Advanced Studies in E-Commerce

Jie Cao

E-Commerce Big Data Mining and Analytics

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

Springer

Advanced Studies in E-Commerce

Advanced Studies in E-Commerce takes a fresh and global viewpoint to E-Commerce development. It encompasses such issues as the basic concepts and principles of E-Commerce, the industry chain of E-Commerce, the security management of E-Commerce; the architecture of E-Commerce; the analytics of E-Commerce; and some cutting-edge topics of E-Commerce.

Jie Cao

E-Commerce Big Data Mining and Analytics

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

Springer

Jie Cao
Hefei University of Technology
Hefei, Anhui, China

Advanced Studies in E-Commerce
ISBN 978-981-99-3587-1 ISBN 978-981-99-3588-8 (eBook)
https://doi.org/10.1007/978-981-99-3588-8

Jointly published with Xi'an Jiaotong University Press
The print edition is not for sale in China (Mainland). Customers from China (Mainland) please order the print book from: Xi'an Jiaotong University Press.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

I. Why Write a Book

Currently, big data fusion and business intelligence are widely concerned in current academic and industrial application fields, and building business intelligence applications based on big data is of great importance in government governance decisions and enterprise business strategies.

Big data for business is a highly practical field with many theoretical directions at the same time. The books on big data in the current market generally focus on big data management in business and rarely introduce all the processes covered by big data in detail. For this reason, this book adopts a step-by-step approach and discusses the background, core technologies and application scenarios of big data in depth, which helps readers deepen their understanding of the knowledge and technologies related to big data in business.

Through this book, readers can understand the specific process of big data for business from scratch and quickly choose the appropriate technology to solve the practical problems in big data for business. Further, by studying this book, readers can narrow the gap in big data integration and business intelligence system construction and provide some necessary guidance for the implementation of business intelligence in industry and government governance.

II. Organization of the Book

This book introduces the basic knowledge and mainstream technologies of big data for business, learning these knowledge and technologies will play a key role in designing and carrying out big data for business applications for practical applications of big data for e-commerce, and at the same time, this book provides a starting point for domestic scholars who are interested in the research work of big data mining and application areas for e-commerce, helping them to understand the current status of

research in this field, grasp the key issues and become familiar with the basic. The book also provides a starting point for domestic scholars who are interested in the research of big data mining and application areas of e-commerce. As an emerging subfield of big data application, it is necessary to understand the necessary process of big data mining for business, and reading this book after understanding data mining and big data analysis will make twice the effort.

We start the discussion of business big data mining-related technologies from Chap. 2. Chapter 2 introduces the knowledge of the underlying collection of business big data, including data types of business big data, collection schemes of business big data, online and offline big data collection schemes, business big data collection cases, etc. These schemes are the basis of business big data mining.

Chapters 3 and 4 discuss business big data pre-processing and business big data storage strategies, respectively. The former discusses the core techniques of business big data pre-processing, including techniques of business big data processing, including data collection, data cleaning, data transformation, data integration and data statute, as well as redundant heterogeneous big data inconsistency elimination and semantic extraction and analysis in business big data. The latter mainly constructs different storage strategies according to different big data types, specifically including key–value storage, column family storage, graph storage and other core contents, and introduces in detail the core tools and core algorithms of three kinds of unstructured data storage and related databases, etc.

Chapter 5 introduces business big data security management technology, mainly discusses the traceability technology, privacy protection, sharing mechanism, blockchain technology and security management cases of business big data and provides related fundamental knowledge.

Chapter 6 addresses the knowledge representation problem of commerce big data and discusses the research work on multi-granularity e-commerce entity construction model, multi-category e-commerce entity relationship extraction, multi-level knowledge representation model and commerce big data knowledge representation cases.

Chapter 7 starts from the relationship of commerce big data knowledge system to reveal the core technologies of commerce big data knowledge fusion and then introduces semantic extraction and semantic association, user portrait construction, knowledge graph construction, knowledge inference and interpretable analysis, and also briefly introduces specific cases of commerce big data knowledge fusion.

Chapter 8 introduces commonly used models for big data management and decision making in commerce, including multi-task learning, recommendation algorithm, community hiding algorithm, purchase prediction model, recommendation based on probability matrix decomposition, indoor location technique, graph K-mean clustering algorithm, combined with the analysis of several e-commerce applications, in order to provide reference for e-commerce big data developers in the actual system design process.

Chapter 9 summarizes the applications related to big data management and decision making in commerce, and looks at the directions worthy of further exploration in the field of big data in commerce, with a view to inspiring readers to carry out more in-depth research work.

III. Target Readers

- Readers who have data mining skills and are interested in big data for business.
- This book can be used as a professional book for senior undergraduate or graduate students in computer science, software engineering, big data, e-commerce and their related majors in colleges of higher education.

IV. Study Suggestions

In order to better read and understand this book, it is recommended that readers learn some simple programming in Python for big data and have some basic knowledge of data mining. The chapters in this book are arranged according to the specific process of business big data processing, and readers are recommended to read them in order from front to back when dealing with business big data specifically.

Hefei, China Jie Cao
March 2023

Acknowledgements

I am grateful to the National Natural Science Foundation of China (NSFC, under Grant Nos. 92046026, 72172057) and the Second Division of Management of NSFC for their support to this project, and to Springer Nature for their suggestions and support to this book. We would like to express our gratitude to Mr. Youquan Wang, Mr. Haicheng Tao, Mr. Guixiang Zhu, Ms. Lei Chen, Mr. Song Li, Mr. Jiangnan Tang, Mr. Xinhao Wang, Ms. Jinjin Cao and Mr. Jiawei Miao from the School of Information Engineering, Nanjing University of Finance and Economics for their participation in writing this book.

Finally, I owe so much gratitude to several of my friends who helped me brainstorm ideas for the book. Due to the author's level and experience, and the shortage of time, omissions are inevitable, so we are grateful for readers' criticism and correction.

Contents

1 Introduction 1
1.1 Overview of Business Big Data Mining and Applications 1
1.2 Big Data Infrastructure 3
1.2.1 Infrastructure Layer 3
1.2.2 Big Data Layer 4
1.3 Overview of Big Data Research in Commerce 4
1.3.1 Big Data Fusion 4
1.3.2 Knowledge Fusion 5
1.3.3 Trajectory Big Data Mining 6
1.3.4 Knowledge Graphs 7
1.3.5 User Portraits 9
1.3.6 E-Commerce Recommendation System 11
References 13

2 Data Collection in the Era of Big Data 19
2.1 Data Types of Business Big Data 19
2.1.1 Structured Data 19
2.1.2 Semi-structured Data 20
2.1.3 Unstructured Data 21
2.2 Online Business Big Data Collection Solution 22
2.2.1 Enterprise Data Collection 22
2.2.2 Web Crawler Data 23
2.2.3 Mobile Device Data 24
2.2.4 Database Data Collection 25
2.3 Offline Business Big Data Collection Solution 26
2.3.1 Physical Data Collection 26
2.3.2 Activity Data Collection 27
2.4 Cases of Business Big Data Collection 27
2.4.1 Precise User Portrait Description 27
2.4.2 Social Platform User Description 28

3 Pre-processing Big Data for Business 29
3.1 Business Big Data Pre-processing Techniques 29
3.1.1 Data Acquisition 29
3.1.2 Data Cleaning 30
3.1.3 Data Transformation 33
3.1.4 Data Integration 34
3.1.5 Data Imputation 35
3.2 Inconsistency Elimination Strategies for Multi-source Heterogeneous Commerce Big Data 36
3.3 Semantic Extraction and Analysis of Business Big Data 38
3.3.1 What Is Semantics 38
3.3.2 Semantic Analysis in Big Data 38
3.4 Business Big Data Pre-processing Case 39

4 Big Data Database for Business 41
4.1 Key-Value Store 41
4.1.1 Background to the Development of Key-Value Store 42
4.1.2 Key-Value Database Versus Relational Databases 42
4.1.3 Key Value Database Advantages 44
4.1.4 Redis 45
4.2 Column Family Store 55
4.2.1 Column Family Database Storage Structure 55
4.2.2 Column Family Database Features 57
4.2.3 HBase 58
4.3 Graph Store 63
4.3.1 The Concept of a Graph 64
4.3.2 Property Graph 64
4.3.3 Graph Database 66
4.3.4 Neo4j 70

5 Security Management on Big Data of Business 75
5.1 Traceability Technology of Business Big Data 76
5.1.1 The Definition of Data Traceability 76
5.1.2 The Definition of PROV 76
5.1.3 The Constraint of PROV Traceability Graph 79
5.2 Privacy Protection of Business Big Data 79
5.2.1 Data Desensitization Technology 80
5.2.2 Differential Privacy Protection 81
5.2.3 K-anonymity 82
5.3 The Data Sharing of Commercial Big Data 82
5.3.1 Access Control 84
5.3.2 Zero Trust Architecture 85
5.3.3 Attribute Based Encryption 86
5.3.4 Homomorphic Encryption 87
5.4 Blockchain Technology 88
5.4.1 Peer to Peer Network 88

5.4.2 Digital Signature ... 89
5.4.3 Hash Function ... 90
5.4.4 SPV Lightweight Verification and Melkel Hash Tree ... 91
5.4.5 Application of Blockchain in Business Big Data ... 91
5.5 Business Big Data Management Case ... 93
5.5.1 Demand Analysis ... 94
5.5.2 Network Architecture Design ... 95
5.5.3 Data Storage Design ... 95

6 Big Commerce Data Knowledge Representation ... 99
6.1 Multi-granularity E-Commerce Entity Construction Model ... 99
6.1.1 Multi-granularity E-Commerce Entity Category ... 99
6.1.2 E-Commerce Entity Recognition ... 101
6.2 Multi-category E-Commerce Entity Relationship Extraction ... 103
6.2.1 Multi-category E-Commerce Entity Relationship Categories ... 103
6.2.2 Multi-category E-Commerce Entity Relationship Extraction Methods ... 104
6.3 Multi-level Knowledge Representation Model ... 106
6.3.1 Knowledge Representation Model Based on Natural Language Processing ... 107
6.3.2 Knowledge Representation Model Based on Relational Network ... 108
6.4 Case Studies of Big Commerce Data Knowledge Representation ... 109
References ... 111

7 Business Big Data Knowledge Fusion ... 113
7.1 Semantic Extraction and Semantic Association ... 113
7.1.1 Subgraph Matching Algorithm for RDF ... 113
7.1.2 Knowledge Graph Keyword Search Algorithm ... 114
7.1.3 Semantic Association Ranking Techniques ... 115
7.2 User Profile Construction ... 115
7.2.1 User Data Collection ... 116
7.2.2 Segmentation of User Groups ... 116
7.2.3 Building a User Profile ... 117
7.2.4 Application of User Profiling ... 117
7.3 Knowledge Graph Construction ... 118
7.3.1 Knowledge Extraction ... 118
7.3.2 Knowledge Integration ... 119
7.3.3 Knowledge Storage and Graph Database Neo4j ... 119
7.4 Knowledge Reasoning and Interpretability ... 120
7.4.1 Knowledge Discovery and Reasoning ... 120
7.4.2 Rule-Based Knowledge Reasoning ... 121
7.4.3 Graph-Based Knowledge Reasoning ... 121
7.4.4 Neural Network-Based Knowledge Inference ... 122
7.4.5 Interpretability Analysis of Knowledge Reasoning ... 122

7.5 Business Big Data Knowledge Fusion Case 123
7.5.1 Introduction to Knowledge Fusion Tools 123
7.5.2 Technical Challenges of Knowledge Fusion 123
7.5.3 A Classic Case of Business Big Data Knowledge Fusion 124

8 Common Business Big Data Management and Decision Model 125
8.1 Robust Multi-task Learning for Clustering 125
8.1.1 Background 125
8.1.2 Problem Formalization 127
8.1.3 Cluster Multitasking Learning Based on Representative Tasks 127
8.2 Recommendations that Integrate User Interests 128
8.2.1 Background 128
8.2.2 Related to the Definition 129
8.2.3 Modeling Endogenous and Exogenous Interests of Users 131
8.2.4 Modeling Missing Data 132
8.2.5 A Recommendation Model that Incorporates User Interests 132
8.3 A Multi-objective Reinforcement Learning Framework for Community Deception 133
8.3.1 Introduction to Community Hiding Algorithms 133
8.3.2 Community Hiding Based on Multi-objective Reinforcement Learning 134
8.4 Mining of Periodic Coactive Populations in Trajectory Data 138
8.4.1 Background 138
8.4.2 Problem Formalization 140
8.4.3 Mining Algorithm for Periodic Populations in Trajectory Data 142
8.5 A Purchase Prediction Method Based on Semi-supervised Multi-view Learning 146
8.5.1 Feature Construction of co-EM-LR Model 146
8.5.2 Online Travel Customer Segmentation 152
8.5.3 Analysis of Online Travel Purchasing Patterns 153
8.5.4 Structure of the co-EM-LR Model 158
8.6 Recommendation Based on Probabilistic Matrix Decomposition and Feature Fusion 159
8.6.1 Application Scenarios of the PMF-MAI Model 159
8.6.2 Feature Construction of PMF-MAI Model 160
8.6.3 Structure of the PMF-MAI Model 162
8.7 Indoor Positioning Technology Based on Asynchronous Sensor 163
8.7.1 Indoor Positioning Technology Background 163
8.7.2 Asynchronous Sensing Method 164

8.7.3 Indoor Area Location Method for Asynchronous Sensing Data 165
8.8 Graph K-means Algorithm Based on Leader Recognition, Dynamic Game and Viewpoint Evolution 169
8.8.1 Study Scenarios, Motivations, and Meanings 170
8.8.2 Basic Knowledge and Problem Definition 171
8.8.3 Specific Framework 172
8.8.4 Experiment 175
8.8.5 Conclusion 179

9 Application of Business Big Data Management and Decision Making 181
9.1 Malicious User Fraud Detection 181
9.1.1 Malicious User Comment Detection 182
9.1.2 Recommended System Support Attack Detection 184
9.1.3 Credit Card Fraud Detection 185
9.2 Online Purchase Decision Model 186
9.2.1 Purchase Prediction Model 186
9.2.2 Personalized Recommendation Model 187
9.2.3 Sales Forecasting Model 189
9.3 Related Applications of Tourism E-Commerce 190
9.3.1 Point of Interest POI and Travel Package Recommendation 191
9.3.2 Travel Itinerary Planning 192
9.4 Business Applications of Location-Based Services 193
9.4.1 APP Takeaway Food 193
9.4.2 Car-Hailing Route Planning 194
9.4.3 Restaurant, Hotel and Gas Station Recommendation Based on Location Service 195
References 196

About the Author

Jie Cao, born in 1969, received his Ph.D. degree in engineering from Southeast University in 2002, is the Dean of School of Information Engineering of Nanjing University of Finance and Economics, Ph.D. supervisor, part-time doctoral supervisor of Nanjing University of Science and Technology and Hohai University, Director of National International Joint Research Center of E-Commerce Information Processing, Director of National Local Joint Engineering Laboratory of E-Commerce Transaction Technology and Member of Teaching Steering Committee of E-Commerce of Ministry of Education. He is also a member of the Teaching Steering Committee of E-Commerce of the Ministry of Education. In the past five years, he has published more than 100 papers in *TKDE*, *TPDS*, *TKDD*, *TWeb*, *TCyb*, *TNNLS*, *TII*, *TIST*, *InfSci*, *KDD*, *ICDM* and other domestic and international journals and conferences, and the total number of citations exceeds 1500, among which more than 70 are indexed by SCI. He has obtained 20 authorized invention patents. He has presided over more than 10 national projects such as the key support projects of National Natural Science Foundation of China, National Natural Science Foundation of China, National Science and Technology Support Program and National Soft Science. He received the second prize of Science and Technology Progress of Ministry of Education in 2015 and the second prize of Science and Technology Progress Award of Jiangsu Province in 2018. He was appointed as an editorial board member of international journals *Neurocomputing* and *World Wide Web Journal*.

Chapter 1
Introduction

1.1 Overview of Business Big Data Mining and Applications

Business intelligence is a collection of software, technologies, and methods to discover valuable laws and patterns from data, transform data into knowledge, and support enterprises' decision making, marketing, and services [1]. With the rapid development of big data, e-commerce is gradually transitioning to the era of big data, i.e., the new generation of e-commerce. According to eMarketer, a leading global market research firm, global e-commerce sales will be $4.938 trillion in 2021, and according to its forecast predicts that global e-commerce sales will reach $5.542 trillion in 2022, accounting for more than one-fifth of total retail sales. Such rapid growth brings broad prospects for the development of a new generation of e-commerce industry, which means a strong market and broad user demand. In the era of big data, e-commerce platforms and resident merchants face new opportunities and challenges on how to exploit the value of data using artificial intelligence and blockchain technology to gain competitive advantages in technology, innovation, and business models [2].

Data is like the blood of business intelligence system, and in the era of Internet+ and Smart+, e-commerce is now widely used in traditional wholesale and retail fields, in addition to the influence of the current epidemic and electronic payment, e-commerce + smart logistics has become the current mainstream model. Due to the order volume as well as the traditional retail enterprises have increased their development efforts in e-commerce websites, enterprises have accumulated a large amount of multi-source heterogeneous data, which directly leads to the widespread existence of data silos, data chimneys and management fragmentation. Specifically, the following aspects of commerce big data analysis are in urgent need of research.

First, the multi-source heterogeneous big data storage and computing infrastructure support strategy, and the semantic modeling of big data in business management are yet to be proposed as complete solutions. Although large enterprises or organizations (such as Amazon, Google, Apache, Taobao, etc.) at home and abroad have launched many open source tools for distributed storage and computation, the

J. Cao, *E-Commerce Big Data Mining and Analytics*, Advanced Studies in E-Commerce,
https://doi.org/10.1007/978-981-99-3588-8_1

semantic modeling, storage model, and computation model of big data are closely related to mining and analysis algorithms and decision-making applications. Before using the existing open source tools, we need to study the storage and computation support strategies and the semantic modeling methods for unstructured data.

Second, the basic problems of data sharing, data ownership, data inconsistency elimination, data missing and sparsity of multi-source heterogeneous multi-modal business big data need to be studied in depth. The data supporting e-commerce analysis and decision-making platform contains user behavior data, user transaction data, mobile data, commodity information, and social platform data, constituting a veritable multi-source heterogeneous multimodal big data. On the one hand, due to data privacy and business security considerations, it is difficult to break the data barriers between e-commerce platforms; data form physical islands within each e-commerce platform, making it difficult to realize efficient data circulation and open sharing, thus restricting the value creation of business big data. On the other hand, business big data are mostly heterogeneous and also face a series of serious challenges such as incomplete information, much noise, poor quality, low credibility, semantic ambiguity, cross-modality, etc., which in turn lead to the formation of logical islands of data at the semantic level, and traditional data mining models are unable to conduct deep association analysis.

Third, the adaptability of data mining methods to big data and to business-specific scenarios requires research efforts. It is easy to lose a large amount of user behavior information by simply obtaining users' opinions and emotions about offline experience from User Generated Content (UGC) text. Global Positioning System (GPS) trajectory data and urban intelligence services, it is difficult to directly migrate to trajectory data mining in a closed and small area, and the adaptability of the method to big data needs to be improved.

Fourth, the knowledge fragmentation in business big data is serious, and the method of multi-channel knowledge fusion needs to be studied systematically. The fusion of multi-channel fragmented knowledge of business big data and the formation of a unified knowledge navigation path is the key issue of whether the intelligence of business big data can be significantly improved, and at present, the granularity, mode and specific algorithm of knowledge fusion in this field are all in a state of great scarcity and need to be studied in depth.

Fifth, the diversified and personalized needs of users make business big data face the challenges of mining users' intentions in multiple marketing scenarios, portraying the behavioral characteristics of users' visits throughout their life cycle and providing interpretable marketing models. The needs of the new generation of business users are becoming more and more diversified and personalized, and the experience scenarios are becoming more and more highly segmented and complex, while the diversification and speed of product design of elemental enterprises obviously lags behind the speed of decentralization of user needs. How to mine the hidden main behavioral intention of users from the spatio-temporal scenes and user access records, model the personalized differences of users in time and space, and improve the performance of mobile recommendation system in the business field has become the core issue of business big data marketing and decision making.

As the application of big data in the new generation of e-commerce, commerce big data also has the characteristics of large data volume, many kinds of data, low value density and fast processing speed. At the same time, business data in the era of big data has the typical characteristics of multiple sources, heterogeneity, and high fragmentation, and its analysis and decision-making are also facing the needs of real-time decision-making, personalized services, diversified decision-making, etc. It is evident that the development of business intelligence is returning to the data at the root of business intelligence [3]. E-commerce platforms have speed, agility, and resilience issues compared to entities in operations. Speed refers to the speed of an organization from strategy planning, deployment, and execution; agility represents the efficiency of e-commerce providers in responding to user needs, i.e., the efficiency of e-commerce providers in identifying user pain points and translating this motivation into strategies; and finally, elasticity, i.e., the ability to adjust to a variety of different e-commerce scenarios. This book focuses on scientific issues in the intersection of big data mining and e-commerce, and its research direction itself represents the development trend of the subject area, with distinctive field characteristics and significant practical application value. This book belongs to the interdisciplinary research of information discipline and management discipline, focusing on big data-driven business data management and decision-making, governance mechanism of big data resources, big data analysis methods and supporting technologies.

1.2 Big Data Infrastructure

The infrastructure layer mainly contains the hardware and software architecture of the system and the underlying services such as security and backup, while the big data layer mainly contains data acquisition, pre-processing, storage and computing support platform, computing models and services.

1.2.1 Infrastructure Layer

- Host and storage system: the underlying hardware, using a hybrid storage architecture of x86 servers, internal server storage and centralized storage, connected via Gigabit/10 Gb/IB Ethernet switches.
- Operating system: using an open source Linux operating system.
- Virtualization platform: using Docker-based lightweight virtualization technology to support flexible deployment and elastic expansion of business, and renting public cloud services to form a hybrid cloud platform together with the internal virtualization platform.
- Security management: managing users, roles and other privileges to secure the system.

- Backup system: through a variety of backup means to ensure the normal operation of the system and the ability to respond well to unexpected situations.

1.2.2 Big Data Layer

- Data acquisition: using a distributed crawler system based on Scrapy, mainly for acquiring data external to the enterprise.
- Data pre-processing: pre-processing of internal and external multi-source heterogeneous data such as format conversion, garbage filtering, correlation and monitoring statistics to ensure the availability of data.
- Big data support platform: the big data storage part provides the function of distributed storage and calculation of data to realize the unified storage and processing of different types of data. Big data computing part provides multiple computing engines including full-text retrieval engine, distributed computing engine, memory computing engine and stream computing engine to meet multimodal computing requirements.
- Model layer: the underlying storage and computing capabilities are encapsulated into various typical data mining tasks to provide data retrieval, statistics, analysis, mining and visualization services to the outside world to realize the analysis and processing of data.

1.3 Overview of Big Data Research in Commerce

This section reviews and analyzes the current state of research in six areas closely related to the mining and application of big data for commerce, including big data fusion, knowledge fusion, trajectory big data mining, knowledge graph, user profiling, and e-commerce recommendation systems.

1.3.1 Big Data Fusion

Big data fusion refers to the use of computer technology to integrate and analyze multiple sources of information under certain guidelines in order to achieve classification tasks for different applications [4]. Traditional big data fusion methods are mainly implemented based on probabilistic models, evidential reasoning, and knowledge-based approaches [5]. The introduction of probability distributions or density functions to deal with data fusion problems [6] allows expressing dependencies between random variables and establishing relationships between different data sets. Probability-based data fusion methods include Bayesian inference, state-space models, Markov models, confidence propagation, maximum likelihood, rough set

theory, and least squares estimation-based methods, but probabilistic fusion algorithms mainly have the following drawbacks: (i) it is difficult to obtain density functions and define prior probabilities; (ii) limited performance when dealing with complex and multivariate data; and (iii) they cannot handle uncertainty. Evidential inference, on the other hand, is generally implemented based on D-S theory and recursive arithmetic, which has the advantages of representing under-informed ignorance and aggregating confidence when collecting new evidence compared to Bayesian inference [7, 8], which introduces the concepts of confidence and likelihood to represent uncertainty in the real world, enabling inference to be performed in dynamic situations. Confidence indicates the extent to which a given piece of evidence supports an event, and likelihood indicates the degree of confidence that a given piece of evidence fails to refute an event. Knowledge-based approaches use machine learning to obtain data fusion results from less consistent data and generally include intelligent aggregation methods, machine learning, fuzzy logic, etc.

Due to the rise of techniques of deep learning, current big data fusion is gradually shifting to multimodal fusion, which mainly includes aligning ontologies and patterns [9], identifying identical entities [10], link relationship prediction [11], and merging conflicting data [12]. Commonly used data-level fusion methods usually use averaging weighting methods, feature matching methods, and pyramid algorithms. The weighted averaging party performs linear weighted averaging of the various data acquired, which is relatively short and fast and can suppress noise well, but its fusion results are less comparable. The feature matching method, on the other hand, establishes the spatial transformation relationship of various modalities and then passes the mapping relationship in space, such as subspace mapping [13] and association matrix [14]. The pyramid method usually constructs multiple tower-level structures for fusing multimodal information for image video analysis [15], and its process is mostly based on algorithms such as convolutional neural networks and attention mechanisms in deep learning to extract information from the bottom to the top layers of data at different levels.

For now, the first challenge still facing big data fusion is the selection of the fused dataset, and the fusion result of the data is closely related to the selected dataset. In addition, multimodal data needs further research, what modality is selected to help obtain the best fusion effect, and how to fuse effectively in multiple e-commerce scenarios deserves further in-depth study.

1.3.2 Knowledge Fusion

Knowledge fusion has been one of the hot topics in the field of data mining and machine learning. The existing research ideas can be broadly categorized into two types: (i) local mining is used in each data source to obtain local patterns, which are fused together after global combinatorial learning to form a globally consistent pattern [16]. This type of approach uses two stages with independent models, the advantage of which is that the local mining model can use existing mature models

and the researcher only needs to work on the model design of the global combined learning stage. The fusion of multiple classifiers [17], and consistent clustering [18] essentially follow the two-stage idea. (ii) Directly using multiple sources of data, fusion is performed from the beginning of model training, and then learning under a unified objective function to obtain probabilities on single or multiple tokens. Models such as multi-example learning [19], multi-tagging learning [20], and multi-example multi-tagging learning [21, 22], which combine both together, fall under the category of model fusion.

While Big Data brings many challenging problems to data mining, it also brings many challenges to the important problem of knowledge fusion. From the model level, the massive quantification of samples makes the imbalance between labeled and unlabeled samples even greater, and there is an urgent need to incorporate partially supervised learning [23] into the learning process of knowledge fusion. literature [24] made a preliminary attempt on the malicious user identification problem by integrating data from two different channels, with partially supervised learning, and proposed a hybrid learning framework using Bayesian inference. A generalization framework that can integrate more channels, multiple examples, multiple labels, and partially supervised learning is worth further exploration. In addition, the huge number of unlabeled samples involved in training makes knowledge fusion learning extremely demanding in terms of algorithmic efficiency, and it is necessary to shift from focusing on model accuracy to considering the balance between accuracy and efficiency in previous studies.

1.3.3 Trajectory Big Data Mining

Trajectory can be regarded as the imprint left by mobile objects in space with the change of time. In recent years, with the rapid development of infrastructure such as global positioning system and wireless communication network and the wide application of handheld and vehicle-mounted wireless communication and positioning devices, especially the application of location check-in, location sharing and location identification of many mobile social networks, a large amount of trajectory data is increasingly accumulated in daily life and supports different types of location-based services (Local-Based Service (LBS)). In fact, trajectory data is one of the most important data in people's daily life, which records users' activities in real-world environment, and these activities reflect users' intention, interest, experience and behavior pattern to some extent. For example, a user's trajectory often appears in sports venues, indicating that the user may be interested in fitness activities; the trajectory between work and life places in a user's daily life largely reflects his or her habits; more fine-grained trajectory analysis can even determine the user's preference in a specific situation based on the type of locations frequented by the user and the contextual information at that time.

In terms of pre-processing of trajectory data, there are a series of data quality problems in trajectory data, mainly including: inaccurate data caused by positioning

device and physical environment; part of data missing caused by equipment, transmission failure or mis-operation; inconsistent data caused by different coordinate representation update strategies and context transformation; data redundancy caused by partial export and backup of trajectory data, etc. Generally speaking, the preprocessing of trajectory data mainly includes Noise Filtering, Stay Point Detection, Trajectory Segmentation, Trajectory Compression and Map Matching, etc. Noise filtering aims to remove noisy points or outliers from trajectories. Existing methods mainly filter noise from the perspective of a single trajectory.

In terms of trajectory pattern mining, the existing trajectory knowledge extraction work is mainly carried out from the perspective of trajectory-based data mining, including Frequent Patterns mining, Trajectory Clustering, Moving Together Patterns mining and Periodical Patterns mining. Periodical Patterns mining, etc. Frequent pattern mining aims to discover temporal patterns from large-scale trajectories, such as the common paths of more than a certain number of objects traveling in a given time interval, which is of great value for destination prediction, path recommendation, and behavior understanding. At present, research in temporal frequent patterns has been relatively well developed, and several mature algorithms including GSP [25], PrefixSpan [26], Span [27], and Spade [28] have been proposed and applied. However, different from the traditional time series, the trajectory data include location dimension, time dimension and semantic dimension [29], so simply using traditional sequence mining methods cannot effectively solve the problem of frequent pattern mining of spatio-temporal trajectories.

In terms of trajectory semantic annotation, traditional trajectory mining research mainly focuses on the extraction of spatio-temporal features of trajectories, often starting from the trajectory data itself for bottom-up mining analysis, and one-sidedly emphasizing the formalization of computational models, resulting in ineffective use of information. Therefore, the effective integration of spatio-temporal information and domain knowledge is an important way to promote the continued development of trajectory mining research. Trajectory semantic annotation aims to semantically enrich the original trajectory data by using spatio-temporal information and domain knowledge, which essentially belongs to the trajectory classification problem, i.e., to distinguish different types of trajectories based on characteristics such as behavior and traffic mode. Relevant literature has designed special algorithms for semantic annotation of different types of spatial objects, mainly including: Enrichment with geographic regions, Enrichment with geographic lines and Enrichment with geographic points. geographic points, etc.

1.3.4 Knowledge Graphs

Knowledge graphs originated from various structured knowledge bases, such as WordNet for linguistic knowledge [30] and Freebase for world knowledge [31], where a lot of effort was spent on organizing human knowledge into structured knowledge systems. Google further proposed the concept of knowledge graphs [32]

on the basis of knowledge bases by representing entities (including concepts and attribute values) as graph nodes, and the connected edges between nodes correspond to the association relationships between entities to characterize the acquired knowledge in a networked structure, and its core content includes three parts: knowledge acquisition, knowledge representation, and knowledge inference.

Knowledge acquisition tasks are divided into three categories, namely knowledge graph complementation [33], relationship extraction [34], and entity discovery [35]. The first one extends the existing knowledge graph and the other two are discovering new knowledge (also known as relations and entities) from the text. Knowledge graph complementation can be classified into the following categories: embedding-based ranking, relational path inference, rule-based inference, and meta-relational learning. Initial studies on knowledge graph complementation focused on triadic prediction methods for low-dimensional embeddings [36]. However, most of them failed to capture multi-hop relations. Therefore, recent work has turned to exploring multi-hop relational paths and combining them with logical rule implementations, referred to as relational path inference and rule-based inference, respectively [37]. Triad classification is a related task of knowledge graph complementation, which evaluates the correctness of factual triad classification [38, 39]. Entity recognition or named entity recognition is a task of tagging entities in text when it focuses on human-defined features of specific named entities, such as capitalization patterns and language-specific resources, such as place names, and has applications in many literatures [40]. Entity types include coarse-grained types and fine-grained types, the latter using tree-structured type categories that are often considered as multi-class, multi-label classifications. Entity discovery includes entity identification, disambiguation, keying and alignment [41]. Relationship extraction is a key task to extract unknown relationship facts from plain text and add them to the knowledge graph, enabling automatic generation of large-scale knowledge graphs. Due to the lack of labeled relational data, weakly supervised or self-supervised creates training data by assuming that sentences containing the same entities can express the same relationships under the supervision of a relational database using heuristic matching [42]. Relational extraction models utilize attention mechanisms, graph convolutional networks, adversarial training, reinforcement learning, deep residual learning, and transfer learning.

Knowledge representation learning is an important research topic in knowledge graphs, which paves the way for many knowledge inference tasks and subsequent applications [43]. Knowledge representation learning models can be divided into four areas: representation space, scoring functions, encoding models, and auxiliary information. The representation space, on the other hand, investigates what dimensionality to represent relations and entities as [44]. The existing literature mainly uses real-valued point state spaces, including vector spaces, matrix spaces, and tensor spaces, while other types of spaces, such as complex vector spaces, Gaussian spaces, and manifold spaces, are also used [45]. Scoring functions are used to measure the reasonableness of facts and are also referred to as energy functions in energy-based learning frameworks. Scoring functions are generally classified into distance-based scoring functions and similarity matching-based scoring functions [46]. Current research on knowledge representation has focused on encoding models, including linear/

bilinear models, factorization, and neural networks [47]. Linear models represent relations as linear/bilinear mappings by projecting head entities into a representation space close to tail entities; factorization aims to decompose relational data into low-rank matrices for representation learning; neural networks encode relational data with nonlinear neural activation and more complex network structures. Multimodal embedding combines external information such as textual descriptions, type constraints, relational paths and visual information with the knowledge graph [48] to facilitate more efficient representation of knowledge, so auxiliary information considers textual, visual and type information.

Knowledge inference mainly inferred new knowledge or identified incorrect knowledge based on existing knowledge, and completed deep analysis and inference of data. The main technical means contain four categories: (i) rule-based inference, which applies simple rules or statistical features on the knowledge graph [49, 50]; (ii) distributed representation-based inference, which learns fact tuples in the knowledge graph through a representation model to obtain a low-dimensional vector representation of the knowledge graph. Then, the inference predictions are transformed into simple vector operations based on the representation model, including transfer-based [51–54], tensor/matrix decomposition-based [55–57], and spatial distribution-based [58, 59], and other multi-class methods; (iii) neural network-based inference, where neural networks are used to directly model the knowledge graph fact tuples to obtain vector representations of fact tuple elements for further inference. This class of methods is still a score function-based method, which is different from other methods in that the whole network constitutes a score function and the output of the neural network is the score value. Representative inference frameworks include NTN [60], ProjE [61], etc.; (iv) hybrid inference, by mixing multiple inference methods to make full use of the advantages of different methods, such as high accuracy of rule-based inference [62], strong computational power of distributed representation-based inference [63], and strong learning and generalization capabilities of neural network-based inference [64, 65]. In the commerce vertical, the mining of mobile commerce points of interest is an important aspect. Using knowledge graphs, on the one hand, a complete knowledge system description of mobile commerce points of interest can be formed through multiple data sources linked, so as to better recognize, understand, and analyze the characteristic advantages of the points of interest. On the other hand, knowledge graph itself is a kind of relationship network based on graph structure, and based on this graph structure can help people identify the potential risk factors existing in complex relationships more effectively.

1.3.5 User Portraits

The concept of user portrait (Persona) was first proposed by Cooper, the father of interaction design [66], who believed that a user portrait should be a virtual representation of a real user in real life, a target user model built on a series of real available data. The model is based on the social attributes and various types of

behaviors of users, and abstracts one or more labels of information. Many scholars have subsequently refined this concept continuously. The literature [67] describes the user portrait through a narrative form, considering relevant information about individuals, such as favorite items, disliked items, occupation and other dimensions, thus making the user portrait fuller. The literature [68] argues that a series of relevant attributes can be predefined according to different application scenarios, and methods such as machine learning and data mining can be used to learn the attribute weights and thus digitize the user portrait. The literature [69] constructs similarity between users of different platforms based on locally sensitive hashing techniques from a multi-platform perspective, and thus fuses multi-platform user portraits. The literature [70] points out that in the field of mobile e-commerce, the construction of user profiles needs to take into account relevant social network information in order to describe user profiles more accurately and ensure the privacy of users. In a nutshell, the core work of constructing user profiles is to determine the corresponding set of tags using machine learning models or relevant rules for different needs, such as user interest profiles, age and gender profiles, crowd profiles, address profiles, life cycle profiles, etc. Building user portraits can help us better understand users and products in personalized recommendation and sorting, user refinement operation, product analysis, and assisted decision making.

From the perspective of social psychology [71], personal likes, retweets, and favorites of content on e-commerce platforms can more intuitively reflect personal recognition of products, which has a supporting role in portraying user consumption portraits. The comments of individuals mostly contain some emotional expressions based on consumption experiences, with positive and negative directions, which have a strong interference effect on capturing the consumption preferences of online users. At present, the mainstream methods used for text sentiment mining are two types of supervised learning and unsupervised learning. The basic idea of supervised learning is to learn a model by training samples with sentiment polarity annotations and use the trained model to classify the sentiment of the test text. For example, in the literature [72], three classifiers were used to classify the text sentiment of movie review data, and the study showed that the accuracy of text sentiment classification based on machine learning could reach 80%, which is a milestone. In addition, a large number of scholars have proposed the use of deep learning for sentiment classification of short texts, such as sentiment-oriented word embedding methods [73], convolutional neural network models [74], and deep confidence network models [75]. Considering the high cost of manual tagging and the inferior quality of machine tagging, unsupervised learning models have started to attract attention in recent years. For example, literature [76] proposed a WordNet-based sentiment lexicon construction method, which first selected sentiment words with known sentiment polarity as seed words, and then iteratively performed synonymy or antonymy search to continuously expand the sentiment lexicon; literature [77] combined supervised multi-label topic model and implicit sentiment topic model to classify social emotions; literature [78] proposed a topic-adaptive semi-supervised model for microblog sentiment analysis. In addition, literature [79] proposed a model to obtain user data from social networks,

mobile networks, and wearable devices, so as to build a user profile based on multi-source heterogeneous data to describe user preferences more comprehensively and precisely.

1.3.6 E-Commerce Recommendation System

1. **E-commerce recommendation system**

With the rapid development of Internet technology and the widespread use of modern e-commerce, users are confronted with the vast amount of information available on the Internet, and it is difficult to meet their individual needs by relying on search engines alone [80]. Given the explosion of available information on the Web, users usually receive a myriad of products, movies, or restaurants. Therefore, personalization is an essential strategy to promote a better user experience. In summary, these systems play a crucial and indispensable role in the various information access systems that facilitate business development and contribute to the decision-making process, and are widely available in numerous web domains such as e-commerce and/or media sites [81, 82].

In general, recommendation lists are generated based on user preferences, product characteristics, previous user/product interactions, and some other additional information such as temporal (e.g., sequence-aware recommendation algorithms [83]) and spatial (e.g., Point-of-Interesting, POI) data. Recommendation models are mainly classified into collaborative filtering, content-based recommendation systems, and hybrid recommendation systems based on input data types [84]. And these recommender systems usually need to solve two problems: how to recall some relevant goods and how to personalize the recommendations according to the user's preferences.

The generation of candidate items is a very big challenge, which requires a recall process to select hundreds of items with strong purchase intent from a large number (billions) of items. Currently, industry solutions focus on product recall through product similarity, but they do not take into account the preferences of individual users and the attributes of the products. For this reason, the literature [85] proposes an attribute-based collaborative filtering approach to achieve interpretable recommendations with guaranteed accuracy. In recent years, there are more and more articles combining knowledge graphs with recommendation systems, and the introduction of external knowledge can easily improve the effectiveness of the whole algorithm from the data level. In the literature [86], a relational graph network with adaptive target behaviors is designed to portray multiple user behaviors and fuse knowledge graphs for efficient recommendation. In the e-commerce domain, especially in practical recommendation scenarios, goods can usually be divided into different domains, such as books and movies. Although they belong to different domains, they can reflect users' preferences in a more consistent way. The literature [87] learns the synergy between different domains through attention mechanism. Also, review information is

introduced to further enhance the user's representation, thus improving the accuracy of recommendations. Recommendations based on multiple views as well as multiple tasks have also received great attention recently. The literature [88] models' auxiliary information from many different domains as a multi-view model, and proposes a multi-view alignment approach by fully mining information from multiple aspects, which better integrates the internal information of a single view and the cross information of multiple views. The literature [89] proposes an algorithm based on graph neural networks to learn product representations from multiple views from both user perspective and entity perspective, and then perform product recommendation.

2. **Deep learning-based recommendation system**

Deep learning is currently evolving rapidly and is currently enjoying great success in many application areas such as computer vision and speech recognition. Academia and industry have been competing to apply deep learning to a wider range of application domains because of its ability to solve many complex tasks while providing state-of-the-art results [90]. In recent years, deep learning has dramatically changed the recommendation architecture, bringing more ways to improve the performance of recommendation systems (e.g., completeness, accuracy, etc.). Deep learning-based recommendation systems have gained wide attention in recent years because they overcome the shortcomings of traditional recommendation models and obtain high recommendation quality. Deep learning can effectively capture nonlinear user/product relationships and can encode more complex abstractions as higher-level data representations [91]. Furthermore, it captures complex relationships in the data itself from a large number of accessible data sources such as context, text, and visual information [92].

In order to provide a better illustration of deep learning-based recommendation methods, existing models can be classified according to the type of deep learning technique used, which is divided into two main categories as follows.

- Neural module-based recommendation methods: in this category, models are divided into nine subcategories based on the eight deep learning models mentioned earlier: multi-layer perceptron (MLP) [93], auto-encoding (AE) [94], Convolutional Neural Networks (CNN) [95], Recurrent Neural Network (RNN) [96], Restricted Boltzmann Machine (RBM) [97], Neural Autoregressive Distribution Estimation (NADE) [98], Attention Mechanism (AM) [99], Adversary Network (AN) [100], and Deep Reinforcement Learning (DRL) [101] for recommender systems. The deep learning technique used determines the applicability of the recommendation model. For example, MLP can easily model nonlinear interactions between users and items; CNNs can extract local and global representations from heterogeneous data sources such as text and visual information; and RNNs enable recommender systems to model the temporal dynamics and sequential evolution of content information.
- Recommendation algorithms based on deep hybrid models: some deep learning-based recommendation models use more than one deep learning technique. The flexibility of deep neural networks makes it possible to combine multiple neural

network modules to form a more powerful hybrid model, e.g., recurrent neural networks combined with convolutional neural networks for user sequence recommendations [102], recurrent neural networks combined with attention models for Taobao e-commerce recommendations [103], combining reinforcement learning with attention models for user long-term preference recommendations [104], etc.

Usually, in order to achieve accurate recommendations an in-depth understanding of the characteristics of the product and the actual needs and preferences of the user is required [105]. Of course, this can be achieved by using a large amount of available auxiliary information. For example, contextualizing information to tailor services and products to the user's environment and context, and mitigating the effects of cold starts [99]. While implicit feedback reflects the implicit intent of the user and is easy to collect, collecting explicit feedback is a resource demanding task. Although existing research work has synthesized the effectiveness of deep learning models in mining user and item description information [106], implicit feedback [107], contextual information [108], and review text [109] for recommendations, these algorithms are not currently utilizing these different forms of auxiliary information in a comprehensive manner for management and decision making. In addition, un-interpretability is a common problem of current deep learning-based recommendation algorithms. Therefore, proposing interpretable recommendation algorithms is currently a daunting task.

References

1. Chen G, Wei Q, Zhang J (2014) Principles and methods of business intelligence, 2nd edn. Electronics Industry Press, Beijing (in Chinese)
2. Feng Z, Guo X, Zeng D et al (2013) On the research frontiers of business management in the context of big data. J Manage Sci China 16(1):1–9 (in Chinese)
3. Chaudhuri S, Dayal U, Narasayya V (2011) An overview of business intelligence technology. Commun ACM 54(8):88–98
4. Meng X, Du Z (2016) Research on the big data fusion: issues and challenges. Comput Res Dev 53(2):231–246 (in Chinese)
5. Ding W, Jing X, Yan Z et al (2019) A survey on data fusion in internet of things: towards secure and privacy-preserving fusion. Inf Fus 51:129–144
6. Pansiot J, Stoyanov D, McIlwraith D et al (2007) Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems. In: 4th international workshop on wearable and implantable body sensor networks, pp 208–212
7. Panigrahi S, Kundu A, Sural S et al (2009) Credit card fraud detection: a fusion approach using Dempster–Shafer theory and Bayesian learning. Inf Fus 10(4):354–363
8. Murphy RR (1998) Dempster–Shafer theory for sensor fusion in autonomous mobile robots. IEEE Trans Robot Autom 14(2):197–206
9. Maedche A, Staab S (2002) Ontology learning for the semantic web. Springer Nature, Switzerland AG
10. Shen W, Han J, Wang J et al (2018) SHINE+: a general framework for domain specific entity linking with heterogeneous information networks. IEEE Trans Knowl Data Eng 30(2):353–366
11. Liu M, Chen L, Liu B et al (2017) DBpedia-based entity linking via greedy search and adjusted Monte Carlo random walk. ACM Trans Inf Syst 36(2):1–34. Article 16

12. Xiao H, Gao J, Li Q et al (2019) Towards confidence interval estimation in truth discovery. IEEE Trans Knowl Data Eng 31(3):575–588
13. Li Z, Liu J, Tang J et al (2015) Robust structured subspace learning for data representation. IEEE Trans Pattern Anal Mach Intell 37(10):2085–2098
14. Wu L, Jin R, Jain AK (2013) Tag completion for image retrieval. IEEE Trans Pattern Anal Mach Intell 35(3):716–727
15. Pu Y, Gan Z, Ren C et al (2017) Variational autoencoder for deep learning of images, labels and captions. In: Proceedings of advances in neural information processing systems (NIPS 2017), pp 2352–2360
16. Wu X, Zhang S (2003) Synthesizing high-frequency rules from different data sources. IEEE Trans Knowl Data Eng 15(2):353–367
17. Avidan S (2007) Ensemble tracking. IEEE Trans Pattern Anal Mach Intell 29(2):261–271
18. Wu J, Liu H, Xiong H et al (2015) K-means-based consensus clustering: a unified view. IEEE Trans Knowl Data Eng 27(1):155–169
19. Ray S, Scott S, Blockeel H (2011) Multi-instance learning. In: Encyclopedia of machine learning. Springer US, pp 701–710
20. Zhang M, Zhou Z (2007) ML-KNN: a lazy learning approach to multi-label learning. Pattern Recogn 40(7):2038–2048
21. Zhou Z, Zhang M, Huang S et al (2012) Multi-instance multi-label learning. Artif Intell 176(1):2291–2320
22. Surdeanu M, Tibshirani J, Nallapati R et al (2012) Multi-instance multi-label learning for relation extraction. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP'12), pp 455–465
23. Liu B, Lee W, Yu P et al (2002) Partially supervised classification of text documents. In: Proceedings of international conference on machine learning (ICML'02), pp 387–394
24. Wu Z, Wang Y, Wang Y et al (2015) Product review spammer detection: a hybrid learning model. In: Proceedings of 2015 IEEE international conference of data mining (ICDM'15), pp 1039–1044
25. Srikant R, Agrawal R (1996) Mining sequential patterns: generalizations and performance improvements. In: Proceedings of 5th international conference of extending database technology (EDBT 1996), pp 3–17
26. Pei J, Han J, Mortazavi-Asl B et al (2001) PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: Proceedings of 17th international conference on data engineering (ICDE 2001), pp 215–224
27. Ayres J, Flannick J, Gehrke J et al (2002) Sequential pattern mining using a bitmap representation. In: Proceedings of 8th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2002), pp 429–435
28. Zaki M (2001) SPADE: an efficient algorithm for mining frequent sequences. Mach Learn J 42(1–2):31–60
29. Zhang C, Han J, Shou L et al (2014) Splitter: mining fine-grained sequential patterns in semantic trajectories. Proc VLDB Endow 7(9):769–780
30. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
31. Bollacker K, Evans C, Paritosh P et al (2008) FreeBase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on management of data (SIGMOD 2008), pp 1247–1250
32. Dong X, Gabrilovich E, Heitz G et al (2014) Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining (SIGKDD 2014), pp 601–610
33. Trouillon T, Dance CR, Gaussier É et al (2017) Knowledge graph completion via complex tensor factorization. J Mach Learn Res 18(1):4735–4772
34. Geng Z, Chen G, Han Y et al (2020) Semantic relation extraction using sequential and tree-structured LSTM with attention. Inf Sci 509:183–192

35. Shi C, Ding J, Cao X et al (2020) Entity set expansion in knowledge graph: a heterogeneous information network perspective. Front Comput Sci 15(1):1–12
36. Vu T, Nguyen TD, Nguyen DQ et al (2019) A capsule network-based embedding model for knowledge graph completion and search personalization. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), pp 2180–2189
37. Che F, Zhang D, Tao J et al (2020) Regarding neural network parameters as relation embeddings for knowledge graph completion. In: AAAI, pp 2774–2781
38. He G, Li J, Zhao WX et al (2020) Mining implicit entity preference from user-item interaction data for knowledge graph completion via adversarial learning. In: Proceedings of the web conference 2020, pp 740–751
39. Akrami F, Saeef MS, Zhang Q et al (2020) Realistic re-evaluation of knowledge graph completion methods: an experimental study. In: Proceedings of the 2020 ACM SIGMOD international conference on management of data (SIGMOD 2020), pp 1995–2010
40. Li M, Lin Y, Hoover J et al (2019) Multilingual entity, relation, event and human value extraction. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations), pp 110–115
41. Wu T, Qi G, Li C et al (2018) A survey of techniques for constructing Chinese knowledge graphs and their applications. Sustainability 10(9):3245
42. Mao Y, Zhao T, Kan A et al (2020) Octet: online catalog taxonomy enrichment with self-supervision. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2247–2257
43. Tang X, Chen L, Cui J et al (2019) Knowledge representation learning with entity descriptions, hierarchical types, and textual relations. Inf Process Manage 56(3):809–822
44. Pham DH, Le AC (2018) Learning multiple layers of knowledge representation for aspect based sentiment analysis. Data Knowl Eng 114:26–39
45. Paulius D, Sun Y (2019) A survey of knowledge representation in service robotics. Robot Auton Syst 118:13–30
46. Kumarasinghe K, Kasabov N, Taylor D (2020) Deep learning and deep knowledge representation in spiking neural networks for brain-computer interfaces. Neural Netw 121:169–185
47. Huo Y, Wong DF, Ni LM et al (2020) HeTROPY: explainable learning diagnostics via heterogeneous maximum-entropy and multi-spatial knowledge representation. Knowl-Based Syst 207:106389
48. Huang Z, Xu X, Ni J et al (2019) Multimodal representation learning for recommendation in Internet of Things. IEEE Internet Things J 6(6):10675–10685
49. Wang WY, Mazaitis K, Lao N et al (2015) Efficient inference and learning in a large knowledge base. Mach Learn 100(1):101–126
50. Cohen WW (2016) TensorLog: a differentiable deductive database. arXiv preprint arXiv:1605.06523
51. Wang Z, Zhang J, Feng J et al (2014) Knowledge graph embedding by translating on hyperplanes. In: Proceedings of twenty-eighth AAAI conference on artificial intelligence (AAAI 2014), pp 1112–1119
52. Bordes A, Usunier N, Garcia-Duran A et al (2013) Translating embeddings for modeling multi-relational data. In: Proceedings of advances in neural information processing systems (NIPS 2013), pp 2787–2795
53. Wen J, Li J, Mao Y et al (2016) On the representation and embedding of knowledge bases beyond binary relations. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence (IJCAI 2016), pp 1300–1307
54. Ji G, Liu K, He S et al (2016) Knowledge graph completion with adaptive sparse transfer matrix. In: Proceedings of thirtieth AAAI conference on artificial intelligence (AAAI 2016), pp 985–961
55. Nickel M, Tresp V, Kriegel HP (2011) A three-way model for collective learning on multi-relational data. In: Proceedings of machine leaning (ICML 2011), pp 809–816

56. Chang KW, Yih W, Yang B et al (2014) Typed tensor decomposition of knowledge bases for relation extraction. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP 2014), pp 1568–1579
57. Nickel M, Jiang X, Tresp V (2014) Reducing the rank in relational factorization models by including observable patterns. In: Proceedings of advances in neural information processing systems (NIPS 2014), pp 1179–1187
58. Xiao H, Huang M, Zhu X (2016) From one point to a manifold: knowledge graph embedding for precise link prediction. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence (IJCAI 2016), pp 1315–1321
59. Nickel M, Rosasco L, Poggio T (2016) Holographic embeddings of knowledge graphs. In: Proceedings of thirtieth AAAI conference on artificial intelligence (AAAI 2016), pp 1955–1961
60. Socher R, Chen D, Manning CD et al (2013) Reasoning with neural tensor networks for knowledge base completion. In: Proceedings of advances in neural information processing systems (NIPS 2013), pp 926–934
61. Shi B, Weninger T (2017) ProjE: embedding projection for knowledge graph completion. In: Proceedings of thirty-first AAAI conference on artificial intelligence (AAAI 2017), pp 1236–1262
62. Han X, Sun L (2016) Context-sensitive inference rule discovery: a graph-based method. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: technical papers (COLING 2016), pp 2902–2911
63. Wang Q, Wang B, Guo L (2015) Knowledge base completion using embeddings and rules. In: Proceedings of twenty-fourth international joint conference on artificial intelligence (IJCAI 2015), pp 1859–1865
64. Toutanova K, Chen D, Pantel P et al (2015) Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP 2015), pp 1499–1509
65. Xie R, Liu Z, Jia J et al (2016) Representation learning of knowledge graphs with entity descriptions. In: Proceedings of thirtieth AAAI conference on artificial intelligence (AAAI 2016), pp 2659–2665
66. Cooper A (2004) The inmates are running the asylum: why high-tech products drive us crazy and how to restore the sanity. Sams, Indianapolis
67. Grudin J, Pruitt J (2002) Personas, participatory design and product development: an infrastructure for engagement. In: Proceedings of participatory design conferences (PDC 2002), pp 144–152
68. Lester JC, Converse SA, Kahler SE et al (1997) The persona effect: affective impact of animated pedagogical agents. In: Proceedings of the ACM SIGCHI conference on human factors in computing systems (CHI 1997), pp 359–366
69. Sharma V, Dyreson C (2018) LINKSOCIAL: linking user profiles across multiple social media platforms. In: Proceeding of 2018 IEEE international conference on big knowledge (ICBK), pp 260–267
70. Garcia-Davalos A, Garcia-Duque J (2020) User profile modelling based on mobile phone sensing and call logs. In: Information technology and systems. ICITS 2020. Advances in intelligent systems and computing, pp 243–254
71. Ellison NB, Steinfield C, Lampe C (2007) The benefits of Facebook "friends:" social capital and college students' use of online social network sites. J Comput-Mediat Commun 12(4):1143–1168
72. Pang B, Lillian L, Shivakumar V (2002) Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP 2002), pp 79–86
73. Tang D, Wei F, Yang N et al (2014) Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (ACL 2014), pp 1555–1565

74. dos Santos CN, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of the 25th international conference on computational linguistics (COLING 2014), pp 69–78
75. Zhou S, Chen Q, Wang X (2014) Active semi-supervised learning method with hybrid deep belief networks. PLoS ONE 9(9):e107122
76. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining (SIGKDD2004), pp 168–177
77. Rao Y, Li Q, Mao X et al (2014) Sentiment topic models for social emotion mining. Inf Sci 266:90–100
78. Liu S, Cheng X, Li F et al (2015) TASC: topic-adaptive sentiment classification on dynamic tweets. IEEE Trans Knowl Data Eng 27(6):1696–1709
79. Musto C, Polignano M, Semeraro G et al (2020) MYRROR: a platform for holistic user modeling. User Model User-Adap Inter 30:477–511
80. Lu J, Wu D, Mao M et al (2015) Recommender system application developments: a survey. Decis Support Syst 74:12–32
81. Qian Y, Zhang Y, Ma X et al (2019) EARS: emotion-aware recommender system based on hybrid information fusion. Inf Fus 46:141–146
82. García-Sánchez F, Colomo-Palacios R, Valencia-García R (2020) A social-semantic recommender system for advertisements. Inf Process Manage 57(2):102153
83. Singh VP, Pandey MK, Singh PS et al (2020) Neural net time series forecasting framework for time-aware web services recommendation. Procedia Comput Sci 171:1313–1322
84. Miller BN, Konstan JA, Riedl J (2004) PocketLens: toward a personal recommender system. ACM Trans Inf Syst (TOIS) 22(3):437–476
85. Chen T, Yin H, Ye G et al (2020) Try this instead: personalized and interpretable substitute recommendation. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval (SIGIR'20), pp 891–900
86. Feng Y, Hu B, Lv F et al (2020) ATBRG: adaptive target-behavior relational graph network for effective recommendation. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval (SIGIR 2020), pp 2231–2240
87. Zhao C, Li C, Xiao R et al (2020) CATN: cross-domain recommendation for cold-start users via aspect transfer network. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval (SIGIR 2020), pp 229–238
88. Wang M, Lin Y, Lin G et al (2020) M2GRL: a multi-task multi-view graph representation learning framework for web-scale recommender systems. In: Proceedings of the 26th ACM SIGKDD conference on knowledge discovery and data mining (SIGKDD 2020), pp 2349–2358
89. Tai C, Wu M, Chu Y et al (2020) MVIN: learning multiview items for recommendation. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval (SIGIR 2020), pp 99–108
90. Goodfellow I, Bengio Y, Courville A et al (2016) Deep learning. MIT Press, Cambridge
91. Zhang S, Yao L, Sun A et al (2019) Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv (CSUR) 52(1):1–38
92. Karatzoglou A, Hidasi B (2017) Deep learning for recommender systems. In: Proceedings of the eleventh ACM conference on recommender systems (SIGKDD 2017), pp 396–397
93. Xu Z, Chen C, Lukasiewicz T et al (2016) Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling. In: Proceedings of the 25th ACM international on conference on information and knowledge management (CIKM 2016), pp 1921–1924
94. Wu W, Zhao J, Zhang C et al (2017) Improving performance of tensor-based context-aware recommenders using bias tensor factorization with context feature auto-encoding. Knowl-Based Syst 128:71–77
95. Tuan TX, Phuong TM (2017) 3D convolutional networks for session-based recommendation with content features. In: Proceedings of the eleventh ACM conference on recommender systems (RecSys 2017), pp 138–146

96. Wu C, Wang J, Liu J et al (2016) Recurrent neural network based recommendation for time heterogeneous feedback. Knowl-Based Syst 109:90–103
97. Pujahari A, Sisodia DS (2019) Modeling side information in preference relation based restricted Boltzmann machine for recommender systems. Inf Sci 490:126–145
98. Zheng Y, Tang B, Ding W et al (2016) A neural autoregressive approach to collaborative filtering. In: Proceedings of the 33rd international conference on international conference on machine learning (ICML 2016), pp 764–773
99. Hu B, Shi C, Zhao W et al (2018) Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining (SIGKDD 2018), pp 1531–1540
100. Tang J, Du X, He X et al (2019) Adversarial training towards robust multimedia recommender system. IEEE Trans Knowl Data Eng 32(5):855–867
101. Zheng G, Zhang F, Zheng Z et al (2018) DRN: a deep reinforcement learning framework for news recommendation. In: Proceedings of the 2018 world wide web conference (WWW 2018), pp 167–176
102. Xu C, Zhao P, Liu Y et al (2019) Recurrent convolutional neural network for sequential recommendation. In: The world wide web conference (WWW 2019), pp 3398–3404
103. Lv F, Jin T, Yu C et al (2019) SDM: sequential deep matching model for online large-scale recommender system. In: Proceedings of the 28th ACM international conference on information and knowledge management (CIKM 2019), pp 2635–2643
104. Zou L, Xia L, Ding Z et al (2019) Reinforcement learning to optimize long-term user engagement in recommender systems. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (SIGKDD 2019), pp 2810–2818
105. Chen L, Wu Z, Cao J et al (2020) Travel recommendation via fusing multi-auxiliary information into matrix factorization. ACM Trans Intell Syst Technol (TIST) 11(2):1–24
106. Xue F, He X, Wang X et al (2019) Deep item-based collaborative filtering for top-n recommendation. ACM Trans Inf Syst (TOIS) 37(3):1–25
107. Yi B, Shen X, Liu H et al (2019) Deep matrix factorization with implicit feedback embedding for recommendation system. IEEE Trans Ind Inf 15(8):4591–4601
108. Ren Y, Tomko M, Salim FD et al (2017) A location-query-browse graph for contextual recommendation. IEEE Trans Knowl Data Eng 30(2):204–218
109. Huang C, Jiang W, Wu J et al (2020) Personalized review recommendation based on users' aspect sentiment. ACM Trans Internet Technol (TOIT) 20(4):1–26

Chapter 2
Data Collection in the Era of Big Data

2.1 Data Types of Business Big Data

Big data includes a variety of different formats and different types of data. According to whether the data has a certain pattern, structure and relationship, the data structure of business big data can be divided into structured data, semi-structured data and unstructured data. With the gradual diversification of data sources, unstructured data has become a major part of the data. According to IDC's survey report: 80% of the data in the enterprise is unstructured data, and the data grows exponentially by 60% every year.

2.1.1 Structured Data

Structured data refers to data that follows standard schemas and structures, mainly uses relational databases to represent and store, and uses two-dimensional table structure logic to express and implement data. Structured data is structured first, and then data is generated. The general characteristics of structured data are: data is in units of behavior; one row of data represents one entity information and each row of data has the same attributes. Due to the relatively mature development of relational databases, the development of structured data storage and analysis methods is also relatively comprehensive. There are a large number of tools to support structured data analysis. The analysis methods are mainly statistical analysis and data mining. The relational database is created based on the relational model. The relational model is a two-dimensional table model. A relational database includes some two-dimensional tables and there is a certain relationship between these tables. Table 2.1 is an example to describe how structured data is represented and stored in the database.

Table 2.1 shows that the storage and arrangement of structured data has certain regularity, and operations such as query and modification are relatively simple. However, structured data has problems such as poor scalability. For example, the

J. Cao, *E-Commerce Big Data Mining and Analytics*, Advanced Studies in E-Commerce,
https://doi.org/10.1007/978-981-99-3588-8_2

Table 2.1 An example of structured data

Id	Name	Age	Gender
1	Sherry	12	Female
…	…	…	…

fields are not fixed and it is difficult to repeatedly change the table structure in practical applications. Therefore, it is very difficult to use relational databases, and it is also easy for the back-end interface to read data from databased incorrectly.

2.1.2 Semi-structured Data

Semi-structured data is a form of structured data, which does not conform to the data model structure associated with relational databases or other data table forms which have certain structure and are composed of fixed structured schema data. It is not inherently relational, and is a data type between fully structured data and fully unstructured data. Semi-structured data contains related markup that separates semantic elements and stratifies records and fields, so semi-structured data is also called self-describing structures, such as storing data in tree or graph data structures. Unlike structured data, which has structure first and data later, semi-structured data has data first and then structure. Common semi-structured data types include xml files and *json* files. Taking *json* files as an example, the storage method of semi-structured data is described in detail (Fig. 2.1).

```
{
  "data": [
            {“title”:”University_of_Notre_Dame”,

             “paragraph”=” Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend \"Venite Ad Me Omnes\". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.”

            }
           {“title”:”…”,
            “paragraph”=”…”}
   ]
}
```

Fig. 2.1 An example of *json* file

As can be seen from the above example, the order of attributes in semi-structured data is not important, and the number of attributes in different semi-structured data is not necessarily the same. More information, including self-describing information (metadata), can be flexibly stored by the data format of semi-structured data. Therefore, the scalability of semi-structured data is better.

2.1.3 Unstructured Data

In the modern world of big data, unstructured data is the most abundant, and this part of the data accounts for up to 80% of enterprise data, and the growth rate is faster. Compared with structured data (that is, the data stored with the logical expression of the two-dimensional table structure), the data that is inconvenient to be represented by the two-dimensional logical table of the database is unstructured data. Unstructured data has no fixed organization principle of unfiltered information, also commonly referred to as raw data. Common examples include web logs, text documents, images, video, and audio files.

Unstructured WEB database is mainly generated for unstructured data. Compared with the popular relational database in the past, the biggest difference is that it breaks through the limitation of relational database structure and fixed length of data, and supports repeated fields, subsections, etc. Fields and variable-length fields and realizes the processing of variable-length data, repeated fields and the variable-length storage management of data items. It has a tradition in processing continuous information (including full-text information) and unstructured information (including various multimedia information which is an unmatched advantage of relational databases). Unstructured data is often stored in relational databases as a whole, in the form of binary large objects (BLOBs, which store binary data as a collection of a single entity), or in non-relational databases such as NoSQL databases.

However, unstructured data is more difficult to be understood by computers, cannot be directly processed or queried with SQL statements, and is also difficult to organize and format. Therefore, the collection, processing and analysis of unstructured data is a big challenge. Table 2.2 are common unstructured data types.

Table 2.2 The common unstructured data types

Text file	Data types
Rich media	Media and entertainment data, surveillance data, geospatial data, audio, weather data, and more
IoT data	Sensor data, ticker data, etc.
Mobile data	SMS, location, etc.
Scientific data	Oil and gas exploration, space exploration, seismic imagery, atmospheric data, and more
…	…

Unstructured data has very diverse formats and standards, and technically unstructured information is more difficult to standardize and understand than structured information. Therefore, storage, retrieval, publication and utilization require more intelligent IT technologies, such as mass storage, intelligent retrieval, knowledge mining, content protection, and value-added development and utilization of information.

2.2 Online Business Big Data Collection Solution

The most basic and important link in the process of data analysis and mining is data collection, which refers to the related technologies and implementation processes of capturing, processing and sending specific user behaviors or events. The technical essence of data collection is to first monitor the events in the running process of the software application, judge and capture when the events that need attention occur. Whether the data collection is abundant, the collected data is accurate, and whether the collection is timely will affect the effect of data analysis. Therefore, how to choose the correct data collection method is very important for data analysis.

At present, the collection of business big data is mainly divided into two collection schemes: online and offline. This section will focus on the online collection scheme of business big data.

2.2.1 Enterprise Data Collection

Commercial enterprises have a large scale of development, have been operating for a long time, and have accumulated a large amount of original data. Collecting enterprise data can be used to better understand consumers' business behavior information.

The data of the Internet mainly comes from network equipment such as Internet users and servers, mainly a large amount of text data, social data and multimedia data, etc., while enterprise data mainly comes from machine equipment data, enterprise informatization data and industrial chain related data.

From the perspective of the type of data collection, it should not only cover basic data, but also include semi-structured user behavior data, meshed social relationship data, and text or audio user opinion and feedback data which mainly include the following:

- **Massive key-value data**: Today, with the rapid development of sensor technology, different types of enterprise sensors including photoelectric and thermal sensors have been widely used in the field.
- **Document data**: Includes a large amount of traditional engineering data such as simulation data.

- **Interface data**: Data of the interface type provided by the established enterprise automation or information system, including txt file, JSON format, XML format, etc.
- **Video, image and audio data**: Various data collected by the device.

The main technical difficulties of enterprise data collection mainly come from the huge amount of data and the non-standard protocol of enterprise data.

The huge amount of enterprise data brings great challenges to data processing. Data specification and cleaning are important tasks in data processing. A large amount of enterprise data is "dirty" data, and direct storage cannot be used for analysis. For processing, massive data has higher technical requirements.

Internet data collection is generally a common HTTP protocol, but in the enterprise field, there will be various types of enterprise protocols such as ModBus, O PC, CAN, ControlNet, and various automation equipment manufacturers and integrators will develop various private enterprises. Agreement, resulting in great difficulty in the interconnection of enterprise agreements. When developers implement comprehensive automation and other projects at the enterprise site, they encounter the problem that they cannot effectively analyze and collect in the face of many enterprise agreements.

2.2.2 Web Crawler Data

With the increasing amount of network data, it is becoming more and more difficult to obtain static data by simply browsing the web. How to effectively extract and utilize information has become a huge challenge. A web crawler, also known as a web spider, is a web robot used to automatically browse the World Wide Web. out and store it as a unified local data file in a structured way.

Web crawler tools are basically divided into the following three categories:

– Distributed web crawler tools, such as Nutch, etc.
– Java web crawler tools, such as C rawler4j, WebMagic, WebCollector.
– Non-Java web crawler tools, such as Scrapy (based on python).

Web crawlers can automatically collect all accessible page content to provide data sources for collection engines and big data analysis. In terms of functions, the crawler has three functions: data collection, processing and storage. Common web crawler strategies include depth-first strategy (DFS) and breadth-first strategy (BFS).

The depth-first strategy means that the crawler starts from a certain URL and crawls link by link until all the lines in a link are processed before switching to other lines.

The basic idea of the breadth-first strategy is to insert the links found in the newly downloaded webpage into the end of the URL queue to be crawled, that is, the web crawler will first crawl all the webpages linked in the starting webpage, and then select the linked webpages. Continue crawling all pages in this web link.

2.2.3 Mobile Device Data

Compared with enterprise data and web crawler data, mobile device data is closer to daily life. With the development of mobile Internet and big data technology and the popularization of smartphones, almost all work, study and life scenarios are inseparable from mobile phones. Mobile APPs have replaced the traditional way of life, allowing people to experience convenient and efficient. Of course, it also carries a lot of rich information. Collecting the data of these APPs, cleaning and analyzing the data in a centralized manner can turn these massive data into valuable data energy.

Different from the PC side, for mobile devices such as mobile phones, iPads, smart watches, TV boxes, etc., the carrier of our mobile device data collection is the APP. The framework of mobile data collection SDK mainly consists of three parts: user interface, business module and control module. Native SDK needs to invest a lot of development resources in multi-language support, and cross-platform application development is gradually becoming a trend, but the implementation of JS SDK in each framework is also different. difficult to support. More difficult is the model adaptation of Android devices. Due to the open source features of the Android system, various manufacturers have made targeted ROM improvements in order to have a better user experience on various models. Especially in recent years, Android has made great changes in virtual machines and compilers. This brings more difficulty to the model adaptation.

Taking Google Analytics for Firebase as an example, we will introduce the collection process of mobile device data.

The types of information that Google Analytics for Firebase collects by default include: number of users and sessions, session duration, operating system, device model, geographic location, first launches, app opens, app updates, and in-app purchases frequency.

In terms of device identification, the Firebase SDK library uses the app instance identifier to identify each unique installed instance of the app. When using the SDK, application instance identifiers are generated at the application level, and by default, the Firebase SDK collects identifiers (such as Android Advertising ID and IOS Advertising Identifier) for mobile devices and uses a technology similar to cookies. On iOS devices, the SDK collects advertising identifiers. To ensure that the Advertising Identifier (IDFA) is available, developers need to associate the AdSupport.framework library. If there is no advertising identifier, the SDK collects the vendor identifier. The SDK will stop collecting vendor identifiers if, after reporting vendor identifiers, there are advertising identifiers that can be collected. By default, the SDK collects advertising IDs on Android devices.

How to protect customer data privacy in the process of mobile device data collection is a major challenge in mobile device data collection. Thus, the object of protecting customer privacy when collecting mobile device data needs to attract the attention of the majority of developers.

2.2.4 Database Data Collection

In the process of managing various information work, a large amount of data is generated or required, and the database can effectively store and manage increasingly important information. Traditional enterprises will use traditional relational databases such as MySQL and Oracle to store data. With the advent of the era of big data, NoSQL databases such as Redis, MongoDB, and HBase are also commonly used in data collection. Enterprises complete big data collection by deploying a large number of databases on the collection side, and performing load balancing and sharing among these databases.

Taking the NoSQL database as an example, we will understand the data collection process of the database. NoSQL (Not only SQL) non-relational database, is suitable for ultra-large-scale data storage, storage of unstructured data, to make up for the deficiencies of relational databases that are not good at writing large amounts of data, and has the characteristics of non-relational and distributed. NoSQL is divided into 4 types: key-value databases, columnar databases, document databases, and graph databases. Due to the complete consistency of the key (key), the key-value database can quickly find the corresponding value through the hash value of the key. According to the data storage method, it can be divided into temporary, permanent and both. Access load, common key values include Redis and Memcached, etc. Columnar databases store data in columns, which are convenient for storing structured and semi-structured data. Common columnar databases include HBase and BigTable. Document database, data is stored in the form of documents, usually in json or xml format, common document databases include CouchDB, MongoDB, etc. The graph database uses the graph structure to store data, and uses the graph structure related algorithm, which is suitable for systems with complex relationships such as social networks.

Google's MapReduce is a classic case of NoSQL database application. Google used cheap commodity hardware to generate a search index, and scaling the data across multiple servers required two stages—Map and Reduce. The data processing is shown in Fig. 2.2.

As shown in Fig. 2.2, the map function stage, also known as the map operation, extracts data from the input data and converts the results into key-value pairs, which are then sent to distribution and sorting. Distribute and sort sorts key-value pairs by key and returns the result. The reduce function stage is responsible for sorting, integrating, and summarizing the sorted results before returning them.

Google's extensions to the map and reduce functions make it possible to blow hundreds of millions of web pages and run them cost-effectively and reliably on low-cost commercial CPUs. The application of MapReduce has also inspired engineers in other organizations to create an opensource version of MapReduce, which has broadened new ideas for using functional programming systems.

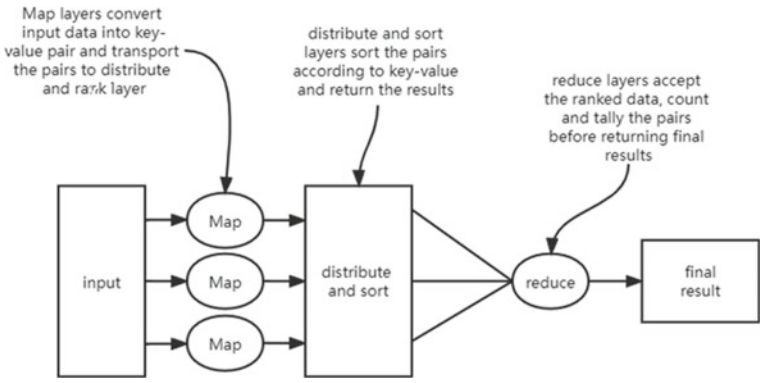


Fig. 2.2 The framework of MapReduce

2.3 Offline Business Big Data Collection Solution

Compared with the virtualization of online data, offline business big data can better reflect the real preferences of users, and is more conducive to comprehensive user analysis, focusing on the value of offline data intelligence. On the one hand, it is due to the digital transformation of the real economy with the demand of digital transformation becoming stronger and stronger, and it must rely on the digitization and data application of offline business scenarios; on the other hand, the data infrastructure of offline scenarios is relatively weak and thus it belongs to the data depression, and the overall market size of the offline real economy far exceeds that of the online, so the application The potential is huge. Retail, marketing, government affairs and other industries are relatively mature application scenarios of offline data intelligence. The offline data collection technology collects people, goods, and scene data. The collection methods include SDK embedded in the APP, wireless signal sensors and cameras. In this section, we will take offline shopping malls as an example to describe the offline business big data collection scheme.

2.3.1 Physical Data Collection

Physical data mainly comes from various information collected by offline sensors. Mainly include Wi-Fi perception technology, human body thermal technology, video portrait recognition technology, etc.

Wi-Fi sensing technology is simple to implement, but it cannot scan devices such as mobile phones with Wi-Fi turned off, and the accuracy of acquisition accuracy is not high. For example, Euclid Zero technology tracks the MAC (media access control layer) addresses of customers' mobile phones in the mall, so as to analyze

the movement route and residence time of specific customers in the store, and use it to improve product display and personalized recommendation services.

Human body thermal technology and video portrait recognition technology need to install cameras and analyze the images in the video. After technical optimization, it can achieve high accuracy, but it can only identify the number of passenger flows, and cannot load more information on passenger flows. A valid connection to the online data could not be established.

2.3.2 Activity Data Collection

The activity data in offline shopping malls is mainly human activity data. In order to obtain more accurate traffic analysis data, retailers can use the various microelectronic systems (MEMS) such as accelerometers and gyroscopes on today's mainstream smartphones. Member app can create an accurate heat map of passenger flow. The NFC membership card handled by shopping mall users is also the best way to track customer behavior. For example, the NFC-enabled membership card developed by FiveStars can grasp the frequency of customer visits, peak hours of visit and information on purchased products. In store, customer flow and customer behavior analysis solutions based on technologies such as 3D sensors and cameras are also human activity data collection solutions. For example, beer brands use Shop perception 3D shopping sensors to track user interactions with products on the shelves, including touching, taking, and putting back. and other actions, and generate a heat map of the commodity.

Compared with online data collection, offline data collection is more difficult, and there are many technical problems that need to be further improved. The difficulty of offline collection lies in the accuracy of data. It is impossible to locate accurate customer information, and it is difficult to form a complete data link. It is difficult to form customized services.

2.4 Cases of Business Big Data Collection

After the data is collected, cleaned and processed, how to further mine the data and apply it to real life is a big proposition. This chapter mainly introduces the application of big data in accurate user portrait description and social platform user description.

2.4.1 Precise User Portrait Description

User portraits are similar to user roles, and are used to outline users (user backgrounds, characteristics, personality tags, behavioral scenarios, etc.) and connect user needs

and product design. Using user portraits label user information, that is, collecting and sharing the data of consumers' social attributes, living habits, consumption behaviors and other main information, companies can perfectly abstract the user's business picture, which is the basic way to apply big data technology in business. User portraits provide enough information for business intelligence, which can help business intelligence quickly find more extensive feedback information such as accurate user groups and user needs.

The data for constructing user portraits comes from all user-related data. For the classification of user-related data, an important classification idea is introduced: closed classification. Such a classification method helps to continuously enumerate and iteratively supplement the missing information dimensions. User data can be roughly divided into two categories: static information data and dynamic information data.

Static information data refers to relatively stable information of users, such as demographic attributes, business attributes and other data. This kind of information is self-labeled. On the basis of real information, there is no need for too many modeling predictions, but more modeling predictions.

Dynamic information data can accurately describe the user's behavior preferences, focusing on user contact points, using offline data and online data to understand user behavior, such as browsing records, search records and click-to-publish information, and how to apply User behavior data to conduct data modeling and analyze user tags is an important part of business big data applications.

2.4.2 Social Platform User Description

With the gradual popularization of smart devices, there is a trend that users' activity on social platforms is gradually greater than that in real life. Therefore, social media has become an unprecedentedly large Internet platform, and hundreds of millions of users' behavior records contain huge scientific and market value. Accurate behavior prediction and detection technology is the core of business big data recommendation and social marketing, and social platform user behavior analysis and modeling is the basis of prediction and detection technology, which has become one of the important issues of computer science.

The existing behavior prediction and detection technologies in social media still have many challenges in the basic links of behavior analysis and modeling, and it is difficult to effectively meet the application requirements of social recommendation systems, personalized search, fraudulent behavior and information manipulation detection. The challenges can be summarized as follows: the high sparsity of adopting information behavior, the multivariate heterogeneity of social media user behavior, the massive and dynamic nature of user behavior data, and the complexity of user behavior intent.

Chapter 3
Pre-processing Big Data for Business

3.1 Business Big Data Pre-processing Techniques

Different types of data can be collected through various methods, and these data want to be used for data mining, its necessary path is data processing, because the collected data is often imperfect, there may be a variety of problems, must be processed through technical means these collected raw data, that is, through the process of data cleaning, data transformation, data integration, data attribution to process the raw data. The data obtained at the data processing stage is the result of preliminary analysis, and the data obtained through subsequent steps such as data mining and visual analysis can be used to assist in decision-making.

3.1.1 Data Acquisition

Data acquisition is the use of some technology or means to collect and store data on some device. Data acquisition is the first step in the life cycle of Big Data, and all subsequent analysis and mining is based on data acquisition. This section will focus on the various methods of data acquisition, the criteria for assessing data quality and the factors that influence it.

Various types of data, including structured, semi-structured and unstructured data, are available through RFID radio frequencies, sensors, social networks, mobile internet, etc. As these data are characterized by large data volumes and heterogeneity, it is important to adopt acquisition methods specifically for big data. This section introduces the following three big data acquisition methods.

1. **System log collection**

Many companies have business platforms that generate large amounts of log data on a daily basis. What log collection systems do is to collect business log data for use in offline and online analysis systems. High availability, high reliability and scalability

J. Cao, *E-Commerce Big Data Mining and Analytics*, Advanced Studies in E-Commerce,
https://doi.org/10.1007/978-981-99-3588-8_3

are the basic characteristics of a log collection system. Currently the commonly used opensource log collection systems are Flume, Scribe, etc.

2. **Network data collection**

Web data collection, i.e. the collection of unstructured data, refers to the acquisition of data information from websites by means of web crawlers or open APIs (Application Programming Interfaces) of websites, for example. This method allows unstructured data to be extracted from web pages and stored in a structured way as a unified local data file. It supports the capture of files or attachments such as images, audio and video, and attachments can be automatically associated with the body of the text. In addition to the content contained in the web, the capture of web traffic can be processed using bandwidth management techniques such as DPI (Deep Message Inspection) or DFI (Deep/Dynamic Flow Inspection).

3. **Database collection**

Some businesses use traditional relational databases such as MySQL, Oracle and SQL Server to store data. In addition to this, NoSQL (non-relational) databases such as Redis and MongoDB are also commonly used for data capture, a situation where a large number of databases are usually deployed on the capture side and load balanced between them.

Data quality is the basis for ensuring the application of data. The raw data collected may have quality issues and needs to be assessed through certain criteria. Data that does not pass the assessment will be dealt with by a range of follow-up methods.

The criteria for assessing data quality can be judged by these 4 aspects. Integrity refers to whether there is any missing information in the data, which may be the absence of the whole data or the absence of information in a particular field in the data. Data integrity is one of the most fundamental assessment criteria for data quality; consistency refers to whether the data follows a uniform specification and whether the logical relationships between the data are correct and complete; accuracy refers to whether the information and data recorded in the data are accurate and whether there are any abnormalities or errors in the information recorded in the data; timeliness refers to the time interval between when the data is generated and when it can be viewed, also known as the delayed duration of the data, which is the degree to which the world is synchronized with the objective world.

3.1.2 Data Cleaning

Problem data can be filtered according to the data quality assessment criteria, and there are three main types of "problem" data: defective data, noisy data and redundant

data. In this section, we will discuss the definition of each of these three types of "dirty data" and explore the cleaning methods for each of them.

1. **Handling of residual data**

Data cleansing is the process of removing "faulty" data and includes three areas: dealing with missing values, dealing with noisy data and dealing with redundant data. This section focuses on the processing of redundant data.

As the name suggests, fragmented data is data that is incomplete. As mentioned above, fragmented data may be missing data as a whole, or it may be missing information from a field in the data. The data can be judged to be incomplete based on the "completeness" of the data quality assessment criteria mentioned above. There are several ways to deal with missing data.

(1) **Ignore the entire tuple**

When an attribute of a tuple is mutilated, the entire tuple is ignored. This method is simple, but has a disadvantage: using the ignore tuple method means that the remaining attribute values of the tuple, which are likely to be necessary to analyse the problem, cannot be used. This method is not very effective unless the tuple has multiple attribute residues. It performs particularly poorly when an attribute has many missing tuples.

(2) **Fill in the residual values**

The content of the missing values can be determined either by manual completion or by setting up a rule. Manual filling is only suitable for small amounts of data with few missing values and may not work when the amount of data is large and many values are missing. The four main methods of manual filling are as follows.

(a) Use global constants to fill in missing values.
(b) Use the mean value of the attribute to fill in the missing values.
(c) Fill in the residual values using the mean values of the attributes of all samples that are in the same class as the tuple with the residual attribute.
(d) Speculate on the most likely value and fill it in: you can use methods such as regression analysis to speculate on the size of that missing value.

2. **Handling of noise data**

Noisy data are possible deviations or errors in the measured value relative to the true value when measuring a variable, which can affect the correctness and effectiveness of subsequent analysis operations. Noisy data includes mainly erroneous data, false data and abnormal data. As the handling of erroneous and spurious data is more complex and involves knowledge of the application area of this data, it will not be

described in this book, which will focus on the handling of abnormal data. Anomalous data are discrete data that have a large impact on the results of data analysis.

(1) **Splitting the box**

Boxing is the process of placing the data to be processed into "boxes" according to certain rules, and using some method to process the data in each box. This book introduces the following three methods of boxing.

(a) Equal depth boxing method: each box has the same number of records and the number of records in each box is called the depth of the box.
(b) Equal-width box splitting method: splitting equally over the entire interval of data values so that each box has an equal interval, this interval is called the width of the box.
(c) User-defined boxing method: boxing is carried out according to user-defined rules.

(2) **Smoothing**

After splitting the boxes, the data in each box is to be smoothed.

(a) By average: averaging the data in the same box and replacing all the data in the box with the mean value.
(b) By median: take the median of all the data in the box and replace all the data in the box with the median.
(c) By boundary value: for each data in the box, the boundary value that is smaller from the boundary value is used instead of all the data in the box.

(3) **Clustering**

The data sets are grouped into clusters, and the values outside the clusters are isolated points. These isolated points are noisy data and should be removed or replaced. Similar or similar data are aggregated together to form clusters, and data outside these clusters are anomalous data.

(4) **Return**

A regression function is constructed by finding a correlation between two related variables that allows the function to satisfy the relationship between the two variables to a greater extent, using this function to smooth the data.

3. **Handling redundant data**

Redundant data includes both duplicate data and data that is not relevant to the problem being analyzed and processed, and is usually dealt with by filtering the data. Repetitive filtering is used for duplicate data and conditional filtering is used for irrelevant data.

(1) **Repeat filter**

On the basis of the known content of the duplicate data, one record from each duplicate is retained and the other duplicates are deleted. Duplicate filtering = identification of duplicate data + filtering operation. Filtering operations can be divided into direct and indirect filtering depending on the complexity of the operation.

Direct filtering: filtering directly on duplicate data, selecting any record to keep and filtering out the rest of the duplicate data.

Indirect filtering: duplicate data is first processed in some way to form a new record before the filtering operation is performed.

(2) **Conditional filtering**

Filtering data based on one or more conditions. A condition is set on one or more attributes, and the records that match the condition are put into the result set, and the data that does not match the condition is filtered out. In effect, duplicate filtering is a form of conditional filtering.

3.1.3 Data Transformation

Data transformations can be divided into attribute type transformations and attribute value transformations, and this section describes both of these in detail.

1. **Property type transformation**

Data transformation is the process of converting data from one representation to another. During data processing, it is often necessary to transform the attributes of the original data into the attribute types of the target data set for the convenience of subsequent work. Attribute transformations can be carried out using methods such as data generalization and attribute construction.

(1) **Data generalization**

Data generalization is the replacement of lower level or raw data with more abstract (higher level) attributes. For example: street attributes can be generalized to the city level, cities can be generalized to the country level, and of course streets can be generalized directly to the country level; age attributes can be generalized to youth, middle age and old age; birth year attributes can be generalized to post-80s, post-90s or post-00s, etc.

(2) **Attribute construction**

Attribute construction refers to constructing a new attribute and adding it to the set of attributes in order to aid mining. This attribute can be an attribute calculated from the original attribute, e.g. a new attribute perimeter and area can be calculated from the radius attribute. In addition, attribute changes can be divided into one-to-one and many-to-one mappings, depending on the mapping relationship between the original attribute and the target attribute.

2. **Attribute value transformation**

Attribute value transformation, or data normalization, refers to the scaling of attribute values so that they fall into a specific interval in order to eliminate the bias in mining

results caused by numerical attributes of varying sizes. There are four main methods of data normalization as follows.

(1) **Maximum-minimum standardisation**

The original range of the known attribute is mapped to the new range. This method is simple, but has the drawback that when the new data added exceeds the original range, the original minimum and original maximum values must be updated, otherwise an error will occur.

(2) **0–1 standardisation**

0–1 normalisation is a special form of max–min normalisation, i.e. the case where the other minimum is 0 and the maximum is 1.

(3) **Zero-mean normalisation**

Suitable for cases where the data fits a normal distribution.

(4) **Standardisation of fractional calibrations**

Normalisation is achieved by moving the position of the decimal point to map the attribute value between [0, 1], using scientific notation for decimals.

3.1.4 Data Integration

Data integration is the logical or physical centralisation of data from different sources, formats and characteristic properties to provide comprehensive data sharing for the enterprise. When integrating data, schema integration and object matching are very important, and how to match equivalent entities from multiple sources of information is the problem of entity identification. When data integration is performed, the same data is repeated several times in the system and data redundancy needs to be eliminated and correlation analysis needs to be performed for different features or relationships between data.

Data integration is the integration of data from different data sources, either logically (generating a view) or physically (generating a new relational table) into a unified data set, on which subsequent analytical processing is carried out.

When integrating data from different data sources, how can pattern matching be achieved and how can equivalent entities from multiple data sources be matched. The essence of the above problem is the entity identification problem. Entity identification is to match real entities from different data sources, e.g. A.user-id = B.customer_ id. Entity identification is usually done based on metadata to avoid errors in schema integration.

For the same real-world entity, data values for the same attribute may differ from system to system, possibly due to different representations of the attribute, different units, etc. For example, the attribute house price uses different currency units in different countries, differences in rating scale between universities, etc. For

conflicting data values, rules for the attribute need to be extracted from the metadata and uniform rules need to be established in the target system to convert the original attribute value to the target attribute value.

In data integration, data redundancy is inevitable: the same attribute uses different field names in different systems, e.g. the same customer ID has the field name Cust_id in system A and Customer_Num in system B; after integration a data attribute can be calculated from other data attributes, e.g. the monthly turnover attribute in system A and the daily turnover attribute in system B. The monthly turnover can be deduced from the daily turnover. Correlation analysis can be used to check the correlation between the attributes and to determine whether there is data redundancy.

3.1.5 Data Imputation

Data imputation has no impact on the subsequent analytical processing, the results of the analytical processing of the data before and after imputation are the same and the time spent on data imputation does not exceed the time saved by data mining after imputation. A necessary prerequisite for data imputation is a good understanding of the mining task and familiarity with the data content. This section focuses on two methods of data imputation: dimensional imputation and numerical imputation.

(1) **Dimensional reversion**

Data subsumption, also known as data reduction, refers to the process of streamlining the maximum amount of data while keeping the data as close to its original form as possible. This section will describe in detail how dimensional normalisation is carried out.

Dimensional normalisation removes unimportant or irrelevant attributes from the original data, or reduces the number of attributes by reorganising them. The aim of dimensional normalisation is to find the smallest subset of attributes whose probability distribution is as close as possible to the probability distribution of the original data set.

(2) **Numerical imputation**

Numerical imputation refers to replacing the original data with a simpler representation of the data, or using smaller units of data, or replacing the data with a data model to reduce the amount of data. Common methods include histograms, representing actual data with clustered data, sampling and parametric regression methods.

3.2 Inconsistency Elimination Strategies for Multi-source Heterogeneous Commerce Big Data

The concept of multi-source heterogeneous information is understood in two main ways.

Multiple sources refer to different sources of information. Information sources can be traditional structured relational database systems and object-oriented database systems, semi-structured XML files, or network information sources with different query interfaces between them. For the integration of heterogeneous information, it is necessary to solve the problem of how to carry out the integrated representation and description of heterogeneous information, and then organise, manage and utilise the information reasonably and effectively according to different application purposes on this basis. The main task of such researchers is to integrate and utilise data from different types of databases, for example. This heterogeneity is reflected in the following areas.

(1) Heterogeneity in computer hardware: the database can run on mainframes, minis, workstations, PCs or embedded systems.
(2) Heterogeneity of operating systems: the more popular operating systems targeted are Unix, Windows, Linux, etc.
(3) The database system itself is also heterogeneous: different database management systems are available such as Oracle, SQL Server, MySQL, etc. Different database data types are available such as relational, reticulated, etc.

Heterogeneity refers to the different compositional structures of information, i.e. semi-structured, structured and unstructured. The main task of this type of researcher is to investigate how to combine information with different structures for analytical research.

Heterogeneous information refers to Internet information in the form of unstructured text and real-time quantitative data in the form of structured, quantitative data. In addition to the completely different forms of information presentation, the sources of information, reliability and the difficulty of deep analysis of information are all different. At present, structured data is more widely used and mature than unstructured information and semi-structured information. Time series analysis based on quantitative information is more maturely developed and has been widely used in the financial industry to assist decision-making. Unstructured information refers to information that exists in a relatively irregular form, often with an irregular internal structure and stored in a variety of forms, such as documents, emails, web pages, video files, comments published by internet users, and so on. There is a wide range of file formats, such as TEXT, HTML, XML, RTF, MSOffice, PDF, PS2/PS, etc. However, the role of unstructured information cannot be ignored, and the archives and expert analysis reports released by the state have a great impact on the industry. At present, the research mainly based on this is still based on quick access to semantic analysis, sentiment analysis, text mining, etc. With the popularity of the Internet in all walks of life, the amount of information available and to be processed by enterprises

and individuals has exploded, and the vast majority of this data is semi-structured as well as unstructured. The means to process these types of data needs to be improved in order to meet user needs for access.

In response to the two notions above, we see two dominant ways of dealing with heterogeneous information.

One way is to integrate and integrate heterogeneous information from different data sources for sharing and processing. The current data sources are in a complex format and processing using this approach is still very difficult and the process is complicated. There are many options for integrating heterogeneous information that have been studied, but the basic approaches can be grouped into two categories.

(1) The warehouse method, where a data warehouse is created and copies of data from various data sources that participate in the integration are stored in the data warehouse. All subsequent operations are carried out directly on the basis of the data warehouse. The advantage is that the data copies are stored in the data warehouse and can be constantly updated in sync with the original data source. The user does not have to access the actual data source when using the data, so access is faster. The disadvantages are obvious: the data is stored more than once, which consumes a lot of storage resources when the volume of data is large enough, and it is more difficult to update and does not synchronise well enough to reflect the information in the data source in a timely manner.
(2) Virtual method, in this form, the data is still stored in each independent data source, the user for the global pattern highlight query, there is a query engine for the user's global query for query parsing, rewriting and decomposition; then each data source according to the query requirements for separate processing, and then use the middleware to pick out the results of each query to integrate and return to the user. It actually uses middleware to process the user's operation, parses the operation and sends it to the relevant data source for execution, and then is responsible for integrating the results of the different data sources and finally returning them to the user. The advantage of this approach is that there is no need to store large amounts of data repeatedly, and the data is updated in a timely manner, making it suitable for businesses with high requirements for real-time data processing.

The other is that the two types of information are processed separately, combining data from two different data sources or in different formats, often with the results from one side to assist the other. Structured data and unstructured data constitute heterogeneous information, and both types of information are processed differently. Structured data analysis methods are relatively more sophisticated. Practical applications are widespread, including stock, financial, trading, and recommendation algorithms. Unstructured data, on the other hand, is still in a state of research and exploration, and some practical cases have emerged, such as the analysis of user evaluation content on Taobao. Therefore, current research tends to focus on the results of structured data analysis, supplemented by the results of unstructured data analysis. Combining the two, there is currently work being done on how to use structured stock data as well as unstructured news media report information to discover financial event hotspots.

3.3 Semantic Extraction and Analysis of Business Big Data

3.3.1 What Is Semantics

In the real world, the meaning of a concept represented by a thing, and its relationship to other concepts, can be thought of as semantics. Semantics is the interpretation of symbols, for example "An apple is a fruit that is rich in minerals and vitamins." This explains the string (symbol, concept) "Deep learning" is the first systematic account of deep learning techniques, written by Ian Goodfellow, Yoshua Bengio and Aaron Courville, published in 2016 describes the meaning of the book (concept) "Deep learning".

The typical features of semantics include objectivity and subjectivity, clarity and vagueness, and domain of semantics. For example, the semantic meaning of "fat" is a vague concept with unclear boundaries, which cannot be defined by simple logic. Domainness refers to the fact that the understanding of the meaning of some words needs to be determined in a specific domain, and there may be different understandings of the same thing in different domains. For example, the word 'apple' has different meanings in the field of fruit and food and in the field of mobile phone communication.

3.3.2 Semantic Analysis in Big Data

A typical characteristic of Big Data is Variety, which has various meanings, the most important of which is the diversity of data types. When representing a book, there can be numeric, date, text and many other forms.

We have represented the two books B1 and B2 using a structured data representation as follows.

B1: (Internet Big Data Processing Techniques and Applications, Zeng Jianping, Tsinghua University Press, 2017, Big Data category).

B2: (The Beauty of Mathematics, Wu Jun, People's Post and Telecommunications Publishing House, 2014, Mathematics).

In big data applications such as book recommendations, one of the most basic problems in deciding which book to recommend to a customer is to calculate the similarity of two books. In the case of books B1 and B2, the similarity between the years of publication 2017 and 2014 is relatively easy to calculate, but for "Big Data" and "Mathematics" it is not possible to calculate accurately simply by relying on strings. The text-based ones such as "Big Data Processing Techniques and Applications on the Internet" and "The Beauty of Mathematics" are even more difficult to determine. These problems are very common in big data analysis and applications, so the importance of semantic analysis and computation for big data analysis applications is self-evident, and directly affects the ultimate value of big data.

As the amount of information provided by words as strings is very small, semantic analysis at the lexical level usually requires the use of a certain semantic knowledge base or corpus, so that the calculation of semantic relevance between "mathematics" and "big data" can be based on the shortest path method based on the structure of the semantic graph, the calculation method based on the amount of information in the concept node, etc.

As vocabulary is also very common in relational data representations, the issue of semantic analysis is also important in structured big data. For example, how to calculate the correlation between "Beijing", "Shanghai", "Xiamen", etc., stored in the city field is a fundamental requirement in applications such as big data mining involving regions. This is a fundamental requirement in applications such as Big Data mining involving regions. In addition to the semantic structure maps constructed by various methods, word2vec-based training methods are also a good choice.

Sentence-level semantic analysis techniques are more common in big data mining than lexical semantics. Typical application scenarios include finding the components of an event and their relationships in the text of a news report, and identifying commentary information in the text of a review. For example, in the sentence "The screen of a mobile phone is very large", "mobile phone" and "screen", "large" and "very" are both a modifying relationship, while "screen" and "large" are a declarative relationship. There is no uniformity in Chinese grammar as to how many semantic relations there are between words and how many names there are for the various semantic relations between real words in Chinese after they have entered a sentence. However, the main semantic relations frequently mentioned at present are giving thing, receiving thing, with thing, instrument, result, direction, time, purpose, manner, reason, colleague, material, quantity, datum, scope, condition, collateral, etc. It is the diversity of semantic relations that also makes the task of semantic analysis research rich and varied. However, due to the limitations of computer processing and reasoning capabilities, only a very small proportion of semantic relation analysis is currently targeted in this area of research.

3.4 Business Big Data Pre-processing Case

The twenty-first century is the century of information technology, and the "ubiquitous" Internet has resulted in an increasingly large amount of user data, and with the growing maturity of big data technology, this huge amount of data has become an inestimable asset. Amazon, Taobao, Facebook and other internet companies are representative of companies that make good use of big data technology.

Amazon is known for its enterprise cloud platform, and although the company has a high quality user data resource and huge website traffic, for quite some time before, Amazon had focused mainly on product sales, with advertising only as a supplement to its sales business. However, in 2013 alone, Amazon's advertising revenue surged by 45.51% and continues to grow. Although the proportion of advertising revenue in the group's total revenue is very low and still far behind other advertising giants,

the emergence of more and more rich advertising products based on huge amounts of user data and the growth in revenue generated by advertising is a reflection of the value of big data analytics applied within the company.

Amazon wants to improve the ROI of advertising and achieve precision marketing by analysing customer behaviour. Amazon has world-class personalised recommendation technology, unparalleled user data and interactive video content. Amazon is currently leveraging these strengths in an effort to build a new generation of advertising products. This new generation of advertising products is based on a powerful demand-side platform (DSP), real-time bidding (RTB) advertising model, which enables the presentation of ads based on the interests of users. The classified user groups are then matched with different types of ads. The ads are most effective if they are placed on products that the user needs or is interested in.

To achieve this, Amazon uses its own infrastructure cloud service with a 100-node on-demand elastic MapReduce cluster on the backend architected with Hadoop technology. Amazon Web Services' Elastic MapReduce runs on the Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3), which is extremely scalable. In this way, processing time can be reduced from 2 days to 8 h, and the return on advertising investment is increased by 500%.

In December 2013, Amazon was granted a patent for "predictive logistics". The patent combines a big data application with a logistics system that allows the company to start delivering items before a customer clicks "buy". The patent predicts what items a customer is likely to buy but has not yet ordered, based on past orders and other relevant factors for a particular region, and begins packaging and delivering these items. These pre-delivered items are stored in the courier company's delivery centre or on a truck before the customer places an order, and can be delivered more quickly once the customer has placed a purchase, effectively reducing delivery times.

When predicting "predictive logistics" items, Amazon takes into account factors such as previous orders, products searched, wish lists, cart contents, returns and even the length of time a customer's mouse hovers over an item to improve the accuracy of the prediction.

Amazon says the predictive logistics approach is particularly applicable to pre-sold items, particularly popular books that are pre-set to open on a certain date, for example. The patent exemplifies an emerging trend—intelligent prediction—whereby technology and consumer companies are increasingly inclined to predict consumer demand before they take a purchasing action, rather than doing the statistics after the consumer has made a purchase. This is where data pre-processing pays off.

Chapter 4
Big Data Database for Business

4.1 Key-Value Store

Key-value databases, one of the most common NoSQL (Not only SQL), are essentially shared sorted arrays or distributed hash tables. The hash table is a data structure that is accessed directly based on key values, that is, it accesses records by mapping key values to a location in a table to speed up lookups. This mapping function is called a hash function and the array that holds the records is called a hash table. Data in a key-value database is stored as a key-value pair, with a particular key pointing to a particular data value. In practice, users store access to data values through a unique primary key, and the time complexity of storing access to data is only $O(1)$. This makes key-value databases simple and efficient. In addition, the extended form of the key-value databases enables the sorting of keys and allows for range queries and the ordering of keys.

Key-value databases can be divided into three categories according to how the data is stored: temporary storage where the data is stored in memory, permanent storage where the data is stored on the hard disk, and temporary storage used in combination with permanent storage. Temporary storage is where data is stored in memory and is extremely fast to read and write, but a disadvantage is that once the application stops executing, the data in memory may be lost and the size of the data is limited by the memory capacity. This type of key-value database represents Memcached, etc. Permanent storage is where the data is stored on the hard disk and is slow to read and write, but data is not lost. Representatives of this type of key-value database: Tokyo Tyrant, Flare, etc. Temporary and permanent storage methods combine the advantages of both, storing data in memory and hard disk at the same time, through the design of data retention mechanisms to achieve both high-speed processing data and permanent storage purposes. A typical representative of this category is Redis.

J. Cao, *E-Commerce Big Data Mining and Analytics*, Advanced Studies in E-Commerce,
https://doi.org/10.1007/978-981-99-3588-8_4

4.1.1 Background to the Development of Key-Value Store

At the beginning of the twenty-first century, with the rise of Web 2.0 technology, the Internet entered a new period of rapid development. Under the Internet Web 1.0 era, Internet content information was published by a small number of website editors. In the Internet Web 2.0 era, the publisher of Internet content information became every Internet participant. From "read-only" in the Web 1.0=-0 Zxcv bnmZxc vbnera to "readable and writable" in the Web 2.0 era, Web 2.0 era technology products include blogs, RSS, Wikipedia, web digests, social networks, P2P, etc. These technology applications generate massive amounts of data. The massive amount of data generated by these technology applications makes the storage and fast access capabilities of relational database systems a huge challenge.

In the era of big data, the servers of subjects such as Taobao, Jingdong, and Weibo generate massive amounts of data every day. With more dimensions of data, each row of data often has partial information missing and other incomplete situations. If we continue to use the traditional storage method, the missing information will be empty, which is equivalent to storing a sparse matrix. That will inevitably lead to high storage costs, which is unacceptable to business entities. In addition, in the era of big data, where needs change rapidly and therefore data models change rapidly, databases should be able to cope with this normality in a highly effective manner. Key-value storage can effectively solve such problems by identifying data instances with a primary key. In the column part of a row of data, each column of data contains metadata (key), data (value), and a timestamp. While the data structure is not fixed, each tuple can have different fields, and each tuple can add some of its key-value pairs as required. So, it is not limited to a fixed data structure and reduces a lot of time and space overhead.

4.1.2 Key-Value Database Versus Relational Databases

The purpose of the key-value database system is to store large amounts of semi-structured and unstructured data to cope with the continuous expansion of data volume and user size, which traditional relational databases cannot meet. This does not mean that traditional relational databases will be replaced by non-relational databases, and the objective of Key-Value database systems is not to eventually replace relational database systems. The two are not antagonistic relationships, but complementary to each other, to jointly solve the database processing problems required in practical applications. Figure 4.1 shows the relationship between relational databases and non-relational databases.

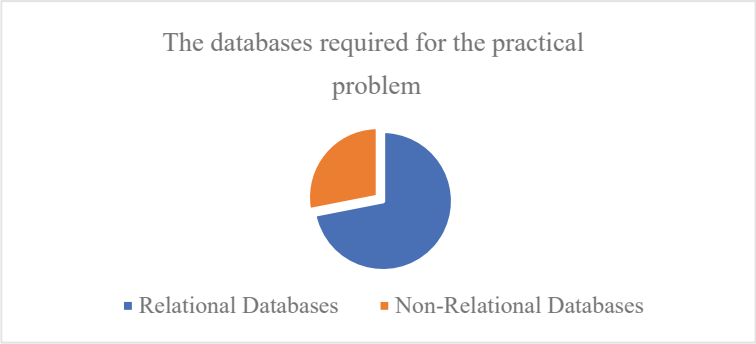


Fig. 4.1 Relationship diagram between relational and non-relational databases

At present, although relational databases still occupy a considerable share of the whole database management system, the demand and application potential of relational databases are still high. The differences between the two are as follows

(1) In a relational database system, the database contains tables, which contain rows and fields, and the rows consist of the data values of individual fields, and the rows in a table all have the same data format. A key-value database storage system does not contain a data table like a relational database, which generally contains domains. Instead, each domain contains several records.
(2) Relational databases have a well-defined data model that contains mechanisms such as policies, relationships between tables, and transactions. Relationships between data are based on the data itself, and relationships exist between tables and tables, tables and fields, and data and data, rather than based on the needs of higher-level applications. In a key-value database storage system, data records are simply identified and accessed by a key-value identifier. However, there is no concept of relationships between data.
(3) Relational databases improve the ability to share data, while key-value databases generally require data redundancy to ensure their reliability.
(4) Relational databases are suitable for storing traditional data, such as strings and numbers, and for querying, while Key-Value storage systems are suitable for storing and querying large amounts of non-relational data. The key-value database system has a clear advantage.

4.1.3 Key Value Database Advantages

Although widely used, traditional relational databases still have their insurmountable bottlenecks, and they struggle when faced with problems such as

(1) When there are more than 10,000 read-and-write requests per second, the hard disk I/O performance becomes a bottleneck and the program execution response becomes less effective.
(2) In the face of massive amounts of data, a single storage node will not be able to meet the storage needs. Data usually need to be divided into libraries and tables, and difficult to maintain large data. When the amount of data reaches a certain level, the query SQL efficiency is extremely low, so the query time will be exponential growth and difficult to expand horizontally. We cannot just simply increase the hardware, and service nodes to improve system performance.

For these cases, key-value databases have shown good performance. Key-value databases often use distributed storage to store large amounts of data, improving system performance by using inexpensive clusters of machines, and eliminating the need for high-performance, costly machines. Key-value database systems eschew some of the key features of relational database systems, such as real-time read/write and strict transactional consistency, which allows them to take full advantage of parallel computing and distributed applications and focus on unstructured data processing, system scalability, reliability, etc. Currently, the key-value database features are mainly in the following areas.

(1) **High scalability**

The key-value database storage system offers a very high degree of scalability, with users typically only having to configure the system to suit the size of their needs, and quotas can be increased as demand grows. Without the strict field structure of data tables and the relationship between tables, the key-value systems can easily be deployed on multiple servers for distributed applications, thus increasing the scalability of the entire system and making it more convenient and flexible.

A key-value database storage system is the perfect partner for cloud computing, which is all about being flexible and responding to users' needs for scalability, and this is where key-value database systems come in. If you are trying to leave the scaling requirements of a large system to hundreds of servers, then a key-value database storage system should be a better solution. The high throughput capability of key-value database storage systems is an excellent choice for data storage solutions that can cope with large-scale parallel computing and load balancing.

(2) **Flexible data tuple format**

As there are no complex data formatting requirements, manipulating fields in data tuples in a key-value database is easy and fast. It does not significantly impact server performance, which is often a performance nightmare in a relational database, especially for fields associated with multiple tables.

include secondary database models — 65 systems in ranking, July 2020

Rank Jul 2020	Rank Jun 2020	Rank Jul 2019	DBMS	Database Model	Score Jul 2020	Score Jun 2020	Score Jul 2019
1.	1.	1.	Redis	Key-value, Multi-model	150.05	+4.40	+5.78
2.	2.	2.	Amazon DynamoDB	Multi-model	64.58	-0.29	+8.17
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	30.40	-0.40	+1.32
4.	4.	4.	Memcached	Key-value	25.84	+1.04	-1.22
5.	↑ 6.		etcd	Key-value	8.57	+0.52	
6.	↓ 5.	↓ 5.	Hazelcast	Key-value, Multi-model	8.45	+0.04	+0.18
7.	7.	↓ 6.	Aerospike	Key-value, Multi-model	6.82	+0.15	+0.23
8.	8.	↓ 7.	Ehcache	Key-value	6.51	+0.23	-0.05
9.	9.	↑ 10.	ArangoDB	Multi-model	5.85	+0.47	+1.19
10.	10.	↓ 8.	Riak KV	Key-value	5.41	+0.42	-0.65
11.	11.	11.	Ignite	Multi-model	4.95	+0.08	+0.67
12.	12.	↓ 9.	OrientDB	Multi-model	4.88	+0.06	-0.81
13.	13.	↓ 12.	Oracle NoSQL	Key-value, Multi-model	4.42	+0.20	+0.96
14.	14.	↓ 13.	InterSystems Caché	Multi-model	3.44	-0.02	+0.14
15.	15.	15.	Oracle Berkeley DB	Multi-model	3.40	+0.20	+0.36

Fig. 4.2 Top 15 key-value databases

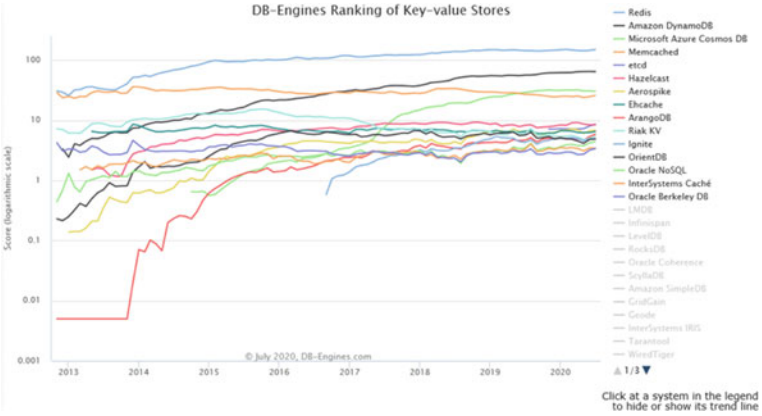


Fig. 4.3 Top 15 key-value database popularity trend chart

4.1.4 Redis

DB-Engines are dedicated to collecting and presenting information on database management systems, covering almost all relational and non-relational databases. The database engine rankings are updated monthly and are based on their respective popularity. As of June 2020, the site has 358 databases included and 217 non-relational databases. Statistics Key-Value Databases has 65 products. Of these, Redis is firmly at the top of the list. Figure 4.2 shows the top 15 Key-value data libraries. Figure 4.3 shows the trend of popularity of the top 15 Key-Value databases from 2013 to 2020.[1]

[1] https://db-engines.com/en/ranking_trend/key-value+store.

Redis is an open-source (following the BSD open-source protocol), networked, memory-based, key-value database storage system that allows for data persistence. Unlike other structured storage systems, it supports multiple types of storage, including String, List, Set, Sort set, and Hash, and you can do a lot of atomic operations on these data types. In terms of operations, the TCP protocol-based nature of Redis allows it to pipe data operations, and Redis itself provides a client that can connect to the server, through which data access operations can be easily performed. Redis supports multiple languages, including C, C++, JAVA, Python, PHP, R, …

(i) Redis data structure

Within Redis, data structure types are supported by efficient data structures and algorithms and are used extensively in the construction of Redis itself. The four Redis data structures are described below: String, Hash, List, and Set.

(1) **String**

Redis does not use the string of the C programming language, but rather SDS (Simple Dynamic String). The SDS is used by the underlying Redis language and used in almost all Redis modules, which replaces the default C language's string data type. Inside Redis programs, SDS is used in the vast majority of cases, rather than the traditional C string. In the C programming language, a string can be an array of characters (char) ending in "\0" (Fig. 4.4). Redis also follows the same rule of C string data ending in "\0". When performing a series of API operations on SDS, an extra byte is always allocated to store "\0" for compatibility with some of the C language string functions.

The C language string type has a single function and a low level of abstraction, and cannot efficiently support some of Redis' common API operations, so Redis chose to develop SDS to meet its needs.

The data structure of an SDS is shown in Fig. 4.5.

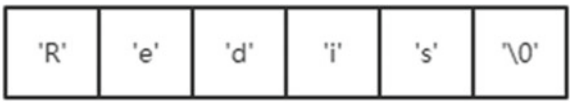


Fig. 4.4 Example of a C language string

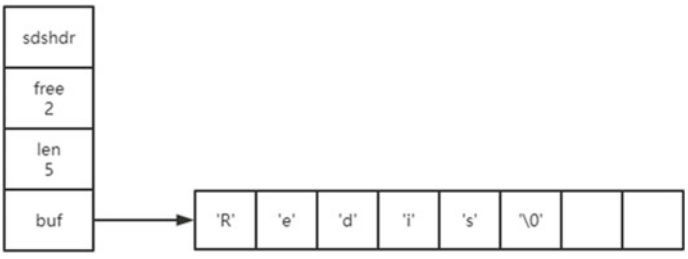


Fig. 4.5 SDS example

```
struct sdshdr{
    // Record the length of the used bytes in the buf array
    int len:
    // Record the length of the remaining space in the buf array
    int free:
    // Byte arrays for storing strings
    char buf[]:
}
```

As shown above, "free" records the number of unused bytes in the "buf" array, "len" indicates the length of the string, and "buf" is the byte array that stores each byte as well as the null character. For example, when the value of the "free" attribute is 2, this means that the SDS has two unused character spaces, i.e. unlike C language strings, the SDS can have a surplus of allocated byte space. When the "len" attribute is 5, the SDS stores a string of 5 bytes. "buf" is an array of characters, and the figure shows the string "Redis", which notably ends in "\0".

Compared to C strings, SDS has the following advantages

(i) **The time complexity of obtaining the length of a string is $O(1)$**. The "len" property directly stores the length of the current string. In addition, setting or updating the SDS string length is done automatically by the SDS API when it is executed. C strings, on the other hand, have to be traversed character by character, with a time complexity of $O(N)$.

(ii) **It avoids overflowing the buffer**. For example, when a C string is appended, the string will grow. If the current string space is not adjusted beforehand, the current string may overflow and the contents of the appended string may be accidentally modified.

(iii) **It reduces the number of memory allocations**. SDS uses a space pre-allocation strategy to reduce the number of memory allocations. After an SDS splice has occurred, if the value of the "len" attribute is less than 1 MB at that point, it allocates more unused space of the same size as "len". At this point, the value of the "len" attribute is the same as the value of the "free" attribute. If the value of the "len" attribute is greater than 1 MB, then an additional 1 MB of unused space is allocated. For example, if the value of the "len" attribute is 20 MB after a certain operation, then 20 MB + 1 MB of byte space will be allocated. Inert space release is used to optimize SDS string-shortening operations. When a string is shortened, the program does not immediately reclaim the freed space but records it as free for later use when SDS splicing occurs.

Redis has to deal with mere byte arrays in addition to C strings, as well as things like server protocols. So for convenience, the "Redis" string representation should also be binary-safe. For the C language's strings, the fact that no null characters can exist anywhere else in the string other than at the end of the string imposes a significant limitation. Because general data information such as images and videos often have null characters, C strings cannot be stored. A good program should not

make any assumptions about the data stored inside the string. The data can be a C language string ending with "\0". "\0" can also appear elsewhere in the string, either in other locations such as the head of the string, simple byte arrays, or other formats of data. Also, as mentioned above, ending SDS with "\0" at the end is compatible with some C string functions.

(2) **Hash**

Hash is an abstract data structure used to hold key-value pairs. Each key in a hash is unique, and programs can find, delete, modify, etc. By querying the unique key, the mapping value is associated with it. In general, the data structures of the hash are built into high-level programming languages, such as Java, Python, etc. The C programming language does not have a built-in hash structure, so Redis has developed its own hashes implementation. Hashes are widely used in Redis, and the Redis database uses hashes as the underlying implementation. The hashes are used to add, delete, change and check operations on the database. For example:

```
//Execution of orders.
redis > SET mes "hello"
OK
```

When this command is executed, a key-value pair is created in the database: the key is "mes" and the value is "hello", and the key-value pair is stored in a hash. There are various ways to implement a hash, but to balance efficiency and simplicity, Redis uses a hash table as its underlying implementation.

The hash table structure used by the Redis dictionary is defined as shown below

```
typedef struct dict{
//hash table arrays
dictEntry **table;
//hash table size
unsigned len size;
// hash table mask, always equal to size-1, index value calculated when stored
unsigned len size mask;
// Number of existing nodes
unsigned len used;
}dict;
```

The "table" attribute in the hash table is a secondary pointer, equivalent to an array of pointer data, where the pointer data is a pointer to a hash table node. "len size" indicates the size of the hash table. "len size mask" indicates the hash table mask, whose value is always equal to "len size" − 1. Together with the hash value, this "len size mask" attribute determines which index a key should be placed on this attribute, and determines which index of the table array a key should be placed on.

The hash table node structure is defined as shown below

```
typedef struct dicten{
//Key
```

```
void *key;
//value
union{
void *val;
unit64_t u64;
    int64_t s64;
    }v;
// points to the next hash table node, forming a chain
struct dicten *next;
}dicten;
```

The pointer key points to the address of the key in the key-value pair to get the key. The attribute "v" holds the value in the key-value pair, which can be either a pointer or an integer of uint64_t or int64_t. The pointer next points to another hash table node pointer.

The Redis dictionary structure is defined as follows

```
typedef struct dic{
// Type-specific functions
dicType *type;
// Private data
void *privdata;
//hash tables
dic t[2];
//rehash index, the value is -1 when rehash is not in progress
int rehashidx;
}dic;
```

As shown in the pseudo-code above, the "type" pointer and "privdata" pointer are set up to cope with different types of key-value pairs to create polymorphic dictionary generation. The "type" pointer points to the "dicType" structure, which contains several functions for specific types of key-value pairs, e.g. functions to calculate hashes, functions to destroy keys, etc. The "privdata" pointer points to optional parameters that need to be passed to specific functions. The attributes "t" are arrays, which store the hash table. In general, dictionaries only use the t[0] table. While the t[1] table is only used in rehash. The following section will talk about rehash.

An example dictionary for Redis is shown in Fig. 4.6.

Figure 4.6 shows that the t array in the dictionary "dic" structure stores two dict structures: dict[0], dict[1]. the attribute "table" in dict[0] points to the pointer "dicten", and the value of the size attribute is 3, which means that there are three hash tables, equal to the number of elements stored in the "dicten" array. "sizemask" value: size-1 = 2. The "dicten" array stores three hash tables, and in the second hash table (the "dicten" array subscript starts with 0) two hash table nodes are stored, i.e.

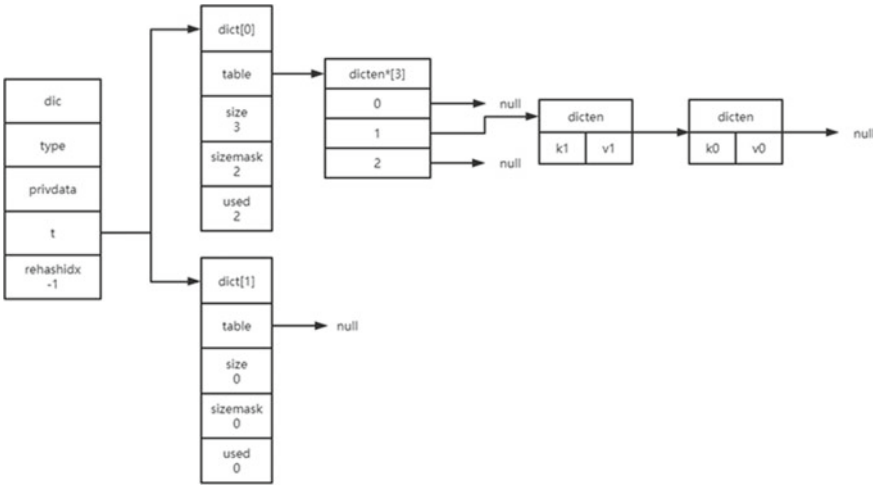


Fig. 4.6 Dictionary example

the used attribute value in the "dict" structure is 2. Each hash table node stores a key-value pair.

If a new key needs to be stored in the hash table during normal operation, the hash value and index value are calculated based on the keys in the key-value pair, and the hash table node containing the new key is placed at the specified index of the hash table array based on the index value. Assuming that when more than one new key is generated to be added to the dictionary, the key conflict problem is raised: two or more keys are assigned to the same index.

Redis uses the chain address method to solve the conflict problem. As shown in Fig. 4.5, hash table nodes are linked by a next pointer to build a single chained table, linking nodes that point to the same hash table index, thus solving the key conflict problem. For hash tables, the performance of the hash table depends on the ratio between the hash table size and the number of nodes saved.

(i) The size of the hash table to the number of nodes, with the best performance of the hash table when the ratio is 1:1.
(ii) If the number of nodes is much larger than the size of the hash table, then the hash table degenerates into multiple chained tables and the performance benefits of the hash table itself cease to exist.

Redis guarantees that when the above ratio reaches a certain value, a rehash operation is performed, i.e. the hash table is expanded or reduced. When expanding, space is traded for time, and when shrinking, time is traded for space. Space is allocated to the t[1] hash table based on whether the expansion or reduction operation is performed and the value of the used attribute and the hash and index values of all key-value pairs in t[0] are recalculated. The key-value pairs are migrated to t[1] based on the hash and index values, and when the migration is complete, t[0] is freed.

rehash operations are generally performed in an incremental manner. Redis uses two incremental rehash methods.

(i) **Each time an add, read or delete operation is performed, the rehash is executed once**.
(ii) **Rehash is performed when Redis' server routine tasks are executed**. The rehash process of database dictionaries is accelerated by performing rehash on those dictionaries in the database dictionary that require rehash as much as possible within a specified time frame.

(3) **List**

The underlying implementation of lists in Redis is built using linked lists, a data structure embedded in many common high-level programming languages, but not in C itself, so Redis builds its own lists. Each node holds a forward pointer (*prev), a backward pointer (*next), and a value. The Redis list can perform operations similar to those performed by many programming languages, such as inserting and removing elements from the head and tail of the list, reading the list attribute values one by one, and returning the list length attribute value.

(4) **Sets and ordered sets**

Collections in Redis are like any other high-level programming language in that the elements held in a collection are not duplicated. This is where collections differ from lists, which can hold the same elements. Redis collections use only the keys in a hash table key-value pair, not the associated values. This is because the keys in a hash table are unique and non-repetitive, which just happens to match the non-repetitive nature of collection elements. The elements in a collection are arranged out of order and the user cannot insert and read data from the head list node or the tail list node as they can with a list. Similar to other high-level programming languages, Redis collection operations include basic add, delete, update, intersection, merge and difference operations between collections.

Unlike collections that use only the keys in a key-value pair, ordered collections also use the values in the key-value pair, which must be floating point numbers. It is important to note that unlike other data structures within Redis, ordered collections can access elements not only from keys but also from values. Ordered collections can also be added, deleted, updated and read according to the official commands provided.

The commands for operating the individual data structures are described in detail in the section on commands below.

(iii) **Redis persistence methods: RDB and AOF**

At runtime, Redis database data is maintained in memory and will be lost when the Redis database is shut down or restarted. Redis provides two persistence modes, RDB (Redis DataBase) and AOF (Append-only file), to ensure that data is kept safe and intact.

RDB saves a snapshot of the database to disk as a binary. When Redis is running, the RDB program saves a snapshot of the database currently in memory to a disk file. A snapshot is a copy of the data stored in memory at a point in time, and in the event of a system crash, the user can restart Redis and use the snapshot to recover the data. Snapshots can be set in a configuration file, e.g.

```
save 300 1000 // If there are 1000 writes in 300 s, a snapshot is generated.
save 60 10000 // If there are 10000 writes in 60 s, a snapshot will be taken.
```

It is worth noting that a snapshot can be used to recover only the data saved in the most recent snapshot. Imagine a scenario where a snapshot is generated at 2.00 pm, then 20 changes are made to the data, but no new snapshot is generated, and the system suddenly crashes.

AOF is a protocol text record of all commands written to the database (and their parameters) in an AOF file, which records the state of the database as if it were a history of all commands run and can be executed to restore the database to the state it was in before the system error. The AOF file grows over time, taking up the entire disk. For this reason, Redis has designed an AOF rewrite compression mechanism that opens a new thread, scans the database data, removes redundant commands from the AOF, and then stores the compressed AOF data in a temporary AOF file. When the scan is complete, the original AOF file is replaced with the temporary AOF file. In this way, the commands recorded in the AOF file are the most concise and therefore do not take up much space.

(iv) **Client and server**

The Redis server is capable of connecting to multiple clients, in a one-to-many format (Fig. 4.7). The client's structure defines several attributes, including the socket descriptor used to communicate with the server, the name, the input and output buffers, information about the database pointers connected to it, etc. By default the client has no name, i.e. the name attribute has an empty value and can be set by the user via a command. The client has an input buffer and an output buffer. The input buffer holds the request commands that need to be sent to the server and is dynamic, expanding or shrinking according to the space required, but cannot exceed a maximum of 1 GB. One of these is a fixed-size buffer and the other is dynamically allocated as required. In general, the fixed-size buffer holds relatively small data and the dynamic buffer holds larger data.

The Redis server is responsible for processing requests from multiple clients and returning the results to the client. At the same time, the server also needs to perform several other necessary operations to keep itself running well. For example, updating the server time cache, updating the LRU (Least Recently Used) clock, managing client resources, managing database resources, etc. There are many functions within the server that require access to the current time of the system, and the server needs to call the current time frequently, so the server needs to cache the current time of the system. In addition, the server itself keeps a record of the last time the object was accessed, and by updating the LRU clock, it can accurately record the idle time

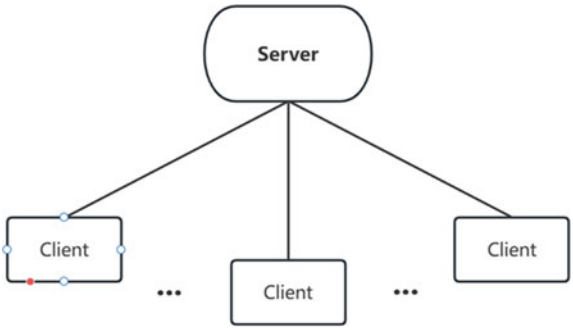


Fig. 4.7 Relationship between client and server

of the object, so that it can discard data that has been idle for too long and optimise space. The server also needs to manage the resources of the client and the database, and check that the connection between them is not broken.

Redis is a database management system that supports clustering. As such, its servers use master–slave replication to achieve coordinated and synchronized consistency across the cluster. Master–slave replication means that one of the multiple servers is selected as the master server and the others as slaves. The master server quickly synchronizes state information by sending snapshot files to new slave servers that join the system, which is then sent to the slave servers with every command executed by the master server, bringing consistency to the entire cluster.

(v) **Redis transactions**

Redis transactions differ from traditional relational databases in that they are implemented through commands such as MULTI, EXEC, WATCH, etc. A Redis transaction begins with the command MULTI, followed by the execution of multiple user-committed commands, and ends with the EXEC command. The MULTI command means that the client executing the command changes from a non-transactional state to a transactional state, and then if the client sends one of the four commands EXEC, DISCARD, WATCH, or MULTI, the server will immediately execute the command. if the client sends one of the four commands EXEC, DISCARD, WATCH and MULTI, the server will execute the command immediately. If it is not one of these four commands, it will not execute it and will place these commands in the transaction queue and return a QUEUED reply to the client. It should be added that if the client is in a non-transactional state, the command sent is executed immediately. During the transaction execution phase, a client in a transactional state wants the server to send an EXEC command, then the server immediately traverses the client's transaction queue, executes all the commands in the queue, and returns the result to the client. The ACID feature of Redis transactions is described below.

(1) **Atomicity**

The atomicity of a transaction means that multiple operations in the transaction are either not executed or all of them are executed, not just half of them, and the operations

are indivisible. Redis transactions do not have a rollback mechanism, so even if a command in the queue is executed incorrectly, the command in the transaction queue will be executed, meeting the requirement of atomicity.

(2) **Consistency**

Consistency in transactions means that the database should be consistent before and after the execution of a command, regardless of whether the execution was successful or not. The consistency problem in Redis can be discussed in two parts: transaction command entry errors, and transaction command execution errors. If a client sends an incorrect command to the server during command queuing, Redis will refuse to execute the transaction and return a failure message. If an error occurs during the execution of a transaction, Redis will only include the error in the result of the transaction, which will not cause the transaction to break or fail in its entirety, and will not affect the result of the executed transaction command, or the transaction command to be executed later, so it does not affect transaction consistency either.

(3) **Isolation**

The isolation of a transaction means that multiple transactions will not affect each other during execution. Redis is a single-process program and it guarantees that there will be no interruptions to transactions while they are being executed and that transactions can run until all the commands in the transaction queue have been executed. Therefore, Redis transactions are always isolated.

(4) **Durability**

Redis transactions are essentially a series of commands stored in a queue and do not provide any additional persistence features. Redis transactions are not persistent in in-memory mode. In RBD persistence mode, snapshots are also not triggered in real-time and data loss can occur, which does not guarantee timely and persistent data retention. In AOF persistence mode, persistence is also not guaranteed in some cases.

In summary, Redis transactions satisfy atomicity, consistency, isolation, and not persistence.

(vi) **Publishing and subscription**

Publish and subscribe is a message communication mode. A client publisher publishes messages to a specified channel using the PUBLISH command, and a client subscriber subscribes to the specified channel using the SUBSCRIBE command to receive these messages. In addition to the above functionality for subscribing to channels, Redis also supports subscription patterns, by subscribing to one or more patterns using the SUBSCRIBE command. When a client sends a message, not only can the channel subscriber receive it, but all pattern subscribers matching that channel can also receive it. This is shown in Fig. 4.8.

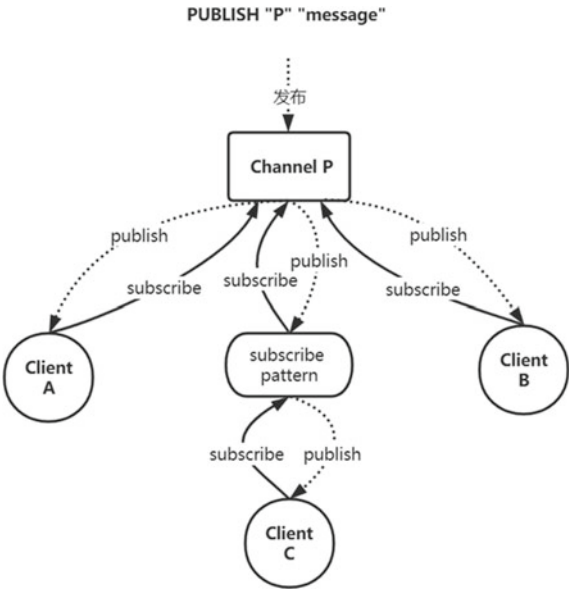


Fig. 4.8 Schematic diagram of the publish and subscribe model

4.2 Column Family Store

In the context of the Big Data era, the traditional relational database with row-oriented storage has exposed the following drawbacks when storing and processing massive amounts of data: insufficient scalability due to structured data, insufficient high concurrent read-and-write capability, slow query speed based on row storage, relatively few query functions, relatively fixed data storage format, etc. Column family databases are the product of the big data era. The introduction of columnar databases has provided a very good solution for handling huge amounts of data. Following the papers "The Google File System" and "MapReduce: Simplified Data Processing on Large Cluster", Google published the book "Bigtable: A Distributed Storage System" in 2006. This paper laid the groundwork for the development of distributed columnar storage systems in the era of big data. The more widely used column family databases: HBase and Cassandra, are all heavily influenced by this paper.

4.2.1 Column Family Database Storage Structure

HBase, a sub-project of the Hadoop project, is an open-source implementation of the BigTable project. As the version is updated, there are some deviations between HBase and BigTable in some details of the functional implementation, but the essential features of both are the same. This section takes the open-source HBase as an example

to introduce the basic elements of a column family database storage structure. The basic elements of the storage structure in the logical data model:

(1) **KeySpace**

The key space is similar to the table name of a traditional relational database and is the top-level data structure of the database. The key space can hold row keys, columns, and column-family.

(2) **Row Key**

A row key is similar to the primary key in a traditional database and is used to identify a row of data in the database. Unlike the primary key in a traditional relational database, a row key is a logical row key for a column family database. In actual disk storage, it is not stored consecutively with the relevant column data in this row, but rather by column, unlike traditional relational databases where the primary key is stored consecutively on disk with the row data in which it is located. In addition, when a piece of data is written to the column family database, it is automatically sorted by row key, with the sorting rule: the row key is sorted by ASCII code.

(3) **Column**

Columns are in a sense equivalent to columns in traditional relational databases, the data structures used to store data values. The difference is that for the column family database HBase, columns do not define what types of data can be stored, any data value that can be represented as a string can be stored, it can be a string, a number, a link and other information.

Column Family

As the name suggests, a column family is a collection of several columns, mostly of related columns, with strong correlations between columns. For example, in the identity information family: name, age, gender, and home address. In the shopping app family: Taobao, Jingdong, Tmall, Suning.

Timestamp

The introduction of the concept of timestamps has changed the traditional concept of two-dimensional extensions to the horizontal and vertical extensions of tables. The data stored in each column can have a before and after version, tagged with a timestamp to identify the data, transforming a two-dimensional data store in a logical sense into a three-dimensional logical one. In the case of HBase, the timestamp is represented by a string of numbers, the larger the number the newer the version. You can manually specify the timestamp to distinguish the priority of the data, otherwise, the system automatically assigns the timestamp based on the time of input. For example, for the home address column, some applications need to save the existing home address and the previous home address at the same time, you can distinguish the before and after the home address by using the timestamp, marking the previous home address with timestamp = 3 and the existing home address with timestamp =

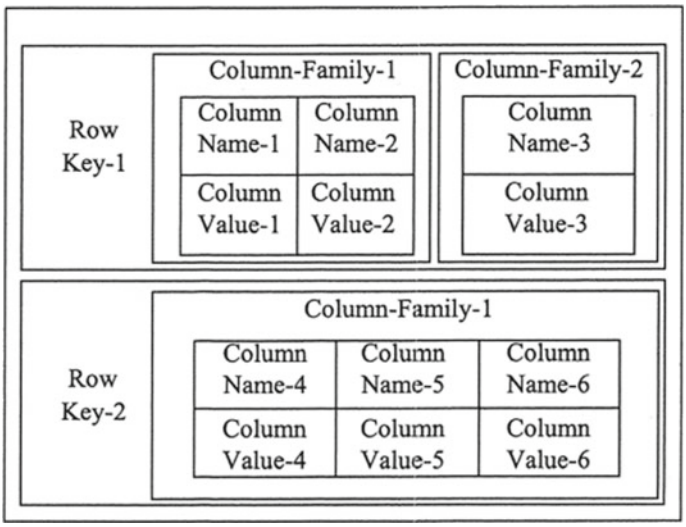


Fig. 4.9 Data structure model of column family database

4. The larger the timestamp, the more recent the data. It is worth noting that the above two home addresses are stored in the same column, whereas traditional relational databases cannot store data in one column (Fig. 4.9).

Compared to columnar databases, relational databases cannot use flexible timestamps to store multiple versions of data information and can only add rows horizontally or vertically to ensure data integrity. In addition, for sparse data, NULL values in relational databases take up disk space, whereas NULL values in columnar databases do not need to be NULL and do not take up space. In an era of increasing data fragmentation and multidimensionality, columnar databases have unparalleled advantages over traditional relational databases.

4.2.2 Column Family Database Features

Google has designed BigTable to reliably handle large amounts of data information. In many projects such as Web Indexing, Google Earth and Google Finance, Google uses BigTable to store huge amounts of relevant data information to provide services to its users. These projects require BigTable not only to store large amounts of data, but also to be low latency, flexible and high performance, and highly scalable. BigTable is a distributed storage system for managing structured data, designed to achieve reliable, flexible and high-performance data storage using the underlying HDFS inexpensive fleet of distributed storage. By analyzing the initial design ideas

of the column family database above and the storage structure elements from the previous subsection, the column family database features are as follows

(1) **Massive data processing**. For massive data processing, BigTable uses an underlying inexpensive distributed fleet of machines to store data, achieving high data throughput, real-time batch processing and high reliability.
(2) **Expertise in handling sparse, fragmented, multidimensional data**
(3) **High concurrent access**. Basically, columnar databases are capable of millions of concurrent writes per second.
(4) **Low redundancy**. The use of timestamps and the fact that null values do not take up storage space makes columnar databases less redundant and more disk efficient than traditional databases.
(5) **Column-oriented and highly scalable**. Columns are added as they are used. Tables can be scaled vertically and horizontally, making them very flexible. This is something that traditional relational databases cannot achieve when dealing with large volumes of data.

4.2.3 HBase

(i) Introduction to HBase

HBase is a sub-project of the Hadoop project, written in Java and responsible for building a high-performance, highly reliable, column-oriented, scalable distributed storage system, developed by the Apache Foundation as a software framework for distributed processing of large amounts of data. Initially, the Hadoop project was not a standalone project, but a subproject of Apache Lucene. In 2004, Google published three papers in quick succession: GFS, MapReduce and Bigtable, all of which were of great importance to the Hadoop project, and Hadoop developed and implemented its framework design along the lines of the three papers: HDFS uses cheap clusters to reliably store large amounts of data, MapReduce provides bulk computing for large amounts of data, and HBase uses HDFS as a tool for processing large amounts of data. Hadoop has been developed and implemented along the lines of three papers: HDFS to reliably store large amounts of data using cheap clusters, MapReduce to provide bulk computational processing of large amounts of data, and HBase to use HDFS as its underlying file storage system. Since 2008, Hadoop has been a top project of the Apache Foundation and is used by more and more Internet companies.

In 2006, the Bigtable paper was published, and as a technical application used by Google itself, the relevant source code for Bigtable was not published. 2007 saw the development of the HBase open-source project, which was heavily influenced by it, and was initially intended to deal with the problem of storing massive amounts of data generated by natural language search. After two years of development, HBase officially became a sub-project of the top-level project Hadoop. As Hbase relies on the underlying distributed file system HDFS, a sub-project of Hadoop, the cluster must deploy a Hadoop system before running the Hbase distributed cluster model.

HBase has now been updated to version 2.3.2. The official download is available at https://hbase.apache.org/downloads.html.

HBase is a type of NoSQL database, but technically it is more of a "data store" than a "database" because it lacks many of the features of a relational database such as predefined columns, auxiliary indexes, triggers and advanced query languages. HBase has several features that support linear and modular scaling, and HBase clusters scale through region servers (HRegionServer). For example, if a cluster scales from 10 to 20 region servers, it doubles its storage capacity and processing power. Relational databases can scale well, but only to a single server and are limited by the size of the individual database servers. For optimal performance, specialised hardware and storage are required. important features of HBase include.

(1) Strongly consistent read/write: HBase is not a "final consistent" data store. This makes it ideal for tasks such as high-speed counter aggregation.
(2) Automatic partitioning: HBase tables are distributed across the cluster via regions, and regions are automatically partitioned and redistributed as data grows.
(3) Automatic regional server failover.
(4) Hadoop/HDFS integration: HBase supports off-the-shelf HDFS as its distributed file system.
(5) MapReduce: HBase supports massively parallelized processing via MapReduce to use HBase as a source and receiver.
(6) Java Client API: HBase supports easy-to-use Java API for programmatic access.
(7) Thrift/restapi: HBase also supports Thrift and REST for non-Java front ends.
(8) Block caching and Bloom filters: HBase supports block caching and Bloom filters to optimize high-volume queries.
(9) Operations Management: HBase provides built-in web pages to provide operational insight as well as JMX metrics.

(ii) **Hbase and Hadoop**

Figure 4.10 shows the Hadoop ecosystem, with HBase playing a role in the ecosystem framework by providing persistent data storage and management tools. HDFS provides the underlying distributed data storage for Hadoop. zookeeper is an open-source distributed orchestration service that implements features such as data publishing/subscription, load balancing, naming services, distributed orchestration/notification, cluster management, master election, distributed locking and distributed queuing. YARN is a resource management system introduced in version 2. x of Hadoop, evolving directly from MapReduce, the new Hadoop resource manager, a universal resource manager that provides unified resource management and scheduling for upper-level applications, and it was introduced for clusters in terms of utilization, resource MapReduce is a programming framework for distributed computing programs that provides an easy way to design parallel programs, with two functions, Map and Reduce, programmed to achieve basic parallel computing tasks. Pig can easily handle data from HDFS and HBase, and like Hive, Pig can

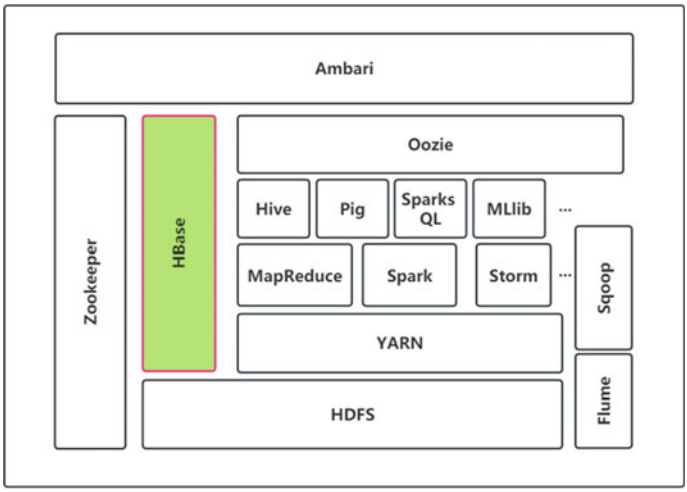


Fig. 4.10 Hadoop ecosystem

handle what it needs to do very efficiently, saving a lot of labour and time by directly manipulating Pig queries.

In short, HBase is a database for Hadoop, a distributed storage system. hbase uses Hadoop's HDFS as its file storage system. hadoop uses MapReduce to process the massive amounts of data in Hbase. Pig and Hive are used for querying and analyzing data, and the underlying layers are transformed into MapReduce programs to run.

(iii) **Introduction to HBase components**

Figure 4.11 shows the Hbase system architecture. The following will introduce in detail the important components or basic concepts such as Client, Zookeeper, Hmaster, HRegionServer, HRegion, HStore, MemHStore, HStoreFile, Hfile, etc. to better understand the principle of HBase implementation and data structure model.

HRegion: HRegion is the basic unit of HBase data storage and management, which consists of storage for each column family. If the table stores millions or even tens of millions of data information, Hbase will slice and dice this data information according to different row keys and store them as HRegion. A complete piece of data must belong to only one HRegion. A table can contain one or more HRegions. each HRegion can only be served by one HRegionServer server, and the HRegionServer server can serve multiple HRegions at the same time, and the HRegions from different HRegionServer are combined to form a logical view of the table as a whole. From the HMaster's point of view, each HRegion records its StartKey and EndKey (the first HRegion has an empty StartKey and the last HRegion has an empty EndKey), and since the RowKeys are sorted, the Client can use the HMaster to quickly locate the RowKey of each The HRegion is allocated to the corresponding HRegionServer by the HMaster, and then the HRegionServer is responsible for the start-up management

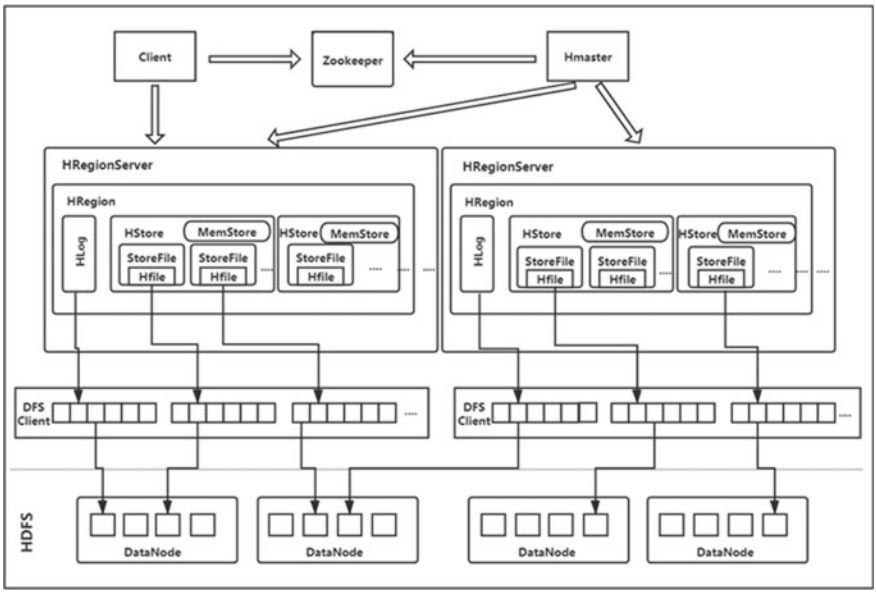


Fig. 4.11 HBase system architecture

of the HRegion and the communication with the Client, and is responsible for the data reading and writing.

Generally, HBase is designed to run a small number (20–200) of HRegions per server with a relatively large storage capacity (5–20 Gb). it is often desirable to keep the number of HRegions low for HBase for many reasons. Typically around 100 HRegions per HRegionServer server will produce the best results.

HStore: An HStore corresponds to a family of columns in an Hbase table, HStore stores MemHStore, and HStorefile, MemHStore is used to store the current data operations. When a write operation is performed, it is not written directly to the HStorefile, but first to the MemHStore for sorting, because the data stored in the underlying HDFS is arranged in an orderly fashion. When the amount of data in the MemHStore reaches a certain threshold, a new MemHStore is created and the old MemHStore is added to the flush queue and flushed (flushed) by a separate thread to the HStoreFile, which is stored in HFile format on HDFS. When the HStoreFile in an HStore reaches a certain threshold, it will be merged (major compact), merging the modifications to the same key together to form a large HStoreFile, and when the size of the HStoreFile reaches a certain threshold, the HStoreFile will be split again. When the size of the HStoreFile reaches a certain threshold, the HStoreFile is split into two HStoreFiles.

HFile: The HFile is the smallest structure in the HBase architecture, and all the data in HBase is in the HFile. section, Opening-time data section and Trailer.

Scanned block section: indicates that when the HFile is scanned sequentially (containing all the data to be read) all blocks will be read, including the Leaf Index Block and Bloom Block.

Non-scanned block section: this part of the data is not read during the sequential scan of the HFile and consists mainly of Meta Block and Intermediate Level Data Index Blocks.

Load-on-open-section: This section of data needs to be loaded into memory when HBase's HRegion server starts. including FileInfo, Bloom filter block, data block index and meta block index.

Trailer: This section records the basic information of the HFile, the offset values of each section and the addressing information.

HLog (WAL log): WAL means Write ahead log and is used for data recovery. Hlog records all changes to the data and can be recovered from the log once the data has been modified.

Client: Users access the Hbase cluster through the client. The client uses HBase's RPC mechanism to communicate with HMaster and HRegionServer, mainly communicating with HMaster for management-type operations and with HRegionServer for data reading and writing-type operations. When a client initiates a request, the process is as follows: the client first queries the meta table to obtain the HRegion location information and then communicates directly with the HRegionServer responsible for the HRegion and issues a read or write request. This location information will be cached in the client so that subsequent requests do not need to go through a lookup process. If the HRegion is reallocated by the main load balancer or because the HRegionServer has died, the client will re-query the directory table to determine the new location of the user area.

HMaster: The HMaster is the master server of the HBase cluster and is responsible for monitoring all HRegionServer instances in the cluster and is the interface for all metadata changes. In a distributed cluster, the HMaster usually runs on the NameNode. Usually, a cluster has multiple HMasters, only one of which is active and the others are in a hot backup state so that in the event of a failure, a new active HMaster can be quickly elected from the hot backup HMaster, making the system much more fault tolerant.

1. managing users' operations of adding, deleting, changing and checking Table tables.
2. managing the load balancing of HRegion servers and adjusting HRegion distribution.
3. responsible for the allocation of the new HRegion after it has split.
4. Responsible for the migration of HRegion on the failed HRegionServer server after the HRegion server is down.

Zookeeper: ZooKeeper (Animal Keeper) is so named because many of Hadoop's subprojects are animal names, including Hadoop which is the name of the founder's son's toy baby elephant. The name Animal Keeper also indicates the role that ZooKeeper plays in Hadoop as a coordinating manager. As can be seen from Fig. 4.11, the Zookeeper is key to contacting the client and the Hmaster. zookeeper acts as the

cluster coordinator for HBase, coordinating the handling of the operation of HBase. zookeeper has the following main roles for HBase.

(1) System fault tolerance ZooKeeper registers each new HRegionServer that joins the cluster and listens for HRegionServer status information. When an HRegionServer fails to send presence status information for a specified period, ZooKeeper deletes the node information and notifies the HMaster.
(2) The election HMaster is responsible for selecting the backup HMaster as the active HMaster in the event of an active HMaster failure, preventing the cluster from being out of action for an extended period.
(3) The HRegion metadata information is stored in the Meta table and the Mate table is stored in ZooKeeper. RegionServer failures, etc., and can feed this information back to the client effectively.

RegionServer HRegionServer is the main component in HBase, responsible for the actual reading and writing of table data and managing Regions. In a distributed cluster, HRegionServer is usually on the same node as the DataNode, to achieve data locality and improve read and write efficiency. The main functions are as follows.

1. RegionServer status is reported to HMaster regularly, including RegionServer memory usage status, online status and HRegion etc.
2. Manage HRegion, perform Flush, Compaction, Open, Close, Load and other operations
3. Managing Hlog
4. Perform data insert, update and delete operations
5. Metrics: externally provides parameters to measure the status of HBase internal services
6. Built-in HttpServer, providing access to the RegionServer interface.

Refer to the Hadoop documentation HDFS Architecture for more information.

4.3 Graph Store

What is the concept of a Graph? What is a graph database? Why do I need a graph database? What are the differences between graph databases and traditional relational databases? What problems do graph databases help solve that are difficult or impossible to solve with relational databases? When you see the term “graph database”, you will probably ask these questions. This chapter will answer these questions in a way that is easy to understand and will enable the reader to better understand graph databases. This chapter will also introduce neo4j, the current mainstream graph database, and present relevant application examples to give the reader a deeper impression of graph databases.

4.3.1 The Concept of a Graph

The concept of 'graph' in the graph database introduced in this chapter is also derived from graph theory, a branch of mathematics. A graph in graph theory is a figure consisting of several given points and a line connecting the two points, which is usually used to describe a particular relationship between certain things, with the points representing the things and the line connecting the two points indicating a particular relationship between the two things.

In short, a graph is a very common and important data structure that can be used to characterize the 'many and many' relationships between natural things. Graph theory is very common in the world of computer science, with a large number of algorithms, models and applications. In everyday life, public transport systems, social applications, navigation systems, etc. can be abstracted into graph data structures. Graph structures can be used to represent almost anything and the relationships between things. The most common graph model at present is the Property Graph, and the Neo4j database to be introduced in this chapter also makes use of the Property Graph model.

4.3.2 Property Graph

The property graph is the dominant graph database model and is the focus of this chapter. A property graph consists of two elements: nodes and edges. Each node represents a natural entity, and nodes can have their attributes and have one or more labels. Edges between nodes represent relationships. Edges have a direction and can be unidirectional or bidirectional. In addition, edges can also have attributes. As shown in Fig. 4.12, the three circular nodes and the two square nodes represent five natural entities. The circle nodes have their own attributes, with attributes such as name, age, grade, and gender. Also, the circular node has a student label. The square node has its attributes such as product name, price, color and brand. In addition, the square node has a stationery label. The relationship between the circular node and the circular node is "friend", and the relationship between the circular node and the square node is "purchase". The use of attribute maps allows for a good abstraction of the connections between natural things and the creation of data models for various types of scenarios. These relationships are abstracted through attribute diagrams and stored in a database, so a variety of data relationship queries can be satisfied.

With the accelerated application of Internet technology, huge scale, complex structure and diverse query requirements of graph data have emerged in the fields of online social networking, traffic navigation, online shopping and movie recommendation. In terms of online social networking, Facebook, the world's largest social networking platform, has hundreds of millions of active users, of which the need to store the edge of the relationship between users' friends has reached the level of a hundred billion. These relationships can be used by websites to recommend users to whom they might

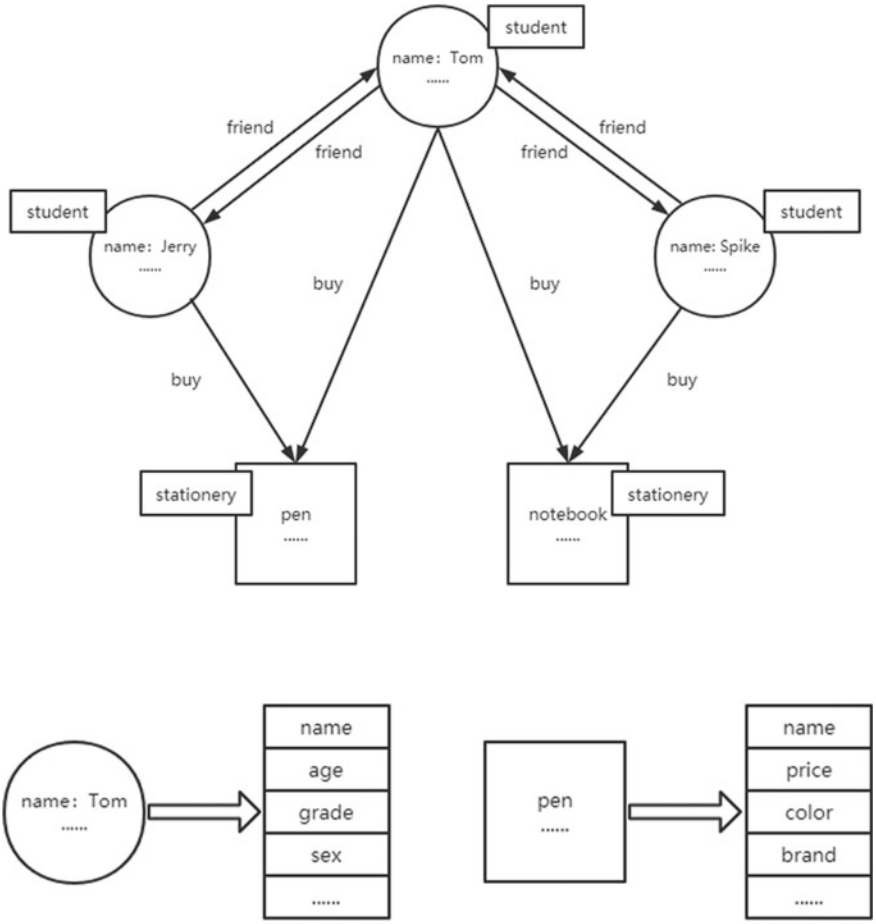


Fig. 4.12 Example of a property map

meet, enriching and expanding their social networks. In online shopping, shopping platforms such as Taobao, Jingdong and Amazon generate a huge number of transactions every day, which contain connections between customer nodes, connections between product nodes, and connections between customer nodes and product nodes. Abstracting these transactions into graph data structures and using graph databases to store, analyze and recommend them can help improve the customer shopping experience. This same approach can be applied in the film recommendation domain by abstracting directors, actors, and each film into node entities. The relationship between the director node and the film festival points is an edge, and the relationship between the actor node and the film festival points is an edge. Imagine a scenario where a film is liked by a user, and a new film is created by the lead actor and director, and the two films may have similar attributes in terms of lead actor, genre,

and style, so the probability of the user liking the new film is high. There are many more scenarios for the use of attribute maps, so the reader can think about them, and I think the reader should have a general understanding of them by now.

4.3.3 Graph Database

A graph database is a database that aims to treat the relationships between data as if they were equally important to the data itself. It aims to preserve data without restricting it to a predefined model. Instead, the data is stored in a way that demonstrates how each entity is related or interrelated to other entities. In summary, the graph domain is divided into two main parts: the graph database and the graph computation engine. Graph databases, like traditional relational databases, are used for On-Line Transaction Processing (OLTP) and can be accessed by applications in real-time. The graph computing engine is similar to Big Data analytics and is used for On-Line Analytical Processing (OLAP), which is suitable for decision makers to analyze data for decision making.

A graph database, short for a graph database management system, is an online graph database management system that supports the Create, Read, Update and Delete (CRUD) approach to the graph database. Graph databases are generally used in transactional (OLTP) systems, so 2 additional characteristics need to be considered when looking at graph database technology.

(1) Basic storage: Basic storage is divided into native graph storage and non-native storage. Native graph storage means that the graph data stored in the database is optimized and specifically designed for storing and managing graphs. Examples of such graph databases are Neo4j, OrientDB, etc. Non-native databases are those that serialise graph data and store it in other databases. Examples of such graph databases are Titan, InfiniteGraph, etc.
(2) Processing engines: Some definitions require graph databases to use index-free adjacencies. This means that the associated nodes are physically pointing to each other in the database. Each node in a database engine that uses index-free adjacency maintains its reference to neighbouring nodes. Each node, therefore, behaves as a micro-index of its neighbouring nodes, which is much less costly than using a global index. This means that query time is independent of the overall size of the graph; it is only proportional to the number of graphs searched. The graph database storage and processing model is simpler and more expressive than other databases. In contrast, a non-native graph database engine uses global indexes to connect individual nodes. These indexes add an indirection layer to each traversal and therefore lead to greater computational costs.

Let's take a look at an example of how graph databases can be used in comparison to traditional relational databases. Suppose a company needs to find out which department an employee Tom currently belongs to to make a personnel transfer. If the query is carried out according to a traditional relational database, three tables need

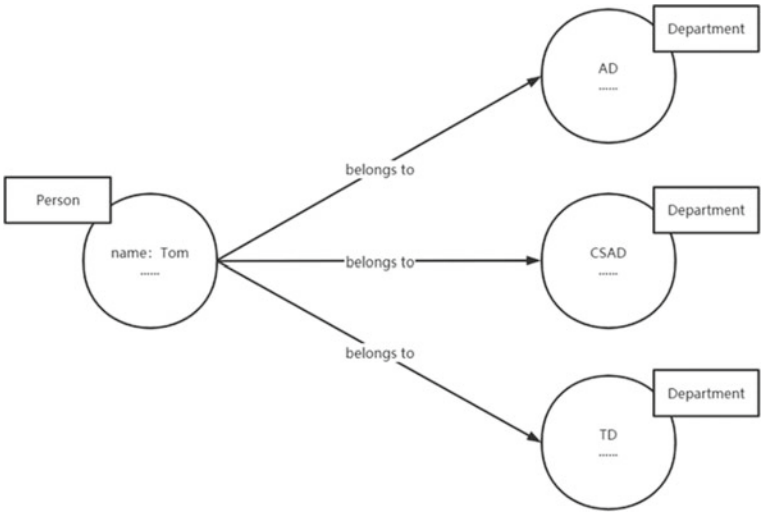


Fig. 4.13 Department and employee relations information sheet

to be created: the employee information table, the department information table, and the employee-department correspondence table. This is shown in Fig. 4.13.

At this point, the search requires a three-step process.

Step 1: Querying Tom's job ID from the employee information table.
Step 2: Querying the corresponding Department ID from the Work ID obtained in step 1.
Step 3: Querying the department name information from the department ID obtained in step 2.

Analyzing these three steps, it can be seen that the first query requires one index lookup. The second query also requires one index lookup, and in the third step, the situation is very different, requiring three index lookups. If the company has a small number of employees, a traditional relational database can achieve this query through a join operation. However, for larger companies, which may have hundreds of thousands of employees, this query requirement will put pressure on a traditional relational database. The next section shows how a graph database can solve this problem. Figure 4.14 shows how a graph database stores data.

For the graph database, the data model is clear at a glance and the query requirement can be well solved by creating employee nodes and department nodes. The employee node includes attributes such as name, and job ID and d has a person tag. The department node includes attributes such as department ID, and department name and has a department tag. The "belongs to" relationship is established directly

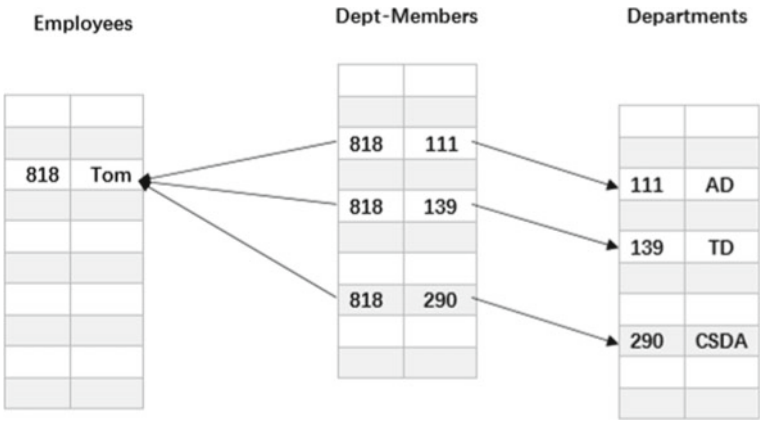


Fig. 4.14 Example of how data is stored in a graph database

between the employee and the department. The search process is also divided into three steps.

Step 1: Create a global index on the employee tag person to query the employee node to which employee Tom belongs.
Step 2: Querying the relationship department node with the edge label 'believes to' on the employee node obtained in step 1.
Step 3: Reading the department name attribute on the department node obtained by the query in step 2.

Now to briefly analyse the step-by-step comparison of the two queries: for the first query, both queries are equally efficient. For the second query step, the relational database will index the dept-members table, while the graph database will fetch directly from the employee node. Fetching the relational node from the employee node with the "believes to" edge is much more efficient than indexing the dept-members table. For the third step, the graph database works better.

A simple experiment is presented here, with more intuitive experimental data to illustrate the efficiency shown by graph databases compared to relational databases. In social networks, users are related to each other as friends, colleagues, couples and lovers. The experiment uses a population of one million households, each with approximately 50 relational users, and the task is to query the friendships of friends with a maximum depth of 5. Neo4j was chosen for the graph database to conduct experimentation with a relational database.

In response to the above experimental results, the reader can feel that the graph database is far more efficient than traditional relational database queries for this query requirement. In the face of huge amounts of data, relational databases are exposed to the problem of not being able to use join queries to handle the relationships between query things, which is a very "expensive" operation involving a lot of I/O operations and memory consumption. Graph databases can solve this problem efficiently, as

David Meza of NASA says: "I love Neo4j because I can explore relationships faster than you can say SQL JOINs". Of course, this is only the first great use of graph databases.

In addition to the obvious performance benefits, graph databases show great flexibility and scalability compared to traditional databases. This is because graph data structures are inherently flexible and extensible. This is demonstrated by the fact that new edges, nodes, labels and subgraphs can be added to an existing graph structure without breaking existing queries and application functionality. It is not easy to achieve with traditional relational databases, because the data model is established at the beginning of the design of a traditional relational database and the need for high reliability and consistency causes relational databases to not easily scale horizontally or vertically. The high flexibility and scalability of graph databases allow those involved to design a final and complete data model without being forced to do so without knowing the true shape and complexity of the data and to change and optimize the data model as requirements change.

On the other hand, some businesses are inherently flexible and changeable. Using a graph database (or other NoSQL database such as MongoDB) can quickly keep up with changes in the business without costly administrative operations such as Schema changes.

It is important to note that graph databases are independent of the total size of the dataset and are not only good at managing highly connected data but are also suitable for complex queries. Using only a schema and a set of starting points, graph databases can explore adjacent data around these initial starting points—collecting and aggregating information from millions of nodes and relationships, and keeping any data outside the scope of the search intact. Accessing nodes and relationships in the native graph database is an efficient, constant-time operation that allows you to quickly traverse millions of connections per kernel per second.

Secondly, the graph database does not use a traditional SQL language as a "Create, Read, Update and Delete (CRUD)" language, this is because traditional SQL languages are not suitable for multi-layer correlation analysis queries. The graph database specifically uses a query language for graph retrieval, which is more efficient than traditional SQL languages. Examples include Gremlin, Cypher, etc. The graph query language greatly facilitates the ongoing development of the association analysis business. Traditional solutions often have to modify the data storage model and modify complex query scripts when requirements change, graph databases have abstracted the business expression and provide basic graph operations APl, which is very convenient.

For example: 2 layer depth friend query, Gremlin language implementation:

g.V(me).out('friend').out('friend').

if you need to change to 2 layer depth classmate query, then adjust the friend for classmate:

g.V(me).out('classmate').out('classmate').

Finally, graph databases also provide professional analysis algorithms and tools. For example, ShortestPath, PageRank, PersonalRank, Louvain, etc. Many graph databases also provide data batch import tools and provide a visual graph display interface, making the analysis results of the data more intuitive to display.

4.3.4 Neo4j

Neo4j follows the property graph data model, focusing not only on data but also on data relationships, and is a high-performance, highly scalable native Nosql graph database. By using Neo4j, developers can build intelligent applications to traverse today's large, interconnected datasets in real time.

Neo4j is powered by a native graphical storage and processing engine that provides an intuitive, flexible and secure database. It stores structured data on the network rather than in tables. It is an embedded, disk-based, Java persistence engine that supports ACID transaction features. At the same time, Neo4j provides methods for visualising data, such as the Neo4j Viewer for developers, the Neo4j Bloom for analysts and others seeking natural language search, and libraries for developers to embed graphics directly into their applications.

Each version of the Neo4j graph database has a community edition and an enterprise edition, of which the community edition is free and open source based on the GPLv3 agreement, but is limited to standalone deployments and has limited functionality. Neo4J is not a distributed database and scalability is not an advantage, but it supports CSV data import online and Neo4j import offline. Neo4j is a native graph database and also has the capabilities of a graph analysis engine.

Neo4j is developed in Java and was originally designed to be based on the Java domain. at the heart of Neo4j is the Java traversal API, which is powerful, flexible and easy to use, designed specifically for graph databases. neo4j is essentially a JVM-based product, which means that theoretically, any programming language that supports the JVM can use Neo4j using a specific library. for example, Java, Spring, Scala, etc.

Neo4j diagram database is more mature than other diagram databases, rich, complete and powerful, and is currently the most mainstream diagram database. According to the DB-Engines Ranking database engine ranking, Neo4j is ranked first among 32 graph databases participating in the popularity ranking https://db-engines.com/en/ranking_trend/graph+dbms. This is shown in Fig. 4.15.

(i) **Neo4j system architecture**

The following describes the Neo4j system architecture, as shown in Fig. 4.16. The Neo4j database system is deployed at the lowest level where the actual physical hard disk is used to store the data. The actual hard drive stores a large number of files, which can be divided into: node files, relationship files, attribute files, index files, transaction log files, etc. We can therefore roughly calculate the space required for the data of a particular Neo4j project as follows.

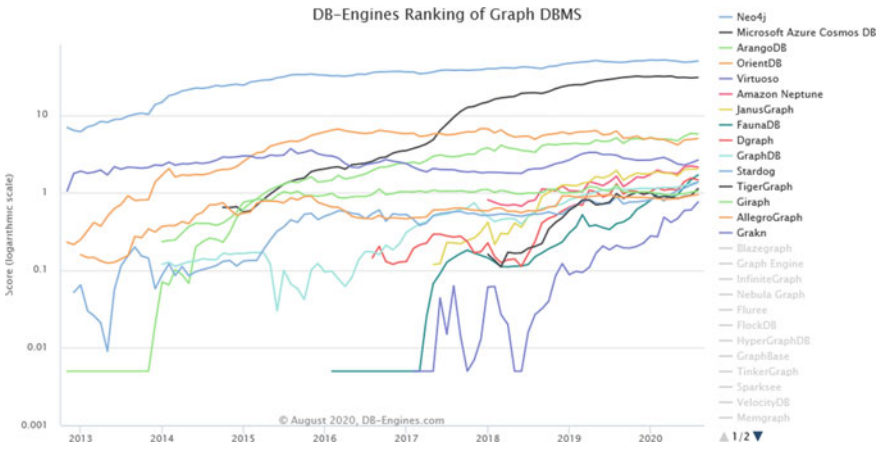


Fig. 4.15 Graph database ranking

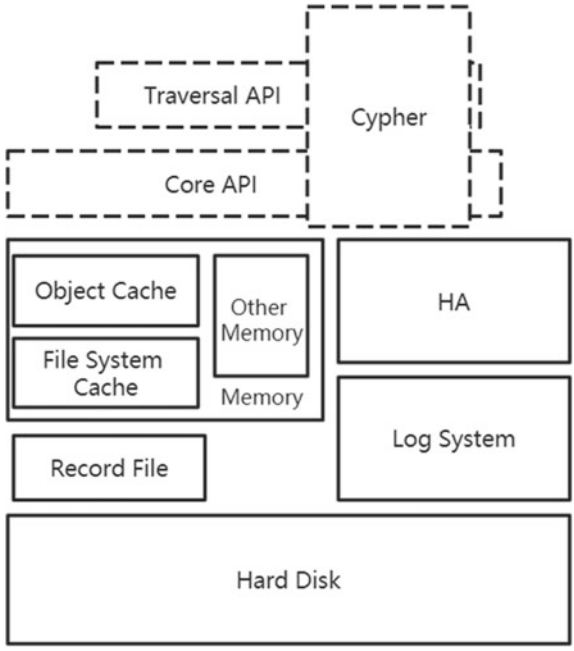


Fig. 4.16 Neo4j's system architecture

Neo4j hard disk space required for a particular project data = number of project nodes * storage space required for a single node + number of project relationships * storage space required for a single relationship + number of project attributes * storage space required for a single attribute + index storage space + transaction log storage space + other space (smaller).

The log files are specially designed and optimized to store nodes, relationships and attribute pointers to actual hard disk addresses. To support the Neo4j graphical search engine for fast traversal lookups. The core approach to fast traversal lookups is the index-free proximity principle, which means that a single node only finds all of its neighbors that satisfy the conditions, regardless of a large number of nodes. For the Neo4j database, it is not feasible to use the hard disk alone for data reading, as the speed of reading data from the hard disk is not sufficient for high performance, and fast querying, and is limited by the speed of disk reading and system I/O capability, so it is necessary to use memory that supports fast data reading. When running a Neo4j system, there are several types of data in memory: file system cache, object cache and other caches (mainly reserved for the operating system).

The file system cache can be thought of as a small percentage of commonly used data being read and written from the hard disk to memory first, reducing the number of accesses to the hard disk and increasing the speed of data access. When the system is running, if file data such as node files and attribute files need to be called, they are first sought from the file system cache, and if they do not exist, then the data is read from the hard disk to the file system cache. Data in the file system cache is written to the hard disk when the conditions are met, and these read-and-write operations are managed by system control.

Object caching is also used as a strategy for fast access to data. As mentioned above, the Neo4j database system is a JVM-based application that can use the methods used to create objects in Java to handle nodes and relationships, and the core Neo4j API to quickly access traversed data, rather than just working with raw files.

The logging system and HA are used to ensure that data is reliable and not lost. the high availability feature of HA provides the ability to customise hardware as well as hardware failure rescue. The logging system records a transaction file which, under certain conditions, is written from memory to the hard disk for persistent storage. This file ensures the ability to recover data in the event of a system crash by re-executing the operations in the file. Of course, as with the Redis database persistence in the previous chapter, when the system runs out of memory and needs to be re-run, the latest operations in the file are not written from memory to the hard disk and will disappear with the memory, and this part of the operation, as well as the data, will be lost and cannot be recovered.

Finally. At the top of the Neo4j system architecture are the core API, the traversal API, and the Cypher, which is used to interact with the underlying data and is highly abstracted, providing explicit instructions to interact with the underlying graph and providing the basis for the traversal API. The traversal API can be found in the section on graph algorithms, and the Cypher is described in more detail below.

(ii) **Neo4j database indexing**

Neo4j can quickly find the value of a given node or relational attribute because of the data indexes set up in Neo4j. In a relational database, indexes are created to find the information about the row you are looking for quickly and efficiently. Similarly,

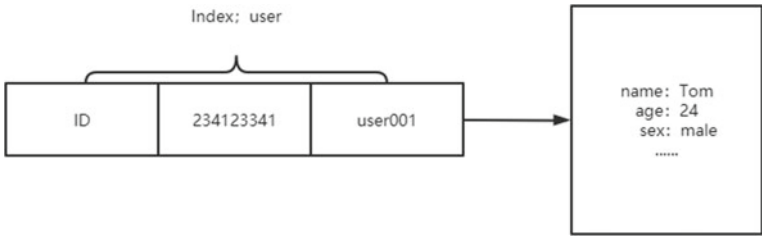


Fig. 4.17 Example of index item construction

in Neo4j, there is a special tool, IndexManager, to create an index entry for a newly added node so that the node can be quickly queried later. An index entry in Neo4j is essentially a pointer to a node or relational attribute value. The value in a newly constructed index entry in Neo4j consists of three pieces, the index key, the value of the index and the node to be indexed. How do you create an index item? An example is given below. In a social network, the user use001 node is logged in with a unique ID number, e.g. ID: 234123341. If we add this ID user to the Neo4j database, then the new index item is constructed: ID for the index key, 234123341 for the value of the index, and user001 for the node to be indexed. As shown in Fig. 4.17, the constructed index item can be seen as a pointer to the value of the user001 node attribute. This newly constructed index item is stored by Lucene, the default index implementation component in Neo4j.

In practice, the key in an index item may not choose a unique ID value as above, or it may choose the age attribute value or some other key value that is not unique as the index item. As shown in Fig. 4.18, if the age attribute is chosen as the key for an index item, then the index may point to many nodes with equal age values. To query all nodes with a certain value of the age attribute, you can simply quickly find the index item with a certain value of the age attribute and access all the nodes associated with it.

For databases, creating indexes increases the efficiency of data retrieval, but also brings with it a certain cost of building and storing indexes. The costs and benefits of building indexes are measured against the needs of the actual application and there is no absolute good or bad. For small amounts of data, it is best not to use indexes. For specific application needs, as discussed above, there are advantages to using either the ID attribute or the age attribute. In general, once an index has been constructed in a database, it is best not to change it or to keep the number of changes to a minimum. there are two types of indexes in Neo4j: b-tree (B-tree indexes) and full-text (full-text indexes). b-tree indexes are good at exact lookups for all types of values, as well as range scans, full scans and prefix searches. The full-text index differs from the B-tree index in that it is optimized for indexing and searching text, allowing the user to

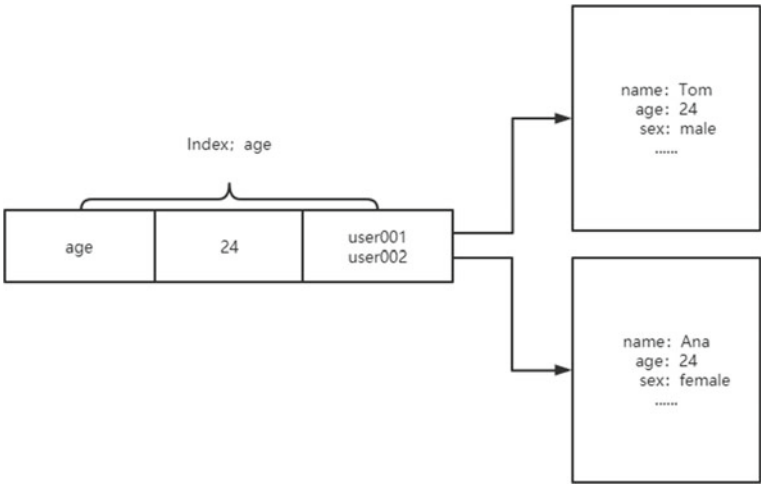


Fig. 4.18 Example of multiple nodes matching the same index item

write queries that match the content of the index string attributes. Also, approximate matches can be returned for a given query in addition to any exact matches.

(iii) **Query language—Cypher**

Cypher is Neo4j's graphical query language, a developer-friendly declarative query language that allows users to store and retrieve data from graphical databases and also incorporates the functionality of other standard data access languages. The language uses ASCII-Art to represent visual graphical patterns for finding or updating data in Neo4j. There are currently several ways to execute the Cypher language for Neo4j queries. Downloading the Neo4j installation package comes with the Neo4j shell, and a web management console (graphical web window), both of which can connect to the Neo4j database and execute Cypher commands to perform database-related operations. It is also common to be able to connect to Neo4j via a programming language using a specified database driver and use the Cypher language to perform database-related operations.

Cypher is not only the best way for data to interact with Neo4j, but it's also open-source! This is much easier than writing complex Java program queries to access the Neo4j database. Cypher, although based on SQL functionality, is specifically optimised for graphics and has a clear and concise syntax, allowing users to easily write all normal CRUD operations in a simple and maintainable way. Cypher provides advanced queries including data aggregation (similar to SQL), function queries and chained queries. Interested readers can read the official documentation to learn https://neo4j.com/developer/cypher/guide-sql-to-cypher/.

Chapter 5
Security Management on Big Data of Business

In recent years, big data technology has been widely used in many ways of daily life, including of business intelligence, medical service, etc. Big data technology can be used to store and analyze valuable historical information of consumers, so as to strengthen the understanding of customer's needs, better grasp the market trend, help enterprise decision makers to better formulate scientific and reasonable marketing strategies, and achieve accurate and personalized recommendation programs. Although big data technology has broad applications, there are still many data security problems that need to be solved: For the consideration of data privacy, it is difficult to break the barriers between different e-commerce platforms, thus forming many "data silos"; Second, due to the electricity business enterprise storing a lot of customer's information, such as user id, contact information, credit card numbers and consumption records, the data leakage will result to a big impact on consumers.

In 2018, with the help of an intermediary, amazon employees provide internal data and other confidential information to the external institutions; in the same year, Tencent Cloud responded to a silent error caused by the firmware version of the physical disk, that is, the data written and read is inconsistent, which damages the system metadata. In addition, the leakage of the enterprise's operational data will also cause great economic risks to the enterprise. Therefore, in the era of big data, how to protect consumers' privacy and ensure the security of enterprise data has become a difficult problem for e-commerce enterprises. Finally, with the continuous development of blockchain technology, due to its decentralized, untamperable, whole-process traceability, openness and transparency, it is suitable for solving problems such as data traceability, distributed storage and trusted computing in e-commerce big data applications, and has attracted increasing attention. Therefore, solving the problem of business data security can enhance the confidence of e-commerce enterprises in big data platform, which is of great significance in the era when big data has become the basis of intelligence business service. This chapter mainly introduces some big data security technologies such as data traceability, privacy protection technology, cryptography and blockchain.

J. Cao, *E-Commerce Big Data Mining and Analytics*, Advanced Studies in E-Commerce,
https://doi.org/10.1007/978-981-99-3588-8_5

5.1 Traceability Technology of Business Big Data

5.1.1 The Definition of Data Traceability

Traceability, which means to find along the upstream and trace the origin, has been applied to the field of information science in recent years. Like the artificial intelligence, Internet of things, etc., the value of "data" has become increasingly prominent, and the data industry has also increasingly become a national basic and cutting-edge pillar industry. However, in the current information construction, various units and systems are separated from each other. Data openness is a good choice, but without effective data governance will cause unacceptable consequences. For example, the data user needs to verify the authenticity of the data and ensure the credibility of data in order to make better use of data. The emergence of data traceability technology is needed to realize the historical analysis of dissemination path and production chain. Data traceability was first used in data warehouse systems, mainly in areas requiring high data authenticity, such as biology, archaeology, medicine, etc. However, with the rapid development of the Internet and the frequent occurrence of network deception, people have higher requirements for the authenticity verification of data authenticity. Therefore, data traceability technology has been applied into more application fields and set off a wave of research upsurge.

Data traceability technology is also known as data kinship or data pedigree. This technology tracks the history, status and change process of data according to the data path, so as to achieve the positioning of abnormal data and causes.

5.1.2 The Definition of PROV

Metadata, also known as intermediary data which means the data's description information can be used for electronic data catalog. Metadata can be used to describe the attributes information of data, record, locate, identify and evaluate the Internet resources for the purpose of data storage, document retrieval or historical traceability. In the field of data traceability, metadata can be used to describe the origin and the flow of data. PROV is the most widely used metadata model for data traceability (Fig. 5.1).

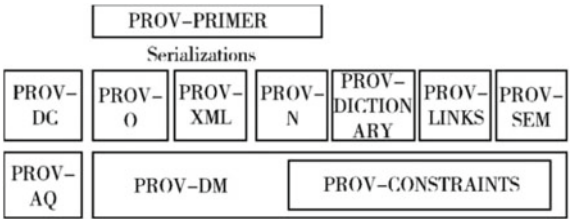


Fig. 5.1 PROV standard

At present, the most widely used model for data traceability was the PROV standard launched by the World Wide Web Consortium (W3C) based on the open origin metadata model in 2013. PROV series standards contain 12 documents, including PROV-O: main body description, PROV-DM: model concept, PROV-CONSTRAINTS: model constraints and other items.

PROV-O's main contents include of entities, activities and agents:

Entity: entity is the description of some conceptual things that objectively exist and can be understood in the world. For example, a photo or a table can be regarded as an entity of different origins when describing from different aspects and states. Such as different versions of a data sheet. Entities are often static individuals and do not have the initiative of behavior. It is the operating object of the agent through the activity. In PROV-O, Entity (E) is used to represent an entity.

Activity: refers to the behaviors that leads to the change of entity states, such as the modification, deletion, insertion and other operations on a document. Usually, the activity of an entity will make the entity become a new entity. In PROV-n, use Activity (a) to represent an activity.

Agency: agency refers to the individual who bears responsibility in the objective world. Usually, an agent is a person or explicitly defined as a software, such as Excel office software and eclipse IDE. Agents usually change the state of entities through active behavior. The relationship between agents, activities and subjects can be compared to human beings, labor tools and labor objects. Human beings transform labor objects through labor tools. In PROV-n, **Agent** (Ag) is used to represent an activity.

In PROV-DM, directed edges are used to represent the relationship between entities. Because there are many complex causal relationships among subjects, agents and activities, there are seven kinds of dependencies summarized as below:

Usage: usage relationship is the dependency relationship between entities and activities, which indicates the behavior of an activity to use entities to achieve a certain goal. For example, the modification of a document. The relationship can be expressed as used (a, e), where a represents behavior and e represents used entity.

Generation: the generation relationship is the dependency relationship between entities and activities, which indicates that a certain activity generates a new entity. For example, the modification on an old document produces a new document. This relationship can be expressed as was generated by (E, a), where e represents the newly generated entity and a represents the activity.

Communication: communication is the dependency between activities, which means that some unspecified entities are shared between two activities. For example, after a document is created, it is modified another party. Then only the created document can be modified, and the modification behavior depends on the creation behavior. This relationship can be described as was informed by ($a2$, $a1$), where a1 affects the $a2$ relationship.

Derivation: a derivation relationship is a dependency relationship between entities, indicating that a new entity is derived from an old entity. For example, an old document is modified to derive a new document. The difference is that the derivation relationship is the dependency between entities, and the generation relationship is the dependency between entities and activities. The derivation relationship can be expressed as was derived from (E2, E1), where the entity E2 is derived from E1.

Association: association is the dependency between an activity and an agent, indicating that the agent is responsible for the activity. For example, if users use Excel to modify the data table, there is an association relationship between excel as a proxy and the data table entity. This relationship can use was associated with (a, Ag) to represent the association between behavior a and agent AG.

Attribution: the attribution relationship is the dependency relationship between an entity and an agent, which means that the agent is responsible for an entity. When there is an attribution relationship between the entity and the agent, it can be inferred that the activity of the entity is triggered by the agent. The attribution relationship can be represented by was attributed to (E, Ag), where e belongs to entity AG.

Authorization: authorization relationship is the relationship between agents. It means the right of the agent to authorize other agents to engage in an activity. For example, a software user can authorize another software or user to access the data of this process, and all these agents are responsible for this activity in some form. The authorization relationship can be represented by acted on behalf of (ag2, AG1). Agent ag2 is authorized by agent AG1 as its principal.

In Fig. 5.2, we describe some simple historical operations of document 1 based on the PROV model: document 1 belongs to the agent "Zhang San", which means that Zhang San has ownership of document 1; Word software uses documents and performs "write" operations to derive "new document 1" from document 1.

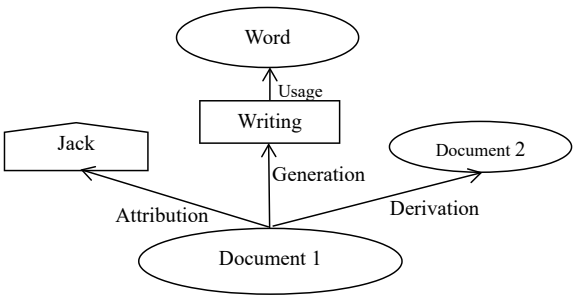


Fig. 5.2 PROV based document traceability model

5.1.3 The Constraint of PROV Traceability Graph

In addition to defining the above entities, PROV also defines basic constraints on entities and dependencies, so as to ensure the cross organizational versatility of the traceability graph and reduce the redundancy of the diagram. Specifically, it includes the following three constraints:

Constraint 1: the origin node is unique. PROV model describes all historical information of data evolution process, and each node has its time attributes. Even if there are two identical nodes, they will be different due to their different time attributes. Therefore, the origin node is unique.

Constraint 2: there is no circuit in the PROV traceability graph. The PROV traceability graph is a data structure that records the origin and historical evolution of data, so there is no loop.

Constraint 3: an entity node in the PROV traceability graph can only have one activity generated. Like entity nodes, activity nodes have time attributes, and in the operating system, even "concurrent" processes still have different sequence order. Therefore, there is no case that two activities produce an entity at the same time.

5.2 Privacy Protection of Business Big Data

Privacy is an important civil right that citizens to enjoy the personal information and private things. Although our constitution does not make clear protective provisions on the right to privacy, it indirectly confirms the inviolability of citizens' privacy in other aspects. Clause 38 of the constitution: "the personal dignity of citizens of the people's Republic of China is inviolable. Humiliation, slander, false accusation and framing of citizens by any means are prohibited." But in the era of big data and artificial intelligence, everyone's privacy in finance, shopping, medicine and so on is controlled under the powerful computing power. With the continuous development of big data technology, the continuous popularization of informatization in all aspects of daily life, and the high integration of the physical world and the digital world, if there is no effective privacy protection technology as a guarantee, our society will develop into a society without any privacy.

However, traditional information security mainly studies the loophole of information system, such as virus detection, Trojan horse detection, etc., which cannot guarantee data security from the root. With the continuous development of information security technology, computer scientists have realized the need to protect information from the data itself. Especially in the era of big data, a large amount of data has been generated. The data is separated from the traditional physical entities like the water and electricity, and the physical boundary is becoming more and more blurred. Technologies such as differential attack and relevance analysis for big data

continue to appear. Traditional privacy protection technologies such as data encryption cannot meet the new security requirements. This section mainly introduces the technologies related to big data privacy protection.

5.2.1 Data Desensitization Technology

One of the characteristics of the big data era is information sharing, certain organization will no longer monopolizes digital resources. Data desensitization is to deform some privacy information in the raw data, such as personal identity, so that malicious attackers cannot obtain sensitive information from the desensitized data. Table 5.1 is a list of common desensitization algorithms. Data desensitization has been widely used in many fields such as medical treatment, finance, electric power, e-commerce and so on. For example, before big data analysis on the e-shopping platform, it is necessary to desensitize and protect sensitive data such as the user's ID number, mobile phone number, bank card number and user ID.

According to the process of data desensitization, it can be divided into two types: dynamic desensitization (DDM) and static desensitization (SDM). Static desensitization refers to desensitization of data when it is stored. It is generally used in non-production environments, such as development, testing, outsourcing and data analysis; Dynamic desensitization is to desensitize the stored data before use and store it in plaintext. Dynamic desensitization is suitable for different users to use different desensitization strategies, which is more flexible. For different privacy sensitivity levels, they can be generally divided into five levels: L1 (public), L2 (confidential), L3 (confidential), L4 (top secret) and L5 (private). The higher the level of data application, the more stringent the approval process, the longer the cycle, and the greater of the difficulty.

Table 5.1 Common data desensitization algorithms

Method	Description
Mask	Use * and other wildcards to replace the symbols that need to be hidden to ensure that the data length remains the same, and the original format length will be retained, which is generally used for mobile phone numbers, ID number, etc.
Rounding	Integer data such as age, income and date
Replace	Use the dictionary to replace, for example, Alice is replaced with A, Bob is replaced with B
Truncation	Truncate the string and retain only part of the information
Encryption	The encryption algorithm is used to deform the data, and the security depends on the encryption algorithm

5.2.2 *Differential Privacy Protection*

Data desensitization technology can ensure the privacy protection of statistical information. However, in addition to obtaining data directly, attackers can also obtain useful information through differential attacks. The principle of differential privacy technology is to prevent data from being speculated by adding noise interference to the source data. To put it simply, it is useless for you to infer more data by obtaining some data. Let's take a look at the following example. Table D shows whether five people buy a good (Table 5.2), in which 1 means yes and 0 means no. Suppose the attacker wants to know about E's information, but the attacker can not directly query, and can only count the total data of certain users. The attacker suspects that the data sum of the first four users and the first five users are obtained respectively, and then subtracted. For example, in this example, Q5 (d) = 3, Q4 (d) = 2, and the obtained difference is 1, thus obtaining the analysis result.

Here we introduce a concept of sensitivity,

$$\Delta \mathrm{f} = \max ||\mathrm{f}(D_1) - \mathrm{f}(D_2)||_1$$

where, D is a data set, $||\cdot||_1$ representing the Manhattan distance, and f () is a query function. In the database example above, if we think f is the query function Qi, then the sensitivity of the function is 1, because changing any entry in the database will cause the output of the function to change 0 or 1.

By adding randomness to the data, differential privacy makes part of the data in the sample have no impact on the overall output probability, and reduces the sensitivity of the query results, so that it can resist the attack. According to the type of differential noise, it can be divided into, Laplace noise (i.e., noise conforming to Laplace distribution) and exponential mechanism. At present, differential privacy is applied in application scenarios such as recommendation systems, social networks and location-based services to protect the user's privacy.

Table 5.2 Consumption statistics

Name	Whether to purchase commodity
A	1
B	1
C	0
D	0
E	1

5.2.3 K-anonymity

K-anonymity is a general privacy model, which is not limited to location privacy protection. The core idea is to replace the precise location of the user with a fuzzy spatial area, and there must be at least k different users in this area. Thus, the user in proposed LBS (location-based service) system can be indistinguishable from other $k-1$ users, thus achieving the purpose of location privacy protection. The method is similar to differential privacy. Differential privacy protection is achieved by adding noise. K anonymous users may be randomly generated noise or other similar legal users.

A secure k-anonymity algorithm should guarantee the following three points, that is, the attacker cannot determine whether a user is in the public data set, given a user and a sensitive attribute, the attacker cannot determine whether the user owns the attribute, and the attacker cannot determine whether a piece of data belongs to a user.

For the above K anonymity standards, generalization algorithms are generally used to process data tables, which can be divided into global generalization and local generalization. Global generalization is to generalize the entire attribute column, which is easy to cause large loss of data accuracy. Local generalization generalizes different elements in the attribute column to different levels, avoiding excessive generalization and reducing data loss. Figure 5.3 is an example of generalization of e-commerce platform's user shopping data to achieve anonymity. We can see that through the continuous generalization of the data (the specific age is changed to "adult" and "minor", the last few digits of the ID number are replaced by *, and the classification of purchased goods is generalized to the commodity category), the privacy of individual users is protected to a great extent, while the results of data analysis are not affected. The specific generalization accuracy can be dynamically adjusted according to different data usage needs.

5.3 The Data Sharing of Commercial Big Data

In the application of commercial big data, a large amount of data generated by various services are stored, used and shared in the background, and the data security risk generated in the process of data sharing is particularly obvious. Especially with the popularity of enterprises on the cloud, more and more commercial enterprises give up building their own data centers and choose cloud service as the carriers of data storage and application deployment, which leads to the possibility of data leakage; Secondly, artificial intelligence applications rely on a large number of data samples to train the model, while the traditional centralized machine learning model needs to obtain a large amount of data accumulated in various industries. However, due to the needs of trade secrets and user privacy protection, this large-scale data sharing is difficult to become a reality, resulting in a large number of "data islands", Therefore, distributed secure multi-party computing and other technologies have also become a

No	Gender	Age	Identify No	Commodity
1	male	56	1234500000	salt
2	male	42	1234500001	soy sauce
3	female	23	1234500100	sneakers
4	male	31	1234500900	cigarette
5	male	31	1234502013	wine
6	female	16	1234503100	cage

age

No	Gender	Age	Identify No	Commodity
1	male	adult	1234500000	salt
2	male	adult	3214500001	soy sauce
3	female	adult	1234500100	sneakers
4	male	adult	1234500900	cigarette
5	male	adult	3214502013	wine
6	female	underage	1234503100	cafe

Identify

No	Gender	Age	Identify No	Commodity
1	male	adult	123450****	condiment
2	male	adult	123450****	condment
3	female	adult	321450****	Clothes & Accessories
4	male	adult	123450****	Tobacco and alcohol
5	male	adult	321450****	Tobacco and alcohol
6	female	underage	123450****	Drinks

commodity

No	Gender	Age	Identify No	Commodity
1	male	adult	123450*****	salt
2	male	adult	123450*****	soy sauce
3	female	adult	321450*****	sneakers
4	male	adult	123450*****	cigarette
5	male	adult	321450*****	wine
6	female	underage	123450*****	cafe

Fig. 5.3 Anonymity and generalization process

new mode of "invisible sharing" of big data, realizing the "availability but invisibility of the data".

In the big data applications, data ownership can be divided into data owners, data centers and data users.

Data owner: the data owner is the data producer, collecting data from various applications and sensors deployed on the client. The data owner has the ownership of the data and needs to protect the privacy for the generated data.

Data user: the data user is the data consumer. The data user needs to obtain accession rights from the data owner or the data center, and can legally access the data after obtaining the security authorization.

Data center: the data center is the physical entity for a large number of users' data. The data center generally is the background server of the different applications. With the rising of the cloud computing technology, public cloud has gradually become the data collection center of the whole society, serving as the role to complete the data storage and data sharing.

According to the mode of data sharing, the big data sharing mode can be generally classified into the following categories:

1. **Data opening**. Data opening generally means that the government opens the data to the society. This method mainly includes non-sensitive data that does not involve personal privacy, and it needs to ensure that sensitive data will not be generated after secondary processing or aggregation analysis.
2. **Data exchange**. Data exchange mainly refers to non-profit data open sharing between government departments or between government and enterprises through signing agreements or cooperation. Generally, there are two methods: one is that the third party provides data exchange, which is suitable for data with low confidentiality; The other is to encapsulate sensitive data in closed-loop business scenarios to ensure that data is visible and unavailable.
3. **Data transaction**. Data transaction is mainly the sale of data with clear price. At present, many third-party data trading platforms in the market, such as big data trading market, provide this mode.

Next, we will introduce access control, zero trust, attribute encryption, homomorphic encryption and other big data security sharing technologies.

5.3.1 Access Control

Access control is an important means of data protection, which plays an irreplaceable role in the protection of information resources. The access authority system achieves the purpose of protecting data resources by restricting the operation authority of visitors. Generally speaking, the core of the access control system is a group of permission engines. This engine only answers one question: who has the permission to perform a certain action on a certain type of resources?

The traditional access control system is mainly based on the static experience and judgment of industry experts. For example, in a medical system, developers can obtain an access control scheme by discussing with hospital experts: for example, doctors can access all the historical data of their patients, pharmacies can query the prescription of each doctor, and the president can access all the data of the hospital. Although this access control system has played a good role in the protection of

traditional information systems, it is not suitable for large data systems with complex and changeable data volume and fast generation speed.

Generally speaking, the access control system includes the following entities:

(1) **Principal**: the principal is the initiator of the access control and the requester of the access resource. Generally speaking, it is a user or a user's initiating process.
(2) **Object**: an object is an accessed resource entity, which can be data, files, and hardware facilities or terminals that open to the outside world.
(3) **Access policy**: an access policy is a collection of access rules from one subject to an object. It specifies the authorities of the subjects to operate on another objects.

The basic purpose of access control is to ensure that the user data can be authorized to access legally, and prevent unauthorized illegal users accessing the network resources. Therefore, the primary task of access control is to authenticate the user's identity. In addition, it is also necessary to monitor and audit ultra vires and users' behaviors:

(1) **Identity authentication**: identify and verify the identity of the applicant visitor.
(2) **Control policy**: the control policy is the core component of the access controller, which ensures the usage of authorized resources by legitimate users and prevents unauthorized users from accessing resources by setting control rules reasonably; at the same time, legal users cannot access resources beyond their authority.
(3) **Security auditing**: the system needs to record all resource accessing events and maintain them as logs. So that all activities on the platform can be systematically and independently inspected in the future and give corresponding evaluation.

5.3.2 Zero Trust Architecture

Based on the traditional access control technology, "Zero Trust Architecture" has been received more and more attentions. Zero trust is established against the background of the gradual failure of the traditional network boundary. It advocates breaking the concept of the single network boundary and conducting comprehensive and dynamic access control for users, devices and applications. Generally speaking, the zero-trust architecture is applied in the distributed data center, and ZTNA is used as the front boundary of the data plane to isolate external access and internal resources, and a data channel is established between the external data requester and the intranet data server in the form of a proxy gateway based on the C/S mode. ZTNA control plane mainly consists of the following components:

Policy execution point (PEP): it exists as a media component between the resource accessing subjects and the data. As an access point of the external network, PEP accepts external data request events and sends the requests to the policy manager PA.

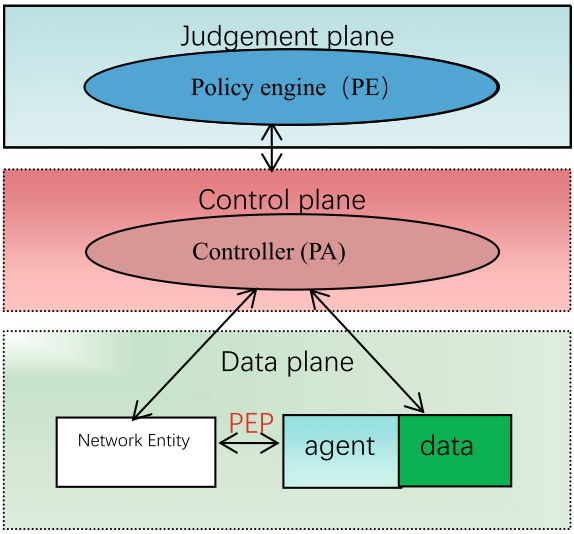


Fig. 5.4 Access control strategy diagram

Policy manager (PA): the PA accesses the policy engine PE according to the data request event sent by the PEP. If an authorization message is received, the PA generates an identity token or credential and sends it to the PEP (also in the form of a cookie) to authorize the user to access the data resources. It is worth noting that the token service generated at this stage adopts the single sign on (SSO) mode to ensure that one authorization and multiple services are available.

Policy engine (PE): the PE obtains the input information of trust decision, network security situation awareness intelligence, etc., conducts dynamic evaluation and access control policy decision in combination with the access control policy library, and finally outputs the decision of the authorizing or rejecting the access request to the PA.

The PE engine can evaluate the internal trust of users based on the identity authentication mechanism of PKI digital signature and the method of combining multiple factors of internal user trust. PE strategy refers to the idea of traditional IDS intrusion detection system and combines the misuse detection mechanism based on host behavior and network to build the behavior trust of network entities; based on the data traceability mechanism, the trust degree of user data usage is built, and the overall trust degree is finally obtained (Fig. 5.4).

5.3.3 Attribute Based Encryption

In addition to the traditional access control methods, attribute encryption technology can also achieve access control for users. ABE encrypts the data and formulates an

access structure that meets the requirements for decrypting data. Because of its excellent characteristics such as fine-grained access control and application prospects, it has become a hot spot pursued by many scholars.

Attribute based encryption, as the name implies, is an encryption technology based on user's attributes. Attribute encryption is also known as fuzzy identity encryption. Compared with identity encryption, it does not use explicit identity information as the public key, but divides users into different attributes, such as {teacher, associate professor, doctor} and other fuzzy information. Only users who meet the specified identity attributes can decrypt the encrypted text, so as to achieve the purpose of data access control. Generally speaking, ABE can be divided into the following two categories:

(1) Key policy attribute-based encryption (kp-abe). In kp-abe, the access policy is associated with the key. Generally, the decryptor formulates the access policy and limits the access resources. Only the ciphertext that conforms to the access policy will be encrypted. Such as a pay video on demand system;
(2) Ciphertext policy attribute-based encryption (cp-abe). In cp-abe, the access policy is related to the ciphertext, and the user defines the access policy. Only the user who meets the policy can successfully decrypt the data, which is generally used in the access control system of the cloud storage environment.

ABE is expected to be applied in many fields, such as cloud storage access control and tele-medicine systems, to achieve fine-grained access control of user's data.

5.3.4 Homomorphic Encryption

Similar to attribute encryption, homomorphic encryption is an encryption technology that can also be applied to data sharing. However, the operations in the ciphertext field can also be mapped to the plaintext field, so the goal of data "available but not visible" can be achieved. In the era of big data, the government, large enterprises and other institutions hold a large number of digital resources. However, due to the heterogeneous platforms and standard of database construction in different parties, it is difficult to connect and share data and form many "data islands". Homomorphic encryption has been widely used in many fields, such as multi-party privacy computing like federated machine learning. The research of homomorphic encryption technology started from the 1970s. Through continuous research by many scholars, homomorphic encryption has developed from single ciphertext operation to fully homomorphic operation supported with multiple operations.

Homomorphic encryption is an encryption algorithm that meets the certain properties. It can calculate a certain convention function on the ciphertext data to get an output, and decrypt the output into plaintext. The function of the calculation result for plaintext can be obtained, this property can be explained with the following formula:

$$\mathrm{Dec}(\mathrm{En}(a) \odot \mathrm{En}(b)) = a \oplus b.$$

Especially, the fully homomorphic encryption refers to an encryption algorithm that satisfies the properties of addition homomorphism and multiplication homomorphism simultaneous, and can perform any number of addition and multiplication operations. his property can be explained with the following formula:

$$\mathrm{Dec}(\mathrm{f}(\mathrm{En}(\mathrm{m}_1), \mathrm{En}(\mathrm{m}_2), \ldots, \mathrm{En}(\mathrm{m}_k))) = \mathrm{f}(\mathrm{m}_1, \mathrm{m}_2, \ldots, \mathrm{m}_k).$$

If $\mathrm{f}(\cdot)$ is an arbitrary function, it is called fully homomorphic encryption.

This feature is of great significance in practical applications. For example, the headquarters of e-commerce platform can perform homomorphic operation on the encrypted sales data to obtain the total sales amount in a certain region, without to obtain the real sales amount of each store, thus achieving a certain degree of privacy protection.

5.4 Blockchain Technology

Blockchain technology was firstly proposed by Nakamoto in 2008, which can provide a data tamper proof mechanism with decentralized management platform. The blockchain technology was firstly applied to Bitcoin to realize the trusted storage of transaction records.

Bitcoin is a kind of virtual currency. Bitcoin relies on the transaction history in the ledger, and the Unspend Transaction Outputs (UTO) is different from the traditional currency. Therefore, the distributed ledger is a collection of big data recorded the transactions. With the continuous recognition and understanding of blockchain technology, blockchain has also been applied into other fields to achieve the trusted storage of data in different applications. Figure 5.5 shows the information stored in one block. We can see that the information stored in one block header includes of the hash value of this block, the hash value of the previous block header, the difficulty value, timestamp, random number and the root node information. The random number is the result obtained by filling in the response difficulty value when mining the blockchain. The root node stores the verification value of all transactions in the entire blockchain. We will introduce the core principles and mechanisms of the blockchain technology in underlying algorithms.

5.4.1 Peer to Peer Network

Blockchain breaks the traditional C-S (client server) architecture and adopts a decentralized P2P network mode to deploy the system. In peer-to-peer networks, there are no absolute clients or servers. Compared with the traditional C-S architecture, each user node in the P2P network is both a client and a server, which can serve other

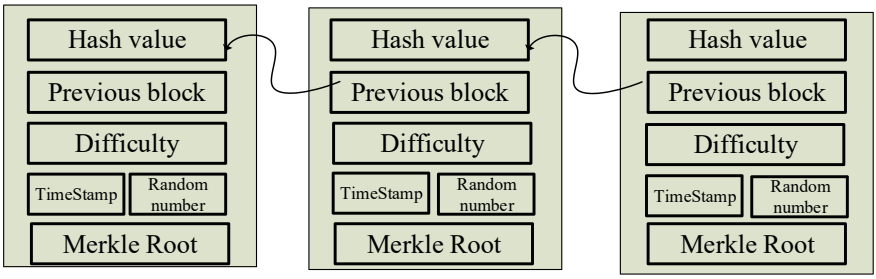


Fig. 5.5 Storage structure of blockchain header

nodes as a server at the same time. Compared with traditional centralized network, P2P network has the following advantages:

(1) **Scalability**. In P2P network, users can join and leave the network at any time. With the increasing number of user nodes, the service capacity can been improved dynamically. And in the blockchain system, with the addition of miners' nodes, the global computing power of the system will be improved, and the speed of "mining" will also be accelerated.
(2) **Robustness**. Because there is no centralized server in P2P network, the system security is greatly enhanced. Even if a node fails, the operation of system services will not be affected. Because of the other nodes can also serve as service provider, this determines that blockchain technology has high security.
(3) **Privacy protection**. In P2P network, data storage is distributed on each node, so users do not need to rely on the centralized server. Users will not worry about the unjustified ability of the central server. This is also the original intention of Nakamoto when designing the Bitcoin system: considering that if there exists a digital service like digital currency, if its server is located and controlled by a certain country, it will not be trusted by other countries and regions.
(4) **Low construction cost**. As P2P network adopts a distributed system, there will not need a big data center. For the reason that the participants share the hardware and software resources, the cost of network deployment is greatly reduced. It can be imagined that if Bitcoin, a global application, is based on the traditional C-S architecture, it needs to build a very large information system.
(5) **Load balancing**. Because of the decentralized storage resources in P2P networks, the situation of excessive burden on central servers is avoided, and the performance bottleneck problem of the entire system is also solved.

5.4.2 Digital Signature

Digital signature is a kind of data digest that can only be produced by the message sender, so it can realize an effective proof of the message sender. Digital signature

depends on asymmetric encryption technology and PKI, and generally includes of two algorithms: sign and verify.

The sign algorithm is executed by the message sender to generate the corresponding signature for the sending message. The sign algorithm inputs the sending message and the user's private key, and outputs the user's signature finally.

The verify algorithm is executed by the message receiver. After receiving the sender's message and the corresponding signature, the message receiver retrieves the user's digital certificate on the certificate server and obtains the user's public key to verify the received message. The verify algorithm inputs the message sent by the user, the digital signature and the user's public key, and outputs "Yes" or "No".

In the Bitcoin system, the installation software of the client is called "wallet", which is similar to the wallet in our life and stores many credit cards; The card in Bitcoin is actually the public/private key pair of the user, corresponding to an address for transaction; different from the miner nodes, the client software only stores user's related information; at the transaction occurrence node, the wallet software will generate a digital signature for the transaction information, then the miner node uses the user's public key to verify the validity of the signature to confirm the authenticity of the transaction, thus realizing the non-repudiation of transaction.

5.4.3 Hash Function

Hash function is a kind of mathematical function which is generally represented by *Hash*() or *H*() with the following properties:

- Messages of any length can be compressed into a fixed length output;
- Hash function is a kind of one-way function, that is, the algorithm can calculate the hash value in a limited time, but it is difficult to recover the original data only through the hash value;
- for any two different data blocks, the possibility of the same hash value is negligible; for a given data block, it is extremely difficult to find a data block with the same hash value.

These characteristics can be used for the purpose of data integrity checking or tampering detection. Users can calculate the unique hash value for the data block. Due to the irreversibility of the hash function, it is impossible for an attacker to calculate a forged hash value for the tampered data block to pass the verification algorithm. In the blockchain, a well-known application of hash function is the Proof of Work (PoW), commonly known as "mining". The miner node finds the answer firstly can obtain right to keep account, and makes other nodes to follow this decision and reach a consensus on this transaction. Due to the irreversibility nature of the hash function, the miners' nodes can only be cracked exhaustively without effective fast methods.

However, the following special situations should be paid attentions.

If two nodes in different regions of the world succeed in mining at the same time, two new sub blocks will try to merge into the main chain, resulting in "bifurcation" and global network bifurcation.

In addition, if a child block appears in a short time which will arrive at the storage node before the parent block, causes the miners to be unable to immediately assemble it into the "main chain", resulting in "orphaned block". The orphan node will be temporarily stored in the "orphan pool" until its parent node is found and then assembled to the "main chain".

We can see that the appearance of "bifurcations" and "solitary blocks" is related to the short interval between blocks. Therefore, Bitcoin designed the block interval as 10 min, which is a compromise between faster transaction confirmation and lower bifurcation probability. Shorter block generation intervals will enable transaction to be more quickly, and will also lead to more frequent block chain branching. In contrast, a longer interval will reduce the number of forks, but will result in a longer transaction time.

5.4.4 SPV Lightweight Verification and Melkel Hash Tree

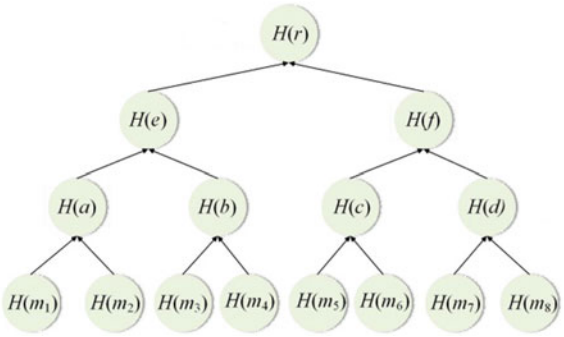
In addition, the hash function in blockchain uses a data structure called Melkel tree to avoid the verification burden with a large number of transactions. The Melkel tree is a binary tree. The leaf node stores the hash value of the transaction. Two adjacent leaf nodes can get the parent node by connecting and computing the hash, and then get the root node. In the blockchain, because each block stores a lot of transaction information, if the validity of each transaction is verified by verifying the hash function, it is necessary to maintain a lookup table to save transaction data and corresponding messages. When we need to verify a lot of transaction information, we must transmit a large number of hash values. Is it possible to verify the correctness in batches only by storing some hash values? Based on the Melkel tree technology, client only need to store the neighbor nodes of the node and the nodes on the path to the root node. In the blockchain network, the wallet node usually only stores the hash value of the transaction information related to the user, while the complete transaction information is generally stored in the service node in the blockchain network (Fig. 5.6).

5.4.5 Application of Blockchain in Business Big Data

1. **Data asset transaction**

With the development of AI technology, more and more applications rely on a large number of data to train models. In the early days, because the whole society was unfamiliar with IT and did not have a high awareness of data assets, personal data

Fig. 5.6 Melkel hash tree



could be transferred and sold at will. With the promulgation of the data security law, more and more people have a demand for the protection of their personal privacy data, and put forward the concept of "digital assets". Similar to traditional tangible assets, all valuable data will be called assets. In addition, the spread of data assets will be constrained by laws and privacy needs.

The traditional data management mainly depends on the original relational database. The data center stores the user's information centrally and authorizes the data security using traditional access control policies. However, this model is only suitable for small-scale data, with poor scalability, and cannot adapt to large-scale data management.

With the development of blockchain technology, blockchain has not only been used in the field of electronic currency, but also can replace the traditional relational database. Distributed storage such as IPFS can effectively avoid the problem of single point collapse of centralized servers. At the same time, blockchain has the characteristics of being tamper resist and traceable, which can become a good carrier of trusted data.

In terms of data trading, there are mainly two different trading platforms: one is a centralized trading platform, and the other is a decentralized trading platform. In the centralized trading platform, the core of this model is that there is a central node as the trading center, and all transactions must be conducted through this center, but there is not a trusted central system. In the blockchain based data trading system, distributed trusted nodes can be used to form a trading system. Individuals such as genetic data can be stored on the chain to achieve traceability and security management of the personal data.

2. **Digital copyright protection**

The network era makes the intellectual property digitalization develop rapidly, such as e-books, digital music, forum post Q&A and short videos. However, with the spread of various kinds of knowledge, new problems related to intellectual property rights have also been concerned:

(1) Internet users in Web 2.0 are not only content readers, but also content manufacturers. Under the new mode, netizens have changed from the simple users

to common maintenance of the network system. However, the random deletion and modification will result in illegal dissemination of original works, random elimination of evidence etc., resulting in difficulties in obtaining evidence for copyright infringement.

(2) Traditional trading platforms are based on third-party custody and coordinated buying and selling. Overreliance on centralized institutions will result in low processing efficiency and high cost; in addition, an untrustworthy third party may cache and repeatedly sell works for profit.

The domestic Internet giants have also deployed blockchain copyright technology for a long time. For example, Baidu has developed the Baidu "TuTeng" project to protect the intellectual copyright of pictures to solve the problem like the high cost of rights confirmation, redundant processes, widespread piracy and difficulties in rights protection; Based on blockchain technology, Tencent has proposed a "LingYu" certificate storage system to obtain trusted blockchain stored digital copyright and eliminate the risk of data tampering and forgery.

3. **NFT**

NFT (Non homogenous Token) is a unique digital asset. In 2014, the famous "cryptoart" website first tried to connect artworks with cryptocurrencies and sell artworks combined with paper wallets: the author printed the public key of the wallet of on the art works in the form of two-dimensional codes, and printed the private key on the reverse side; early enthusiasts tried to write the information of art into the blank space of Bitcoin, so as to store the information of works of art. With the continuous development of NFT technology, people realize that the traceability of digital art is similar to other traceability systems based on blockchain technology. They only need to store the author information and work descriptions related to the copyright on the chain, and create many NFT applications.

However, in blockchain based artwork storage, art circulation records (ACR) have the following characteristics:

(1) **Large amount of data storage**. ACR data often requires text records and high-definition information, resulting in excessive storage of blockchain nodes;
(2) **Data security issues**. Due to the investment attribute of artworks, ACR information is likely to be attacked and tampered by hackers, so it should be stored in a more secure way;
(3) **The data storage cycle is long**. Due to the aging problem, artworks will have a long storage period, and the number of queries will increase with the increase of popularity. Therefore, the traceability efficiency is challenged.

5.5 Business Big Data Management Case

This section mainly introduces a grain data traceability system based on the blockchain.

The grain supply chain is an integrated network composed of grain producers, storage enterprises, processing enterprises, transportation enterprises, distributors, retailers, and final consumers. Because the traditional grain traceability system adopts centralized data storage mode, the central server can tamper with the data at will, so it can not realize the reliable grain traceability process; secondly, the traditional traceability method needs to collect all the data of each enterprise, which will cause the disclosure of confidential information of each enterprise; the blockchain technology can make it have more advantages in privacy protection, so it can be used in the field of food traceability. This section introduces a case of grain big data traceability system based on blockchain.

5.5.1 Demand Analysis

The grain data traceability system involves a wide range of data holders and circulation links, mainly including of the following roles and information elements:

Planting enterprises: In order to ensure the traceability of planting, planting enterprises need to record agricultural products related to the planting process, including seeds, fertilizers and pesticides; During the planting process, small weather stations, monitoring equipment and soil sensors are used to collect the temperature and humidity, light quality, soil moisture content and other information during the planting process; In the process of collecting products, information such as trade name, collection date, collection place and collection personnel also need to be recorded.

Warehousing enterprises: In order to ensure the traceability in the storage stage, the information like goods placement, delivery, shipment, etc. in all the storage facilities, must be kept in the container level, pallet level or trade level.

Processing enterprises: Different batch numbers need to be assigned to different products (different commodity names/variety names) or from another field. After the products are graded and packaged into cartons, the processing enterprise will send a delivery notice to the customer. The delivery notice will list the logistics identification of the pallet of this batch of goods, as well as the trade identification and batch number of the products on each pallet.

Distributor: In the distribution phase, the distributor is responsible for receiving the ordered goods and their related trade identification code, batch number, accepted quantity and date information, and storing them in the data management system. The distributor has associated the batch information provided by the planting enterprise in advance with the field, processing factory or product production information.

Retailer: In order to ensure the traceability between distributors and retail enterprises, both companies record the trade identification code, batch number and flow information of logistics units of products.

In view of the above actual needs, the specific uplink data list of the grain traceability system is as in Table 5.3.

5.5.2 Network Architecture Design

According to the above analysis, the grain traceability system can be built on Kadelima distributed peer-to-peer network and Ethereum to store, verify and provide user data query services for grain supply chain. Compared with the traditional centralized traceability system, the "Blockchain + IPFS + DB" architecture can be used. There is no centralized server, and the nodes responds to the user's data uploading or querying needs. The system architecture is shown in Fig. 5.7.

Considering the data permission requirements of different roles in the blockchain network, the system defines the following roles and permissions for different participants:

Data provider node: the data provider node does not join the blockchain network, and is generally an enterprise node. If the enterprise is small and does not have the ability to set up a data holder node, it can delegate the task of data upload to the associated data creator node; If the enterprise is large and has the ability to join the blockchain, it can independently become a data holder node and join the decentralized network.

Data viewer node: the data viewer node does not join the blockchain network, and is generally a consumer node. The data viewer node can query the data holder node in the blockchain network through the user interaction module established by the system, and trace the information of purchased products.

Data holder node: the data holder node joins the blockchain network, which is generally an enterprise node. An enterprise can join the blockchain backbone network as a data provider node or a data holder node. The data holder node is the core of the whole system. As a participant of the data persistence layer, it jointly maintains the data storage of the whole system.

5.5.3 Data Storage Design

(1) First of all, there are many participants in the food supply chain, and the amount of transaction data is very large. It will cost a lot to store all the data in the blockchain, so additional databases are needed for auxiliary storage. In consideration of which database to use for storage, there are several options: relational database and non-relational database (such as file storage, key value storage, etc.).

Table 5.3 Requirements for food traceability system uplink data and privacy protection

Stage	Event	Uploaded data	Privacy data
Retailer	Product storage	Product name and code, retailer identification code, logistics code, transaction code, batch number, transaction date, transaction address	Quantity
	Product procurement	Salesperson number, order number, trade item identification code, serial number, commodity name, quantity, payment method, order amount, retailer address	\
Distributor	Product storage	Shipper identification code, logistics order number, trade order number, commodity name, origin, grade, delivery location, delivery date	Quantity
	Product information	Packaging worker number, logistics number, packaging time, packaging location	\
	Logistics information	Shipper No., logistics No., trade item identification code, batch No., commodity name, delivery location, delivery date	Quantity
	Distribution data	Buyer's name, business contact, purchase order number, trade name, buyer's address, order time, delivery time	Quantity
Processing enterprises	Raw material receiving	Shipper code, logistics code, commodity name/variety name, origin, shipment location, shipment date	Quantity
	Raw material processing	Batch, trade name, place of origin, grade, year, temperature, disinfection, processing equipment, additive type, additive dosage, etc.	Weight
	Product packaging	Trade code, product serial number, trade name, trade description, enterprise identification code, place of origin, grade	Weight
	Downstream purchase	Buyer's enterprise name, enterprise contact, purchase order number, commodity name, grade, buyer's enterprise address, order time, delivery date	Quantity
	Product transportation	Shipper No., logistics No., trade No., batch No., commodity name, origin, grade, delivery location, delivery date	Quantity

(continued)

Table 5.3 (continued)

Stage	Event	Uploaded data	Privacy data
Warehousing enterprises	Confirm receipt	Freight train information, delivery time, delivery personnel, name and enterprise identification code of the enterprise to which the freight train belongs, the person in charge of the freight train and contact information	\
	Raw grain warehousing	Batch, commodity name, place of origin, grade, warehousing time, warehousing clerk, storage location code	Quantity
	Downstream purchase	Enterprise contact, buyer's enterprise name, commodity name, origin, grade, buyer's enterprise address, order time	Quantity
	Out of raw grain	Worker identity, location code, batch, commodity name, origin, grade, and issue time	Quantity
	Logistics information	Shipper code, logistics code, commodity name/variety name, place of origin, grade, delivery location, delivery date	Quantity
Planting enterprises	Purchase of agricultural materials	Fertilizer trade name, seed trade name, pesticide trade name, purchase time	Information and purchase time of agricultural products
	Planting link	Growers, land information, sowing time, meteorological data, temperature and humidity information, etc.	Land owner information
	Harvesting link	Land information, reapers, grain varieties and quantity, harvesting time	Land owner information
	Downstream purchase	Buyer No., trade name, variety, grade, quantity	Buyer enterprise and other information
	Logistics information	Truck, delivery time, shipper	The enterprise to which the truck belongs and the information of the person in charge of the truck

(2) Secondly, we can see that the traceability data of the grain supply chain: products, participants, locations, and events are interrelated. The products, participants, locations, and events are regarded as nodes, and the edges with directions are used to bind different entities. It is easy to build complex relationships within the supply chain. The node4j graph database can take the relational information as the first-class entity of information storage. The built-in optimized graph

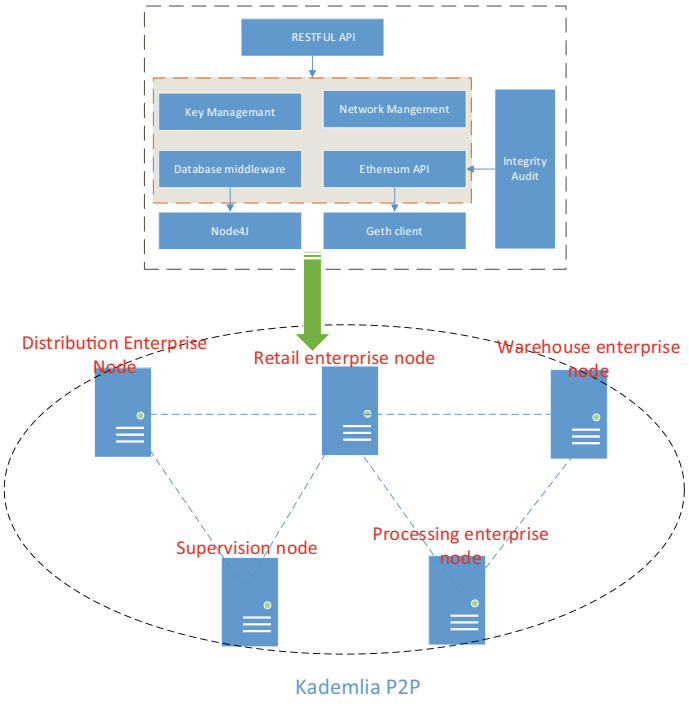


Fig. 5.7 Architecture of grain traceability system based on blockchain

algorithm can provide high-performance fast traversal, and is flexible in creating data models. The cost of graph expansion and modification is low. Therefore, the node4j graph database can be used to store additional information beyond the key uplink information.

(3) Data integrity module design: use remote data integrity detection algorithm to check the integrity and consistency of the data uploaded by the data provider. When the data check fails, you can feed back the error to the user interaction module through the Node Service Module. When the data check passes, you can further perform data persistence operations through the Node Service Module and store the data hash value (data fingerprint) to the blockchain node.

(4) Privacy protection requirements: in order to prevent the rapid growth of the blockchain ledger and the disclosure of data privacy, the system only needs to store the hash value of user information on the chain when storing on the chain, and specific information can be stored on the enterprise's own server outside the chain; In order to be compatible with blockchain features, IPFS is used as the off-chain database.

Chapter 6
Big Commerce Data Knowledge Representation

6.1 Multi-granularity E-Commerce Entity Construction Model

E-commerce (electronic commerce) refers to share business information and engage in goods and service trading using various information technology, including computer technology, Internet technology and telecommunication technology, etc. E-commerce could help to realize electronic business processes between enterprises and enterprises, enterprises and individuals, and individuals and individuals, thus facilitating efficient and effective transaction activities between suppliers, customers and logistics.

The whole process of e-commerce includes all aspects of production, inventory, distribution and finance. The main entities and interrelationships of e-commerce are shown in Fig. 6.1. The E-commerce provide different sales platform for e-tailers (online shops), thus providing customers with a wide range of goods that meet different levels of demand. Meanwhile, Suppliers provide various goods and services. Moreover, they can update and change the attributes of goods according to feedbacks from e-tailers and the e-commerce market. Customers can complete their transactions with e-tailers through the online payment system, i.e. electronic banking. And e-tailers can provide logistics services for customers with their cooperative logistics. The entire process is supervised by commercial regulators and supervisors, including the State Administration for Industry and Commerce, State Taxation Administration and People's Bank of China, etc. This process ensures the orderly, stable and efficient development of the e-commerce market.

6.1.1 Multi-granularity E-Commerce Entity Category

As can be shown in above figure, E-commerce is consisted of six entities including customers, e-tailers, suppliers, goods and services, e-banking and distribution

J. Cao, *E-Commerce Big Data Mining and Analytics*, Advanced Studies in E-Commerce,
https://doi.org/10.1007/978-981-99-3588-8_6

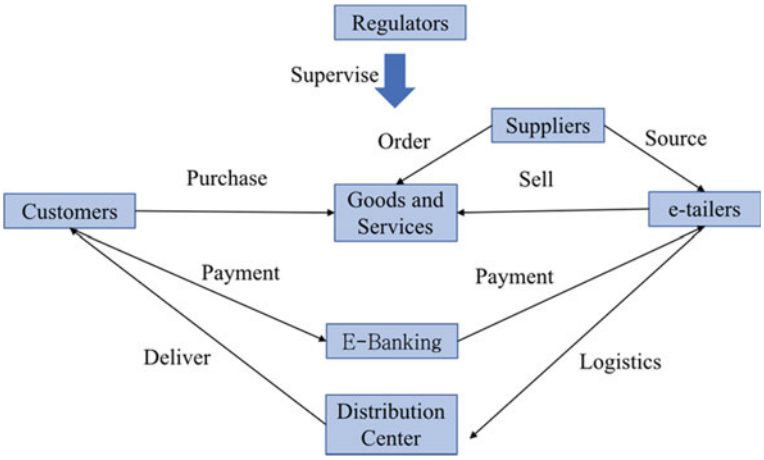


Fig. 6.1 E-commerce entities and their relationships

centers. Depending on the attributes of the entities, the six entities can be further divided into sub-entities of different granularity. These sub-entities can not only provide the accurate representation of entities so as to predict link and relationship among entities, but are also helpful to build downstream applications such as recommendation and fraud detection in e-commerce platforms. This chapter focuses on the subdivision methods of entities from the perspective of customers and e-tailers.

From the perspective of customers, subdivision methods for entities are mostly in two ways, i.e., based on customers' demographics and customers' consumption behavior. customers' demographics include gender, age, education, geographical location and so on. In particularly, demographics are important guidelines for e-commerce platforms to conduct targeted marketing. For example, from a survey study for online shoppers in Beijing,[1] 67.8% of customers were women, while were far more than man shoppers. In terms of the number of online purchases, women accounted for 64.9% of the total amount of online purchases, while men accounted for 35.1%. However, men are more inclined to use the Internet to search for product information and services than women, and the number of online transactions from men is generally 2.4 times higher than that of women. but there is no significant difference in the evaluation of e-commerce platforms by men and women. In terms of age group, nearly 60% of customers are between 30 and 49 years old. The amount of online purchases from this group accounted for 69.1%, with 45.5% of customers aged 30–39 and 23.6% aged 40–49. Moreover, the distribution of education shows that the highly educated group is more fond of online shopping. The proportions of online shopping customers with junior high school and below, high school, college and undergraduate education and above were 7.6%, 19.1%, 23.0% and 50.3%, respectively, and the proportions of online shopping amount were 3.3%, 11.8%, 20.3% and

[1] https://www.chyxx.com/industry/202006/871657.html.

64.6%, respectively. The consumer behavior of online customers in e-commerce platforms plays a key and important role in the operation of the platform and can be broken down according to the purchase time, the price and how active customers are on the platform. For example, customers who regularly browse e-commerce websites are usually not averse to advertising and marketing on the platform, so they can be pushed more advertisement. And for customers who like to shop late at night, target marketing should be carried out actively at that time.

From the perspective of e-tailers, according to the categories of goods and services, e-tailers can be divided into clothing shops, food shops and digital products shops, etc. And then according to the different groups of target customers, e-tailers can be divided into women's clothing shops, men's clothing shops and maternity shops, etc. Moreover, according to the different sales method, e-tailers can be divided into pure online shops, pure offline shops and online-offline shops. According to the Taobao index, the top 10 categories are: women's clothing/women's boutique, beauty care/body care/essential oils, alcohol, outdoor/mountaineering/camping/travel supplies, network equipment/network-related, spirulina/algae extract, health products/dietary supplements, residential furniture, women's underwear/men's underwear/home and snacks/nuts/specialties. The category of online shops usually influences the exposure and click-through rate of their goods and services, which is also an important factor for e-commerce platforms to build search engines and recommendation systems.

6.1.2 E-Commerce Entity Recognition

The rapid development of E-commerce has brought about a huge amount of e-commerce data. Identifying and classifying entities relying only on manual labor are time-consuming and labor-intensive. Meanwhile, the accuracy of entity recognition determines the effectiveness of downstream tasks including product retrieval, recommendation systems and customer service question and answer systems, etc. In E-commerce systems, entity recognition should include not only the name of the entity, but also the category of the entity, the attributes of the entity and some other related components. Take an example of the Apple phone, the content of the identified entity should include: entity name: Apple/mobile phone, category: digital products, attributes: memory, version number and price, etc. However, different e-commerce platforms have different system architectures which lacking a unified schema representation. And the descriptions of goods and services vary widely and the quality of data also varies, which greatly affects the performance of entity recognition. Currently, there are several difficulties in entity recognition, i.e., the difficulty of defining entities, the slow annotation of entities and the existence of ambiguity among entities. The definition of an entity refers to the start and end position of an entity in a sentence. For example, is the item "iphone 11" identified as "iphone" or "iphone 11"? The slow annotation of entities is mainly due to the proliferation of Internet terms resulting in incomplete coverage of entities. The ambiguity of entities is mainly reflected in two aspects: firstly, there are multiple words with

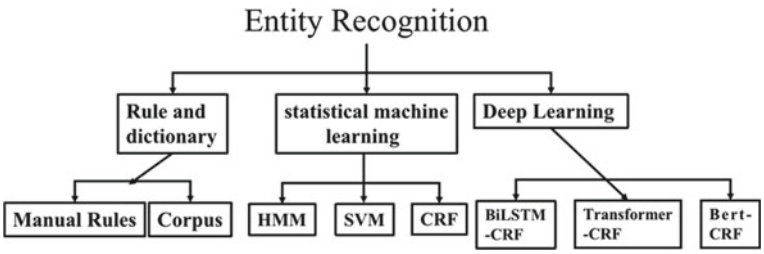


Fig. 6.2 Entity recognition methods

the same meaning. secondly, the same entity has different meanings. For example, "apple" means not only fruit but also mobile phone. To solve the problem of entity recognition, a large number of approaches have been proposed in academia and industry, including rule-based and dictionary-based approaches, statistical machine learning-based approaches and deep learning-based approaches, as shown in Fig. 6.2.

The main idea of rule-based and dictionary-based approaches is to match predefined templates, i.e. templates constructed manually by experts, such as the word formation patterns of entity words and high frequency context words. Then entities can be identified by matching word strings from the predefined templates using regular expressions. However, such methods rely too much on the construction of corpora and dictionaries which are time-consuming and infeasible, e.g., different corpus are needed for different domains. In the public dataset, the Chinese corpus includes about 1.5 million words from Chinese news, government documents, magazines and online blogs.[2] While the corpus of Enron includes over 500,000 emails in English marked with name and date.[3] And the Ontonotes corpus includes 1,745,000 English, 900,000 Chinese and 300,000 Arabic text data, including telephone records, news, radios and weblogs.[4]

Statistical machine learning based approaches aim to build multi-classification tasks or sequence annotation tasks by using a large number of labelled corpora. And widely used statistical-based machine learning approaches include: Hidden Markov Mode (HMM) and Maximum Entropy Markov Models (MEMM), which are based on probabilistic graphical models, and Support Vector Machine (SVM), Conditional Random Fields (CRF), etc.

In recent years, deep learning techniques have greatly developed and made considerable progress, especially in the field of natural language processing. A large number of deep learning models have emerged to represent words using vectors through neural networks. These models solves the problem of data sparsity. Meanwhile, dense vectors for words contain more semantic information than manually selected features, which could also be helpful for entity recognition. Moreover, word features can be represented under a uniform vector space from heterogeneous texts [1]. In

[2] https://catalog.ldc.upenn.edu/LDC2013T21.

[3] http://www.cs.cmu.edu/~enron/.

[4] https://catalog.ldc.upenn.edu/LDC2013T19.

the literature, widely-used deep learning-based models include the BiLSTM-CRF model, Transformer_CRF model, and Bert-CRF model, etc.

Open-source Chinese natural language processing tools include Jieba, SnowNLP, PkuSeg, THULAC, HanLP, FoolNLTK, Harvard LTP, and Stanford CoreNLP. HULAC (THU Lexical Analyzer for Chinese) is a Chinese lexical analysis toolkit developed by the Laboratory of Natural Language Processing and Social Computing at Tsinghua University for Chinese word tokenization and lexical annotation. Based on the THULAC tool, which integrates the world's largest corpus of manual word tokenization and lexical annotation in Chinese (containing about 58 million words), it can achieve an F1 value of 97.3% for word tokenization and 92.9% for lexical annotation on the Chinese Treebank (CTB5) dataset. Moreover, it can process about 150,000 words per second.

6.2 Multi-category E-Commerce Entity Relationship Extraction

In the field of e-commerce, entities can belong to different categories, for example, entities can be customers as well as goods. There are not only explicit and implicit relationships among entities of the same category, but also among entities of different categories. And these relationships can be extracted from structured and unstructured databases. With the huge amount of e-commerce data, extracting relationships manually is not practical, which requires to extract relationships by means of information technology. Recently, based on machine learning and deep learning techniques, a large number of researchers have propose various methods for extracting relationships from unstructured information including text, images and speech, etc. These methods provide great support for search engine systems, recommendation systems and automated question and answer systems for e-commerce platforms.

6.2.1 Multi-category E-Commerce Entity Relationship Categories

Customers, products and e-tailers are the three main categories as entities in e-commerce platforms. The interactions among them bring about a complex network of relationships, as shown in Fig. 6.3. Firstly, there are multiple relationships between entities of the same category. For instance, customers can follow each other to form a friend relationship. Meanwhile, a customer can reply to another customer's message in the comment section of a product to form another relationship. Moreover, there are competitive relationships among products of the same category, such as Coca-Cola and Pepsi. Also, there are complementary relationships among products of different types, such as toothbrushes and toothpaste. As for e-tailers, there are also competitive

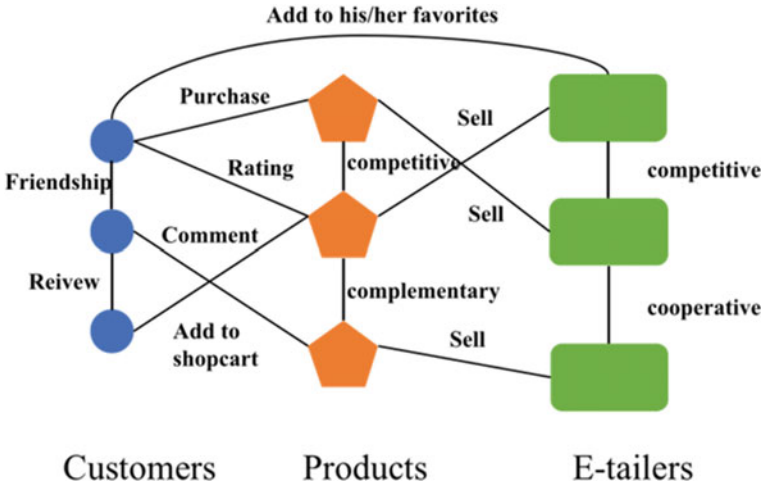


Fig. 6.3 Complex relationships among customers, products and e-tailers

and cooperative relationships. Secondly, there are also different relationships among entities of different categories. For instance, customers can buy products from e-tailers, which forms a purchase relationship. And customers would comment on the corresponding product, thus forming comment relationship. Moreover, customers would like to add some e-tailers to his/her favorites as they think products which e-tailers sell have great value to them. Meanwhile, if customers think the price of the product will drop or they don't need it for the time being, they will put it in their shopping cart for easy access later. Finally, there are other implicit relationship extraction, including implicit friend relationships among customers, implicit relationships among sub-categories and parent categories of products by inferring through product titles. And implicit relationships among products can also be inferred through reviews.

6.2.2 Multi-category E-Commerce Entity Relationship Extraction Methods

In E-commerce systems, explicit relationships are easier to obtain. For example, friend relationships among customers can be determined by parsing the customer's profile page through HTML and XML technologies. However, implicit relationships are difficult to be extracted, especially for relationships based on reviews. For instance, customers usually mention the advantages and disadvantages of the product, the quality of e-tailers' service and the logistics in their reviews. Thus, how to explore such relationships among different products, between products and

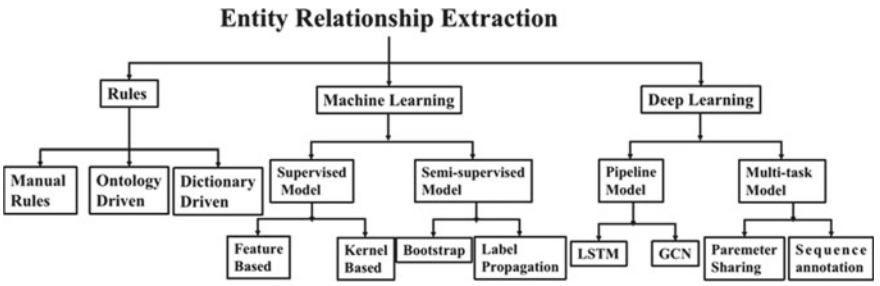


Fig. 6.4 Entity relationship extraction methods

merchants and between logistics remains a challenging task. These implicit relationships play a crucial role in E-commerce retrieval systems, recommendation systems and customer service Q&A systems. To this end, this section will focus on relationship extraction techniques based on reviews. Currently, the mainstream relationship extraction techniques in industry and academia are shown in Fig. 6.4. There are three main categories, namely rule-based approaches similar to entity recognition, traditional machine learning-based approaches and deep learning-based approaches.

Early methods for entity relationship extraction are based on rules, which are mainly based on the manual constructed rules by domain experts. These rules usually are consisted of a collection of patterns based on words, lexicality or semantics. In the process of relation extraction, the defined rules are matched against each other with pre-processed utterances to complete the relevant relation extraction. Similar to the rule-based approaches for entity recognition, experts in different domains are required to determine related rules, resulting in poor portability of this approach. while the main idea of those dictionary driven approaches lies in the construction of verbs in the lexicon, which then determines the relationships among entities. Ontology-driven approaches mainly rely on the existing ontology hierarchy and the relationships among the entity concepts to assist in the relationship extraction.

Traditional machine learning based methods fall into two main categories: supervised learning-based models and semi-supervised learning based models. Among them, supervised learning-based algorithms, which treat the relationship extraction task as a classification problem, can be further divided into two categories: feature based algorithms and kernel based algorithms. The feature based algorithms focus on constructing features explicitly. While the kernel based approaches computer the inner product between feature vectors learned implicitly through the kernel function. The biggest problem for supervised learning based algorithms is that they require a large number of labelled training corpus, which are time consuming and expensive. While, semi-supervised learning based algorithms only use a small amount of labelled information to update the unlabelled information. For example, bootstrap based algorithms first identify a small number of relationships as seeds and automatically obtain new relationships from a large training corpus through iterative approaches. The main idea of label propagation based algorithms is to predict unlabelled data using labelled information. This algorithm treats the classification

problem as the propagation of labels over a graph, with all entities treated as nodes and the relationships between pairs of entities as edges. However, this algorithm has high uncertainty and is not suitable for text data with particularly complex relationships [2].

In recent years, as deep learning techniques are widely used in the fields of image and natural language processing, they have gradually become a hot topic in the relationship extraction area. Deep learning-based methods can integrate different low-level features to form more abstract high-level features, which can better represent individual words, sentences and documents, which also reduce the error and workload associated with manual feature selection. These methods can be divided into two categories, i.e., pipeline models and multi-task learning models. The main process of pipeline models can be described as follows: the relationship is firstly extracted from the sentences that have been annotated with the target entity pairs, and then the triples with entity relationships are output as the prediction results. The widely used pipeline models include CNN, LSTM and GCN. The main idea of the multi-task learning model is to build a joint model for entity recognition and relationship extraction, so that the relationships between different entities can be extracted directly. Those methods can be further divided into parameter sharing methods and sequence annotation methods. Among them, the parameter sharing method models entities and relations respectively. While the sequence annotation method models entity relations directly [3].

6.3 Multi-level Knowledge Representation Model

The first two sections focus on two key factors in e-commerce platforms, i.e., entities and their relationships. And how to mine these two factors and subsequently provide effective knowledge representation for downstream tasks plays a crucial role for search engine systems, recommendation systems and automated question and answer systems in e-commerce platforms. The attributes of entities in E-commerce platforms consist of text, especially information about product descriptions, reviews, etc. And relationships can be described as graph structures. Based on this, this section introduces knowledge representation models in terms of text based on natural language processing and relationship representation models based on graphs.

6.3.1 Knowledge Representation Model Based on Natural Language Processing

Early knowledge representations for texts are mainly based on discrete symbolic representations. Among them, each word is expressed as a vector through one-hot encoding, and the length of the vector is the number of words. In this vector, the position of the current word is 1, and the rest are 0. And the sentence can be represented by the bag of word model, TF-IDF model, etc. However, the one-hot encoding only symbolizes words, which does not contain any semantic information and distance information. For example, the cosine distances of words and sentences represented by the one hot encoding are all 0s. Meanwhile, this encoding method leads to data sparseness, which decreases the performance of downstream tasks.

The distributed representations of texts assume that two words are believed to be similar if their contexts are similar. This context-based assumption can be realized through the co-occurrence matrix, which describes the number of co-occurrences in the same context. Based on the co-occurrence matrix, there are many ways to obtain continuous word representation, such as latent semantic analysis model (Latent Semantic Analysis, LSA), Latent Dirichlet Allocation (LDA), Singular Value Decomposition (SVD), etc. Thus, the distributed representation captures the association information between words by representing words as low-dimensional dense vectors. This representation method can efficiently calculate the semantic association between words in a low-dimensional space, and effectively solve the problem of data sparsity. Generally speaking, the distributed representation of text can be divided into two categories: the model based on the above-mentioned matrix decomposition method and the distributed representation model based on deep learning.

The early work for deep learning based distributed model is the neural network based probabilistic language model (Neural Probabilistic Language Model, NPLM), which constructs the word sequence $[w_1, w_2, \ldots, w_n]$ based on the following equation. This method is directly based on the neural network structure, as shown in Fig. 6.5. In order to improve training efficiency and adapt to larger-scale data, the researchers split a local context window during the training process, thus constructing a shallow window-based Word2Vec model, i.e., Skip-Gram and CBOW algorithm. Among them, the Skip-Gram model mainly uses the target word to predict the context word. While the CBOW model mainly predicts the target word through the context word. Although algorithms based on shallow windows have improved model efficiently, they are limited by the size of the window and fail to contain global information. To address this issue, the GLOVE model combines the advantages of matrix and shallow windows, exploiting both the global statistics of the corpus and the local contextual information.

$$\mathrm{p}(\mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_\mathrm{n}) = \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{p}(\mathrm{w}_\mathrm{i} | \mathrm{w}_1, \mathrm{w}_2, \ldots, \mathrm{w}_{\mathrm{i}-1})$$

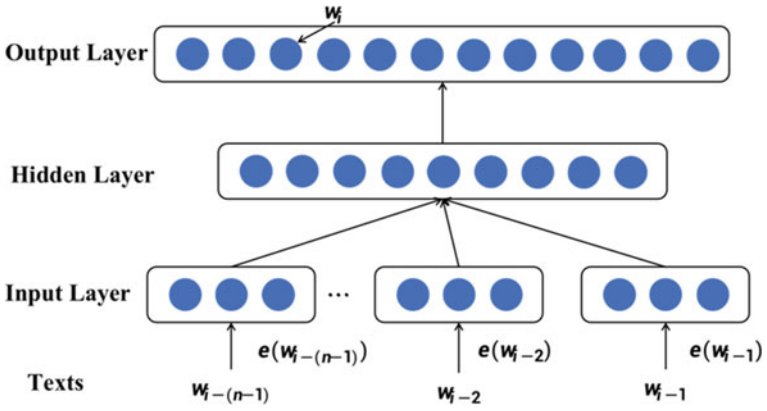


Fig. 6.5 Probabilistic language modeling based on neural networks

6.3.2 Knowledge Representation Model Based on Relational Network

In the e-commerce system, there are not only interactions among entities of the same type, but also complex interactions among different entities. Hence, how to model this interaction to better support the e-commerce system is the focus of this section. For example, interactions between customers and products can be exploited by graph learning systems to make very accurate recommendations so as to provide better support for e-commerce platforms. There are massive amounts of data with graph structures in e-commerce systems, i.e., massive entities and complex relationships. These complex data present a highly nonlinear structure and have serious problems of sparsity. Therefore, this section focuses on the knowledge representation model from the perspective of network embedding or graph embedding. Graph embedding can be regarded as a dimensionality reduction process, i.e., high-dimensional graphs data could be represented as low-dimensional vectors. Currently, knowledge representation methods based on graph embedding in industry and academia can be divided into the following categories: matrix factorization-based methods, random walk-based methods, and deep learning-based methods.

Most of the early graph embedding methods are based on matrix factorization. This type of method forms an adjacency matrix by connecting edges between nodes, and then factorizes the matrix to reduce the dimensionality to obtain a low-dimensional vector. Moreover, based on the adjacency matrix, Laplacian matrix, probability matrix, and similarity matrix can be derived for matrix factorization. For matrices that cannot be decomposed by eigenvalues, such as positive semi-definite matrices, the dimensionality reduction can be obtained by the gradient descent method. Related algorithms for graph embedding include Laplacian Eigenmaps, Structure Preserving Embedding, Graph Factorization, etc.

The main idea of the random walk-based methods is a series of steps to select nodes in the graph, i.e., select a starting node, randomly select a neighbor node, and repeat this process many times. Among them, the DeepWalk algorithm adopts the idea of the Word2Vec algorithm by treating the nodes as words and generating short random walks as sentences. And then the embedding of the graph can be obtained through the Skip-gram language model. This type of algorithm combines local information and high-order information. Node2Vec is a variant of DeepWalk which is used for weighted graphs. Node2Vec introduces the AliasSampling to perform random walks on weighted graphs, and combines breadth-first and depth-first algorithms to obtain local and global network information. Some other variants include hierarchical representation learning for networks, Walklets, Metapath2vec for heterogeneous networks, etc.

In recent years, deep learning has achieved great success in Euclidean space, such as images, videos, and texts. However, deep learning techniques are rarely applied to graph structures. To this end, this section focuses on the graph representation algorithms based on deep learning, including Structural deep network embedding (SDNE), Graph convolutional networks (GCN) and Variational graph auto-encoders (VGAE). Among them, SDNE uses a deep autoencoder to preserve the first-order and second-order network proximity. SDNE consists of two parts: the unsupervised component and the supervised component. The former consists of an autoencoder that aims to find an embedding can reconstruct its neighborhood. While the latter is based on a Laplacian eigenmaps. Graph Convolutional Networks (GCNs) iteratively aggregate the neighborhood embeddings by defining a convolution operator on the graph, and use the embeddings obtained in previous iterations to obtain new embeddings. This algorithm is scalable since the embedding is aggregated only local neighborhoods. Meanwhile, multiple iterations allow this algorithm could learn global information. VGAE uses a graph convolutional network (GCN) encoder and inner product decoder, which relys on GCN to learn high-order dependencies between nodes.[5]

6.4 Case Studies of Big Commerce Data Knowledge Representation

With the development of the Internet, the e-commerce industry has also made rapid progress in the past few years. Not only that, social e-commerce based on social networks has gradually penetrated into people's lives. However, on social network platforms, criminals have formed a mature industrial chain by employing a large number of paid posters to engage in illegal activities. These paid posters use a large number of fake accounts disseminate false information and conspiracies so as to influence public sentiment and maximize the social influence of employers, or realize their improper political intentions. Recently, false information about COVID-19 vaccines

[5] https://cloud.tencent.com/developer/article/1610049.

on social platforms was disseminated everywhere and received a large number of irrational attacks, causing a large number of people to resist vaccination which resulted in the COVID-19 epidemic. In addition, the paid posters also conduct unfair competition through malicious ratings of an event or product on social platforms, resulting in damage to the reputation of relevant personnel and malicious marketing of related products. This seriously disrupts the social order and network environment. So, it is great important to crack down on illegal activities especially fraudulent activities on social networks and e-commerce platforms.

At the same time, for features implicit in the network relationships, especially between sailors and sailors in order to avoid "a bunch", there is usually no direct connection, through the graph embedding and graph neural network and other techniques, the different types of relationship abstraction into graph representation tasks, and then mining the implicit characteristics of the sailors.

In a social network, user-centered social circles can be built based on common hobbies, mutual friends, and common workplace. These social circles contain rich user information which can be helpful for detecting paid posters. Compared with normal users, the proportion of paid posters' follows and followers is seriously unbalanced. In particularly, the proportion of abnormal users in paid posters' social circle is very high. Detecting paid posters based on social circles can not only identify a single abnormal user, but can also detect groups of paid posters. In the literature, various studies for detecting paid posters are based on social circles. Among them, most works focus on single relationship in social circles and are based on isomorphic networks. This section pays more attention on multiple relationships, e.g., comments, likes and forwarding, etc., so as to build a heterogeneous network among users, comments, and events. Especially, events can be replaced by topics extracted through topic models. Figure 6.6 shows the relationships extraction from the heterogeneous network. Based on this network, the explicit relationship can be extracted to describe the differences of paid posters. Meanwhile, in order to be unnoticeable, there are usually no direct connections among the paid posters. These implicit relationships can be extracted based on graph embedding and graph neural network methods.

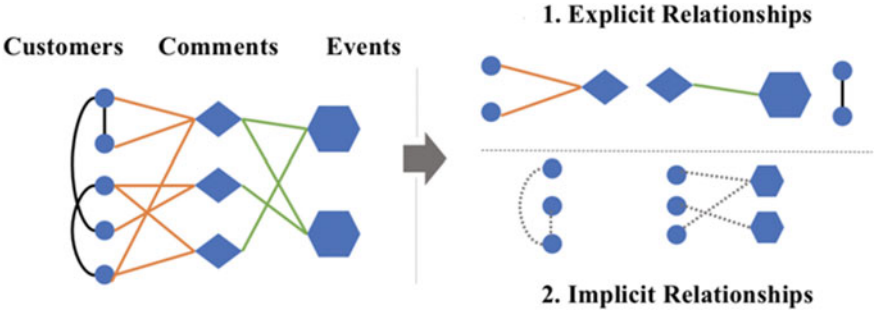


Fig. 6.6 Relationship extraction based on heterogeneous network

In heterogeneous networks, explicit relationships can be extracted based on the explicit features of customers, comments, events. However, these explicit features are easy to be attacked by paid posters to cover up their malicious purposes. While features constructed based on network structures have certain stability, and are not easy to be concealed. Hence, this section intends to construct more distinguishable and stable features from various network structures. Based on different relationships, e.g., likes, comments, forwarding, etc., we can construct explicit features in following ways: (1) Based on existing network structures, we can construct network-based indicators such as degree, clique, density, etc. (2) For different relationships, we can construct multiple networks. For example, if both of two customers like the same comment, we can add an edge between these customers, and thus construct an isomorphic network. Then based on this network, relational indicators can be constructed based on statistics, such as the proportion of two customers who like the same comment.

In heterogeneous networks, implicit feature extraction can be based on following methods. Indicators on explicit network structures described above cannot fully represent the characteristics of paid posters. To address this issue, it is necessary to construct the implicit heterogeneous relationship between customers and customers, and between customers and events. Existing works usually tend to focus on the relationship between customers and comments, which still have limitations. While by analyzing the relationship between users and events, we can identify paid posters more accurately. To this end, it is necessary to establish the implicit relationship between customers and customers, and between customers and customers. Most of the existing works are based on homogeneous networks. That is, there is only one type of entity in the network. While the heterogeneous network in this section contains three types of entities. And implicit features can be constructed based on following methods: (1) we can transform the heterogeneous network into multiple homogeneous networks using multi-level random walks. Then we can construct features based on the DeepWalk model, graph neural network and graph embedding models in the homogeneous network; (2) we can define heterogeneous networks using different meta-paths e.g. U for customer, R for comment and E for event, then meta-path "URE" means that one customer commented on a comment belonging to an event. And meta-path "UERU" means that both users commented on the same comment of the same event.

References

1. Liu L, Wang D (2018) A review of named entity recognition research. J Intell vol. 37, no. 3
2. Li D, Zhang Y, Li D et al (2020) A review of research on entity relationship extraction methods, Computer Research and Development 57(7):1424–1448
3. E HH, Zhang WJ, Xiao SQ et al (2019) A review of deep learning entity relation extraction research. J Software 30(6):1793–1818

Chapter 7
Business Big Data Knowledge Fusion

7.1 Semantic Extraction and Semantic Association

Knowledge graph is essentially a semantic network. Semantics is already an abused word, and each has a different understanding of what semantics is. The semantic web is an application of artificial intelligence in the era of big data, specifically, the application of knowledge representation and reasoning, with its back-end theoretical support of description logic and modal logic. The emergence of the Semantic Web is causing the Web to evolve from a document network to a data network. The content on the Web has made a major shift from just human-readable text to a computer-acceptable data structure for describing knowledge. The vast amount of machine-readable, triadically structured semantic data is enabling an increasing number of intelligent applications. Among others, modern Web search engines use semantic data to improve search accuracy and enhance their search results with straightforward and rich answers. This section introduces some semantic extraction and semantic association techniques based on the Semantic Web's knowledge representation framework (e.g., RDF) and evaluates existing semantic association ranking techniques.

7.1.1 Subgraph Matching Algorithm for RDF

RDF (Resource Description Framework) that is, the resource description framework, is developed by the W3C. It is a standard data model for describing entities/resources. In the knowledge graph, we use RDF to formally represent the ternary relationship (Subject, predicate, object).

For complex queries, most of today's research cannot effectively solve the multiple join problems associated with multiple triplet queries, and these problems can seriously affect query efficiency. Also, querying these knowledge bases is usually done using structured queries that use graphical schema languages such as SPARQL.

J. Cao, *E-Commerce Big Data Mining and Analytics*, Advanced Studies in E-Commerce,
https://doi.org/10.1007/978-981-99-3588-8_7

however, such structured queries require some expertise from the user, which limits the access to such data sources. To overcome this problem, keyword search must be supported. This section presents a check-subgraph matching algorithm for keyword queries on RDF graphs.

The first step is to retrieve the set of subgraphs that match the user's keyword query. To avoid retrieving arbitrarily long subgraphs, we restrict the retrieved subgraphs to those with the following two attributes.

1. Subgraphs should be unique and maximal. That is, each subgraph retrieved should not be a subset of any other subgraph retrieved.
2. Subgraphs should contain triples that match different sets of keywords. That is, triples in the same subgraph will not match the exact same set of keywords. If two triples match the same set of keywords, they are part of two different possible results of the user's query and should be considered as part of two separate subgraphs.

The algorithm is as follows

Given a query $q = \{q_1, q_2, \ldots, q_{\mathrm{m}}\}$, represent it as a set of keywords.

Our subgraph retrieval algorithm first retrieves a list of all triples that match the query keywords.

We use the reverse index to retrieve the list $\{L_1, L2, \ldots, L_m\}$ (Li is the list of all triples that match q_i).

(1) Find all unique triples in $\{L_1, L2, \ldots, L_m\}$, find all unique triples and form them into a set E. The set E can be considered as a disconnected graph, which we call a query graph.
(2) Adjust the backtracking algorithm to retrieve subgraphs from the query graph.

The subgraph matching algorithm in RDF, which is an important part of semantic extraction and semantic association, is an inescapable problem in the vast majority of Semantic Web applications. The specific study of this part involves a lot of knowledge related to graph theory, which will not be discussed too much here.

7.1.2 Knowledge Graph Keyword Search Algorithm

Our knowledge base consists of a set of SPO triples, and to be able to handle keyword queries, we construct a virtual document for each triple. contains a set of keywords extracted from the subject and object of the triad, as well as representative keywords for the predicates.

Given a keyword query, we use a reverse index to retrieve a list of matching triples for each query keyword. We then concatenate the triples in the different lists based on topics and objects to retrieve subgraphs with one or more triples.

However, we only construct subgraphs containing triples from different lists, corresponding to matches for different (groups of) keywords. The intuition behind this is that we assume that the user has a precise information need in mind that can be

precisely represented using a set of triple patterns. Since keyword queries introduce additional ambiguities that do not exist in structured triple pattern queries, result ranking becomes important. To provide efficient ranking, the system must infer the most likely structured query in the user's mind and rank the subgraphs according to how well they match this implicit structured query.

7.1.3 Semantic Association Ranking Techniques

Searching for associations between entities is required in many fields such as e-commerce. In recent years, it has been facilitated by the emergence of graph-structured semantic data on the Web, which provides more explicit associations than structured semantic associations hidden in unstructured text for computer discovery. The increase in the volume of semantic data often generates too many semantic associations and requires the use of ranking techniques to identify more important associations for users. Despite fruitful theoretical research on innovative ranking techniques, a comprehensive empirical evaluation of these techniques is still lacking.

Earlier search engines that did not have semantic data had to analyze documents and discover associations hidden in the text, and thus inevitably suffered from imprecision and incompleteness. Using triadic semantic data, which can also be represented as entity-relationship graphs, finding and returning paths and subgraphs connecting user-specified entities would be very simple, as recent specialized search engines, including RelFinder and Explass, have done. Such graph-structured semantic associations are relatively easy to discover compared to associations hidden in unstructured text.

Known technologies fall into two main categories: data-centric technologies and user-centric technologies. Data-centric technologies analyze all aspects of semantic data. User-centric techniques focus on user preferences. In addition, diversity-based re-ranking can improve the quality of the highest ranked results, considering that semantic associations may overlap with each other.

The highest ranked semantic associations generated by the above techniques may not include the best results because they provide information that may overlap with each other and thus exhibit redundancy. To improve the diversity of results, the semantic associations are re-ranked to allow only the top-ranked results to have limited overlap.

7.2 User Profile Construction

Alan Cooper, the father of modern interaction design, first introduced the concept of user portrait, which refers to the use of virtualized representatives to identify real users, and is a user model built from a series of actually generated data, which can also be called user information labeling. User profiles usually describe the basic

attributes, social characteristics, consumption habits, life preferences and other labels of users in easy-to-understand life terms.

7.2.1 User Data Collection

User data is generally classified into two categories: static information data and dynamic information data.

Static information data refers to the relatively stable information of users, mainly including data on demographic attributes, business attributes and other aspects. If there is a God, everyone's behavior is monitored by God's invisible eyes all the time. In a broad sense, a user opens a webpage, buys a cup; the same as the user slipped the dog in the evening, took a trip in the daytime In a broader sense, a user who opens a web page and buys a cup is the same as a user who walks the dog in the evening, picks up money during the day, yawns, etc. These are all user behaviors in God's eyes. When the behavior is focused on the Internet, or even e-commerce, the user behavior will be focused a lot, as shown in the picture above: browsing the home page of Vancl, browsing the single product page of casual shoes, searching for canvas shoes, posting microblogs about the quality of shoes, praising the microblog message of "Double Eleven Promotion". All these can be regarded as Internet user behaviors.

There are several general methods on how to capture user data.

(1) **Questionnaires**: If you don't know where to start, try a questionnaire. The advantages are that it is quick, cheap, highly relevant, and will give you a good guide for your qualitative research.
(2) **Big data collection**: Generally speaking, the data for building user portraits comes from website transaction data, user behavior data, and web log data. Of course, it is not only limited to these data, but also personal credit data on some platforms.
(3) **Scenario survey**: to go deep into the user's daily life environment to survey your users. This is the closest way to survey the user, you can find out what specific problems the user will encounter in the process of using the product.

7.2.2 Segmentation of User Groups

By tagging different users with static data and dynamic data, through the weighting and arrangement of tags, you can get a lot of user tags, and select the user tags needed to find the corresponding users according to the needs of their own products. For example, e-commerce companies choose users' basic data in addition to their interests, consumption habits, spending power, access records, store browsing time, historical purchase records, etc. to recommend to users goods with higher matching

degree with their tags, and promote users to visit other goods more often so as to generate re-purchase behavior.

7.2.3 Building a User Profile

There are already mature methods for constructing user profiles, such as Alen Cooper's "7-step persona method" and Lene Nielsen's "10-step persona method", which are very practical and professional methods for user These are very practical and professional methods of user profiling.

The general process of constructing a user profile is as follows

(1) Data source: Generally speaking, the data for building user portraits comes from website transaction data, user behavior data and web log data. Of course, it is not only limited to these data, but also personal credit data on some platforms.
(2) Data pre-processing: The first step is cleaning, cleaning some messy and disordered data, then grouping them into structured data, and finally, standardizing the information. We can understand the data pre-processing simply as classifying data in a table, this step is to lay the cornerstone of data analysis.
(3) Behavioral modeling: text mining, natural language processing, machine learning, predictive algorithms, clustering algorithms. Machine learning requires some mathematical foundation, such as what statistics, linear algebra, etc.
(4) User profiling: Through the previous series of means, we can classify the data into dimensions such as basic attributes, purchasing ability, behavioral characteristics, interests, heart characteristics, and social networks.

7.2.4 Application of User Profiling

(1) **Personalized recommendation**: for e-commerce and content platforms, the user will visit the subdivision into many attribute tags, according to the user real-time tag changes and constantly refresh the user model, so that and constantly will refresh the recommended content.

Common examples: e-commerce platform product (Taobao) recommendations, and today's headlines content recommendations, according to the label composition of the user portrait model, and then use their recommendation algorithm mechanism to match the user's interest in the content, to achieve personalized recommendations, thousands of people.

(2) **Advertising precision marketing**: today's mobile advertising has been fully applied to the user profile as the basis for placement, whether it is e-commerce, games or other brand exposure, the use of user profile data to guide advertising, not only to reduce costs, but also can greatly promote the click-through rate and conversion rate, to improve the overall advertising effect.

(3) **Personalized services**: Some industries pinpoint their services to find users and recommend customized services. For example, a clothing design company targets working men over 25 years old and provides them with quarterly clothing matching services, recommending clothes matching sets for them every quarter according to their budget and preferences.

7.3 Knowledge Graph Construction

The original data, according to the different structured forms of the data, adopt different methods to convert the data into the form of triads, and then perform knowledge fusion on the data of the triads, mainly entity alignment and combination with data models, and after the fusion, a standard data representation is formed, and in order to discover new knowledge, implicit knowledge can be generated based on certain inference rules, and all the formed knowledge After certain quality assessment, the knowledge formed finally enters the knowledge graph, and based on the data platform of the knowledge graph, some applications such as semantic search, intelligent Q&A, and recommendation system can be realized.

7.3.1 Knowledge Extraction

Knowledge extraction, i.e., knowledge extraction from different sources and structures of data to form knowledge (structured data) deposited into Knowledge mapping. We classify the raw data into structured data, semi-structured data and unstructured data, and according to different data types, we use different methods to process them.

(1) **Structured data**: For structured data, usually data from relational databases with a clear data structure, converting data from relational databases to RDF data (linked data), the commonly used technology is D2R technology.
(2) **Semi-structured data**: mainly refers to those data that have a certain data structure, but need further extraction and collation. For example, data from encyclopedias, data from web pages, etc. For this kind of data, wrappers are mainly used for processing.
(3) **Unstructured data**: For unstructured text data, we extract knowledge including entities, relationships, and attributes. There are three corresponding research problems, one is entity extraction, also known as named entity recognition, where entities include concepts, people, organizations, place names, time, etc. The second is relationship extraction, that is, the relationship between entities and entities, which is also an important knowledge in the text and requires certain technical means to extract the relationship information. The third is attribute extraction, which is the attribute information of an entity, and is similar to relationship.

Knowledge acquisition aims to construct knowledge graphs from unstructured text and other structured or semi-structured sources, complete existing knowledge graphs, and discover and identify entities and relationships. Well-constructed and sizable knowledge graphs can be used in many downstream applications and augment knowledge-aware models with knowledge inference, thus paving the way for artificial intelligence.

7.3.2 Knowledge Integration

Knowledge fusion, i.e., merging two Knowledge mapping (ontologies), the basic problem is to study how to fuse descriptive information about the same entity or concept from multiple sources. The main problem to be solved in this process is entity alignment. Different knowledge bases have different focuses of collecting knowledge, and for the same entity, some knowledge bases may focus on the description of some aspect of itself, while others may focus on describing the relationship between the entity and other entities, and the purpose of knowledge fusion is to integrate the descriptions of entities from different knowledge bases to obtain a complete description of the entity.

For example, there are some differences in the descriptions of the historical figure Cao Cao in different knowledge bases such as Baidu, Interactive Encyclopedia and Wikipedia. Cao Cao's main achievements are listed in Baidu as "implementing the cantonment system, pacifying the exiles and eliminating the others, unifying the north, laying the foundation of the Cao Wei regime, creating Jian'an literature, and advocating thin burial", in Interactive Encyclopedia as "unifying the north", and in Wikipedia as "unifying the core areas of the Eastern Han Empire".

It can be seen that there are still some differences between different knowledge bases for the description of the same entity, the difference in the description of the era to which they belong lies in the specific degree of chronology, the difference in the main achievements lies in the different scope of achievements, etc. Through knowledge fusion, the knowledge in different knowledge bases can be complementarily fused to form a comprehensive, accurate and complete description of the entity. The main work involved in the process of knowledge fusion is entity alignment, which also includes relationship alignment, attribute alignment, and can be realized by similarity calculation, aggregation, clustering and other techniques.

7.3.3 Knowledge Storage and Graph Database Neo4j

The knowledge store, i.e., the triples obtained and schema how it is stored in the computer. Using relational databases for storage, especially for simple knowledge graphs, is technically no problem at all. However, with the advent of the era of big data, the conventional relational data technology seems to be overwhelmed, so new

databases that can cope with massive amounts of data have emerged, and Neo4j is one of them.

Neo4j, a mainstream graph database, is relatively new to most people, but Neo4j has the following features and advantages.

(1) Native Graph is used to store and process data: providing optimized relational traversal execution efficiency, thousands of times faster than relational database table joins.
(2) (Label)-based attribute graph model: supports rich data semantic descriptions and is flexible.
(3) Based on pure Java implementation, it supports the widest range of operating systems and the most convenient deployment, supporting cloud and container deployment.
(4) Provides graph-oriented analysis and pattern matching, declarative Cypher query language that is intuitive, brief, and easy to understand.
(5) Causal Clustering-based distributed database that provides high availability, failover, data redundancy and scalable throughput.
(6) Rich driver language support: the official release of Java, JavaScript, Python,.Net and GO. In addition, there are community users provide C/C++, R, JDBC, Python and other language drivers.

7.4 Knowledge Reasoning and Interpretability

The so-called intellectual reasoning is the process of inferring unknown knowledge on the basis of existing knowledge. By starting from the known knowledge, new facts are obtained from it through the already acquired knowledge, or generalization from individual knowledge to general knowledge is made by generalizing from the large amount of existing knowledge.

7.4.1 Knowledge Discovery and Reasoning

After obtaining the representation of the knowledge graph, we have a part of the facts, and knowledge inference of the knowledge graph is to infer new knowledge or identify the errors of the existing knowledge on the knowledge graph based on the facts of the existing knowledge graph. According to the two roles of inference, we can naturally think of two downstream tasks. The first task is the complementation of the knowledge graph, and the second task is the denoising of the knowledge graph, and the so-called denoising of the knowledge graph is the identification of the wrong triples in the knowledge graph.

From the perspective of knowledge graph complementation, what we want is to use the existing complete triples to complement the triples of indeed entities or relations. That is, given two elements, use the existing triples to reason about the missing parts.

For example, given a head entity and a relation, use other triples on the knowledge graph to reason about the tail entity. Another example is that given the head and tail entities, use the triples on the knowledge graph to deduce the relationship between the two.

7.4.2 Rule-Based Knowledge Reasoning

Rule-based reasoning mines and reasons by defining or learning the rules that exist in knowledge. AMIE and AMIE+ algorithms derived from the earlier ILP (Inductive Logic Programming) system emphasize on learning rules with high confidence from large-scale knowledge graphs quickly and efficiently by automated rule learning methods and applying them to reasoning tasks.

7.4.3 Graph-Based Knowledge Reasoning

Path Ranking Algorithm (PRA) is a kind of knowledge reasoning based on graph structure, which is usually used for link prediction tasks in knowledge graphs. Because the relational paths it acquires actually correspond to a kind of Horn clause, the path features computed by PRA can be converted into logical rules that facilitate one to discover and understand the hidden knowledge in the knowledge graph. To learn an inference model for a particular edge type in the knowledge base, PRA finds sequences of edge types of frequently linked nodes that are instances of the edge type being predicted. PRA then uses these types as features in a logistic regression model to predict the missing edges in the graph. A typical PRA model consists of three components: feature extraction, feature computation, and relationship-specific classification.

(1) The first step is to find a set of potentially valuable path types to link entity pairs. To do this, PRA performs a path constrained random walk on the graph to record a finite length starting from h to t.
(2) The second step is to calculate the values in the feature matrix by computing the random walk probabilities. Given a node pair (h, t) and a path π, PRA computes the eigenvalues as the random walk probability $p(t \mid h, \pi)$, i.e., the probability of reaching t when given a random value starting from h, and the relationship contained in π.
(3) The last step is to train each relationship using logistic regression algorithm to get the weights of the path features.

The PRA model not only has high accuracy, but also greatly improves the computational efficiency, providing an effective solution to the problem of reasoning about large-scale knowledge graphs.

7.4.4 Neural Network-Based Knowledge Inference

Knowledge graphs are nowadays an important research direction in cognition and artificial intelligence due to their ability to characterize structured relationships between entities. Graph neural networks use deep neural networks to integrate topological structure information and attribute feature information in graph data, which in turn provides a more fine-grained feature representation of nodes or substructures and can be easily combined with downstream tasks in a decoupled or end-to-end manner, skillfully meeting the requirements of knowledge graphs for learning attribute features and structural features of entities and relationships.

Research related to combining knowledge graphs and graph neural networks has become a hot direction in the field of artificial intelligence. Knowledge graphs can provide good a priori knowledge for various learning tasks, and graph neural networks can better support the learning tasks of graph data. However, there are relatively few studies on knowledge graph learning, computation and application based on graph neural networks, and there is still huge room for future development, such as automatic construction of knowledge graphs based on graph neural networks, knowledge fusion based on heterogeneous graph neural networks, complex inference of knowledge graphs based on meta-paths or graph neural networks, and interpretable learning based on graph neural networks.

7.4.5 Interpretability Analysis of Knowledge Reasoning

With the development of deep learning, the model structures of knowledge inference methods are becoming more and more complex, and just one network may contain hundreds of neurons and millions of parameters. Although these inference models outperform humans in many aspects such as speed, stability, portability, and accuracy, users still cannot trust the prediction results of the models in risk-sensitive domains because they cannot intuitively understand the parameters, structures, and features in such models, and they know little about the decision-making process of the models and the inference basis of the models, and they know little about the decision-making process of the models and do not know when they will be wrong. In risk-sensitive domains, users still cannot trust the model's prediction results. Therefore, in order to build trust between users and inference models and to balance the contradiction between model accuracy and interpretability, interpretable knowledge inference has become a hot topic in scientific conferences in recent years.

7.5 Business Big Data Knowledge Fusion Case

7.5.1 Introduction to Knowledge Fusion Tools

(1) **Ontology alignment-FaIcon-AO**

Falcon-AO is an automatic ontology matching system that has become a practical and popular choice for matching Web ontologies expressed by RDF(S) and OWL. The programming language is Java.

The matching algorithm library contains four algorithms, V-Doc, I-sub, GMO, and PBM. Among them, V-Doc is virtual document based linguistic matching, which is a collection of entities and their surrounding entities, nouns, text and other information as a form of virtual document. This allows us to operate with algorithms such as TD-IDF. i-Sub is string matching based on edit distance, which we have described in detail earlier. As you can see, both I-Sub and V-Doc are based on string or text level processing. Further on there is GMO, which is matching done on the graph structure of the RDF ontology. pbm does it based on the idea of divide and conquer.

The output of the GMO, together with the output of the V-Doc and I-Sub, is selected by the final greedy algorithm.

(2) **Limes entity matching**

Limes is a metric space-based entity matching discovery framework for large-scale data linkage, the programming language is Java.

Its main process is: 1. given the source data set S, target data set T, threshold θ; 2. sample selection, select sample points E from T to represent the data in T. The so-called sample points, that is, the points that can represent the distance space. Should be uniformly distributed in the distance space, the distance between each sample as large as possible; 3. filtering, calculate the distance between $s \in S$ and $e \in E$, m(s, e), using the triangle inequality for filtering; 4. calculate the similarity; 5. serialization, stored in the format specified by the user.

7.5.2 Technical Challenges of Knowledge Fusion

There are two main technical challenges to knowledge fusion today.

(1) Data quality challenges: e.g. naming ambiguities, data entry errors, data loss, inconsistent data formats, abbreviations, etc.
(2) Challenges of data size: large volume of data (parallel computing), diversity of data types, no longer just by name matching, multiple relationships, more links, etc.

7.5.3 A Classic Case of Business Big Data Knowledge Fusion

IMDB lists the author of the film Don't Stop Dreaming as Aditya Raj, but Freebase's database lists the author as Adiya Raj Kapoor—are they the same person?

The mission of Amazon's Knowledge Graph is to answer all questions about products and related knowledge. Knowledge graphs have a very important application at Amazon. Recently, Amazon, in order to combine their different knowledge graphs, ended up using the novel cross-graph-attention and self-attention mechanisms, which were found to achieve state-of-the-art performance.

At Amazon, the Knowledge Graph is used to represent hierarchical relationships between product types on Amazon.com, relationships between creators and content on Amazon Music and Prime Video, and general information about Alexa's Q&A service, among other things.

In tests involving the integration of two movie databases, Amazon's system improved by 10% over the best performance of 10 benchmark systems, a metric known as area under the precision-recall curve (PRAUC), which evaluates the trade-off between true-positive and true-negative rates. PRAUC, which evaluates the tradeoff between true-positive and true-negative rates.

Amazon's work specifically addresses the problem of merging multiple types of knowledge graphs, or the merging of knowledge graphs where nodes represent more than one type of entity. For example, in the movie dataset we are dealing with, nodes may represent actors, directors, movies, movie genres, and so on. And edges indicate the relationship between entities, such as play, director, screenwriter, etc.

Chapter 8
Common Business Big Data Management and Decision Model

8.1 Robust Multi-task Learning for Clustering

In this chapter, we study the application of self-expression based subset selection method in cluster multi-task learning. Multitasking learning aims to learn multiple tasks together by sharing information among related tasks, which can improve the generalization performance of different tasks. Although multitasking has been shown to yield performance gains compared to single-tasking learning, the major challenge of learning which features to share with which tasks is still not fully addressed. In this chapter, we propose a robust clustering multi-task learning method, which divides tasks into different groups by learning representative tasks. The main assumption behind our approach is that each task can be represented by a linear combination of representative tasks that characterizes all tasks. The correlation between tasks can be represented by the corresponding combination coefficient. By applying row sparse constraints to the correlation matrix, our approach can select representative tasks and encourage information sharing among related tasks. In addition, the norm is used to represent losses to enhance the robustness of our method.

8.1.1 Background

Multitask learning is a hot research topic in the fields of data mining, machine learning, natural language processing and computer vision because many practical applications in these fields involve learning many related tasks, such as entity recommendation, travel time estimation, image description, human activity recognition, and so on. In order to improve the generalization performance of different tasks, multitask learning can learn multiple tasks simultaneously by transferring knowledge between multiple tasks. Specifically, multitask learning uses the intrinsic relationship between multiple tasks to share information between related tasks. For example, in human activity identification tasks, many activities are related and share

J. Cao, *E-Commerce Big Data Mining and Analytics*, Advanced Studies in E-Commerce,
https://doi.org/10.1007/978-981-99-3588-8_8

basic actions. Because of the experimental and theoretical advantages of multitask learning over single-task learning, multitask learning has developed dramatically in the past few decades. As a sub-domain of transfer learning, the key challenge of multitask learning is how to selectively transfer information between related tasks while preventing information from being transferred between unrelated tasks. Information transfer between unrelated tasks deteriorates the generalization performance of multitask learning, which is called negative migration. In multitask learning, the traditional ways to deal with this challenge can be divided into two categories. The first method assumes that all tasks are related to each other and can be implemented by two strategies: joint feature selection and multitask feature learning based on low rank structure. On the other hand, the second method assumes that all tasks can be clustered into groups, and only tasks within the same group are relevant and share information to a certain extent. This chapter focuses on cluster multitask learning. Although much progress has been made in task grouping in the past few years, there are still some major limitations in the existing clustering multitask learning methods. First, the number of clusters is usually unknown, which makes the task grouping method mentioned above less flexible in practical applications. Second, many task grouping methods divide tasks into disjoint groups. This hard allocation is not necessarily reasonable and can result in inefficient information sharing between tasks. At the same time, since the assumption that tasks in the same group are close to each other is based on the distance between tasks, negatively related tasks will be clustered into different groups, which prevents information sharing between them. Finally, few task grouping methods take into account robustness against abnormal tasks that do not share information with all other tasks.

Inspired by the selection of structured sparse subsets, we propose a new clustering multitask learning method for task grouping by selecting representative tasks. A subset of the representative tasks in the task set is selected and used to represent all tasks in a linear combination. The key point of the proposed method is that all tasks can be represented by representative tasks. Specifically, we first represent each task by its linear combination, where the linear combination coefficients form a task correlation matrix. Then, by imposing a row sparse constraint (norm) on the related matrix to encourage information sharing between related tasks, you can select the most representative and informative tasks at the same time. Finally, to enhance the robustness of the proposed method to abnormal tasks, norms are used to measure the loss of representation between each task and its linear combination of representative tasks, where norms and norms are used to constrain tasks and features, respectively. A recent work related to us identifies representative tasks by minimizing the weighted distance between each task and its representative task. However, since negatively related tasks are assigned to different groups, this method cannot completely discover the cluster structure of tasks.

The idea of representing each task as a linear combination of tasks is not new. However, our approach differs from existing methods in two respects. First, we represent each task with a linear combination of representative tasks by applying a norm to the correlation matrix, which makes each task share information only appropriately with related tasks. Secondly, we replace the square norm on the loss

with the cumulative norm to enhance the robustness of our method to the anomalous task. Our approach has the following advantages:

- The method can automatically learn the number of task groups from the data, without the need to be given in advance.
- Each task can be grouped into different groups based on representative tasks.
- Shared information can be passed between negatively correlated tasks.
- Since represents the cumulative norm on the loss. Our approach reduces the impact of unusual tasks.
- The objective function of the proposed method is an unconstrained double convex optimization problem.

8.1.2 Problem Formalization

Suppose we have a multitasking learning problem with m tasks, where each task i has a set of instances $D_i = \left\{\left(x_1^i, y_1^i\right), \ldots, \left(x_{ni}^i, y_{ni}^i\right)\right\} \subset R^d \times R$ and a linear function $f_i : f_i\left(x_j^i\right) = w_i^T x_j^i$, Where w_i is the weight of the i task, d is the dimension of data and the number of instances included in the i task. $W = [w_1, \ldots, w_m] \in R^{d \times m}$ is the weight matrix to be estimated, then the empirical risk is as follows:

$$L(W) = \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(W_i^T x_j^i, y_j^i\right)$$

The loss function $l(.,.)$ is the square loss for the regression problem and the logical loss for the dichotomy problem. In order to learn m tasks simultaneously, we follow the well-developed method of searching the weight matrix w in order to minimize the following regularized empirical risks:

$$\min_W L(w) + \lambda \Omega(W)$$

where Ω is the regularization term used to encode the prior knowledge of the task group structure.

8.1.3 Cluster Multitasking Learning Based on Representative Tasks

We propose a novel approach to integrate the idea of robust representative task selection into cluster multitasking learning. Specifically, those tasks with common representative tasks are considered a group, and all tasks can be grouped into

different groups based on their representative tasks. Formally, the proposed method is expressed as follows:

$$\min_{W,C} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(W_i^T x_j^i, y_j^i\right) + \lambda_1 W_F^2 + \lambda_2 (W - WC)_{1,2}^T + \lambda_3 C_{0,p}$$

The second regularization item controls the complexity of each task, the third regularization item represents all tasks through representative tasks, and the last regularization item controls the number of representative tasks. Note that we absorb the bias b_i into the weight w_i by defining an additional feature $x_0^i = 1$ for each instance in the task.

The optimization problem in the above formula involves calculating the number of non-zero rows of a matrix C, which is usually non-convex and NP. According to the recent theoretical advances in group variables, we relax the l_0 norm to its convex proxy l_1 norm. On the other hand, the typical value of p is $p \in \{2, \infty\}$, and we set $p = 2$ so that each task can be represented by a representative task with different weights. Therefore, the final optimization problem is as follows:

$$\min_{W,C} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(w_i^T x_j^i, y_j^i\right) + \lambda_1 W_F^2 + \lambda_2 (W - WC)_{1,2}^T + \lambda_3 C_{1,2}$$

In contrast to previous work on cluster multitasking learning, our approach automatically learns the number of clusters from the data. Each task can be grouped into different groups based on representative tasks. In addition, by representing each task with a linear combination of representative tasks, you can put negatively related tasks into the same group and share information.

8.2 Recommendations that Integrate User Interests

8.2.1 Background

With the development of high-precision positioning technology and the maturity and wide application of all kinds of mobile devices (smart phones, sports bracelets and car positioning devices, etc.), it has greatly changed all aspects of people's lives. More and more social media sites are springing up as new Internet platforms that connect cyberspace with the physical world. For example, Sina Weibo allows users to share information in the form of text, pictures, videos and other multimedia in real time, and add the user's current geographical location; Foursquare, Dianping, Jiepang and other apps provide users with the "check-in" function when they arrive at hotels, restaurants and scenic spots, and users can evaluate the products or services provided by the businesses. Apps such as Flicker, Panoramio and Instagram allow users to

upload and share photos online, adding text descriptions and geographic locations. Users actively or passively leave geographical location information in location-based social networks, resulting in large-scale spatio-temporal trajectory data (location check-in data, travel trajectory data, etc.). Massive temporal and spatial trajectory data record users' mobile activities in the real world, and contain rich personalized interest preference information.

However, the complexity of user interests and the sparsity of check-in data pose significant challenges to developing POI recommendation systems. For one thing, it's hard to explain what prompted a user to check in to a location based on a check-in record alone. Thus, how to model a user's true, explainable interests becomes a tricky problem. For example, when an unaccessed POI is far away from the user, the user may not access it, even if she likes the POI, due to external geographic restrictions. This presents a challenge in interpreting and modeling user decisions at POI check-in. Second, because the data is extremely sparse, recommendation systems face another key challenge. In a real system, there are millions of locations and users. However, each user has only a limited number of historical check-ins, making recommendations much more difficult. Existing methods have two limitations. First, their modeling of user preferences is so general that it fails to capture the real interests of users, let alone explain their check-in decision-making process. For example, a user's low predictive rating of an unvisited POI does not reveal why the user dislikes the location. Because he/she does not like the POI and will be limited by the external environment. Second, most existing methods treat all unseen feedback from users as negative in the same way, and thus fail to capture the inherent property of missing data, which is a mixture of missing negative and positive values.

In response to the above problems, we propose a unified approach that effectively learns fine-grained, interpretable user interests and adaptively models missing data. Each user's overall interest is a mixture of its endogenous interest, which is driven by personal interest and displays preferences without any restrictions, and exogenous interest, which is driven and influenced by external circumstances (i.e. geographical distance). To capture and distinguish the two, we first define a user's active region as a set of geographically accessible locations for that user, and then form them into paired ranking constraints in the unified recommendation approach. Specifically, based on endogenous interest, a user prefers each POI visited to any unvisited POI in the corresponding active area. On the other hand, in terms of outside interests, she preferred every POI she visited to any of the outdoor activity areas she hadn't visited.

8.2.2 Related to the Definition

Definition 1 (*endogenous interest*) is an intrinsic form of user interest, driven by personal interest preferences. For example, users can access a POI based on their own interests and preferences.

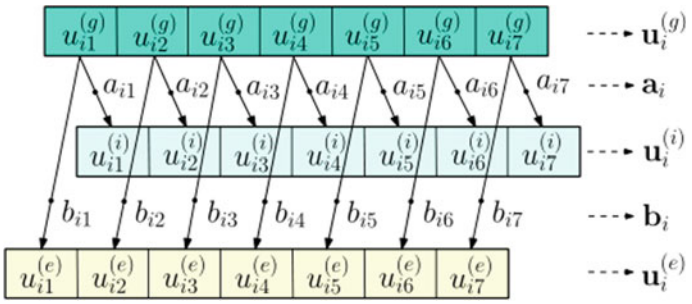


Fig. 8.1 $u_i^{(g)}$, $u_i^{(i)}$ and $u_i^{(e)}$

Definition 2 (*exogenous interest*) An external form of user interest that is context-driven. For example, users' interest preferences for POI are influenced by geographical distance.

These two types of user interests have different effects on each user's check-in decision. Thus, the overall interests of each user are considered a mixture of their intrinsic and extrinsic interests. Assume that the general, endogenous and exogenous interests of user i are represented by the d-dimensional vectors $u_i^{(g)}$, $u_i^{(i)}$ and $u_i^{(e)}$, respectively. As shown in Fig. 8.1, their relationship can be expressed as:

$$u_i^{(g)} = a_i \odot u_i^{(i)} + b_i \odot u_i^{(e)}$$

$$s.t.\, a_{ik} \in [0, 1], b_{ik} \in [0, 1], a_{ik} + b_{ik} = 1, \forall k \in \mathbb{R}_d$$

where $\odot$ represents the multiplication operation at the element level. $a_i \in \mathbb{R}^{d\times 1}$ and $b_i \in \mathbb{R}^{d\times 1}$ are the mixed weights of endogenous and exogenous interests respectively.

We also propose to capture features of missing data (or unobserved data), i.e. a mixture of negative and missing positive values, to solve the problem of data sparsity in location recommendation. It is assumed that in the matrix decomposition technique, each position j also has the characteristics of the potential vector $v_i \in \mathbb{R}^{d\times 1}$. Therefore, the overall loss function of our framework for endogenous and exogenous interests of modeling users and missing data is expressed as:

$$\min \mathcal{L}\big(\mathrm{P}, U^{(g)}, \mathrm{V}\big) + \theta^c\big(U^{(e)}, U^{(i)}, V\big) + \theta^r(\cdot)$$

8.2.3 Modeling Endogenous and Exogenous Interests of Users

According to Definitions 1 and 2, the distinction between endogenous and exogenous interests of users reveals whether there are external influences involved. In the POI recommendation task, the user's check-in decision process is significantly influenced by geography. Therefore, we focus on modeling user interests that have a geographic impact. Before describing how to model the two types of interests of a user, let's first define the user's activity area as follows:

Definition 3 (*user active areas*) Each user has one or more active areas within which the user can access each POI without any geographic restrictions.

The active area of each user can be calculated in various ways according to the actual application scenario. One approach is to use clustering techniques based on a user's historical location. In each cluster, we first select all locations within a circle with a specified distance radius, with each access location as the center, and then combine them into active regions. Modeling user endogenous interest $\boldsymbol{U}^{(i)}$. According to Definition 1, each user's intrinsic interest takes into account only his/her intrinsic interest preferences, regardless of any external geographic constraints. Within each active area, users can access each location geographically. This indicates that the user's check-in decision-making process is free at these locations within their active area. Thus, within each active area, the user's intrinsic interest in the observed sites, as opposed to those not observed, plays an important role in her decision-making process. In other words, based on intrinsic interest, each individual user i prefers each observed position j to any unobserved position l in each active region, which can be expressed as:

$$\left(u_i^{(i)}\right)^T v_j > \left(u_i^{(i)}\right)^T v_l$$

Modeling user exogenous interest $\boldsymbol{U}^{(e)}$. For locations that are far away from the user's area of activity, he/she has less opportunity to access these POIs. For example, although a user who lives in California likes a restaurant in New York City, he/she does not eat at the restaurant due to distance restrictions. Therefore, compared with the unobserved locations outside the user's activity area, the user's external interest in the observation location has a greater impact on their check-in decision than their endogenous interest. Therefore, based on exogenous interest, each user i prefers each observed position j to any unobserved position l outside the active region, expressed as follows:

$$\left(u_i^{(e)}\right)^T v_j > \left(u_i^{(e)}\right)^T v_l$$

The benefits of modeling a user's overall interest in two ways are: (1) By explaining the user's choice of location from both an endogenous and an exogenous perspective,

the behavior of the location recommendation system can be interpreted. (2) Provides a fine-grained and accurate way to understand user interests. Finally, the sorting constraint θ^c is derived as follows:

$$\theta^c(\cdot) = \sum_{i,h,j \in \mathcal{T}} \left(\lambda_c^i \sum_{l \in \overline{\mathcal{A}}_{ih}} \left(\hat{r}_{il}^{(i)} - \hat{r}_{ij}^{(i)} \right)_+ + \lambda_c^e \sum_{l \in \mathcal{A}_i^*} \left(\hat{r}_{il}^{(e)} - \hat{r}_{ij}^{(e)} \right)_+ \right)$$

8.2.4 Modeling Missing Data

A user's failure to access a POI does not necessarily indicate that he/she does not like it, but it may be due to her failure to discover the POI. In other words, some unvisited POIs may be of interest to those users, while others are actually of no interest to them. Thus, each user's preference for unobserved locations is a mixture of negative and unobserved positive values. Based on this intuition, we propose a location-oriented approach for adaptively learning the potential values of missing items, rather than treating them equally as predefined values. To this end, we introduce an augmented matrix matrix $P \in \mathbb{R}^{n \times m}$, which is only used for unobserved feedback and learned during training. Assume that user i predictive preference for location j is approximately $\hat{r}_{ij} = \left(u_i^{(g)} \right)^T v_j$. The empirical loss function under the square error can be expressed as:

$$l(\cdot) = \frac{1}{2} \left\| W \odot \left(\mathrm{R} + \mathrm{P} - \hat{R} \right) \right\|_F^2$$

8.2.5 A Recommendation Model that Incorporates User Interests

So far, we've introduced solutions that capture both endogenous and exogenous interests of users and address the missing data issues in POI recommendations. Integrate the above solutions into a unified framework.

$$\begin{aligned} \underset{\mathbf{U}^{(i)}, \mathbf{U}^{(e)}, \mathbf{V}, \mathbf{A}, \mathbf{B}, \mathbf{q}}{\operatorname{argmin}} & \frac{1}{2} \left\| \mathbf{W} \odot \left(\mathbf{R} + \mathbf{P} - \hat{\mathbf{R}} \right) \right\|_F^2 + \theta^r(\cdot) \\ & + \sum_{i,h,j \in \mathcal{T}} w_{ij} \left(\lambda_c^i \sum_{l \in \overline{\mathcal{A}}_{ih}} \left(\hat{r}_{il}^{(i)} - \hat{r}_{ij}^{(i)} \right)_+ + \lambda_c^e \sum_{l \in \mathcal{A}_i^*} \left(\hat{r}_{il}^{(e)} - \hat{r}_{ij}^{(e)} \right) \right), \end{aligned}$$

8.3 A Multi-objective Reinforcement Learning Framework for Community Deception

With the rise of social networks, community discovery algorithms play an important role in social network analysis. However, over-mining social network information can expose personal privacy, including personal interests, circles and related partners. How to effectively protect your privacy from being tracked by the community discovery algorithm, which hides the social relationships (community hiding) of your community. So far, there is little research work in this area. Existing algorithms for hiding communities are based on adding or reducing edges between points that exist in the network, but in real networks, such as FACEBOOK-based social networks, users are expected to change the structure information associated with them, especially specifying who should pay attention and who should be taken off, which is less operational and costly. To this end, this chapter intends to add points (i.e., fake accounts) and related edges (i.e., fake social relationships related to the account) to the existing network for the purpose of hiding the community; Secondly, this method (i.e. adding points and edges) intelligently transforms the hidden problems of the community into the growth problems of the network, and then, based on different network growth models, studies their different performances on the hidden of the community. Finally, the way of adding points and edges can essentially be described as a decision-making process, that is, according to the structure (state) of the current network, selecting the appropriate adding point and edge strategy to maximize the final indicator, which fits the framework of reinforcement learning. For this reason, this chapter proposes a community hiding algorithm based on multi-objective reinforcement learning for two hidden indicators and different network growth models.

8.3.1 Introduction to Community Hiding Algorithms

Currently, the research on community hiding at home and abroad mainly focuses on changing the network structure, that is, changing the connections of nodes in the network, which can be divided into two main categories: the hidden algorithm (G1) which only changes the structure within a specific community and the community hidden algorithm (G2) which can change the structure in the global network. Both of these algorithms achieve the purpose of community hiding by optimizing the hidden indicators proposed by each. For the G1 algorithm, Nagaraja et al. divides the network into two communities, the target community and the main community (excluding the points in the target community), whose hidden indicators are defined as Miss-ratio, also known as false Negative Rate. Based on this, the authors take Edge-Adding methods to target and main communities to improve the invalidation rate of community discovery algorithms. For Edge-Adding strategies, based on centrality measures, such as median centrality, degree centrality, and eigenvector centrality, the

points in the two communities are sorted in descending order by the centrality index, and two points are selected to edge in order to update the network structure. Waniek et al. proposed another hidden indicator (Concealment measure), which consists of two parts. The first part addresses the above-mentioned inefficiency problem, fully considering that the points in the target community should be dispersed among the rest of the communities, rather than just for the main community, that is, they should be evenly distributed among the communities. The second part of the indicator requires that as many points as possible be dispersed across communities. For the community discovery algorithm, a fast heuristic algorithm is proposed to reduce modules by edge reduction within the target community and edge addition between different communities. Fionda et al. proposed three criteria that are more suitable for community concealment indicators: after changing the network structure, the points within the target community should be as connected as possible, and the points within the target community should be evenly distributed among other communities and dispersed among larger communities. Based on this, the author proposes a deception score, and optimizes a security function to reach the purpose of community hiding. At the same time, the author proves that the function can only be optimized if the edge is reduced within the target community. For G2-class algorithms, Chen et al. proposed three heuristic algorithms and a genetic algorithm based on index to change the network structure while keeping the degree of each point unchanged, that is, adding an edge to a point while subtracting the related edge. From the perspective of information entropy, Liu et al. defined the information entropy of the original network structure and the information entropy of the network structure based on community division. The difference between the two information entropy (residual error) is the amount of information that the community structure brings to the network structure, and it minimizes the standard residual error by only adding edges to achieve the purpose of community hiding.

Table 8.1 shows the differences between the proposed model and the existing community hiding algorithms. The model proposed in this paper has three main differences: First, it is the first time to add points and edges for community hiding, which has less impact on users and is more convenient for site managers to operate; Second, learning the Edge-Adding strategy through the framework of reinforcement learning is also the first application of reinforcement learning in the field of community hiding. Third, unlike the existing work, which only optimizes a single goal, this paper optimizes both goals simultaneously by strengthening the learning framework, and can also expand to multiple goals.

8.3.2 Community Hiding Based on Multi-objective Reinforcement Learning

Different from existing algorithms that only add and subtract edges between existing nodes in the network, multi-objective reinforcement learning based community hiding is to add points (i.e., fake accounts) and related edges (i.e., fake social relations

Table 8.1 Differences from existing community hiding algorithms

Existing algorithms	Update network structure	Applying technology	Hide indicators	Category
Nagaraja	Edge only	Centrality measure	Miss-ratio (False negative rate)	G1
Waniek	Add or subtract edges	Modularity	Concealment	G1
Fionda	Add or subtract edges	Safeness and modularity	Deception score	G1
Chen	Add edge and subtract edge	Modularity	Modularity and NMI	G2
Liu	Edge only	Normalized residual entropy	Partition similarity and Mutual information and Query accuracy	G2
Algorithms for this article	Adding points and edges	Multi-objective reinforcement learning	Negative ratio association and ratio cut	G2

related to the account) in the existing network to achieve the purpose of community hiding, that is, to hide the relationship between users in the community. Secondly, the method (point and edge) cleverly transforms the problem of community hiding into the problem of network growth, and then based on different network growth models, the different performance of community hiding can be studied. Finally, the method of adding points and edges can be described as a decision-making process in essence, that is, according to the structure (state) of the current network, select appropriate strategies of adding points and edges to maximize the final index, which is exactly in line with the framework of reinforcement learning. Therefore, this chapter proposes a community hiding algorithm based on multi-objective reinforcement learning for two kinds of hiding indicators and different network growth models.

First of all, two indexes that can be optimized directly are defined. In the general network structure, communities are closely connected, but the connections between communities are sparse (i.e., community definition). Assume that there are P communities as $P_i = (V_i, E_i)$, V and E respectively represent the point set and edge set, A represents the adjacency matrix, $\mathrm{L}(V_i, V_j) = \sum_{i \in V_i, j \in V_j} A_{ij}$, $\overline{V_i}$ represents the points except community i. The index Ratio Association (RA) can be written as Formula (8.1), which represents the density of edges inside the community. Ratio Cut (RC) can be written as Formula (8.2), which represents the density of edges between communities. However, the hidden purpose of communities is to destroy this structure, namely, the inner communities are increasingly sparse. Communities are getting tighter, so this article defines hidden metrics as and based on RA and RC.

$$RA = \sum_{i=1}^{P} \frac{\mathrm{L}(V_i, V_i)}{|V_i|} \tag{8.1}$$

$$RC = \sum_{i=1}^{P} \frac{\mathrm{L}\left(V_i, \overline{V_i}\right)}{|V_i|} \tag{8.2}$$

The purpose of community hiding is to maximize the two indicators defined above, as shown in Formula (8.3), where $G = (V, E)$ represents the original network structure, $G' = \left(V', E'\right)$ represents the network structure adjusted by the community hiding algorithm, $V' = \left(V \cup V^{+}\right)$ and $E' = \left(E \cup E^{+}\right)$, V^{+} and E^{+} distribution represent the set of added points and the set of edges, $|*|$ represents the number of points in the set, and O represents the above two hidden indicators.

$$\begin{aligned} &\max_{G'} O\left(f(G), f\left(G'\right)\right) \\ &\text{s.t.} \quad \left|V^{+}\right| \leq d \\ &\left|E^{+}\right| \leq b \end{aligned} \tag{8.3}$$

In order to solve the above multi-objective optimization problem, from the perspective of reinforcement learning, the quadruple $(\mathcal{A}, \mathcal{S}, \mathrm{r}, \mathcal{P})$ is defined as follows:

- Action space $\mathrm{a} \in \mathcal{A}$: four edging strategies based on random, degree-based power-law distribution, degree-based and uniform distribution and log-plus distribution are selected as action space.
- The state space $\mathrm{s} \in \mathcal{S} = \{1, -1, 0\}^{\mathrm{k}}$: consists of the consistency policy of the nearest k index values. Formula (8.4) shows the consistency policy of two index values. If there are more than two indicators, the consistency of the policy can be determined by voting $\mathcal{S}^{\mathrm{t}} = \left\{\mathrm{s}_0^{\mathrm{t}}, \mathrm{s}_1^{\mathrm{t}}, \ldots, \mathrm{s}_{\mathrm{k}-1}^{\mathrm{t}}\right\}$

$$s_i^t = \begin{cases} 1, & O_{nra}^{t-i} \text{ and } O_{rc}^{t-i} \text{ Increase at the same time} \\ 0, & O_{nra}^{t-i} \text{ and } O_{rc}^{t-i} \text{ One goes up and one goes down} \\ -1, & O_{nra}^{t-i} \text{ and } O_{rc}^{t-i} \text{ Decrease at the same time} \end{cases} \tag{8.4}$$

- Reward value r: Set as the index value hidden by the community. If there are multiple indicators, it is a vector composed of multiple index values. Two indicators are used here, namely O_{nra} and O_{rc}, and the reward value obtained after each addition of points and edges is (O_{nra}, O_{rc}).
- State transition probability $\mathcal{P}$: Set this probability as 1, which is a deterministic decision.

To solve the multi-objective optimization problem, that is, select suitable points and edges according to the network structure to add the maximum hidden index. However, because there are multiple hidden indexes (based on two indexes in this paper, it can be extended to a maximum number), different hidden indexes will produce different reward values, thus corresponding to different strategies. In order to optimize multiple indicators at the same time to obtain the optimal consistency

strategy, two data structures are constructed based on the Q-learning algorithm in reinforcement learning to represent, as shown in Formula (8.5), where represents the average reward value based, represents the non-dominant policy set of the next state reached by taking actions under the state, represents the vector summation, and represents the loss rate. The multi-objective Q-learning algorithm based on Pareto optimality is specific, such as Algorithms 1 and 2. Among them, Algorithm 1 selects actions based on Hypervolume index through greedy algorithm. (Hypervolume index is an evaluation method consistent with Pareto, that is to say, if one strategy set π_1 is better than another strategy set π_2, Then the Hypervolume value of PI 1 will be greater than the Hypervolume value of PI 2). In Algorithm 6 represents the number of times it is selected.

$$Q(s,a) = \overline{R}(s,a) \oplus \gamma N D(s,a) \tag{8.5}$$

Algorithm 1 Community deception based on scalarized mulit-objective Q-learning

Input: The number of edge additions b, the set of node additions V^+, the number of iterations T, $\epsilon \in [0, 1]$.

Output: The rewired network.

Initialize $Q(\boldsymbol{s}, \boldsymbol{a}) \leftarrow 0, \forall(\boldsymbol{s}, \boldsymbol{a}) \in S \times \mathcal{A}$;
for episode $= 1$ to T **do**
  $\boldsymbol{s}^1 \leftarrow \{-1, 0, 1\}^k$;
  for $\tau = 1$ *to* b **do**
    Call the Algorithm 4 based on Eq. (12), i.e., SelectAction$(f(\boldsymbol{Q}, \boldsymbol{w}), Q, \boldsymbol{s}^\tau, \epsilon)$ and get the action $\boldsymbol{a}^\tau \in \mathcal{A}$;
    Update the network based on the action $\boldsymbol{a}^\tau \in \mathcal{A}$, i.e., add links between two nodes in V^+ and V, respectively;
    Get the next state $\boldsymbol{s}^{\tau+1}$ and reward $\mathbf{r}$ based on the action $\boldsymbol{a}^\tau \in \mathcal{A}$;
    Call the SelectAction$(Q_w, Q, \boldsymbol{s}^{\tau+1}, \epsilon)$ in which $\epsilon = 1$ and then get the action $\boldsymbol{a}^\tau \in \mathcal{A}$;
    for each objective o **do**
      update $Q(\boldsymbol{s}, \boldsymbol{a})$:
      $Q_0(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau) \leftarrow Q_0(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau) + \ell(\boldsymbol{r}_o + \gamma Q_0(\boldsymbol{s}^{\tau+1}, \boldsymbol{a}^{\tau+1}) - Q_0(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau))$;
    end for
    $\boldsymbol{s}^\tau \leftarrow \boldsymbol{s}^{\tau+1}$;
  end for
end for
Return the rewired network.

Algorithm 2 Community deception based on Pareto mulit-objective Q-learning

Input: The number of edge additions b, the set of node additions V^+, the number of iterations T, $\epsilon \in [0, 1]$.

Output: The rewired network.

2.1 Initialize $Q(\boldsymbol{s}, \boldsymbol{a}) \leftarrow 0, \forall(\boldsymbol{s}, \boldsymbol{a}) \in S \times \mathcal{A}$;
2.2 **for** episode $= 1$ to T **do**
2.3 $\boldsymbol{s}^1 \leftarrow \{-1, 0, 1\}^k$;
2.4 **for** $\tau = 1$ *to* b **do**
2.5 Call the Algorithm 4 based on the Hypervolum, i.e., SelectAction(Hypervolum, $Q, \boldsymbol{s}^\tau, \epsilon$) and get the action $\boldsymbol{a}^\tau \in \mathcal{A}$;
2.6 Update the network based on the action $\boldsymbol{a}^\tau \in \mathcal{A}$, i.e., add links between two nodes in V^+ and V, respectively;
2.7 Get the next state $\boldsymbol{s}^{\tau+1}$ and reward $\mathbf{r}$ based on the action $\boldsymbol{a}^\tau \in \mathcal{A}$;
2.8 Update the non-dominated set, i.e., $\mathrm{ND}(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau) = \mathrm{ND}(\mathrm{U_{a'}Q}(\boldsymbol{s}^{\tau+1}, \boldsymbol{a}'))$;
2.9 Update $\overline{\mathrm{R}}(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau) = \overline{\mathrm{R}}(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau) + \frac{\mathrm{r} - \overline{\mathrm{R}}(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau)}{\mathrm{n}(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau)}$;
2.10 Update $Q(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau) = \overline{R}(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau) \oplus \gamma \mathrm{ND}(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau)$;
2.11 **end for**
2.12 **end for**
2.13 **Return** the rewired network.

8.4 Mining of Periodic Coactive Populations in Trajectory Data

Moving objects equipped with positioning devices constantly generate a large amount of space–time trajectory data. One interesting finding about orbital flow is that groups of objects move together over a period of time. We observed that existing studies on co-moving population mining did not take into account an important correlation between co-animals, namely the recurrence of co-moving patterns. In this study, we pose the problem of discovering periodic co-motion patterns from flow tracks, allowing us to discover recent co-motion patterns that repeat over a given time period.

8.4.1 Background

With the popularity of positioning equipment, a lot of spatiotemporal data of moving objects are generated. Systematically analyzing trajectory data of moving objects can extract a variety of interesting patterns that can be used in many real-life applications, such as facility deployment and urban computing. One interesting finding

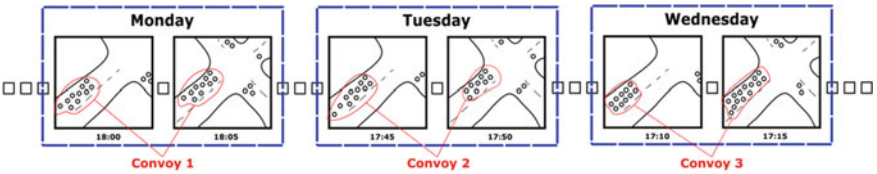


Fig. 8.2 An example of a traffic management application

in the track database is the exploration of the fleet of vehicles. In more detail, escort requires a continuous timestamp of at least a certain threshold for densely-connected group objects to move together. Essentially, interest escort is defined by the number of targets () and time spent moving together (), where and are user-specified parameters. A range of different escort modes have been proposed in the literature, including offline or online trajectory data processing. The definition and techniques for mining cooperative moving object patterns vary depending on the parameters and scenarios considered. However, the existing literature on co-moving object pattern mining usually deals with each mined pattern independently, ignoring the possible correlations between them. Considering the interrelationships between convoys, the importance of each convoy can be effectively assessed. In addition, real-time analysis of trajectory data is required in many practical applications. It is therefore important to respond to demining convoys that have occurred in the most recent window of time and to take corresponding measures in the light of historical events. Figure 8.2 shows an example of a traffic management application. This programme shows data on congestion occurring on a particular section of road on a given working day (i.e. convoys in close proximity exceeding a critical number of vehicles) and similar congestion patterns repeated on other working days. These patterns were found to be examples of patterns that are repeated on a daily basis in terms of time span, group size and spatial tightness. Our goal is to find these teams that show up regularly. Many applications can benefit from a fleet of historical events found in a sliding window. To name a few:

- Scenario 1: Transmission management system. Traffic apps can distinguish between unusual traffic congestion caused by accidents and regular traffic congestion during rush hour.
- Scenario 2: Military surveillance. A real-time military surveillance system can detect the repetition of a suspicious group with a common movement pattern.
- Scene 3: Marches and protests. The occasional large crowd needs to be distinguished from the average commuter.

Unofficially, a series of similar convoys form a regular fleet. The importance of periodic escorts varies with parameters such as the number of objects forming the convoy (prominence), the duration of the convoy (time span), and the time interval between the occurrence of two consecutive escorts (relapse). The value of the parameter defining the fleet interest may change over time or domain/context. Therefore, because the parameter values are not evenly distributed throughout the search space,

the exploration of cycle fleets of interest is an iterative process. Setting the appropriate values as query inputs gives us a new insight into the data set.

The main challenges of identifying recurring fleets in a track database are threefold. First, the convoy may not repeat at a fixed rate, that is, at the same time stamp and with the same object. The characteristics that make up a convoy may change from time to time. Therefore, similarity measures that use common goals to find collaborative motion patterns in most related work are not suitable to find correlations between fleets. Second, we can find the number of fleets that meet the query parameters in a sliding window. However, the number of times each resulting fleet is repeated, i.e., the recursion threshold is unknown. As a result, the search for previous occurrences within the recurrence threshold for each fleet varies in the resulting fleet. In order to speed up the mining, an efficient algorithm is needed to retrieve only potential candidate fleets. Third, the user may not know the appropriate value for each threshold to find the fleet of interest. This is an iterative exploration process, not a one-time parameter setting task. Therefore, we need an efficient index structure to facilitate excavation work to explore cycling fleets of interest.

8.4.2 Problem Formalization

Given a set of moving objects $O = \{o_1, o_2, \ldots, o_{|O|}\}$, $\tau = \{t_1, t_2, \ldots, t_\infty\}$ in the time domain, the trajectory of moving object $o \in O$ is expressed as a finite sequence of positioning samples based on the data in the time interval $[t_i, t_j] \subseteq \tau$, namely $o = \{loc_i, loc_{i+1}, \ldots, loc_j\}$. Where loc_a is the record position of t_a in two-dimensional space when the time stamp o is located. We assume that the trajectory of an object is recorded on each timestamp $[t_i, t_j]$ of its lifetime. Different objects' trajectories may have different lengths. Let $C = \{C^{t_1}, C^{t_2}, \ldots, C^{t_{|C|}}\}$ be the cluster set generated by applying the selected clustering algorithm on the trajectory database with different time stamps. $C^t (\in C) = \{c_1^t, c_2^t, \ldots, c_{|Ct|}^t\}$ represents the cluster set obtained with timestamp t, where O is the non-empty cluster satisfying the clustering condition. Since we assume that each trajectory of an object only corresponds to a finite time interval $[t_i, t_j]$, it is possible to have certain timestamp nonexistent clusters.

Example 1 Set the time span of the track database as $[1, 2]$, as shown in Fig. 8.3a. Suppose we found a total of 8 cluster $C = \{C^1\{c_1, c_2\}, C^2\{c_3, c_4\}, C^{10}\{c_5, c_6\}, C^{11}\{c_7, c_8\}\}$ in the database at 4 different timestamp $\{t = 1, t = 2, \ldots, t = 10, t = 11\}$, and no cluster was found from $t = 3$ to $t = 9$.

The goal of our approach is to find a cycle fleet that meets the threshold given by the user. In our work, an accepted definition of escort has been adopted.

Definition 1 (*fleet*) Given a set of cluster C and the threshold of the protrusions (τ) and the threshold of the timestamp (k), the convoy $g = \{c^{t_i}, c^{t_{i+1}}, \ldots, c^{t_j}\}$ is defined as a cluster sequence that satisfies the following constraints on the continuous

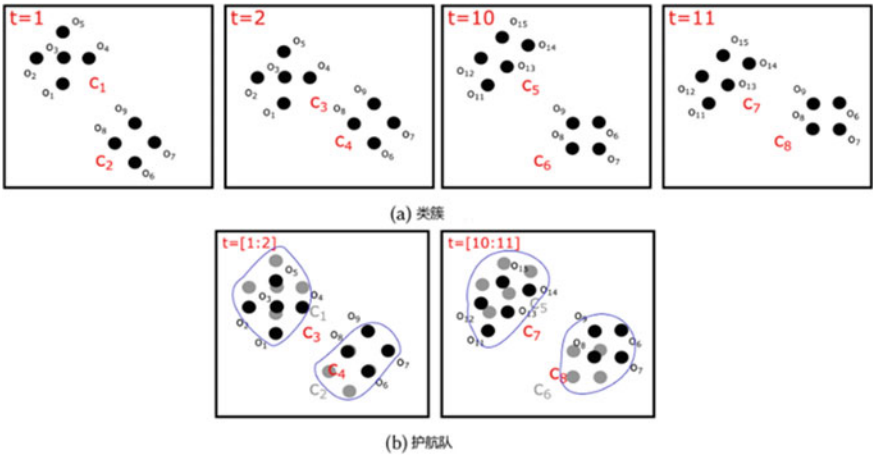


Fig. 8.3 The difference in threshold for cycling fleets

timestamp: (i) $\forall c^{t_a} \in g, \exists C^{t_a} \in C$ makes $c^{t_a} \in C^{t_a}$; (ii) The number of common objects shared by all clusters, denoted as $g.\tau$, does not exceed τ, that is, $g.\tau = \left|c^{t_i} \cap c^{t_{i+1}} \cap \ldots \cap c^{t_j}\right| \geq \tau$; (iii) The duration is denoted as $g.k$, not exceeding k, e.g. $g.k = \left|t_j - t_i + 1\right| \geq k$, where $k > 1$.

Example 2 Let $\tau = 4, k = 2$. Suppose we use the cluster in case 1 to generate a fleet that meets a given threshold. We got four fleets, and we intentionally included time stamps in the form of c_i^t in each cluster to represent the time stamps formed by the cluster.

Next, we define something like a fleet based on the interest threshold in Definition 2. If the similarity between the two object clusters c_1 and c_2 meets the minimum similarity threshold δ, the Boolean function $SIM(c_1, c_2, \delta)$ returns 1. There are multiple similarity indexes which can quantify the similarity between two object clusters and the choice of similarity measurement is application dependent. For the sake of illustration, Hausdorff distance is used to calculate the similarity between the convoy clusters. However, the problem definition and our methodology can be easily adjusted to other similar measures.

Definition 2 (*similar fleet*) Given thresholds τ, k and two convoys $g_a = \left\{c^{t_i}, c^{t_{i+1}}, \ldots, c^{t_{i+u-1}}\right\}, g_b = \left\{c^{t_j}, c^{t_{j+1}}, \ldots, c^{t_{j+v-1}}\right\}$. Convoy g_a is similar to Convoy g_b. (i) $MIN(g_a \cdot \tau, g_b \cdot \tau) \geq \tau$; (ii) $MIN(g_a \cdot k, g_b \cdot k) \geq k$; (iii) $\exists g_a' = \left\{c^{t_a}, c^{t_a+1}, \ldots, c^{t_a+k-1}\right\} \subseteq g_a, \exists g_b' = \left\{c^{t_b}, c^{t_b+1}, \ldots, c^{t_b+k-1}\right\} \subseteq g_b$, makes $\forall l \in [0, k-1], SIM\left(g_a', c^{t_a+1}, c^{t_b+1}, \delta\right) = 1$. As shown in Definition 2, two similar convoys contain at least τ objects and at least k timestamps. In addition, the corresponding clusters in the two k length subsequences of similar vehicles satisfy the given similarity measure. We are now ready to introduce a recurring escort in Definition 3.

Definition 3 (ρ-*loop escort*) Given the convoy sequence $p_{i,j} = \{g_i, \ldots, g_j\}$, where each escort meets the threshold τ and k (by Definition 1) and the recursive threshold ρ, ρ, $p_{i,j}$ is a cyclic escort, two continuous escort $g_a, g_{a+1} \in p_{i,j}$ are similar (by Definition 2), and the difference between their start time stamps is no more than ρ. $|g_a \cdot t_s - g_{a+1} \cdot t_s| \leq \rho$, where $g \cdot t_s$ indicates the start time stamp of the escort g.

Example 3 Suppose we search for frequent escorts, namely $\tau = 5, k = 2$ and $\rho = 10$. We found that two fleets met the given thresholds, namely $g_1\{c_1, c_3\}$ and $g_3\{c_5, c_7\}$. If we assume that convoy $t = 1$ is similar to convoy $t = 10$, with time stamps from $g_3 \cdot t_s - g_1 \cdot t_s = 10 - 1 = 9 \leq \rho$.

From the point of view of data mining, mining cycle fleet is a one-time task. However, the parameters that define fleet interest (time span, significance, repeatability) may change over time as we accumulate more data. Therefore, we present the cyclic fleet query on a sliding window in Definition 4, which takes varying parameters as input and only considers the fleet in the sliding window (note that we skipped the parameter δ in Definition 4).

Definition 4 (*circular escort query*) Given a constantly updated track database and a window of length I, $RCQ\langle k, \tau\rangle$ aims to discover ρ-cyclic convoy P, where $\forall p_i = \{g_a, \ldots, g_b\}$ satisfies recursive constraint ρ (according to Definition 3), $\forall g_b$ is inside sliding window I, and $\forall g_j \in p_i$ is a valid escort.

8.4.3 *Mining Algorithm for Periodic Populations in Trajectory Data*

We use convoy index convective trajectory data for RCQ processing algorithm. The query processing (i.e., Algorithm 5) consists of two phases: (i) The cluster generated from the current timestamp examines the existing fleet candidates that meet the significance threshold. The fleet that expands and meets the time span threshold (i.e., Algorithm 3) is passed to the second stage. (ii) At a given time interval and threshold, the escort index (i.e., Algorithm 4) is used to perform a historical event check for each result escort received in the first stage. Therefore, the historical escort generation is performed only when an escort that satisfies τ and k is found in the sliding window.

Algorithm 3 Convoy generation using a lookup table

Input: set of convoys G^{t-1} at timestamp $t - 1$, clusters C^t identified at timestamp **t**, and prominence threshold τ

Output: set of convoys G^t extended at timestamp t

3.1 $G^t \leftarrow \emptyset, C_{ext} \leftarrow \emptyset$

3.2 *lookupTable* $\leftarrow$ mapObjectsToCluster(C^t)

3.3 **foreach** $g \in G^{t-1}$ **do**
3.4 $clusterMap \leftarrow \emptyset$
3.5 O $\leftarrow$ *getConvoyObject*(g)
3.6 **foreach** *object* $o \in O$ **do**
3.7 $c \leftarrow$ getOjbectCluster(*lookupTable*)
3.8 $clusterMap.push(c, clusterMap.get(c)+1)$
3.9 **foreach** *pair*(c, *count*) $\in$ *clusterMap* **do**
3.10 **If** count $\geq \tau$ **then**
3.11 Update object of g with $O \cap C^t.get(c)$
3.12 Add c to C_{ext}
3.13 Push *pair*(g, c) to G^t
3.14 Add $C^t \backslash C_{ext}$ to G^t as new convoys
3.15 return G^t

Algorithm 3 describes the formation generation process in the sliding window according to the given threshold τ and k lookup table. C_{ext} declares the G^{t-1} cluster to store the expansion candidate fleet C^t. First, create a lookup table, store the objects as keys, and use the cluster obtained by timestamp t as the value to store the corresponding cluster identifier. Objects in each candidate fleet are grouped according to the cluster identifier obtained by the time stamp and stored in the clusterMap. There is no cluster identifier for objects whose current timestamp is not recorded. Each candidate fleet's clusterMap contains clusters in time stamps. Since fleets can be dispersed into multiple fleets, we check each cluster τ meeting the threshold c in clusterMap, and add a pair of candidate fleets g and cluster c to G_t. Clusters that do not extend any candidate fleets will be added to the candidate set as new fleets.

Algorithm 4 Historic convoy generation using the convoy index
Input: R-tree *tree*, prominence threshold τ, timespan threshold k, recurrence threshold ρ, time interval offset t_{end}
Output: set of historic convoys S

4.1 $skip \leftarrow \emptyset, G \leftarrow \emptyset$
4.2 $\cup(C^t, t) \leftarrow retrieveClusters(tree, \rho, t_{end}, \tau)$
4.3 **foreach** $C^t \leftarrow$ *each cluster set is accessed in ascending order of timestamp t* **do**
4.4 $newSkip \leftarrow \emptyset, G^t \leftarrow \emptyset$
4.5 **foreach** $c^t \in C^t$ **do**
4.6 **foreach** *convoy* $g \in G$ *that intersects* C^t **do**
4.7 ***If*** *skip contains* g **then**
4.8 push c^t to g, push g to G^t
4.9 $newSkip.put$ (g, $skip.get(g)$-1)

4.10 ***If*** c^t *extends nothing* **then**
4.11 **foreach** $cprvious\ cluster\ c\ of\ c^t$ **do**
4.12 $g_{new} \leftarrow$ generate a convoy using clusters c and c^t if it satisfies τ
4.13 push g_{new} to G^t
4.14 $count \leftarrow$ derive the value using embedding of c^t
4.15 $newSkip.put\ (g_{new}, count)$
4.16 $S \leftarrow S \cup \{g \in \left(G \backslash G^t\right) | timespan(g) \geq k\}$
4.17 $G \leftarrow G^t$, $skip \leftarrow newSkip$
4.18 return S

Algorithm 4 uses the escort index to generate pseudo-code of historical escort according to the threshold given in the time interval $[t_{end} - \rho + 1, t_{end}]$. The lookup table in line 2.1 skips tracking the fleet time span currently being evaluated. The key in the lookup table represents the escort identifier, while the value represents the number of timestamps the escort lasts. We retrieve the cluster set in line 2.2 by timestamp sorting within a given time interval. Each cluster set of a particular timestamp retrieved from the index contains a set of previous clusters, common objects, and time intervals for objects. We then generate the fleet from the cluster by evaluating the cluster in ascending order according to its timestamp (line 2.3). Suppose we are currently evaluating the cluster-set C^t corresponding to the timestamp t. For each cluster $c^t \in C^t$, we evaluated whether C^t could extend any candidate fleet g (retained by G) in the evaluation (lines 2.5–2.6). For each retrieved candidate convoy g intersecting C^t, if it is indexed through the query table skip, we can learn about the common objects and their time span through the escort index. Therefore, we can skip the detailed intersection operation between C^t and g. In the absence of crossover, c^t can expand the escort through g, and retain the expanded escort g as one of the next generation's candidate escorts (Lines 2.7–2.8). Additionally, we insert the extended escort g as a new entry in new Skip, which is the query table used in the next iteration of timestamp $t + 1$ (line 2.9). Note that the count value g is reduced by 1, reflecting the fact that g has been expanded by the cluster c^t corresponding to timestamp t, so the remaining time of the cluster is also reduced. If the class cluster c^t does not extend any candidate fleets, and if they meet the threshold τ, we add this cluster with previous class clusters as new candidate fleets. The value represents the number of timestamps, at least for the objects observed in the same class cluster in subsequent timestamps (line 2.14). If k (line 2.16) is satisfied, the convoys that were not extended at timestamp t are added to the result set, and any convoys that the cluster expanded/generated at timestamp t are passed to the next iteration at timestamp t + 1 (line 2.17).

Algorithm 5 Recurrent convoy query in a sliding window

Input: sliding window W_I of length I, R-tree index *tree*, Convoys G^{t-1} at timestamp $t-1$, Object O^t recorded at timestamp t, prominence threshold τ, timespan threshold k, recurrence threshold ρ

Output: set of recurrent convoys P

5.1 $C^t \leftarrow getCluster(O^t, \tau)$
5.2 $G^t \leftarrow generateConveys(G^{t-1}, C^t, \rho)$
5.3 $G \leftarrow filterConveys(G^t, k)$
5.4 $t_{end} \leftarrow t-k$, $p \leftarrow \emptyset$
5.5 **while** G *is not empty* **do**
5.6 $g_{remove} \leftarrow \emptyset$
5.7 $g_{past} \leftarrow getHistoricConveys(tree, \rho, t_{end}, \tau, k)$ using Algorithm 4
5.8 **foreach** $g \in G$ **do**
5.9 **if** $isSimilar(g, g_{past}, \tau, k)$ **then**
5.10 push (g, g_{past}) to P, $found \leftarrow true$
5.11 **if** $found = false$ **then**
5.12 push g to g_{remove}
5.13 $G \leftarrow G \backslash g_{remove}$
5.14 $t_{end} \leftarrow t_{end} - \rho$
5.15 return P

Algorithm 5 Outlines the complete process of recurrent fleet queries when a slide window move and a set of object position updates arrive. The updated position of the objects at the current timestamp t is clustered using the selected clustering algorithm (line 3.1). The generated class cluster is filtered over the threshold τ. The existing candidate fleet G^{t-1}, which ends with timestamp $t-1$, will examine the possible extensions against the filtered cluster set, using the ideas presented in line 3.2. The candidate teams meeting the threshold k are added to the result team set G (line 3.3). Trend line 3.4 defines the upper limit of the historical fleet search time interval. We then assess whether any of the escorts in the resulting convoy set G are actually periodic escorts meeting the specified requirements (lines 3.6–3.16). More specifically, it initializes G_{remove}, which is a time set that stores a fleet of results that no longer need to be searched for again in the next time interval (line 3.6). Next, it uses Algorithm 4 on the escort index (Line 3.7) to retrieve the historical escort within the interval $[t_{end} - \rho + 1, t_{end}]$. The similarity between the historical and result teams in G is then evaluated (lines 3.8–3.11) and the historical events of each result team are added to the corresponding list (lines 3.12). If a resulting fleet has not been extended by any historical fleet within the time interval of the search, it will be added to G_{remove}, so it will not be evaluated in the next iteration (lines 3.13–3.14). Thereafter, the resulting fleet set is updated by removing those in the middle, and the next time interval for the historical fleet search is also updated by moving another

timestamp ρ (lines 3.15–3.16). This process is repeated until the resulting escort set G becomes empty. Finally, the P is returned to terminate the query (line 3.17).

8.5 A Purchase Prediction Method Based on Semi-supervised Multi-view Learning

Similar to other industries in the field of e-commerce, the primary concern of travel e-commerce platforms is to understand and predict users' online purchasing behavior in order to improve the conversion rate from "visit" to "purchase". Clearly, a small improvement in the order Conversion Rate (CR) can generate millions of dollars. For example, on Amazon, a 1% increase in order conversion (CR) translates into $1 million in sales revenue. This chapter will propose co-EM-LR, a purchase prediction method based on semi-supervised multi-view theory, which have been published in ECRA in 2019.

8.5.1 Feature Construction of co-EM-LR Model

For the online travel purchase forecasting task, Table 8.2 shows the characteristics of co-EM-LR input of the model and gives a concise explanation.

(1) ***Member$_i$***: Member$_i$ is a feature in user Statistics, which describes whether user i is a VIP member of the platform. From the perspective of behavioral research, user membership information can reflect users' trust in the platform, so it will affect the intention of online purchase. As the online tourism data in this paper has been desensitized, the user's personal information can only be used if the user is a member of the platform. Therefore, we take whether the user i is a member as the first feature of the online tourism feature project.

(2) **LVDays$_i$**: LVDays$_i$ is a feature of clickstream metrics that describes the number of days since user i last accessed a session. The frequency and time of users' recent visits are the key factors that affect users' online purchase intention, and these factors are also the key indicators to describe the degree of user activity. At the same time, the higher the frequency of recent visits and the shorter the interval, the more significant the purchase intention of users. LVDaysi is taken as the second feature of the online tourism feature project for the user i who has a recent click stream.

(3) **TotV$_i$**: TotV$_i$ is a feature of the "clickstream metric", which describes the total number of times (i.e. frequency) that user i visits the platform in the last month. Similar to features, features can also be used to describe user i's activity. This feature has been widely used in the research of online purchase prediction of

Table 8.2 Definition of characteristics (variables)

	ID	Variable	Description	*p*-value
Variables for recency clickstream	1	$Member_i$	Whether user I is a member (1 = yes, 0 = No)	√ ***
	2	$LVDays_i$	Days elapsed since user *i*'s last visit (Sigmoid)	√ ***
	3	$TotV_i$	Number of visits made by user *i* in last month	√ ***
	4	*PAvgVi*	Average price in dollars of products browsed by user *i* in last month	√ **
	5	$PDevV_i$	Stand. Dev. of price of browsed-products for user *i* in last month	**
	6	$LDwell_i$	Dwell time in seconds of the last session of user *i*	√ *
	7	$DwellAvg_i$	Average dwell time in seconds of user *i*'s sessions in last month	√ *
	8	$TotP_i$	Number of purchases made by user *i* in last month	√ ***
	9	$LPDays_i$	Days elapsed since user *i*'s last purchase (Sigmoid)	√ ***
	10	$MAvgP_i$	Average monetary spending in dollars of user *i* in last month	√ ***
Variables for current clickstream	11	$PAvg_j$	Average price of products within session *j* (Log)	√ ***
	12	$PDev_j$	Stand. Dev. of price of products within session *j* (Log)	√ **
	13	$Length_j$	Length of session *j* (Log)	√ ***
	14	$Dwell_j$	Dwell time in seconds of session *j* (Log)	√ ***
	15	$Search_j$	Which search engine is used in session *j* (0 = none, 1 = offsite, 2 = onsite)	√ ***
	16	$TRegions_j$	The entropy of travel destination distribution in session *j*	***
	17	$PTypes_j$	The entropy of travel type distribution in session *j*	**
	18	$RPages_j$	Percentage of pages containing travel product displays in session *j*	***

(continued)

Table 8.2 (continued)

	ID	Variable	Description	*p*-value
	19	$Location_j$	Average semantic Sim. between user *i*'s living city and departure cities of products in session *j*	***
	20	$Holiday_j$	Number of days between log-time of session *j* and the latest holiday	**
	21	*Weekendj*	Whether current time of session j is weekend (1 = yes, 0 = no)	√ **

Note (1) "Sigmoid" means the variable is normalized into [0, 1] by Sigmoid function $1/\left(1+e^{-x}\right)$
(2) "Log" means the variable is taken its logarithm as $\log_{10} x$
(3) Mann–Whitney U test: * < 0.05; ** < 0.01; *** < 0.001

traditional goods. Here, for user i with recent click stream, it will be the third feature of the online tourism feature project.

(4) **PAvgV$_i$**: PAvgV$_i$ is a feature of "clickstream metrics" that captures the average price (in dollars) of products that User *i* has viewed in the last month. The existing research on the purchase intention of traditional goods has confirmed that price is the key factor affecting users' online purchase, and the vast majority of users are also sensitive to the price of goods. For user *i* who has a recent click stream, we consider PAvgV$_i$ as the fourth feature of the online tourism feature project

(5) **PDevV$_i$**: PDevV$_i$ is a feature of "clickstream measurement", which describes the standard deviation of the product price that user *i* browsein recent one month. The existing researches on the online purchase decision of traditional commodities and tourism commodities only consider the price factor of the current browsing products, which deceivers the sensitivity of users to the price in the near future. At the same time, it does not consider the degree of dispersion or aggregation of recently focused product prices. Therefore, it is necessary to measure the degree of dispersion and agglomeration of package prices. For this reason, PDevV$_i$ is taken as the fifth feature of the online tourism feature project for the user *i* who has a recent click stream.

(6) **LDwell$_i$**: LDwell$_i$ is a feature of "clickstream metrics" which describes the dwell time of user *i* last session in seconds in the last month. Many sites often use time spent rather than clicks as a measure of how much a particular user likes a product. At the same time, existing research on traditional product purchase decision has taken the recent session stay time of users as a measure of user's liking for the product. Here, aiming at user i with recent click streams, we will consider LDwell$_i$ as the 6th feature of the online Tourism Feature project.

(7) **LDwellAvg$_i$**: LDwellAvg$_i$ is a feature of "clickstream metrics" which describes the average dwell time (in seconds) of all sessions that user *i* has accessed in the last month. Similar to feature LDwell$_i$, LDwellAvg$_i$ describes user's liking

from the perspective of average recent residence time. Here, aiming at user *i* with recent click streams, we will consider LDwellAvg*i* as the 7th feature of the online Tourism Feature project.

(8) **TotP$_i$**: TotPi is a feature of "buying behavior", which describes the total number of purchases user i has made in the past month. Existing studies have repeatedly mentioned that the recent purchase behavior of users is closely related to the current purchase intention. This feature reflects both the purchasing power of users and their trust in the platform. Here, for user i who has a recent click stream, we consider TotPi as the 8th feature of the online Travel Feature Project.

(9) **LPDays$_i$**: LPDays$_i$ is a feature of "purchase behavior", which describes the time interval (unit: day) between user i and the last purchase within a month. Considering that the purchase of traditional daily necessities has explosive power and periodic demand, for example, users have periodic demand for milk powder and diapers, existing studies have taken the recent purchase interval as one of the basis for online purchase prediction in the purchase prediction of these products. Due to the significant differences between travel bags and traditional products in price, demand and other aspects, the impact of the recent purchase interval of travel bags on online purchase prediction is worth in-depth analysis. Here, LPDays$_i$ is taken as the 9th feature of the online tourism feature project for the user *i* who has a recent click stream.

(10) **MAvgP$_i$**: MAvgP$_i$ is a feature of "purchase behavior", which describes the average spending of user *i* on travel packages in the last month (unit: USD). Existing studies have confirmed that the average spending of users can reflect the purchasing power of users and is one of the important factors affecting the purchase intention. Since package prices vary widely, it is important to measure the purchasing power of online travel users in terms of recent average spending. Here, for user *i* who has a recent click stream, we take MAvgP$_i$ as the 10th feature of the online tourism feature project.

(11) **PAvg$_j$**: PAvg$_j$ is a feature of "clickstream metrics" that describes the average price (in US dollars) user *i* paid to browse travel packages in the current session *j*. Similar to feature MAvgP$_i$ and PAvgV$_i$, PAvg$_j$ measures a user's purchasing power and price sensitivity in terms of product price, except that PAvg$_j$ measures the price of the product being browsed in the current session. Existing studies have also used PAvg$_j$ as a factor to measure users' current purchasing power and price sensitivity. Here, for all user *i* with the current clickstream, we consider its PAvg$_j$ as the 11th feature of the online tourism Feature project.

(12) **PDev$_j$**: PDev$_j$ is a feature of "clickstream measurement", which describes the standard deviation of the price of travel package that user *i* browsed in the current session j. Similar to characteristic PDev$_j$, PDev$_j$ considers the degree of dispersion or aggregation of these product prices that users are currently focused on. This feature has been used to measure the sensitivity of users to product price. Here, for all user i with the current clickstream, we take PDev$_j$ as the 12th feature of the online tourism feature project.

(13) **Length**$_j$: Length$_j$ is a feature of the "clickstream metric", which covers the length of user i's current session j (i.e., number of clicks). Previous studies have repeatedly mentioned user session length as a key factor in measuring users' online purchase intentions. At the same time, this feature has been widely applied to the feature construction of customers' online behavior data, for example, in the study of online purchase prediction on Tmall. At the same time, existing studies generally believe that the number of session pages and their stay time have a positive impact on purchase intention. Here, we include Length$_j$ as the 13th feature in the online tourism feature project for all users i who have the current clickstream.

(14) **Dwell**$_j$: Dwell$_j$ is a feature of "clickstream metrics" which describes the amount of time user i is staying in session j currently. Many sites often use time spent rather than clicks as a measure of how much a particular user likes a product. At the same time, existing researches also use the user residence time as the feature input classifier in the purchase prediction task. In our Tourism Characteristics project, conversational dwell time is used to measure visitor interest. Here, we will Dwell$_j$ as the 14th feature of the online Tourism Feature project to all user i with the current clickstream.

(15) **Search**$_j$: Search$_j$ is a feature of "clickstream metrics" that characterizes the search engine type used by user i in the current session j. In the online travel data in this paper, search behavior can be further divided into in-site search and off-site search. In-site search is the internal search engine of Tuniu Travel, while off-site search is transferred to Tuniu Travel through third-party search engines (such as Baidu, Sogou and 360 search, etc.). Existing studies have also confirmed that users' search behavior has a positive impact on purchase intention. Here, we take Search$_j$ as the 15th feature of the online tourism feature project for all users i who have the current click stream.

(16) **TRegions**$_j$: TRegionsj is a feature of "clickflow metric", which describes the entropy of tourist destination distribution of user i in the current session j. Unlike general merchandise (such as books, music, movies, etc.), the text description of travel packages is more trivial, meaning that 2 different travel packages may be very similar except for some subtle differences. For example, the route and schedule of scenic spots, hotel and transportation options and other factors change. Therefore, we do not directly use the distribution of the travel package ID (that is, Item_ID) to measure the degree of aggregation and dispersion of the session. Since travel packages can be classified and described from the perspectives of tourism region and tourism type, generally speaking, users with strong purchase intention may browse the travel package and take the region they are interested in as the destination or the tourism type, they are interested in as the target. Therefore, we use entropy to measure the degree of dispersion or concentration of travel regions and travel types associated with travel packages in the conversation.

$$TRegions_j = -\sum_{t \in T_j} \pi(t) \log_2 \pi(t)$$

where T_j represents the set of tourism regions extracted from session j. For each tourism region t in T_j, the tourism region distribution π specifies a rough π(t) obtained by calculating the ratio of the number of tourism regions t in T_j to the number of all tourism regions. For all users i who have the current clickstream, **TRegions**$_j$ is taken as the 16th feature of the online tourism feature project.

(17) **PTypes**$_j$: PTypes$_j$ is a feature of the "clickstream metric", which describes the entropy of the distribution of travel types of user *i* in the current session *j*. Similar to TRegions$_j$, PTypes$_j$ measures the degree of aggregation or dispersion of travel package attributes that the user browses in the current session *j*. PTypes$_j$ is also computed in the same way as the TRegions$_j$ definition. Here, PTypes$_j$ is the 17th feature of the online travel feature project for all user *i* that has the current clickstream.

(18) **RPages**$_j$: RPages$_j$ is a feature of "click flow metrics", which describes the proportion of travel package-related display pages that user *i* includes in the current session *j*. Customers with strong purchase intentions tend to browse relevant product pages to obtain rich information to help them make online purchase decisions. In the online travel data set of this article, the category page usually shows a group of travel packages with a specific theme (such as a trip to Beijing, an amusement park, an island, etc.). Therefore, in addition to the product pages themselves, category pages are also important for identifying customers' purchase intentions.

$$RPages_j = \frac{\#Category_j + \#Product_j}{\#Length_j}$$

where $\#Category_j$ and $\#Product_j$ are the number of classified pages and the number of travel product pages in session j, respectively. $\#Length_j$ is the length of session j. For all user i with the current clickstream, $RPages_j$ is taken as the 18th feature of the online tourism feature project.

(19) **Location**$_j$: Location$_j$ is a feature of "travel space–time measurement", which describes the distance similarity between the city where user *i* resides and the city where Session j travel package leaves. In order to reduce the expense and time cost, the customer usually chooses a city near his/her city of residence (inferred from the IP address) as the route departure city to start the trip. Here, we use the structural similarity of trees to define the semantic relationship between two cities.

(20) **Holiday**$_j$: Holiday$_j$ is a feature of "travel space–time metric", which describes the interval between the time of user *I*'s current session *j* and the time stamps of recent holidays (such as Mid-Autumn Festival, National Day, New Year's Day and Spring Festival, etc.). In fact, the travel industry is heavily influenced by holidays and weekends, so we have also factored holidays into our online travel features. Typically, as the holiday season approaches, not only does CR

increase significantly, but so does the volume of orders. Here, Holiday j is listed as the 20th feature of the online Travel feature project for all user i who have the current clickstream.

(21) **Weekend$_j$**: Weekend$_j$ is a feature of the "travel space–time metric", which describes whether the date of user i current session j is a weekend. According to the research on the online purchasing behavior of traditional products, the weekend factor is the important factor that affects the online purchasing behavior of users. Similarly, in order to measure the influence of weekend factors on online travel purchase decisions, Weekend$_j$ was selected as the 21st feature of the online travel feature project for all user i with the current clickstream.

8.5.2 Online Travel Customer Segmentation

From the perspective of customer needs, customers have different needs. In order to satisfy different customers, the platform must provide products and services that meet customer needs. In order to meet such diversified needs, customer groups need to be segmented according to different standards. From the perspective of customer value, different customers can provide different values for the platform. If the platform wants to know which are the potential customers of the enterprise, which are the most valuable customers, which are the loyal customers of the platform and which customers are the most likely to be lost, it must subdivide the customers. From the perspective of platform management, how to optimize the application of limited resources to different customers is a problem that must be considered by every platform. Therefore, it is very necessary to conduct statistics, analysis and segmentation of customers in customer management. Only in this way can the platform carry out targeted marketing according to different characteristics of customers, win, expand and maintain high-value customer groups, attract and cultivate customer groups with great potential, and provide basis for marketing decisions of the platform. In order to deeply understand and accurately predict the purchasing behavior of online travel users, the research in this chapter only takes browsing history as the basis of online user segmentation. Therefore, online users can be divided into three disjoint categories in the observation window:

- First-Time Visitors. Such users visit the site for the first time, which means that there is no historical information available to learn about these users other than the current click stream.
- Ever-Visited Users. Such a user has visited the site before, but has not visited it in the recent past, for example, in the last month. So in addition to some demographic information, only the current clickstream information is available.
- Recent-Visited Users. This type of user is very active, meaning they visited the site recently. So, recent and current click streams, and some demographic information is available.

8.5.3 Analysis of Online Travel Purchasing Patterns

In order to study the prediction of online travel purchase, the 3-week clickstream data of the typical travel season is selected as the research object. Then, necessary of clickstream data was done, including: (i) deleting short sessions with session length of 1; (ii) Delete suspected crawler users with abnormal online operations. The final data size is shown in Table 8.3. Specifically, D1 refers to a week during the summer vacation, D2 refers to the week before China's Golden Week (National Day and Mid-Autumn Festival), and D3 refers to a week during a typical working day (i.e. the off-season for travel). In addition, each session consists of pages clicked by the user over a certain period of time, also known as a sample (i.e. an instance). Each session can also be marked as whether or not a purchase was made based on whether or not the reservation page is included. Thus, the number of purchased sessions and the corresponding order Conversion Rate (CR) can be obtained.

The influence of holidays and weekends on online purchases: Considering the time cost factor, tourists usually choose to travel during holidays. Specifically, due to the proximity of China's national holidays (such as National Day and Mid-Autumn Festival), the order conversion rate (CR) of data set D2 is significantly higher than that of D1 and D3. Secondly, it can be seen from Fig. 8.4 that weekend factors also affect the order conversion rate (CR) of travel e-commerce. Since the weekend in D2 data is also the Mid-Autumn Festival and National Day, the order conversion rate (CR) of these two days is generally higher than the baseline. In order to avoid the effects of holidays, only D1 and D3 data sets are divided. It can be observed that the order conversion rate (CR) for both data sets declines from Friday and is generally below the baseline (i.e. the mean CR of the data set), while on weekdays, the order conversion rate (CR) is higher than the baseline. In real life, tourists usually arrange their travel itinerary in advance, so this phenomenon is reasonable. Finally, we observe the change in the number of access sessions per week, as shown in Fig. 8.4. Unlike traditional products, more access sessions are generated on weekdays (i.e., Monday through Friday). These results indicate that travel e-commerce platforms are greatly affected by holidays. Therefore, holiday and weekend factors must be taken into account when conducting online travel purchase forecasting research.

The correlation between user segmentation and purchase behavior is further divided into recently-purchased Users and recently-not-purchased Users by collecting the purchase behavior of recently-visited Users. Table 8.4 shows the order conversion rate (CR) for each type of user. Most of the users are First-Time Visitors, with D1 accounting for 52.3%, D2 61% and D3 55.6%. This is similar to the cold start

Table 8.3 Description of online purchase forecast data

Data set	Date time	Record	User	Session	Order	Order rate (%)
D1	2012 08.01–08.07	2,022,633	364,067	431,321	7284	1.69
D2	2012 09.24–09.30	1,980,299	341,878	403,032	10,236	2.54
D3	2012 11.01–11.07	941,930	190,292	217,692	2731	1.26

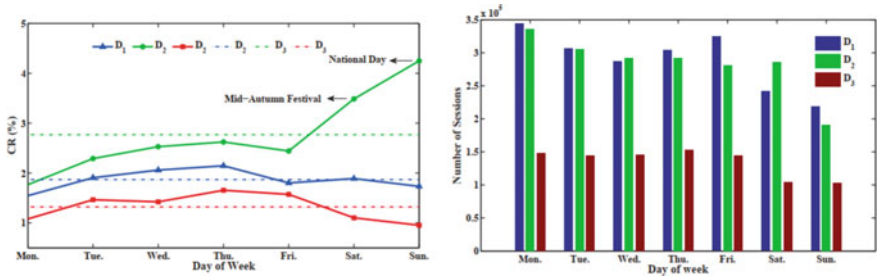


Fig. 8.4 Holiday factors in purchase forecast data

problem in the recommendation system. For First-Time Visitors, due to the lack of rich information available, Forecasting online purchases is complicated. In addition, as expected, existing customers (i.e., members) lead to a higher order conversion rate (CR). However, it is worth noting that Recent-Purchased Users have an unusually high order conversion rate (CR) of over 17% on all three data sets. This suggests that historical buying behavior is closely related to predicting future purchases. By comparison, we are fully aware that 60% of purchases are made by non-member users. The raw data used to understand the behavior of non-members is the current clickstream, and they view very few pages. Therefore, how to make the most of the clickstream information of the current session is very important for predicting tasks.

The impact of search behavior on purchase intent: In general, users often use search engines to find the information they need, and each current session also records whether users use search engines or not. We further divide the search behavior into in-site search and off-site search. It is worth noting that off-site search jumps to the platform page through external search engines (such as Baidu, 360 and Google, etc.). To quantitatively describe the influence of search behavior, we use two random variable indicators: $X = 1$ means that a session contains search behavior; Y is related to the purchase event, $Y = 1$ means that the related purchase behavior occurred, otherwise $Y = 0$. Then, we compared the conditional probabilities of the two sets of data. Table 8.5 shows the comparison results. It can be seen that the in-site search behavior is a strong purchase signal, because more than 95% of customers use in-site

Table 8.4 Influence of search behavior on order conversion rate

		D_1		D_2		D_3	
		Onsite (%)	Offsite (%)	Onsite (%)	Offsite (%)	Onsite (%)	Offsite (%)
$P(Y\|X = 1)$	Buy	95.12	3.74	96.23	4.19	94.78	4.69
$Y = \{0, 1\}$	Not-buy	4.88	96.26	3.77	95.81	5.22	95.31
$P(X = 1\|Y)$	Buy	21.76	61.47	25.61	53.25	20.39	59.65
$Y = \{0, 1\}$	Not-buy	1.67	56.83	1.78	49.37	1.22	58.89

Note "Onsite" and "Offsite" stand for *onsite search* and *offsite search*, respectively

search to buy products and more than 20% of consumers use in-site search in the purchase process.

In order to examine the impact of Recent visits and purchases on online purchase decisions, we focus on analyzing recent-visited users. To do this, a set of purchased user sessions were extracted and the time interval between the most recent access and purchase was counted. As can be seen from the cumulative distribution (CDF) in Fig. 8.5, both distributions have a significant long tail, which means that the recent behavioral influence is relatively significant. In addition, the effect of the last visit was stronger, meaning that approximately 90% of Recent-Visited users completed their purchase on their second visit within 5 days. However, the time between purchases has grown relatively slowly. Compared with Japanese department stores, which have explosive power and periodic demand, the purchase interval of tourism products is shorter. For example, about 60% of users make a second purchase within 3 days. We find that most of these products are low-cost travel packages (such as surrounding Tours, scenic spots and tickets). In addition, once a user buys a travel product, there is a relatively long interval until the second purchase. For example, about 18% of users buy a travel product for the second time 11 days after the first purchase. We find that most of these products are high-priced travel packages (such as domestic long or short line, overseas long or short line).

Next, the impact of the history interval on online purchases is analyzed. As mentioned above, users tend not to pay long-term attention to travel packages, that is, to leave a visit record on e-commerce travel websites, but tend to browse travel packages after they have travel goals and arrangements, which leads to a large number of cold start users in online travel data. Therefore, this paper attempts to analyze the effect of the time interval between the non-cold start user and the last access session on the online purchase behavior. As shown in Fig. 8.6, the CR of D1, D2, and D3 drops rapidly within 5 days since the last interview, and then slowly drops to a 30-day interval. In addition, on D1, D2, and D3, the session CR within 3 days of the last access was higher than the baseline (that is, the mean CR of the data set). In other words, users who returned to the site within three days were more likely to place an order. To sum up, just like traditional products, the history of travel e-commerce users is also crucial for the research of online purchase prediction.

Influence of geographic similarity on purchase intent: Then, measure a domain-specific factor that influences online travel purchase decisions, namely the distance between the customer's city of residence (inferred from the IP address) and the city of departure he/she clicks on. This factor corresponds to the variable Location_j. Figure 8.7 shows the results of buying versus not buying. It is clear that sessions with no purchase behavior mainly have smaller similarity values [e.g., fall on the interval $(0, 0.25)$ and $(0.25, 0.5)$], whereas the vast majority of sessions with purchase behavior have larger similarity values [e.g., fall on the interval $(0.75, 1.0)$]. The influence of variable Location_j explains that users who frequently browse travel packages from cities of departure close to their cities of residence tend to have strong purchase intentions.

Influence of entropy of travel region distribution on purchase intention: The travel package region visited by customers is represented by variable TRegions_j. The box

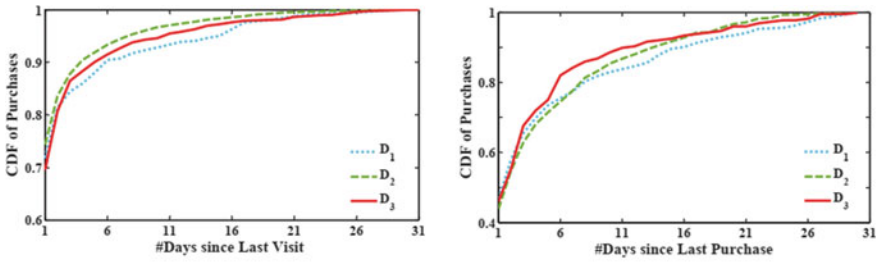
Table 8.5 An example of the fingerprint RSSI vector

Region	Fingerprint	S1	S2	S4	S5	S6	S7	S8	S9
Region 1	$\vec{f}_1$	62	78	89	66	77	62	73	82
	$\vec{f}_2$	53	72	88	67	76	65	77	85
	$\vec{f}_3$	63	68	90	59	74	66	74	83
	$\vec{f}_4$	40	65	85	60	72	66	79	86
Region 2	$\vec{f}_5$	42	61	81	62	74	72	77	83
	$\vec{f}_6$	55	50	85	66	72	78	84	90
	$\vec{f}_7$	61	44	79	61	70	73	81	88
	$\vec{f}_8$	66	41	83	67	75	77	83	89
Region 3	$\vec{f}_9$	72	48	75	67	70	77	82	88
	$\vec{f}_{10}$	76	55	70	68	73	78	85	90
	$\vec{f}_{11}$	70	50	74	62	69	75	80	85
	$\vec{f}_{12}$	73	65	72	66	67	75	78	82
Region 4	$\vec{f}_{13}$	79	72	67	70	68	73	74	79
	$\vec{f}_{14}$	77	70	66	70	65	70	71	77
	$\vec{f}_{15}$	81	76	63	75	67	75	72	77
Region 5	$\vec{f}_{16}$	81	78	59	76	70	77	75	78
	$\vec{f}_{17}$	82	78	53	77	72	80	73	79
Region 6	$\vec{f}_{18}$	64	53	75	53	66	69	73	80
	$\vec{f}_{19}$	67	54	74	55	65	68	71	78
	$\vec{f}_{20}$	61	65	77	38	67	61	73	81
	$\vec{f}_{21}$	66	70	75	45	53	58	69	77
Region 7	$\vec{f}_{22}$	70	68	74	62	59	71	75	80
	$\vec{f}_{23}$	74	70	68	66	58	73	70	75
	$\vec{f}_{24}$	68	71	73	63	42	68	66	71
	$\vec{f}_{25}$	73	74	71	69	45	68	64	70
Region 8	$\vec{f}_{26}$	63	69	79	57	66	50	65	72
	$\vec{f}_{27}$	68	72	77	60	58	52	61	69
	$\vec{f}_{28}$	66	71	81	65	68	47	64	75
	$\vec{f}_{29}$	71	74	78	66	65	45	60	68
Region 9	$\vec{f}_{30}$	70	75	74	67	63	64	52	63
	$\vec{f}_{31}$	75	77	79	70	65	66	54	60
	$\vec{f}_{32}$	73	75	77	69	68	62	49	66
	$\vec{f}_{33}$	78	77	80	70	67	68	47	58

(continued)

Table 8.5 (continued)

Region	Fingerprint	S1	S2	S4	S5	S6	S7	S8	S9
Region 10	$\vec{f}_{34}$	77	79	83	72	65	69	58	48
	$\vec{f}_{35}$	80	82	85	75	71	70	62	47
	$\vec{f}_{36}$	78	81	85	74	68	69	57	47
	$\vec{f}_{37}$	81	85	88	77	70	72	65	46



(a) Last Visit (b) Last Purchase

Fig. 8.5 Effect of recent visits and purchases on purchase intent

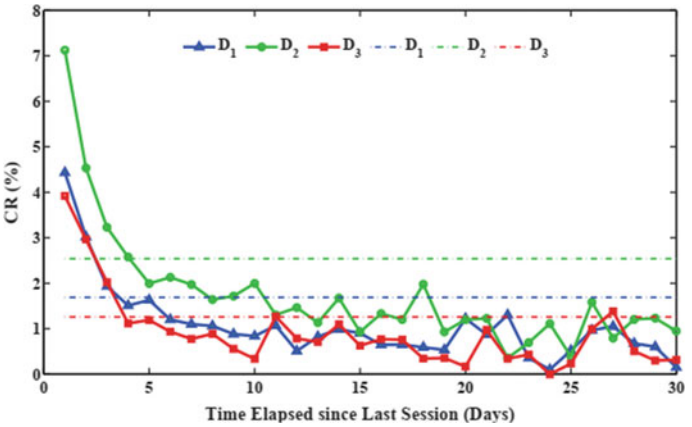
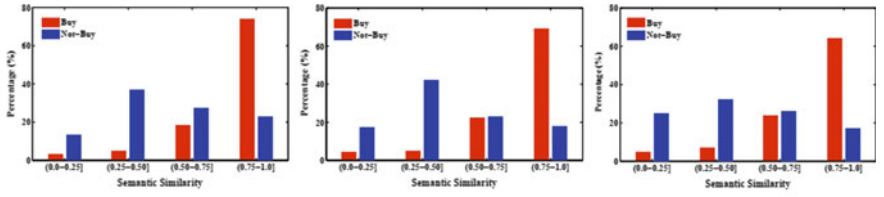


Fig. 8.6 Influence of history access interval on purchase intent



(a) D_1 (b) D_2 (c) D_3

Fig. 8.7 Influence of geographic location on purchase intention

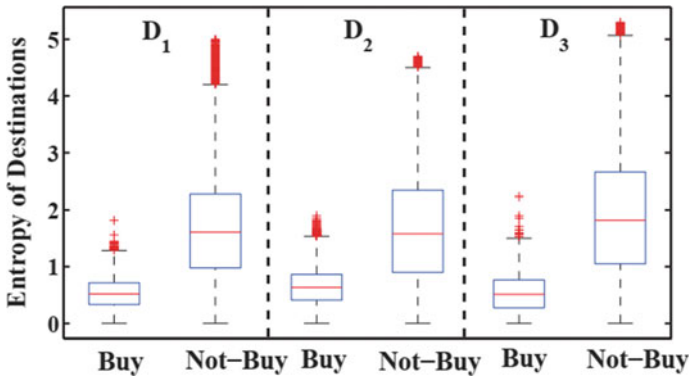


Fig. 8.8 Influence of entropy of tourism regional distribution on purchase intention

plot in Fig. 8.8 compares the entropy of the distribution of tourist regions, where a smaller entropy means that users focus on fewer tourist regions. Obviously, in the three tourism data sets, the median and coefficient of variation of the sample group with purchasing behavior are relatively small. This means that customers with a strong intention to buy will prefer to browse packages that are destined for the region they are interested in. In contrast, online users who browse a wide range of travel packages find it difficult to make a purchase decision.

8.5.4 Structure of the co-EM-LR Model

In order to complete online travel purchase forecasting, co-EM-LR has to address the following issues within its internal structure.

(1) The number of tag users used for training (that is, whether the user is tagged or not to purchase) is limited compared to untagged users, which may reduce the generalization ability of the supervised classifier.
(2) How to deal with two types of variables: one for the current session and the other for the recent (historical) session. Expect the two types of variables to work together to produce a consistent result.

To solve the above problems, a novel learning model called co-EM Logistic Regression (Co-EM-LR) is proposed for online travel purchase prediction. co-EM (co-Expectation Maximization) is a classic multi-view semi-supervised learning method. It is learning through the Expectation Maximization parameter estimation of the generative model jointly on two views. It is a semi-supervised classification model which combines multi-view learning and maximum expectation EM algorithm. co-EM-LR models are useful for combining Regression models with probabilistic models (maximum expectation EM algorithms). The double difficulty of the basic logistic regression model is extended to deal with the purchase prediction

problem. First, the number of tag users used for training (that is, whether users are tagged or not to buy) is limited compared to untagged users, which may reduce the generalization ability of the supervised classifier. Therefore, we hope to design a prediction model that can self-promote weak classifiers by using a large number of unlabeled samples and initially small-scale labeled samples. Second, two classes of variables are defined: one for the current session and the other for the recent (historical) session. Inspired by multi-view learning: Different views (i.e. categories) can learn collaboratively to improve the performance of the model. For this reason, the two types of variables are expected to work together to produce a consistent result.

8.6 Recommendation Based on Probabilistic Matrix Decomposition and Feature Fusion

This chapter introduces PMF-MAI, a recommended model for probabilistic matrix decomposition and feature fusion. Because of its simple model structure and clear and easy to understand model principle, PMF-MAI is very suitable for tourism product recommendation scenarios.

8.6.1 Application Scenarios of the PMF-MAI Model

The application scenario of PMF-MAI model is the tourism product recommendation scenario of Tuniu Travel, a large tourism e-commerce company in China. Users collect rich and comprehensive tourism product information through the online tourism platform for travel itinerary planning. In order to alleviate the problem of information overload, the online travel platform adopts the personalized travel recommendation system. Through analyzing and modeling the user behavior data, it predicts and recommends the tourism products that users may be interested in, so as to improve the service quality and attract and retain users. At the same time, users with purchase intentions can quickly find tourism products that meet their personalized needs. The ultimate optimization goal of PMF-MAI model is to increase the click-through rate of users as much as possible and accurately predict the click-through rate of users.

Different from traditional product recommendation, there are various types and huge quantities of tourism products in online tourism clickstream data. These factors lead to sparse explicit feedback data between users and travel packages (that is, the interaction matrix of click or purchase is extremely sparse). If the purchase matrix of users and travel packages is established, only about 0.1% of the elements are non-zero. This is much smaller than the data set for traditional products. For example, the recommendation algorithm predicts that the data matrix in the Netflix7 competition corresponds to 1.17% of non-zero values, and the density of the user-item score

matrix in the popular MovieLens100K data is 6.3%. Therefore, the sparsity of tourism data makes it difficult to directly apply traditional recommendation technologies to realize personalized travel package recommendation (such as collaborative filtering or matrix decomposition, etc.), which poses new challenges for personalized travel package recommendation. At the same time, the elements of tourism products are complex, and there are differences among different products, including the origin, destination and travel time, etc. The choice of tourism products is also affected by seasons and holidays.

8.6.2 Feature Construction of PMF-MAI Model

For this usage scenario, six global features are constructed. Different from user features and product features, global features are the interaction features between users and products. Global characteristics play an important role in the field of tourism product recommendation. For example, the city of residence of tourists has little effect on modeling user preferences, but when it is associated with the city of departure of tourism products, it is of great significance. The first type of feature is used to describe the distance between the residence of the user and the departure city of the tourist product, as well as the distance between the intended destination of the user and the destination city of the tourist product. Specifically, the residence of the user is obtained according to the corresponding IP address of the session; the product page's property Departure is to obtain the departure city of the tourism product; the user searches the Keyword to extract the intended Destination of the user; the product page's property destination is to obtain the destination city of the tourism product. We use two methods to measure the distance between two locations: geographic similarity and semantic similarity.

(1) **Degree of geographical similarity**

Given a set of places of p_i and p_j, use the Google maps API[1] geographic distance between computing place names, as the dd. Then, the minimum maximum normalization method was used to convert the distance into geographical similarity:

$$S_1(p_i, p_j) = 1 - \frac{D(p_i, p_j) - \min}{\max - \min}$$

[1] https://developers.google.com/maps/.

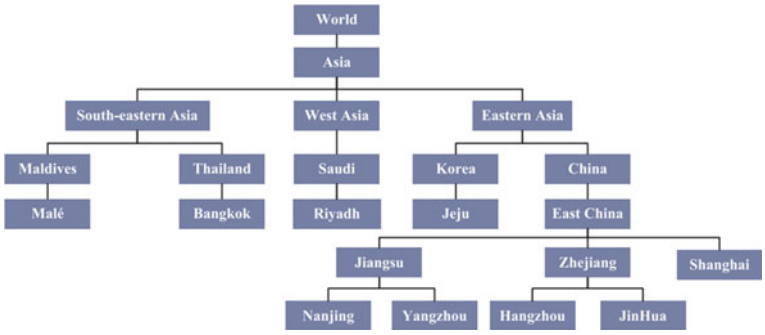


Fig. 8.9 Geographic tree structure

The geographical similarity between place names p_i and p_j is $S_1(p_i, p_j)$; $D(p_i, p_j)$ is the distance between place name p_i and p_j calculated by longitude and latitude data; max (min) is the maximum (minimum) value of $D(p_i, p_j)$.

(2) **Degree of semantic similarity**

Organize place names into a tree structure. For example, one possible path for this tree is: "Nanjing City, Jiangsu Province, East China." Node similarity in tree structure is used to define the semantic relationship between two place names. Specifically, the geographic hierarchy is constructed using the United Nations Geoscheme (UNG).[2] Figure 8.9 shows an example of this hierarchy, with semantic similarity defined as:

$$S_2(p_i, p_j) = \frac{H(p_i \cap p_j)}{H(p_i) + H(p_j)}$$

The semantic similarity between place names p_i and p_j is $S_2(p_i, p_j)$; Where $H(p_i \cap p_j)$ is the path length between the nearest common parent node of place name p_i and place name p_j in the geographic tree and the root node; $H(p_i)$ is the path length of the place name from the root node in the geographic p_i tree. Then according to Fig. 8.9, S2 (Phuket Island, Thailand) = 0.889, S2 (Jeju Island, Thailand) = 0.444. This means that the tourist has a higher preference for Phuket than Jeju in the choice of destination.

Prices and travel times vary widely between different travel products. For example, in the data set, the selling price of tourism products ranges from several hundred yuan to tens of thousands of yuan, and the travel time ranges from one day to more than a dozen days. Therefore, another type of feature is designed to express users' preference for tourism products in terms of economy and time cost. By referring to the methods in other literature, the user's preference for financial and time cost is modeled as Gaussian priori. Specifically, the maximum minimum normalization method is used to normalize the price attributes and calculate the mean value and

[2] https://en.wikipedia.org/wiki/united_nations_geoscheme.

standard deviation. The price probability function for each session is then obtained by assuming that the price follows a one-dimensional Gaussian distribution. Finally, two construct features are obtained to model user preferences in terms of price and time.

8.6.3 Structure of the PMF-MAI Model

In the last summary, PMF-MAI model has solved the problem of complex spatial and temporal characteristics of tourism by constructing global features. For the problem of sparse tourism, the PMF-MAI model will be mitigated by missing data (i.e. unobserved user-product interaction data).

The dimensional preference matrix $\boldsymbol{X}$ of $N \times M$, given N users and M items, aims to recommend new items of interest but no interaction to each user. In essence, the recommendation task is equivalent to predicting the missing values in the matrix $\boldsymbol{X}$ and recommending corresponding items to users by sorting the predicted values. Figure 8.10 Probability diagram representation of PMF-MAI model. PMF-MAI is divided into the following three modules. The following describes the functions and implementation of each module:

Probability decomposition based on user-product interaction matrix: The probability matrix decomposition model is used to decompose the user-product interaction matrix and establish the loss objective function:

$$\mathcal{J}_1 = \frac{1}{\sigma_{X1}^2} \left\| I^X \odot \left(X - U^T V \right) \right\|_F^2$$

where U is the user implicit factor matrix, V is the product implicit factor matrix, σ_{X1} is the parameter, $\|\cdot\|_F^2$ is the Frobenius norm, $\odot$ is the Hardamard product.

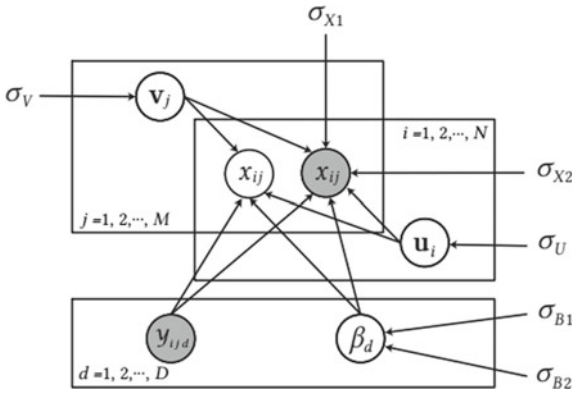


Fig. 8.10 Probability diagram representation of PMF-MAI model

Linear regression based on feature fusion: the constructed global features are aggregated into tensor form $\boldsymbol{Y}$, where $\boldsymbol{y}_{ij} = \left[y_{ijd}\right]_{D\times 1}$ represents D-dimension feature vector and y_{ijd} corresponds to the d feature value of user-product interaction. Similarly, linear regression is used to establish the loss function:

$$\mathcal{J}_2 = \frac{1}{\sigma_{X2}^2}\left\| I^X \odot (X - \boldsymbol{Y}_{\times d}\boldsymbol{\beta})\right\|_F^2 + \frac{1}{\sigma_{B1}^2}\|\boldsymbol{\beta}\|_F^2$$

Missing data processing: Using the missing data to update the parameters of the probability matrix decomposition and linear regression model to establish the loss function:

$$\mathcal{J}_3 = \frac{1}{\sigma_{B2}^2}\left\| \overline{I}^X \odot \left(\boldsymbol{Y}_{\times d}\boldsymbol{\beta} - U^T V\right)\right\|_F^2$$

where $\overline{I}_{ij}^X = 1 - I_{ij}^X$.

The loss function $\mathcal{J} = \alpha_1\mathcal{J}_1 + \alpha_2\mathcal{J}_2 + \alpha_3\mathcal{J}_3$ established by the three modules is combined. The training of the model can be completed by using gradient descent. U and V can be solved, and $\hat{X}_{ij} = U_i^T V_j$ of the users' clicks on the product predicted by the system can be recommended. According to the predicted clicks, products with high K before the predicted value can be recommended.

8.7 Indoor Positioning Technology Based on Asynchronous Sensor

This chapter introduces indoor positioning technology based on asynchronous sensor. Indoor positioning refers to the realization of location positioning in an indoor environment. At present, a variety of indoor positioning technologies have emerged. Among them, the wireless location method based on fingerprint has been widely concerned. The basic idea of this method is to infer the location of the user according to the Received Signal Strength Indicator (RSSI) of the wireless communication device.

8.7.1 Indoor Positioning Technology Background

In recent years, track data mining has attracted more and more attention, and has been widely used in urban calculation, such as traffic prediction, route prediction and so on. At the same time, by mining indoor tracking data and developing indoor location-based services, it has greatly promoted the development of indoor navigation, intelligent retail, social networking, health care and other fields. However, most

existing location methods require sensors deployed in different locations to simultaneously detect wireless signals to determine the user's real-time location. This requires not only the introduction of synchronization mechanism in the positioning system, but also the accurate measurement of two-dimensional coordinates of each reference point in the offline phase, which greatly increases the cost of human, material and financial resources. On the other hand, in many cases, people just want to know where the user is at a given time interval, not where the user is at a given point in time. For example, a manager of a large shopping mall might want to know which products are a big draw for customers and which are not. So we can predict which counters customers are more interested in by counting how many and how long they stay at each counter. In other words, we only need to know which counter the customer stops at over a period of time.

For this kind of problem, it can be solved by localization of indoor areas. Contrary to the traditional localization problem, the purpose of indoor area localization is to predict the location of the area where the object stays within a given time interval. For such problems, this book introduces the indoor area location method based on Bayesian probability model asynchronous sensing data. The core idea of this approach is to create a time window for each timestamp, and then introduce a time-decay aggregation method for user modeling, which translates the asynchronous induction data into a series of RSSI measurement vectors. The conditional probabilities for each fingerprint are then estimated and all of these probabilities are put together to obtain the predicted probabilities corresponding to the different regions. Finally, the region with the greatest probability of prediction within a given time interval is regarded as the residence region of the object. Since the area of the object is predicted, only the relative position of each measurement point needs to be recorded, rather than the exact coordinates, which greatly reduces the workload of the data acquisition process.

8.7.2 Asynchronous Sensing Method

With the popularity of wireless LAN and WiFi devices, WiFi positioning system is a passive positioning technology based on devices among many wireless technologies. The WiFi location system does not require additional infrastructure modification, but only requires multiple WiFi sensors to be placed in the indoor space. Each WiFi sensor is typically equipped with a 4G communication module, which then sends sensing data back to the cloud server.

At present, most WiFi positioning systems work through multiple sensors in an asynchronous way, that is, a user cannot be detected by nearby sensors at the same time. In order to achieve quasi-synchronous monitoring, the scanning interval can only be reduced, but because 4G communication is relatively expensive, the overall cost is too high. Many commercial systems do not have very short scan intervals. Meshlium, for example, scans WiFi frames every 20 s. But predicting the area of the room where the user will stay within a specific time interval means that it is

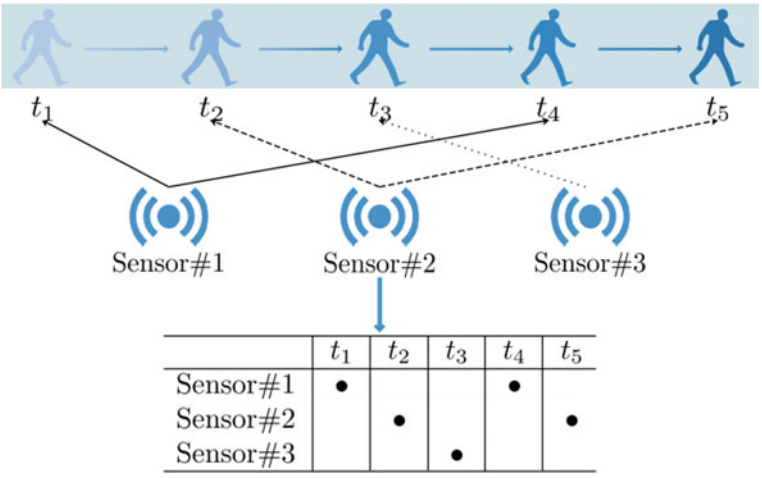


Fig. 8.11 Illustration of asynchronous scenario

not necessary to predict every stop the user will make. Therefore, each sensor is allowed to detect the WiFi signal at relatively long intervals (for example, 20 s), using asynchronous sensing to obtain data information.

Figure 8.11 shows a typical example of an asynchronous sensing scheme. Three WiFi sensors are deployed near the path, with detection intervals set to multiples of three. The user then walks along the path, carrying a WiFi-connected phone. During this process, he will be detected by one or more sensors. Sensor # 1 will detect the user at time points t1 and t4; sensor # 2 will detect the user at time points t2 and t5; sensor # 3 will detect the user at time t3. We can clearly observe that the detection events of these sensors occur at different times, or in other words, the user is detected asynchronously.

8.7.3 Indoor Area Location Method for Asynchronous Sensing Data

This chapter will introduce an indoor area positioning method based on asynchronous sensing data. The core idea is to arrange the asynchronous sensing data corresponding to a given user into a sequence X according to the detection time. According to the given time point t and time interval Δt, the records of detection time in the interval $(t - \Delta t, t]$ constitute the sub-sequence $(t - \Delta t, t]$; Calculate the weight of each record in the sequential records and convert the sequential records to a user RSSI vector; For each fingerprint RSSI vector, the similarity between the user RSSI vector and the fingerprint RSSI vector was calculated. For each small area, the similarity between the user's RSSI vector and the area is calculated. The area with the largest similarity

is the stay area of the user corresponding to Xt. This method can effectively solve the indoor location problem in asynchronous scenario. The following details how this method is implemented.

1. First of all, given the divided target region and several fingerprint RSSI vectors, the divided target region consists of several non-overlapping small regions, each cell domain has several reference points, each reference point corresponds to a fingerprint RSSI vector $\vec{f}_i = \left[s_{i1}, s_{i2}, \ldots, s_{ij}, \ldots, s_{id}\right]$, s_{ij} is the received signal strength RSSI of the wireless communication device detected by the j sensor. d is the number of sensors.
2. Given time point t and time interval Δt, the records within the detection time interval (t − Δt, t] constitute sub-sequence Xt.
3. The weight of each record in the sequential records is calculated and the sequential records is converted into a user RSSI vector $\vec{o}^t = \left[s_1^t, s_2^t \ldots, s_j^t, \ldots s_d^t\right]$, where s_j^t is the RSSI corresponding to the j sensor.
4. For each fingerprint RSSI vector, the similarity between the user RSSI vector and the fingerprint RSSI vector is calculated.
5. For each small area, the similarity between the user's RSSI vector and the area is calculated. The area with the largest similarity is the stay area of the user corresponding to *Xt*.

In a specific embodiment, the indoor location method based on asynchronous sensing data is described in detail.

Figure 8.12 shows the divided target area. Circles in the figure represent the position of reference points, and solid dots represent the position of sensors. The fingerprint vectors corresponding to each reference point are as follows. In particular, for example, $\vec{f}_1$ is the fingerprint RSSI vector corresponding to reference point 1.

When the user carries a wireless communication device into the area and turns on the Wi-Fi, the sensor deployed in the area can detect the received signal strength RSSI of the wireless communication device, and store the detected information to form asynchronous sensing data. The asynchronous sensor data corresponding to the user is as follows:

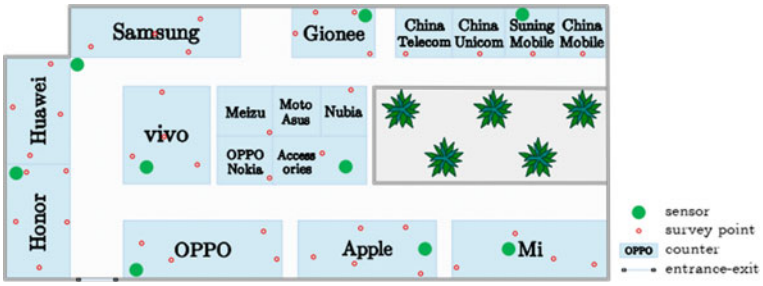


Fig. 8.12 Floor plan of the mobile phone store

Sensor ID	RSSI	Detection time
S1	65	2018/10/01 08:52:03
S1	69	2018/10/01 08:52:25
S2	72	2018/10/01 08:52:09
S2	73	2018/10/01 08:52:30
S3	75	2018/10/01 08:52:05
S3	78	2018/10/01 08:52:26
S4	80	2018/10/01 08:52:00
S4	75	2018/10/01 08:52:18
S5	62	2018/10/01 08:52:03
S5	63	2018/10/01 08:52:23
S6	67	2018/10/01 08:52:01
S6	60	2018/10/01 08:52:21
S7	48	2018/10/01 08:52:07
S7	55	2018/10/01 08:52:26
S8	62	2018/10/01 08:52:06
S8	66	2018/10/01 08:52:24
S9	73	2018/10/01 08:52:00
S9	69	2018/10/01 08:52:23

The asynchronous sensor data were arranged into a sequence X successively according to the detection time, and the results were as follows:

$$\begin{aligned}\chi = \{&\langle\langle S4, 80\rangle, 2018/10/01\ \ 08{:}52{:}00\rangle\\ &\langle\langle S9, 73\rangle, 2018/10/01\ \ 08{:}52{:}00\rangle\\ &\langle\langle S6, 67\rangle, 2018/10/01\ \ 08{:}52{:}01\rangle\\ &\langle\langle S1, 65\rangle, 2018/10/01\ \ 08{:}52{:}03\rangle\\ &\langle\langle S5, 62\rangle, 2018/10/01\ \ 08{:}52{:}03\rangle\\ &\langle\langle S3, 75\rangle, 2018/10/01\ \ 08{:}52{:}05\rangle\\ &\langle\langle S8, 62\rangle, 2018/10/01\ \ 08{:}52{:}06\rangle\\ &\langle\langle S7, 48\rangle, 2018/10/01\ \ 08{:}52{:}07\rangle\\ &\langle\langle S2, 72\rangle, 2018/10/01\ \ 08{:}52{:}09\rangle\\ &\langle\langle S4, 75\rangle, 2018/10/01\ \ 08{:}52{:}18\rangle\\ &\langle\langle S6, 60\rangle, 2018/10/01\ \ 08{:}52{:}21\rangle\\ &\langle\langle S5, 63\rangle, 2018/10/01\ \ 08{:}52{:}23\rangle\\ &\langle\langle S9, 69\rangle, 2018/10/01\ \ 08{:}52{:}23\rangle\\ &\langle\langle S8, 66\rangle, 2018/10/01\ \ 08{:}52{:}24\rangle\\ &\langle\langle S1, 69\rangle, 2018/10/01\ \ 08{:}52{:}25\rangle\end{aligned}$$

$$\langle\langle S3, 78\rangle, 2018/10/01\ \ 08{:}52{:}26\rangle$$
$$\langle\langle S7, 55\rangle, 2018/10/01\ \ 08{:}52{:}26\rangle$$
$$\langle\langle S2, 73\rangle, 2018/10/01\ \ 08{:}52{:}30\rangle\}$$

Given time point t = 2018/10/01 08:52:30 and time interval $\Delta t = 25$ s, the records of detection time within the interval $(t - \Delta t, t]$ constitute sub-sequence χ^t, and the results are as follows:

$$\begin{aligned}\chi^t = \{&\langle\langle S8, 62\rangle, 2018/10/01\ \ 08{:}52{:}06\rangle\\ &\langle\langle S7, 48\rangle, 2018/10/01\ \ 08{:}52{:}07\rangle\\ &\langle\langle S2, 72\rangle, 2018/10/01\ \ 08{:}52{:}09\rangle\\ &\langle\langle S4, 75\rangle, 2018/10/01\ \ 08{:}52{:}18\rangle\\ &\langle\langle S6, 60\rangle, 2018/10/01\ \ 08{:}52{:}21\rangle\\ &\langle\langle S5, 63\rangle, 2018/10/01\ \ 08{:}52{:}23\rangle\\ &\langle\langle S9, 69\rangle, 2018/10/01\ \ 08{:}52{:}23\rangle\\ &\langle\langle S8, 66\rangle, 2018/10/01\ \ 08{:}52{:}24\rangle\\ &\langle\langle S1, 69\rangle, 2018/10/01\ \ 08{:}52{:}25\rangle\\ &\langle\langle S3, 78\rangle, 2018/10/01\ \ 08{:}52{:}26\rangle\\ &\langle\langle S7, 55\rangle, 2018/10/01\ \ 08{:}52{:}26\rangle\\ &\langle\langle S2, 73\rangle, 2018/10/01\ \ 08{:}52{:}30\rangle\}\end{aligned}$$

Calculate the weight of each record in the sub sequence according to the formula in Step 4, the result is as follows:

t_i	$w(t_i,t)$
2018/10/01 08:52:06	0.51
2018/10/01 08:52:07	0.52
2018/10/01 08:52:09	0.54
2018/10/01 08:52:18	0.68
2018/10/01 08:52:21	0.74
2018/10/01 08:52:23	0.78
2018/10/01 08:52:24	0.81
2018/10/01 08:52:25	0.83
2018/10/01 08:52:26	0.86
2018/10/01 08:52:30	1

Each component of the user's RSSI vector is calculated according to the formula in Step four, and the result is $\vec{o}^t$ = [69, 72.65, 80, 75, 63, 60, 52.36, 64.45, 69].

According to the formula in Step 5, for each fingerprint RSSI vector, calculate the similarity between the user RSSI vector and the fingerprint RSSI vector, the results are as follows:

	$\text{Sim}(\vec{f}_i \mid \vec{o}^t)$		$\text{Sim}(\vec{f}_i \mid \vec{o}^t)$		$\text{Sim}(\vec{f}_i \mid \vec{o}^t)$		$\text{Sim}(\vec{f}_i \mid \vec{o}^t)$
$\vec{f}_1$	0.097	$\vec{f}_{11}$	0.013	$\vec{f}_{21}$	0.220	$\vec{f}_{31}$	0.261
$\vec{f}_2$	0.041	$\vec{f}_{12}$	0.049	$\vec{f}_{22}$	0.154	$\vec{f}_{32}$	0.285
$\vec{f}_3$	0.079	$\vec{f}_{13}$	0.002	$\vec{f}_{23}$	0.131	$\vec{f}_{33}$	0.095
$\vec{f}_4$	0.010	$\vec{f}_{14}$	0.021	$\vec{f}_{24}$	0.190	$\vec{f}_{34}$	0.069
$\vec{f}_5$	0.010	$\vec{f}_{15}$	0.004	$\vec{f}_{25}$	0.202	$\vec{f}_{35}$	0.032
$\vec{f}_6$	0.002	$\vec{f}_{16}$	0.006	$\vec{f}_{26}$	0.657	$\vec{f}_{36}$	0.043
$\vec{f}_7$	0.005	$\vec{f}_{17}$	0.003	$\vec{f}_{27}$	0.871	$\vec{f}_{37}$	0.015
$\vec{f}_8$	0.001	$\vec{f}_{18}$	0.064	$\vec{f}_{28}$	0.635		
$\vec{f}_9$	0.005	$\vec{f}_{19}$	0.092	$\vec{f}_{29}$	0.689		
$\vec{f}_{10}$	0.003	$\vec{f}_{20}$	0.058	$\vec{f}_{30}$	0.361		

According to the formula in Step 6, for each small region, calculate the similarity between the user's RSSI vector and the region, and the results are as follows:

k	$\text{Sim}(\hat{r} = k \mid \vec{o}^t)$	k	$\text{Sim}(\hat{r} = k \mid \vec{o}^t)$	k	$\text{Sim}(\hat{r} = k \mid \vec{o}^t)$
1	0.042	5	0.005	9	0.225
2	0.003	6	0.093	10	0.034
3	0.010	7	0.167		
4	0.006	8	0.707		

8.8 Graph K-means Algorithm Based on Leader Recognition, Dynamic Game and Viewpoint Evolution

This chapter introduces a graph k-means algorithm based on leader recognition, dynamic game and opinion evolution. Graph clustering task is a basic task in the field of graph data mining. However, the formation and evolution mechanism of the groups with different opinions hidden in graph data (which can also be regarded as a community structure) is still poorly studied. Therefore, it can be further studied by completing this task.

8.8.1 Study Scenarios, Motivations, and Meanings

With the proliferation of online social media, people's behavior is profoundly influenced by the interactions of online users and the structural networks that bind them together. Most Internet users have opinions on various topics ranging from public politics to daily life. These different points of view will be the result of careful thought, formed through interaction with others who hold opinions on particular issues.

Research on public opinion dynamics and community detection in social media networks has a wide range of application prospects, including public opinion prediction in political elections, advertising and public opinion prediction in social media. In the past two decades, although a lot of efforts have been made in these two fields, there are still some problems to be solved, which can be summarized as follows:

The vast majority of community detection methods in the literature focus on the topology of graphs, whereas real-world entities can be associated with multiple attributes (here they are called opinion vectors). How to integrate topology information and attribute information in real network effectively for community detection is still a problem to be solved. Although some multi-source information fusion methods have been proposed, it is difficult to determine the weight of structure and attribute in advance in the process of graph clustering. At the same time, the selection of prior distributions in statistical models requires a great deal of expertise.

The implementation mechanism of most community detection schemes often relies on greedy strategy, which is to design a target function up front and optimize it. However, real world opinion communities are often formed naturally and systematically from the bottom up, and in-depth understanding of the formation and dynamics of real opinion communities with huge individual diversity is still missing.

The existing viewpoint dynamic model only focuses on the binary choice of viewpoint, which is inconsistent with the actual observation. In addition, they tend to ignore the influence of individual intrinsic attributes, such as social influence, trustworthiness of interactions, and autoimmunity. In addition, the global community structure also affects the opinion dynamics, since each individual may trust interactive users in the same opinion community more than people in other opinion communities.

To address the above challenges, this section introduces a new and powerful graph k-means framework (hereinafter referred to as GK-means) that defines opinion community detection in social media networks as a multi-objective optimization problem in discrete time dynamic systems. Each discrete time period of GK-means consists of three coupling stages. In the first stage, a fast heuristic method is proposed to identify opinion leaders with high local reputation. In the second stage, the output of the first stage is regarded as the initial community structure of the t stage, and a new dynamic game model is used to solve the local Pareto optimal of the multi-objective optimization problem. Finally, in the third stage, the output of the second stage is regarded as the local Pareto optimal group structure of the t stage, and the opinion matrix of the next stage is obtained by using the opinion dynamic model. In conclusion, the GK-means proposed in this section can not only identify a series of

local Pareto optimal community structures, but also provide insights into the evolution process of opinion vectors and opinion leaders in real social media networks. The three theoretical contributions of this section are summarized below:

- We model the social media network as a physical system where all participants will interact. A broad definition of individual reputation is given, which can be used to detect potential opinion leaders. The model also has good flexibility and can be adjusted by specifying different forms of topological similarity and opinion proximity functions.
- Community detection in social media networks naturally creates a dynamic game in which all users and communities participate. By carefully defining the policy mapping function and utility function relevant to each user, we show that the local Pareto optimal community structure can be found by classical K-means at the end of some continuous optimization path.
- The generalization of the belief function of the classical bounded confidence model is studied so that the new model can deal with multi-dimensional continuous opinion space. It is further proved that any belief function suitable for opinion dynamics (that is, guaranteeing that the opinion vector of each user converges to a relatively stable state in a finite number of iterations) can be derived from continuously differentiable non-negative convex functions.

8.8.2 Basic Knowledge and Problem Definition

Definition 1 A Social Media Network (SMN) can be defined as a quad: $\mathcal{G} = \langle t, \mathcal{A}, \mathcal{O}^t, \mathbf{x}^t \rangle$, where $t = 0, 1, 2, \ldots$ is the periodic index of a discrete temporal dynamic system. $\mathcal{A} = \left[A_{ij}\right] \in \mathbb{R}^{n\times n}$ is an adjacency matrix of $n \times n$, and $\mathcal{O}^t = \left[o_{ei}^t\right] \in \mathbb{R}^{d\times n}$ is an opinion matrix of $d \times n$ in period t, $\mathbf{x}^t = \left\{x_1^t, x_2^t, \ldots, x_n^t\right\} \in \mathbf{X} = \mathcal{X}^n$.

Question 1 Opinion Community Detection (OPD) based on multi-objective optimization.

$$\min_{\mathbf{x}^t\in\mathbf{X}} \mathcal{Q}\left(\mathbf{x}^t\right) = \left(\mathcal{Q}^1\left(\mathbf{x}^t\right), \mathcal{Q}^2\left(\mathbf{x}^t\right)\right)^T$$

$$\mathcal{Q}^1\left(\mathbf{x}^t\right) = \sum_{p=1}^{K} \sum_{i\in\mathcal{C}_p^t(\mathbf{x}^t)} \varpi\left(o_i^t, c_p^t\left(\mathbf{x}^t\right)\right)$$

$$\mathcal{Q}^2\left(\mathbf{x}^t\right) = -\frac{1}{2m}\sum_{i=1}^{n}\sum_{j\neq i}^{n}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta\left(x_i^t, x_j^t\right)$$

$$\varpi(\mathbf{y}, \mathbf{z}) = \varphi(\mathbf{y}) - \varphi(\mathbf{z}) - (\mathbf{y} - \mathbf{z}) \otimes \nabla\varphi(\mathbf{z}), \forall \mathbf{y}, \mathbf{z} \in \mathbb{R}^d$$

The task now becomes opinion community detection based on multi-objective optimization.

8.8.3 *Specific Framework*

In SMN, there are four assumptions:

- A1: The adjacency matrix $\mathcal{A}$ remains stable, while the opinion matrix $\mathcal{O}^t$ changes with time;
- A2: Each period t in SMN can be further sustained when unsteady, according to $\tau = 0, 1, 2, \ldots$;
- A3: Each user can only join one community;
- A4: Opinion matrix $\mathcal{O}^t$ and community structure $\mathbf{x}^t$ will influence each other.

This introduces a new, powerful Figure K-means framework that consists of three parts, as shown in Fig. 8.13. The first stage uses a quick heuristic to identify those opinion leaders who have a high local reputation. The second stage is the output of stage one which can be seen as the initial community structure in period t, where each opinion leader stays in his own dominant community and other users do not belong to any community. In the second stage, the novel dynamic game model is carefully designed to find local Pareto, that is, the multi-objective optimization problem we are concerned with. Finally, the output of stage 2 is taken from stage 3, that is, the local Pareto optimal community adopts the robust opinion dynamic structure model in stage t, and the opinion matrix $\mathcal{O}^{t+1}$ of the next stage is obtained. GK-means repeats the above iterative process until the generated opinion matrix converges to a relatively stable state.

Remarks Different from existing AGC methods, the GK-means proposed in this section defines community detection in SMN as a multi-objective optimization problem, and attempts to use a dynamic game model to find the local Pareto optimal community structure in each discrete time period. GK-means is almost a non-parametric graph clustering method without the need for pre-determined structure and attribute weights for process graph clustering. It also does not need to estimate

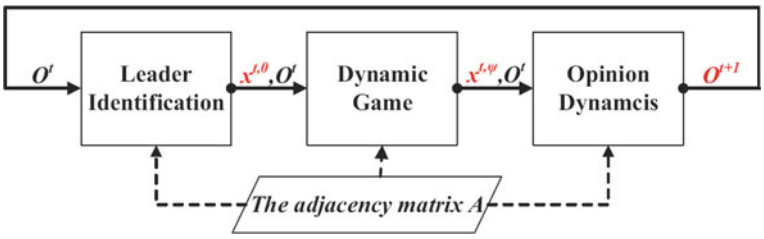


Fig. 8.13 GK-means algorithm framework

any prior probability distribution of the property graph generated model. With the help of the proposed dynamic model, GK-means can process both ordinary graphs and attribute graphs. In addition, subsequent experiments show that GK-means has better scalability compared with the subspace-based AGC method. To sum up, the GK-means framework proposed in this section can not only deeply understand the evolution mechanism of community structure, but also simulate the dynamics of opinion matrix and opinion leaders in the real SMS network.

In the real world, SMNS have different levels of structure at different scales. Communities usually form around a few opinion leaders who have a high reputation locally. When looking at a particular community, one can also observe a clear hierarchy, with opinion leaders at the top. Accordingly, the area where the opinion leader has a high local reputation can be regarded as his/her dominant community. Since hierarchy is a natural result of reputation dissemination and community formation, we believe that identifying opinion leaders plays an important role in revealing the underlying structure of the wechat community. A simple way to locate these opinion leaders is to calculate user centrality. There are many ways to measure centrality in the field of statistical physics, including degree centrality, intermediate centrality, proximity centrality and so on. The main disadvantage of these measures is that they only focus on the connection and ignore the user opinion vector.

Here, we view the SMN of each cycle as a physical system in which all users can interact and interact with each other. We assume that the influence will cause any two users to be directly attracted to each other by the force. At the same time, we have the following three position hypotheses: (1) Opinion leaders are expected to have higher local centrality; (2) The more similar two adjacent groups of users are, the more attractive they are to each other; (3) The number of gravitational users between any two entities will decrease rapidly with the increase of the number of users. Based on the above three assumptions, the reputation of each user during t can be defined as:

$$R_i^t = \sum_{j \in N_i} g_{ij}^t = \sum_{j \in N_i} \lambda_{ij} \exp^{-\varpi\left(o_i^t, o_j^t\right)}$$

where $g_{ij}^t = \lambda_{ij} \exp^{-\varpi\left(o_i^t, o_j^t\right)}$ is the attraction between j and i at time t, and $\varpi(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_0^+$ is the Bregman divergence. Therefore, if a user has a high local reputation, then he is more likely to become an opinion leader. Stage One uses a heuristic process similar to SCD to find a group of opinion leaders. Specifically, the reputation of each user is calculated in parallel. All users are then sorted by decreasing reputation value. All users in the sorted list are then accessed iteratively: if a user has not been accessed before, he is considered the new opinion leader; We then mark the user and all of its neighbors as visited. The process ends when all users have been accessed, or when K opinion leaders have been identified. Therefore, users associated with the non-zero community label will be regarded as opinion leaders during t. We notice that in the first stage the value of K is self-tuning. For example,

in a practical application, if the real number of the community is unknown, the first stage can even autonomously determine the best K by setting the input K = n.

In real life, the formation of wechat public opinion community is a bottom-up, natural and systematic process. It inspired us to model interactions between individuals in SMNS as dynamic games, where the decisions of one user can influence the decisions of others. In this section, to solve problem 1, the idea of dynamic game theory will be used to find the local Pareto optimal population structure in each discrete time period. Suppose that in the SMN of period t, there are n users, each of whom is associated with a view vector. We assume that each community can be associated with a set of dependent cohesion indicators about its internal connections and/or opinion vector users. At the beginning of each phase (for example, a week), all allowed groups set certain access thresholds to keep out those "bad" users in order to maintain their good cohesion metrics. After that, each user is free to join his or her accessible community throughout the scope (for example, from Monday to Sunday of that week), and we further assume that each user would prefer to join a group that is closest to it. The elusive question, then, is under what conditions the above iterative process converges to a stable state. Therefore, the constraint function is designed:

$$\mathcal{F}_{\mathrm{i}}^{t,\tau} = \{p | p \in \mathcal{X} \wedge g_i\left(p, x_i^{t,\tau}, x_{-i}^{t,\tau}\right) \leq 0\}$$

And utility function:

$$u_i^{t,\tau}\left(\mathbf{s}^{t,\tau}\right) = u_i^{t,\tau}\left(s_i^{t,\tau}, \mathbf{s}_{-i}^{t,\tau}\right)_{s_i^{t,\tau}=p} = \varpi\left(o_i^t, c_p^{t,\tau}\left(p, \mathbf{s}_{-i}^{t,\tau}\right)\right)$$

where $\forall s_i^{t,\tau} \in \mathcal{F}_i^{t,\tau}$, $\forall s_{-i}^{t,\tau} \in \mathbf{F}_{-i}^{t,\tau}$.

In addition, we found that in real life, a person's opinion is influenced by the opinion of his neighbors; In addition, if a pair of users has a relatively high LHN index, they are more likely to influence each other's opinions. In other words, a vector of ideas develops dynamically over time. Recent studies on opinion dynamics show that opinion coexistence in social networks often shows local effects. This means that community structure can also have an impact on opinion dynamics, since everyone may trust interacting users in the same community more than people in other communities. In this section, we propose a robust opinion dynamic model, which can be used to simulate the evolution of the opinion matrix in real SMS networks. In the proposed GK-means framework, the opinions of each user i at the time period t are dynamically determined by the following function:

$$o_i^{t+1} = \hat{\alpha}_i^t o_i^t + \left(1 - \hat{\alpha}_i^t\right) \sum_{j \in N_i^t} w_{ij}^t o_j^t,$$

where

$$N_i^t = \left\{j | j \in N_i \wedge \varpi\left(o_i^t, o_j^t\right) \leq \beta \rho^t\right\}, \beta > 0, \rho \in (0, 1),$$

$$\hat{\alpha}_i^t = \begin{cases} rand(1) & \sum_{j \in N_i^t} \lambda_{ij} > 0 \\ 1 & \sum_{j \in N_i^t} \lambda_{ij} = 0 \end{cases},$$

$$\forall j \in N_i^t, w_{ij}^t = \frac{\lambda_{ij} \exp^{\delta\left(x_i^{t,\psi}, x_j^{t,\psi}\right)}}{\sum_{l \in N_i^t} \lambda_{il} \exp^{\delta\left(x_i^{t,\psi}, x_j^{t,\psi}\right)}}$$

First, each user updates its trusted neighbor set; Each user then calculates his/her normalized belief values among the neighboring set of users; Finally, each user updates their opinion vector.

8.8.4 Experiment

Nine real benchmark networks were used in the experiment, including Karate, PolBK, FaceBK, Twttr, Gplus, Dblp, Amazon, YouTube and LiveJ, where PolBK, FaceBK, Twttr and Gplus were property graphs. Each node describes a D-dimensional vector with a value of 0/1. Some basic statistical data of the experimental data set are shown in Table 8.6, where K*, O*, C* and acc respectively represent community number, overlap rate, hand-marked real community coverage rate and average clustering coefficient. In all data sets except Karate and PolBK, (1) Each node can belong to multiple communities with uneven overlap rate, that is, in some data sets, one node may belong to more communities than other nodes; (2) Most benchmark networks are only partially marked, resulting in varying coverage.

Ten graph clustering algorithms are selected as the comparison base. Louvain is a multi-level clustering algorithm based on modular optimization. Infomap combines

Table 8.6 Data sets

Datasets	*n*	*m*	*d*	*K**	*O**	*C**	acc
karate	34	78	–	2	1.00	1.00	0.59
PolBK	105	441	3	3	1.00	1.00	0.49
FaceBK	4k	84k	89	193	1.46	1.00	0.61
Twttr	76k	1.2m	23k	3k	2.22	0.29	0.57
Gplus	102k	12.1m	610	438	2.69	0.23	0.49
Ddlp	317k	1.0m	–	13k	3.21	0.63	0.63
Amazon	335k	926k	–	75k	7.13	0.57	0.40
YTube	1.1m	3.0k	–	8k	2.40	0.04	0.08
LiveJ	4.0m	34.7m	–	287k	5.87	0.27	0.28

random walk and weighted modular optimization; Walktrap is a hierarchical clustering method based on random walk. Metis is a classical clustering method based on spectral cutting model. Cluto first divides the graph into subgroups, and then combines these subgroups repeatedly to get the final cluster. SCD partitions the graph by maximizing weighted community clustering. SA-Cluster is a graph clustering algorithm which can adaptively adjust the contribution degree of structural similarity and attribute similarity. CESNA is a high-performance overlapping community detection method, which is based on the network generation model with node attributes. EDCAR uses the established greedy stochastic adaptive search principle to approximate the optimal clustering solution. K-means++ extends the classical K-Means algorithm by selecting the initial clustering centroid according to the D2 metric.

Now, using different Bregman divergence, we illustrate the convergence of GK-means by observing the trend of the target vector during each iteration in Stage 2. Two property graphs were selected in the experiment—FaceBK and Twttr. For each Bregman divergence, the GK-means algorithm is repeated 100 times, randomly generating the initial community structure each time. Figure 8.14 shows the relative convergence rate of the k-means objective function values for this trend. As shown in Fig. 8.14, all of the RCR-Kmeans curves have a very similar downward trend, that is, the values continue to decline regardless of the data set, the initial community structure, or the use of Bregman divergence. As you can see from the figure, all modularity curves climb along the corresponding COP. This shows that both of the objective function values can be optimized continuously at the same time using the proposed dynamic model. Furthermore, it can be observed that such an optimization process results in a relatively stable structure of the community after about 20 iterations. The resulting solution is the local Pareto optimal problem 1, in which the two objective functions can be optimal because there is no neighborhood solution ψ. It can also be seen from Fig. 8.14 that the convergence rate of GK-means is quite satisfactory KL-divergence. For example, GK-KL can usually achieve the maximum reduction of the target function value within 10 iterations.

In this experiment, we further compare the clustering quality. For the different methods of all 9 data sets, Fig. 8.15 summarizes the comparison results. As the size of the data set increases, some algorithms are memory constrained or take longer than 24 h to compute, which results in missing values in the corresponding subgraph. GK-means can process all test data sets. The other Bregman divergence, by contrast, means using the squared Euclidean distance to produce more accurate clustering results. Although GK-KL converges faster at the second stage, it is slightly lower than GK-SE and GK-CD AvgF1 and NMI. Compared to other baseline tools, GK-means achieves better clustering quality in all attribute graphs; For ordinary graphs without node attributes, GK-means is slightly lower than SCD, but performs better than others. The idea behind Louvain, Infomap, and Walktrap is to optimize a single global objective function, known as modularity or weighted modularity. However, modular optimization does not have a resolution limitation, that is, it does not solve the problem of small communities. Metis and Cluto are two representative hierarchical clustering tools. Both methods work well with small data sets, such as Karate

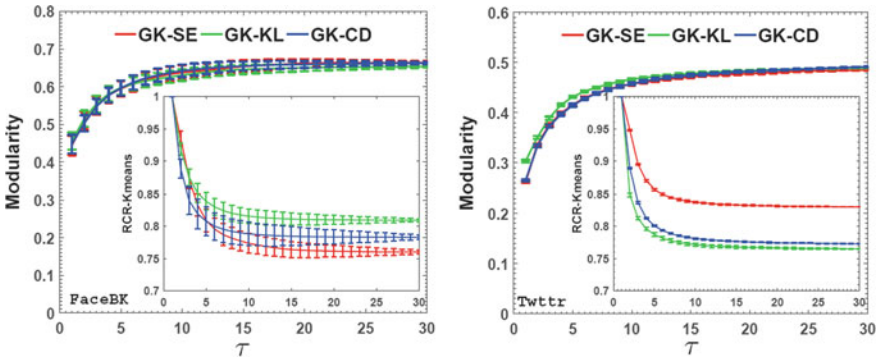


Fig. 8.14 Motion of RCR-Kmeans and modularity along associated cop on FaceBK and Twttr during t = 0

and PolBK, but lose their advantage on very large graphs. SCD detects community clusters by maximizing the weighted value WCC, where WCC is a recently proposed concept based on triangular analysis of community detection metrics. It does best on Dblp and Amazon; On the other hand, because of the property graph, SCD performs poorly and completely ignores the attributes of the node. Despite SACluster, CESNA and EDCAR consider both topology and attribute information and do not integrate the two data modes effectively. Their clustering quality is even inferior to that of some single-data based schema methods. K-means++ does not perform well on most data sets it completely ignores the topological information in it. Although the communities it detects include very similar attribute vectors, the node topology is relatively quiet and distant from each other in each detected community.

Next, the algorithm was tested for ablation. Table 8.7 summarizes the performance comparison between the above two variant algorithms and GK-means (where the best result s for the corresponding metric is highlighted in bold type), where the Bregman divergence function is based on the classical squared Euclidean distance function. Specifically, where

- GK-woLI: Random initialization is used to replace K leaders detected by a community in the algorithm stage.
- GK-woOD: Use the detected community structure directly.

GKwoLI and GK-means' results using the squared Euclidean distance are based on the statistical average of 30 replicates. In general, GK-SE performs best on all attribute graphs, followed by GK-woLI and GK-woOD. Compared with GK-woLI, the average clustering quality of GK-SE was increased by more than 24%, and the NMI was increased by more than 13%. Meanwhile, compared to GK-woOD, the GK-SE improved by more than 85% in average F1 and more than 30% in NMI. These results also demonstrate the effectiveness of two modules within GK-means, namely leader identification and opinion dynamics.

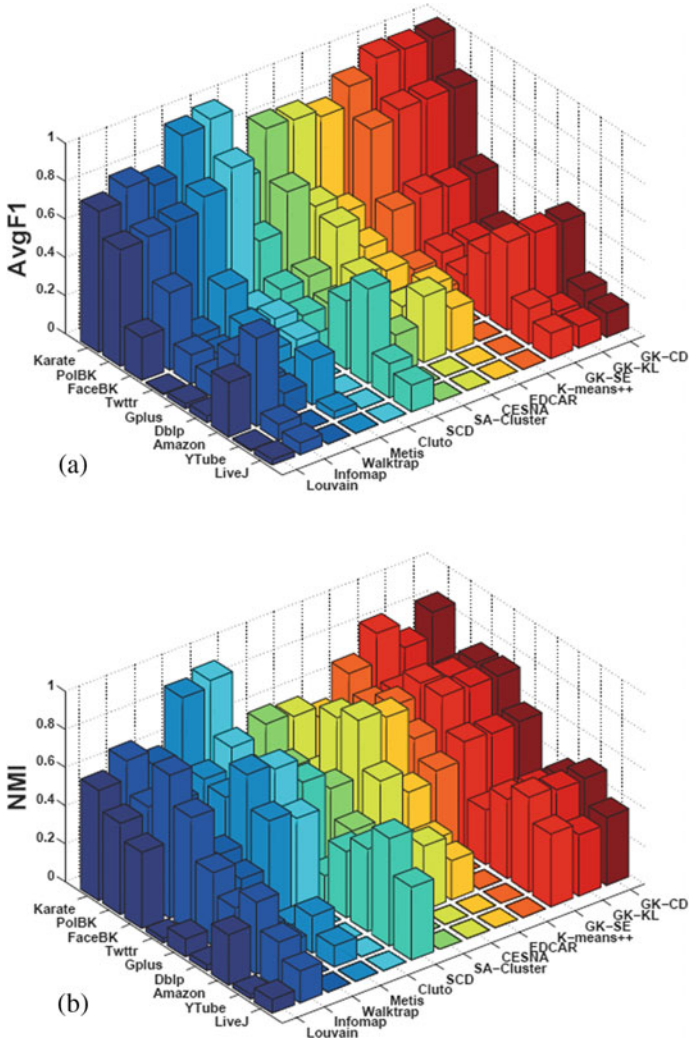


Fig. 8.15 **a** Comparison of AvgF1 values; **b** comparison of NMI values

Finally, GK-means is compared with 10 comparison algorithms in terms of running time. In order to make a fair comparison, the parallel versions of Infomap, SCD and CESNA, k-means++ are considered in this experiment. It is worth noting that GK-means also works well for parallelization. Each algorithm runs on four threads, which keeps the processor's four cores active. Since the difference between fast and slow algorithms is orders of magnitude, multithreading does not change the conclusion. Figure 8.16 shows the execution time of the different tools, with missing flags indicating that the tools cannot process a given data set within 24 h as the scale of the figure changes. As can be seen from the figure, four tools, Louvain, Infomap,

Table 8.7 Comparison of clustering quality of GK-means and its two variants on three attribute graphs

Metric	Dataset	GK-woLI	GK-woOD	GK-SE
AvgF1	FaceBK	0.362	0.225	**0.408**
	Twttr	0.188	0.134	**0.228**
	Gplus	0.137	0.104	**0.215**
	Average	0.229	0.154	**0.284**
NMI	FaceBK	0.600	0.516	**0.697**
	Twttr	0.724	0.627	**0.776**
	Gplus	0.556	0.480	**0.644**
	Average	0.627	0.541	**0.706**

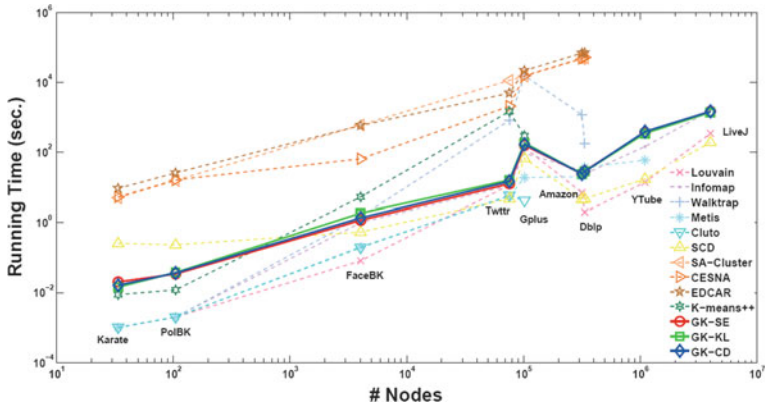


Fig. 8.16 Comparison of running times of different algorithms

SCD, and GK-means, can process all the data sets. For larger pure topologies, Louvain is the fastest method, followed by SCD, Infomap, and GK-means. The three selected attribute graph clustering tools, SA-Cluster, CESNA, and EDCAR, scale poorly and GK-means is three orders of magnitude faster than them. As K-means++ is based on the classical K-means++ algorithm, its time complexity depends on the number of clusters, so it cannot process data sets with large K. Broadly speaking, GK-means is not as scalable as Louvain and SCD, but is comparable to or even better than the other eight comparison algorithms.

8.8.5 Conclusion

This section introduces a new and powerful graph K-means framework, which can effectively integrate topological information and attribute information in social

networks for community detection. Firstly, the broad definition of personal reputation is given, and it is used to detect potential opinion leaders in wechat network. Then, we carefully study a new dynamic game model and prove that the local Pareto optimal community structure can be found by the classical K-means two-stage iteration at the end of a continuous optimization path. Finally, it is proved that any belief function suitable for opinion dynamics can be derived from continuously differentiable non-negative convex functions, and a robust opinion dynamic model is proposed to simulate the evolution of opinion matrix. A large number of experimental results verify the performance of the proposed Figure K-means framework.

Chapter 9
Application of Business Big Data Management and Decision Making

9.1 Malicious User Fraud Detection

The high socialization and ubiquity of emerging e-commerce brings together huge user groups and potential business opportunities. A large number of malicious users gain economic benefits by generating and spreading false opinions and junk information. The analysis and detection of malicious user behavior has become a hot field in the interdisciplinary field of e-commerce. With the rapid development of mobile Internet, the gradual deepening of the concept of Web 2.0, the rapid expansion of e-commerce logistics and the flourishing of social networks, e-commerce is gradually moving into a new form, showing significant features such as mobility, virtuality, sociality, personalization and extreme data richness [1]. In this new e-commerce environment, the types, forms and quantities of information have been greatly enriched, which has promoted the development of various e-commerce models and businesses, but also brought more opportunities to other malicious users [2]. Fake user behavior, driven by profit and aimed at advertising and marketing, has become rampant.

In recent years, employers' websites that organize online mercenaries to write comments to promote product sales or ranking have emerged both at home and abroad, such as Bike.com, Mechanical Turk, etc. Employers publish tasks regularly, users participate in random, publish and spread false opinions and comments to get corresponding remuneration. In a rectification action of Taobao.com, thousands of credit merchants were published. Behind them were organized credit companies and platforms, which forged a large number of purchase records and favorable comments for the merchants who purchased their services by clearly marking prices, so as to improve the credit and level of the merchants. A media reporter revealed that an illegal online public relations agency of Dianping made huge profits by writing false reviews for its cooperative merchants on the website. It hired nearly 10 "Internet water soldiers" to open hundreds of fake accounts, each of whom was responsible for "filling water" and hyping comments every day. In addition, half of the merchants it investigated had been blackmailed by professional bad reviewers. As many as 60% of the merchants finally choose to lose money to eliminate the disaster, the cost

J. Cao, *E-Commerce Big Data Mining and Analytics*, Advanced Studies in E-Commerce,
https://doi.org/10.1007/978-981-99-3588-8_9

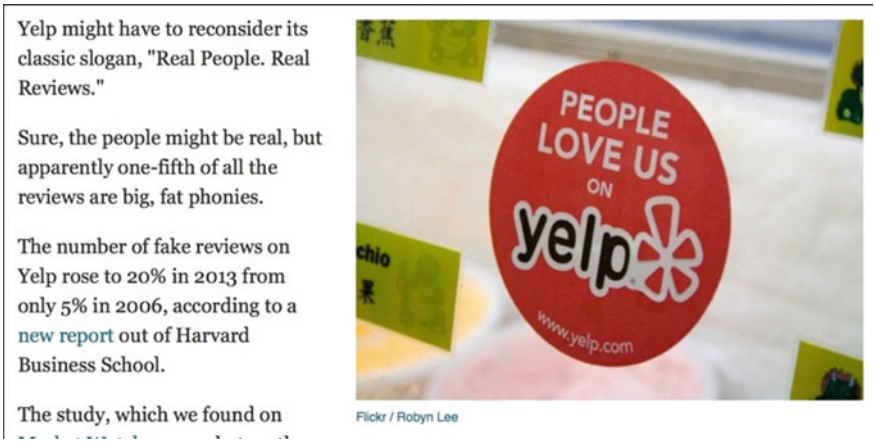


Fig. 9.1 Related reports of fake reviews on Yelp

from tens of yuan to tens of thousands of yuan; In 2002, when Amazon received a complaint, it found that a malicious user used a malicious attack to make the site recommend a book on sex alongside a Christian classic [3]. As shown in Fig. 9.1, Harvard Business School reported that 20% of Yelp reviews were fake in 2013. Such behavior seriously interferes with normal users' trading activities, reduces the credibility of the commercial platform, and causes serious harm to the order of e-commerce operation. Therefore, the research on Spammer Detection has become a hot topic in academia and industry, which has great theoretical and application value.

9.1.1 Malicious User Comment Detection

The development of electronic commerce makes the form and quantity of information in the Internet have been greatly enriched and developed. However, the open and interactive characteristics of e-commerce make it increasingly become the main target of attack and a way to spread malicious comments and false information by malicious users such as cyber watermen [4], who launch attacks through commodity comments and ratings. At present, for the detection of malicious comments in e-commerce, researchers have proposed a variety of models and methods based on different strategies [5–7]. Various types of e-commerce and applications have also been studied specifically, including Twitter, Amazon, Yelp, Foursquare, etc. In general, the existing detection methods for single malicious comments are divided into four types: content-based malicious user detection, graph-based malicious user detection, time-based malicious user detection and behavior-based malicious user detection.

Content based malicious comment user detection method: As malicious users publish fake comments or microblog and other text content, this is completely

different from user-item scoring model in recommendation system, which requires semantic analysis to complete the content analysis of malicious comments, so many research works are conducted based on content detection [8]. Mainstream detection methods are also based on the classification of feature attributes, but in terms of the construction of feature attributes, in addition to the behavioral characteristics of the publisher [9, 10], more natural language information of the content text is utilized, such as n-gram features, word frequency and speech features [11, 12] and word label features [13]. In terms of detection, McCord et al. trained classifiers by combining features of text content with features of users, and compared the performance differences of decision trees, naive Bayes and other classification models in detecting malicious users [14].

Graph based malicious comment user detection method: Most of the current graph-based anomaly detection is based solely on the graph, that is, there is no point and edge label in the graph. In terms of the definition of false points or outliers, Akoglu et al. considered near-clique structure and star structure as false points [15]. Muller et al. believed that most of the points closely linked to spurious points were spurious points [16]. Li et al. believed that the points distributed with the attributes of most points were outliers [17]. Gao et al. analyzed false points from the perspective of community mining, and believed that points not belonging to any community were community isolated points [18]. Davis et al. described the outliers from the perspective of graph and time, and believed that the outliers in topology and time were outliers. In the detection process, spectral methods (i.e., feature decomposition or singular value decomposition) are mainly used to gather similar points in the figure together [19–21].

Time-based malicious comment user detection method: A lot of work has been done to implement anomaly detection based on multivariate time series. Cheng et al. applied the kernel matrix alignment method to multivariable time series to capture the independent relation in time serie [22]. Li et al. mapped multidimensional time series data to a time series data cube to capture multidimensional space, and proposed to iteratively select the subspace of the original high-dimensional space to detect outliers [23]. Izakian et al. used time window to divide time series into a series of subsequences, and found the spatio-temporal structure of each time window based on Fuzzy C-Means algorithm, and characterized each cluster through outliers [24]. In social network anomaly detection, Beutel et al. constructed a graph from the user-page-time relationship, characterized abnormal behaviors through graph structure and edge constraints, and proposed optimization algorithms and parallel optimization algorithms to construct abnormal synchronization behaviors in social networks [25]. Literature [26] analyzed the distribution patterns of the time Windows of microblog release in social networks, including positive correlation, long tail, attack periodicity and bimodal distribution, and proposed the Rest-Sleep-and-Comment algorithm to detect outliers and Non-Human Behavior.

Malicious comment user detection method based on user behavior pattern: Since most malicious comment detection methods are based on the behavior patterns of malicious users and normal users, many research results are focused on malicious comment detection methods based on user behavior patterns. Aiming at the detection

problem of malicious commenters, Lim et al. proposed a method to model user fraud behavior based on different comment patterns of users. By defining abnormal patterns and measuring the degree of users in different abnormal patterns, they finally predicted a real value for users indicating that the users were malicious users [27]. Mukherjee et al. assumed that malicious users have different distribution of behavior patterns from normal users, and proposed the unsupervised Author Spamicity Model (ASM) model to model the potential distribution of users' behavior patterns under the framework of Bayes theory [28]. Lin et al. analyzed Sina Weibo data, determined the typical behavior pattern of malicious comment users, and established a malicious user detection model [29]. Xie et al. also found that compared with normal users, the behavior pattern of malicious comment users often changes, and proposed a sequential behavior pattern for detecting malicious associations. By establishing a multi-dimensional sequential pattern sequence, the detection problem of abnormal users is transformed into the detection problem of abnormal association pattern [10].

9.1.2 Recommended System Support Attack Detection

Since the concept of malicious user Attack (also known as Shilling Attack) was proposed in 2004 [30], scholars at home and abroad have proposed many detection algorithms to enhance the robustness and security of the recommendation system. Based on the assumption that the behavior characteristics of malicious users are different from those of most normal users, the most mainstream detection methods of malicious users define the behavior characteristics that are significantly different between malicious users and normal users. These characteristics are represented by feature space vector, and then a classifier is constructed to judge the categories of unmarked users [31–33]. Most of the research on malicious user detection of recommendation system is carried out around three basic problems: (1) definition of behavior characteristics; (2) feature modeling, evaluation and selection; (3) classifier learning. First of all, behavioral characteristics mainly depend on differences in users' scoring behaviors, and score statistical characteristics are defined based on this [34]. At present, a large number of achievements have focused on the scoring behavior of the recommendation system, and the reliability of their achievements has been proved by frequentist [35] and Bayesian statistical school [36]. In addition, some current models have applied the time feature [37] and the bimodal model [38] to the recommendation score. In terms of detection indexes of malicious users in the recommendation system, Chirita et al. proposed the average deviation RDMA between the average similarity DegSim and user model rating items and their average value by observing the distribution rules of various characteristic indexes of marked users [39]. Subsequently, Williams et al. from DePaul University in the United States systematically defined scoring Entropy entropy, average length variable LengthVar, TMF of user model's attention to target project, etc. [40, 41].

The recommendation system malicious user detection is essentially how to train a classifier (that is, normal user class and malicious user class) or sorting problem

(that is, to set a threshold as the dividing line between normal user and malicious user). From the use of prior knowledge, detection algorithms can be divided into three types: supervised learning, unsupervised learning and statistical learning. It is an intuitive idea for people to train detectors with known category users as reference when dealing with malicious attack detection. Its essence is to construct classifiers based on supervised learning. Most of these detection algorithms take feature indexes as classifier attributes. Chirita et al. proposed the first malicious attack detection algorithm by observing the distribution rule of various characteristic indexes of tagged users, and proposed for the first time the empirical algorithm of malicious attack detection combined with DegSim and RDMA to detect average and random attacks by using two indexes of average similarity and RDMA [39]. Subsequently, scholars such as Mobasher, Burke and Williams from DePaul University in the United States defined detection indicators systematically and did a lot of work in detecting malicious attacks based on decision trees [40, 41]. Williams' technical report summarized their work.

9.1.3 Credit Card Fraud Detection

With the development of modern information technology and globalization, credit card transactions become more and more. Meanwhile, credit card fraud is on the rise. According to the report of China Banking Association, the illegal cases of credit card fraud are increasing year by year, and the economic loss caused by credit card crimes has reached 10 billion yuan every year [42]. Credit card fraud detection has become an urgent problem. With billions of euros lost to credit card fraud worldwide every year, financial institutions urgently need a well-designed fraud detection system to prevent fraudulent transactions. In recent years, machine learning technology has been widely used in credit card fraud detection, and achieved good results. However, due to the high imbalance and concept drift of the credit card data set, the credit card fraud detection system needs to be improved continuously.

In fact, credit card fraud detection is essentially a binary classification problem. Most of the existing researches have designed different classifiers to realize credit card fraud detection on the basis of constructing feature engineering. Feature engineering mainly constructs features from two dimensions: cardholder user information and transaction record historical data. Cardholder user information mainly includes cardholder's card number, ID number, marriage, education background, occupation, annual income, etc. Transaction record historical data mainly includes transaction date, transaction amount, transaction type, historical repayment, etc. At present, the mainstream classification methods for credit card fraud detection can be divided into two categories: traditional machine learning algorithm and neural network method. Traditional machine learning methods include random forest [43–45], support vector machine [46] and boosting [47] algorithm, etc. Although these machine learning algorithms above can achieve good prediction or classification effects for small sample data sets, they cannot achieve ideal effects when encountering large amounts of

high-dimensional data. However, deep learning algorithms can better solve high-dimensional complex data, and algorithms based on deep learning can more accurately extract effective features from big data. To build a more perfect model. Deep learning algorithms commonly used for classification include convolutional neural network [48–50], deep neural network, etc. Among them, CNN is outstanding in feature selection in the face of high-dimensional data, so it is widely used in detection in various fields. For example, Liu et al. [51] proposed an improved fuzzy neural network algorithm based on the traditional fuzzy neural network with the help of the Grey Wolf algorithm, and applied it to the study of credit card default prediction. Hsu et al. [50] used recursive neural networks (RNN) as feature extractors and extracted dynamic features and static features to train the enhanced RNN model (RNn-RF) to predict credit card defaults. In addition, considering the existence of extreme class imbalance in credit card fraud detection data, there are two solutions: under-sampling and over-sampling. Under-sampling is the random selection of the same number of samples as the small sample from the large sample. For the unbalanced problem, underfitting will occur because the sample size is too small. Oversampling is using the use of a small sample to produce the same number of samples as SMOTE, there are two methods: Random oversampling and SMOTE oversampling. In credit card fraud detection of the imbalance data, SMOTE was used extensively to balance the SMOTE data by treating a small sample of the original data.

9.2 Online Purchase Decision Model

The core of the purchasing decision model in e-commerce model is the insight of customer behavior, that is to provide intelligent marketing plan service and personalized product recommendation according to customers' purchase intention and different interests and preferences. The low conversion rate of e-commerce model "browse-buy" determines that the insight and prediction of customer purchase intention become important issues affecting revenue. In other words, further pushing personalized services (such as recommending specific products) to customers with purchase intention is easier to achieve marketing objectives. Existing studies on customized promotion have refined the supermarket purchase decision model [52, 53] into three prediction stages: (i) Whether to buy; (ii) What to buy; (iii) How much to buy.

9.2.1 Purchase Prediction Model

With the rapid development of information technology and the popularization of the Internet, e-commerce has been widely used in the tourism industry and has become a new mode of tourism transaction in the era of e-commerce. At the same time, due to the gradual increase of information on the Internet, various types of

online tourism resources have also brought data disasters, making it particularly difficult for online users to find the products they need accurately and timely [54–56]. More importantly, facing a large number of online users, how to accurately find potential users of orders has become a difficult problem for e-commerce platforms to conduct precision marketing [57, 58]. Similarly, the tourism e-commerce platform also generates massive and rich data during its operation, such as log data in the website server, user and product information in the background database, and a large number of order and transaction data. These data contain a large amount of valuable information, aiming at these massive data, how to use big data analysis technology to build a purchase prediction model has become a hot topic.

9.2.2 Personalized Recommendation Model

At present, many mature recommendation algorithms have been widely used for traditional commodity recommendation. For example, Collaborative Filtering (CF) [59], Content-based recommendation [60], Singular Value Decomposition (SVD) [10] and Latent Factor and Matrix decomposition (LF) model [35]. Among them, collaborative filtering algorithm is by far the most famous and widely used recommendation algorithm. Typical applications include Amazon, the world's largest B2C e-commerce site; MovieLens, an experimental movie recommendation site for research purposes; StumbleUpon, a social web recommendation algorithm; and a mobile App that recommends popular news stories Platform Toutiao and social music clothing Last.fm, etc. The most basic collaborative filtering algorithm takes the "user-item" scoring matrix as the input, according to the User's score value of the Item, and then adopts the similarity principle to predict the user's score of the unknown item based on similar users or similar items. To be specific, Collaborative Filtering algorithm [59] is divided into User-based Collaborative Filtering (UCF) [11] and Item-based Collaborative Filtering (ICF) [60]. The idea of UCF is to find users who are similar to the current user, and then recommend items that these similar users like to the current user, while the basic idea of ICF is to recommend items that are similar to the user's current items directly (but the similarity between items is also determined by the "user item" relationship). Cryptic meaning and matrix decomposition model [35] is an important collaborative filtering method. This model provides the concept of hidden factor, which is the bridge between users and items. By iteratively adjusting the parameters in the model, the low-dimensional approximate matrix is learned, and the optimal recommendation model is finally established. Content-based recommendation model [60] refers to the recommendation of other items with similar attributes according to the items selected by users. This recommendation model is derived from information retrieval method. By analyzing the content features of items and the features of items interested in by users, the matching degree between users and target items is calculated, and the items with high matching degree are recommended to users. For example, recommending a travel package relies on information such as

the destination, price and number of days on the trip. Therefore, content-based recommendations require extensive domain knowledge, which often needs to be specified by domain experts [61, 62]. However, the collaborative filtering model only relies on the user's rating of the project to generate recommendations, which has nothing to do with specific project attributes. The above single-type recommendation algorithms (such as UCF, ICF and SVD) are often troubled by problems such as data sparse [63], project cold start [61] and user cold start [64–66]. In order to deal with the above deficiencies, more and more scholars pay attention to the research of Hybrid Recommendation Model algorithms [67]. One of the most important principles of hybrid recommendation algorithms is to avoid or make up for the weaknesses of their recommendation algorithms through the combination of models. For example, the hybrid recommendation algorithm [67], which combines content-based and collaborative filtering algorithms, is the most widely used in practice. The performance of collaborative filtering algorithm based on matrix decomposition can be greatly improved by adding some content information of users or projects. According to different application scenarios, content information such as purchase time [68], social relationship [69] and even project net profit [70] can be used to design hybrid recommendation algorithm.

Deep learning can obtain the deep unified representation of users and items by learning a deep nonlinear network structure, and has a powerful ability to learn the essential characteristics of data sets from samples. In recent years, deep learning has pushed the research and application of artificial intelligence to a new upsurge, and also brought new opportunities for the research of recommendation system. Based on collaborative filtering and matrix decomposition algorithms, a lot of research work integrates deep learning theory to carry out recommendation research based on deep learning. At first, Salakhutdinov et al. [71] proposed a method of using the depth hierarchy model for collaborative filtering algorithm for movie recommendation in the ICML meeting in 2007, which set a precedent for applying deep learning to the recommendation system. Since then, more and more scholars have applied the depth model to the recommendation system. By taking advantage of the effectiveness of deep learning to extract hidden features and relationships, some scholars have proposed a series of alternative solutions to solve the challenges of recommendation systems (such as accuracy, sparsity and cold start problems [72–75]). For example, Devoogh et al. [76] improved the neural network RNN model by converting the prediction problems of collaborative filtering and matrix decomposition into sequential prediction problems to improve the accuracy of short-term and long-term interest prediction. Sedhain et al. [77], with the help of the autoencoder, improved the accuracy by predicting the severely missing "user-item" scoring matrix. He et al. [73] proposed a neural network structure to simulate the potential characteristics of users and projects, and designed a universal collaborative filtering framework NCF based on neural networks. Xue et al. [78] propose to use multi-layer neural networks to learn potential user and project factors in matrix decomposition. However, the above methods do not make full use of the relevant content information of the user and the project, and in fact this neglected information is crucial to the recommendation system. For this reason, instead of using neural networks in the traditional matrix

decomposition framework, many studies use neural networks to learn the original feature representation of users [79–81]. For example, Wang et al. [72] proposed a hierarchical Bayesian model to solve the problems that only ID features or scoring features are used in the collaborative filtering algorithm, some information features of the algorithm are rarely used, and it is difficult to be effective on sparse matrix. Collaborative Deep Learning (CDL) makes use of deep representation learning of content information and collaborative filtering of scoring (feedback) matrix. Guo et al. [80] proposed a DeepFM algorithm. DeepFM effectively combined the advantages of Factorization Machines (FM) and Deep Neural Network (DNN) in feature learning. It can extract low-order combination features and high-order combination features at the same time, and share the same input and embedded vector, and get better training effect. Elkahky et al. [81] used the deep learning method to map users and items into a potential space, and extended it by introducing a multi-view deep learning model to learn from items and user characteristics in different fields. The representation of such features enables the model to learn relevant user behavior patterns efficiently. Effectively alleviates the user cold start problem.

9.2.3 Sales Forecasting Model

In recent years, our country electronic commerce develops rapidly, the competition between electric commerce is increasingly fierce, at the same time, electronic commerce has more dynamic and complex relationship with the traditional offline retail business environment, providing the brand-new development opportunity for each electric commerce at the same time also put forward the brand-new challenge. In the face of the dynamics and complexity of domestic and foreign e-commerce markets, sales forecast has become an urgent application problem for most e-commerce enterprises. Forecasting the sales volume of electronic goods can largely compress the inventory turnover cycle of relevant product manufacturers, reduce the squeeze of goods on the working capital of enterprises, and thus improve their own economic benefits [82].

At present, commodity sales forecasting is mainly divided into methods based on statistics, machine learning, data mining and deep learning [83]. Sales forecasting algorithm has been studied by many scholars at home and abroad for quite a long time. In recent years, with the development of online business model, the importance of e-commerce sales forecasting has been gradually discovered. Luo et al. adopted a brand new calculation model aiming at the characteristics of the double trend change of cigarette sales in time series and the multiple impact factors affecting cigarette sales. This model is mainly a calculation model to predict the sales trend of related products in the future period of time through the application of ARMA algorithm [84]. The model accuracy of this method is poor when predicting nonlinear relation.

Chang et al. proposed a forecasting model combining time series linear regression model and intelligent nonlinear regression method to improve the forecasting effect of ARMA model under stationary and non-stationary time series, aiming at the

inability to capture the non-stationary and nonlinear characteristics of commodity sales in the time series model [85]. Feng and Chen adopted a weighted combination method of ARIMA-XGBoost-LSTM to predict the results of the original data series more accurately in combination with the correlation algorithm [86]. Higher accuracy can be achieved by using the XGBoost model for prediction, but the complexity of XGBoost's pre-sort space is too high, which will consume too much memory, resulting in slow training process. Ma and Wang used LSTM model to learn the nonlinear dependence relationship in data and solve the long-term dependence problem. They predicted the sales volume of dishes, and the final prediction result was much higher than the traditional ARIMA model [87]. Li and Wu built LSTM model under the Pytorch framework to forecast the highest prices of Shanghai and Shenzhen indices and four specific domestic stocks, and found that in the process of predicting the stock results, the shorter the time, the more accurate the final forecast results, and the longer the time, the bigger the difference [88]. The traditional LSTM model has the problem of gradient disappearing when predicting time series problems. Li and Zhang combined BP algorithm and LSTM algorithm in their research on problems related to automobile sales prediction [89]. Aiming at the complexity of manual parameter adjustment in traditional feature extraction, Wang proposed an adaptive LSTM prediction model and improved the prediction accuracy through automatic feature extraction [90]. Jiang proposed a WaveNet-LSTM model aiming at the instability of hidden layer when LSTM is used as feature extraction. WaveNet network is used for feature extraction of time series, and then LSTM model is used as prediction module for output, which improves the accuracy of sales prediction for retailers [91]. The above research is aimed at the offline commodity sales forecast of entity merchants. This paper designs an improved prediction model with multiple layers of LSTM superimposed based on LSTM neural network on the basis of the existing prediction of ordinary commodity sales. Aiming at the characteristics of rapid collection, convenient data processing and huge data volume of e-commerce commodity sales, this paper makes more accurate forecast of e-commerce commodity sales.

9.3 Related Applications of Tourism E-Commerce

With the improvement of national income level and consumption concept, tourism consumption has become one of residents' daily life. According to the 44th Statistical Report on China's Internet Development, by June 2019, the number of online booking users in China's tourism industry had exceeded 400 million. With low pollution, high linkage, consumption promotion and structural adjustment, tourism has become an important part of the current economic transformation, and has risen to the level of national strategy. The Outline of the 13th Five-Year Plan for China's National Economic and Social Development mentions "tourism" in 19 places, making clear the economic significance of developing tourism and implementing supply-side reform. At present, China's tourism industry has completed the structural transformation

from niche market to mass market, from ticket economy to all-region tourism, and the tourism market is increasingly showing the characteristics of personalized demand, mobile consumption, destination IP, product differentiation and so on. In the digital context, with the deep integration of the Internet, big data, cloud computing, artificial intelligence, virtual reality technology and real economy. Major Online Travel agencies (OTAs), such as Tuniu Travel, Ctrip and Qunar, as well as major free travel Online travel platforms (OTP), such as Hornet's Nest, Tripadvisor and Qiongyou.com, have developed rapidly. It realizes the connection and sharing of tourism users and tourism destination information, promotes the integration of tourism service resources and full-factor collaborative services with related industries.

9.3.1 Point of Interest POI and Travel Package Recommendation

Personalized recommendation applications in the field of tourism are mainly divided into two types: POI recommendation and travel package recommendation. By treating POI as a common commodity, most researchers apply traditional collaborative filtering methods to POI recommendation, and extensively explore a large amount of situational information, such as geographical information [92], social relations [93], time information [94, 95] and user preference order [96]. Among them, geographic information is the most important. Users' travel is mostly concentrated around the living area and radiates, reducing the probability of visiting remote locations. Cheng et al. assumed that users check-in in several centers, used multi-source Gaussian distribution to model the check-in trajectory between the two places, and then integrated these information into matrix decomposition, and then proposed a point of interest recommendation algorithm. Zhang et al. believed that the influence of geographic information on the movement trajectory is personalized, instead of assuming a distribution for all users and using kernel density estimation (KDE) method to model the influence of geographic location [94]. Ye et al. eliminated the friend-based collaborative filtering method based on friends' common visit records to make POI recommendation [97]. Compared with POI recommendation, there are few studies related to travel package recommendation, and traditional recommendation methods are difficult to be directly applied to travel package recommendation. In view of the strong sparse tourism data and other features, most of the existing studies [98–102] first extract multidimensional features describing tourism products from heterogeneous data. For example, the implied theme, the relationship between scenic spots and regions, and the relationship between time and price extracted from the text are expected to strengthen the correlation between users and travel packages, so as to design some novel collaborative filtering or content-based recommendation methods, and finally effectively alleviate some problems faced by tourism data. For example, Liu et al. [96] tried to explore potential themes from the description text

of STA Travel travel package, and proposed a hybrid collaborative filtering recommendation method combining seasonal and price factors, which effectively solved the problem of collaborative filtering recommendation applied in extremely sparse travel data. In addition, considering the travel cost (i.e. money and time), Ge et al. [58, 59] conducted a special study on matrix decomposition in the potential factor model of cost perception.

9.3.2 Travel Itinerary Planning

Tourism products contain complex and diverse elements, small parameter changes will lead to completely different tourism products, such as scenic spot visit route and schedule, hotel and transportation options and other factors. Most consumers can easily find a variety of tourism products that roughly meet their needs on the OTA platform, but this fixed customization mode is difficult to meet the personalized needs of users. In order to obtain perfect new products, it is often necessary to fine-tune or integrate the components of these products (such as the addition, deletion or order of scenic spots, hotel stars, etc.). Therefore, the personalized customization of tourism products is in urgent demand. In fact, many online travel platforms are offering personalized product customization by telephone customer service. Personalized product customization has long been studied by scholars in the field of economic management [100–103]. Most of these researches are from the perspective of enterprises, that is, they study product customization and its pricing strategy in order to maximize benefits. From the perspective of technology, this project intends to carry out innovative exploration on the personalized customization and combination of tourism products based on the modeling of user behaviors and preferences. Lim et al.'s work [104, 105] involved the customization of tourist paths. For example, in literature [106], optimization objectives were formed based on user queries, preferences, and temporal and spatial constraints of paths, and new tourist paths were dynamically generated by Monte Carlo tree search algorithm. In addition, the research work in the field of service portfolio [107–110] also provided many inspirations for the research of product customization, including: the goal of product portfolio customization is to meet the hard requirements and optimize the soft requirements as much as possible, and there is often more than one combination of feedback output (often in the form of Skyline queries). Finally, different from manual customization, technology-driven automated product customization mainly completes communication through man–machine interaction, which requires more advanced and perfect service mode to accurately obtain user needs and preferences, and enhance user enthusiasm for participation through safe, friendly and easy-to-use operation experience.

At present, the research on the customization service model mainly focuses on the mass product customization [111, 112]. For example, Gilmore and Pine studied the ways of enterprises to provide mass customization services and proposed multiple paths to achieve mass customization [113]. Park et al. compared the influence of

two options presentation modes of "addition" and "subtraction" on user decision difficulty in customized services, and pointed out that in the subtraction customization mode, customers would perceive more value and be more satisfied with the customized results [113]. Wang et al. further considered the role of customer self-efficacy in product customization and analyzed the relationship between customer self-efficacy, option presentation and customization satisfaction through empirical research [114]. In spite of this, the research on the service model for online tourism product customization still stays at the level of simple service system design, lacking a comprehensive and systematic study on the service model and in-depth insight into its internal mechanism. In conclusion, personalized customization of products is a further extension of personalized recommendation. Both theoretical research, application system and service model design in subdivided fields are in the initial stage, showing openness, which provides researchers with good research opportunities and challenges.

9.4 Business Applications of Location-Based Services

9.4.1 APP Takeaway Food

The fast pace of urban life has led to a boom in the food delivery industry, with many restaurants near the city center staying open all night to deliver delicious food to customers after receiving delivery orders, called "deliverymen". As customers' patience is limited, the delivery boy needs to arrive at the restaurant as quickly as possible to pick up the food and deliver it to the customer's address, otherwise he risks being sued by the customer and losing his pay or even being fired. So how to shorten customer waiting time has become the central issue of take-out delivery. In addition, under the trend of consumption upgrading, people have increasingly higher requirements for delivery service, which also encourages the network delivery platform to continue to increase size to provide a more efficient service system. In order to develop a modern and systematic take-out delivery system, distribution nodes and distribution routes should not be ignored. However, the location of distribution centers and the number of distribution personnel are important issues for the construction planning of distribution centers, which not only directly affect the time of take-out delivery, but also affect the operation performance and future development of the take-out industry [115]. Therefore, it is of great practical significance to reasonably select the address of distribution center, increase the number of delivery personnel and shorten the average waiting time of customers, which is conducive to the rapid response of restaurants to customer demands, improve the service level of take-out industry and enhance customer satisfaction with the distribution link.

Due to the widespread existence of distribution problem in real life, scholars in various countries have carried out partial research on this problem, but there are still many problems worth discussing. Yang et al. [116] took the distribution route

optimization problem of a well-known express delivery enterprise in Hefei as the research object, analyzed the distribution problem of the enterprise, and established a mathematical model with minimum distribution network cost as the optimization objective. Fan et al. [117] made reasonable planning of the distribution routes inside and outside the campus of Shenyang University by using the genetic algorithm and the related theories of TSP problem and the improved adaptive genetic ant colony hybrid algorithm, which effectively shortened the length of the delivery route on campus and effectively improved the delivery efficiency of the deliverymen. Huang et al. [118] applied the theory of travel salesman problem, modeled the distribution route problem of outbound sales, established the distance matrix between customer rooms, and obtained the short distance between each two customer rooms and the shortest path back to the restaurant after delivery by using genetic algorithm and MATLAB programming. Zhai et al. [119] under the constraints of satisfying customer demand and time window, established a delivery route optimization model with the shortest delivery time as the goal, and obtained the optimal delivery route by using genetic algorithm. There are numerous such researches, but few of them take the location of distribution center and the number of distribution personnel as the optimization object.

9.4.2 Car-Hailing Route Planning

By the end of the second quarter of 2018, there were 1.51 billion mobile phone users in China. Such a large group of mobile phone users and the development of mobile communication technology have brought the vigorous development of peer-to-peer sharing transportation industry. As a typical example, today, the convenience of online car booking has made it an important means of transportation for the public to travel. According to the city operation report of Didi Platform in the second quarter of 2018, the number of users of the platform has exceeded 550 million, providing 30 million trips for users in more than 400 cities in China every day [120]. While bringing convenience to users, the emergence of online car hailing is likely to increase traffic congestion. Specifically, due to the unbalanced relationship between supply and demand in the ride-hailing market, ride-hailing drivers are usually unable to successfully pick up passengers in the area where they drop off passengers, so they need to drive empty cars to other areas to find customers, which increases traffic congestion and environmental pollution. Cramer and Krueger [121] surveyed 2000 ride-hailing drivers in five major U.S. cities (Boston, Los Angeles, New York, San Francisco, and Seattle) and found that the empty load rate was 45–57%. In a 2017 analysis of ride-hailing data in San Francisco, the empty load rate in San Francisco was 20%, and this no-load behavior resulted in an increase in vehicle mileage of 6.5–10% [122].

In view of this, it is important to consider the impact of ride-hailing behavior on the traffic distribution of the transportation network. Ban et al. [123] established variational inequalities to describe the operation path selection behavior of online

ride-hailing. In their model, all ride-hailing operations are assumed to be uniformly subject to the scheduling of the platform in order to optimize the total revenue of the platform. This assumption is quite different from the actual ride-hailing business model, which is usually operated on the principle of optimizing the personal income of ride-hailing drivers. Xu et al. [124] established a user equilibrium model that can capture no-load and passenger-seeking behaviors of online car-hailing through a series of inequality equations to reflect the real operating behaviors of online car-hailing, and described the impact of no-load behaviors of online car-hailing on road network traffic flow through this model. Domestic scholars have also done a lot of research on e-hailing, but mainly focused on the usage characteristics and selection preferences [125, 126], pricing [127, 128], parking behavior [129], etc. In this study, a simple variational inequality model is proposed to describe the influence of passenger demand on ride-hailing drivers' passenger-seeking strategies and routing behaviors, and then a road network equilibrium model considering ride-hailing operation behaviors is described. This variational inequality model can be used in traffic planning of related departments.

9.4.3 Restaurant, Hotel and Gas Station Recommendation Based on Location Service

With the rapid popularization of intelligent mobile devices, a large amount of data of Location-Based Social Networks (LBSNs) has accumulated. How to make use of these data has become a hot issue of common concern in academia and industry. In LBSNs, users can share the places they have visited, such as restaurants, celebrity attractions and shops, by checking in at POI (Point-of-Interest). The general task of recommending POIs is to recommend new and interesting POIs that users are interested in [130]. However, in the campus environment, student users' behaviors have obvious regularity. Therefore, when recommending POIs to users, we should not only recommend new and interesting POIs that students have not visited, but select POIs that students may be interested in from the whole data set. It is better to select POIs that students may be interested in from the whole data set for recommendation. The main goal is to obtain the top-N POI that students are likely to visit at a given time by mining the student body's check-in records and other available information.

Unlike traditional no-context recommendation systems, the interaction between the POI and the user requires the user to visit a real place in the real world. Therefore, spatio-temporal information (including longitude and latitude coordinates of locations and time factors, etc.) is the key factor affecting users' actual check-in behavior. For example, students usually go to the playground for walking and physical exercise in the afternoon or evening, and usually in the dormitory, library and study room area on weekends. In conclusion, temporal and spatial information is crucial to analyzing user behavior for POI recommendations. POI recommendation is of great value in urban planning, commercial advertising and service industries,

and many methods have been proposed to improve the quality of POI recommendation [131, 132]. However, how to accurately predict users' POI at a given time based on complex spatio-temporal information is still a challenging problem [133]. Many researches solve the problem of POI recommendation by adopting traditional methods (such as Matrix Factorization, MF, etc.). MF obtains the user and POI potential factors according to the user POI frequency matrix, which displays the number of user check-ins [134]. Due to the low density of POI check-in data in general check-in entertainment data sets (such as Foursquare, Yelp, etc.), MF-based POI recommendation has a data sparsity problem [135–137]. In order to solve this problem and improve the accuracy of POI recommendation, other contextual information, such as geography, time and category information, should be combined in the recommendation process [138–140]. Analysis of user behavior shows that geographical information has a greater impact on user preferences than other contexts [141, 142]. Therefore, several POI recommendation algorithms based on geographic information are proposed [143–145]. However, these algorithms only consider geographical information from the perspective of users. For example, geographic distance is the distance between the user's location and the POI.

References

1. Chen G, Wang J, Guo X, Xu X, Hu L, Liao X (2013) Behavior of emerging E-commerce participants. Tsinghua University Press
2. Mo Q, Yang K (2014) Research on network spammers identification. J Softw 25(7):1505–1526
3. Zhang F, Xu S (2008) Survey on security issues and technology of recommender systems. Appl Res Comput 25(3):656–659
4. Lee K, Caverlee J, Webb S (2010) The social honeypot project: protecting online communities from spammers. In: Proceedings of the 19th international conference on world wide web (WWW 2010). ACM, pp 1139–1140
5. Hu X, Tang J, Liu H (2014) Online social spammer detection. In: Twenty-eighth AAAI conference on artificial intelligence (AAAI 2014). AAAI Press, pp 59–65
6. Hu X, Tang J, Zhang Y et al (2013) Social spammer detection in microblogging. In: Proceedings of the twenty-third international joint conference on artificial intelligence (IJCAI 2013). AAAI Press, pp 2633–2639
7. Benevenuto F, Magno G, Rodrigues T et al (2010) Detecting spammers on twitter. In: Collaboration, electronic messaging, anti-abuse and spam conference (CEAS 2010), pp 6–12
8. Lecun Y, Boser B, Denker J, Henderson D (2014) Backpropagation applied to handwritten zip code recognition. Neural Comput 1:541–551
9. Mukherjee A, Venkataraman V, Liu B (2013) What Yelp fake review filter might be doing. In: Proceedings of the 7th international AAAI conference on weblogs and social media (ICWSM 2013), Cambridge, MA, pp 409–418
10. Xie S, Wang G, Lin S et al (2012) Review spam detection via temporal pattern discovery. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2012), Beijing, pp 823–831
11. Jindal N, Liu B (2008) Opinion spam and analysis. In: Proceedings of the first ACM international conference on web search and data mining (WSDM 2008), Stanford, pp 219–230
12. Jindal N, Liu B, Lim EP (2010) Finding unusual review patterns using unexpected rules. In: Proceedings of the 19th ACM conference on information and knowledge management (CIKM 2010). ACM, pp 1549–1552

13. Shojaee S, Murad MAA, Bin Azman A et al (2013) Detecting deceptive reviews using lexical and syntactic features. In: Proceedings of the 13th IEEE international conference on intelligent systems design and applications (ISDA 2013), Selangor, pp 53–58
14. Mccord M, Chuah M (2011) Spam detection on twitter using traditional classifiers. In: Autonomic and trusted computing. Springer Berlin Heidelberg, pp 175–186
15. Akoglu L, McGlohon M, Faloutsos C (2010) Oddball: spotting anomalies in weighted graphs. In: 14th Pacific-Asia conference on advances in knowledge discovery and data mining. Springer Berlin Heidelberg, pp 410–421
16. Muller E, Sánchez PI, Mulle Y et al (2013) Ranking outlier nodes in subspaces of attributed graphs. In: 2013 IEEE 29th international conference on data engineering workshops (ICDEW). IEEE, pp 216–222
17. Li N, Sun H, Chipman KC et al (2014) A probabilistic approach to uncovering attributed graph anomalies. In: Proceedings of the 2014 SIAM international conference on data mining (SDM 2014). SIAM, pp 82–90
18. Gao J, Liang F, Fan W et al (2010) On community outliers and their efficient detection in information networks. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 813–822
19. Jiang M, Cui P, Beutel A et al (2014) Inferring strange behavior from connectivity pattern in social networks. In: 18th Pacific-Asia conference on advances in knowledge discovery and data mining (PAKDD 2014). Springer International Publishing, pp 126–138
20. Prakash BA, Sridharan A, Seshadri M et al (2010) EigenSpokes: surprising patterns and scalable community chipping in large graphs. In: 14th Pacific-Asia conference on advances in knowledge discovery and data mining (PAKDD 2010). Springer Berlin Heidelberg, pp 435–448
21. Shah N, Beutel A, Gallagher B et al (2014) Spotting suspicious link behavior with fBox: an adversarial perspective. In: 2014 IEEE international conference on data mining (ICDM 2014). IEEE, pp 959–964
22. Cheng H, Tan PN, Potter C et al (2009) Detection and characterization of anomalies in multivariate time series. In: Proceedings of the SIAM international conference on data mining (SDM 2009). SIAM, pp 413–424
23. Li X, Han J (2007) Mining approximate top-k subspace anomalies in multi-dimensional time-series data. In: Proceedings of the 33rd international conference on very large data bases (VLDB 2007). VLDB Endowment, pp 447–458
24. Izakian H, Pedrycz W (2014) Anomaly detection and characterization in spatial time series data: a cluster-centric approach. IEEE Trans Fuzzy Syst 22(6):1612–1624
25. Beutel A, Xu W, Guruswami V et al (2013) CopyCatch: stopping group attacks by spotting lockstep behavior in social networks. In: Proceedings of the 22nd international conference on world wide web (WWW 2013). ACM, pp 119–130
26. Ferraz Costa A, Yamaguchi Y, Juci Machado Traina A et al (2015) RSC: mining and modeling temporal activity in social media. In: Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2015). ACM, pp 269–278
27. Lim EP, Nguyen VA, Jindal N et al (2010) Detecting product review spammers using rating behaviors. In: Proceedings of the 19th ACM international conference on information and knowledge management (CIKM 2010). ACM, pp 939–948
28. Mukherjee A, Kumar A, Liu B et al (2013) Spotting opinion spammers using behavioral footprints. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2013). ACM, pp 632–640
29. Lin C, He J, Zhou Y et al (2013) Analysis and identification of spamming behaviors in sina weibo microblog. In: Proceedings of the 7th workshop on social network mining and analysis (SNAKDD 2013). ACM, Article No. 5
30. Lam SK, Riedl J (2004) Shilling recommender systems for fun and profit. In: Proceedings of the 13th international conference on world wide web (WWW 2004). ACM, New York, pp 393–402

31. Günnemann S, Günnemann N, Faloutsos C (2014) Detecting anomalies in dynamic rating data: a robust probabilistic model for rating evolution. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2014). ACM, pp 841–850
32. Zhou W, Koh YS, Wen J et al (2014) Detection of abnormal profiles on group attacks in recommender systems. In: Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval (SIGIR 2014). ACM, pp 955–958
33. Zhang Y, Tan Y, Zhang M et al (2015) Catch the black sheep: unified framework for shilling attack detection based on fraudulent action propagation. In: Proceedings of the 24th international conference on artificial intelligence (IJCAI 2015). AAAI Press, pp 2408–2414
34. Gunes I, Kaleli C, Bilge A et al (2014) Shilling attacks against recommender systems: a comprehensive survey. Artif Intell Rev 42(4):767–799
35. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2008). ACM, pp 426–434
36. Salakhutdinov R, Mnih A (2008) Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In: Proceedings of the 25th international conference on machine learning (ICML 2008). ACM, pp 880–887
37. Koren Y (2010) Collaborative filtering with temporal dynamics. Commun ACM 53(4):89–97
38. Beutel A, Murray K, Faloutsos C et al (2014) CoBaFi: collaborative Bayesian filtering. In: Proceedings of the 23rd international conference on world wide web (WWW 2014). ACM, pp 97–108
39. Chirita PA, Nejdl W, Zamfir C (2005) Preventing shilling attacks in online recommender systems. In: Proceedings of the 7th annual ACM international workshop on web information and data management (WIDM 2005), Bremen, pp 67–74
40. Williams CA, Mobasher B (2006) Profile injection attack detection for securing collaborative recommender systems. Technical report, DePaul University
41. Burke R, Mobasher B, Williams C et al (2006) Classification features for attack detection in collaborative recommender systems. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2006). ACM, pp 542–547
42. Bank Card Professional Committee of China Banking Association (2018) Blue book on the development of China's bank card industry 2018. China Finance Press, Beijing
43. Breiman L (2001) Random forests. Mach Learn 45:5–32
44. Dong S, Huang Z (2013) Analysis of random forest theory. Ensemble Technol 2(1):1–7
45. Wang Y, Xia S (2018) Integrated study of a random forest algorithm review. J Inf Commun Technol 12(1):49–55
46. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
47. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55:119–139
48. Lecun Y, Bottou L, Bengio Y et al (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324
49. Lecun Y, Boser B, Denker J et al (2014) Backpropagation applied to handwritten zip code recognition. Neural Comput 1:541–551
50. Hsu TC, Liou S, Wang Y et al (2019) Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction. In: ICASSP 2019–2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), Hof, 12–17 May 2019, pp 1572–1576
51. Liu M, Zhang S, He Y (2017) Credit card customer default prediction based on improved fuzzy neural network. Fuzzy Syst Math 31(1):143–148
52. Wan M, Wang D, Goldman M et al (2017) Modeling consumer preferences and price sensitivities from large-scale grocery shopping transaction logs. In: Proceedings of the 26th international conference on world wide web, pp 1103–1112
53. Zhang J, Wedel M (2009) The effectiveness of customized promotions in online and offline stores. J Mark Res 46(2):190–206

54. Cheng AJ, Chen YY, Huang YT et al (2011) Personalized travel recommendation by mining people attributes from community-contributed photos. In: Proceedings of the 19th ACM international conference on multimedia, pp 83–92
55. Khan MUS, Khalid O, Huang Y et al (2015) MacroServ: a route recommendation service for large-scale evacuations. IEEE Trans Serv Comput 10(4):589–602
56. Wen YT, Yeo J, Peng WC et al (2017) Efficient keyword-aware representative travel route recommendation. IEEE Trans Knowl Data Eng 29(8):1639–1652
57. Liu Q, Chen E, Xiong H et al (2014) A cocktail approach for travel package recommendation. IEEE Trans Knowl Data Eng 26(2):278–293
58. Ge Y, Liu Q, Xiong H et al (2011) Cost-aware travel tour recommendation. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, pp 983–991
59. Ge Y, Xiong H, Tuzhilin A et al (2011) Collaborative filtering with collective training. In: Proceedings of the 5th ACM conference on recommender systems, pp 281–284
60. Pazzani MJ, Billsus D (2007) Content-based recommendation systems. Springer, Berlin, pp 325–341
61. Agarwal D, Chen BC (2009) Regression-based latent factor models. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp 19–28
62. Koohi H, Kiani K (2016) User based collaborative filtering using fuzzy C-means. Measurement 91:134–139
63. Linden G, Smith B, York J (2003) Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput 7(1):76–80
64. Ge Y, Xiong H, Tuzhilin A et al (2014) Cost-aware collaborative filtering for travel tour recommendations. ACM Trans Inf Syst 32(1):Article 4
65. Gantner Z, Drumond L, Freudenthaler C et al (2010) Learning attribute-to-feature mappings for cold-start recommendations. In: Proceedings of the 10th IEEE international conference on data mining, pp 176–185
66. Ahn HJ (2008) A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Inf Sci 178(1):37–51
67. Burke R (2007) Hybrid web recommender systems. Springer, Berlin, pp 377–408
68. Koren Y (2009) Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp 447–456
69. Ma H, King I, Lyu MR (2011) Learning to recommend with explicit and implicit social relations. ACM Trans Intell Syst Technol 2(3):Article 29
70. Wang J, Zhang Y (2011) Utilizing marginal net utility for recommendation in ecommerce. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval, pp 1003–1012
71. Salakhutdinov R, Mnih A, Hinton G (2007) Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th international conference on machine learning, pp 791–798
72. Wang H, Wang N, Yeung DY (2015) Collaborative deep learning for recommender systems. In: Proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining, pp 1235–1244
73. He X, Liao L, Zhang H et al (2017) Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web, pp 173–182
74. He X, He Z, Song J et al (2018) NAIS: neural attentive item similarity model for recommendation. IEEE Trans Knowl Data Eng 30(12):2354–2366
75. Chen J, Zhang H, He X et al (2017) Attentive collaborative filtering: multimedia recommendation with item- and component-level attention. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, pp 335–344
76. Devooght R, Bersini H (2017) Long and short-term recommendations with recurrent neural networks. In: Proceedings of the 25th conference on user modeling, adaptation and personalization, pp 13–21

77. Sedhain S, Menon AK, Sanner S et al (2015) AutoRec: autoencoders meet collaborative filtering. In: Proceedings of the 24th international conference on world wide web, pp 111–112
78. Xue HJ, Dai X, Zhang J et al (2017) Deep matrix factorization models for recommender systems. In: Proceedings of the 26th international joint conference on artificial intelligence, pp 3203–3209
79. Covington P, Adams J, Sargin E (2016) Deep neural networks for YouTube recommendations. In: Proceedings of the 10th ACM conference on recommender systems, pp 191–198
80. Guo H, Tang R, Ye Y et al (2017) DeepFM: a factorization-machine based neural network for CTR prediction. In: Proceedings of the 26th international joint conference on artificial intelligence, pp 1725–1731
81. Elkahky AM, Song Y, He X (2015) A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: Proceedings of the 24th international conference on world wide web, pp 278–288
82. He W, Xu F (2013) Promotional goods inventory model with demand dependent on inventory and shortage partially delayed. J Comput Appl 33(10):2950–2959
83. Huang Y, Zhang Y (2021) Sales forecasting of E-commerce industry based on GM(1, N)-Prophet combination model. J Southwest Univ Natl (Nat Sci Ed) 47(3):317–325
84. Luo Y, Lv Y, Li B (2009) Hybrid cigarette sales forecasting model based on ARMA. Appl Res Comput 26(7):2664–2668
85. Chang B, Zhang H, Liao C et al (2018) Retail sales forecasting based on selective ensemble ARMA combination model. Comput Meas Control 26(5):132–135
86. Feng C, Chen Z (2019) Application of weighted combination model based on XGBoost and LSTM in sales forecasting. Appl Comput Syst 28(10):226–232
87. Ma C, Wang X (2018) Dish sales prediction based on LSTM network model. Mod Comput (Prof Ed) 23:26–30
88. Li Z, Wu Q (2019) Research on stock prediction algorithm based on LSTM neural network. Fujian Comput 35(7):41–43
89. Li Z, Zhang K (2020) Comparison research of car sales forecasting model based on BP algorithm and LSTM algorithm. Econ Res Guide 20:84–88+93
90. Wang Y (2018) Research on E-commerce demand forecasting based on LSTM neural network. Master's thesis, Shandong University, Jinan
91. Jiang W (2019) Research on commodity sales forecasting based on WaveNet-LSTM network. Master's thesis, Guangdong University of Technology, Guangzhou
92. Law M, Kwok R, Ng M (2016) An extended online purchase intention model for middle-aged online users. Electron Commer Res Appl 20:132–146
93. Li D, Zhao G, Wang Z et al (2015) A method of purchase prediction based on user behavior log. In: Proceedings of the 15th IEEE international conference on data mining workshop, pp 1031–1039
94. Zhang Y, Pennacchiotti M (2013) Predicting purchase behaviors from social media. In: Proceedings of the 22nd international conference on world wide web, pp 1521–1532
95. Young Kim E, Kim YK (2004) Predicting online purchase intentions for clothing products. Eur J Mark 38(7):883–897
96. Liu G, Nguyen TT, Zhao G et al (2016) Repeat buyer prediction for ecommerce. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 155–164
97. Guo S, Wang M, Leskovec J (2011) The role of social networks in online shopping: information passing, price of trust, and consumer choice. In: Proceedings of the 12th ACM conference on electronic commerce, pp 157–166
98. Yin H, Cui B, Zhou X et al (2016) Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. ACM Trans Inf Syst 35(2):Article 11
99. Pálovics R, Szalai P, Kocsis L et al (2015) Solving RecSys challenge 2015 by linear models, gradient boosted trees and metric optimization. In: Proceedings of the 9th ACM conference on recommender systems challenge, pp 1–4

100. Lee J, Kim E, Lee S et al (2019) Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation. In: Proceedings of the IEEE international conference on computer vision (ICCV 2019), pp 6808–6818
101. Dewan R, Jing B, Seidmann A (2003) Product customization and price competition on the Internet. Manage Sci 49(8):1055–1070
102. Bernhardt D, Liu Q, Serfes K (2007) Product customization. Eur Econ Rev 51(6):1396–1422
103. Basu A, Bhaskaran S (2018) An economic analysis of customer co-design. Inf Syst Res 29(4):786–787
104. Guo S, Choi TM, Shen B et al (2018) Inventory management in mass customization operations: a review. IEEE Trans Eng Manage 66(3):412–428
105. Lim KH (2016) Recommending and planning trip itineraries for individual travellers and groups of tourists. In: Proceedings of the 26th international conference on automated planning and scheduling (ICAPS 2016), pp 115–120
106. Lim KH, Chan J, Karunasekera S et al (2017) Personalized itinerary recommendation with queuing time awareness. In: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval (SIGIR 2017), pp 325–334
107. Yu Q, Bouguettaya A (2011) Efficient service skyline computation for composite service selection. IEEE Trans Knowl Data Eng 25(4):776–789
108. Wagner F, Ishikawa F, Honiden S (2013) Robust service compositions with functional and location diversity. IEEE Trans Serv Comput 9(2):277–290
109. Zhang F, Hwang K, Khan SU et al (2015) Skyline discovery and composition of multi-cloud mashup services. IEEE Trans Serv Comput 9(1):72–83
110. Pine BJ, Victor B, Boynton AC (1993) Making mass customization work. Harv Bus Rev 71(5):108–111
111. Novemsky N, Ravi D, Norbert S et al (2007) The effect of preference fluency on consumer decision making. J Mark Res 44(8):347–357
112. Gilmore JH, Pine BJ (1997) The four faces of mass customization. Harv Bus Rev 75(1):91–102
113. Park CW, Jun SY, MacInnis DJ (2000) Choosing what I want versus rejecting what I do not want: an application of decision framing to product option choice decisions. J Mark Res 37(2):187–202
114. Wang Y, Han D (2012) How customers perceive mass customization: an empirical study based on customer self-efficacy, option presentation and customization satisfaction. J Soft Sci 26(4):140–144
115. Wu L (2014) Research on multi-objective distribution center location method based on customer satisfaction. J Logist Technol 6:95–97
116. Yang S, Yu L (2020) Research on express distribution route optimization problem based on genetic algorithm. Mod Inf Technol 4(9):99–103
117. Fan L, Lv P (2021) Campus takeaway distribution path planning based on improved genetic algorithm. Logist Sci Technol 1:14–19
118. Huang C, Huang G, Zhu X (2016) Research on the shortest path of delivery based on genetic algorithm. Sci Technol Commun 12(6):94–95
119. Zhai J, Tai Y (2018) Takeaway delivery route optimization based on time window constraint. Logist Sci Technol 3:15–18
120. Drops travel city traffic report: in the second quarter of 2018 (2018). https://sts.didistatic.com/official-website/reports/2018 in Q2 city run reports-finally.pdf
121. Cramer J, Krueger AB (2016) Disruptive change in the taxi business: the case of Uber. Am Econ Rev 106:177–182
122. Castiglione J, Chang T, Cooper D et al (2016) TNCs today: a profile of San Francisco transportation network company activity. San Francisco County Transportation Authority
123. Ban JX, Dessouky M, Pang J et al (2018) A general equilibrium model for transportation systems with e-hailing services and flow congestion. Working manuscript, University of Washington, Seattle, WA
124. Xu Z, Chen Z, Yin Y (2019) Equilibrium analysis of urban traffic networks with ride-sourcing services. SSRN 3422294. https://doi.org/10.2139/ssrn.3422294

125. Tang L, Zou T, Luo X et al (2017) Research on choice behavior of online car-hailing based on mixed logit model. Transp Syst Eng Inf Technol 18(1):108–114
126. Yuan L, Wu P (2018) Study on the choice intention and influencing factors of urban residents for online ride-hailing and taxi: logistic analysis based on survey data in Jiangsu Province. J Soft Sci 32(4):120–123
127. Lu K, Zhou J, Lin X (2019) Research on market pricing of ride-hailing platform considering cross-network externalities. Oper Res Manag 28(7):169–178
128. Li Y (2018) Research on pricing strategy of online ride-hailing platform based on two-sided market theory. Master's thesis, Chang'an University, Xi'an
129. Xu Z, Yan H (2019) Research on parking choice behavior and management strategy of hub online car-hailing. Transp Technol 8(3):155–165
130. Zhao S, King I, Lyu MR (2016) A survey of point-of-interest recommendation in location-based social networks. arXiv:1607.00647
131. Cheng C, Yang H, Lyu MR et al (2013) Where you like to go next: successive point-of-interest recommendation. In: Proceedings of 23rd international joint conferences on artificial intelligence, Beijing, pp 2605–2611
132. Zhao S, Zhao T, Yang H et al (2016) STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation. In: Proceedings of 30th AAAI conferences on artificial intelligence. AAAI Press, Menlo Park, CA, pp 315–322
133. Liu Q, Wu S, Wang L et al (2016) Predicting the next location: a recurrent model with spatial and temporal contexts. In: Proceedings of 30th AAAI conferences on artificial intelligence. AAAI Press, Menlo Park, CA, pp 194–200
134. Johnson CC (2014) Logistic matrix factorization for implicit feedback data. In: Workshop of advances in neural information processing systems. Springer, Berlin
135. Ahmadian S, Afsharchi M, Meghdadi M (2019) A novel approach based on multi-view reliability measures to alleviate data sparsity in recommender systems. Multimed Tools Appl 1–36
136. Ahmadian S, Meghdadi M, Afsharchi M (2018) A social recommendation method based on an adaptive neighbor selection mechanism. Inf Process Manage 54:707–725
137. Ye M, Yin P, Lee WC et al (2009) Exploiting geo-graphical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 325–334
138. Hang M, Pytlarz I, Neville J (2018) Exploring student check-in behavior for improved point-of-interest prediction. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, KDD'18-exploring student check-in behavior for improved point-of-interest prediction, pp 321–330
139. Xie M, Yin H, Wang H et al (2016) Learning graph-based POI embedding for location-based recommendation. In: Proceedings of the 25th ACM international on conference on information and knowledge management, Indianapolis, IN, pp 15–24
140. Liu Y, Pham T, Cong G et al (2017) An experimental evaluation of point-of-interest recommendation in location-based social networks. Proc VLDB Endow 10:1010–1021
141. Stepan T, Morawski JM, Dick S et al (2016) Incorporating spatial, temporal and social context in recommendations for location-based social networks. IEEE Trans Comput Soc Syst 3:164–175
142. Aliannejadi M, Rafailidis D, Crestani F (2018) A collaborative ranking model with multiple location-based similarities for venue suggestion. In: Proceedings of the 2018 ACM SIGIR international conference on theory of information retrieval. ACM, New York, pp 19–26
143. Cheng C, Yang H, King I et al (2016) Fused matrix factorization with geographical and social influence in location-based social networks. In: Twenty-sixth AAAI conference on artificial intelligence. Canada at the Sheraton Centre Toronto, Ontario, pp 17–23

144. Guo L, Wen Y, Liu F (2019) Location perspective-based neighborhood-aware POI recommendation in location-based social networks. Soft Comput 23:11935–11945
145. Guo L, Jiang H, Wang X (2018) Location regularization-based POI recommendation in location-based social networks. Information 9:85–95