# Design of a Book Management System Based on Recommendation Technology

**Authors:** Zhou Qiang, Xu Liyuan, Kong Beibei, Li Xi, Zhou Qiang

**Date:** 2016-01-25T00:00:00+00:00

## Abstract

Existing book management systems have not integrated book retrieval with book recommendation. The book management system presented in this paper categorizes books and provides retrieval by title and author, with recommendation results displayed on the retrieval results page. This paper introduces the design of the book management system; its recommendation engine employs item-based collaborative filtering algorithms and content-based recommendation algorithms. This paper provides a detailed exposition of these two algorithms, followed by a comprehensive introduction to the design of the recommendation system component within the book management system. At present, the book management system has completed both its preliminary and detailed design.

## Full Text

### Design of a Book Management System Based on Recommendation Technology

**Zhou Qiang[1], Xu Liyuan[1], Kong Beibei[1], Li Xi[2]**
[1] National Science Library, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Science and Technology of China, Anhui 230026, China

**Abstract:** Existing library management systems do not integrate book search and recommendation functionality. The system presented in this paper classifies books and provides search by title and author, with recommendation results displayed on the search results page. This article introduces the design of this library management system, whose recommendation engine employs an item-based collaborative filtering algorithm and a content-based recommendation algorithm. Both algorithms are described in detail, followed by a comprehensive overview of the recommendation system design within the library management system. The system has currently completed both preliminary and detailed design phases.

**Keywords:** Item-based collaborative filtering algorithm, Book, Recommendation, Content-based recommendation algorithm

## 1. Introduction to the Book Management System

Books are the ladder of human progress and serve as vital tools for disseminating knowledge, scientific technology, and preserving culture. Given the vast number of books available, the library management system designed in this paper includes books in fields such as electronic technology, computer science, and library and information science. The collection comprises both relatively classic works, such as canonical textbooks and masterpieces, and books on cutting-edge technologies like cloud computing and mobile internet. The proposed design integrates book search and recommendation functionality as an experimental solution to demonstrate the effectiveness of combining these two features.

## 2. Overview of Existing Library Management Systems

### 2.1 Peking University Library Retrieval System

The Peking University Library boasts an extensive, comprehensive, and precious collection. By the end of 2011, its total holdings reached approximately 11 million items, including over 8 million print volumes and more than 3 million digital resources introduced or developed in recent years, encompassing various databases, e-journals, e-books, and multimedia materials. In the search results, the book detail page displays information such as cover image, title, author, publication date, call number, total pages, collection location, and borrowing status. However, the book detail page does not provide any recommendation information.

### 2.2 Tsinghua University Library Retrieval System

By the end of 2013, the Tsinghua University Library collection totaled approximately 4.63 million items, forming a comprehensive system centered on natural sciences and engineering literature while also incorporating humanities, social sciences, and management science materials across multiple types and formats. In addition to Chinese and foreign language books, the collection includes ancient thread-bound books, journals, doctoral and master's theses from the university, and microform materials. Search results provide a list view with book covers and reservation options. The book detail page includes author, title, publisher, publication date, abstract, table of contents, collection location, and borrowing status. Like the Peking University system, the book detail page lacks recommendation features.

### 2.3 Dangdang.com

Dangdang.com (www.dangdang.com) is a globally renowned comprehensive online shopping platform. Books constitute one of its primary business categories,

with over 800,000 book and audio-visual products in its inventory. The platform's registered users span 32 provinces, municipalities, autonomous regions, and directly-administered municipalities across China. Search results display a list view with book covers. The book detail page includes cover image, title, author, publisher, publication date, ISBN, page count, abstract, author biography, table of contents, and online preview. Notably, the book detail page provides two types of recommendation data: "Customers who viewed this item also viewed" and "Customers who bought this item also bought."

## 3. Design of the Book Management System

The library management system integrates search and recommendation functionalities, with this paper focusing primarily on the recommendation system [1-2,5-9]. The recommendation algorithm employs an item-based collaborative filtering approach. Data is stored in a relational database, specifically MySQL in this implementation. To accelerate retrieval speed, Lucene indexes are created, and searches read data from these indexes. The web server utilizes Tomcat, with Squid configured at the front end for reverse proxying and load balancing. The system is deployed in two parallel sets, each comprising a database, Lucene index, and web server, maintaining consistency across both sets. Database operations are append-only and can be performed in real time, while updates to Lucene indexes and web server code are scheduled alternately during early morning hours.

[Figure 1: see original paper] illustrates the architecture of the library management system. The book data table stores comprehensive information including title, author, publisher, publication date, ISBN, page count, abstract, author biography, preface, and table of contents. Book cover files are stored in the file system. Separate author and publisher tables store respective names and biographical information. Data from these tables is provided to the browsing and retrieval system, with indexes created to expedite search operations.

The logging system is crucial for the recommendation engine. User behaviors—including browsing and searching actions, as well as borrowing transactions—are recorded in logs through the user interface. The recommendation system computes and returns results based on these user logs, displaying them in two modules on the book detail page: "Users who viewed this book also viewed" and "Users who borrowed this book also borrowed."

## 4. Brief Introduction to Recommendation Algorithms

With an enormous number of books available, locating personally interesting titles can be challenging. The recommendation system's role is to bridge users and information, helping users discover valuable content while ensuring information reaches interested audiences. This system employs two recommendation algorithms [1-13]: item-based collaborative filtering and content-based filtering. The content-based approach utilizes vector representation through the cosine

theorem. The algorithm proceeds in two stages: first, generating an item list using item-based collaborative filtering, then re-ranking this list using the cosine theorem.

### 4.1 Item-based Collaborative Filtering Algorithm

In this system, items refer to books. The item-based (book-based) collaborative filtering algorithm consists of two main steps: (1) calculating correlations between books, and (2) generating personalized recommendation lists based on these correlations and user history. The item similarity formula is given by: N(i) represents the number of users who like book i, and N(j) represents the number of users who like book j. The steps for calculating item similarity are: (1) construct a user-book inverted index (a list of liked books for each user); (2) for each user, increment by 1 the co-occurrence matrix C for every pair of books in their list, where C[i][j] records the number of users who like both book i and book j; (3) normalize matrix C to obtain the cosine similarity matrix W between books. After obtaining book similarities, user u's interest in a book is calculated using the following formula: Here, N(u) represents the set of books liked by user u, S(i,K) represents the set of K books most similar to book i, w_{ji} represents the similarity between books j and i, and r_{ui} represents user u's interest in book i. These computations are performed offline, enabling the construction of a book-book inverted index.

### 4.2 Cosine Theorem

The cosine theorem approach requires pre-loading a vocabulary into the database, with the total vocabulary size denoted as m. The process consists of four steps: (1) calculate TF/IDF (term frequency/inverse document frequency) values for all content words in the book's abstract and preface; (2) sort these TF/IDF values according to the words' positions in the vocabulary; (3) if a vocabulary word does not appear in the abstract or preface, its corresponding value is zero, forming an m-dimensional vector that represents the book as its feature vector. Books with similar feature vectors have similar content and thus appear closer in the recommendation list ranking. (4) Use the cosine theorem to compute vector similarity: for books X and Y with vectors (x1, x2, …, xm) and (y1, y2, …, ym) respectively, the cosine of the vector angle equals (x1*y1 + x2*y2 + … + xm*ym) / (sqrt(x1^2 + x2^2 + … + xm^2) sqrt(y1^2 + y2^2 + … + ym^2)). When the cosine equals 1, the books are identical; when the cosine approaches 1, the books are similar and thus ranked closely; smaller cosine values indicate greater distance in the recommendation list.

## 5. Design of the Recommendation System

[Figure 2: see original paper] illustrates the data flow in the recommendation system [14-22]. User features consist of two types: demographic characteristics extracted from registration information, and behavioral features computed from

user actions. An item feature vector comprises features and their weights, with calculation considering several factors: (1) User behaviors are categorized as browsing or borrowing, with borrowing assigned greater weight. (2) The timing of user behavior, where recent browsing and borrowing activities are more significant. (3) The frequency of user behavior—users may view a book multiple times, and this frequency reflects interest level. Books with more views receive higher feature weights; similarly, books viewed by more users gain higher weights. For the same book, borrowing carries higher weight than browsing. (4) Book popularity—behavior toward popular books may result from trend-following rather than genuine interest, thus less popular books receive higher weights.

After obtaining the user feature vector, an initial book recommendation list is generated from offline correlation tables, stored in the format: feature ID, book ID, title, author, and weight. This preliminary list is then filtered to remove low-quality books. The filtered results undergo further recommendation processing using the content-based algorithm. TF/IDF values are calculated for all content words in book abstracts and prefaces, sorted by vocabulary position to generate book feature vectors, followed by cosine similarity computation. The final recommendation list is generated offline and written to Lucene indexes to accelerate query performance.

This paper first introduced the requirements for the library management system, followed by an overview of existing systems. Building upon this foundation, the system design was presented, with detailed explanations of the item-based collaborative filtering algorithm and content-based filtering algorithm, culminating in a comprehensive description of the recommendation system design.

## References

[1] Marco Degemmis, Pasquale Lops, Giovanni Semeraro. A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation[J]. User Modeling and User-Adapted Interaction. 2007, 3: 217-255.

[2] Rosario Girardi, Leandro Balby Marinho. A domain model of Web recommender systems based on usage mining and collaborative filtering[J]. Requirements Engineering. 2007, 1:

[3] Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach[J]. Data Mining and Knowledge Discovery. 2004, 1: 53-87.

[4] Ken Goldberg, Theresa Roeder, Dhruv Gupta, Chris Perkins. Eigentaste: A Constant Time Collaborative Filtering Algorithm[J]. Information Retrieval. 2001, 2: 133-151.

[5] Ma Hongwei, Zhang Guangwei, Li Peng. Survey of Collaborative Filtering

Recommendation Algorithms[J]. Mini-Micro Computer Systems, 2009, 07: 1282-1288.

[6] Wang Yanran, Chen Mei, Wang Hanhu, Zhang Xin. A Content-Based Filtering Recommendation Algorithm for Scientific and Technical Literature[J]. Computer Technology and Development, 2011, 02: 66-69.

[7] Liu Meilian, Liu Tongcun, Li Xiaolong. Research on Recommendation Algorithm Based on User Interest Feature Extraction[J]. Application Research of Computers, 2011, 05: 1664-1667.

[8] Yang Bo, Zhao Pengfei. Survey of Recommendation Algorithms[J]. Journal of Shanxi University (Natural Science Edition), 2011, 03: 337-350.

[9] Xing Zhe, Liang Jingfan, Zhu Qing. Multi-dimensional Adaptive Collaborative Filtering Recommendation Algorithm[J]. Mini-Micro Computer Systems, 2011, 11: 2210-2216.

[10] Liu Xudong, Chen Deren, Wang Huimin. An Improved Collaborative Filtering Recommendation Algorithm[J]. Journal of Wuhan University of Technology (Information and Management Engineering Edition), 2010, 04: 550-553.

[11] Zhao Liang, Hu Naijing, Zhang Shouzhi. Design of Personalized Recommendation Algorithm[J]. Journal of Computer Research and Development, 2002, 08: 986-991.

[12] Guo Yanhong, Deng Guishi. Research on a Personalized Recommendation Algorithm for Collaborative Filtering[J]. Application Research of Computers, 2008, 01: 39-41+58.

[13] Li Cong, Liang Changyong, Ma Li. A Collaborative Filtering Recommendation Algorithm Based on Domain Nearest Neighbors[J]. Journal of Computer Research and Development, 2008, 09: 1532-1538.

[14] Wang Guoxia, Liu Heping. Survey of Personalized Recommendation Systems[J]. Computer Engineering and Applications, 2012, 7: 66-76.

[15] Xu Hailing, Wu Xiao, Li Xiaodong, Yan Baoping. Comparative Study of Internet Recommendation Systems[J]. Journal of Software, 2009, 2: 350-

[16] Liu Jianguo, Zhou Tao, Wang Binghong. Research Progress on Personalized Recommendation Systems[J]. Progress in Natural Science, 2009, 01: 1-15.

[17] Liu Lu, Ren Xiaoli. Research Progress and Prospects of Recommendation Systems[J]. China Journal of Information Systems, 2008, 1: 82-90.

[18] Chen Dingquan, Zhu Weifeng. Association Rules and Library Bibliographic Recommendation[J]. Information Studies: Theory and Application, 2009, 6: 81-84.

[19] Li Shuqing, Xu Xia, Xu Minjia. A Method for Measuring Book Recommendation Quality Based on Reader-Book Bipartite Networks and Personalized

Book Recommendation Services[J]. Journal of Library Science in China, 2013, 3: 83-95.

[20] Fan Xu. A Case Study of Online Bibliographic Recommendation Systems: Douban and National Library of China[J]. Library Science Research, 2008, 8: 44-48.

[21] Ding Xue. Research on Intelligent Book Recommendation System Based on Data Mining[J]. Information Studies: Theory and Application, 2010, 5: 107-110.

[22] Wang Yi, Ma Shangcai. Design and Application of Personalized Recommendation System Based on User Behavior[J]. Computer Systems and Applications, 2010, 8: 29-33.

*Note: Figure translations are in progress. See original paper for figures.*