Information and Knowledge Management (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/infkm.2025.060108
ISSN 2523-5893 Vol. 6 Num. 1

Research on Learning Resource Recommendation Based on Latent Factor Model

Juan Wang[1,2,a,*], Lida An[1,b]

[1]*College of Computer Science, Haojing College of Shaanxi University of Science & Technology, Xi'an, Shaanxi, China*
[2]*Faculity of Electronic Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, China*
[a]*434897487@qq.com*, [b]*ald2010@163.com*
[*]*Corresponding author*

Keywords: Latent Factor Model; Learning Resource Recommendation; Cognitive Tutors

Abstract: Research on personalized learning supported by recommendation technologies must address challenges such as acquiring authentic educational datasets and adapting these technologies across different domains. This paper adopts the Latent Factor Model in the real dataset, which is collected from the intelligent tutoring system Cognitive Tutors, to achieve personalized exercise resource recommendations. This study implements the latent factor model recommendation for learning materials with Python. In addition to achieving personalized recommendations for educational content, the study also explore the theoretical foundation, motivation, and feasibility of domain-transferable algorithm design to provide a reference for subsequent researchers to conduct related comparative experiments and optimization of our algorithm. Moreover, the study hope that this exploration can enlighten the basic theoretical research on the application of recommendation algorithms in the field of education. This exploration work will also motivate research and development in personalized education technology; The study believe that more advanced computational models have the potential to improve educational practices and learner experiences and achieve effective and scalable personalized learning.

1. Introduction

Since the inception of the Netflix Prize competition, model-based collaborative filtering recommendation algorithms have received widespread attention in the industry and have been gradually applied to recommendations in various fields such as news and music, among which the implicit semantic model stands out the most. Compared to neighborhood-based collaborative filtering algorithms, many models, including the implicit semantic model, can achieve a high coverage rate even for sparse rating matrices. Moreover, the implicit semantic model utilizes data redundancy in correlated data to create low-dimensional approximations, significantly reducing the space complexity of offline computations[1].

In this context, this paper attempts to apply the Latent Factor Model to the real public online teaching dataset released by KDD Cup 2010, taken from the intelligent tutoring system Cognitive

Tutors[2], to reveal the hidden features between learners and exercises from the data itself, thus achieving personalized exercise resource recommendations. While implementing the learning resource Latent Factor Model recommendation based on Python, the paper also tries to explain the theoretical basis, motivation, and applicability of the algorithm application domain migration design, in order to provide a reference for comparison and testing for subsequent algorithm improvements, and to bring some inspiration to the basic theoretical research of the recommendation algorithm's migration application in the field of education

2. Latent Factor Model

2.1 Basic Idea of the Latent Factor Model

For the issue of how to recommend items of interest to target users, if one knows the interest category to which an item belongs and the interest classification of the target user, recommendations can be made from the collection of items in the corresponding interest category. To implement the above recommendation, three issues need to be resolved:

(1) How to categorize items based on interest differences?

(2) How to determine the categories of interest to the target user and the degree of interest in those categories?

(3) How to select items for recommendation within a specific category? And how to determine the weight of an item within that category?

The conventional solution to the above problem is through manual classification, such as having editors categorize books according to the Chinese Library Classification. However, manual classification is not only time-consuming and labor-intensive but also has many limitations. Therefore, automatic clustering based on user behavior statistics has been adopted, utilizing implicit feature connections between user interests and items through latent semantic analysis techniques.

According to the study[3], compared to manual classification, the categories derived from Latent Semantic Analysis (LSA) technology come from the statistics of user behavior and represent the users' perspective on item categorization. This technology allows for specifying the final number of categories; the larger the number, the more fine-grained the categorization, and vice versa. The technology calculates the weight of items belonging to each category, rather than rigidly assigning items to a specific class. Each category provided by this technology is not based on the same dimension; it is calculated based on the common interests of users. If the common interest of users is a certain dimension, then the categories given will also be of the same dimension[4]. This technology can determine the weight of items in each category by statistically analyzing user behavior. If users who like a certain category also like a particular item, then the weight of that item in that category may be relatively high[5].

The Latent Semantic Analysis technique has derived many models and methods, including the Latent Factor Model (LFM). For example, Probabilistic Latent Semantic Analysis (pLSA)[6], Latent Dirichlet Allocation (LDA) [7], Matrix Factorization (MF) and so on. These methods are essentially interconnected and can all be used for personalized recommendations. This paper will use the Latent Factor Model to find the connection between user interests and hidden item features.

2.2 Basic Principles of the Latent Factor Model

From a matrix perspective, the Latent Factor Model can be represented as follows:

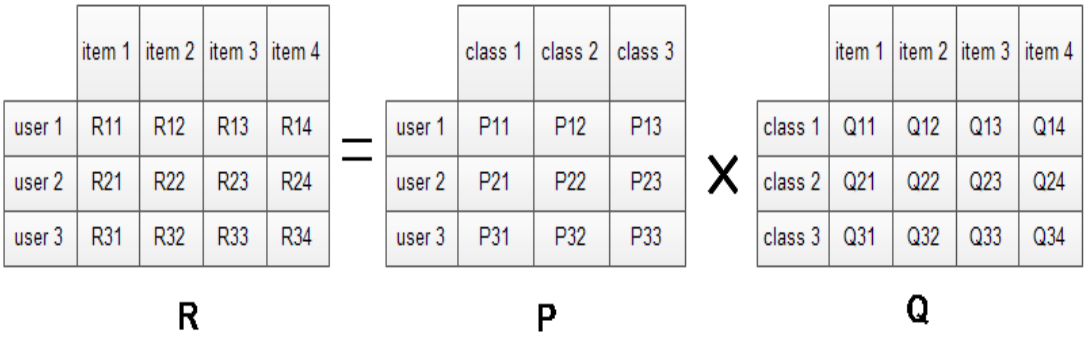

Figure 1: Latent Factor Model Principle Matrix Representation Diagram

In Figure 1, matrix R represents the preference information of users towards items, where R_{ij} represents the interest of User i in Item j; matrix P represents the preference information of users towards different categories of items, where P_{ij} represents the preference degree of User i for Class j; matrix Q represents the weight information of items belonging to various categories, where Q_{ij} represents the probability size of Item j belonging to Class i.

The Latent Factor Model aims to decompose matrix R into the product of matrices P and Q, linking users (User) and items (Item) through item categories (Class) in the matrix. For a user, if their interest in all items can be calculated, the items can be ranked by interest level and recommendations can be made. Therefore, matrices P and Q are used to derive the complete matrix R with filled-in information. Below is the formula for calculating user u's interest in item i in the Latent Factor Model.

$$\text{Preference}(u,i) = r_{ui} = p_u^T q_i = \sum_{f=1}^{F} p_{u,k} q_{i,k} \tag{1}$$

In the formula, $p_{u,k}$ and $q_{i,k}$ are the parameters of the model, where $p_{u,k}$ measures the relationship between user u interest and the k latent class, and $q_{i,k}$ measures the relationship between the k-th latent class and item i. The number of latent classes k can be determined independently. The calculation of the model parameters $p_{u,k}$ and $q_{i,k}$ requires solving the optimization loss function based on the existing data in matrix R.

2.3 Differences in applying Latent Factor Model to learning recommendations

Transferring the Latent Factor Model, which is maturely applied in the commercial field, to the educational field reveals significant differences in the goals and motivations of recommendations[8]. In the commercial field, the ultimate goal of personalized customer needs is to provide products and services that meet customer demands. In contrast, the concept of personalization in the educational field not only includes fostering students' individual interests but also involves the objective evaluation of the knowledge logic of academic content and students' existing knowledge levels to achieve specific educational and academic goals. Therefore, the goal of teaching resource recommendations must balance individual preferences with the normative requirements of achieving educational objectives.

Beyond highlighting the differences in the application of recommendation algorithms across fields, applying neighborhood-based recommendation methods to the educational field also shares a common methodological foundation. Since the Latent Factor Model inherently discovers latent factors that connect users and items from the data itself, it inspires us to use this model to uncover hidden features that link learners and learning resources. Taking exercise resources as an example, the information represented by these hidden features could be the potential knowledge points tested

by the exercises, the potential difficulty of the exercises, or the potential learning level of the learners. By transferring the Latent Factor Model to student groups and leveraging historical learning data, we can identify factors that promote the academic progress of target students. Based on this[9], targeted learning resource recommendations can be made, which theoretically represents a valuable approach to educational resource recommendation.

3. Experiment and result analysis

3.1 Data Set

The data used in this paper is the Cognitive Tutors dataset released by KDD Cup 2010, which contains interaction data between students and the math learning module in the system. The original data includes 18 data items, involving basic information about students, exercise-related information, and human-computer interaction records. For example, "Correct Step Duration (sec)" indicates the time taken from the beginning of a step to correctly solve it, while "Incorrects" represents the number of incorrect attempts.

To achieve the goal of this paper, which is to recommend exercises that align with learners' learning objectives based on their existing learning experience, namely, the mastery of knowledge reflected by the completion of exercises, it is necessary to construct a "student-knowledge point-mastery" matrix from the original data as shown in Table 1.

This matrix describes the mastery of various knowledge points by students, with rows representing a student and columns representing a knowledge point. The data in the matrix represents the error rate of exercises done by student i involving knowledge point j, formulated as:

$$s_{ij} = \frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - avg_i\right) \tag{2}$$

In this formula, n represents the total number of exercises involved in knowledge point j; x_{ij} represents the error rate of exercises involving knowledge point j done by students i; avg_i represents the average error rate of all questions done by student i. This data ranges from -1 to 1, the larger the value, the worse the student's mastery of the subject.

Table 1. "Student-Knowledge Point" mastery matrix

Student	Knowledge			
	k_1	k_2	…	k_N
S_{1N}	0.000	0.019	…	0.681
S_{2N}	0.002	0.135	…	0.426

3.2 Results and analysis

This paper randomly selects 20% of the data set from the "student-knowledge point-mastery" matrix as the test set before model training, and uses offline experimental methods to evaluate the recommendation results from three indicators: accuracy, recall, and coverage.

In the Latent Factor Model, there are four important parameters: the number of hidden features F, learning rate alpha, regularization parameter lambda, and the ratio of negative samples to positive samples ratio. This study focuses on examining the impact of the number of hidden features F on the recommendation results when alpha is fixed at 0.02, lambda at 0.01, and ratio at 1, as specifically shown in Table 2.

Table 2. Performance of the LFM algorithm on the Cognitive Tutors dataset under F

F	Accuracy	Recall	Coverage
20	0.02%	0.0039%	11.26%
40	0.04%	0.0098%	17.57%
60	0.09%	0.0196%	22.42%
80	0.10%	0.0235%	25.89%
100	0.11%	0.0254%	28.35%
120	0.05%	0.0117%	30.15%
140	0.08%	0.0176%	31.58%
160	0.03%	0.0078%	33.40%
180	0.03%	0.0078%	34.29%
200	0.06%	0.0137%	33.97%

Turning the data from Table 2 into a graph, we can see more clearly the performance changes of the LFM algorithm in the Cognitive Tutors dataset under F.

As shown in Figure 2, with the increase of the number of hidden features, the accuracy and recall rate of LFM have certain improvements, among which the improvement of accuracy is significantly higher than that of recall rate. Moreover, when the number of hidden features is 100, the accuracy and recall rate reach their maximum values; afterward, as the number of hidden features continues to increase, both the accuracy and recall rate decrease and exhibit a fluctuating state. From the data indicators, it is evident that the recommendation of the implicit semantic model applied to this experimental education dataset has a significantly lower accuracy and recall rate compared to the common experimental datasets. According to the research conclusions of reference [2], under the parameter settings of 100 hidden features, alpha=0.02, lambda=0.01, and ratio=1 for the MovieLens dataset, its accuracy and recall rate are 21.74% and 10.50%, respectively. In this experiment, under the same parameters, the accuracy and recall rate are 0.11% and 0.0254%, respectively, which is two orders of magnitude lower.

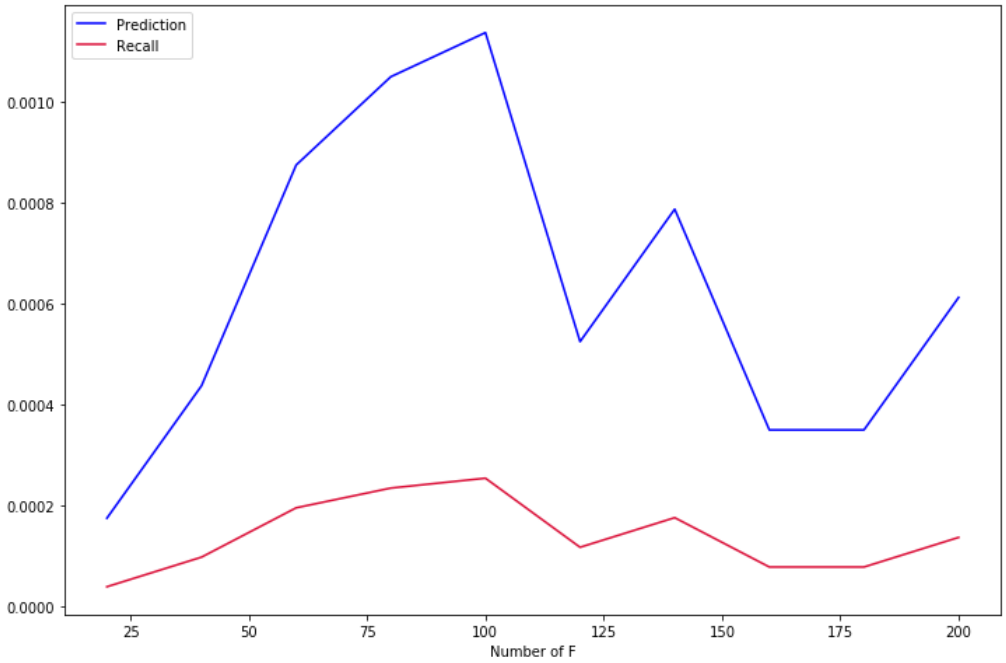

Figure 2: Latent Factor Model Recommendation Performance - Accuracy and Recall

Figure 3 shows the curve of the coverage rate of recommendations by the implicit semantic model as the number of implicit features changes. It can be seen that as the number of implicit features increases, the coverage rate gradually improves; when the number of implicit features is 180, the coverage rate reaches its peak, and then it shows a downward trend. Additionally, the coverage rate metric does not show significant order-of-magnitude differences with the performance of other common datasets in the implicit semantic model recommendation, such as the research results in literature [2]. With 100 implicit features and other parameters consistent with this study, the coverage rate of LFM in the MovieLens dataset is 51.19%, while in this study it is 28.35%. Although the coverage rate in this study is significantly lower than that of the MovieLens dataset, there is no order-of-magnitude difference.

Analyzing the possible reasons for the low accuracy and recall rate in this study: one is the impact of data sparsity, the second is the selection and use of implicit feedback data, and in addition, the requirements of specific application fields on recommendation strategies will also affect the recommendation effect.

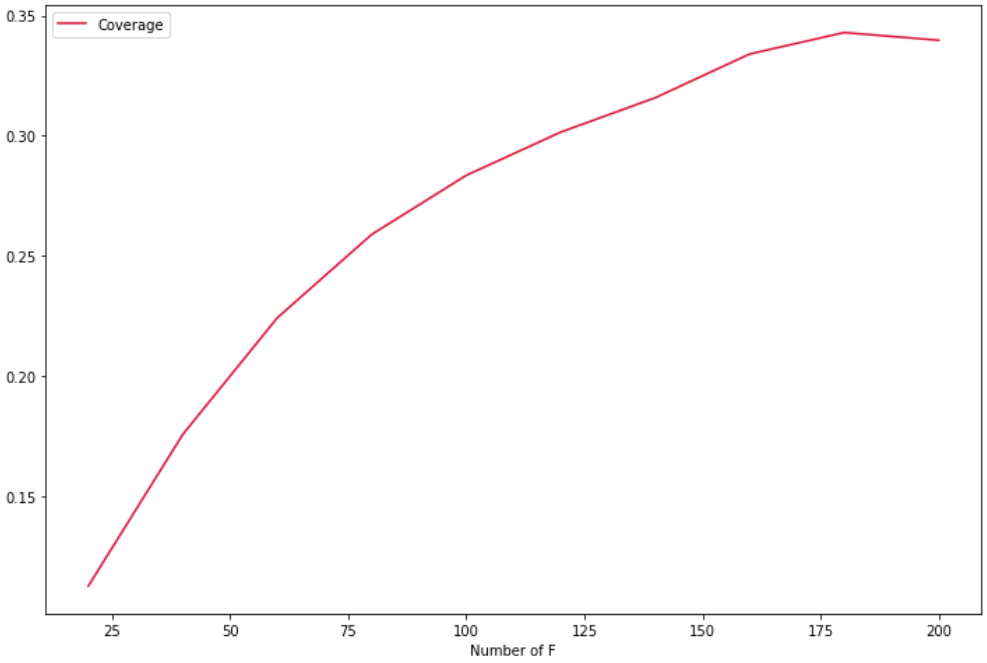

Figure 3: Latent Factor Model Recommendation Performance - Coverage

The literature research points out that when the dataset is very sparse, the performance of LFM will significantly decline, and it may even underperform the performance of neighborhood-based collaborative filtering recommendation algorithms. In the Cognitive Tutors dataset used in this study, after deduplication, the number of learners is 1146, the number of exercises is 19355, and the effective answer records of learners are 255665. This corresponds to a data missing rate of 98.85% in the "student-knowledge point-mastery" matrix. Such a high rate of missing data will inevitably make the model's low-order approximation inaccurate, thereby affecting the recommendation results.

Furthermore, this study refers to the common practice in commercial fields and uses implicit feedback datasets to sample positive and negative cases for model training. That is, for a learner, exercises with practice behavior are regarded as positive samples, and some exercises without practice behavior are sampled as negative samples. However, during sampling, the number of positive and negative samples for each learner is ensured to be comparable. The main reason for using implicit feedback data in commercial fields is that explicit feedback data is not always

available, and implicit feedback data is larger in quantity. This study uses the "student-knowledge point-mastery" matrix extracted from the Cognitive Tutors dataset. Explicit feedback data can clearly reflect whether learners have mastered the knowledge points involved in related exercises. Literature research points out that LFM can achieve good accuracy in solving rating prediction problems on explicit feedback data. Therefore, this study should consider directly using explicit feedback data for model training and exercise mastery prediction.

Finally, in the commercial field, for items that users have not interacted with, it is generally either because the users are not interested in the item, or they are interested but have not directly found the item in the system. In this study, exercises that learners did not practice might be due to their subjective feeling that they are too difficult or too easy and do not want to attempt them. Another possibility is that learners want to practice but did not directly see the exercise in the system. These two scenarios are similar to those in the commercial field. However, there is also a possibility that the knowledge points involved in the exercise are far from the cognitive stage required by the learner, and are not even in the learner's list of alternative exercises. This difference naturally requires that when recommending exercises based on prediction scores, there should be corresponding strategic constraints, because learners will only practice exercises related to their learned topics. Recommending exercises on topics that learners have not studied is meaningless and will naturally affect the effectiveness of the recommendation.

4. Conclusion

This project applies the Latent Factor Model to learning resource recommendations on real datasets in the educational field. From the perspective of recommendation results, there is still much room for improvement. Future enhancements can be made in the following areas: 1) Cluster learners based on their knowledge level similarity, or cluster exercises based on the logic of the knowledge itself, in order to reduce the volume of data and indirectly address the issue of overly sparse data. 2) Directly use explicit feedback data from the dataset for model training. 3) Improve recommendation algorithm evaluation metrics and formats, emphasizing the objective characteristic of educational field recommendations to achieve learners' learning goals. User experiments can also be used to evaluate recommendation effects. 4) Considering the time complexity of offline calculations for the Latent Factor Model and its inability to perform real-time online recommendations, an effective hybrid recommendation strategy is also a method to enhance recommendation performance.

Acknowledgement

This work was supported by

1) Key Education Reform Project of Haojing College, Shaanxi University of Science & Technology - "Research on the Training Model of New Engineering Specialty Talents under the Background of Independent College Transformation" (2023JG04)

2) Xianyang City Science and Technology Innovation Team - Innovation Team for Information Platform of Applied Universities (Project ID: L2023CXNLCXTD010)

References

[1] Charu C. Aggarwal. Recommender Systems: The Textbook [M]. Translated by Li Lingli et al. Beijing: China Machine Press, 2018: 21-52.

[2] KDD CUP 2010.Educational Data Mining Challenge [EB/OL].https://pslcdatashop. web.cmu.edu/KDDCup/,2010.

[3] Tianyi Pan, Yan Song. Non-linear representation of latent factor model based on trust relationship [J/OL]. Electronic Technology, 1-10 [2025-01-31]. https://doi.org /10.16180 /j.cnki.issn1007-7820.2025.02.007.

[4] Qi Wu, Wenhui Nie. Research on recommendation algorithm based on user clustering and time-implicit semantic model [J]. Computer and Digital Engineering, 2023, 51(03):561-565.
[5] Kong Huan, Huang Shucheng. Optimization of recommendation algorithm based on hidden semantic model [J]. Computer and Digital Engineering, 2022, 50(10):2197-2201.
[6] Lin Jianghao, Gu Yeli, Zhou Yongmei, et al. An automatic annotation method for news commentary sentiment categories based on PLSA [J]. Computer System Applications, 2019, 28(01):207-211. DOI:10.15888/j.cnki.csa.006687.
[7] Zhenzhen Wang, Ming He, Yongping Du. Text similarity calculation based on LDA topic model [J]. Computer Science, 2013, 40(12): 229-232.
[8] X Pan, X Li and M Lu. A MultiView courses recommendation system based on deep learning. Proc. Int. Conf. Big Data Informatization Educ. (ICBDIE), 2020, (4): 502-506.
[9] B D Walker. Course difficulty and student rating of teaching. Improving College Univ. Teaching, 1974, 22(1): 18-20.