Proceedings of the 2004 IEEE
Conference on Cybernetics and Intelligent Systems
Singapore, 1-3 December, 2004

A Movie Recommender System Based on Inductive Learning

PENG LI
CISS, IGSSE, Tokyo Institute of Technology, Japan
4259 Nagatuta, Midori-ku, Yokohama, Japan
liorlee@nii.ac.jp

SEIJI YAMADA
National Institute of Informatics, Japan
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
seiji@nii.ac.jp

***Abstract*—Recommender Systems apply intelligent access technologies to large information systems. These systems, especially collaborative filtering based ones, are achieving widespread success on the Web. In recent years, the amount of available information and the number of visitors to Web sites are increasing enormously. New recommender system technologies are needed that can quickly produce high quality recommendations, even for very large-scale information resources. In this paper we apply inductive learning algorithm to the recommendation process. Instead of computing user-user or item-item similarities, we construct a decision tree to represent user preference. Recommendations are performed by decision tree classification. To inspect the effectiveness of this technology, we set up a movie recommender system based on inductive learning and make online experiments for evaluation. Our results suggest that inductive-learning-based technology is promising for the solution of the very large-scale problems and high-quality recommendations can be expected.**

Keywords—recommender system; inductive learning

I. INTRODUCTION

With the dramatic development of computer and internet technologies, everyone is capable of accessing the tremendous amount of information easily. Moreover, the amount of information in the world is increasing far more quickly than our ability to process it. Especially in recent years, many contents providers begin to provide news, music and movie resources. It is difficult to obtain relevant contents through the flood of these resources.

One of the approach to solve this problem is the development of recommender systems[1]. Recommender systems can automatically apply personalized recommendations for information, products or services during a live interaction.

The elemental technology of recommender systems is collaborative filtering. Collaborative filtering is a technique by which the interest of a user for an object is predicted from the knowledge of the interest of other users for this object. Many business systems such as LikeMinds (Macromedia,Inc), GroupLen(NetPerceptions,Inc), AuraSearch(AuraLine&FEG) are based on collaborative filtering techniques. Collaborative filtering has achieved great success in both academic and business fields[2,3,4]. However, the tremendous growth in the amount of available information and the number of users poses some potential challenges such as *sparsity* and *scalability* problems[5]. In addition, collaborative filtering has a typical problem that the reason and the adequacy of recommendation is *not transparent*.

In this research, we propose a new recommendation technology based on inductive learning in the attempt of solving the sparsity, scalability and transparency problems. Our basic idea can be described in three steps: First, correspond an item's evaluated value and its attributes to the class and attributes of a decision tree. Secondly, perform inductive learning to construct decision trees that represent user's preferences. Finally, obtain predicted value of a new item through decision tree classification.

Inductive-learning-based recommendation technology solves the sparsity problem by sharing attributes of each item(we call them contents preferences). The computation cost during inductive learning process is low enough to construct decision tree instantly. Moreover, the computation cost during recommendation process has a linear relationship to the total amount of available items. Therefore it will not become enormous with the growth of scalability.

In order to inspect the effectiveness of this technology, we set up a movie recommender system based on inductive learning and make online experiments for evaluation. There are two reasons why we choose movie data as our target. One reason is the compilation of movie database is sophisticated and many solid movie databases like IMDb[1] (Internet Movie Database) are available on the Web. The other reason is that we can obtain the contents preferences of each item from an online database called PAOON [2] directly, which play an important role in the recommendation process.

We begin with a brief overview of traditional recommendation technologies. Inductive-leaning-based recommendation technology is then introduced, followed by a detailed description of our movie recommender system. Evaluation experiment is then discussed. Finally, we present some tentative conclusions and recommendations for further study.

[1] http://www.imdb.com/

[2] http://www.paoon.com/

0-7803-8643-4/04/$20.00

II. OVERVIEW OF TRADITIONAL RECOMMENDATION TECHNOLOGIES

A. CONTENTS-BASED FILTERING

Information filtering techniques fall in two independent categories: content-based filtering and collaborative filtering. *Content-based filtering* is based on content analysis of the considered objects, e.g. term frequency for documents, and its relation to the user's preferences. For content-based filtering it is therefore necessary that the results of content analysis and user preferences can reliably and automatically be determined. While recent research shows good results for the content-based filtering of documents, filtering of other media, as audio and video, is hard due to the limitations of content analysis technology available. *Collaborative filtering*, on the other hand, does not show this limitation.

B. COLLABORATIVE FILETERING

Collaborative filtering is based on the premise that people looking for information should be able to make use of what others have already found and evaluated. In collaborative filtering, objects are selected for a particular user when they are also relevant to similar users and, in general, the content of the objects is ignored. Therefore, collaborative filtering is especially interesting for items for which content analysis is difficult or impossible.

Collaborative filtering based recommender systems are achieving widespread success on the Web. However, the tremendous growth in the amount of available information and the number of users poses some potential challenges. These are:

●SPARSITY

The sparsity problem occurs under the situation that the amount of evaluated items of each user is by far less than the total amount of available items. Many commercial recommender systems are used to evaluate large item sets(e.g.,Amazon.com recommends books an CDnow.com recommends music albums). In these systems, even active users may have purchased under 1% of the items[5]. It is difficult to find neighbors in this situation. As a result, the accuracy of recommendations may be poor.

●SCALABILITY

Nearest neighbor algorithms require computation that grows with both the number of users and the number of items. With millions of users and items, a typical web-based recommender system running existing algorithms will suffer serious scalability problems[5].

●TRANSPARENCY

The relationship between the recommended information and user preferences is not clear. It is difficult for a user to understand how the items are recommended.

III. INDUCTIVE-LEARNING BASED RECOMMENDATION TECHNOLOGY

A. INDUCTIVE-LEARNING AND DECISION TREE

Inductive Learning is a typical learning task in machine learning. Given a data set, inductive learning aims to discover patterns in the data and form concepts that describe the data. Research in inductive learning has sustained for decades.

```
C4.5 [release 8] decision tree generator  Tue Aug 19 16:28:02 2003
-------------------------------------

Read 48 cases (7 attributes) from movie13.data

Decision Tree:
story = 2: 2 (5.0/1.0)
story = 1: 3 (0.0)
story in {3,4,5}:
|   story in {1,2}: 3 (0.0)
|   story in {3,4}:
|   |   story = 3: 3 (13.0/1.0)
|   |   story in {1,2,5}: 3 (0.0)
|   |   story = 4:
|   |   |   performance = 3: 3 (2.0)
|   |   |   performance in {1,2}: 4 (0.0)
|   |   |   performance in {4,5}:
|   |   |   |   emotion = 2: 3 (1.0)
|   |   |   |   emotion = 5: 4 (2.0/1.0)
|   |   |   |   emotion = 1: 4 (0.0)
```

Figure 1. Decision tree constructed by C4.5.

A decision tree is a tree in which each "branch node"(attribute) represents a choice between a number of alternatives, and each "leaf node"(class) represents a classification or decision. Two well-known algorithms for constructing decision trees are C4.5 [Quinlan 1993] [6]and CART (Classification and Regression Tree) [Breiman et al. 1984].

C4.5 is a typical machine learning program for the efficient synthesis of decision trees. Figure 1 shows an example of decision tree constructed by C4.5.

B. RECOMMENDATION WITH C4.5

The detailed recommendation process based on inductive learning is listed below:

(1) Input of training data

The user evaluates several items through GUI.

(2) Construction of decision tree

Correspond an item's evaluated value and its contents preference to the class and attributes of a decision tree. The decision tree constructed is used as a user profile.

(3) Classification of unvalued items

Use the decision tree constructed at step (2) to classify the items that are not evaluated by the user. The decision tree classes are corresponded to predicted values. Each unvalued item is classified according to its contents preferences and gets a particular predicted value.

(4) Building of candidates list

TABLE I. CREDIT PREFERENCE

	Credit Preference		
	Genre	Director	Starring
A Beautiful Mind	Drama; Romance	Ron Howard	Russell Crowe;Jennifer Connelly
Top Gun	Action; Romance	Tony Scott	Tom Cruise; Kelly McGillis
the Godfather	Drama; Suspense/Crime	Francis Ford Coppola	Marlon Brando; Al Pacino
Saving Private Ryan	War; Drama	Steven Spielberg	Tom Hanks; Matt Damon

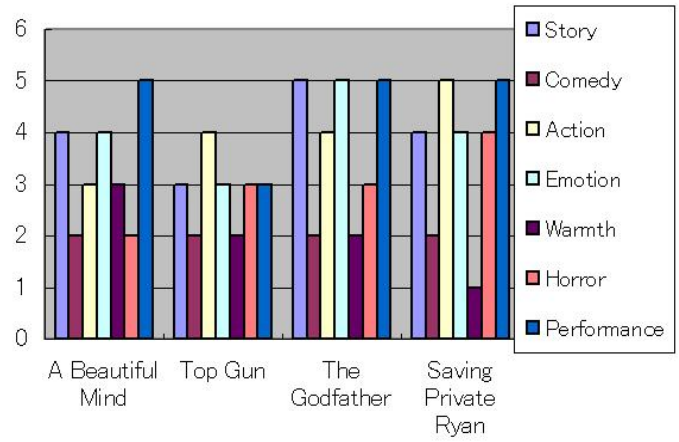

Figure 2. Contents preference.

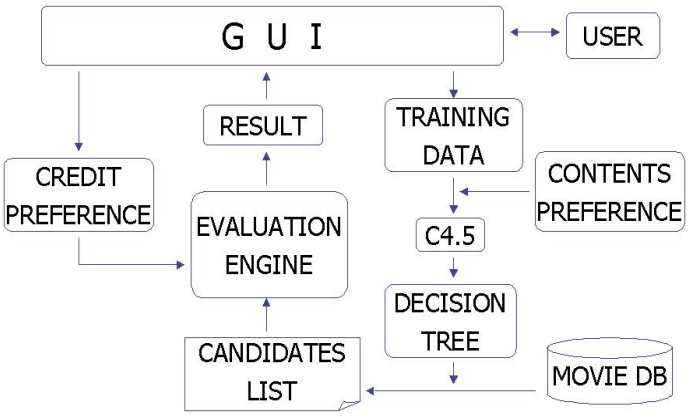

Figure 3. System construction.

Sort the unvalued items by its predicted values to build candidates list.

(5)Result presentation

Sort candidates list and present it to users through GUI.

(6)Training data addition

If a recommended item is evaluated by the user, add this item to the training data. It takes part in the construction of the decision tree in the next recommendation.

C. *ADVANTAGES OF INDUCTIVE-LEARNING-BASED RECOMMENDATION*

Inductive-learning-based recommendation technology is available for the solution of the following problems.

●SPARSITY

Instead of computing user-user or item-item similarities, inductive-learning-based recommendation technology looks into the contents preference of each item. Contents preferences are set up in advance by users or administrators and are shared among users. Data will not get sparse because any item which has a contents preference can obtain its predicted value through decision trees.

●SCALABILITY

In C4.5 learning algorithms, the computation cost depends on the number of training data. If ***m*** stands for the number of training data, the computation cost is $O(m^2)$. In practice, the number of items a user evaluates is no more than one hundred, so a decision tree can be generated immediately. In the case of m>1000, we use alternative sampling method[7] to reduce the computation cost.

The computation cost during recommendation process depends on the total amount of available items. If ***n*** stands for the total amount of available items, the computation cost is $O(n)$ because they have a linear relationship. Therefore the computation cost will not become enormous with the growth of the number of users and items.

●TRANSPARENCY

High readability is one of the advantages of decision trees. The relationship between the recommended information and user preferences is clear. Furthermore, the structure of the decision tree itself represents user preferences, so detailed information can be offered.

IV. A MOVE RECOMMENDER SYSTEM BASED ON INDUCTIVE LEARNING

We setup a movie recommender system to inspect the effectiveness of inductive-learning-based recommendation technology. In this section, We give detailed description about this system.

A. *CONTENTS PREFERENCE AND CREDIT PREFERENCE*

We use two types of movie data in this system. They are contents preference and credit preference. Contents preference has 7 attributes, as shown in Figure 2, story, comedy, action, emotion, warmth, horror and performance.

Figure 4. User interface.

Each attribute has an integral value from 1 to 5. Higher value stands for better quality. Contents preference plays an important role in decision tree construction and classification.

Credit preference is the credit information of a movie, as shown in TAB I, such as genre, director and starring. Recommendation with same training data always provides a static result. Specifying credit preference enables dynamic recommendations.

B. SYSTEM CONSTRUCTION AND RECOMMENDATION FLOW

System construction is shown in Figure 3. Through GUI interface, a user evaluates several movies and provides them to C4.5 as training data. C4.5 apply these training data x to construct the decision tree $h_t : x -> y(1 \le t \le 5)$. Use the decision tree to classify unvalued items and build candidates list. Evaluation engine analyzes user specified credit preference, sorts candidates list and presents recommendation results to the user

C. USER INTERFACE

Figure 4 is the user interface of our movie recommender system. A user will come to this page after login. Each page presents 10 items. “Predictions for you” shows the system predicted value. 5 stars is the highest recommendation. “Your Ratings” stands for the evaluated value by the user. If one of the movies on the list has been previously experienced, a user can perform evaluation by selecting a value and “Submit” it to the server. “Movie Information” displays credit information. Users can click “IMDd” for more detailed information. This design is for better usability[8].

Evaluations by the user are saved after logout. The evaluated items take part in the construction of the decision tree in the next recommendation.

D. SORTING CANDIDATES LIST WITH CREDIT PREFERENCE

Credit preference is information such as genre, director and starring. Evaluation engine first checks whether genre is specified or not. If specified, delete items in candidates list

that are not in this genre. Next, collect all items that are directed or cast by the specified directors or stars. Take the average of each attribute $Attribute_k$ of the contents preference $E_k = \frac{\sum_i Attribute_{ki}}{n}$. Use the attribute that gets the highest score $Attribute_{\max}$ to sort items with the same predicted values. Then carry out sorting with the second highest attribute $Attribute_{\max-1}$. For example, if director is specified to "Steven Spielberg", items that have the same predicted value as 5 are sorted by the attribute "story". Then items that have the same predicted value and the same value of the attribute "story" are sorted by the attribute "performance".

V. EVALUATION EXPERIMENT

A. PARTICIPANTS

A total of 20 university students participated in our experiment. *Age range*: 19 to 23 years. *Gender ratio*: 18 males and 2 females. *Technical background*: 14 were students in technology-related fields, the other 6 were studying in non-technical fields.

B. PROCEDURE

(1) Online registration

(2) Evaluate items in order to get recommendations

(3) Review the recommendation list. Items that have been previously experienced should be evaluated

(4) Specify credit preference. Repeat (3)

(5) Experiment finishes when the evaluated items reach a certain number

C. EVALUATION METRICS

Statistical accuracy metrics evaluate the accuracy of a system by comparing the numerical recommendation scores against the actual user ratings. MAE(Mean Absolute Error) between ratings and system predictions is a widely used metric. MAE is a measure of the deviation of recommendations from their true user-specified values. For each ratings-prediction pair $< p_i, q_i >$ this metric treats the absolute error between them equally. The MAE is computed by first summing these absolute errors of the N corresponding ratings-prediction pairs and then computing the average[5]. Formally,

$$MAE = \frac{\sum_{i=1}^{N} | p_i - q_i |}{N}$$

The lower the MAE, the more accurately the recommendation engine predicts user ratings. We use MAE as our choice of evaluation metric to report prediction experiments because it is most commonly used and easiest to interpret directly.

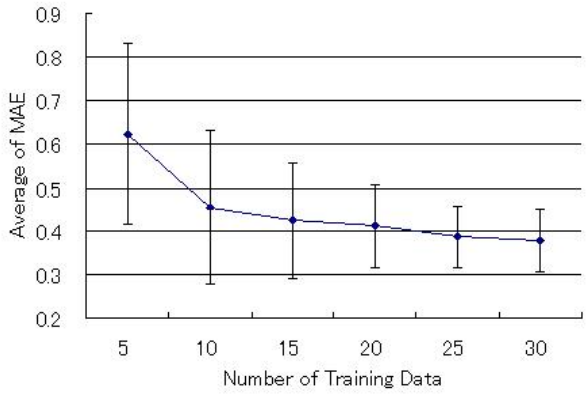

Figure 5. Transition of MAE(1).

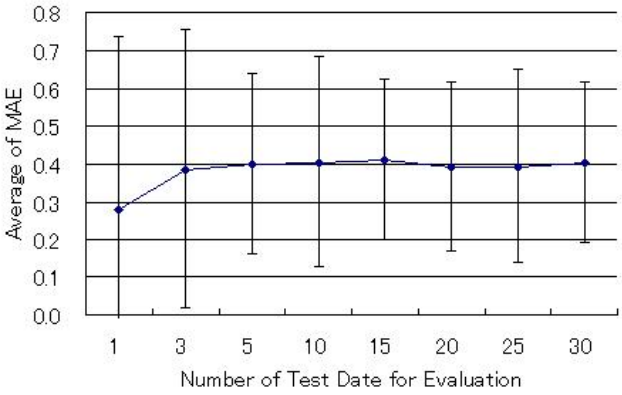

Figure 6. Transition of MAE(2).

D. PILOT STUDY

Before evaluation experiment, we should fix 2 parameters, which affect the experimental result materially. One is the number of training data. It's necessary to investigate the minimum number of the training data required in the decision tree construction process. The other is the number of test data for evaluation. During the evaluation experiment, we should figure out how many items a user should evaluate in order to make efficient evaluation.

Figure 5 shows the transition of the average and the standard deviation of MAE when the number of training data is 5, 10, 15, 20, 25 and 30. 25 times of cross-validation were executed for each number of training data. We can see that the average of MAE drops with the increase of the number of training data. Furthermore, the average of MAE is virtually constant after 10 in spite of the rapid fall between 5 and 10. The average of MAE is 0.455 when the number of training data is 10, which decreases 0.167 compared to 5, but only exceeds 30 by 0.077. Therefore, the efficient decrease of MAE cannot be expected even if we boost the number of training data. The number of training data was fixed to 10 in the later experiments.

Figure 6 shows the transition of the average and the standard deviation of MAE when the number of training data is 10 and the number of test data for evaluation is 1, 3, 5, 10,15, 20, 25 and 30. 25 times of cross-validation were

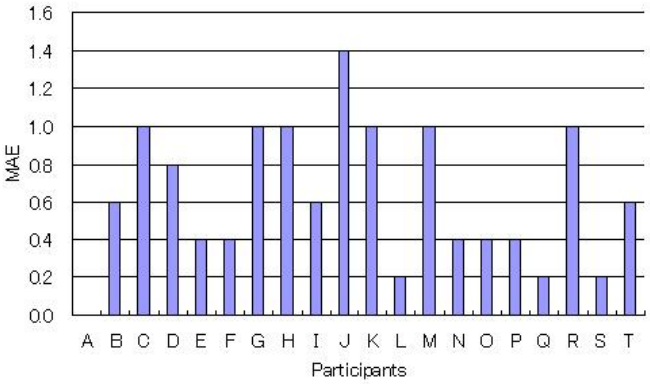

Figure 7. Experimental result.

executed for each number of test data. We can see that the average of MAE is virtually constant with the increase of test data. When the number of test data is 1 and 3, the standard deviation exceeds the average of MAE. Considering the reliability, we fixed the number of test data for evaluation to 5 in the later experiments.

E. EXPERIMENTAL RESULT

Figure 7 shows the result of our experiment. The number of the items a participant should evaluate is 15 in total, of which 10 are training data and 5 are test data for evaluation. The average of MAE is 0.63, and the standard deviation is 0.37.

F. DISCUSSION

In this experiment, we used only 10 training data to moderate MAE under 0.7. MAE was 0.7 to 0.8 in the evaluation experiment by MovieLens [Sarwar,2001][5]. We cannot perform quantitative comparison because the data set and the evaluation metrics are very different. However, an under 1.0 MAE means that most of the time, system predictions agrees user ratings or have the difference of 1 rank. It is almost certain that our system provided high quality recommendations.

We did this experiment online. The execution speed of the server program is 2 to 3 seconds. This is an acceptable speed for online users.

One limitation of our experiment design is that we choose 20 university students as our participants. If the age range stretches and the number of participants grows, we do not have the assurance that we will obtain the same result. More experimental data are needed to improve the reliability.

The experimental result has gaps between some users. We looked into the cases that received high MAE and found that the recommendation accuracy drops if the evaluated value of training data has large bias. During the decision tree classification, 14% of the test data failed to be classified. Increasing the number of attributes of contents preference will lead to a more accurate classification.

VI. CONCLUSION

In this research, we pose a new recommendation technology based on Inductive Learning in the attempt of solving the potential challenges of collaborative filtering based recommender systems. We set up a movie recommender system to inspect the effectiveness. Our experimental result shows that this new recommendation technology is available of solving the Sparsity and Scalability problems, while at the same time providing high quality recommendations.

REFERENCES

[1] P. Resnick and H. R. Varian: "Recommender Systems", Communications of the ACM, vol.40, pp.56-58, 1997

[2] Pennock, D. M., Horvitz, E., Lawrence, S., and Giles, C. L.: "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach", In Proc. of the Sixteenth Conference on Uncertainty in Artificial Intelligence, pp.473-480, 2000

[3] J.S.Breese, D.Heckerman, and C.Kadie: "Empirical Analysis of Predictive Algorithms for Collaborative Filtering", Uncertainty in Artificial Intelligence 14, pp.43–52, 1998

[4] M.Balabanovic and Y.Shoham: "Fab: Content-based, collaborative recommendation.", Communications of the ACM, vol.40, no.3, pp.66-72, 1997

[5] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J.: "Item-Based Collaborative Filtering Recommendation Algorithms", In Proc. of the 10th International World Wide Web Conference (WWW10), Hong Kong, 2001

[6] Ross J. Quinlan. C4.5. "Programs for Machine Learning.", Morgan kaufmann Publishers Inc., San Francisco, California, 1993

[7] N. Abe and H. Mamitsuka: "Query Learning Strategies Using Boosting and Bagging", Proc.of the 15th Int. Conf. on Machine Learning (ICML98), pp:1-9, 1998

[8] Kirstn Swearingen, Rashmi Sinha: "Beyond Algorithms: An HCI Perspective on Recommender Systems.",ACM SIGIR Workshop on Recommender Systems, 2001