Hybrid Recommendation: Combining Content-Based Prediction and Collaborative Filtering

Ekkawut Rojsattarat and Nuanwan Soonthornphisaj

Department of Computer Science
Faculty of Science, Kasetsart University
Bangkok, Thailand
{g4464024, fscinws}@ku.ac.th

Abstract. Recommender systems improve access to relevant products and information by making personalized suggestions based on historical data of user's likes and dislikes. They have become fundamental application in electronic commerce and information access, provide suggestions that effective prune large information spaces so that users are directed toward those item that best meet their needs and preferences. Collaborative filtering and content-based recommending are two fundamental techniques that have been proposed for performing recommendation. Both techniques have their own advantages however they cannot perform well in many situations. To improve performance, various hybrid techniques have been considered. This paper proposes a framework to improve the recommendation performance by combining content-based prediction based on Support Vector Machines and conventional collaborative filtering. The experimental results show that SVMs can improve the performance of the recommender system.

1 Introduction

The original motivation of a recommender system is to solve the information overload problem. The system aims to filter out insignificant materials but provides to the user with more important information. Nowadays, the recommender system also play important role in the business sector. Many e-commerce web sites are already using recommender systems to help their customers find product to purchase. Recommender systems employ historical data on users' preferences to predict items that fit the users. There are two traditional approaches to construct recommender systems: collaborative filtering and content-based recommending. Collaborative Filtering or Social Filtering uses explicit user feedback in the form of rating for item in given domain and utilize similarities and differences among profiles of several users in determining how to recommend an item. On the other hand, Content-Based Recommending learns individualized profile from descriptions of examples and provides recommendation by comparing representations of content contained in an item to representations of content that interest the user. As a result, content-based methods can uniquely make recommendations without having to match their interest to the others.

Content-based recommending and collaborative filtering have been implemented in various domains. Both methods have their own advantages but they cannot perform well in many situations. For example, collaborative filtering cannot provide an efficient recommend if the rating matrix is sparse or have many items that have not been rated by users. Moreover, the content-based recommending lacks of the ability to provide serendipitous recommendations from learned user preference from descriptions of rated item and recommend items that have contents close to user preference. Our hypothesis is that the combination of both techniques should be able to enhance the accuracy of the recommendation.

In this paper, we propose an alternative technique for combining content-based prediction using Support Vector Machines and collaborative filtering based on neighborhood-based algorithm.

2 Background and Related Work

In this section we briefly review some of the previous works related to our work.

2.1 Collaborative Filtering (CF)

CF is the most familiar, most widely implemented and most mature of the recommender system technology. CF aggregates ratings or recommendations of items, recognizes commonalities between users on the basis of their ratings, and generate new recommendations based on inter-user comparison. A variety of collaborative filtering algorithms have designed and deployed. Tapestry [1] is one of earliest implementations of collaborative filtering. GroupLens [2, 21] applied collaborative filtering to email and Usenet news. Ringo [3] and Video Recommender [4] used collaborative filtering to recommend music and movies. Herlocker, et al. presents an analysis of exist collaborative filtering algorithms [19].

The goal of the collaborative filtering is to predict the users' preference, referred to as the active user, based on the preference of a group of users. Collaborative filtering works by collecting human judgment (known as rating) for items in a given domain and matching with people who share the same information need or the same tastes. The problem space of the collaborative filtering can be formulated as a matrix of users versus items, with each cell representing a user's rating on specific item. Under this formulation, the problem is to predict the values of the specific empty cell.

2.2 Neighborhood-Based Algorithm

In neighborhood-based algorithm, a subset of users is chosen based on their similarities to the active user, and a weighted aggregate of their ratings is used to generate predictions for the active user. The neighborhood-based algorithm consists of three steps.

Step 1 Weight all users with respect to similarity of the active user. The similarities between users are measured using the Pearson correlation between their rating vectors as shown in equation 1.

$$P_{a,u} = \frac{\sum_{i=1}^{m}\left(r_{a,i} - \bar{r}_a\right)\times\left(r_{u,i} - \bar{r}_u\right)}{\sqrt{\sum_{i=1}^{m}\left(r_{a,i} - \bar{r}_a\right)^2 \times \sum_{i=1}^{m}\left(r_{u,i} - \bar{r}_u\right)^2}} \qquad \textbf{(1)}$$

where $r_{a,i}$ is the rating given to items *i* by user *a*; and $\bar{r}_a$ is the mean rating given by user *a*; and *m* is the total number of items.

Step 2 Select *n* users that have the highest similarity to the active user. These users form the neighborhood.

Step 3 Compute a prediction from a weight combination of the selected neighbor's ratings. (See equation 2)

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}\left(r_{u,i} - \bar{r}_u\right)\times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}} \qquad \textbf{(2)}$$

where $p_{a,i}$ is the prediction for the active user *a* for item *i*; $P_{a,u}$ is the similarity between active users *a* and *u*; and *n* is the number of user in neighborhood.

Neighborhood-based algorithms have been used successfully to build recommender system in various domains. However they suffered from two fundamental problems.

The Sparse Matrix Problem is a problem that most users do not rate most items and hence the rate matrix is typically very sparse. Therefore, the probability of finding a set of users with significant similar rating is usually low. This is often the case when systems have very high item-to-user ratio this problem is highly affected when the system is in the initial stage of use.

Another problem of neighborhood-based algorithm is *The First-rater Problem*. In the neighborhood-based technique, the recommendation scores of each item are the weighted combination of the neighbor's rating. An item cannot be recommended unless a user has rated it before. This problem will occur when we apply new item into the system.

2.3 Content-Based Recommending (CB)

CB is an outgrowth and continuation of information filtering research [5]. In the content-based system, the objects of interest are defined by their associated features. For example, the newsgroup filtering system NewsWeeder [6] uses the words of their text as features. CB learns a profile of the user's interests based on the features that present in items the user has rated. For that reason, it can make recommendations without having to match their interest to someone else's.

CB has been applied to build recommender systems in various domains. LIBRA [11] is content-based book recommender system which applies automated text categorization method, naive Bayes classifier, by using a database of book information extracted from web pages at Amazon.com. Syskill&Webert [10] suggest web pages that might interest the user.

The drawback of CB is lacking of “cross-genre” or “outside the box” recommendation ability from learned user preference from descriptions of rated item and recommended items that have contents close to user preference.

2.4 Support Vector Machines (SVMs)

Support Vector Machines is a new binary classification technique which based on the principle of the structural risk minimization [7]. The approach aims to minimize the upper bound of the generalization error through maximizing the margin between the separating hyperplane and data.

In the basic form, SVMs learn linear decision rules $h(\mathrm{x}) = sign\{\mathrm{w}\cdot\mathrm{x} + \mathrm{b}\}$ described by a weight vector w and a threshold b. Input is a sample of n training example $S_n=((\mathrm{x}_1,\mathrm{y}_1),\ldots,(\mathrm{x}_n,\mathrm{y}_n))$, $\mathrm{x}_i \in \Re^N$, $\mathrm{y}_i \in \{-1,+1\}$. For linearly separable S_n, SVMs find the optimal margin hyperplane with maximum Euclidian distance to the closet training examples by solving an optimization problem [15].

$$\begin{aligned}&\text{minimize} \quad 1/2\,\|\mathrm{w}\|^2\\&\text{subject to the constraints: (1)} \quad y_i(\mathrm{w}\cdot\mathrm{x}_i + b) \geq 1 \quad, i = 1,\ldots,l\end{aligned} \qquad \mathbf{(3)}$$

The main problem with the optimal margin hyperplane is that it always produces perfectly a consistent hypothesis, which is a hypothesis with no training error. Now we introduce the soft margin hyperplane which allow us to construct the maximum margin hyperplane with some training error. By adding a penalty term (the sum of deviations ξ_i) to the minimization problem [13].

$$\begin{aligned}&\text{minimize} \quad 1/2\,\|\mathrm{w}\|^2 + C\textstyle\sum_{i=1}^{l}\xi_i\\&\text{subject to the constraints: (1)} \quad y_i(\mathrm{w}\cdot\mathrm{x}_i + b) \geq 1-\xi_i \quad, i = 1,\ldots,l\\&\qquad\qquad\qquad\qquad\qquad\quad \text{(2)} \quad \xi_i \geq 0,\ \forall_i.\end{aligned} \qquad \mathbf{(4)}$$

where C is the weight of penalty term for misclassifying training examples. Selecting large value of C spend more computation time since it requires more exhaustive search to minimize the number of misclassified examples.

For a multi-class classification problem, SVMs treats the problem as a collection of binary classification problems. In this subsection, we discuss two multi-class classification approaches for SVMs.

The one-against-the-rest approach is approach works by constructing a set of k binary classifiers. The i^{th} classifier is trained with all of the examples in the i^{th} class with positive labels, and all other example with negative labels. The final output is the class that corresponds to the classifier with the highest output value [15].

The one-against-one approach constructs all possible two-class classifiers from a training set of k classes. Each classifier is trained on only two out of k classes. Thus, there will be $k(k$-1)/2 classifiers. Thubthong and Kijsirikul employs two algorithms, Max Wins algorithm and Decision Directed Acyclic Graph, to evaluate the final output

Max Wins algorithm. A test set is classified by all classifiers each classifier provides one vote for its preferred class and the majority vote is used to make the final output [16]. However, if there is more than one class giving the highest score, a class will be randomly select as the final output.

Decision Directed Acyclic Graph [14] uses a rooted binary direct acyclic graph with k leaves labeled by the classes where each of the $k(k\text{-}1)/2$ internal nodes is labeled with an element of Boolean function. The nodes are arranged in triangle with the single root node at the top, two nodes in the second layer and so on until the final layer of k leaves. The i^{th} node in layer $j<k$ is connected to the i^{th} and $(i+1)^{th}$ nodes in the $(j+1)^{th}$ layer. To evaluate a Decision Directed Acyclic Graph, start at the root node, the binary function at a node is evaluated. The node is then exited via the left edge, if the binary function is -1; or the right edge, if the binary function is 1. The next node's binary function is then evaluated. The value of the decision function is the value associated with the final leaf node.

3 Our Approach

In our approach, we try to improve the performance of the recommendation process by unifying the content-based recommending and the collaborative filtering. In pure collaborative filtering, recommendation scores are the combination of ratings that given by n users who are chosen based on the Pearson correlation [19]. The sparsity of rating matrix becomes the difficulty in finding the way to choose n users in neighborhood. However, we can apply a useful property of content-based methods to defeat this problem. Accordingly, in our hybrid technique, we try to predict the rating of items that has not rated by the user using Support Vector Machines and perform neighborhood-based collaborative filtering on the dense rating matrix.

4 Experiment

4.1 Data Collection and Data Preprocessing

We utilize the EachMovie data set which provided by the GroupLens Research Project at the University of Minnesota as the data set. It consists of 943 users, 1682 movies in 18 genres (movies can be in several genres at once.) and 100,000 ratings (1-5). For each user has rated at least 20 movies. The content for each movie will be collected from the Internet Movie Database (IMDb). By crawling follow the IMDb hyperlinks provided for every movie in EachMovie data set and collecting information from various links of the main URL. We download contents such as director, cast, user comment, tagline, award, plot key-word and plot summary. For the preprocessing step, we eliminated stop words and other non-informative parts and modify several forms of name to unique token form (First name_Last name).

The TF-IDF [17] is applied to weight each term in feature vector. Finally, we put them into a vector of words (one vector for each movie)

4.2 Content-Based Prediction

We treat user ratings 1-5 as one of five class labels and utilize the content of the movies in each class as training data. We apply both naive Bayes text classifier [12] and SVMs to construct a pseudo user-ratings vector for every user. The pseudo user-ratings vector, v_u consists of the item rating provided by user *u*, where available, and those predicted by the content-based predictor otherwise. The pseudo user-rating of all user put together give the dense pseudo rating matrix

4.3 Similarity Weighting

We now perform the collaborative filtering using dense rating matrix from the content-based prediction. The similarity between an active user, *a*, and another user, *u*, is computed using the Pearson correlation coefficient. Since the significant of similarities depends on the accuracy of content-based prediction, thus we multiply the similarities between the active user *a* and another *u* by hybrid correlation weight $hw_{a,u}$ [18].

$$hw_{a,u} = hm_{a,u} + sg_{a,u} \tag{5}$$

The $sg_{a,u}$ is the significant weighting factor. If two users have less than 50 co-rated items, $sg_{a,u}$ is $n/50$, where n is the number of co-rated items. If the number of overlapping items is greater than 50, then $sg_{a,u}$ is 1. $hw_{a,u}$ is the harmonic mean weighting factor computed by Equation 6.

$$hm_{a,u} = \frac{2m_a m_u}{m_a + m_u} \quad , \qquad m = \begin{cases} n/50 : \text{if } n < 50 \\ 1 \ \ : \text{otherwise} \end{cases} \tag{6}$$

4.4 Producing Prediction

The prediction of the combination is computed as follows:

$$p_{a,i} = \bar{v}_a + \frac{sw_a(c_{a,i} - \bar{v}_a) + \sum_{u=1}^{n} hw_{a,u} P_{a,u}(v_{u,i} - \bar{v}_u)}{sw_a + \sum_{u=1}^{n} hw_{a,u} P_{a,u}} \tag{7}$$

In Equation 9, $c_{a,i}$ corresponds to the content-based prediction for the active user, *a*, and item *i*. $v_{u,i}$ is the pseudo rating for user *u* and item *i* and sw_a is self weighting factor for content-based prediction in the final prediction.

5 Results

We compare the performance of our approach to hybrid technique using naive Bayes Classifier, pure content-based methods and pure collaborative filtering. MAE (Mean Absolute Error) and ROC (Receiver Operating Characteristic) [19] are applied for the evaluation metrics. After we ran the experiment five times using 5-fold cross validation [12], the results of the experiment are summarized in table 1.

Table 1. Summary of experimental results

Algorithm	MAE	ROC-4
Pure content-based using NB	0.9113	0.6279
Pure content-based using SVMs	0.8840	0.6448
Pure collaborative filtering	0.9236	0.6517
Hybrid approach using NB	0.8604	0.6782
Hybrid approach using SVMs	0.8312	0.6947

The results show that our hybrid approach present better performances than the other algorithms on both metrics. On the MAE metric, our approach performs 6.6% better than pure content-based using NB, 3.7% better than pure content-based using SVMs, 7.8% better than pure CF, 2.2% better than hybrid approach using NB.

On the ROC-4 metric, our approach performs 10.6% better than pure content-based using NB, 7.7% better than pure content-based using SVMs, 6.6% better than pure CF, 2.4% better than hybrid approach using NB. This means that our approach has higher ability to recommend high quality items than other algorithms.

6 Conclusion and Future Work

We found that the combination of content-based prediction and collaborative filtering obtains better performance than either content-based methods or collaborative filtering. The content-based prediction part of our approach can solve the sparse matrix and the first-rater problem of pure CF by predicted ratings of unrated items using prior knowledge about user preference from rated items. Moreover, hybrid technique also has "cross genre" recommendation ability which is the constraint of pure CB. Since the performance of the hybrid approach usually depends on the accuracy of the content-based prediction, the experimental results show that Support Vector Machines contributes to a better performance than the naive Bayes classifier in content-based prediction tasks. This leads to the performance enhancement of the hybrid recommendation.

Although, our hybrid approach improves the performance of recommendation, but the difference is not very large. We are currently attempting to improve the performance of hybrid technique by applying Transductive Support Vector Machines [20], to enhance the accuracy of content-based prediction. Besides, we also plan to test the performance of hybrid approach with other collaborative filtering algorithms in the future.

References

1. Goldberg, D., Nichols, D., Oki, B. M., Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM 35(12). (1992) 61-70
2. Resnick, P., Iacovou N., Suchak M., Bergstorm P., Riedl J.: GroupLens: An open architecture for collaborative filtering of netnews. In Proceedings of 1994 Conference on Computer Supported Collaborative Work. (1994) 175-186
3. Shardanand, U., Maes P.: Social information filtering: Algorithms for automating "Word of Mouth". In Proceeding of ACM CHI'95. (1995) 210-217
4. Hill, W., Stead, L., Rosenstein, M., Riedl, J.: Recommending and Evaluating Choice in a Virtual Community of Use. In Proceedings of CHI'95. (1995) 194-201
5. Belkin, N. J., Croft, W. B.: Information Filtering and Information Retrieval: Two Sides of the Same Coin? Communications of ACM 35(12). (1992) 29-38
6. Lang, K.: Newsweeder: Learning to Filter News. In Proceeding of the 12th International Conference on Machine Learning, Lake Tohoe, CA, (1995) 331-339.
7. Burges, C.: A tutorial on Support Vector Machines for pattern recognition. Data Mining and Knowledge Discovery 2(2) (1998) 121-167
8. EachMovie dataset. http://research.compaq.com/SRC/eachmovie.
9. Internet Movie Database. http://www.imdb.com.
10. Pazzani M., Muramatsu J., Billsus D.: Syskill & Webert: Identifying interesting web sites. In Proceedings of the Thirteenth National Conference on Artificial Intelligence. (1996) 54-61
11. Mooney, R. J., Roy L.: Content-based book recommending using learning for text categorization. In Proceedings of the Fifth ACM Conference on Digital Libraries. (2000) 195-204
12. Mitchell, T. M.: Machine Learning. Mc Graw Hill. (1997)
13. Cristianini, N., Taylor J.S.: An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press. (2000)
14. Thubthong, N., Kijsirikul B.: Support Vector Machines for Thai phoneme recognition. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. (2001)
15. Vapnik, V.: Statistical Learning Theory. Wiley. (1998)
16. Friedman, J. H.: Another approach to polychotomous classification. Technical report, Department of Statistics, Standford. (1996)
17. Salton, G.: Automatic text processing: The transformation, analysis, and retrieval of information by Computer. Wesley. (1989)
18. Melville, P., Mooney R. J., Nagarajan R. Content-boosted collaborative filtering. In Proceeding of the SIGIR-2001 Workshop on Recommender Systems. (2001)
19. Herlocker, J., Konstan J., Borchers A., Riedl J.: An algorithmic framework for performing collaborative filtering. In Proceedings of the 1999 Conference on Research and Development in Information Retrieval. (1999)
20. Joachims, T. Transductive inference for text classification using Support Vector Machines. In Proceeding of 16th International Conference on Machine Learning. (1999)
21. Konstan, J. A., Miller B. N., Maltz D., Herlocker J. L., Gordon L. R., Riedl J.: GroupLens: Applying collaborative filtering to Usenet news. Communication of the ACM 40(3). (1997) 77-87