How do I Know What You Like?
The Ways of Collaborative Filtering

Bottyán Németh

Department of Telecommunication and Media Informatics
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics
Műegyetem rkp. 3-9, H-1111 Budapest, Hungary
bottyan@gmail.com

Abstract: Recommendation systems are more and more needed because of the huge amount of information available on the internet. The method of making automatic predictions about the interests of a user by collecting taste information from many users is called collaborative filtering. The underlying assumption of collaborative filtering is that: If there are people with similar preferences they would rate analogous things similar. There are many different methodologies to predict an unknown user rating based on other user ratings. In this article I give an overview about the various approaches. Memory-based collaborative filtering algorithms predict the rating of an item based on the ratings of users who are similar to the selected user. It's possible to describe the problem as finding a low-rank approximation to a partial observed rating matrix. And there are other researchers who were using neural networks or other additional data to overcome the above mentioned challenge.

1 Introduction

Content-based filtering (CBF) and collaborative filtering (CF) are two technologies used in recommender systems. CBF systems analyze the contents of a set of items together with the ratings provided by individual users to infer which non-rated items might be of interest for a specific user. Examples include [1], [2], [3]. In contrast, collaborative filtering methods typically accumulate a database of item ratings cast by a large set of users, and then use those ratings to predict a query user's preferences for unseen items. Collaborative filtering does not rely on the content descriptions of items, but purely depends on preferences expressed by a set of users. These preferences can either be expressed explicitly by numeric ratings, or can be indicated implicitly by user behaviors, such as clicking on a hyperlink, purchasing a book or reading a particular news article.

One major difficulty in designing CBF systems lies in the problem of formalizing human perception and preferences. Why one user likes or dislikes a joke, or prefers one CD over another is virtually impossible to formalize. Similarly it is difficult to derive features which represent the difference between an average news article and one of high quality. CF provides a powerful way to overcome these difficulties. The information on personal preferences, tastes, and quality are all carried in (explicit or implicit) user ratings.

CF-based recommender systems have successfully been applied in areas ranging from e-commerce (for example, Amazon and CDnow1) to computer-supported collaborative work [4]. CF research projects include Grouplens (the first automatic CF algorithm, [5]), Ringo [6], Video Recommender [7], Movielens [8], and Jester [9].

In this paper we focus in collaborative filtering techniques. We introduce different classes of methods, and describe some concrete implementation. Regrettably there isn't test results aren't comparable because the different test data and test conditions. For that reason we are lack of global comparison, what should be made in the future. By the rating of these methods, beside the accuracy the scalability has an important role because there are very large databases available with hundreds of million ratings.

In the rest we will get known different kind of probabilistic methods and methods using dimensionality reduction. In general, most collaborative filtering approaches assume that users with similar 'tastes' would rate items similarly, and the idea of clustering has been exploited in all approaches either explicitly or implicitly. Compared with memory-based approaches, model-based approaches provide a more principled way of performing clustering, and is also often much more efficient in terms of the computation cost at the prediction time. The basic idea of a model-based approach is to cluster items and/or training users into classes explicitly and predict ratings of a test user using the ratings of classes that fit in well with the test user and/or items. Several different probabilistic models have been proposed and studied in the previous work. These models have succeeded in capturing user/item similarities through probabilistic clustering in one way or the other, and have all been shown to be quite promising [10].

Other common approach is to fit a factor model to the data, and use it in order to make further predictions. The premise behind a low-dimensional factor model is that there is only a small number of factors influencing the preferences, and that a user's preference vector is determined by how each factor applies to that user. In a linear factor model, each factor is a preference vector, and a user's preferences correspond to a linear combination of these factor vectors, with user-specific coefficients. Thus, for n users and d items, the preferences according to a k-factor model are given by the product of an $n \times k$ coefficient matrix $\mathbf{U}$ (each row representing the extent to which each factor is used) and a $k \times d$ factor matrix $\mathbf{V}'$ whose rows are the factors. The preferencematrices which admit such a factorization are matrices of rank at most k. Thus, training such a linear factor model amounts to approximating the observed preferences **Y** with a low-rank matrix **X** [11].

Finally we describe an algorithm, which use artificial neural network to approximate the user rating function. The learning algorithm is paired with feature extraction techniques to deal with high dimensional, sparse data [12].

2 Probabilistic-based Methods

2.1 Memory-based Probabilistic Method

This approach focuses on scalability and accommodation to new data. They used a density model for the description of user preferences. A generative probabilistic model is used in which the ratings of an actives user are generated based on a probability density on a selected user space. The user space is carefully selected to reach the same accuracy but lower computational requirements. The user space is build by the help of an efficient greedy algorithm which trays to minimize the distance between the set of selected users and the whole user space. The distance measure is the Kullback-Leibler divergence.

This method is significant more accurate than other methods using Naïve Bayes or Pearson correlation, and it's computational efficient. [13]

2.2 Probabilistic Models for Collaborative Filtering

In general, in order to model the similarity among different users, items and ratings given the difficulty of sparse ratings provided by users, we need to cluster each component into groups and model the interactions between different components appropriately. More specifically, the following three important issues must be addressed [10]:

1 *How should we model user similarity and item similarity?* Generally, we may regard users and items as being from different types of entities and they couple with each other through rating information. Therefore, a good clustering model for collaborative filtering is expected to explicitly model both the classes of users and the classes of items and be able to leverage their correlations.

2 *Should a user or an item be allowed to belong to multiple clusters?* Since a user can have diverse interests and an item may have multiple aspects,

intuitively, it is desirable to allow both items and users to be in multiple classes simultaneously.

3 *How can we capture the variances in rating patterns among the users with similar interest of items?* One common deficiency in most existing models for collaborative filtering is that they are all based on the assumption that users with similar interests would rate items similarly. This is incorrect because the rating pattern of a user is determined not only by his/her interests but also by the rating strategy/habit.

2.2.1 Bayesian Clustering (BC)

In Bayesian Clustering, we assume that the same type of users would rate items similarly, thus users can be automatically grouped together into a set of user clusters, or user classes, according to their ratings of items. This model assigns each user to a single user class. The probability of the item ratings in the classes can be learned automatically using the Expectation-Maximization algorithm. Bayesian Clustering appears to be the simplest mixture model: only cluster the users; each user is assumed to be of a single cluster; no separation for preference and rating patterns.

2.2.2 Aspect Model (AM)

The aspect model is a probabilistic latent space model, which models individual preferences as a convex combination of preference factors. Intuitively, this model means that the preference pattern of a user is modeled by a combination of typical preference patterns. Unlike the Bayesian Clustering algorithm, which only models ratings, the aspect model is able to model both users and items with conditional probabilities. The aspect model is still a preliminary model: a simple way to model users and items but without clustering them separately; allowing each user and item to be in multiple clusters; no attempt for modeling intrinsic preference of users separately from their rating patterns.

2.2.3 Joint Mixture Model (JMM) and Flexible Mixture Model (FMM)

In this section, we examine two models for collaborative filtering, namely Joint Mixture Model (JMM) and Flexible Mixture Model (FMM). They differ from both the Bayesian Clustering algorithm and the Aspect Model in that users and items are clustered separately. For both models, the goal is to model the joint probability $P(\{R_y(x)\}_{x \in X(y)} \mid y)$ where R*y*(*x*) is the rating of item *y* by user *x*.

Both models apply separate clustering to users and items and thus satisfy the property 1. The Flexible Mixture Model satisfies the second property since it leaves each rated item the freedom to choose the appropriate user class while the Joint Mixture Model does not. Neither of the two models makes any attempt to explicitly model the difference between the rating patterns and the intrinsic preference of users.

2.2.4 Decoupled Models for Rating Patterns and Intrinsic Preference (DM)

All mixture models that have been discussed so far fail to explicitly account for the fact that users with similar interests may have very different rating patterns. In this section, we discuss decoupled model (DM), which extends the Flexible Mixture Model by introducing two hidden variables ZP and ZR that account for rating patterns and intrinsic preference of users, respectively.

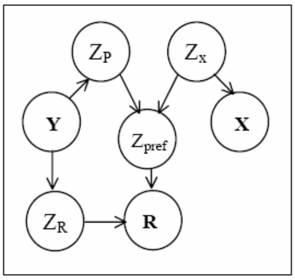

Figure 1
Graphical model representationfor the decoupled model (DM)

Unlike the previous mixture models where the user type is modeled by a single class variable *Zy*, in the new model, users are clustered from two different perspectives, i.e., the clustering of intrinsic preference by hidden variable ZP and the clustering of rating patterns (or habits) by hidden variable ZR. This new model satisfies all the three desirable properties: cluster users and items separately; allow each user to be in multiple clusters; and model the difference between preference patterns and rating patterns.

2.2.5 Comparsion

Training Users Size	Algorithms	5 Items Given	10 Items Given	20 Items Given
20	PCC	1.26	1.19	1.18
	VS	1.24	1.19	**1.17**
	PD	1.25	1.24	1.23
	AM	1.28	1.24	1.23
	BC	1.46	1.45	1.44
	DM	**1.20**	**1.18**	**1.17**
	FMM	1.23	1.22	1.22
	JMM	1.37	.135	1.34
200	PCC	1.22	1.16	1.13
	VS	1.25	1.24	1.26
	PD	1.19	1.16	1.15
	AM	1.27	1.18	1.14
	BC	1.25	1.22	1.17
	DM	**1.07**	**1.04**	**1.03**
	FMM	1.08	1.05	1.04
	JMM	1.17	1.15	1.14
400	PCC	1.22	1.16	1.13
	VS	1.32	1.33	1.37
	PD	1.18	1.16	1.15
	AM	1.28	1.19	1.16
	BC	1.17	1.15	1.14
	DM	**1.05**	**1.03**	**1.02**
	FMM	1.06	1.04	1.03
	JMM	1.10	1.09	1.09

Figure 2
Mean absolute deviation for eight different models on the 'EachMovie' dataset, including a Pearson Correlation Coefficient approach (PCC), a Vector Similarity approach (VS), a Personality Diagnosis approach (PD), a Aspect Model (AM), a Bayesian Clustering approach (BC), a Decoupled Model (DM), a Flexible Mixture Model (FMM) and a Joint Mixture Model (JMM). A smaller value means a better performance.
The results are compared with memory-based methods, like Pearson Correlation Coefficient and Vector Similarity method. We used a method called Personality diagnosis [14].

2.3 Maximum Margin Matrix Factorization

Finally we examine a method which describes the users not in the dimension of the rated items but in a reduced space. As mentioned in the introduction we approximate the original observation matrix **Y** with a lower rank matrix **X**.

The low-rank matrix X that minimizes the sum-squared distance to a fully observed target matrix Y is given by the leading singular components of Y and can be efficiently found. However, in a collaborative prediction setting, only some of the entries of Y are observed, and the low-rank matrix X minimizing the sum-squared distance to the observed entries can no longer be computed in terms of singular value decomposition. In fact, the problem of finding a low-rank approximation to a partially observed matrix is a difficult non-convex problem with many local minima, for which only local search heuristics are known.

Recently was suggested a formulation termed 'Maximum Margin Matrix Factorization' (MMMF), constraining the norms of **U** and **V** instead of their dimensionality (X = UV'). Viewed as a factor model, this corresponds to constraining the overall "strength" of the factors, rather than their number. That is, a potentially infinite number of factors is allowed, but only a few of them are allowed to be very important. Mathematically, constraining the norms of **U** and **V** corresponds to constraining the trace-norm (sum of singular values) of **X**. Interestingly, this is a convex constraint, and so finding a matrix **X** with a low-norm factorization minimizing any convex loss versus a partially (or fully) observed target matrix **Y**, is a convex optimization problem.

It's possible to perform gradient-based local search on the matrices **U** and **V**. Using such methods, it's possible to find maximum margin matrix factorizations for a realistically sized collaborative prediction data set. [11]

Conclusions

We introduced the main methods used for collaborative filtering. Of course there are other possibilities to solve this problem but these are the most commonly used approaches to deal with it. After studying the different solutions it's hard to say which the best is because it depends hardly on the details of the implementation and on small tricks, but because the high dimensionality of the dataset, and because its sparsity clustering of users and items seems to be profitable. It would be necessary in the future to make an empirical comparison of the above mentioned methods on the same dataset. The accuracy and the computational requirement should be measured as well.

References

[1] M. Balabanovic, Y. Shoham: Fab: Content-based, Collaborative Recommendation, *Communications of the ACM*, Vol. 40, No. 3, pp. 66-72, 1997

[2] R. J. Mooney, L. Roy: Content-based Book Recommending Using Learning for Text Categorization, in Proceedings of the 5th ACM Conference on Digital Libaries, San Antonio, US, 2000, pp. 195-204, ACM Press, New York, US

[3] M. Pazzani, J. Muramastsu, D. Billsus: Syskill and Webert: Identifying Interesting Web Sites, in Proceedings of the 13th National Conference on Artificial Intelligence, Portland, OR, August 1996, pp. 54-61

[4] B. Sarwar, G. Karypis, J. Konstan, J. Riedl: Analysis of Recommendation Algorithms for E-commerce, in Proceedings ACM E-Commerce Conference, 2000, pp. 158-167

[5] P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, J. Riedl: Grouplens: An Open Architecture for Collaborative Filtering of Netnews, in Proceedings of the 1994 Computer Supported Collaborative Work Conference, Chapel Hill, North Carolina, 1994, pp. 175-186, ACM

[6] U. Shardanand, P. Maes: Social Information Filtering Algorithms for Automating 'Word of Mouth', in Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, 1995, Vol. 1, pp. 210-217

[7] W. Hill, L. Stead, M. Rosenstein, G. Furnas: Recommending and Evaluating Choices in a Virtual Community of Use, in Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems, 1995, pp. 194-201

[8] B. J. Dahlen, J. A. Konstan, J. L. Herlocker, J. Riedl: Jump-Starting Movielens: User Benefits of Starting a Collaborative Filtering System with Dead Data, Tech. Rep. 7, University of Minnesota, 1998

[9] K. Goldberg, T. Roeder, D. Gupta, C. Perkins: Eigentaste: A Constant Time Collaborative Filtering Algorithm, Information Retrieval Journal, Vol. 4, No. 2, pp. 133-151, 2001

[10] Rong Jin, Luo Si, Chengxiang Zhai: A Study of Mixture Models for Collaborative Filtering, Journal Of Information Retrieval (In Press)

[11] Jason D. M. Rennie, Nathan Srebro: Fast Maximum Margin Matrix Factorization for Collaborative Prediction, in Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 2005

[12] Daniel Billus, Michael J. Pazzani: Learning Collaborative Information Filters. Proceedings of the International Conference on Machine Learning. Morgan Kaufmann Publishers. Madison, Wisc.

[13] Yu et al.: Probabilistic Memory-based Collaborative Filtering, IEEE Transactions on Knowledge and Data Engineering, Jan. 2004, Volume 16, Issue 1, pp. 56-69

[14] Pennock, D. M., Horvitz, E., Lawrence, S., & Giles, C. L. (2000) Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-based Approach. In the Proceeding of the 16th Conference on Uncertainty in Artificial Intelligence