Collaborative Filtering of Multi-component Rating for Recommender Systems

Abstract

The dependency structure among the rating components is discovered and incorporated into a mixture model and parameters of the model were estimated using Expectation Maximization. The algorithm is evaluated using data collected from Yahoo Movies. Improved recommendations were found using multiple components over using only one component when very little training data is used. However, no gain was found when enough training data were available.

1 Introduction

Collaborative filtering is becoming popular as a method to generate recommendations for users of an online meeting place. The recommendations are generated from the collective user experiences and preferences, yet personalized for each user—hence, the name *collaborative filtering* [7]. Examples of community sites where members use collaborative filtering to find potentially interesting items include Last.fm (a music recommender system) and StumbleUpon (a web page recommender system). Used by a retailer collaborative filtering is a useful tool to target-advertise items to its customers; items that a customer is likely to like, yet was unlikely to have discovered on his own because of the seer number of items available. Thus, a merchant can induce demand for its less known items. For example, Netflix, a movie-rental company, uses collaborative filtering to create demand for older and less known movies by advertising them to users who might like those movies. Other stores who use collaborative filtering to manage demand and enhance user experience include Amazon.com, iTunes, and Tivo.

There is a large research literature on collaborative filtering from a number of perspectives [1]. The goal of a collaborative filtering based recommender system is: given a user's ratings on a subset of items and its peers' ratings on possibly different subsets of items, to predict among the items that the user has not yet rated which ones he would rate highly. The algorithm treats two users as similar if they rate common items similarly. Two items are considered similar if they have received similar ratings from users who have rated them both. The idea is to make a recommendation from the items liked by a user's similar users. This can be thought of as automating the spread of information through word-of-mouth [8].

One approach in collaborative filtering is to learn a model that explains how the ratings are generated and then using this model to make predictions about the users. Notable among the model based collaborative filtering works is the Flexible Mixture Model (FMM) where two latent variables are used to model the user behavior and item characteristics separately [9]. This leads to improved performance over the models, such as Aspect Model, where single latent variable is used to characterize the user and the item distribution[5]. Often the model based algorithms are designed using Bayesian Networks. This framework allows us to systematically incorporate our intuition about the rating generation process in the model.

Most of the current literature discuss methods to use ratings with only one component. However, when we have ratings with multiple components spanning more than one aspect of an item, we have more information about the user's preferences and an opportunity to generate more accurate recommendations. Recently there has been a growing interest in the research community in effective use of such multi-component ratings [1]. Recently Yahoo Movies has started collecting ratings along multiple aspects of a movie, such as, *story, acting, visuals, direction*. Such developments and the limited amount of work that has been done in integrating multiple components of ratings to generate improved recommendations has been the motivation behind this work.

In this work we have described a model based approach to generate recommendations using a mixture model that uses multiple components of the ratings. Experiments on a movie dataset show that it leads to improved recommendation when we use limited amount of training data. However, we did not find any gain from using multiple components when we use more data for training. We believe that our ability to generate better predictions with small amounts of traiing data meets a real world requirement in the use of collaborative filtering systems.

2 Multi-component rating collaborative filtering

Data This work has been facilitated by the availability of component rating data from Yahoo Movies web-site. Each record of the rating data consists of seven variables: item or movie id (I), user id (U), ratings on story (S), acting (A), visuals (V), direction(D) and overall (O) quality of the movie. The ratings were converted to $0-4$ integers ($A \rightarrow 4$, $B \rightarrow 3$, $C \rightarrow 2$, $D \rightarrow 1$, $F \rightarrow 0$). Ratings from users who have rated more than 20 movies were used so that we shall have enough training and test data for each user. After this filtering there were 45892 records, 1058 unique users and 3430 unique movies.

The intuition We expect that by rating multiple aspects of an item users provide richer information about their preferences. Consider the following example, where two users have given similar low Overall ratings to a movie.

User	Movie	story	acting	visuals	direction	overall
u_1	m_1	4	0	1	0	1
u_2	m_1	1	0	0	4	1

Table 1. An example of multi-component rating (all ratings are out of 5)

The component ratings suggest that the user u_1 might like other movies that have a story similar to m_1, while user u_2 might like a movie that has been directed by the same director or a director with similar style. Hence, if we can effectively use the information in the component ratings provided by the users, we should be able to make more accurate recommendations.

The problem When a user likes a movie, he, in general, rates the components of the movie higher. Therefore, the components will be correlated (see Figure 1). From the correlation matrix it seems that the components vary together and do not give much independent units of information. In fact, a principal component analysis on the correlation matrix shows that the first principal component explains 84.5% of the total variance. This phenomenon of observing a higher than expected correlation between ratings is known as the Halo effect in the psychometric literature. One important reason for Halo is that the users rate components, based on their overall impression [10].

	S	A	V	D	O
S	1.00	0.79	0.82	0.74	0.87
A		1.00	0.81	0.73	0.83
V			1.00	0.79	0.88
D				1.00	0.80
O					1.00

Figure 1. Correlation among rating variables.

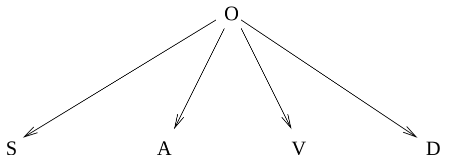

Figure 2. The BayesNet encoding conditional independence

The solution There are two intuitions outlined in last two paragraphs:

1. There is distinguishing information in the variation of components of ratings *even when Overall ratings agree*,
2. The Overall rating might be the inducer of high correlation among components.

This lead us to compute the partial correlation among the component ratings while controlling for the effect of Overall rating. We found that the average inter-component correlation among variables S, A, V, D reduces from 0.78 to 0.26. As all correlations are positive we should expect some reduction in correlation when computing partial correlations. However, the average partial correlation among the variables is the least when we control for the variable O among the possible five variables. The average partial correlations when we controlled for S,A,V,D were 0.47, 0.53, 0.35 and 0.60 respectively. This confirms our intuition that the Overall rating is the highest correlation inducing variable. A similar approach is taken by Holzbach who asserts that global impression rating is the cause of high correlation among the components and ameliorates this by partialing out the effect of global impression rating [6].

The application The observation that controlling for the Overall rating leads to lowest dependence among the component variables leads to the BayesNet shown in Figure 2. It states that the component variables are independent conditional on Overall variable. Note that we do not make an independent assertion among component. Instead we assert that the components are dependent with each other via only the Overall rating. Empirically we shall observe some residual dependence among the rating components (partial correlations are not all zero). This is a limitation of assuming that a single variable can explain the correlation between pairs of components. A model where we allow multiple nodes to explain dependency among pairs of variables would explain more of these dependencies, but, will be more complex. Embedding this BayesNet in the Flexible Mixture Model we get the model shown in Figure 3(b).

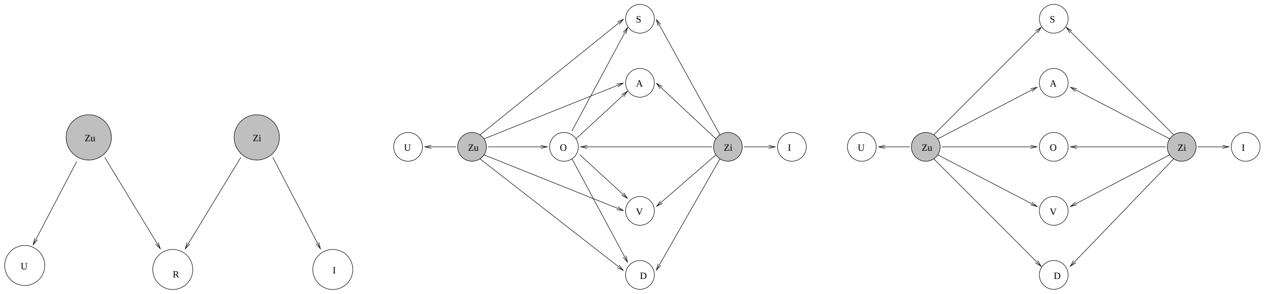

Figure 3. (a) FMM with one rating component by Luo Si and Rong Jin[9], (b) FMM with multiple component with dependency structure, (c) Naive FMM with multiple component without dependency structure.

In a BayesNet diagram a variable is independent of its non-descendents conditional on its parent(s). The original FMM asserts that rating variable R, user variable U, and item variable I are conditionally independent of each other given latent variables Z_u and Z_i. Latent variable Z_u is used to characterize the distribution of U and the latent variable Z_i is used to characterize the distribution of I.

We embed the distribution of five rating components in the FMM BayesNet (Figure 3(b)), while making use of the observed conditional independence among the rating components (Figure 2). This modified model additionally asserts that the component ratings are conditionally independent of each other given the latent variables Z_u, Z_i and the Overall rating. In Figure 3(c) we present the naive model which assumes the components to offer independent units of information.

Learning and prediction The conditional independence assumptions allow us to factorize the joint distribution over all variables as a product of conditional probability tables(CPT). To estimate these CPTs we need to follow some iterative approximation algorithm such as Expectation Maximization [3] because we have two latent variables. After estimating the CPTs to make a prediction about the Overall rating (O) for a user-item pair (U, I), we marginalize away all other variables from the joint distribution and get the joint distribution of these three. Then we can find conditional distribution over O given values of U and I, and make a prediction using this conditional distribution. In the original FMM, mean of this distribution was used to make a prediction. But, we believe that the mode is more appropriate since the ratings are treated as multinomial.

3 Results and discussion

We use a randomly selected fraction of each user's ratings for training and the remaining for testing. Mean Absolute Error (MAE) of the predictions were computed to evaluate the suitability of the methods when the task is to predict the future rating accurately. These algorithms were also evaluated for their effectiveness in retrieving the items that the active user has rated highly. This was done using a precision-recall curve, in which fraction of retrieved items that were relevant (precision) was plotted against the fraction of all relevant items that were retrieved (recall) [2].

The MAE was plotted against the fraction of data that was used for training (Figure 4). For each point in the plot 30 random train-test splits were made, keeping the ratio at the given value, and average is taken. The same train-test set was used for all the models. Hence, we got paired readings. We did a pairwise t-test and found that all differences between algorithms are significant.

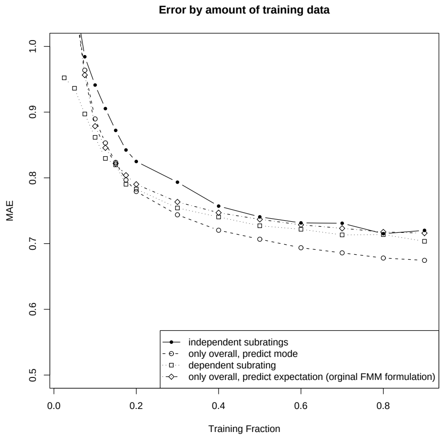

Figure 4. Plot of errors by fraction of data used for training.

We can see from the Figure 4 that naively embedding component ratings as independent variables in FMM give the highest error. This is not surprising since the component ratings are highly correlated. Treating them as independent variables leads to over-counting of the evidence.

Using mode of the distribution to make a prediction leads to fewer error than using the expectation (done in original FMM model).

Comparing the model that uses only Overall rating with the model that uses five component structure with the dependence among them we find that when we use less data for training using multiple components leads to lower error. But, when we use more data for training using only overall component leads to lower error.

The precision-recall curve for the algorithms are plotted in Figure 5.

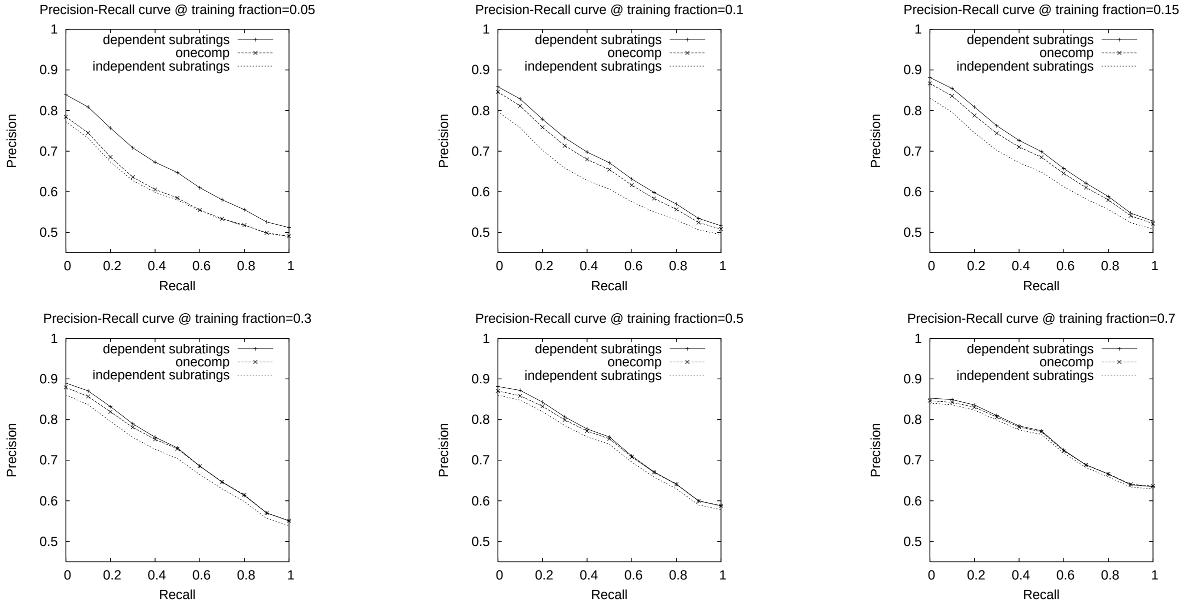

Figure 5. Retrieval performance of algorithm using dependency structure among the components, using only overall rating and algorithm using components as if they were independent.

As we can see when the training fraction is low the difference between the the three algorithms is the most pronounced. The algorithm with discovered structure gives the highest precision at each recall level followed by the method using only the Overall component. The algorithm that assumes independence among the component ratings leads to lowest precision. As we use more and more training data the difference between these algorithms diminishes. The interesting point to note here is that although when using only overall as we use more and more training data we get a lower Mean Absolute Error than using all components, it does not perform better in selecting top-N items. As pointed out in [4] these metrics measure two different aspects of the performance of the algorithms and are often not correlated. One must use the appropriate evaluation metric to measure the suitability of an algorithm for the task at hand.

4 Scope for future work

We have illustrated a way to use multiple components of ratings in a collaborative filtering algorithm. However, further research should be done using a more complex model that can explain the remaining residual dependencies. Another interesting problem would be to examine the stability of ratings of individual rater over time. A formal analysis of usefulness of components of rating in predicting ratings on unknown (user, item) pair would certainly be interesting.

Bibliography

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng*, 17(6):734–749, 2005.

[2] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[4] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[5] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In Dean Thomas, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol2)*, pages 688–693, S.F., July 31–August 6 1999. Morgan Kaufmann Publishers.

[6] R. L. Holzbach. Rater bias in performance ratings: Superior, self, and peer ratings. *Journal of Applied Psychology*, 63:579–588, 1978.

[7] Paul Resnick and Hal R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, 1997.

[8] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating ẅord of mouth.̈ In *CHI*, pages 210–217, 1995.

[9] Luo Si and Rong Jin. Flexible mixture model for collaborative filtering. In *ICML*, pages 704–711. AAAI Press, 2003.

[10] F. L. Wells. A statistical study of literary merit. *Archives of psychology*, 1, 1907.