PERSONAL RECOMMENDATION USING WEIGHTED BIPARTITE GRAPH PROJECTION

MING-SHENG SHANG, YAN FU, DUAN-BIN CHEN

School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu 610054, China
E-MAIL: msshang@uestc.edu.cn

Abstract:

This work is a study of personal recommendation algorithm employing the projection of weighted bipartite consumer-product network. The weight of the edges is directly the rate that a customer giving on a product. Following a network based resource allocation process we get similarities between every pair of consumers, which is then used to produce prediction and recommendation. We show this is also a two step random walk process in the bipartite. Since the weighted graph is more informative, we would expect higher predict accuracy.

Keywords:

Personal recommendation; Graph analysis; Bipartite graph projection; Random walk; Similarity computing

1. Introduction

Recommender systems are widely used techniques that attempt to present information productions, such as movies, music, books, news, images, web pages, that are likely of interest to the consumers. Two known examples are the web merchant Amazon.com and the online movie rental company Netflix. Recommender systems are also likely to provide value for other information overload problems. With their significance in economy and society, recommender systems is now a rich research area, various kinds of recommender systems have been created, and they are still a joint focus in many fields. In [1], Adomavicius et al. provides a recent survey of recommender systems.

One of the earliest and most successful technologies behind recommender systems is collaborative recommendation. The underlying assumption of collaborative recommendation (filtering) approach is that those who agreed in the past tend to agree again in the future. The most common form of collaborative recommendation is the neighborhood-based approach, which works by first computing the similarity between all pairs of users (user-based) or items (item-based), and then predictions are made by aggregating ratings of the target item by the user's neighbors or the user's ratings of items that are neighbors of the target item. Algorithms within these families differ in the definition of similarity, formulation of neighborhoods and the computation of predictions.

Finding a particular consumer's neighborhood with similar taste or interest and quantifying the strength of similarity are the most crucial steps for collaborative filtering. Different techniques have been proposed for this task including various similarity functions, correlation, and clustering etc. Among these techniques, Pearson correlation coefficient is reported as one of the most commonly used technique and gain best performing. Despite their widely used, there remains room to improve accuracy of collaborative filtering. Very recently, network-based inference for recommendation system has attracted a lot of attention in this literature. In [2], algorithm based on a weighted projected projection of bipartite user-object network is proposed, which shows remarkable higher accuracy than the classical collaborative filtering algorithm.

Inspired by [2], in this paper, we study the weighted bipartite projection following a network based resource allocation progress. We will show that the network based resource allocation process is also a two step random walk between two consumers in the bipartite graph. The main difference of this paper and [2] are three aspects: First, we use the rates giving by user on item as the weight of edge, which is more informative than the simple method of collected or not collected relationship between user and item, thus we can expect further higher predict accuracy; Second, this is a weighted resource allocation progress according to customer's interest, not the simple approach that equally distribute customer's recommendation power. Third, we evaluation this proposed approach by a commonly used measure: mean absolute error (MAE).

2. The fundamental of collaborative recommendation

There are two main algorithmic techniques have been proposed in collaborative recommendations: user-based and item-based. The user-based algorithm is the first

collaborative filtering algorithm proposed in the literature. It has been proved by a number of researches that this algorithm can achieve competitive performance with many other algorithms, and often serves as a comparison benchmark for later proposed algorithms. The item-based algorithm has also been shown to outperform the user-based algorithm for many datasets, and it is computational efficiency when customers substantially outnumber the products in the system. Since the difference of user-based approach and the item-based are mathematic equivalent by interchanging the role of user and item, we will only consider the user-based technique in the rest.

Denoting the customer set as $C=\{c_1,c_2,\ldots,c_m\}$ and product set as $P=\{p_1,p_2,\ldots,p_n\}$, a collaborative recommendation system can be fully described by an $m*n$ matrix R, where the (c,p)-th entry of this matrix stands for the customer c's rating on product p, or null, depending on whether the customer c has rated the product p, or not, respectively.

The user-user algorithm can be thought of working in two stages. In the first stage, similarities between every pair of users are computed and stored as a model. There are a variety of formulations for similarity weight calculations; two common used model are Pearson correlation coefficient and vector similarity, both have been demonstrated to have competitive performance with other designs. The Pearson correlation coefficient between two customers, c, and v is measured by equation

$$sim(c,v)=\frac{\sum_{p\in P_c\cap P_v}\left(r_{c,p}-\overline{r_c}\right)\left(r_{v,p}-\overline{r_v}\right)}{\sqrt{\sum_{p\in P_c\cap P_v}\left(r_{c,p}-\overline{r_c}\right)^2}\sqrt{\sum_{p\in P_c\cap P_v}\left(r_{v,p}-\overline{r_v}\right)^2}} \quad (1)$$

where P_c is the set of productions rated by customer c, $r_{c,p}$ is rating of customer c on production p, and $\overline{r_c}$ is the average rating of customer c (similarly for v). The vector similarity differences the Pearson correlation coefficient in that it does not minus the average rating $\overline{r_c}$.

In the next step, a prediction for customer c and product p is computed by a weighted average of the ratings by the neighbors.

$$r_{c,p}=\overline{r_c}+\frac{\sum_{v\in N_c\cap C_p}\left(r_{v,p}-\overline{r_v}\right)\cdot sim(c,v)}{\sum_{v\in N_c\cap C_p} sim(c,v)} \quad (2)$$

where N_c is the set of the top K customers most similar to customer c, C_p is the set of customers who rated product p, so $N_c\cap C_p$ is the set of customers most similar to c who have rated p.

3. Bipartite graph projection based recommendation

3.1. Weighted bipartite graph model

A recommendation system can be presented as a bipartite graph, where the vertices are the customers and the products, a link between a product and a user if and only if this product has been rated by this user. Connections between two users or two products are not allowed. The procedure of graph construct seems intuitive, which has been widely adopted. Model in this paper different from those models is that it is a weighted graph.

In this paper, we adopt a simple way to weight the edge: The weight of the edges is directly the rate that a customer giving on a product. That is, graph $G=(V,E,r)$, where $V=C\cup P$, E is the set of edge between C and P, w is a weighting function $r:E->R^+$, that measure the preference of customer on products (a higher value means more interested in). Here we directly use the ratings as the weight. Table 1 gives and instance of rating matrix, Figure 1(a) is the associated bipartite graph. Obviously, there are many other improvements can be adapted to this method, for instance, a better alternative is to use the *z-scores* instead of explicit ratings. But those extensions are not the main focus of this paper.

3.2. Weighted bipartite graph projection

We are interested in the computation of similarity between customer u and customer v. Our method is motivated by the diffusion process presented by Zhou et al. [2], but here we will use a weighted model of the bipartite graph and a weighted resource allocation process instead of equally allocation customer's interests. Assume that a certain amount of resource (e.g. recommendation power) is associated with each user, and the weight $rp(u,v)$ represents the proportion of the resource customer u would like to distribute to v.

There are two steps resource allocation process. At the first step, each user distributes his initial resource in proportion to his preference to all the products he has rated, each product receive resource from users rated this product. After this step product p gets $r_{u,p}/R_u$ of user u's resource, where $R_u=sum_{q\in P}(r_{u,q})$ is the number of products u rated. Then, at the second step, user u gets resource from all products he has rated, user u gets $r_{u,p}/R_p$ resource from product p, where $R_p=sum_{v\in U}(r_{v,p})$ is the number of customers who rated product p. Thus the weight $rp(u,v)$ can be expressed as

Table 1: The users-items matrix of ratings

	Rating				Recommendation power			Projection-based prediction			
	p_1	p_2	p_3	p_4	c_1	c_2	c_3	p_1	p_2	p_3	p_4
c_1	3	2		1	0.78	0.11	0.11	2.33	1.72	**0.72**	1.2
c_2		1	3		0.17	0.65	0.19	**0.33**	0.87	2.19	**0.61**
c_3			1	2	0.22	0.25	0.53	**0.33**	**0.41**	1.09	1.17

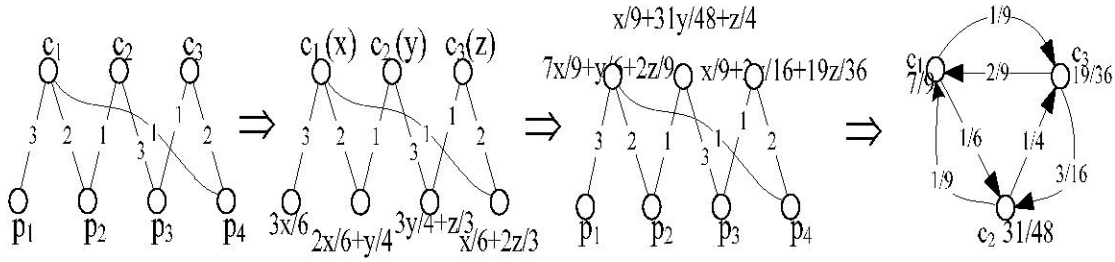

Figure 1. Bipartite graph projection for user-based algorithm.

$$rp(u,v)=\sum_{p\in P}\frac{r_{v,p}}{R_p}\frac{r_{u,p}}{R_u}=\frac{1}{R_u}\sum_{p\in P}\frac{r_{u,p}r_{v,p}}{R_p} \qquad (3)$$

This recommendation power between any two users has some interesting characteristic:

First, it is not necessary symmetric, i.e., *rp(u,v)≠ rp(u,v)*, which is quite distinct from the traditionally similarity based recommendation; the difference between this two value comes from the denominator, which results the lower recommendation power for those customers who rated a lot of products. That makes sense contrast to our realistic thinking that the more products a customer rated the more authoritative that customer is.

Secondly, the sum of each row is 1, we can think of it as a normalized value, which is the distribution of the recommendation power of a user;

Third, the user's recommendation power decreases with the sum of rating he made, which is in accordance with our daily experience: whether the large sum of rating comes from a lot of products showing user likes to rate everything, or comes from high rating of products show user likes to rate everything with high rating, both have little value on recommendation.

It also should be noted that *rp(u,v)* is an extension of results in [2]: if $r_{c,p}$ is binary value, then *rp(u,v)* and w_{ji} have the same expression.

Once the recommendation power between users is computed, we can use it to generate personal recommendation, which is similar to the procedure of using similarity for prediction in standard collaborative recommendation algorithms. Further, if all users' recommendation is considered, since *rp* is normalized with user, the prediction of user *u* on product *p* can be written as

$$r_{c,p}=\frac{\sum_{v\in U}rp(v,c)r_{v,p}}{\sum_{v\in U}rp(v,c)}=\sum_{v\in U}rp(v,c)r_{v,p} \qquad (4)$$

Because of the sparsity of rating matrix, we can simply use the matrix formation, which reads *RP*R*, where *RP=(rp)'* is the transpose of recommendation power matrix, since we want to receive instead of distribution of the power. Recommendation is then made by selecting the largest value of unrated products.

For example, consider the example in Table 1. The left part of Table 1 is the rating matrix The procedure of the projection for user-based recommendation is described in Figure 1. Item-based approach is antithesis thus omitted. The mid part of table 1 is the RP matrix between users, and the prediction is in the right part.

3.3. Random walk explanation of the projection

A random walk is a natural stochastic process on graphs. Given a graph and a start node, we select a neighbor of the node at random, and 'go there', after which we continue the random walk from the newly chosen node. The probability of a transition from node *i* to node *j*, is w_{ij}/d_i, where w_{ij} is the weight of edge <i,j>, $d_i=sum_j(w_{ij})$ is the *weighted degree* of node *i*. Given a weighted graph *G(V,E,w)*, the associated transition matrix is a matrix in which, if *i* and *j* are connected, the *(i,j)*'th entry is simply probability of a transition from node *i* to node *j*.

In the case of our collaborative recommendation task, it is easy to see that the probability of a transition from customer *u* to product *p* is $r_{u,p}/R_u$, where $R_u=sum_{q\in P}(r_{u,q})$ is the number of products *u* rated, i.e., the degree of node *u* in graph. Similarly, the second step of resource allocation

from product p to customer is the probability of a transition from product p to customer v is $r_{v,p}/R_p$, where $R_p=sum_{v \in U}(r_{v,p})$ is the number of customers who rated product p, i.e., the degree of node p in weighted bipartite graph. Combine this two steps, we can easily obtain equation (4). Thus, the network based resource process is equivalent to a two steps of random walk from customer to customer in the weighted bipartite graph.

This equivalence gives us lots of opportunities to further understand collaborative recommendation since it was always verified empirically, and the random walk in graph is a well studied field in mathematics. For example, the two-step of random walk can obviously be extended to multi-step of random walk to reduce the sparse problem.

4. Experiment

In order to measure the quality of an algorithm we use a benchmark data set, namely MovieLens[3] , which consists of 943 users, 1682 movies, and 10^5 discrete ratings from 1 to 5. As most studies, we divide this data set into two parts: one is the training set (containing 90% of the data) that is treated as known information, and the other one is the probe (the remaining 10%), whose information is not allowed to be used for prediction. Then we make a prediction for every entry contained in the probe and measure the difference between the predicted rating and the actual rating.

A lot of metrics, including coverage, accuracy have been proposed in the past research and been evaluated in [5]. Coverage is a measure of the percentage of items for which a recommendation system can provide predictions such hit-rate or rank score; accuracy is a measure of recommendation system by comparing the numerical prediction values against user ratings for the items that having both predictions and ratings. In this paper, we use root mean square error (RMSE), mean absolute error (MAE) and ranked evaluation defined in [4].

Table 2. Performance comparison of different collaborative filtering strategies (for MAE and RMSE, lower value is better, while for ranked evaluation higher value is better)

	Bipartite projection	Pearson correlation	vector similarity
MAE	**0.8213**	0.8335	0.9003
RMSE	**1.0724**	1.0873	1.2544
ranked evaluation	3.9588	**3.9942**	3.9117

The results are presented in Table 2 with average of 5 runs. As shown in Table 2, the proposed weighted bipartite graph projection is much better than the other two similarity measures in the case of prediction accuracy. While for ranked evaluation measure which measures the expected true preference of the chosen item when the probability to choose a recommended item decays exponentially with its location in a sorted list of recommendations (k=10), it is slight lower performance.

5. Conclusions

We have presented a personal recommendation algorithm employing the projection of weighted bipartite consumer-product network, which outperformance the commonly used similarity measures such as Pearson correlation coefficient and vector similarity in prediction a accuracy.

Acknowledgements

This project was supported by China Postdoctoral Science Foundation (20080431273) and the 863 Project (2006AA01Z414)、863 program (2007AA01Z440)、Sichuan province science and technology research project(2008GZ0009).

References

[1] Adomavicius, G. and A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 2005. 17(6): p. 734-749.

[2] Zhou, T., et al., Bipartite network projection and personal recommendation. Physical Review E, 2007. 76(046115).

[3] http://www.grouplens.org/

[4] Breese J. S., Heckerman D. and Kadie C., Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Uncertainty in Artificial Intelligence, 1998

[5] Herlocker, J.L., et al., Evaluating collaborative filtering recomm`ender systems. ACM Transactions on Information Systems, 2004. 22(1): p. 5-53.

[6] Francois, F. and P. Alain, et al.. Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation." IEEE Transactions on Knowledge and Data Engineering, 2007, 19 (3): 355-369.

[7] Perugini, S. and M. A. Goncalves, et al.. Recommender systems research: a connection-centric

survey.Journal of Intelligent Information Systems,2004, 23 (2): 107-143.