An Enhanced Semi-supervised Recommendation Model Based on Green's Function

Dingyan Wang and Irwin King

Department of Computer Science and Engineering,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
{dywang,king}@cse.cuhk.edu.hk

Abstract. Recommendation, in the filed of machine learning, is known as a technique of identifying user preferences to new items with ratings from recommender systems. Recently, one novel recommendation model using Green's function treats recommendation as the process of label propagation. Although this model outperforms many standard recommendation methods, it suffers from information loss during graph construction because of data sparsity. In this paper, aiming at solving this problem and improving prediction accuracy, we propose an enhanced semi-supervised Green's function recommendation model. The main contributions are two-fold: 1) To reduce information loss, we propose a novel graph construction method with global and local consistent similarity; 2) We enhance the recommendation algorithm with the multi-class semi-supervised learning framework. Finally, experimental results on real world data demonstrate the effectiveness of our model.

Keywords: Green's function, semi-supervised learning, recommender system, item graph.

1 Introduction

In modern days, people are usually overwhelmed with various of choices on the web and waste much time searching the wanted one. Recommendation, as the technology to suggest personalized items to meet special needs and tastes of different persons [1][6], has been widely applied into many e-commercial and entertainment web sites, like Amazon and IMDb. The feedback shows that recommender systems not only improve consumer satisfaction but also increase the profit of e-commercial systems. On the other hand, recommendation with rating information from recommender systems, as an application of machine learning, has been studied widely in academic. Normally, recommendation can be regarded as a prediction task : given a partially observed user-item rating matrix $R_0 \in \mathbb{R}^{M \times N}$, whose rows represent M users, columns represent N items, non-zero elements represent observed ratings and zero elements represent those unknown ratings, the goal is to predict unknown ratings to complete the matrix, with each element r_{jk} $(1 \leq j \leq M, 1 \leq k \leq N)$ in the range of rating $1, ..., R$ $(R > 1, R \in \mathbb{Z})$. Various recommendation methods have been proposed, mainly

K.W. Wong et al. (Eds.): ICONIP 2010, Part I, LNCS 6443, pp. 397–404, 2010.

divided into two categories: memory-based and model-based methods [10]. In our paper, we focus on the memory-based methods, which assume that similar users rate the same item similarly or similar items get similar ratings from the same users. Standard memory-based methods can be further divided into user-based [5] [12]and item-based methods [2][11].

Recently, one model applying Green's function learning framework [3] is proposed as an item-based recommendation method. Green's function for the Laplace operator represents the propagation of influence of point sources. This model takes a novel view by treating recommendation as the process of label information propagation from labeled data (i.e., items with ratings) to unlabeled data (i.e., items without ratings). Although this model is based on memory-based assumption, it utilizes the item graph and label propagation to make prediction. Moreover, it obtains a higher prediction accuracy compared to standard memory-based methods while its algorithm is more compact and simpler.

However, there are still some limitations with the Green's function recommendation model. One essential issue of the model is that the Green's function is sensitive to the item graph. The previous model constructs the item graph with the simple *cosine similarity* between items. Due to the sparse data in real recommender systems, one item with few ratings and the other with many ratings may be considered to be with low similarity using *cosine similarity.* In addition, *cosine similarity* computes the similarity in a local view since each similarity is inferred from the concurrent ratings of only two items. As a result, the item graph by *cosine similarity* suffers from information loss so that it can degrade the performance of Green's function method. Likewise, some other standard similarity computation methods, like *Pearson Correlation Coefficient* (PCC) [7] and *conditional probability* [11], also suffer the same local similarity problem. Another limitation is that the previous algorithm needs normalization and post-processing to get the prediction value and it is more preferable to the two-rating case.

In this paper, aiming at resolving the local similarity problem in item graph construction, we first propose an enhanced item Graph construction method. More specially, we derive latent features from ratings with one popular latent feature model PMF (probabilistic matrix factorization), and impose the consistency between the global similarity from latent features and the local similarity from standard similarity methods. At the same time, considering the process of Green's function recommendation similar to semi-supervised learning, we extend the previous algorithm with multi-classes semi-supervised learning, making it more suitable for recommendation with a larger rating range. Finally, we conduct a series of experiments on the famous real world dataset (i.e., MovieLens) to evaluate the performance of our model. Comparing to previous Green's function model and other memory-based methods, the experimental results demonstrate the effectiveness of our model.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of Green's function recommendation framework and item graph construction. Section 3 illustrates our novel recommendation model. The results of empirical analysis are presented in Section 4, followed by the conclusion in Section 5.

2 Related Works

2.1 Green's Function Recommendation Model

Originally Green's function is involved with the Laplace operator in many physics problems, like the diffusion process. The physical explanation is that Green's function represents the propagation of influence of point sources. Considering the similarity between label propagation and the diffusion process, Green's function is applied into machine learning with label propagation. As a graph-based learning model, Green's function is applied into recommendation [3] to predict unknown ratings for items. This recommendation model is memory-based while it is different from other standard memory-based methods which only use simple averaging ratings of similar items or users.

In this graph-based recommendation model, an item graph must be constructed first. An item graph is an undirected graph $\mathcal{G} = (\nu, \varepsilon)$ with a weight $w \in W$ in each edge $e \in \varepsilon$ and each node $v \in \nu$ as an item, where $w_{jk} = w_{kj}$, $0 \leq w_{jk} < 1$ when $j \neq k$ and $w_{jk} = 1$ when $j = k$. An item graph is constructed by calculating similarity between items. This model to construct an item graph is to utilize standard similarity methods, such as *cosine similarity* and PCC. *Cosine similarity* is simple to calculate but it is in favor of frequently rated items and ignore the different rating styles of different users. Pearson Correlation Coefficient (PCC) is proposed to consider different rating styles of users. PCC between two items is based on the common users rating the two items. However, PCC can overestimate similarities between items which happen to be rated by a few users identically in the sparse dataset [8]. Therefore, they both suffer from information loss in large sparse datasets from real recommender systems. What's more, both *cosine similarity* and PCC only take a local view of similarity, which is based on the common items or users. All these can reduce the prediction accuracy for recommendation.

When the item graph is constructed, Green's function is calculated in this way: given the item graph with edge weights W, the combinatorial Laplacian is defined as $L = D - W$, where D is the diagonal matrix with sums of each row of W, and then Green's function is defined as $G = L^{-1}_{(+)} = \frac{1}{(D-W)_+} = \sum_{i=2}^{n} \frac{v_i v_i^T}{\lambda_i}$ without the zero-value eigenvalue, where $Lv_i = \lambda_i v_i, \quad v_p^T v_q = \delta_{pq}$, and $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ are the eigenvalues of the corresponding eigenvectors $v_1, v_2, ..., v_n$. With Green's function from the item graph, recommendation is viewed as a linear influence propagation. The algorithm is defined as $\widehat{R}^T = GR_0^T$, where $R_0, \widehat{R}^T$ are the original and predicted matrix respectively. The algorithm is direct and simple while the results from this formula need much postprocess to get the integer ratings. As a matter of fact, the label propagation process is similar to semi-supervised learning which makes label prediction of unlabeled data only with labeled data. This scenario of semi-supervised learning with only a small number of labeled data is in favor of data sparsity in recommender systems.

2.2 Latent Feature Models

Latent features are usually inferred from the whole rating information and hence they are in a global view to describe items or users. In spite being implicit, some latent features can best describe the characteristics of items or users. Among latent feature models, probabilistic matrix factorization (PMF) [9][4] is used widely recently. PMF assumes the probabilistic linear model into the observations and it works well on a large and sparse dataset. The objective of PMF is to maximize the log likelihood of the posterior distribution. The gradient algorithm is applied to calculate the results.

3 Our Enhanced Model

In this section, we detail our proposed enhanced Green's function recommendation model based on: 1) a novel item graph construction with global and local consistent similarity (GLCS); 2) extended multi-classes semi-supervised recommendation algorithm.

3.1 Item Graph Based on Global and Local Consistent Similarity (GLCS)

The item graph in previous model suffers from information loss, with *cosine similarity* only in the view of local similarity. To construct a more precise item graph, we should impose the consistency between local and global view of similarity. The latent features are able to describe items in a global view since they are referred from the whole rating information, which is different from *cosine similarity* only based on ratings of two items. We adopt the popular PMF to calculate the item latent features from the user-item rating matrix R_0.

The idea of Global and Local Consistent Similarity (GLCS) is that the similarity between two items consists of two parts: one is the classical similarity and the other is similarity between two item latent feature vectors derived by PMF. The representation of GLCS is given in the following Eq.(1):

$$GLCS(j,k) = \mu sim(\mathbf{v}_j, \mathbf{v}_k) + (1-\mu) sim(j,k), \tag{1}$$

where μ is a parameter to control the weight of consistency between the global and local view, $sim(j,k)$ is *cosine similarity* or PCC as the local view, and $sim(\mathbf{v}_j, \mathbf{v}_k)$ is the *cosine similarity* of two item latent feature vectors $\mathbf{v}_j$, $\mathbf{v}_k$ as the global view.

Let r_ij be the rating of user i for item j, and each rating r_{ij} values from 0 to 1 using the mapping function $f(x) = (x-1)/(\mathbf{R}-1)$. $U \in \mathbb{R}^{k \times M}$ and $V \in \mathbb{R}^{k \times N}$ represent the latent user-specific and item-specific feature matrices, with column vectors $\mathbf{u}_i$ and $\mathbf{v}_j$ denoting user and item feature vectors respectively. We define the conditional distribution over the observed ratings as $p(R_0|U,V,\sigma_R^2) = \prod_{i=1}^{M} \prod_{j=1}^{N} \mathcal{N}[(r_{ij}|g(\mathbf{u}_i^T \mathbf{v}_j), \sigma^2)]^{I_{ij}}$, where I_{ij} is the indicator function equal to 1 if user i rated item j. Besides, we also place zero-mean spherical Gaussian priors on user and item feature vectors, with $p(U|\sigma_U^2) = \prod_{i=1}^{M} \mathcal{N}(\mathbf{u}_i|0, \sigma_U^2\mathbf{I}),\ p(V|\sigma_V^2) = \prod_{j=1}^{N} \mathcal{N}(\mathbf{v}_j|0, \sigma_V^2\mathbf{I})$. The objective of PMF is to minimize the function

$$\mathcal{L}(R_0, U, V) = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}(r_{ij} - g(\mathbf{u}_i^T\mathbf{v}_j))^2 + \frac{\lambda_U}{2}\sum_{i=1}^{M} ||\mathbf{u}_i||_{Fro}^2 + \frac{\lambda_V}{2}\sum_{j=1}^{N} ||\mathbf{v}_j||_{Fro}^2, \tag{2}$$

where $\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_V = \sigma^2/\sigma_V^2$ and $\|.\|_{Fro}^2$ denotes the Frobenius norm. A local minimum of the objective function can be obtained with gradient descent algorithm in $\mathbf{u}_i$ and $\mathbf{v}_j$.

3.2 Semi-supervised Recommendation Algorithm Based on GLCS

In our paper, we apply the algorithm of the multi-class semi-supervised learning with Green's function to recommendation. This model treats each rating in the rating sets $\{1, ..., \mathbf{R}\}$ as a label and there are R ratings corresponding to $\mathbf{R}$ labels. When we construct the item graph based on GLCS, we calculate its Green's function $G = \sum_{i=2}^{n} \frac{v_i v_i^T}{\lambda_i}$. With G, we apply the multi-classes semi-supervised learning model to our recommendation model. The algorithm is presented as:

$$y_{jk} = \begin{cases} 1, & k = \arg\max_k \sum_{i=1}^{l} G_{ji} y_{ik} \\ 0, & otherwise \end{cases}, \quad 1 \leq j \leq n. \tag{3}$$

Here we set the background to be that the ratings are discrete values. When we get the resulting $\widehat{R}^T$, we can do some data process to make sure that the elements are integers in the range $\{1, ..., \mathbf{R}\}$. One frequently used method is to set a threshold for each rating value.

4 Experimental Analysis

Our experimental analysis is expected to address the following questions: 1) What is the performance of our Green's function recommendation with GLCS comparing with previous Green's function recommendation algorithms with *cosine similarity* or PCC? 2) How does our approach comparing to traditional memory-based recommendation methods? 3) How does the parameter μ in calculating GLCS affect the performance of our approach?

4.1 Dataset and Metrics

We conduct our experiments over the MovieLens[1] dataset. The data were collected through the MovieLens, a famous Web-based recommender system, during the 7-month period from September 19th, 1997 through April 22nd, 1998.

[1] http://www.cs.umn.edu/Research/GroupsLens/

This dataset contains 100,000 integer ratings (1-5 scales) from 943 users on 1682 movies. Each user on average rated at least 20 movies with a sparsity level $1 - \frac{100000}{943 \times 1682} = 93.70\%$. The data are split 80%/20% into training data and test data, with $80,000$ ratings in training dataset and $20,000$ ratings in test dataset.

We use three most widely used metrics to measure the prediction quality of recommendation approaches in our experiments: *Mean Absolute Error*(MAE), *Mean Zero-one Error*(MZOE) and *Rooted Mean Squared Error*(RMSE).

4.2 Experimental Results

In our paper, we compared our model (GGLCS) with 6 representative methods from memory-based methods in order to measure the prediction accuracy. We implement all the 6 methods on the same dataset. Four of these baseline methods are item-based (ICOS, IPCC) [2][11] and user-based (UCOS, UPCC) [5] method with *cosine similarity* and PCC respectively. GCOS and GPCC are the previous model using *cosine similarity*and PCC respectively. Besides, we also conduct some experiments to measure the impact of parameter weight μ with different latent feature dimensionality k. In our experiments, the rating background is set to be discrete-valued.

- **Impact of Weight Parameter μ.** There is a weight parameter μ to balance the similarity from global view with latent features and local view from *cosine similarity* or PCC. When $\mu = 0$, the item similarity is only the classical *cosine similarity* or PCC, and when $\mu = 1$, the item similarity is only derived from latent features. In other cases that μ is between $(0, 1)$, we obtain the item similarity combining with global and local views.

  Fig. 1 shows the impacts of μ on MAE, MZOE and RMSE with the dimensionality $k = 5$. We can observe from this figure that the value of μ affects the performances of our model significantly. As shown in all the three charts in Fig 1, when μ increases, the three prediction errors decrease first. But when μ passes 0.5, the prediction error begins to decrease with further increase of μ. From Fig. 2, we can observe the similar sensitivity of weight parameter μ to performance of our model when the $k = 10$. The optimal value of weight parameter is near 0.5 to get the best prediction accuracy.

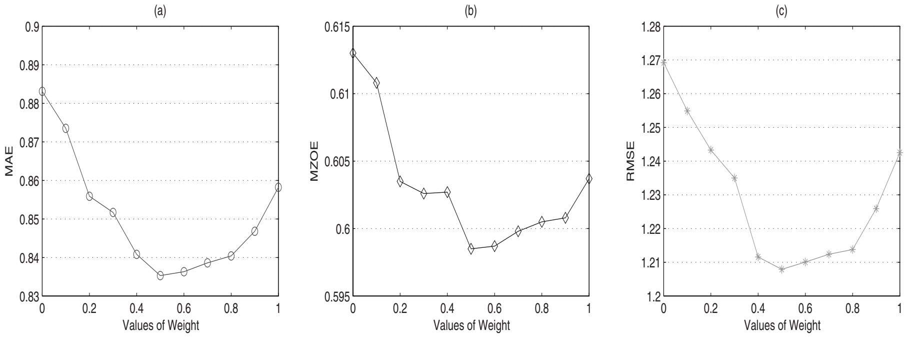

Fig. 1. Performance with Different Values of Weight Parameter μ at k=5

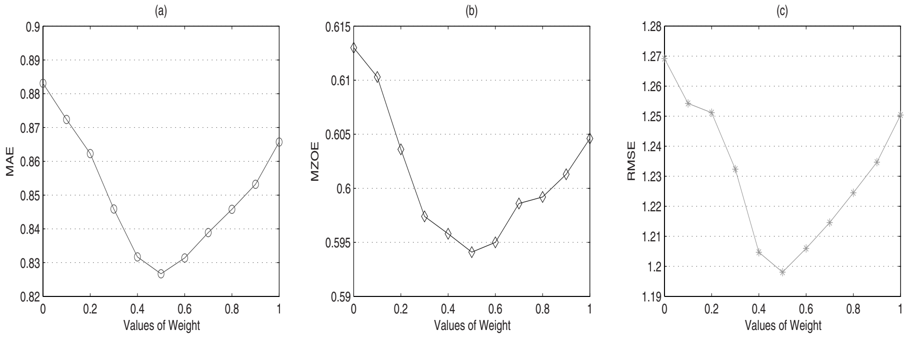

Fig. 2. Performance with Different Values of Weight Parameter μ at k=10

Table 1. Comparison with Different Recommendation Methods on MAE, MZOE and RMSE

Methods **Metrics**	UCOS	UPCC	ICOS	IPCC	GCOS	GPCC	GGLCS(k_5)	GGLCS(k_10)
MAE	0.962	0.943	0.890	0.889	0.883	0.861	**0.835**	**0.827**
MZOE	0.653	0.639	0.621	0.618	0.613	0.604	**0.599**	**0.594**
RMSE	1.281	1.278	1.275	1.271	1.269	1.249	**1.208**	**1.198**

This phenomenon coincides with the intuition that item graph construction with both global and local similarity can help to obtain better performances. We can conclude that when the weight μ between global and local similarities is equal and balanced, our enhanced model can have a better performance in the MovieLens dataset.

- **Performance Comparison.** In order to evaluate the efficiency and superiority of our model, we set the parameter $\mu = 0.5$ in our model which achieves the best consistency between global and local similarity and the best accuracy. Table 1 shows the results of performance of different recommendation methods. We observe that our model GGLCS has the lowest MAE, MZOE and RMSE among these errors in both $k = 5$ and $k = 10$. Compared to MAE of previous model with *cosine similarity*, the accuracy of our model improves about 6.4% when $k = 10$ and 5.4% when $k = 5$. Compared to ICOS, our model at $k = 10$ can improve the accuracy about 7.2%. The results demonstrate efficiency and superiority of our model over other memory-based methods. Another observation from this table is that user-based recommendation is worse than any item-based recommendation in MovieLens dataset and also much worse than any Green's function recommendation.

5 Conclusions

Previous Green's function recommendation model with *cosine similarity* suffers from information loss during the item graph construction in sparse data due to the local similarity problem. In this paper, we propose an enhanced method to construct the item graph by imposing global and local consistence on similarity computation, where the global view is provided by latent features. What's more, we extend the Green's function recommendation algorithm with multi-classes semi-supervised learning framework. Finally, we conduct some experiments with a famous dataset to demonstrate that our approach outperforms the previous Green's function recommendation and other memory-based methods.

Acknowledgment

The work described in this paper is supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No.: CUHK 4128/08E).

References

1. Adomavicius, G., Tuzhilin, A.: 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 17(6), 734–749 (2005)
2. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22(1), 177 (2004)
3. Ding, C., Simon, H., Jin, R., Li, T.: A learning framework using Green's function and kernel regularization with application to recommender system. In: Proceedings of the 13th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, p. 269. ACM, New York (2007)
4. Dueck, D., Frey, B.: Probabilistic sparse matrix factorization. University of Toronto technical report PSI-2004-23 (2004)
5. Jin, R., Chai, J., Si, L.: An automatic weighting scheme for collaborative filtering. In: Proceedings of the 27th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 337–344. ACM, New York (2004)
6. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer 42(8), 30–37 (2009)
7. Ma, H., King, I., Lyu, M.: Effective missing data prediction for collaborative filtering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 46. ACM, New York (2007)
8. McLaughlin, M., Herlocker, J.: A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In: Proceedings of the 27th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 329–336. ACM, New York (2004)
9. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. Advances in neural information processing systems 20, 1257–1264 (2008)
10. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, p. 295. ACM, New York (2001)
11. Wang, F., Ma, S., Yang, L., Li, T.: Recommendation on item graphs. In: Perner, P. (ed.) ICDM 2006. LNCS (LNAI), vol. 4065, pp. 1119–1123. Springer, Heidelberg (2006)
12. Wang, J., De Vries, A., Reinders, M.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, p. 508. ACM, New York (2006)