ICIC Express Letters
Volume **8**, Number **12**, December **2014**
 ISSN 1881-803X
pp. **3299–3306**

LOTMAP: LEARNING TO OPTIMIZE TOP-N RECOMMENDATION WITH MEAN AVERAGE PRECISION

Yehui Yang[1], Wensheng Zhang[1], Yuan Xie[1], Limin Zhu[1]
and Yuanhua Tan[2]

[1]Institute of Automation, Chinese Academy of Sciences
No. 95, Zhongguancun East Road, Beijing 100190, P. R. China
{ yehui.yang; wensheng.zhang; yuan.xie; limin.zhu }@ia.ac.cn

[2]Karamay Hongyou Software CO., LTD
No. 22, Changzheng Road, Karamay 834000, P. R. China
tanyh66@petrochina.com.cn

Received January 2014; accepted April 2014

Abstract. *People tend to pay attention to the top few items given by recommendation system, and it is crucial to ensure the precision and personalization at the top of recommendation list. In this paper, we propose a ranking-oriented Collaborative Filtering (CF) algorithm LOTMAP, which aims to optimize the top-N recommendation by directly maximizing Mean Average Precision (MAP). As the relevance between users and items are not clarified in rating datasets, we also introduce a new method that judges the relevance through the explicit rating scores. Experiments on real world datasets show that our algorithm is effective, and outperforms the state-of-the-art CF baselines.*
Keywords: Top-N recommendation, Collaborative filtering, Optimize, Matrix factorization, Mean average precision

1. **Introduction.** Recently, the information on the internet begin to explode, and it is a tough work for people to find what he/she really wants. Personalized recommendation systems aim to provide useful information according to the interests of the target users, facilitate our web life, and have been intensely studied by both academies and industries [1, 2].

Collaborative Filtering (CF) is one of the core methods in the construction of recommendation system [3, 4], and it is often adopted in two application areas [5, 6]. In one case, target user is provided with one item in one recommendation, then we measure the error between predicted rating of the item and its hold-out truth. The evaluation metrics for this scenario used to be Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) [7, 8]. The other case is ranking-oriented CF, it gives the top-N recommendation by generating an ordered list to the target user. The main metrics to for this case are MAP, Precision/Recall, and Mean Reciprocal Rank (MRR). As users tend to pay attention to some top items given by the recommenders [9, 10], it is more reasonable to tackle top-N recommendation.

The relationships between users and items are mainly showed in two scenarios. First, the relationship is binary (0-1); 1 means that the user is relevant with the item, 0 means that the user is irrelevant with the item, and a typical example of this scenario is the friendship in social network. Second, the user shows his/her preference on the items by explicitly rating on them. The rating scores are according to the preference which the user presents on the items, in other words, items may get higher scores if the user is more fond of them. However, the relevance between users and items are not defined in this case, in order to calculate MAP metric in this scenario, we need to clarify whether the item in

the recommendation list is relevant with the user or not. In this paper, we propose a new method to judge the relevance between users and items through explicit rating scores.

All the recommendation methodologies may start from a user-item matrix [11], which records either the ratings of items given by the users, or the binary relationships between users and items. However, the user-item matrix is often extremely sparse, for a user can only rate on a small percentage of items among all the possible candidates. In order to find meaningful information in such sparse data, some latent factor models based on Matrix Factorization (MF) are shown to be particular effective [10,12-14]. [10,14] try to use MF to minimize RMSE between the predicted ratings and the groundtruths, but we are different from them for LOTMAP is a ranking-oriented CF which aims to optimize MAP metric. [15, 16] are also ranking-oriented CF based on MF, however, [15] can only provide few relevant items because of the "less-is-more" limitation brought by MRR. [16] has been recently proved to be an effective ranking-oriented CF, but it can only tackle binary relationships.

In this paper, we propose an effective ranking-oriented CF algorithm for top-N recommendation by directly maximizing the MAP metric, and we implement MF to deal with the sparsity in user-item matrix, additionally, in order to extend the MAP matric to rating datasets, we also introduce a new method to clarify relevance between users and items through their rating scores.

The paper is organized as follows. In Section 2, we present details of LOTMAP algorithm. The experimental evaluation will be reported in Section 3 to show the effectiveness, outperformance of LOTMAP. Finally, conclusions and future work are followed in Section 4.

2. The Details of LOTMAP.

In this section, we present the details of LOTMAP. Firstly, we introduce MAP into the algorithm, then we optimize the MAP function to give the objective function of LOTMAP, finally, we derive a learning algorithm to construct our recommendation model.

2.1. The use of MAP in LOTMAP.

MAP is a well known metric to evaluate the performance of top-N recommendation, and it has been shown of good discrimination and stability compare to some other evaluation measures [17]. Given a recommendation list to target user i in a descending order according to the items' predicted scores, the MAP of the list can be calculated as

$$MAP = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{m_i}\sum_{j=1}^{N} precision(R_{ij}) \tag{1}$$

where M is the number of the users, m_i denotes the number of relevant items for user i in the list, N is the number of all target items, and $precision(R_{ij})$ denotes the precision of top-j items in the recommendation list if the j-th item is relevant with user i, otherwise $precision(R_{ij}) = 0$.

LOTMAP aims to generate a recommendation list for each user by maximizing the MAP metric, and the core methodology of LOTMAP is based on MF. Inspired by [7], we use $p_{ij} = u_i v_j^T$ to fit the observed groundtruth r_{ij}, where p_{ij} denotes the predicted rating score of item j given by user i, and r_{ij} is the hold-on groundtruth, u_i and v_j are vectors which only associate with user i and item j respectively. In order to count m_i and $precision(R_{ij})$, we define $y_{ij} = 1$ if user i is relevant with item j, otherwise $y_{ij} = 0$. Additionally, we also define an auxiliary variable ρ_{kj} as

$$\rho_{kj} = \begin{cases} 1, & rank_{ik} < rank_{ij} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $rank_{ij}$ denotes the rank of item j in the recommendation list given by user i. Then, we can calculate the precision in top-j items as:

$$precision(R_{ij}) = \frac{y_{ij}}{rank_{ij}} \sum_{k=1}^{N} y_{ik}\rho_{kj} \tag{3}$$

Substituting $m_i = \sum_{j=1}^{N} y_{ij}$ and Equation (3) into Equation (1), we can obtain:

$$MAP = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{y_{ij}}{\sum_{j=1}^{N} y_{ij} rank_{ij}} \sum_{k=1}^{N} y_{ik}\rho_{kj} \tag{4}$$

2.2. **Optimizing the function of MAP.** In order to maximize the MAP function of Equation (4), we need to solve the next two problems: (a) How to define the relevance between users and items? (b) How to smooth discrete indexes in Equation (4) and make it available for some optimization methods like gradient descent?

2.2.1. *Judging the relevance.* As introduced in Section 1, we can easily find the relevance between users and items in binary dataset, but the relevance is not clarified in rating datasets. In such datasets, item trends to get a higher score if the user is more interested in it, so it is reasonable to infer that the user is relevant with the item which he/she rates high. We can take the judgement as a classification problem that classifies the rating data into relevance or irrelevance.

In this paper, we construct the classifier based on logistic function: if $f(r_{ij}) \geq \mathrm{T}$, user i is relevant with item j, otherwise they are irrelevant, where T is the threshold to judge relevance, $f(.)$ is logistic function: $f(\mathrm{x}) = \frac{1}{1+e^{-\theta x}}$, θ is the parameter to control the slope of logistic function. As different users may have different rating habits – some users trend to give high scores, and some always give low scores, we alleviate the bias caused by users' habits, and derive the final judgement as follows:

$$y_{ij} = \begin{cases} 1, & f(\mathrm{r_{ij}} - \overline{\mathrm{r_i}}) \geq \mathrm{T} \\ 0, & f(\mathrm{r_{ij}} - \overline{\mathrm{r_i}}) < \mathrm{T} \end{cases} \tag{5}$$

where, $\overline{\mathrm{r_i}}$ denotes the average scores given by user i.

2.2.2. *Smoothing the discrete index.* Now we look back to Equation (4), ρ_{kj} and $rank_{ij}$ are discrete indexes both related with the items' rank in the recommendation list. Based on the work of learning to rank [18], approximately, we can use logistic function to smooth ρ_{kj} and $rank_{ij}$:

$$\rho_{kj} \approx f(r_{ik} - r_{ij}) \tag{6}$$

$$\frac{1}{rank_{ij}} \approx f(r_{ij} - \overline{\mathrm{r_i}}) \tag{7}$$

Substituting Equation (6), Equation (7) into Equation (4), we can get a smoothed function of MAP:

$$MAP = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{y_{ij} f(r_{ij} - \overline{\mathrm{r_i}})}{\sum_{j=1}^{N} y_{ij}} \sum_{k=1}^{N} y_{ik} f(r_{ik} - r_{ij}) \tag{8}$$

2.3. **Generating objective function and the learning algorithm of LOTMAP.**

2.3.1. *Generating objective function.* We have got the smooth version of MAP in the last section. Our goal is to find an optimum predicting model $U = [u_1, u_2, \ldots, u_M]$, $V = [v_1, v_2, \ldots, v_N]$ to maximize Equation (8). By adding the regularization term to avoid overfitting, we obtain the objective function of LOTMAP:

$$L(U,V) = \max_{U,V} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{y_{ij} f(u_i v_j^T - \overline{r_i})}{\sum_{j=1}^{N} y_{ij}} \sum_{k=1}^{N} y_{ik} f(u_i v_k^T - u_i v_j^T) - \frac{\lambda}{2}(||U||^2 + ||V||^2) \tag{9}$$

where we substitute the score r_{ij} by $u_i v_j^T$, $||U||$ and $||V||$ denote the Frobenius norm of U and V, and λ is the regularization coefficient. Since the number of user M is a constant, we can omit it during the process of maximizing the objective function.

2.3.2. *The learning algorithm of LOTMAP.* We use gradient descent to maximize the objective function; for each user i, the partial derivative of Equation (9) with respect to u_i and v_j are computed as follows:

$$\begin{aligned} \frac{\partial \mathrm{L}}{\partial \mathrm{u_i}} = \frac{1}{\sum_{j=1}^{N} y_{ij}} \sum_{j=1}^{N} \sum_{k=1}^{N} y_{ij} y_{ik} \Big[& f'(\mathrm{u_i v_j^T} - \overline{\mathrm{r_i}}) f(\mathrm{u_i v_k^T} - \mathrm{u_i v_j^T}) \mathrm{v_j} \\ & + f(\mathrm{u_i v_j^T} - \overline{\mathrm{r_i}}) f'(\mathrm{u_i v_k^T} - \mathrm{u_i v_j^T})(\mathrm{v_k} - \mathrm{v_j}) \Big] - \frac{\lambda}{2} \mathrm{u_i} \end{aligned} \tag{10}$$

$$\begin{aligned} \frac{\partial \mathrm{L}}{\partial \mathrm{v}_j} = \sum_{i=1}^{M} \frac{y_{ij} u_i}{\sum_{j=1}^{N} y_{ij}} \sum_{k=1}^{N} y_{ik} \Big[& f(u_i v_k^T - u_i v_j^T) f'(u_i v_j^T - \overline{r_i}) \\ & - f'(u_i v_k^T - u_i v_j^T) \left(f'(u_i v_j^T - \overline{r_i}) - f'(u_i v_k^T - \overline{r_i}) \right) \Big] - \frac{\lambda}{2} \mathrm{v}_j \end{aligned} \tag{11}$$

Using the partial derivatives list above, U, V can be learned by solving Equation (9) using gradient descend algorithm. The learning algorithm of LOTMAP is presented in Algorithm 1.

Given the learned U and V, we can predict the rating score between user i and item j by $p_{ij} = u_i v_j^T$, and then generate the top-N recommendation lists according to the predicted scores.

3. **Experiments.**

In this section, we set up a series of experiments to evaluate our algorithm. First, we introduce the datasets briefly. Second, we present the experiment protocol and evaluation metrics. Finally, we show the comparison between LOTMAP and alternatvie baselines on benchmark datasets.

3.1. **Datasets.**

The datasets for our experiment are Movielens [19], Yahoo Music [20] and Epinions [21]. They are all popular datasets for recommendation system. The first two are rating datesets. Movielens contains 10 000 054 ratings (scale 1-5) which applied to 10 681 movies by 71 567 users at the online movie recommender service Movielens. Yahoo Music contains 11 557 943 ratings of 98 211 artists by 1 948 882 anonymous users, the ratings are scale from 1 to 100. Epinions is a binary datasets which contains trust relationships between 49 288 users. Statistics on the three datasets are summarized in Table 1.

Algorithm 1 Learning Algorithm of LOTMAP

Input: The user-item matrix for training R, slope parameter θ, the relevance threshold T, regularization coefficient λ, the number of iterate $NumIter$

Output: The predicting latent factors U, V

% Using logistic function to project R into pR

Set pR as all zero matrix with the same dimension of R;
for All the nonzero entries r_{ij} in R **do**
  $pr_{ij} = f(\mathrm{r_{ij}} - \overline{\mathrm{r_i}})$;
end for
Initialize $U^{(0)}$, $V^{(0)}$ with random values, set $t = 0$;
while $t < NumIter$ **do**
  for $i = 1, 2, 3, \ldots, M$ **do**
    Find the relevant items of user i according to pR through Equation (5);
    Count $\frac{\partial L}{\partial u_i}$ in the positions of relevant items based on Equation (10);
  end for
  % Update U
  $U^{(t+1)} = U^{(t)} + \gamma \frac{\partial L}{\partial U^{(t)}}$
  for $j = 1, 2, 3, \ldots, N$ **do**
    Find the relevant users of item j according to pR through Equation (5);
    Count $\frac{\partial L}{\partial v_j}$ in the positions of relevant users based on Equation (11);
  end for
  % Updata V
  $V^{(t+1)} = V^{(t)} + \gamma \frac{\partial L}{\partial V^{(t)}}$;
  $t = t + 1$;
end while
$U = U^{(t)}$, $V = V^{(t)}$

Table 1. Statistics of the data sets

Dataset	Movielens	Yahoo Music	Epinions
Num. non-zeros	10 000 054	11 557 943	346 035
Num. users	71 567	1 948 882	4 718
Num. items	10 681	98 211	49 288
Sparseness	98.69%	99.99%	99.85%

3.2. **Experiment protocol and evaluation metric.** We use both rating and binary datasets to evaluate the performance of LOTMAP. In binary dataset, we directly get the relevance between users and items from the user-item matrix, while in rating datasets, we use Equation (5) to judge the relevance. In each dataset, we randomly select 10% data to carry out validation experiments, and tune the following parameters to yield the best performance in the validation test. We set slope coefficient $\theta = 3$, relevance threshold $T = 0.9$ in Equation (5), and set the latent dimension of user and item vector $d = 10$. regularization parameter $\lambda = 0.001$ in Equation (9), the learning rate γ is set to 0.0001 in Algorithm 1.

Three metrics are used in our experiments. Besides MAP, we also measure other two metrics: $P@5$, F-measure. $P@5 = \frac{rel\_5}{5}$ is the precision of the top 5 items in the recommendation list, where rel_5 refers the number of relevant items in top 5 retrieval. F-measure is a metric which considers both precision and recall in the retrieval, $F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R}$, where P and R denote the precision and the recall in top-N retrieval. β is a coefficient to balance the weight between precision and recall, and we set $\beta = 1$ in this paper.

3.3. **Experimental procedure.** In this part, we focus on the following research questions: (a) Is the learning algorithm convergent and can it obtain the maximum MAP during the learning process? (b) Does our algorithm outperform other alternative state-of-the-art CF approaches in top-N recommendation?

3.3.1. *The convergence of the learning algorithm.* A successful learning algorithm must be convergent, and can get the best result in finite time, so we conduct an experiment to test the effectiveness of the learning algorithm. We implement LOTMAP on Movielens and Yahoo Music dataset and plot the variation of MAP metric during iterations. As presented in Figure 1, the MAP in both datasets gradually increases with iterations at the beginning and trends to convergence at last, then we confirm that LOTMAP is an effective top-N recommendation approach which can get the maximum of MAP during the iterations. As we can see in Figure 1, the algorithm becomes convergence after approximate 20 iterations, so we set the iteration number $NumIter = 20$ in the follow experiments.

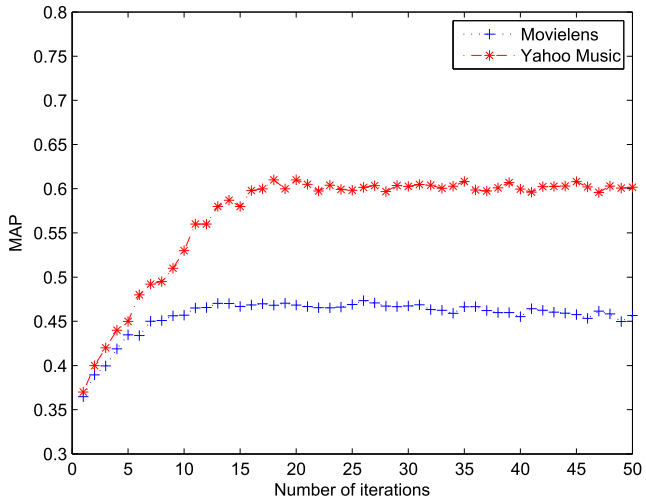

FIGURE 1. The convergence of learning algorithm for LOTMAP

3.3.2. *Performance comparison.* We have proved that the learning algorithm is convergence during the iterations; in this section, we wonder if LOTMAP outperforms the state-of-the-art recommenders. We compare LOTMAP with the next three baselines: PopRec, Basic-MF [22], Item-based CF [23].

- PopRec: A naive non-personalized baseline recommends items based on their popularity. This approach provides the same items to all users. In this paper, we define the popularity of an item by counting how many users have rated on the item. In this paper, PopRec returns 5 most popular items to all users.
- Basic-MF: A basic matrix factorization for recommendation system by Koren *et al.*, this approach aims to produce recommendations by minimizing RMSE metric.
- Item-based CF: A neighborhood-based method, it provides items to the users by analyzing the similarity among items. Here the similarity metric between the items is adjusted cosine similarity.

The performance of LOTMAP and the baselines on Movielens and Yahoo Music are summarized in Table 2. The comparison on Movielens is listed in the first place, and the number in parentheses is the results on Yahoo Music. The comparison on Epinions is presented in Table 3. The best results are shown in bold. We have not computed MAP metric in PopRec, for it is unreasonable to compare MAP when PopRec provides users

Table 2. Performance comparison on Movielens and Yahoo Music dataset

Method	MAP	P@5	F_1
PopRec	– (–)	0.21204(0.24804)	0.1155(0.0834)
Basic-MF	0.33262(0.60519)	0.21568(0.49572)	0.26630(0.49348)
Item-based CF	0.35110(0.59886)	0.23340(0.52336)	0.35350(0.48731)
LOTMAP	**0.47339(0.60815)**	**0.33968(0.52888)**	**0.39178(0.49199)**

Table 3. Performance comparison on Epinions dataset

Method	MAP	P@5	F_1
PopRec	– (–)	0.23547	0.15830
Basic-MF	0.33816	0.27931	0.36514
Item-based CF	0.56372	0.36429	0.47258
LOTMAP	**0.58137**	**0.39206**	**0.52697**

with only 5 popular items while LOTMAP generates an ordered recommendation list to the users.

3.4. **Discussions.** We can find two main observations in Table 2 and Table 3. First, LOTMAP preforms better than all the baselines on P@5 and F_1 metric besides MAP. It means that by maximizing MAP, LOTMAP gets corresponding elevation in precision and recalls metrics for top-N recommendation. Second, nearly all the three methods perform better on the Yahoo Music dataset than on Movielens dataset under the same condition. We own this to the difference of rating scale between these two datasets. Movielens scores scales from 1 to 5, while Yahoo Music scales from 1 to 100, obviously, and Yahoo Music can describe the interests of users more explicitly than Movielens do, so the CF algorithms work better on Yahoo Music dataset. However, there also exit two exceptions, one is that the F_1 metric of PopRec on Yahoo Music is smaller than on Movielens, the other is LOTMAP convergence is faster on Movielens than on Yahoo Music as presented in Figure 1. The reason for this exceptions may lie in Yahoo Music dataset which is more sparse than Movielens dataset.

4. **Conclusions.** In this paper, we propose an effective CF approach LOTMAP for top-N recommendation based on MF, and we optimize the recommender by directly maximizing the MAP metric. To extend LOTMAP to explicit rating datasets, we give a new method to judge the relevance between users and items through rating scores. Experiments on real world datasets show that LOTMAP is effective and outperforms the alternative baselines. By the way, we also find that the rating scale of datasets may affect the performance of CF algorithms. Future work will be done to make the better use of the context information to construct a better recommender.

Acknowledgment. The work is supported by the National Natural Science Foundation of China (Nos. U1135005, 90924026), the knowledge innovation program of the Chinese academy of sciences (No. Y1W1031PB1), Hi-Tech Research and Development Program of China (863) (2013AA01A607), and the Project for the National Basic Research 12th Five Program (No. 0101050302).

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Trans. on Knowledge and Data Engineering*, pp.734-749, 2005.

[2] R. Qumsiyeh and Y. K. Ng, Predicting the ratings of multimedia items for making personalized recommendations, *Proc. of ACM SIGIR*, pp.475-484, 2012.

[3] J. L. Herlocker, J. A. Konstan, A. Borchers and J. Riedl, An algorithmic framework for performing collaborative filtering, *Proc. of ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, pp.230-237, 1999.
[4] K. Niemann and M. Wolpers, A new collaborative filtering approach for increasing the aggregate diversity of recommender systems, *Proc. of ACM SIGKDD*, Chicago, USA, pp.955-963, 2013.
[5] N. N. Liu and Q. Yang, EigenRank: A ranking-oriented approach to collaborative filtering, *Proc. of ACM SIGIR Conference on Research and Development in Information Retrieval*, 2008,
[6] J. S. Breese, D. Heckerman and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, *Proc. of UAI*, pp.43-52, 1998.
[7] Y. Koren, Factorization meets the neighborhood: A multifaceted collaborative filtering model, *Proc. of the 14th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp.426-434, 2008.
[8] T. Hofmann, Collaborative filtering via Gaussian probabilistic latent semantic analysis, *Proc. of ACM SIGIR*, 2003.
[9] M. Deshpande and G. Karypis, Item-based top-N recommendation algorithms, *ACM Trans. on Information Systems*, vol.22, no.1, pp.143-177, 2004.
[10] P. Cremonesi, Y. Koren and R. Turrin, Performance of recommender algorithms on top-N recommendation tasks, *Proc. of RecSys, ACM*, Barcelona, Spain, pp.39-46, 2010.
[11] A. Rajaraman and J. Ullman, *Mining of Massive Datasets*, http://i.stanford.edu/ullman/mmds.html.
[12] L. Pu and B. Faltings, Understanding and improving relational matrix factorization in recommender systems, *Proc. of RecSys, ACM*, Hong Kong, China, pp.41-48, 2013.
[13] O. Koyejo, S. Acharyya and J. Ghosh, Retargeted matrix factorization for collaborative filtering, *Proc. of RecSys, ACM*, Hong Kong, China, pp.49-56, 2013.
[14] R. Salakhutdinov and A. Mnih, Probabilistic matrix factorization, *NIPS*, vol.20, 2008.
[15] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver and A. Hanjalic, CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering, *Proc. of RecSys, ACM*, New York, USA, pp.139-146, 2012.
[16] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic and N. Oliver, TFMAP: Optimizing MAP for top-N context-aware recommendation, *Proc. of ACM SIGIR*, New York, USA, pp.155-164, 2012.
[17] C. D. Manning, P. Raghavan and H. Schtze, *Introduction to Information Retrieval*, Cambridge University Press, New York, 2008.
[18] O. Chapelle and M. Wu, Gradient descent optimization of smoothed information retrieval metrics, *Inf. Retr.*, pp.216-235, 2010.
[19] *Grouplens*, http://www.grouplens.org/taxonomy/term/.
[20] *Yahoo*, http://research.yahoo.com/AcademicRelations.
[21] *Epinions*, http://www.epinions.com/, 2013.
[22] Y. Koren, R. Bell and C. Volinsky, Matrix factorization techniques for recommender systems, *Computer*, vol.42, no.8, pp.30-37, 2009.
[23] B. Sarwar, G. Karypis, J. Konstan and J. Reidl, Item-based collaborative filtering recommendation algorithms, *Proc. of Int. Conf. World Wide Web*, Hong Kong, China, pp.285-295, 2001.