Online Event Recommendation for Event-based Social Networks

Xiancai Ji [1,3], Mingze Xu [4], Peng Zhang[2,1], Chuan Zhou[1], Zhi Qiao [1,3] and Li Guo [1]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100190, China
[2]Quantum Computation and Intelligent Systems, University of Technology, Sydney (UTS), Australia
[3]University of Chinese Academy of Sciences, Beijing 100190, China
[4]Shandong University, Shandong, China
{jixiancai,zhouchuan,guoli}@iie.ac.cn,Peng.Zhang@uts.edu.au, qiaozhi@nelmail.iie.ac.cn

ABSTRACT

With the rapid growth of event-based social networks, the demand of event recommendation becomes increasingly important. While, the existing event recommendation approaches are batch learning fashion. Such approaches are impractical for real-world recommender systems where training data often arrive sequentially. Hence, we present an online event recommendation method. Experimental results on several real-world datasets demonstrate the utility of our method.

Categories and Subject Descriptors H.2.8 [Database Management]: Database Applications - Data Mining
General Terms Theory, Algorithms, Performance
Keywords Event Recommendation, Online Learning.

1. INTRODUCTION

With the rapid growth of event-based social networks (EBSNs) like Meetup, the demand for event recommendation becomes increasingly urgent. Different from traditional recommendation problems, event recommendation encounters the implicit feedback challenge. In [3], the HeSig model is presented to make accuracy event recommendation. Additionally, collaborative filtering and matrix factorization technologies also can be applied into model the task by assuming the rating as 0/1. But, the existing recommendation approaches usually assume a collection of users rating data is given as the priori to train the model in a batch learning fashion. Typically, the model has to be re-trained whenever there is new training data. Such approaches are impractical for real-world recommender systems where training data often arrive sequentially as new users are being added daily or even hourly, and new products items are being offered dynamically. This calls for an urgent need of efficient and scalable learning technique for the online event recommendation task in the real-world recommender systems.

2. OUR METHOD

In this section, we present the online event recommendation framework, which constitutes with two parts, one for the basis model and the other for the online model. The basis model is used to learn the initial observed data. The online model is used to learn the realtime arriving data.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742742.

2.1 Basis Model

We apply matrix factorization latent factor model (short for MF) as basis model, which has spawned a large body of researches and been proved efficiently. In the setting of matrix factorization, the fundamental idea is to embody user i and item j with low-dimension latent factors vectors U_i and V_j. Then the dyadic rating $r(u_i, v_j)$ of user i to item j is usually approximated according to inner product $U_i^T V_j$.

However, feedback of a user against an event is implicit, where the presence or absence is represented by binary value 1 or 0. The Bayesian Personalized Ranking (BPR)[5] emphases on predicting the dyadic rating $r(u, v)$ and make the items with higher ratings rank higher, which is efficient for the implicit feedback recommendation problem. Actually, the **B**PR-based **M**atrix **F**actorization (short for **BMF**) method has been presented in the [5]. Here, we just apply it into the event recommendation problem. The positive event set is constituted with the events participated by user u and the negative event set is constituted with the left events. Then the best optimization result for user is that all events participated by him should rank higher than other events that he didn't participate, which approximately can be expressed by the above optimization problem,

The ***Parameter Learning*** use the stochastic gradient descent as in [5]. Owe to the space limits, the detailed is neglected.

2.2 Online Model

The online learning is an effective way to handle large-scale data, especially streaming data[6]. Hence, we firstly proposed an extended online learning method according to online Passive-Aggressive learning [4]. However, the online learning usually introduces noise when process data one by one. In order to improve stability of the online learning algorithm, a useful technique to reduce the noise in data is the use of mini-batches as in the work [4], which means that the practitioners typically use multiple samples to compute gradients at a time. In our case, suppose that we have a mini-batch of data points at time t, denoted as $Y_t = \{y_{ij}\}$. Our model online updates based on the mini-batch observations instead of a data point at each time stamp for stream data analysis.

Suppose that current mini-batch observation is $Y^t = \{y_{ij}\}$ at time t. We can obtain the objective function as follows,

$$min_{\Theta} ||\Theta - \Theta^{t-1}||_2 - \epsilon logp(\Theta|Y_t) \tag{1}$$

The goal of objective function is to minimize the posterior probability based on the sequentially arriving training samples. Intuitively, if Θ get maximum posterior probability, the algorithm passively assigns $\Theta_{t+1} = \Theta_t$; otherwise, it aggressively projects Θ to the feasible zone of parameter vectors that attain maximum objective function.

Then, we apply the **BMF** method into the **on**line learning framework (short for **OnBMF**) for online event recommendation. As referred, we can get the objective function as follows.

$$min_{U,V,M,\Pi}||U - U^{t-1}||_2 + ||V - V^{t-1}||_2 + ||M - M^{t-1}||_2 + ||\Pi - \Pi^{t-1}||_2 - \epsilon \sum_{(i,j,k)\in D_s(Y_t)} ln \frac{1}{1 + e^{r(u_i,v_k)-r(u_i,v_j)}} \tag{2}$$

Parameter Learning. In order to fast optimize the objective function, we also use the stochastic gradient descent. Instead of computing the gradient of entire objective function exactly, each iteration estimates this gradient on the basis of a single randomly picked example, $(i, j, k) \in D_s(Y_t)$, which represents user i participated into event j not event k in time t, the parameters updating is similar as follows.

$$U_i = U_i - \lambda[C\alpha F(i,j,k)(V_k - V_j) + 2(U_i - U_i^{t-1})] \tag{3}$$

$$V_j = V_j - \lambda[2(V_j - V_j^{t-1}) - \epsilon\alpha\{F(i,j,k)\} \cdot U_i] \tag{4}$$

Algorithm 1: The algorithm of OnBMF

Input: rating set at $t + 1$ time
$Y_t = \{< u_i, v_j, v_k > |newly\ rating - pairs\ on\ current\ time\}$, last users' latent factors U^{t-1} learnt by history data, last items' latent factors V^{t-1}
Output: U^t, V^t

```
If t==0
    Learn U^t, V^t with basis model as [2].
Else
    Initialize U = U^{t-1}; V = V^{t-1};
    For each <u_i, v_j, v_k> in Y_t
        Update U_i according Equation 4;
        Update V_j according Equation 5;
    End For
End If
U^t = U; V^t = V;
```

3. EXPERIMENTS

In this section, we analyze the performance of our proposed online event recommendation method. We first got the five data sets as in Table 1 for the five American cities in Meetup by extracting them from the data sets published in [2]. We then create 5 new datasets by preprocessing the 5 original datasets. For each datasets, we firstly split the original rating data ($<u_i, v_j, time>$) into basis data, online data and test data, where the ratio is 3:6:1. Then, we split the online data into many time splices ($(u_i, v_j) \in Y_t$) according to the min-batch size setting, where each splice has the corresponding time flag.

Hence, we firstly use ***the basis data*** to learn the basic model by applying the basis model. For ***the online data***, suppose current time t, we apply the OnBMF to update the model. In order to verify the algorithm performance, the learnt model can be applied into test data.

Measurement: AUC measures [3] the overall results of classification. It is suitable for highly imbalanced data set, as in our case where the negative events take a high proportion.

Firstly, as referred, the min-batch suppose is efficient in the online learning. Hence, we firstly discuss the effect of the min-batch setting. Suppose the latent factor dimension size is 100. The experimental results are shown in the Table 2. We can find that the accuracy firstly increase and then decrease, when the batch size increases on each dataset. In the following experiments, we experimentally set the min-batch size as 16.

Table 1: Statistics of the Data sets in Meetup.

Meetup	Houston	Chicago	NYC	LA	SF
users	36199	89796	338144	124040	119569
events	16694	36009	108170	54538	45213

Table 2: AUC of test sets

Datasets	size=8	size=16	size=32	size=64	size=128
Houston	0.581	0.635	0.661	0.647	0.578
Chicago	0.686	0.716	0.736	0.735	0.736
LA	0.672	0.703	0.705	0.691	0.647
NYC	0.657	0.707	0.716	0.719	0.711
SF	0.706	0.698	0.706	0.711	0.702

Then, we compare our proposed OnBMF with several representative online collaborative recommendation methods. Specifically, the compared algorithms in our experiments include:

- OCF: the Online Collaborative Filtering algorithm by online gradient descent method described in [1];
- OM^3F: the presented online maximum margin matrix factorization learning as shown in [4].

All the experimental results are reported by averaging over these 5 runs as in table 3. We can find that our proposed algorithm $OnHeSig$ has better performance than other methods

Table 3: AUC of test sets

Method	Houston	Chicago	LA	NYC	SF
OCF	0.568	0.615	0.584	0.603	0.551
OM3F	0.486	0.473	0.527	0.488	0.492
OnBMF	**0.635**	**0.716**	**0.703**	**0.707**	**0.698**

Acknowledgement. This work was supported by the NSFC (No. 61370025), and the Strategic Leading Science and Technology Projects of CAS (No.XDA06030200), 973 project (No. 2013CB329605) and Australia ARC Discovery Project (DP140102206).

4. REFERENCES

[1] J. Abernethy, K. Canini, J. Langford, and A. Simma. Online collaborative filtering. In *Proceedings of Technical Report of University of California, Berkeley*, 2009.

[2] X. Liu, Q. Hey, Y. Tiany, W.-C. Lee, J. McPhersony, and J. Han. Event-based social networks linking the online and offline social worlds. In *Proceedings of KDD-12*, 2012.

[3] Z. Qiao, P. Zhang, Y. Cao, C. Zhou, and B. Fang. Combining heterogenous social and geographical information for event recommendation. In *Proceedings of AAAI-14*, pages 145–151, 2014.

[4] Z. Qiao, P. Zhang, W. Niu, C. Zhou, P. Wang, and L. Guo. Online nonparametric max-margin matrix factorization for collaborative prediction. In *Proceedings of ICDM-14*, 2014.

[5] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of UAI-14*, 2009.

[6] P. Zhang, C. Zhou, P. Wang, B. J. Gao, X. Zhu, and L. Guo. E-tree: An efficient indexing structure for ensemble models on data streams. In *TKDE*, pages 461–474, 2015.