ples. 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.

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

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.

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

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

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.