Proc. of the 2nd International Research Conference on Emerging Information Technology and Engineering Solutions (EITES 2015), Pune, India, pp. 23 – 26, Feb. 20-21, 2015.

Bounded-SVD: A Matrix Factorization Method with Bound Constraints for Recommender Systems

Bang Hai Le
Intelligent Computer Entertainment Laboratory, GISE, Ritsumeikan University, Shiga, Japan
is0097sk@ed.ritsumei.ac.jp

Kien Quang Nguyen
Intelligent Computer Entertainment Laboratory, GISE, Ritsumeikan University, Shiga, Japan
is0050sv@ed.ritsumei.ac.jp

Ruck Thawonmas
Intelligent Computer Entertainment Laboratory, GISE Ritsumeikan University, Shiga, Japan
ruck@ci.ritsumei.ac.jp

***Abstract*— In this paper, we present a new matrix factorization method for recommender system problems, named bounded-SVD, which utilizes the constraint that all the ratings in the rating matrix are bounded within a pre-determined range. In our proposed method, the bound constraints are included in the objective function so that both the task of minimizing errors and the constraints are taken into account during the optimization process. For evaluation, we compare the performance of bounded-SVD with an existing method, called Bounded Matrix Factorization (BMF), which also uses the bound constraints on the ratings. The results on major real-world recommender system datasets show that our method outperforms BMF in almost cases and it is also faster and more simple to implement than BMF. Moreover, the way the bound constraints are integrated in bounded-SVD can also be applied to other optimization problems with bound constraints as well.**

Keywords—Collaborative filtering; Matrix factorization; Bound constraints; Recommender systems; Stochastic gradient descent.

I. Introduction

Matrix factorization [1] is one of the state-of-the-art collaborative filtering approaches to recommender systems. In matrix factorization method, the data of interest are formed as a sparse rating matrix whose rows and columns correspond to the users and the items, and each cell is the evaluation (or rating) of a user on an item. In order to predict the unknown ratings, the rating matrix is estimated as the product of two smaller size matrices called the user feature matrix and the item feature matrix, respectively, with the number of latent features decided in advance. Once these feature matrices are learned, they can be combined to estimate the unknown ratings of the original rating matrix. Matrix factorization has been proved to have high prediction performance in many domains of recommender systems while not requiring much additional information about users and items [1].

There are many matrix factorization methods for recommender systems proposed over the last decade. In those methods, the main task is to learn the user feature matrix and the item feature matrix so as to minimize the estimation errors on the set of known ratings. Different methods use different constraints and optimization techniques for the feature matrices. For example, the SGD method proposed by Funk [2] used a stochastic gradient descent technique for optimization and a constraint that biases the search to the feature matrices with small element values. Bias-SVD [3] and SVD++ [3] proposed by Koren extended matrix factorization with baseline estimation and implicit ratings, and used gradient descent to learn the feature matrices. Another well-known matrix factorization approach is non-negative matrix factorization (NMF) [4] that tries to learn the feature matrices with non-negative elements.

In many recommender system problems, their ratings are typically bounded within a range of possible rating values (e.g., one to five in a five-star rating system). For the estimated ratings which are out of bounds, the methods proposed so far often just artificially truncated them to fit within the bounds. Recently, a new matrix factorization method, called Bounded Matrix Factorization (BMF), which takes into account the bound constraints on the ratings, has been proposed by R. Kannan et al. [5]. In BMF method, the rating bounds were used as the constraints for optimizing the latent feature matrices so that the estimated ratings are guaranteed to be within the bounds. As reported by its authors, the method outperformed many state-of-the-art algorithms for recommender systems on some real-world datasets.

However, in BMF, in order to guarantee that the estimated ratings are always in the bounds, the method has to compute the lower bound and upper bound vectors for each element of the features matrices and only updates the element if the new value is bounded by the two bound vectors. This process can make BMF take very long time to finish an iteration, and in many cases, it wastes the computational resources because the new value for the element does not always satisfy the bound constraints. In this paper, we introduce a new matrix factorization method with bound constraints called bounded-SVD. In our method, the bound constraints are included in the objective function of the problem, so by minimizing the objective function, the method not only minimizes the estimation errors but also biases the search towards the parameter values that satisfy the bound constraints. The results on major real-world datasets show that our method outperforms BMF and is faster to converge and more simple to implement than BMF.

II. Notations

In the following sections, we use the notations as in Table I to describe the proposed method.

TABLE I. NOTATIONS

Notations	Meaning
N	Number of users
M	Number of items
K	Number of latent features
$\boldsymbol{R} \in \mathbb{R}^{N\times M}$	Ratings matrix
$\boldsymbol{P} \in \mathbb{R}^{N\times K}$	User feature matrix
$\boldsymbol{Q} \in \mathbb{R}^{K\times M}$	Item feature matrix
u	A user
i	An item
r_{ui}	The rating of item i made by user u
$\boldsymbol{p}_u{}^T \in \mathbb{R}^{1\times K}$	The feature vector of user u (the u^{th} row of $\boldsymbol{P}$)
$\boldsymbol{q}_i \in \mathbb{R}^{K\times 1}$	The feature vector of item i (the i^{th} column of $\boldsymbol{Q}$)
$\boldsymbol{S_r}$	Set of known ratings
$\boldsymbol{S}$	Set of known and unknown ratings
r_{min}	Minimal rating
r_{max}	Maximal rating

III. PROPOSED METHOD

A. Objective function

In our method, we try to minimize the estimation errors on the known ratings with the constraint that the estimated ratings are bounded between the minimal and the maximal rating. Therefore, the problem can be written as

$$\min_{P,Q} \sum_{r_{ui}\in \boldsymbol{S_r}} (r_{ui} - \boldsymbol{p}_u^T\boldsymbol{q}_i\ )^2$$

$$\text{subject to} \quad r_{min} \leq \boldsymbol{p}_u^T\boldsymbol{q}_i\ \leq r_{max}$$

In typical matrix factorization methods, in order to avoid the elements of the feature matrices getting too large value (a sign of overfitting), a regularization term is often added to the objective function. In our case with the bound constraints, the regularization term should penalize the estimations which go outside of the bounds. Therefore, we came up with a new objective function for bounded-SVD as shown in (1).

$$E = \sum_{r_{ui}\in \boldsymbol{S_r}} (r_{ui} - \boldsymbol{p}_u^T\boldsymbol{q}_i\ )^2 + \lambda\left(e^{\alpha(\boldsymbol{p}_u^T\boldsymbol{q}_i - r_{max})} + e^{\alpha(r_{min} - \boldsymbol{p}_u^T\boldsymbol{q}_i)}\right) \quad (1)$$

In (1), α and λ are constants and the term

$$R = e^{\alpha(\boldsymbol{p}_u^T\boldsymbol{q}_i - r_{max})} + e^{\alpha(r_{min} - \boldsymbol{p}_u^T\boldsymbol{q}_i)}$$

has the characteristic that its value is small if $\boldsymbol{p}_u^T\boldsymbol{q}_i \in [r_{min}, r_{max}]$ and increases very quickly once $\boldsymbol{p}_u^T\boldsymbol{q}_i$ goes outside of the bounds. Fig. 1 and Fig. 2 visualize that effect when $r_{min} = 1, r_{max} = 5$ and α is set to 1 and 2 respectively.

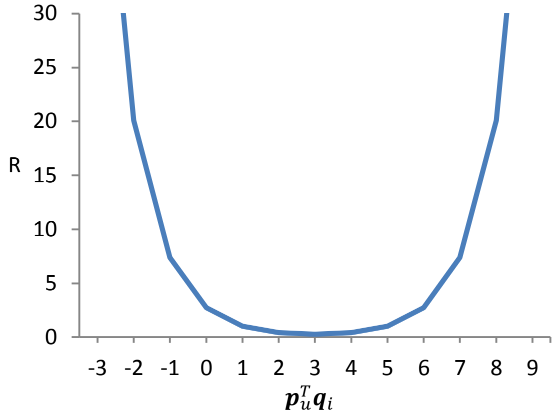

Fig. 1. The change of the regularization term R when $r_{min} = 1, r_{max} = 5$ and $\alpha = 1$

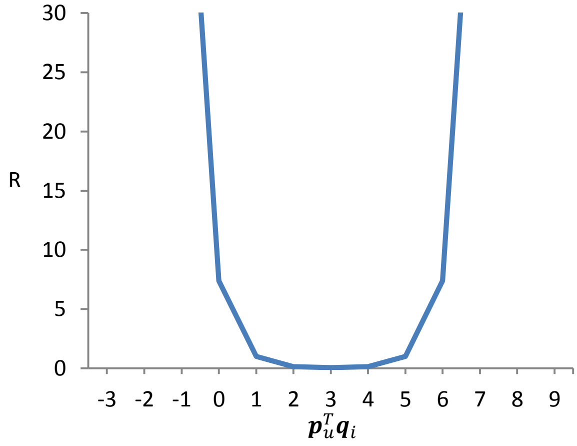

Fig. 2. The change of the regularization term R when $r_{min} = 1, r_{max} = 5$ and $\alpha = 2$

B. Training process

Given the objective function is as (1), we used stochastic gradient descent algorithm to optimize the elements of the feature matrices. For each known rating $r_{ui} \in \boldsymbol{S_r}$, the feature vectors $\boldsymbol{p}_u^T$, $\boldsymbol{q}_i$ were updated by

$$\boldsymbol{p}_u = \boldsymbol{p}_u + \eta\boldsymbol{q}_i[e_{ui} - \alpha H\ ] \quad (2)$$

$$\boldsymbol{q}_i = \boldsymbol{q}_i + \eta\boldsymbol{p}_u[e_{ui} - \alpha H\ ] \quad (3)$$

where

- η is the learning rate
- $e_{ui} = r_{ui} - \boldsymbol{p}_u^T\boldsymbol{q}_i$
- $H = e^{\alpha(\boldsymbol{p}_u^T\boldsymbol{q}_i\ - r_{max})} - e^{\alpha(r_{min} - \boldsymbol{p}_u^T\boldsymbol{q}_i\ )}$

That process was iterated until the stop criteria (as discussed in the experiments section) were met.

C. Implementation

We implemented bounded-SVD in C++ by using a C++ implementation of GraphChi [6] – an open source disk-based framework for graph computation. GraphChi can run very fast on large-scale recommender system datasets with millions of ratings while not requiring much computation resources. The pseudo-code for bounded-SVD is shown in Fig. 3.

```
READ R ∈ ℝ^{N×M}, r_min, r_max, K
INIT features matrices P ∈ ℝ^{N×K}, Q ∈ ℝ^{K×M}
WHILE stopping criteria not met
   FOR each r_ui ∈ S_r
      update p_u using (2)
      update q_i using (3)
   END FOR
END WHILE
```

Fig. 3. The pseudo-code for bounded-SVD

IV. EXPERIMENTS

A. Experiment settings

In order to compare the performance of bounded-SVD with the existing method BMF, we followed the same experiment framework used in [5]. The datasets we used for the experiments are: Jester [7], MovieLens 10M [8], Online dating [9] and Book crossing [10]. Table II shows the characteristics of each dataset. The values for the number of latent features (K) used in the experiments were 10, 20 and 50. For each setting of K, we ran the algorithm 5 times; each time the dataset was randomly divided to training set, validation set and test set whose proportions were 85%, 5% and 10% respectively.

TABLE II. DATASETS INFORMATION

Dataset	*Users*	*Items*	*Ratings (millions)*	*Density*	*Ratings range*
Jester	73,421	100	4.1	0.5584	[−10,10]
Movielens	71,567	10,681	10	0.0131	[1,5]
Dating	135,359	168,791	17.3	0.0007	[1,10]
Book	278,858	271,379	1.1	0.00001	[1,10]

The final result for each setting of K was the average RMSE on the test sets of the corresponding 5 runs. The RMSEs were computed without truncating the estimated ratings as shown in (4). In that formula, $\boldsymbol{T}_r$ is the set of known ratings of a validation/test set and $n(\boldsymbol{T}_r)$ is the number of known ratings in $\boldsymbol{T}_r$.

$$\sqrt{\frac{\sum_{r_{ui}\in\boldsymbol{T}_r}\left(r_{ui}-\boldsymbol{p}_u^T\boldsymbol{q}_i\right)^2}{n(\boldsymbol{T}_r)}} \qquad (4)$$

In each experiment run, we decided to stop updating the feature matrices once the RMSE on the validation set started increasing or decreased with too minor amount (i.e. lower than *1e-5*).

In matrix factorization methods, a reasonable initialization of the feature matrices is very important for the algorithm to converge to a good solution and there is no exception with our method. Therefore, in the experiments, we tried with a random and a baseline initialization method which were proposed in [5].

B. Parameters tunning

In this section, we discuss about how to choose the values for the hyper parameters α, λ and η. In bounded-SVD, the role of α and λ is to balance the task of minimizing the estimation errors and the bound constraints. Reasonable choices for α and λ are $\alpha = 1$ and $\lambda = 1$. The learning rate η can be set to a small enough value and in our experiments, it was gradually reduced by 5% after each iteration. The initialized learning rate values for each initialization method on each dataset are shown in Table III.

TABLE III. THE INITIALIZED LEARNING RATE VALUES FOR EACH DATASET

Dataset	*η (random)*	*η (baseline)*
Jester	0.0005	0.0005
Movielens	0.005	0.001
Dating	0.0005	0.0005
Book crossing	0.001	0.0001

C. Experiment results

Table IV shows the average RMSEs of bounded-SVD on the benchmark datasets compared with BMF method, the lower RMSE the better. We followed the same experiment settings reported in [5] so the results of BMF were re-used for comparison.

TABLE IV. DATASETS INFORMATION

Dataset	*K*	*BMF baseline*	*bounded-SVD baseline*	*BMF random*	*bounded-SVD random*
Jester	10	4.3320	**4.3204**	4.6289	**4.3007**
Jester	20	4.3664	**4.3251**	4.7339	**4.2288**
Jester	50	4.5046	**4.3189**	4.7180	**4.2498**
Movielens	10	0.8531	0.8638	0.8974	**0.8337**
Movielens	20	0.8526	0.8634	0.8931	**0.8272**
Movielens	50	0.8553	0.8655	0.8932	**0.8387**
Dating	10	1.9309	**1.8857**	2.1625	**1.9751**
Dating	20	1.9337	**1.8871**	2.1617	**1.9654**
Dating	50	1.9434	**1.8963**	2.1642	**1.9535**
Book	10	1.9355	**1.8081**	2.8137	**2.3479**
Book	20	1.9315	**1.8319**	2.4652	**2.0826**
Book	50	1.9405	**1.8596**	2.1269	**1.8221**

From Table IV, we can see that with random initialization, bounded-SVD outperformed BMF on all the benchmark datasets, and in some cases, bounded-SVD with random initialization was better than BMF with baseline initialization. With baseline initialization setting, except Movielens 10M dataset on which the performance of bounded-SVD was slightly worse than BMF, in other datasets bounded-SVD performed better.

Additionally, it is clear that given the same configuration of computational resources (i.e., processors, memory), bounded-SVD should run faster than BMF. That is because for each element of the feature matrices, in order to fit the estimated ratings within the bounds, BMF has to compute the lower and upper bound vectors whose size are the number of users or items. However, in bounded-SVD, the bound constraints are included in the objective function so that by minimizing the objective function, the bound constraints are naturally taken into account.

V. Conclusions

In this paper, we have introduced a new matrix factorization method with bound constraints called bounded-SVD. In this method, the bound constraints are included in the objective function and are naturally taken into account during the optimization process. The experiment results on real world recommender system datasets showed that our proposed method was better than an existing method both in performance and convergence speed.

Our proposed method is also flexible to include additional information. Therefore, in future work, we would like to extend the current model by integrating additional information about users and items. In the current version of bounded-SVD, the learning rate was manually adjusted and adapted using a simple rule. Although the performance was still good but it will be better if we can come up with an automatic tuning method and more flexible adaptive rule for the learning rate. That is also left as future work.

In this paper, we only applied the proposed method to recommender systems problems but we believe that our method can be appropriate for other optimization problems with bound constraints as well.

References

[1] Koren, Y., Bell, R. and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30-37.

[2] S. Funk, "Netflix Update: Try This at Home," Dec. 2006; http://sifter.org/~simon/journal/20061211.html.

[3] Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 426-434.

[4] Cichocki, A. and Anh-Huy, P. 2009. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, *92*(3), 708-721.

[5] Kannan, R., Ishteva, M. and Park, H. 2014. Bounded matrix factorization for recommender system. *Knowledge and Information Systems*, *39*(3), 491-511.

[6] Kyrola, A., Blelloch, G. E. and Guestrin, C. 2012. GraphChi: Large-Scale Graph Computation on Just a PC. In *OSDI* (Vol. 12), 31-46.

[7] Goldberg, K. 2003. Jester collaborative filtering dataset. http://ieor.berkeley.edu/~goldberg/jester-data/.

[8] Movielens dataset. http://grouplens.org/datasets/movielens/.

[9] Brozovsky, L. and Petricek, V. 2007. Recommender system for online dating service. In *Proceedings of conference Znalosti 2007*, Ostrava, VSB.

[10] Ziegler, C. N., McNee, S. M., Konstan, J. A. and Lausen, G. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*, 22-32. ACM.