

# Incremental Methods in Collaborative Filtering for Ordinal Data

Elena Polezhaeva

Lomonosov Moscow State University, Moscow  
lena\_polejaeva@mail.ru

**Abstract.** In modern collaborative filtering applications initial data are typically very large (holding millions of users and items) and come in real time. In this case only incremental algorithms are practically efficient. In this paper a new algorithm based on the symbiosis of Incremental Singular Value Decomposition (ISVD) and Generalized Hebbian Algorithm (GHA) is proposed. The algorithm does not require to store the initial data matrix and effectively updates user/item profiles when a new user or a new item appears or a matrix cell is modified. The results of experiments show how root mean square error (RMSE) depends on the number of algorithm's iterations and data amount.

**Keywords:** Collaborative filtering, singular value decomposition, Generalized Hebbian algorithm, sparse matrix, large data, ordinal data, incremental data.

## 1 Introduction

Collaborative Filtering (CF) is used in recommender systems and Customer Relationship Management System (CRM) for personalization. Initial data are represented by a sparse matrix  $Y = (y_{ur})_{n \times d}$ , in which rows correspond to  $n$  users, columns — to  $d$  items (documents, films, etc.). Each matrix cell contains information about the usage of an item by a user. So matrix cell may be a rating, a sum paid by user, a mark about visiting site (0 or 1), etc.

The aim is to predict for any user his preferences towards items or, in other words, to fill any empty cell in the initial matrix.

The thin singular value decomposition (SVD) [1] is effective because it uses a compressed representation of the data. Reduced data (profiles) are formed for users and items, then not all data are held. In modern collaborative filtering applications initial data are typically very large (holding millions of users and items), sparse and come in real time. So the requirement of incrementality is very important. It should be possible to update stored data in case of appearing of a new user/item(1) or a new value in a matrix cell(2). Usually it is proposed to solve only one of these incremental problems. In works of Brand [1] new items and ratings are added efficiently. In paper [2] algorithm efficiently incorporates into the model new users or new ratings but doesn't handle the addition of new items.

In this paper an algorithm is proposed which updates stored data in both cases of incrementality. It is based on incremental singular value decomposition (ISVD) [1] and Generalized Hebbian algorithm (GHA) [3]. The advantage of using GHA is that only known elements of the initial matrix are involved in calculations. As a result it gives data without empty elements for which ISVD may be used. Two situations are considered, when  $Y$  contains real data and when  $Y$  contains ordinal data (for example, ratings), where  $\Omega \subseteq \{1, \dots, n\} \times \{1, \dots, d\}$  is a set of non-empty elements in  $Y$ . Section 2 describes ISVD where data in  $Y$  are real. Section 3 outlines a GHA algorithm for ordinal data. Section 4 presents experiments which show how root mean square error (RMSE) [2] depends on the number of the algorithm's iterations, the number of users and the number of known elements in the initial matrix. When the number of users increases the rate of convergence also increases (from 1000 iterations on 600 users to 40 iterations on 940 users). When a data amount increases the number of iterations decreases thus minimizing the working time of the algorithm.

## 2 Incremental Singular Value Decomposition (ISVD)

SVD is a decomposition of a matrix  $Y$  into two orthogonal matrixes  $U$  and  $R$  and a diagonal matrix  $S$ , so that  $USR^T = Y, U^T Y R = S$ . The elements of  $S$  are called singular values, the columns of  $U$  and  $R$  are called left and right singular vectors respectively. Usually all but  $r$  largest singular values with respective singular vectors in  $U$  and  $R$  are rejected which leads to a considerable data compression at the cost of negligible loss of information. This rank  $r$  approximation is used for further work.

### 2.1 Problem Definition

Let  $USR^T = Y$  be SVD of the matrix  $Y_{n \times d}$  of the rank  $r$  where  $Y$  is sparse and  $U^T U = R^T R = I$ . The aim is to modify  $U, S, R$  to get new SVD  $Y + AB^T$  where  $A$  and  $B$  have  $c$  columns ( $A_{n \times c}, B_{d \times c}$ ). In a special case  $c = 1$  and  $A$  and  $B$  are vectors. Deletion and addition of a row or a column may decrease or increase respectively the rank of the matrix  $Y$  by 1. Let we have two column vectors  $a$  and  $b$  and known SVD  $USR^T = Y$ , where  $b$  is a binary vector which indicates which columns should be modified and a vector  $a$  is derived from the requirement delete or add new values or modify values in chosen rows and columns. The aim is to find SVD of the matrix  $Y + ab^T$  [1].

For example, when a new column is added, the known decomposition may be written with addition of zero column:  $US[R^T \ 0] = [Y \ 0]$ , then  $U'S'R'^T = [Y \ c]$  — desired decomposition, where  $U', S', R'$  are to be found,  $a = c, b = [0, \dots, 0, 1]^T$ .

### 2.2 New SVD

Let us find the SVD of the following block matrix:

$$Y + AB^T = [U, A] \begin{bmatrix} S & 0 \\ 0 & I \end{bmatrix} [R, B]^T. \tag{1}$$

Let  $P$  be an orthogonal basis of column space  $\tilde{A} = (I - UU^T)A = A - UU^T A$ . Then QR-decomposition is applied to  $\tilde{A}$ :  $\tilde{A} = PR_A$ , where  $P$  is orthonormal,  $R_A$  is upper triangular,  $R_A = P^T(I - UU^T)A$ . QR-decomposition of matrix  $[U, A]$  is:

$$[U, P] \begin{bmatrix} I & U^T A \\ 0 & R_A \end{bmatrix} = [U, A], \tag{2}$$

which can be found from modified Gram–Schmidt orthogonalization [4].

Similarly, let  $Q$  be an orthogonal basis of column space  $B - RR^T B$ , then  $QR_B = (I - RR^T)B$  where  $Q$  is orthonormal,  $R_B$  is upper triangular,  $R_B = Q^T(I - RR^T)B$ .

Let us represent  $Y + AB^T$  as a product of three matrixes:

$$\begin{aligned} Y + AB^T &= [U, P] \begin{bmatrix} I & U^T A \\ 0 & R_A \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & R^T B \\ 0 & R_B \end{bmatrix}^T [R, Q]^T \\ &= [U, P] \underbrace{\left( \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} U^T A \\ R_A \end{bmatrix} \begin{bmatrix} R^T B \\ R_B \end{bmatrix}^T \right)}_K [R, Q]^T; \\ K &= \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} (U^T A)(B^T R) & (U^T A)R_B^T \\ R_A(B^T R) & R_A R_B^T \end{bmatrix}. \end{aligned}$$

Let  $U'S'R'^T$  be SVD of matrix  $K$  of rank  $(r + c)$ . Making diagonalization of  $K$ :  $U'^T K R' = S'$ , new matrixes  $U', S', R'$  are formed. Then the modification of SVD from rank  $r$  to rank  $(r + c)$  can be written as:

$$Y + AB^T = ([U \ P]U')S'([R \ Q]R')^T. \tag{3}$$

But instead of making operations with matrixes with big sizes new SVD of  $Y + AB^T$  containing five matrixes is used:

$$U_{n \times r} U'_{r \times r} S_{r \times r} R'^T_{r \times r} R^T_{r \times d}, \tag{4}$$

where  $UU', RR', U, U'$  are orthonormal.

If only one column or row are added following modifications are done: for new SVD ( $a$  —  $n$ -dimensional vector,  $b$  —  $d$ -dimensional vector) modified Gram–Schmidt orthogonalization is used:

$$\begin{aligned} z &= U^T a; & p &= a - Uz; \\ w &= R^T b; & q &= b - Rw; \end{aligned}$$

Then  $K$  have simple representation and effective diagonalization may be found:

$$K = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} zw^T & z\|q\| \\ \|\|p\|\|w^T & \|\|p\|\|\|q\| \end{bmatrix}.$$

### 3 Generalized Hebbian Algorithm

Algorithm GHA [3] calculates SVD for sparse data matrix  $Y$ . Its advantage is that vectors of SVD are found from  $\Omega \subseteq \{1, \dots, n\} \times \{1, \dots, d\}$  a set of only non-empty elements of  $Y$ . This method uses the gradient descent iterations in contrast to classical SVD. As a result it gives data without empty elements for which ISVD defined in previous section may be used.

Matrix factorization is used to estimate missing values [2]:

$$u_i r_j = \sum_{p=1}^L u_{ip} r_{pj},$$

where  $L$  — number of features for factorization.

Functional is minimized, where  $Y$  is given and  $U, R, \beta$  are to be found:

$$\sum_{(i,j) \in \Omega} (\beta_{y_{ij}} - u_i r_j)^2 \rightarrow \min_{u,r,\beta}, \tag{5}$$

where  $\beta_{y_{ij}}$  is a parameter which approximates ordinal ratings in  $Y$ ;  $u_i$  is the  $i$ -th row in  $U$ ;  $r_j$  is the  $j$ -th column in  $R$ ;  $[y_{ij} = m]$  equals 1 if  $y_{ij} = m$  and equals 0 if  $y_{ij} \neq m$ .

Let us represent:

$$\bar{y}_m = \sum_{(i,j) \in \Omega} [y_{ij} = m].$$

$\beta_m$  is found analytically:

$$\beta_m = \frac{1}{\bar{y}_m} \sum_{(i,j) \in \Omega} [y_{ij} = m] u_i r_j,$$

Error in a rating calculated by the model is:

$$\varepsilon_{ij} = \beta_{y_{ij}} - u_i r_j.$$

In (5) optimal matrixes  $U$  and  $R$  minimize the sum of squared errors over non-empty elements of  $Y$ . To avoid overfitting regularization is applied by penalizing the square of the Euclidean norm of weights.

The optimization task is:

$$\sum_{(i,j) \in \Omega} (\beta_{y_{ij}} - u_i r_j)^2 + \lambda_1 \sum_{i=1}^n \|u_i\|^2 + \lambda_2 \sum_{j=1}^d \|r_j\|^2 \rightarrow \min_{u,r,\beta},$$

where  $\lambda_1, \lambda_2 \geq 0$  — regularization parameters.

The incremental gradient descent method is used for finding new weights. Such iterations are made:

$$\varepsilon_{ij} = \beta_{y_{ij}} - u_i r_j,$$

$$\tilde{u}_{ip} = (1 - \eta\lambda_1)u_{ip} + \eta \sum_j r_{pj}\varepsilon_{ij}, \tag{6}$$

$$\tilde{r}_{pj} = (1 - \eta\lambda_2)r_{pj} + \eta \sum_i u_{ip}\varepsilon_{ij}, \tag{7}$$

$$\beta_m = \frac{1}{\bar{y}_m} \sum_{(i,j) \in \Omega} [y_{ij} = m]u_i r_j,$$

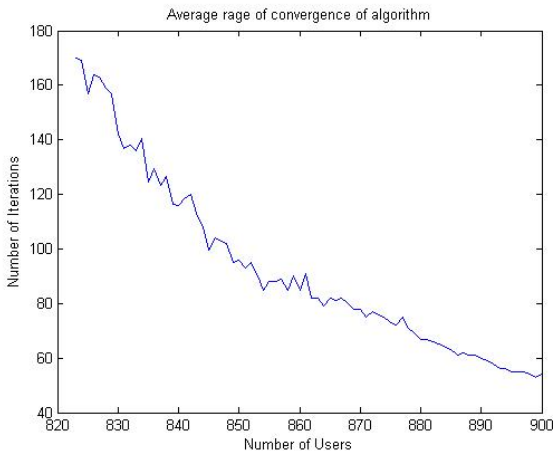
where  $\eta$  is learning rate.

The aim is to minimize RMSE (root mean square error):

$$RMSE = \sqrt{\frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \varepsilon_{ij}^2};$$

The result of GHA is data without empty elements for which SVD is used. On the first step of the algorithm matrixes  $U$  and  $R$  are initialized with small random values. On each step of algorithm we update  $u_i$  —  $i$ -th row of  $U$  and  $r_j$  —  $j$ -th column of  $R$  from (6) and (7). If RMSE doesn't increase on next iterations the algorithm stops.

When a new user is added the matrix  $R$  is not updated but the matrix  $U$  is initialized with small values. In this case only (6) is updated.



**Fig. 1.** Graphic shows how the number of iterations of the algorithm depends on the number of users

## 4 Experiments

Experiments were made on MovieLens data: 943 users, 1682 items, 100000 known ratings. Time factor and data amount have an effect on the rate of convergence of the algorithm. The best results were achieved on rank-15 basis,  $\eta = 0.001$  and  $\lambda = 0.025$ : after 41 iterations  $RMSE = 0.923$ . Brand in his work [5] achieved  $MAE = 0.7910$  using only ISVD and rank-14 basis. It is very interesting that all  $\beta \geq 0$  and they increase (even if we don't impose conditions on  $\beta$ ):  $\beta_1 = 2.52$ ;  $\beta_2 = 3.03$ ;  $\beta_3 = 3.28$ ;  $\beta_4 = 3.63$ ;  $\beta_5 = 3.96$ . When the number of users increases the rate of convergence also increases (from 1000 iterations on 600 users to 40 iterations on 940 users) (Fig.1). When a data amount increases the number of iterations decreases thus minimizing the working time of the algorithm.

## 5 Results

The proposed method solves two incremental problems in CF and enables to update users' and items' profiles when new rows and columns are added and elements in  $Y$  are modified. Two tasks have been solved: when data are real or ordinal. When a new user is added there is no need to retrain the entire model. Addition of a new item is a simple task in which a diagonalizable matrix is calculated.

## References

1. Brand, M.: Fast Low-rank modifications of the thin singular value decomposition. *Linear Algebra and Its Applications* 415(1), 20–30 (2006)
2. Takacs, G., Pillaszy, I., Nemeth, B., Tikk, D.: Scalable Collaborative Filtering Approaches for Large Recommender Systems. *The Journal of Machine Learning Research* 10, 623–656 (2009)
3. Gorrell, G.: Generalized Hebbian Algorithm for Incremental Singular Value Decomposition in Natural Language Processing. In: *Proceedings of EACL* (2006)
4. Golub, G.: *Matrix Computations*, p. 728. Johns Hopkins University Press (1996)
5. Brand, M.: Fast online SVD revisions for lightweight recommender systems. In: *SIAM International Conference on Data Mining (SDM)* (2003)