Context-Aware Recommender System Based on Boolean Matrix Factorisation

Marat Akhmatnurov and Dmitry I. Ignatov

National Research University Higher School of Economics, Moscow
dignatov@hse.ru

Abstract. In this work we propose and study an approach for collaborative filtering, which is based on Boolean matrix factorisation and exploits additional (context) information about users and items. To avoid similarity loss in case of Boolean representation we use an adjusted type of projection of a target user to the obtained factor space. We have compared the proposed method with SVD-based approach on the MovieLens dataset. The experiments demonstrate that the proposed method has better MAE and Precision and comparable Recall and F-measure. We also report an increase of quality in the context information presence.

Keywords: Boolean Matrix Factorisation, Formal Concept Analysis, Recommender Algorithms, Context-Aware Recommendations

1 Introduction

Recommender Systems have recently become one of the most popular applications of Machine Learning and Data Mining. Their primary aim is to help users to find proper items like movies, books or goods within an underlying information system. Collaborative filtering recommender algorithms based on matrix factorisation (MF) techniques are now considered industry standard [1]. The main assumption here is that similar users prefer similar items and MF helps to find (latent) similarity in the reduced space efficiently.

Among the most often used types of MF we should definitely mention Singular Value Decomposition (SVD) [2] and its various modifications like Probabilistic Latent Semantic Analysis (PLSA) [3]. However, several existing factorisation techniques, for example, non-negative matrix factorisation (NMF) [4] and Boolean matrix factorisation (BMF) [5], seem to be less studied in the context of Recommender Systems. Another approach similar to MF is biclustering, which has also been successfully applied in recommender system domain [6,7]. For example, Formal Concept Analysis (FCA) [8] can be also used as a biclustering technique and there are several examples of its applications in the recommender systems domain [9,10]. A parameter-free approach that exploits a neighbourhood of the object concept for a particular user also proved its effectiveness [11]; it has a predecessor based on object-attribute biclusters [7] that also capture the neighbourhood of every user and item pair in an input formal context. Our previous approach based on FCA exploits Boolean factorisation based on formal concepts and follows user-based k-nearest neighbours strategy [12].

The aim of this study is to continue comparing the recommendation quality of several aforementioned techniques on the real dataset and investigation of methods' interrelationship and applicability. In particular, in our previous study, it was especially interesting to conduct experiments and compare recommendation quality in case of a numeric input matrix and its scaled Boolean counterpart in terms of Mean Absolute Error (MAE) as well as Precision and Recall. Our previous results showed that the BMF-based approach is of comparable quality with the SVD-based one [12]. Thus, one of the next steps is definitely usage of auxiliary information containing users' and items' features, i.e. so called context information (for BMF vs SVD see section 4).

Another novelty of the paper is defined by the fact that we have adjusted the original Boolean projection of users to the factor space by support-based weights that results in a sufficient quality increase. We also investigate the approximate greedy algorithm proposed in [5] in the recommender setting, which tends to generate factors with large number of users, and more balanced (in terms of ratio between users' and items' number per factor) modification of the Close-by-One algorithm [13].

The practical significance of the paper is determined by the demand of recommender systems' industry, that is focused on gaining reliable quality in terms of average MAE.

The rest of the paper consists of five sections. Section 2 is an introductory review of the existing MF-based approaches to collaborative filtering. In section 3 we describe our recommender algorithm which is based on Boolean matrix factorisation using closed sets of users and items (that is FCA). Section 4 contains results of experimental comparison of two MF-based recommender algorithms by means of cross-validation in terms of MAE, Precision, Recall and F-measure. The last section concludes the paper.

2 Introductory review

In this section we briefly describe two approaches to the decomposition of real-valued and Boolean matrices. In addition we provide the reader with the general scheme of user-based recommendation that relies on MF and a simple way of direct incorporation of context information into MF-based algoritms.

2.1 Singular Value Decomposition

Singular Value Decomposition (SVD) is a decomposition of a rectangular matrix $A \in \mathbb{R}^{m \times n}(m > n)$ into a product of three matrices

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T, \tag{1}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{n\times n}$ is a diagonal matrix such that $\Sigma = diag(\sigma_1, \dots, \sigma_n)$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$. The columns of the matrix U and V are called singular vectors, and the numbers σ_i are singular values.

In the context of recommendation systems rows of U and V can be interpreted as vectors of user's and items's attitude to a certain topic (factor), and the corresponding singular values as importance of the topic among the others. The main disadvantages are the dense outputted decomposition matrices and negative values of factors which are difficult to interpret.

The advantage of SVD for recommendation systems is that this method allows to obtain a vector of user's attitude to certain topics for a new user without SVD decomposition of the whole matrix.

The computational complexity of SVD according to [2] is $O(mn^2)$ floating-point operations if $m \geq n$ or more precisely $2mn^2 + 2n^3$.

2.2 Boolean Matrix Factorisation based on FCA

Description of FCA-based BMF. Boolean matrix factorisation (BMF) is a decomposition of the original matrix $I \in \{0,1\}^{n\times m}$, where $I_{ij} \in \{0,1\}$, into a Boolean matrix product $P \circ Q$ of binary matrices $P \in \{0,1\}^{n\times k}$ and $Q \in \{0,1\}^{k\times m}$ for the smallest possible number of k. We define Boolean matrix product as follows:

$$(P \circ Q)_{ij} = \bigvee_{l=1}^{k} P_{il} \cdot Q_{lj}, \tag{2}$$

where $\bigvee$ denotes disjunction, and $\cdot$ conjunction.

Matrix I can be considered a matrix of binary relations between set X of objects (users), and a set Y of attributes (items that users have evaluated). We assume that xIy iff the user x evaluated object y. The triple (X, Y, I) clearly forms a formal context[1].

Consider a set $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$, a subset of all formal concepts of context (X, Y, I), and introduce matrices $P_\mathcal{F}$ and $Q_\mathcal{F}$:

$$(P_\mathcal{F})_{il} = \begin{cases} 1, i \in A_l, \\ 0, i \notin A_l, \end{cases} \quad (Q_\mathcal{F})_{lj} = \begin{cases} 1, j \in B_l, \\ 0, j \notin B_l. \end{cases},$$

where (A_l, B_l) is a formal concept from F.

We can consider decomposition of the matrix I into binary matrix product $P_\mathcal{F}$ and $Q_\mathcal{F}$ as described above. The theorems on universality and optimality of formal concepts are proved in [5].

There are several algorithms for finding $P_\mathcal{F}$ and $Q_\mathcal{F}$ by calculating formal concepts based on these theorems [5]. The approximate algorithm we use for comparison (Algorithm 2 from [5]) avoids computation of all possible formal concepts and therefore works much faster [5]. Time estimation of the calculations in the worst case yields $O(k|G||M|^3)$, where k is the number of found factors, $|G|$ is the number of objects, $|M|$ is the number of attributes.

1 We have to omit basic FCA definitions; for more details see [8].

2.3 Contextual information

Contextual Information is a multi-faceted notion that is present in several disciplines. In the recommender systems domain, the context is any auxiliary information concerning users (like gender, age, occupation, living place) and/or items (like genre of a movie, book or music), which shows not only a user's mark given to an item but explicitly or implicitly describes the circumstances of such evaluation (e.g., including time and place) [15].

From the representational viewpoint context[2] can be described by a binary relation, which shows that a user or an item possesses a certain attribute-value pair. In case the contextual information is described by finite-valued attributes, it can be represented by finite number of binary relations; otherwise, when we have countable or continuous values, their domains can be split into (semi)intervals (cf. scaling in FCA). As a result one may obtain a block matrix:

$$I = \begin{bmatrix} R & C_{user} \\ C_{item} & O \end{bmatrix},$$

where R is a utility matrix of users' ratings to items, C_{user} represents context information of users, C_{item} contains context iformation of items and O is zero-filled matrix.

Table 1. Adding auxialiry (context) information

	Movies						Gender		Age		
	Brave Heart	Terminator	Gladiator	Millionaire from ghetto	Hot Snow	Godfather	M	F	0-20	21-45	46+
Anna	5		5	5		2		+	+		
Vladimir		5	5	3		5	+			+	
Katja	4		4	5		4		+		+	
Mikhail	3	5	5			5	+			+	
Nikolay			2		5	4	+				+
Olga	5	3	4	5				+	+		
Petr	5			4	5	4	+				+
Drama	+		+	+	+	+					
Action		+	+		+	+					
Comedy	+			+							

In case of more complex rating's scale the ratings can be reduced to binary scale (e.g., "*like/dislike*") by binary thresholding or by FCA-based scaling.

[2] In order to avoid confusion, please note that formal context is a different notion.

2.4 General scheme of user-based recommendations

Once a matrix of ratings is factorised we need to learn how to compute recommendations for users and to evaluate whether a particular method handles this task well.

For the factorised matrices already well-known algorithm based on the similarity of users can be applied, where for finding k nearest neighbors we use not the original matrix of ratings $R \in \mathbb{R}^{m \times n}$, but the matrix $I \in \mathbb{R}^{m \times f}$, where m is a number of users, and f is a number of factors. After the selection of k users, which are the most similar to a given user, based on the factors that are peculiar to them, it is possible, based on collaborative filtering formulas to calculate the prospective ratings for a given user.

After generation of recommendations the performance of the recommender system can be estimated by measures such as MAE, Precision and Recall.

Collaborative recommender systems try to predict the utility (in our case ratings) of items for a particular user based on the items previously rated by other users.

Memory-based algorithms make rating predictions based on the entire collection of previously rated items by the users. That is, the value of the unknown rating $r_{u,m}$ for a user u and item m is usually computed as an aggregate of the ratings of some other (usually, the k most similar) users for the same item m:

$$r_{u,m} = aggr_{\tilde{u} \in \tilde{U}} r_{\tilde{u},m},$$

where $\tilde{U}$ denotes a set of k users that are the most similar to user u, who have rated item m. For example, the function $aggr$ may be weighted average of ratings [15]:

$$r_{u,m} = \sum_{\tilde{u} \in \tilde{U}} sim(\tilde{u}, u) \cdot r_{\tilde{u},m} / \sum_{\tilde{u} \in \tilde{U}} sim(u, \tilde{u}). \tag{3}$$

The similarity measure between users u and $\tilde{u}$, $sim(\tilde{u}, u)$, is essentially an inverse distance measure and is used as a weight, i.e., the more similar users c and $\tilde{u}$ are, the more weight rating $r_{\tilde{u},m}$ will carry in the prediction of $r_{\tilde{u},m}$.

The similarity between two users is based on their ratings of items that both users have rated. There are several popular approaches: Pearson correlation, cosine-based, and Hamming-based similarities.

We further compare the cosine-based and normalised Hamming-based similarities:

$$sim_{cos}(u, v) = \sum_{m \in \tilde{M}} r_{um} \cdot r_{vm} / \left(\sum_{m \in \tilde{M}} r_{um}^2 \sum_{m \in \tilde{M}} r_{vm}^2 \right)^{1/2} \tag{4}$$

$$sim_{Ham}(u, v) = 1 - \sum_{m \in \tilde{M}} |r_{um} - r_{vm}| / |\tilde{M}|, \tag{5}$$

where $\tilde{M}$ is either the set of co-rated items (movies) for users u and v or the whole set of items.

To apply this approach in case of FCA-based BMF recommender algorithm we simply consider the user-factor matrices obtained after factorisation of the initial data as an input.

For the input matrix in Table 1 the corresponding decomposition is below:

$$\begin{bmatrix} 1&0&0&1&0&0&0&1&1\\ 0&1&0&0&0&1&1&0&0\\ 1&0&0&0&0&1&0&1&1\\ 0&1&0&0&0&1&1&0&0\\ 0&0&1&0&1&0&0&0&0\\ 1&0&0&1&0&0&0&1&1\\ 1&0&1&0&1&0&0&0&0\\ 1&0&0&0&1&0&0&1&0\\ 0&0&0&0&1&0&1&0&0\\ 1&0&0&0&0&0&0&0&0 \end{bmatrix} \circ \begin{bmatrix} 1&0&0&1&0&0&0&0&0&0&0\\ 0&1&1&0&0&1&1&0&0&1&0\\ 0&0&0&0&1&1&1&0&0&0&1\\ 1&0&1&1&0&0&0&1&1&0&0\\ 0&0&0&0&1&1&0&0&0&0&0\\ 0&0&1&0&0&1&0&0&0&1&0\\ 0&1&1&0&0&1&0&0&0&0&0\\ 1&0&1&1&0&0&0&0&0&0&0\\ 1&0&1&1&0&0&0&1&0&0&0 \end{bmatrix}$$

3 Proposed Approach

In contrast to [5], for the recommender setting we mostly interested whether the concepts of more balanced extent and intent size may give us an advantage and use the following criterion to this end:

$$W(A, B) = (2|A||B|)/(|A|^2 + |B|^2) \in [0; 1], \tag{6}$$

where (A, B) is a formal concept.

In subsection 2.2 we recalled that finding Boolean factors is reduced to the task of finding of covering formal concepts for the same input matrix.

To this end we modified *Close-by-One* ([13]). This algorithm traverses the tree of corresponding concept lattice in depth-first manner and returns the set of all formal concepts, which is redundant for the Boolean decomposition task. The deeper the algorithm is in the tree, the larger the intents are, and the smaller the extents of formal concepts. Thus, for every branch of the tree the proposed measure in eq. (6) is growing until some depth and then (in case the traverse continues) goes down.

The proposed modifications are: 1) the traverse of a certain branch is carried out until W is growing with the covered square (size of extent $\times$ size of intent); 2) at each iteration we do not accept concepts with intents that contained in the union of intents of previously generated concepts.

In case the intent of a certain concept is covered by its children (fulfilling condition 1), then this concept is not included into $\mathcal{F}$.

For *Close-by-One* there is a linear order $\succ$ on G. Assume $C \subset G$ is generated from $A \subset G$ by addition $g \in G$ ($C = A \cup \{g\}$) such that $g \succ \max(A)$, then the set C'' is called *canonically generated* if $min(C'' \setminus C) \succ g$.

Algorithm 1: Generation of balanced formal concepts

```
Data: Formal context (U, M, I)
Result: The set of balanced formal concepts F
foreach u ∈ U do
    A ← {u};
    stack.push(A');
    g ← u; g++;
    repeat
        if g ∉ U then
            if stack.Top ≠ ∅ then
                add (A'', A') to F;
                stack.Top ← ∅;
            while stack.Top = ∅ do
                g ← max(A);
                A ← A \ {g};
                stack.pop;
                g++;
        else
            B ← A ∪ {g};
            if (B'' is a canonical generation) and W(C'', C') ≥ W(A'', A') and
            |C'' × C'| ≥ |A'' × A'| then
                stack.Top ← (stack.Top \ C');
                A ← C;
            g++;
    until A ≠ ∅;
return F;
```

The obtained set $\mathcal{F}$ is still be redundant, that is why we further select factors with maximal coverage until we have covered the whole matrix or required percentage.

The main aim of factorisation is the reduction of computation steps and revealing latent similarity since users' similarities are computed in a factor space. As a projection matrix of user profiles to a factor space one may use *"user-factor"* from Boolean factorisation of utility matrix (P in (2)). However, in this case in the obtained user profiles most of the vector components are getting zeros, and thus we lose similarity information.

To smooth the loss effects we proposed the following weighted projection:

$$\tilde{P}_{uf} = \frac{I_{u\cdot} \cdot Q_{f\cdot}}{||Q_{f\cdot}||_1} = \frac{\sum\limits_{v \in V} I_{uv} \cdot Q_{fv}}{\sum\limits_{v \in V} Q_{fv}},$$

where $\tilde{P_{uf}}$ indicates whether factor f covers user u, $I_{u\cdot}$ is a binary vector describing profile of user u, $Q_{f\cdot}$ is a binary vector of items belonging to factor f

(the corresponding row of Q in decomposition eq. (2)). The coordinates of the obtained projection vector lie within $[0; 1]$.

For Table 1 the weighted projection is as follows:

$$\tilde{P} = \begin{bmatrix} 1 & \frac{1}{5} & 0 & 1 & 0 & \frac{1}{3} & \frac{1}{3} & 1 & 1 \\ 0 & 1 & \frac{1}{2} & \frac{1}{5} & \frac{1}{2} & 1 & 1 & \frac{1}{3} & \frac{1}{4} \\ 1 & \frac{3}{5} & \frac{1}{4} & \frac{4}{5} & \frac{1}{2} & 1 & \frac{2}{3} & 1 & 1 \\ 0 & 1 & \frac{1}{2} & \frac{1}{5} & \frac{1}{2} & 1 & 1 & \frac{1}{3} & \frac{1}{4} \\ 0 & \frac{2}{5} & 1 & 0 & 1 & \frac{2}{3} & \frac{1}{3} & 0 & 0 \\ 1 & \frac{1}{5} & 0 & 1 & 0 & \frac{1}{3} & \frac{1}{3} & 1 & 1 \\ 1 & \frac{2}{5} & 1 & \frac{1}{5} & 1 & \frac{1}{3} & \frac{1}{3} & \frac{2}{3} & \frac{1}{2} \\ 1 & \frac{2}{5} & \frac{1}{2} & \frac{2}{5} & 1 & \frac{2}{3} & \frac{2}{3} & 1 & \frac{3}{4} \\ 0 & \frac{2}{5} & \frac{1}{2} & \frac{1}{5} & 1 & \frac{2}{3} & 1 & \frac{1}{3} & \frac{1}{4} \\ 1 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & \frac{2}{3} & \frac{1}{2} \end{bmatrix}.$$

4 Experiments

The proposed approach and compared ones have been implemented in C++ and evaluated on the MovieLens-100k data set. This data set features 100000 ratings in five-star scale, 1682 Movies, Contextual information about movies (19 genres), 943 users (each user has rated at least 20 movies), and demographic info for the users (gender, age, occupation, zip (ignored)). The users have been divided into seven age groups: under 18, 18-25, 26-35, 36-45, 45-49, 50-55, 56+.

Five star ratings are converted to binary scale by the following rule:

$$I_{ij} = \begin{cases} 1, & R_{ij} > 3, \\ 0, & \text{else} \end{cases}$$

The scaled dataset is split into two sets according to bimodal cross-validation scheme [16]: training set and test set with a ratio 80:20, and 20% of ratings in the test set are hidden[3].

Measure of users similarity First of all, the influence of similarity has been compared. As we can see in the Fig. 4, Hamming distance based similarity is significantly better in terms of MAE and Precision. However it is worse in Recall and F-measure. Even though, given the superiority in terms of MAE (widely adopted in the RS community measure), we decided to use Hamming distance based similarity.

Projection into factor space In the series of tests the influence of projection method has been studied. The weighted projection keeps more information and as a result helps us to find similar user of higher accuracy. That is why this method has significant primacy in terms of all investigated measures of quality.

[3] This partition into test and training set is done 5 times resulting in 25 hidden submatrices and differs from the one provided by MovieLens group; hence the results might be different.

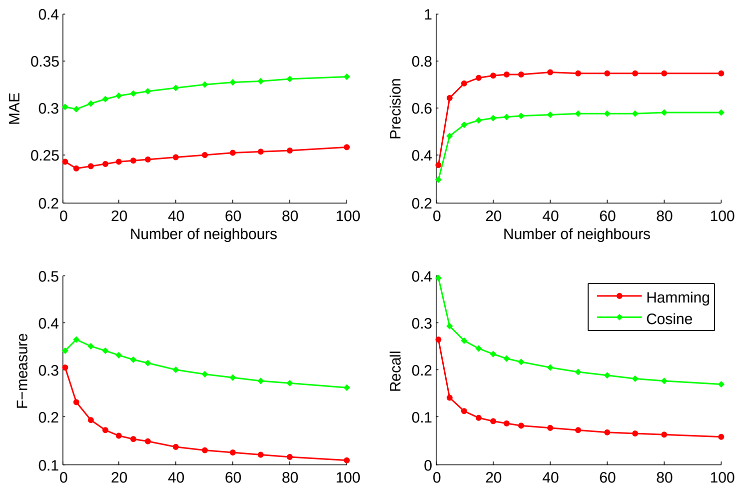

Fig. 1. Comparison of two similarity measures (BMF at 80% coverage)

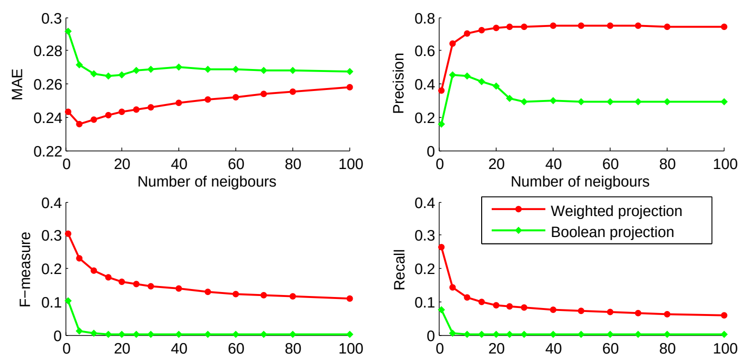

Fig. 2. Comparison of two types of projection into factor space

FCA-based algorithm and factors number The main studied algorithm to find Boolean factors as formal concepts is a modified algorithm *Close by One*. It was compared with greedy algorithm from [5] in terms of factors number and final RS quality measures.

Coverage	50%	60%	70%	80%	90%
Modified Close by One	168	228	305	421	622
Greedy algorithm	222	297	397	533	737

CbO covers the input matrix with a smaller count of factors, but it requires more time (in our experiments, 180 times more on average with one thread calculations). At the same time we have to admit that there is no influence to RS quality: thus Recall, Precision and MAE mainly differ only in the third digit.

Incorporation of context information and comparison with SVD For the SVD-based approach additional (context) information has been attached in a similar way, but there we use maximal rating (5 stars) in the attached columns and rows.

Coverage	50%	60%	70%	80%	85%	90%
BMF	168	228	305	421	508	622
BMF (No context information)	163	220	294	401	479	596
SVD	162	218	287	373	430	496
SVD (No context information)	157	211	277	361	416	480

BMF and SVD give similar number of factors, especially for small coverage; context information does not significantly change their number, but it gives an increase of precision (1-2% more accurate predictions in Table 4).

Table 2. Influence of contextual information (80% coverage)

Number of neighbours	Precision		Recall		F-measure		MAE	
	clean	cntxt	clean	cntxt	clean	cntxt	clean	cntxt
1	0.3589	0.3609	0.2668	0.2647	0.3061	0.3054	0.2446	0.2434
5	0.6353	0.6442	0.1420	0.1412	0.2321	0.2317	0.2371	0.2359
10	0.6975	0.7045	0.1126	0.1114	0.1938	0.1924	0.2399	0.2388
15	0.7168	0.7258	0.0994	0.0979	0.1746	0.1726	0.2422	0.2411
20	0.7282	0.7373	0.0911	0.0903	0.1619	0.1610	0.2442	0.2429
25	0.7291	0.7427	0.0861	0.0853	0.1540	0.1531	0.2457	0.2445
30	0.7318	0.7426	0.0823	0.0818	0.1480	0.1474	0.2472	0.2459
40	0.7342	0.7508	0.0767	0.0759	0.1389	0.1379	0.2497	0.2484
50	0.7332	0.7487	0.0716	0.0712	0.1304	0.1301	0.2518	0.2504
60	0.7314	0.7478	0.0682	0.0678	0.1247	0.1243	0.2536	0.2522
70	0.7333	0.7477	0.0658	0.0654	0.1208	0.1202	0.2552	0.2538
80	0.7342	0.7449	0.0632	0.0624	0.1164	0.1151	0.2567	0.2553
100	0.7299	0.7461	0.0590	0.0583	0.1092	0.1081	0.2594	0.2580

With a similar number or factors (SVD at 85% coverage and BMF at 80%) Boolean Factorisation results in smaller MAE and higher Precision where number of neighbours is not high. It can be explained by different nature of factors in these factorisation models.

5 Conclusion

In the paper we considered two modifications of Boolean matrix factorisation, which are suitable for Recommender Systems. They were compared on real datasets with the presence of auxiliary (context) information. We found out that MAE of our BMF-based approach is sufficiently lower than MAE of SVD-based approach for almost the same number of factor at fixed coverage level of BMF and SVD. The Precision of BMF-based approach is slightly lower when the number of neighbours is about a couple of dozens and comparable for the

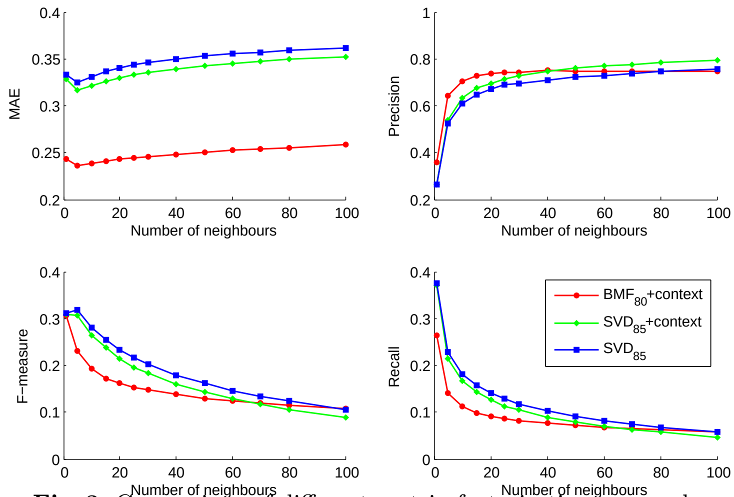

Fig. 3. Comparison of different matrix factorisation approaches

remaining part of the observed range. The Recall is lower than results in lower F-measure. The proposed weighted projection alleviates the information loss of original Boolean projection resulting in a substantial quality gain.

We also revealed that the presence of contextual information results in a small quality increase (about 1-2%) in terms of MAE, Recall and Precision.

We studied the influence of more balanced factors in terms of ratio of number of users and items. Finally, we should report that greedy approximate algorithm [5], even though that it results in more factors with larger user's component, is faster and demonstrates almost the same quality. So, its use is beneficial for recommender systems due to polynomial time computational complexity.

As a future research direction we would like to investigate the proposed approach with the previously ([9,6,10,7]) and recently introduced FCA-based ones ([11,12,17]). As for Boolean matrix factorisation in case of context-aware information, since the data can be naturally represented as multi-relational, we would like to continue our collaboration with the authors of the paper [18]. We definitely need to use user- and item-based independent information like time and location, which can be considered as pure contextual in nature and treated by n-ary methods [19].

Acknowledgments. We would like to thank Alexander Tuzhilin, Elena Nenova, Radim Belohlavek, Vilem Vychodil, Sergei Kuznetsov, Sergei Obiedkov, Vladimir Bobrikov, Mikhail Roizner, and anonymous reviewers for their comments, remarks and explicit and implicit help during the paper preparations. This work was supported by the Basic Research Program at the National Research University Higher School of Economics in 2014-2015 and performed in the Laboratory of Intelligent Systems and Structural Analysis. First author was also supported by Russian Foundation for Basic Research (grant #13-07-00504).

References

1. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8) (2009) 30–37
2. Trefethen, L.N., Bau, D.: Numerical Linear Algebra. 3rd edition edn. SIAM (1997)
3. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Machine Learning **42**(1-2) (2001) 177–196
4. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Comput. **19**(10) (October 2007) 2756–2779
5. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. Journal of Computer and System Sciences **76**(1) (2010) 3 – 20 Special Issue on Intelligent Data Analysis.
6. Symeonidis, P., Nanopoulos, A., Papadopoulos, A., Manolopoulos, Y.: Nearest-biclusters collaborative filtering based on constant and coherent values. Information Retrieval **11**(1) (2008) 51–75
7. Ignatov, D.I., Kuznetsov, S.O., Poelmans, J.: Concept-Based Biclustering for Internet Advertisement. In: Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on. (Dec 2012) 123–130
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin/Heidelberg (1999)
9. du Boucher-Ryan, P., Bridge, D.: Collaborative recommending using formal concept analysis. Knowledge-Based Systems **19**(5) (2006) 309 – 315 {AI} 2005 {SI}.
10. Ignatov, D.I., Kuznetsov, S.O.: Concept-based recommendations for internet advertisement. In Belohlavek, R., Kuznetsov, S.O., eds.: Proc. of The Sixth International Conference Concept Lattices and Their Applications (CLA'08), Palacky University, Olomouc (2008) 157–166
11. Alqadah, F., Reddy, C., Hu, J., Alqadah, H.: Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. Knowledge and Information Systems (2014) 1–17
12. Ignatov, D.I., Nenova, E., Konstantinova, N., Konstantinov, A.V.: Boolean Matrix Factorisation for Collaborative Filtering: An FCA-Based Approach. In: Artificial Intelligence: Methodology, Systems, and Applications - 16th Int. Conf., AIMSA 2014, Varna, Bulgaria, September 11-13, 2014. Proceedings. (2014) 47–58
13. Kuznetsov, S.O.: A fast algorithm for computing all intersections of objects in a finite semilattice. Automatic Documentation and Math. Ling. **27**(5) (1993) 11–21
14. Birkhoff, G.: Lattice Theory. 11th edn. Harvard University, Cambridge, MA (2011)
15. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. **17**(6) (June 2005) 734–749
16. Ignatov, D.I., Poelmans, J., Dedene, G., Viaene, S.: A New Cross-Validation Technique to Evaluate Quality of Recommender Systems. In Kundu, M., Mitra, S., Mazumdar, D., Pal, S., eds.: Perception and Machine Intelligence. Volume 7143 of LNCS. Springer (2012) 195–202
17. Ignatov, D.I., Kornilov, D.: Raps: A recommender algorithm based on pattern structures. In: Proceeding of FCA4AI 2015 workshop at IJCAI 2015. (2015)
18. Trnecka, M., Trneckova, M.: An algorithm for the multi-relational boolean factor analysis based on essential elements. In: Proceedings of 11th International Conference on Concept Lattices and their Applications. (2014)
19. Ignatov, D., Gnatyshak, D., Kuznetsov, S., Mirkin, B.: Triadic formal concept analysis and triclustering: searching for optimal patterns. Machine Learning (2015) 1–32