Evaluating Collaborative Filtering: Methods within a Binary Purchase Setting

Stijn Geuens, Kristof Coussement, Koen W. De Bock

IESEG School of Management – Université Catholique de Lille (LEM, UMR CNRS 8179), Lille, France

{s.geuens,k.coussement,k.debock}@ieseg.fr

Abstract. The study of recommender systems based upon implicit binary purchase data constitutes an under-investigated area. The objective of this study is to evaluate configuration options of memory-based collaborative filtering (CF) for generating recommendations based upon online binary purchase data with different characteristics. First different algorithm configurations are identified. More specifically, three important algorithm parameters are investigated: the data reduction technique, the user- versus item-based CF and the similarity measure. Results on synthetic datasets show that these three factors influence the accuracy results of an algorithm. In a second analysis, extended experiments are set up to gain more insight into the influence of input data characteristics on the relative success of the CF configuration options. In particular, three input characteristics, sparsity level, item purchase distribution and item/user ratio are manipulated to analyze the impact on the algorithm's best configuration. Results show that the best performing algorithm is consistent, independently from the input data characteristics.

Keywords: Collaborative filtering, purchase data, synthetic data, evaluation metrics, data reduction, similarly measure

1 Introduction

In a typical E-commerce setting a customer receives an overload of information. In these situations, it is impossible for a customer to make optimal or even good product evaluations and purchase decisions. To cope with this overload of information, recommendation systems are designed. Besides the clear advantage for the customers, E-tailors also benefit from these systems, because they increase sales, revenue and loyalty [1].

A common used technique in recommendation systems is collaborative filtering (CF). CF systems propose a personalized set of items based on the customer's past behavior and activities of its peers. A good overview of the main existing algorithms is given in [2] and [3]. In a vast part of the literature, recommendations are based upon explicit customers' ratings [4]. These systems base product propositions upon ratings explicitly given by customers and its peers on a rating scale. Although these ratings clearly represent the customer's preference, it demands the user's effort, time and cost [5]. Additionally, results can be biased because customers have difficulties expressing their interest, leading to arbitrarily given or incorrect ratings. In most systems only a small fraction of the products purchased is rated, resulting in only a partial view of a customer when using explicit data [6].

To overcome these problems, implicit ratings can be used [6]. In contrast to explicit ratings, implicit ratings do not require user feedback, but derive affinity with items from actual user behavior. In an online retail setting characterized by a broad, deep and fast-changing product offering, explicit feedback is often hard to (sufficiently) collect. The collection of implicit feedback, on the other hand, is objective and non-intrusive. Implicit ratings come in all kinds of forms. For an overview we refer to Palanivel and Sivakumar [6]. In this study binary purchase data is used as a recommendation basis [7, 8]. In contrast to recommender systems based on explicit user feedback, systems using implicit purchase information remain under-investigated. Purchases are an important KPI for every commercial company, which makes it interesting to investigate a recommender based on purchase data.

The proposed study investigates recommendation systems based on implicit binary purchase data. In particular, different variations of the memory-based collaborative filtering algorithm are tested on binary purchase datasets having distinct input characteristics. Based on this setup, an experimental design is constructed to analyze the impact of input characteristics of a purchase dataset on the optimal algorithm configuration.

The remainder of this paper is organized as follows. The next section discusses some related work to better grasp the background of this study. A third section describes the setup of the experimental design. A fourth section presents the accuracy results based on 54 synthetic datasets. Finally, section five highlights the conclusions and next steps of the project.

2 Related Research

Recommendation systems use available customer information to create relevant personalized product suggestions. These recommendations can be based on different kinds of data. A typical classification of algorithms is presented by Bobadilla et al. [3]. First, *content-based* systems use past buying behavior of a customer to link these purchases to similar products based on characteristics. Second, *demographic-based* systems exploit socio-demographic data to predict relevant recommendations based on similarity in customers' characteristics. Third, *collaborative filtering* algorithms use past behavior of a customer, but, in contrast to content-based systems, they do not use product characteristics. Collaborative systems identify customers exhibiting the same behavior. Based on actions of these peers, products are suggested. Fourth, by combining different algorithms, *hybrid solutions* can be created, using different kinds of data to optimize the performance of the algorithms.

This study uses collaborative filtering algorithms based upon implicit binary purchase datasets pre-processed by a data reduction technique, as further elaborated in this related research.

2.1 Implicit Binary Purchase Data

The collaborative filtering literature typically refers to situations in which a customer expresses a preference by giving an explicit rating to a certain item [4]. In these cases recommendations are based upon the ratings explicitly given by a customer. As discussed in the introduction, using explicit data can have some flaws.

This study focuses though on implicit data and in particular on binary purchase data [7, 8]. This particular information, directly related to purchase behavior remains under-investigated in literature. As in the general case of recommender systems, algorithms possibly suffer from problems related to the input characteristics of the binary purchase matrix [9]. This study investigates sparsity, purchase distribution and item/user ratio.

Sparsity. A common recommender problem is the curse of dimensionality [9]. Typically an input matrix is very sparse, since a customer only buys a limited number of products and so products are only purchased by a limited number of customers. CF configurations tend to have difficulties with this scalability and sparsity, possibly influencing the model performance. A sparser input dataset leads in many cases to a decrease in accuracy and coverage of the proposed algorithm [9]. Typically datasets are very sparse, indicating this characteristic could indeed be an important factor to bear in mind.

Purchase Distribution. A second possible input characteristic problem is the distribution of purchases. Typically some items are very popular, but most products are only bought a few times. CF has the tendency to be less accurate towards the long tail and promoting the popular products [10].

Item/User Ratio. A third factor influencing the performance of CF is the item/user ratio. In settings with a low item/user ratio, it might be beneficial to use item-based algorithms since they are less computationally expensive. Moreover a user-based algorithm could be preferable in a setting with a high item/users ratio [11]. On top can the prediction accuracy of an algorithm depend on the ratio between items and users.

2.2 Data Reduction

Data reduction has as indirect advantage that sparsity is reduced [9]. One of the possibilities is using data reduction in the pre-processing phase of the algorithm's building process [5]. Although very popular in an explicit ratings context, reduction techniques are less used on implicit binary data [8, 12,13], while other fields of research frequently use binary reduction techniques [4, 14].

In this study reduction techniques are only used as pre-possessing step of the collaborative filtering procedure to gain efficiency and memory. Although direct imputation methods based on decomposed matrices are used in the past, this technique is not replicated in this study. The main reason for not using the direct method is the structure of the input matrix. Since direct imputations have as goal to estimate blanks in the original matrix and our input matrix only contains 0/1's, no purchase/purchase, and no missing values, direct imputation is less useful.

Four data reduction techniques are used in the pre-processing phase of the algorithm building process. Three popular reduction techniques in literature are singular value decomposition (SVD) [5], nonnegative matrix factorization (NMF) [4] and logistic principal component analysis (LPCA) [15]. These methods will be used in this study. Additionally a fourth reduction technique, correspondence analysis (CA), is applied. This method is conceptually similar to PCA, but can only be applied to binary data [16]. CA is never used in recommendation settings. Next to the reduced input matrices, the non-reduced purchase matrix is used as input.

2.3 Memory Based Collaborative Filtering

This paper focuses on memory-based collaborative filtering based on binary purchase data. Memory based CF is one of the two main distinctions in CF. In contrast to model-based CF [2], memory-based CF does not estimate a model to make recommendations. It only uses the user-item input matrix to calculate recommendations [17].

Until today memory-based CF remains one of the most popular algorithms in literature. The fact that it is only using the user-item matrix as input is a big advantage, since no extra data has to be gathered. The algorithm uses actual customer behavior as input for making the recommendations. Based on the user-item matrix, similarity is calculated and predictions are often based on the k-nearest neighbor algorithm [17].

CF Methods. Two possible distinctions exist in terms of the used CF method. An algorithm can be user- or item-based. The former type of system calculates similarity between customers. Products are proposed to users based on the behavior of their nearest neighbors [17]. In contrast, an item-based system calculates similarity between products and proposes similar items compared to the items purchased by a customer.

Similarity Measure. To identify nearest neighbors, similarity has to be calculated. In literature, many measures are considered [18]. In binary data settings cosine, Pearson correlation [13] and Jaccard's similarity [19] are often used. In contrast to cosine and Pearson correlation similarity, Jaccard's measure is based on set theory and can only be used on binary data.

$$\text{Cosine Similarity}(x,y) = \frac{\sum_{i\in I_{xy}} p_{x,i}\, p_{y,i}}{\sqrt{\sum_{i\in I_x} p_{x,i}^2}\sqrt{\sum_{i\in I_y} p_{y,i}^2}} = \frac{p_{x,i}\cdot p_{y,i}}{\sqrt{\sum_{i\in I_x} p_{x,i}}\sqrt{\sum_{i\in I_y} p_{y,i}}}. \tag{1}$$

$$\text{Pearson Correlation Similarity}(x,y) = \frac{\sum_{i\in I_{xy}}(p_{x,i} - \overline{p_x})(p_{y,i} - \overline{p_y})}{\sqrt{\sum_{i\in I_{xy}}(p_{x,i} - \overline{p_x})^2 \sum_{i\in I_{xy}}(p_{y,i} - \overline{p_y})^2}}. \tag{2}$$

$$\text{Jaccard Similairty}(x,y) = \frac{x \cap y}{x \cup y}. \tag{3}$$

Equations (1) and (2) represent respectively the cosine and Pearson correlation similarity measures. In these formulas $p_{x,i}$ and $p_{y,i}$ represent the purchase of item i by respectively customer x and y in a user-based setting. In an item-based setting, $p_{x,i}$ and $p_{y,i}$ represent respectively the purchase of product x and y by customer i. $\overline{p_x}$ and $\overline{p_y}$ represent respectively the mean purchase rate of customer x and customer y in a user-based setting. In an item-based setting, $\overline{p_x}$ and $\overline{p_y}$ refer to the mean purchase rate of product x and product y. I_{xy}, I_x and I_y represent the set of products bought by respectively customer x and y, customer x and customer y.

Equation (3) represents Jaccard's formula for calculating similarity between two binary purchase vectors. In this formula $x \cap y$ is the number of products bought by both customer x and y in a user-based setting. In an item-based setting, $x \cap y$ represents the number of customers purchased both product x and y. $x \cup y$ represents the number of products bought by at least one of both customers in a user-based setting. In an item-based setting, $x \cup y$ represents the number of customers who purchased at least one of the products x and/or y.

3 Setup of an Experimental Design

To analyze the link between input characteristics of the binary purchase input matrix and algorithm variations, a 5 x 2 x 3 between-subjects experimental design is constructed. The conditions of the experiment are the algorithm variations as discussed in related research. All algorithm variations are tested on 54 synthetic datasets with different input characteristics [20, 21], which are discussed in paragraph 3.1. Afterwards, results are calculated and algorithms are compared to obtain robust results identifying the best algorithm combination for given input characteristics.

3.1 Input Data Characteristics

Every binary implicit data matrix is unique and has its own characteristics. These characteristics can influence the optimal algorithm configuration, discussed in paragraph 3.2 and the final results of the model. In this proposed study, three important data characteristics are investigated on synthetic datasets: sparsity level, purchase distribution over products and item/user ratio.

Synthetic Data Characteristics. In order to mimic real-life situations, 54 synthetic datasets, characterized by six levels of sparsity, three different purchase distributions and three distinct item/user ratios, are created.

Sparsity levels. Sparsity levels are artificially created to mimic possible real-life situations. Since a typical recommendation setting consists of very sparse input matrices, six sparsity levels are variable between 95% and 99.50% [20, 22, 23].

Purchase Distribution. Purchases typically show a long tailed distribution over products. Some popular items are bought frequently, but most items are purchased only a few times [10]. In the study, the input is varied from a logarithmic distribution, having a very long tail over a linear distribution with a moderate tail to a uniform distribution.

Item/User Ratio. A last manipulated input characteristic is the item/user ratio. By simply varying the number of rows of the binary input matrix, the ratio adjusts [20]. The synthetic datasets consist of 1 000 items, but the number of customers is adjusted. The number of users is set to 500, 1 000 and 2 000, resulting in item/user ratios of respectively 2, 1 and 0.5.

Synthetic Data Generation. Although the aim of the datasets is to be generic and generalizable, a certain structure needs to be inherently present in the data to be able to discover patterns. To create this structure, the correlation matrix of a binary purchase dataset of a European E-tailor is mimicked in the synthetic datasets.

$$r_{ij} = \frac{p_{ij} - p_i p_j}{\sqrt{p_i q_i p_j q_j}} \Leftrightarrow p_{ij} = r_{ij} \sqrt{p_i q_i p_j q_j} + p_i p_j, \text{ where } i, j \in \{1, \dots, I\}\,. \tag{4}$$

Equation 4 shows the relationship between the correlation, from the test dataset, the marginal probabilities and pairwise joint probabilities between two item vectors i and j [24]. r_{ij} indicates the correlation between vector i and j, p_i and p_j represent the marginal probabilities of item i and j, q_i and q_j denote $(1 - p_i)$ and $(1 - p_j)$ and p_{ij} represents the pairwise joint probabilities between vector i and j. Based on equation 4 purchase probabilities of each synthetic product (item vector) can be generated for a certain number synthetic customers. For each synthetic product i a purchase probability vector $P_i = [p_{i1} \dots p_{iC}]$, where C stand for the number of customers, can be generated.

Next to taking into account the correlation structure, the described step gives the ability to control the item/user ratio and sparsity level. By setting the number of customers (C), the item/user ratio is regulated. By adjusting the marginal purchase probabilities by a constant, sparsity is set. The same constant should be applied to every item to not to disrupt the purchase probability structure.

Purchase probabilities are continuous elements in [0, 1], whereas a binary input matrix should be created. Transforming the purchase probability vector P_i to a binary purchase vector $B_i = [b_{i1} \dots b_{iC}]$ can be done by applying a threshold. Every element b_{ic} is defined by equation 5.

$$b_{ic} = \begin{cases} 1, p_{ic} \geq f(i) \\ 0, p_{ic} < f(i) \end{cases}, \text{where } c \in \{1, \dots, C\}. \quad (5)$$

Implementing equations 4 and 5 gives the possibility to create binary item purchase vectors accounting for item/user ratio and sparsity. To be able to vary the purchase distribution, functional thresholds f(i), as described in equations 6, 7 and 8, are applied. Each of these equations results in different structures of the binary input matrix.

Firstly, a logistic function is applied as threshold to create a logarithmic distribution of purchase frequencies of each item. The resulting distribution is characterized by a limited number of products frequently purchased and a lot of products only purchased a few time.

$$f(i) = \beta_1 \log_{(C+\gamma_1)}(i+1), \text{where } i \in \{1, \dots, I\}. \quad (6)$$

Secondly, a linear function is imposed to give the purchase distribution a linear structure.

$$f(i) = \beta_2 \frac{i}{(C+\gamma_2)}, \text{where } i \in \{1, \dots, I\}. \quad (7)$$

Finally, a constant is set as threshold to create a uniform purchase distribution.

$$f(i) = \beta_3 \quad (8)$$

3.2 Experimental Setup

To get a better notion which algorithms perform best in combination with certain input characteristics, different algorithm configurations are calculated on input datasets discussed above. The algorithms are based on memory-based CF, but differ in dimension reduction method, CF method (item- vs. user-based) and similarity measure, as discussed in the related research section.

The used data reduction methods are SVD, NMF, LPCA and CA. Together with the none reduced purchase matrix, five different input matrices for the collaborative filtering algorithm are created. Item- and user based CF are used as CF-method and cosine, Pearson correlation and Jaccard's similarity constitute the three measures for similarity calculation.

Combining the discussed algorithm configuration elements gives the opportunity to create a 5 x 2 x 3 between-subjects experimental design. Different experimental conditions are constructed by combining each time one of the five proposed reduction techniques with user- or item-based CF and one of the three discussed similarity measures. To make the results generalizable and valid, tests are run on the 54 synthetic datasets**.** In total 1620 individual runs with different input characteristic – algorithm configuration combinations are executed.

3.3 Evaluation Metrics

Results of different CF configurations on different binary input matrices can be assessed using accuracy measures. This allows comparing the performance of the used algorithm variations on input matrices with a variety of specific characteristics.

To evaluate the algorithms, Top-N recommendations consisting of 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150 and 200 items are considered. The accuracy of the predictions is expressed in terms of F1-measure [11], calculated on a random test sample consisting of 20% of the input dataset.

4 Results

Firstly, this study investigates the best algorithm configuration (data reduction technique, CF-method and similarity measure) in general, without taking dataset characteristics into account. This will result in an evaluation of the best overall configuration. Secondly, the best algorithm configuration for each dataset separately is investigated. By comparing different champion models and good alternatives, for datasets with distinct input characteristics, conclusions can be draw with respect to the best input characteristic – algorithm variation combination.

Method. To evaluate the effect of the different algorithm configurations on the different synthetic datasets an ANCOVA analysis is run.

$$\mathrm{F1}_i = \beta_{0i} + \beta_{1i}\mathrm{RM} + \beta_{2i}\mathrm{CFM} + \beta_{3i}\mathrm{SM} + \beta_{4i}\mathrm{RM} * \mathrm{CFM} + \beta_{5i}\mathrm{RM} * \mathrm{SM} + \beta_{6i}\mathrm{CFM} * \mathrm{SM} + \beta_{7i}\mathrm{RM} * \mathrm{CFM} * \mathrm{SM} + \beta_{8i}\mathrm{SS} + \varepsilon_i \ . \tag{9}$$

Equation 9 presents the ANCOVA model that allows analyzing the main and interaction effects of reduction technique (RM), CF-method (CFM) and similarity measure (SM). Selection size (SS) is included as a covariate to control for different top-N selections. The inclusion of the covariate makes it possible to evaluate the F1-measures for every selection size in one analysis instead of repeating single ANOVA's for each top-N.

Best overall configuration. Analyzing the main effects of reduction technique, CF-method and similarity measure results in an evaluation of the best overall setting of each of the three algorithm variation parameters. Results show reduction method ($F_{4, 252}$ = 121.95, p = < .0001), CF-method ($F_{1, 252}$ = 154.59, p = < .0001) and similarity measure ($F_{2, 252}$ =368.09, p = < .0001) have a significant impact on the F1 performance of the model.

CA is the significantly best performing data reduction method, followed by NMF and SVD, which do not significantly differ from each another. LPCA performs significantly worse than mentioned reduction techniques, but significantly outperforms models based on the none reduced matrix. In terms of CF-method, item-based CF significantly outperforms the user-based method. Similarity measures also show a significant difference. Correlation and cosine similarity are performing similar, while both measures significantly outperform Jaccard's similarity.

Best configuration within datasets. The ANCOVA analysis is run separately for each of the 54 synthetic datasets with 299 different model configurations (observations). The analysis results in a champion model, based on F1 performance, for each synthetic dataset with different input characteristics. Tables 1, 2 and 3 visualize the results. Beneath the tables, models not significantly differing from the champions are listed.

CF-Method. In the tables, only item-based algorithms are listed as champions. This indicates that, regardless the characteristics, item-based CF outperforms user-based CF for each dataset.

Similarity Measure. The absence of models using Jaccard's measure in the tables indicates that these algorithms perform significantly worse than models with cosine and Pearson correlation similarity measures. For each dataset having a model with cosine (correlation) as champion model, the correlation (cosine) alternative does not significantly differ. This observation indicates that cosine and correlation similarity are statistically interchangeable. For 24 datasets, correlation constitutes the champion model. In 30 cases, cosine delivers the best results.

Table 1. Champion models for datasets with an **item/user ratio of 2** (= 500 users) in function of sparsity and distribution

Sparsity	Distribution: Logistic	Linear	Uniform
0.95	CA / Item / Corr [1]	CA / Item / Cos [1/3/4/5]	CA / Item / Cos [2/3/4/5/6]
0.96	CA / Item / Corr [1]	CA / Item / Cos [1/3/4/5]	CA / Item / Cos [2]
0.97	CA / Item / Corr [1]	CA / Item / Corr [1]	CA / Item / Corr [1/3]
0.98	CA / Item / Corr [1]	CA / Item / Corr [1/3]	CA / Item / Corr [1]
0.99	CA / Item / Corr [1/3]	CA / Item / Cos [2/3]	CA / Item / Corr [1/3]
0.995	CA / Item / Corr [1/3]	CA / Item / Corr [1/3/4]	CA / Item / Cos [2/3]

[1] CA / Item / Cos; [2] CA / Item / Corr; [3] NMF / Item / Cos, Corr; [4] None / Item / Cos, Corr; [5] SVD / Item / Cos, Corr; [6] LPCA / Item / Cos, Corr

Table 2. Champion models for datasets with an **item/user ratio of 1** (=1 000 users) in function of sparsity and distribution

Sparsity	Distribution: Logistic	Linear	Uniform
0.95	CA / Item / Corr [1/5]	CA / Item / Cos [2/4/5]	CA / Item / Cos [2/3/4/5/6]
0.96	CA / Item / Cos [2]	CA / Item / Cos [2/4/5]	CA / Item / Cos [2]
0.97	CA / Item / Cos [2]	CA / Item / Corr [1/5]	CA / Item / Corr [2]
0.98	CA / Item / Cos [2]	CA / Item / Cos [2]	CA / Item / Cos [2]
0.99	CA / Item / Corr [1]	CA / Item / Cos [2]	CA / Item / Corr [2/3]
0.995	CA / Item / Corr [1]	CA / Item / Cos [2/3]	CA / Item / Cos [2/3/4]

[1] CA / Item / Cos; [2] CA / Item / Corr; [3] NMF / Item / Cos, Corr; [4] None / Item / Cos, Corr; [5] SVD / Item / Cos, Corr; [6] LPCA / Item / Cos, Corr

Table 3. Champion models for datasets with an **item/user ratio of 0.5** (=2 000 users) in function of sparsity and distribution

Sparsity	Distribution: Logistic	Linear	Uniform
0.95	SVD / Item / Cos [1/2/5]	SVD / Item / Corr [1/2/4/5]	CA / Item / Cos [2/3/4/5/6]
0.96	CA / Item / Cos [2/5]	SVD / Item / Cos [1/2/4/5]	CA / Item / Cos [2/3/4/5/6]
0.97	CA / Item / Corr [1/5]	CA / Item / Cos [2/5]	CA / Item / Cos [2/5]
0.98	CA / Item / Cos [2]	CA / Item / Cos [2/5]	CA / Item / Corr [1/3]
0.99	CA / Item / Cos [2]	CA / Item / Cos [2]	CA / Item / Corr [1]
0.995	CA / Item / Corr [1]	CA / Item / Cos [2]	CA / Item / Corr [1/3]

[1] CA / Item / Cos; [2] CA / Item / Corr; [3] NMF / Item / Cos, Corr; [4] None / Item / Cos, Corr; [5] SVD / Item / Cos, Corr; [6] LPCA / Item / Cos, Corr

Data Reduction Method. In 51 out of 54 datasets the champion model is based on a CA reduced matrix, indicating this is the overall best reduction technique. For three datasets an algorithm based on the SVD decomposed matrix gives the best results. For these models CA configurations are not performing significantly worse, indicating that CA configurations are good alternatives for the SVD based models in these three cases. This logic also goes the other way round. Although CA reduction is the clear champion, in most cases other configurations are performing statistically similar, meaning these models can serve as good alternatives.

Item-based models using cosine or correlation similarity based on the *NMF reduced matrix* serve as good alternatives for CA configurations in 19 cases. Especially in datasets with a high item/user ratio (2), NMF seems a good alternative. For 11 out of 18 datasets having an item/user ratio of 2, NMF is not performing significantly worse than CA.

Creating a model based on the *SVD reduced matrix* is a statistically good input basis for 18 out of 54 datasets. Lower sparsity, gives a higher chance that SVD is a good alternative for CA, especially in cases with a lower item/user ratio. The good performance of SVD in these cases is emphasized by being the champion in datasets with sparsity 0.95 (and 0.96 only for linear distribution) and an item/user ratio of 0.5.

Using the the full *none reduced matrix* as input is a good alternative in 12 cases. All cases represent linear or uniform distributions with mainly low sparsity levels (0.95 – 0.96). Watch out that for datasets with these characteristics most configurations based on item-based CF and cosine or correlation are statistically good alternatives.

LPCA reduced input matrices serve only a good alternative in 4 cases. For these 4 datasets, being cases with low sparsity and a linear or uniform distribution, all models using item-based CF and cosine or correlation similarity are performing significantly similar. This indicates that LPCA can be seen as the data reduction technique resulting in the least good accuracy results.

5 Discussion and Future Work

The presented prediction accuracy results, measured by the F1-statistic, indicate that item-based CF outperforms user-based CF and cosine and Pearson correlation give the same results but beat Jaccard based models. As data reduction technique CA gives the best results, but in some specific input characteristic cases NMF, SVD or the none reduced matrix can serve a good alternative.

Results presented in this study only take the F1-measure into account. To broaden the scope of the analysis, extra evaluation metrics will be analyzed in the next steps of the project. For prediction accuracy recall, precision and ROC-curve will be considered next to the F1-measure. Ranking accuracy will be measured using Kendalls Tau-C. Next to accuracy, item coverage, computation time and diversity (Intra-List Similarity) will be analyzed to get a complete outlook of each input characteristic – algorithm configuration combination.

To validate the results of this study, the discussed experimental design will be replicated on 10 real-life datasets of a leading European E-tailor.

References

1. Dias, M. B., Locher, D., Li, M., El-Deredy, W., Lisboa, P. J. G.: The Value of Personalised Recommender Systems to E-Business: A Case Study. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 291-294, ACM, Lausanne, Switzerland
2. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of yhe State-of-the-Art and Possible Extensions. IEEE Trans. Knowl. Data Eng. 17, 6, 734-749 (2005)
3. Bobadilla, J., Ortega, F., Hernando, A., Gutierrez, A.: Recommender Systems Survey. Knowledge-Based Syst. 46, 109-132 (2013)
4. Su, X., Khoshgoftaar, T. M.: A Survey of Collaborative Filtering Techniques. Adv. in Artif. Intell. 2-21 (2009)
5. Kellar, M., Watters, C., Duffy, J.: Effect of Task on Time Spent Reading as an Implicit Measure of Interest. In: Bryans, J. B. (eds.): Asist 2004: Proceedings of the 67th Asis&T Annual Meeting: Managing and Enhancing Information: Cultures and Conflicts, Vol 41, pp. 168-175. Information Today Inc, Medford (2004)

6. Palanivel, K., Sivakumar, R.: A Study on Implicit Feedback in Multicriteria E-Commerce Recommender System. J. Electron. Commer. Res. 11, 2, 140-156 (2010)
7. Bodapati, A. V.: Recommendation Systems with Purchase Data. J. Mark. Res. 45, 77-93 (2008)
8. Pradel, B., Sean, S., Delporte, J., Guérif, S., Rouveirol, C., Usunier, N., Fogelman-Souli, F., Dufau-Joel, F.: A Case Study in a Recommender System Based on Purchase Data. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 377-385, ACM, San Diego, California, USA (2011)
9. Sarwar, B., Karypis, G., Kostan, J., Riedl, J. T.: Application of Dimensionality Reduction in Recommender Systems - A Case Study. In: WebKDD Workshop (2000)
10. Steck, H.: Item Popularity and Recommendation Accuracy. In: Proceedings of the Fifth ACM Conference on Recommender Systems, pp. 125-132, ACM, Chicago, Illinois, USA (2011)
11. Herlocker, J. L., Konstan, J. A., Terveen, K., Riedl, J. T.: Evaluating Collaborative Filtering Recommender Systems. ACM Trans. Inf. Syst. 22, 1, 5-53 (2004)
12. Khanna, R., Zhang, L., Agarwal, D., Chen, B. C.: Parallel Matrix Factorization for Binary Response. IEEE, New York (2013)
13. Breese, J. S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the Fourteenth conference on Uncertainty in Artificial Intelligence, pp. 43-52, Madison, Wisconsin, USA (1998)
14. Belohlavek, R., Vychodil, V.: Discovery of Optimal Factors in Binary Data via a Novel Method of Matrix Decomposition. J. of Comp. and Sys. Sc. 76, 1, 3-20 (2010)
15. Lee, S., Huang, J. Z., Hu, J. H.: Sparse Logistic Principal Components Analysis For Binary Data. Ann. Appl. Stat. 4, 3, 1579-1601 (2010)
16. Greenacre, M.: Correspondence Analysis in Practice. In Series: Keiding, N., Morgan, B., Speed, T., van der Heijden, P. (eds.): Interdisciplinary Statistics Series, Edition 2, Chapman & Hall/CRC, Boca Raton (2007)
17. Papagelis, M., Plexousakis, D.: Qualitative Analysis of User-Based and Item-Based Prediction Algorithms for Recommendation Agents. Eng. Appl. Artif. Intell. 18, 7, 781-789 (2005)
18. Choi, S.-S., Cha, S.-H., Tappert, C.: A Survey of Binary Similaruity and Distance Measures. J. on Systemics, Cybermetics and Informatics 8, 1, 5 (2010)
19. Candillier, L., Meyer, F., Fessant, F.: Designing Specific Weighted Similarity Measures to Improve Collaborative Filtering Systems. In: Proceedings of the 8th Industrial Conference on Advances in Data Mining: Medical Applications, E-Commerce, Marketing, and Theoretical Aspects, pp. 242-255, Leipzig, Germany (2008)
20. Aggarwal, C. C., Wolf, J. L., Wu, K.-L., Yu, P. S.: Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 201-212, ACM, San Diego, California, USA (1999)
21. Tso, K. H. L., Schmidt-Thieme, L.: Empirical Analysis of Attribute-Aware Recommendation Algorithms with Variable Synthetic Data. Springer-Verlag, Berlin (2006)
22. Deshpande, M., Karypis, G.: Item-Based Top-N Recommendation Algorithms. ACM Trans. Inf. Syst. 22, 1, 143-177 (2004)
23. Popescul, A., Ungar, L. H., Pennock, D. M., Lawrence, S.: Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, pp. 437-444, Seatle, Washington, USA (2001)
24. Leisch, F., Weingessel, A., Hornik, K.: On the Generation of Correlated Artificial Binary Data. Working Paper Series: Adaptive Information Systems and Modelling in Economics and Management Science, Vienna (1998)