FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 38 (2013) No. 1

DOI: 10.2478/v10209-011-0021-x

COLLABORATIVE FILTERING BASED ON BI-RELATIONAL DATA REPRESENTATION

Andrzej SZWABE *, Pawel MISIOREK *, Michal CIESIELCZYK *, Czeslaw JEDRZEJEK *

Abstract. Widely-referenced approaches to collaborative filtering (CF) are based on the use of an input matrix that represents each user profile as a vector in a space of items and each item as a vector in a space of users. When the behavioral input data have the form of *(userX, likes, itemY)* and *(userX, dislikes, itemY)* triples one has to propose a representation of the user feedback data that is more suitable for the use of propositional data than the ordinary user-item ratings matrix. We propose to use an element-fact matrix, in which columns represent RDF-like behavioral data triples and rows represent users, items, and relations. By following such a triple-based approach to the bi-relational behavioral data representation we are able to improve the quality of collaborative filtering. One of the key findings of the research presented in this paper is that the proposed bi-relational behavioral data representation, while combined with reflective matrix processing, significantly outperforms state-of-the-art collaborative filtering methods based on the use of a 'standard' user-item matrix.

Keywords: collaborative filtering, personalized recommendation, Statistical Relational Learning, RDF, propositional data representation

1 Introduction

In many real-world cases, a recommender system is expected to predict user choices, rather than provide exact ratings. Therefore, it should not be surprising that many researchers working on personalized recommendation systems study the so-called 'find good items' task [9],[24].

The natural consequence of investigating the 'find good items' task is the widespread use of 'selection-oriented' recommendation quality measures (originally developed by the IR community), such as precision, recall, and F_1 [9],[11]. It is worth noting that these measures reflect a special case of the 'find good items' task, called the 'find all good items' task.

*Institute of Control and Information Engineering, Poznan University of Technology, M. Sklodowskiej-Curie Square 5, 60-965 Poznan, Poland

The Area Under the Receiver Operating Characteristic curve (AUROC) is regarded as the best recommendation quality measure, as long as one assumes that the purpose of an evaluated recommender is to 'sort' all items according to their estimated attractiveness [9],[24]. AUROC, as a measure that allows one to abstract from any particular precision-recall proportion, is frequently used in off-line experiments aimed at simulating the recommendation quality [9].

In contrast to behavioral datasets, in which negative user feedback data are usually much more incomplete than positive feedback data [19], a rating-based dataset allows the researcher to use AUROC in a more reliable way [24]. A relatively reliable AUROC measurement may be realized based on the conversion of ratings into propositions (each one of the form *(userX, likes, itemY)* or *(userX, dislikes, itemY)*). The most straightforward solution is to treat the average rating for each user as the threshold to discriminate between attractive and non-attractive items.

Some researchers find it reasonable to address the application scenario, in which a recommender system is provided with data about user choices, rather than with ratings [13],[24]. When the purpose of the recommender system is to predict user choices rather than real-valued ratings, the entries of the input data matrix should have a form of binary numbers representing propositions of the form of *(userX, likes, itemY)*. Such a format is frequently regarded as the most convenient to model user actions [13],[10], especially in on-line retailing and news/items recommendation scenarios. In such scenarios, the binary information about user actions (e.g., about purchase or page view) is the only data available to the system. An example is One-Class Collaborative Filtering system [13],[17], for which only the data on positive user preferences are given and negative examples are absent. It is also worth noting that YouTube has simplified its multi-level rating schema to a binary one.

The observations stated above motivate the treatment of behavioral data for CF tasks as a binary input (i.e., as a set of propositions). In order to address such a scenario of using binary input data, we propose to use a new data representation and processing framework that consists of the following two elements:

- a data representation method based on a binary element-fact matrix, for which rows represent elements (i.e., users, items, and relations playing the roles of RDF subjects, objects, and predicates, respectively) and columns represent facts (i.e., RDF-like triples),
- the vector similarity quasi-measure based on the 1-norm length of the Hadamard product of the given tuple of vectors.

The Hadamard product refers herein to the entrywise product of matrices of the same size (in our case the entrywise product of the given tuple of vectors). This quasi-measure may be regarded as a generalization of the dot product introduced in order to extend the domain beyond the case of only two vectors.

2 Related Work

Behavioral data processed by some of the recommender systems presented in the relevant literature have the form of propositional statements about user preferences and the user-system

interaction history. As such data have a natural relational representation, researchers working in the area of Machine Learning investigate collaborative filtering as one of the applications of Statistical Relational Learning (SRL) [21][18].

The input data for a recommender system sometimes have the form of relational data. In order to enable the representation and processing of RDF triples of more than one relation, Singh *et al.* [18] proposed an approach based on the collective matrix processing. This approach was followed by proposals of similar models based on 3rd-order tensors [12],[21],[8],[23]. In this paper, we propose to use an incidence matrix data representation (more precisely, an incidence matrix of a weighted 3-uniform hypergraph) for which columns represent behavioral data triples (hypergraph edges) and rows represent users, items and relations (hypergraph vertices). To our knowledge, such an approach is innovative, at least, in the area of research on collaborative filtering.

SRL addresses one of the central questions of Artificial Intelligence (AI) – the integration of probabilistic reasoning and first-order relational representations [20]. In such a context, the approach to collaborative filtering proposed in this paper may be seen as forming a basis for bi-relational SRL, in the case of which a structurally unconstrained vector-space data representation (making the data alternatively interpretable as a heavily-connected weighted hypergraph or a tensor) is used, rather than a graphical model [20].

The propositional nature of the algebraically represented data makes our proposal relevant to the challenge of unifying reasoning and search, which is sometimes referred to as the challenge of Web scale reasoning [6]. As far as algebraic representation of graph data is concerned, our method may be regarded as very similar to the one described in [5] – both the methods are based on the assumption that a graph node may be represented as a virtual document expressed as a sparse vector in a space of dimensions that correspond to the triples' constituents (i.e., that correspond to subjects, predicates and objects). On the other hand, our method features a new kind of vector-space quasi-similarity measurement that allows us to effectively estimate the likelihood of unknown RDF triples, rather than limiting the system functionality to the RDF graph nodes search, based on a traditional (i.e., bilateral) similarity measure [5].

Although the results presented in the paper are focused on methods based on vector representations, some of the compared techniques are based on the reflective data processing, thus sharing the features of graph-based solutions. In the case of Reflective Random Indexing method (RRI) [3], the vector representations are updated in steps referred to as reflections. Performing multiple reflections may be regarded as a process of exploring the graph neighborhood, in which the first reflection corresponds to the exploration of direct vertex neighbors whereas the further reflections extend the exploration to distant neighbors of the vertex. From such a perspective, the results presented in the paper are highly relevant to research on the graph-theoretical approaches to collaborative filtering including the work on graph-theoretical modeling of correspondences between user preferences [1], the kernels-on-graphs methods [7], and the already-mentioned methods based on Spreading Activation [5],[6]. Moreover, it is worth pointing out that collaborative filtering is also related to the problem of social network graph analysis [15]. Both the problems deal with the use of behavioral data, although in the case of collaborative filtering obtaining such data is automatic, i.e., does not require explicit users' interventions.

3 Methodology

The issues mentioned above have motivated the proposal of a new input data representation and a data processing model for collaborating filtering. Both the data representation and processing model are fact-based rather than rating-based. Apart from a few simplified recommendation scenarios, the proposed model has been investigated in a bi-relational scenario involving the use of two predicates: *Likes* and *Dislikes*. Such an investigation involved, among other steps, a special dataset preparation and a recommendation list generation. The evaluation of the proposed model has been realized with the use of state-of-the-art collaborative filtering data processing algorithms and the AUROC measure.

3.1 Proposed CF Data Representation Model

We propose to model the input data of a recommendation system (i.e., the information on all modeled elements: users, items, and predicates) as a set of RDF triples representing propositions (facts) stored in the form of an element-by-fact matrix referred to as A. It is worth noting that a fact-based data model can be used to represent graded user preference data since each level of a discrete-value rating scale may be modeled as a separate predicate. Furthermore, a conversion of the input data into the fact-based representation (i.e., to RDF triples) simplifies the integration of the data on user preferences, gathered from different sources. Taking into account the widespread adoption of the RDF technology, we believe that the proposed fact-based data representation model allows for flexible representation of highly heterogeneous data, leading to new successful applications of semantically-enhanced recommender systems [22].

3.2 Application of the Proposed Model in the Bi-Relational CF Scenario

In this paper, we focus on a bi-relational recommendation scenario. The data representation model, which is commonly used in such a scenario, is based on a classical user-item matrix. In order to store the information about the two predicates in a single matrix, the rating scale is used in such a way that a high value indicates that a user likes an item, whereas a low value indicates the presence of the *Dislikes* relation. In such a case, it is assumed that the relations representing the 'positive' and the 'negative' user feedback are linearly dependent. In this paper, we propose a solution which enables a more flexible (not linear) modeling of the relationship between triples of two predicates. In our model, both the relations are modeled as separate predicates represented by separately trained vectors. Such an approach allows us to exploit the similarities between users and items that are observable within the set of triples including the *Likes* predicate or within the set of triples including the *Dislikes* predicate. As we show in the paper, such an ability may lead to an increase in the recommendation performance, which is not possible when the classical user-item matrix is used.

Although the proposed data representation model assumes representing both *Likes* and *Dislikes* relations exactly in the same way, one may realize that as far as the recommendation generation task is concerned, the propositional data of the *Dislikes* relation is used only indi-

rectly, i.e., in a way resembling the use of a semantic enhancement of CF [22]. We believe that such an approach to multi-relational data integration is justified by the basic assumption about the purpose of any recommender system, which is the naturally 'biased' estimation of items that are 'liked', not 'disliked' by a given user. Moreover, by showing a successful (i.e., recommendation quality improving) use of *Dislikes* triples as a secondary source of relational data (supporting the use of the primary *Likes* relational data), we feel encouraged to propose our model as the basis for future developments of recommender systems taking the advantage of using multi-relational data.

3.3 Data Representation Based on an Element-Fact Matrix

We introduce a data representation model based on the binary matrix A representing subject-predicate-object RDF-like triples. The rows of the matrix represent all the entities used to define triples, i.e. the elements playing the role of a subject, object or predicate, whereas columns represent propositions corresponding to the triples. We define a set $E = S \cup P \cup O$ as a set of elements referred to by the triples, where S is a set of subjects, P is a set of predicates, and O is a set of objects. We assume that $|S| = n$, $|O| = m$, and $|R| = l$. The model allows each modeled entity to freely play the role of a subject or an object, so in the general scenario we have $|S| = |O|$. In the case of the application scenario presented in this paper, sets S and O (i.e., sets of subjects and objects) are disjoint and correspond to sets of users and items, respectively. Additionally, we define set F as a set of all facts represented as the input dataset triples, such that $|F| = f$.

Finally, we define the binary element-fact matrix $A = [a_{i,j}]_{(n+m+l)\times f}$ as follows:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,f} \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,f} \\ a_{n+1,1} & a_{n+1,2} & \dots & a_{n+1,f} \\ \vdots & \vdots & & \vdots \\ a_{n+m,1} & a_{n+m,2} & \dots & a_{n+m,f} \\ a_{n+m+1,1} & a_{n+m+1,2} & \dots & a_{n+m+1,f} \\ \vdots & \vdots & & \vdots \\ a_{n+m+l,1} & a_{n+m+l,2} & \dots & a_{n+m+l,f} \end{bmatrix}. \tag{1}$$

As a consequence of the fact that the columns of matrix A represent the triples, each column contains precisely three non-zero entries, i.e., for each j there are exactly three non-zero entries $a_{k_1,j}, a_{k_2,j}, a_{k_3,j}$, such that $1 \leq k_1 \leq n$, $n+1 \leq k_2 \leq n+m$, and $n+m+1 \leq k_2 \leq n+m+l$ – the entries corresponding to that correspond to the subject, object, and predicate of the modeled triple. At the same time, the number of non-zero entries in each row denotes the number of triples containing the element that corresponds to this row. Such a model is convenient to represent an RDF dataset, which consists of a finite number of predicates $l \geq 1$.

3.4 Generation of Prediction Lists

When using the element-fact matrix as the data representation, one has to perform the prediction generation step in a special way. Initially, as in many of the most accurate collaborative filtering methods, the missing entries of the input matrix are estimated. In order to achieve this, the input matrix A is processed into its reconstructed form $\widehat{A}$ using one of the evaluated recommendation algorithms, presented in section 3.7. Afterwards, we propose to calculate each of the predictions as the 1-norm length of the Hadamard product of row vectors (in explicit, the vector for which each coordinate is calculated as the product of corresponding coordinates of the original tuple of vectors of the same dimensions, i.e., the entrywise product of vectors) corresponding to the elements of the given RDF triples. Each RDF triple forms the proposition which is the subject of likelihood estimation. More formally, the prediction value $p_{i,j,k}$ is calculated according to the formula:

$$p_{i,j,k} = \left\| \widehat{a}_i \circ \widehat{a}_j \circ \widehat{a}_k \right\|_1,$$

where $\widehat{a}_i$, $\widehat{a}_j$ and $\widehat{a}_k$ are the row vectors of the reconstructed matrix $\widehat{A} = [\widehat{a}_{i,j}]_{(n+m+l)\times f}$ corresponding to the subject, predicate, and object of the given RDF triple, and the symbol $\widehat{a}_i \circ \widehat{a}_j \circ \widehat{a}_k$ denotes the Hadamard product of vectors $\widehat{a}_i$, $\widehat{a}_j$ and $\widehat{a}_k$.

The proposed formula may be seen as a generalization of the dot product formula, as in the hypothetical case of measuring quasi-similarity of two (rather than three) vectors, the formula is equivalent to the dot product of the two vectors. The interpretation of the proposed formula as the likelihood of the joint incidence of two or more events represented as vectors is based on the quantum IR model (see [26] and [14]).

The underlying model of computing the entities group similarity follows the quantum probability approach presented in [14], [26]. The central role in the model is played by the Hilbert space which is used to represent the structure of coincidences between the real-world entities modeled in the system. The probability definition is based on Gleason's theorem [14], [26], which explains the correspondence between probability measure on the lattice of closed subspaces of the Hilbert space and the density operator on this space [14]. The procedure of the probability calculation may be described in terms of quantum observables (which are represented by Hermitian operators) corresponding to the real-world entities modeled in the system. As a result of the modeling assumptions (in particular the operators' compatibility assumption), each vector representation may be used to build a diagonal operator being a quantum observable of the probability of given entity occurrence in the dataset. Observables of probability of entities coincidences (i.e., observables of propositions) are obtained as products of its constituents observables. The probability calculation is done as a result of quantum measurement procedure, i.e., as an expectation value of the respective observables [14].

It has to be admitted that for the methods presented in this paper, the coordinates of modeled entities' representations do not formally denote probabilities. Therefore, formally speaking the proposed method may be regarded as a technique for providing the likelihood of the joint incidence of two or more events represented as vectors, which is inspired by the quantum IR model of probability calculation.

3.5 Evaluated Scenarios

In this paper, we evaluate two matrix-based methods for the representation of RDF datasets that are applicable in the collaboration filtering scenario: the classical user-item matrix model and the novel element-fact matrix model. Both the models have been tested in two experimental scenarios: the one-class collaborative filtering scenario (with the use of RDF triples of the *Likes* predicate only) and the bi-relational collaborative filtering scenario (with the use of RDF triples of both the *Likes* and *Dislikes* predicates). In particular, the following four scenarios S1-4 have been investigated:

- S1 – the application of a binary user-item matrix $B = [b_{i,j}]_{n \times m}$ representing RDF triples of the *Likes* predicate, where n is the number of users, and m is the number of items,
- S2 - the application of a ternary $\{-1, 0, 1\}$ user-item matrix $B = [b_{i,j}]_{n \times m}$ representing RDF triples of the *Likes* predicate (denoted by positive numbers) and RDF triples of the *Dislikes* predicate (denoted by negative numbers),
- S3 - the application of a binary element-fact matrix $A_{i,j} = [a_{i,j}]_{(n+m+l) \times f}$ representing RDF triples of the *Likes* predicate, where n is the number of subjects, m is the number of objects, and $l = 1$ is the number of predicates (only the *Likes* predicate is represented),
- S4 - the application of a binary element-fact matrix $A_{i,j} = [a_{i,j}]_{(n+m+l) \times f}$ representing RDF triples of the *Likes* predicate or the *Dislikes* predicate, where the number of predicates is equal to 2 ($l = 2$).

3.6 Dataset Preparation

In order to evaluate the collaborative filtering methods, we have used one of the most widely referenced data sets – the MovieLens ML100k set [4], which contains 100 000 ratings of 1682 movies given by 943 unique users. Each rating that is higher than the average of all ratings has been treated as an indication that a given user likes a given movie, i.e., as a fact (proposition) denoted by an RDF triple of the form (userA, *likes*, movieA). Analogically, each rating lower than the average, has been used as an indication that a given user dislikes a given movie, i.e., as a fact (proposition) denoted by an RDF triple of the form (userB, *dislikes*, movieB). In other words, only the above-average ratings were considered as known or potentially known 'hits'. Finally, we have generated five datasets by randomly dividing the set of all known facts (propositions) into two groups – a train set and a test set. The data were divided according to the specified training *ratio*, denoted by x. To compensate for the impact that the randomness in the data set partitioning has on the results of the presented methods, each plot in this paper shows a series of values that represent averaged results of individual experiments.

3.7 Evaluated Recommendation Algorithms

We have compared several collaborative filtering data processing algorithms that are presented in the literature: popularity-based (favoring items having the higher number of ratings in the train set) [24], [4], Reflective Random Indexing (RRI) [3], PureSVD [4], and Randomized SVD (RSVD) [2]. All these methods have been applied to the input data matrix (in the case of both matrix-based and triple-based data representations) [4], [2], [24]. When the classical matrix-based model is used, the input matrix is decomposed, and then reconstructed in order to generate predicted ratings as presented in [22]. We have tested each of the methods, using the following parameters (where applicable):

- vector dimension: 256, 512, 768, 1024, 1536, 2048,
- seed length: 2, 4, 8,
- SVD k-cut: 2, 4, 6, 8, 10, 12, 14, 16, 20, 24.

The number of dimensions (i.e., the SVD k-cut value), which we have used in our experiments, may appear as quite small (for example, when compared to a typical LSI application scenario). Nevertheless, this choice has been made in order to avoid overfitting, in accordance to the assumptions concerning the dimensionality reduction sensitivity presented in [16]. Moreover, we have observed that for each investigated scenario the optimal algorithm performance was achieved for the SVD k-cut value that was less or equal to 16, so experiments for k-cuts higher than 24 were not necessary.

The combinations of parameters that lead to the best recommendation quality (i.e., the highest AUROC value) were considered optimal, and used in experiments illustrated in this paper. Table 1 shows the number of reflections used in the RRI and RSVD algorithms for each training ratio.

Table 1: **Number of RRI/RSVD reflections used for each training ratio.**

x (training ratio)	Method							
	S1		S2		S3		S4	
	RRI	RSVD	RRI	RSVD	RRI	RSVD	RRI	RSVD
0.2	3	7	5	7	8	15	3	3
0.3	3	5	3	7	10	9	3	3
0.4	3	5	3	7	10	9	3	3
0.5	3	5	3	5	8	9	3	3
0.6	3	5	3	5	8	11	3	3
0.7	3	5	3	5	10	5	3	3
0.8	3	3	3	5	8	11	3	3

Finally, we have compared the above-specified complex algorithms with a non-personalized baseline method - a popularity-based recommendation method (referred to as the 'Popularity'), following [4]. In this case the recommendation is based on the number of a given

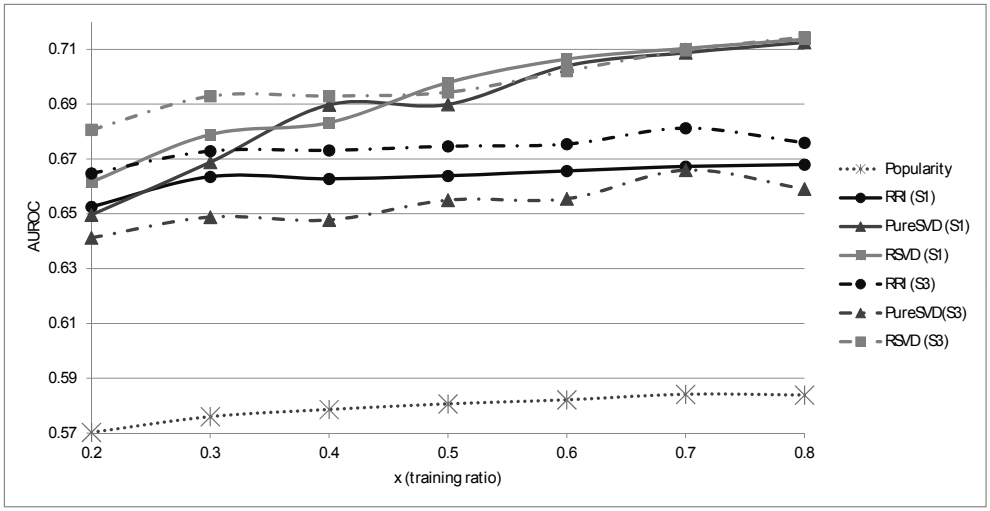


Figure 1: **AUROC results achieved in the S1 and S3 scenarios (the scenarios of using only the *Likes* predicates).**

item's ratings that appear in the train set. Items having the largest number of ratings (highest popularity) are recommended first.

3.8 Recommendation Accuracy Measure

We have compared the proposed data model with the standard one from the perspective of the application of each of these models within a complete collaborative filtering system. To obtain quantitative results of such an analysis, we have evaluated an ordered list of user-item recommendations by means of the AUROC measure.

4 Experiments

The scope of the experiments presented in the paper has been limited to the bi-relational recommendation scenario. Figures 1 and 2 show a comparison of the investigated recommendation algorithms (explicitly: popularity-based, RRI, PureSVD, RSVD), each using either the classical user-item or the element-fact matrix data representation. The comparison has been performed using the AUROC measure and datasets of various sparsity. The presented results have been obtained using optimized parameters for each method and each data model. Figure 1 presents AUROC evaluation results obtained for the case of using the dataset containing only *Likes* predicates (i.e., only the positive user feedback), whereas Figure 2 presents the analogical results obtained for the case of using the full dataset, i.e., the one containing both the *Likes* and *Dislikes* predicates (i.e., both the positive and the negative feedback).

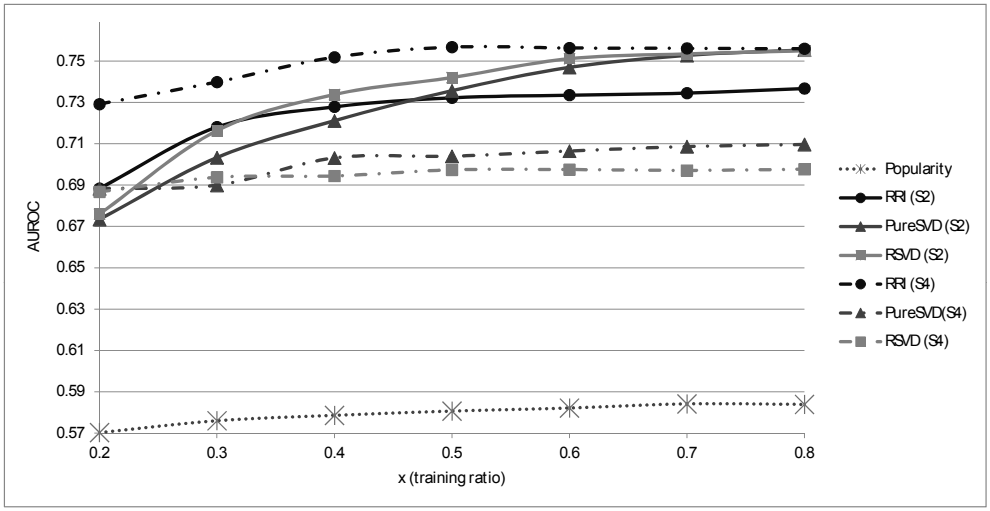


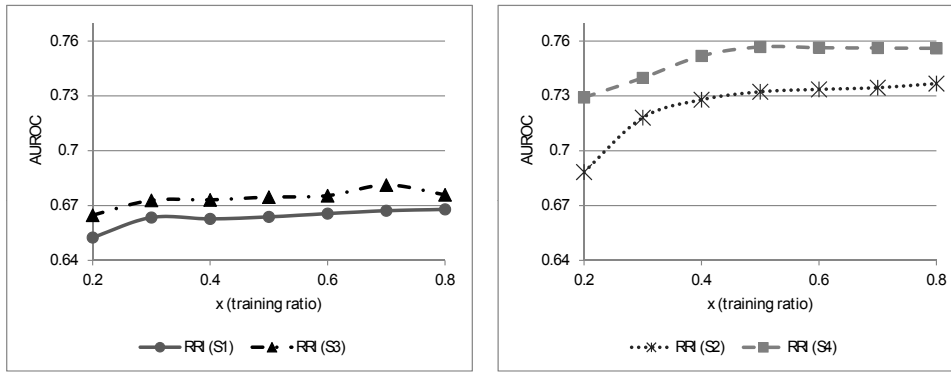
Figure 2: **AUROC results achieved in the S2 and S4 scenarios (the scenarios of using the full dataset - both the *Likes* and *Dislikes* predicates).**

As we have confirmed experimentally, the element-fact data representation matrix enables us to obtain a recommendation quality that is higher than the analogical results obtained with the use of the classical user-item matrix data model. It can be observed that the advantage of the proposed model is especially visible in the case of employing the full dataset (containing both *Likes* and *Dislikes* predicates) and the RRI method. Such behavior is a result of the more 'native' ability to represent multiple relations that is provided by the element-fact model.

One may realize that the popularity-based algorithm, instead of modeling users' preference profiles, simply reflects the ratio between positive ratings ('hits') and negative ratings ('misses') for the most popular items in a given dataset. Since we use a random procedure to divide the data set into a train set and a test set, the values of AUROC observed for the popularity-based algorithm are almost identical for the case of both $x = 0.2$ and $x = 0.8$, what additionally confirms the reliability of our AUROC measurement (see Figures 1 and 2).

In Figures 3, 5, and 4 the impact of the data representation method on the performance evaluation results is presented. We may conclude that the application of the new triple-based data representation method, accompanied with the Hadamard-based reconstruction technique, improves the results of using RRI for both mono- and bi-relational datasets (see Figures 3a and 3b). Moreover, for the case of using the bi-relational dataset, RRI outperforms any other method presented in the paper. We may conclude that, in the context of the proposed data representation scheme, the calculation of the 1-norm length of the Hadamard product is an operation that is synergic to the reflective data processing.

On the other hand, the application of the new representation method, accompanied with the reconstruction technique based on the Hadamard product, decreases the quality of results of using PureSVD for both mono- and bi-relational datasets (see Figures 4a and 4b). The reason of such a behavior is the fact that the prediction method based on the Hadamard product is not compatible with the data processing techniques based on the SVD decom-



(a) Mono-relational dataset

(b) Bi-relational dataset

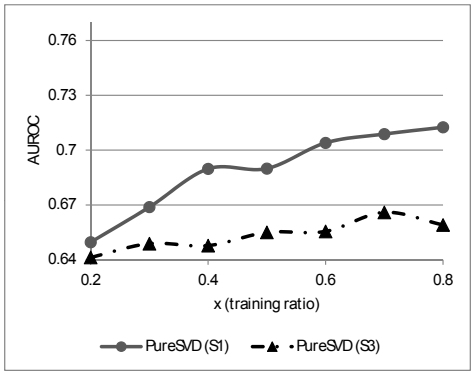
Figure 3: **Impact of the input data representation method on AUROC results achieved using the RRI method.**

position – in the case of using the dimensionality reduction, an input matrix reconstruction result should rather be used directly as the set of the prediction values. The comparatively low quality of the method based on PureSVD and Hadamard product may be explained by the non-probabilistic nature of the SVD results: it is especially evident in cases when the vectors multiplied together (by means of the Hadamard product) have negative coordinates, what indicates that they obviously have no probabilistic interpretation.

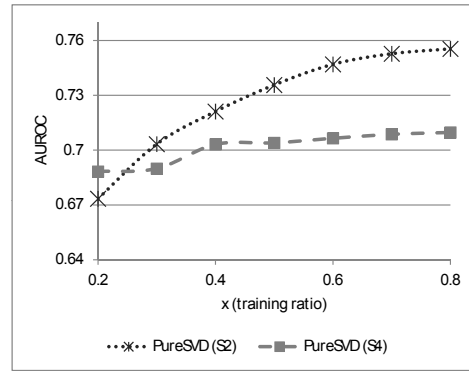
Furthermore, in the case of using RSVD (see Figures 5a and 5b), which is a combination of RI-based pre-processing and SVD-based vector space optimization, the application of the new data representation method improves the performance when the mono-relational dataset is concerned (especially for small numbers of ratio x). On the other hand, the application of the new data representation method decreases the system performance for the bi-relational dataset scenario (see Figure 5b) for the same reasons as in the case of using PureSVD.

It may be additionally concluded that when the methods based on the dimensionality reduction are used, the new representation method performs relatively (i.e., with respect to results obtained for standard representation methods) better for smaller values of ratio x, i.e., for sparser datasets for which the recommendation task is harder than for bigger values of x.

Figure 6 presents the performance of the CF algorithms as compared in the investigated scenarios (i.e., in scenarios S1-4). We may conclude that, as it was already shown in [2], the RSVD method outperforms other methods (i.e., pureSVD and RRI) when the standard input data representation is used (see Figures 6a and 6b). As far as the S1 scenario is concerned, i.e., the one with the standard data representation based on the user-item coincidence matrix containing mono-relational data, it may be seen that, in general, the decomposition-based methods (i.e., PureSVD and RSVD) achieve comparable recommendation quality and that, in general, these methods perform better than RRI (for various values of x). It may also be seen that the decomposition-based methods behave quite differently in the S3 scenario, in which the novel, triple-based data representation is used: in such a case, RSVD is the method that not only outperforms all the other methods compared in the scenario (including PureSVD), but also provides a high recommendation quality for various values of x. When

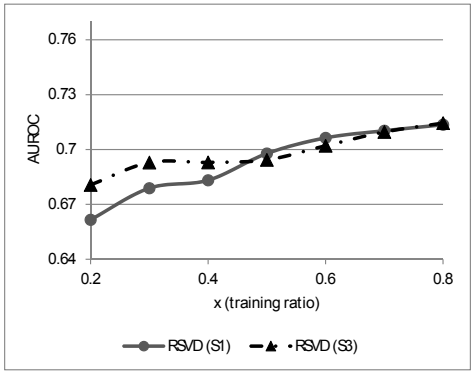


(a) Mono-relational dataset

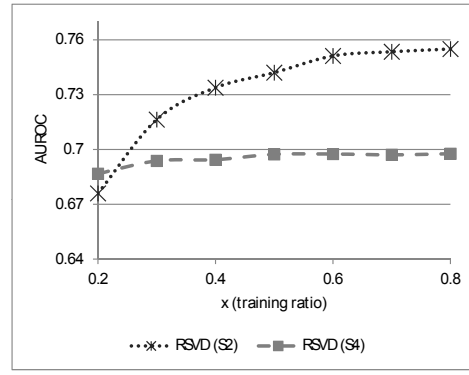


(b) Bi-relational dataset

Figure 4: **Impact of the input data representation method on AUROC results achieved using the PureSVD method.**

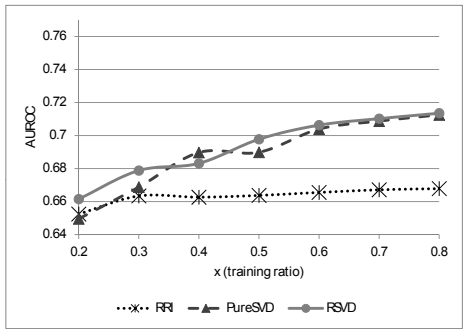


(a) Mono-relational dataset

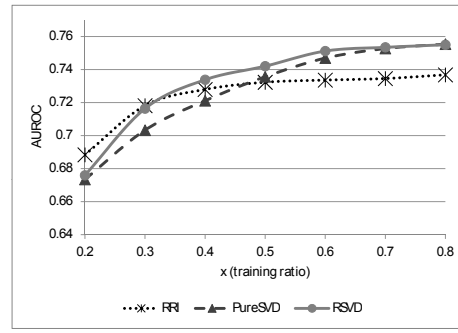


(b) Bi-relational dataset

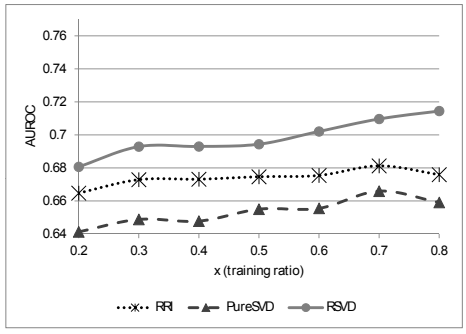
Figure 5: **Impact of the input data representation method on AUROC results achieved using the RSVD method.**



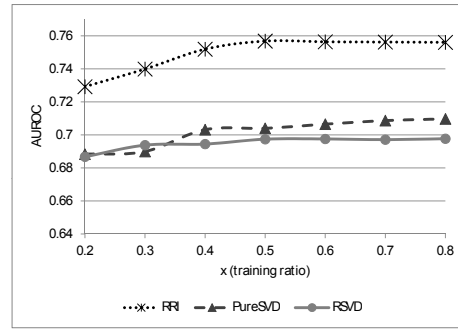
(a) Comparison of the presented CF algorithms for scenario S1



(b) Comparison of the presented CF algorithms for scenario S2



(c) Comparison of the presented CF algorithms for scenario S3



(d) Comparison of the presented CF algorithms for scenario S4

Figure 6: **Comparison of the presented CF algorithms for various scenarios.**

analyzed together, S1 and S3 scenarios show the superiority of RSVD in cases when mono-relational input data is used. Moreover, it may be seen that as long as RSVD is combined with the triple-based data representation, it provides recommendation quality that is the most reliable, i.e., which is higher than the quality observed when any other method is used for the most of the investigated values of x.

In the case of scenario S4 (i.e., for the application of the novel representation scheme for bi-relational data) RRI method outperforms both the decomposition-based methods, what shows the compatibility of the Hadamard-based reconstruction technique with the reflective processing of multi-relational data. Such a combination, i.e., the application of RRI together with the triple-based bi-relational input data representation, provides the highest recommendation quality among all the combinations presented in the paper. As the RRI method does not involve any computationally expensive spectral decomposition, we believe that this result is very valuable from the perspective of the practical applicability of the RRI-based recommendation systems in real-world scenarios.

The results of the experiments presented in the paper indicate that the presence of the additional information about the *Dislikes* relation improves the recommendation quality. The results for S2 and S4 scenarios (see Figure 2) are significantly better than the results obtained

in S1 and S3 scenarios (see Figure 1). However, the main conclusion from the experiments is that the best quality is observed in scenario S4 (in which we applied the proposed data representation and prediction method) for the case of the RRI-based data processing application (see Figure 3).

It may be concluded that the new framework proposed in this paper consists of two core elements: the new data representation method based on the element-fact matrix, and the new prediction calculation technique based on the Hadamard product of vectors. The 1-norm length of the vector Hadamard product may be seen as a natural extension of the vector dot product (in our case - as a 'kind' of 'group inner product' of the three vectors representing the RDF subject, object, and predicate) whereas the dot product may be seen as an elemental step of the matrix multiplication, i.e., the basic operation used in reflective matrix processing (each cell of a matrix multiplication result is in fact a dot product). Therefore, the calculation of the 1-norm length of the vector Hadamard product may be regarded as an operation compatible with the reflective matrix processing, and seen as an 'additional reflection' (i.e., the next step of the reflective data exploration process). This observation may additionally explain why the optimal number of reflections for the RRI method in the S4 scenario is relatively small (equal to 3 for each training ratio) - see Table 1. On the other hand, the prediction based on the Hadamard product does not suit well the data processing techniques based on SVD decomposition. This explains the relatively weak results of the dimensionality reduction methods in the scenarios in which the proposed data modeling method is used. In the case of using the techniques based on the dimensionality reduction, the input matrix reconstruction result is used directly as the set of the prediction values and an additional step of the Hadamard product calculation procedure is not required.

5 Conclusions

We have shown that the proposed triple-based approach to data representation allows us to improve the quality of collaborative filtering. We have investigated the application of the proposed data representation scheme in systems featuring the most widely-known methods for input data processing, such as the SVD-based dimensionality reduction [4] and the reflective matrix processing [3], [2]. We have shown that using the proposed bi-relational matrix data representation together with reflective data processing enables the researcher to design a collaborative filtering system outperforming systems based on the application of the dimensionality reduction technique.

Similarly to [25], we have demonstrated the superiority of multiple matrix data 'reflections' by realizing a new kind of spreading activation. However, the purpose of our spreading activation mechanism is not to prime some selected nodes (e.g., representing keywords) for search purposes (e.g., to identify the most activated nodes representing documents or web pages), but to realize the probabilistic reasoning about any RDF triple that may be composed of the subjects, predicates, and objects appearing in the input RDF graph.

While taking the perspective of related areas of research (such as SRL and web scale reasoning), one may find particularly interesting to investigate our proposal of using the 1-norm length of the Hadamard product (a 'quasi-similarity 'of a given triple's constituents) as an unknown triple likelihood measure. We believe that, as a result of realizing probabilistic

reasoning as a vector-space technique, our solution provides a basic means for extending the capacity for reasoning on RDF data beyond the boundaries provided by widely-known non-statistical methods.

In our opinion, the application of our methods (in particular, the new data representation and the new Hadamard-product-based unknown fact likelihood calculation) leads to a significant recommendation quality improvement, at least, for the case of using the reflective matrix processing. Although in the paper we focus on the bi-relational collaborative filtering scenario (with *Likes* and *Dislikes* as the only predicates), one may also find the proposed approach to matrix-based propositional data representation promising from the perspective of its extendibility to multi-relational applications.

Acknowledgments. This work is supported by the Polish Ministry of Science and Higher Education, grant N N516 196737, and by Polish National Science Centre under grant DEC-2011/01/D/ST6/06788.

References

[1] 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*, San Diego, CA, USA, 1999, pp. 201–212.

[2] Ciesielczyk M., Szwabe A., RSVD-based Dimensionality Reduction for Recommender Systems, *International Journal of Machine Learning and Computing*, vol. **1**, no. 2, 2011, 170–175.

[3] Cohen T., Schaneveldt R., Widdows D., Reflective Random Indexing and Indirect Inference: A Scalable Method for Discovery of Implicit Connections, *Journal of Biomedical Informatics*, **43**, 2, 2010, 240–256.

[4] Cremonesi P., Koren Y., Turrin R., Performance of Recommender Algorithms on Top-n Recommendation Tasks, in: *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*, New York, NY, USA, 2010, pp. 39-–46.

[5] Damljanovic D., Petrak J., Lupu M., Cunningham H., Carlsson M., Engstrom G., Andersson B., Random Indexing for Finding Similar Nodes within Large RDF graphs, in: R. Garcia-Castro, D. Fensel, G. Antoniou (eds.), *The Semantic Web: ESWC 2011 Workshops*, LNCS, vol. 7117, Springer, Berlin Heidelberg, 2011, 156–171.

[6] Fensel D., van Harmelen F., Unifying reasoning and search to web scale, *IEEE Internet Computing*, **11**(2), 96, 2007, 94—95.

[7] Fouss F., Francoisse K., Yen L., Pirotte A., Saerens M., An experimental investigation of kernels on graphs for collaborative recommendation and semi-supervised classification, *Neural Networks*, **31**, 2012, 53–72.

[8] Franz T., Schultz A., Sizov S., Staab S., Triplerank: Ranking Semantic Web Data by Tensor Decomposition, in: *Proceedings of The Semantic Web-ISWC*, 2009, pp. 213–228.

[9] Herlocker J.L., Konstan J.A., Terveen L.G., Riedl J.T., Evaluating Collaborative Filtering Recommender Systems, *ACM Trans. Information Systems*, vol. **22**, no. 1, 2004, 5–53.

[10] Li Y., Hu J., Zhai C.X., Chen Y., Improving one-class collaborative filtering by incorporating rich user information, in: *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM '10)*, ACM, New York, NY, USA, 2010, pp. 959–968.

[11] Manning Ch. D., Raghavan P., Schütze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[12] Nickel M., Tresp V., Kriegel H.P., A Three-Way Model for Collective Learning on Multi-Relational Data, in: *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 809–816.

[13] Pan R., Zhou Y., Cao B., Liu N. N., Lukose R., Scholz M., Yang Q.: One-Class Collaborative Filtering. Technical Report. HPL-2008-48R1, HP Laboratories, 2008.

[14] Pitowsky I., Quantum Probability, Quantum Logic, *Lecture Notes in Physics*, **321**, Heidelberg, Springer, 1989.

[15] Saganowski S., Brodka P., Kazienko P., Influence of the User Importance Measure on the Group Evolution Discovery, *Foundations of Computing and Decision Sciences*, 2012, 295–305.

[16] Sarwar B., Karypis G., Konstan J., Riedl J., Application of Dimensionality Reduction in Recommender System-A Case Study, in: *Proceedings of the ACM EC'00 Conference*, Minneapolis, 2000, pp. 158–167.

[17] Sindhwani V., Bucak S.S., Hu J., Mojsilovic A., A Family of Non-negative Matrix Factorizations for One-Class Collaborative Filtering Problems, in: *Proceedings of the ACM Recommender Systems Conference, RecSys '2009*, New York, 2009.

[18] Singh A.P., Gordon G.J., Relational Learning via Collective Matrix Factorization, in: *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 650–658.

[19] Steck H., Training and testing of recommender systems on data missing not at random, in: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '10)*, ACM, New York, NY, USA, 2010, pp. 713–722.

[20] Struyf J., Blockeel H., Relational Learning, in: C. Sammut, G. Webb (eds.), *Encyclopedia of Machine Learning*, Springer, 2010, 851–857.

[21] Sutskever I., Salakhutdinov R., Tenenbaum J. B.: Modelling Relational Data Using Bayesian Clustered Tensor Factorization, *Advances in Neural Information Processing Systems*, **22**, 2009.

[22] Szwabe A., Ciesielczyk M., Janasiewicz T., Semantically enhanced collaborative filtering based on RSVD, in: P. Jedrzejowicz , N.-T. Nguyen, K. Hoang (eds.), *Computational Collective Intelligence, Technologies and Applications*, LNCS, vol. 6923, Springer, Berlin Heidelberg, 2011, 10–19.

[23] Szwabe A., Misiorek P., Walkowiak P., Reflective Relational Learning for Ontology Alignment, in: S. Omaru et al. (eds.), *Distributed Computing and Artificial Intelligence, Advances in Intelligent and Soft Computing*, vol. 151, Springer-Verlag Berlin/Heidelberg, 2012, 519–526.

[24] Szwabe A., Ciesielczyk M., Misiorek P., Long-tail Recommendation Based on Reflective Indexing, in: D. Wang, M. Reynolds (eds.), *AI 2011: Advances in Artificial Intelligence*, LNCS/LNAI, vol. 7106, Springer, Berlin/Heidelberg, 2011, 142–151.

[25] Todorova P., Kiryakov A., Ognyano D., Peikov I., Velkov R., Tashev Z., Conclusions from Experimental Data and Combinatorics Analysis. LarKC project deliverable 2.7.3, Technical Report, The Large Knowledge Collider (LarKC), 2009.

[26] van Rijsbergen C.J.: The geometry of IR, *The Geometry of Information Retrieval*, Cambridge University Press, New York, USA, 2004, pp. 73–101.

Received November, 2012