

Research Article

On Efficient Link Recommendation in Social Networks Using Actor-Fact Matrices

Michał Ciesielczyk, Andrzej Szwabe, and Mikołaj Morzy

Poznan University of Technology, Piotrowo 2, 60-965 Poznan, Poland

Correspondence should be addressed to Mikołaj Morzy; mikolaj.morzy@put.poznan.pl

Received 28 February 2014; Revised 21 November 2014; Accepted 21 November 2014

Academic Editor: Reda Alhajj

Copyright © 2015 Michał Ciesielczyk et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Link recommendation is a popular research subject in the field of social network analysis and mining. Often, the main emphasis is put on the development of new recommendation algorithms, semantic enhancements to existing solutions, design of new similarity measures, and so forth. However, relatively little scientific attention has been paid to the impact that various data representation models have on the performance of recommendation algorithms. And by performance we do not mean the time or memory efficiency of algorithms, but the precision and recall of recommender systems. Our recent findings unanimously show that the choice of network representation model has an important and measurable impact on the quality of recommendations. In this paper we argue that the computation quality of link recommendation algorithms depends significantly on the social network representation and we advocate the use of actor-fact matrix as the best alternative. We verify our findings using several state-of-the-art link recommendation algorithms, such as SVD, RSVD, and RRI using both single-relation and multirelation dataset.

1. Introduction

Link recommendation, along with link prediction, is a popular research topic in the domain of social network analysis and mining [1]. Numerous algorithms have been proposed over the years [2]. The main objective of link recommendation and prediction is to predict, based on the historical data, unobserved relationships and interactions between actors of a social network [3]. It should be stressed here that the term “link” is used here freely, as the task can refer to predicting possible (existing or future) relationships between people, recommending interesting resources to actors of the network, or discovering latent similarities between objects. Usually, a distinction is drawn between link prediction (where the task is to evaluate the probability of a given relationship’s existence between actors) and link recommendation (where the task is to select top k resources relevant to a given actor). One can see however that it is relatively easy to combine the two tasks under a single framework. For the sake of brevity we will refer to both problems as “link recommendation” throughout this paper. Link recommendation is predicated on the existence of data, either panel data or event data [4]. Panel data refer

to snapshots of the social network taken at certain intervals and representing possibly a coarse-grained view of existing relationships. In contrast, event data refer to detailed records of activities between actors in the network. Event data is time-stamped and fine-grained and often results from automated measurements or transactions. These two types of data are merged and processed and split into a training set and a test set for the purpose of training of link recommendation models.

Although link recommendation tasks have attracted significant attention of the scientific community over the last years, in our opinion relatively little work has been done on the impact of data representation models on the quality of recommendations. By far the most popular data representation model is an actor-object matrix, where actors of the social network are represented as rows and objects that are the subject of recommendations are represented as columns. The cells of such matrix may contain either binary flags to denote the existence of a relation (e.g., Adam likes “The Police”), or a value of the relation, both discrete and numerical (e.g., Beth ate at “Pizza Paradise” and rated it with 4.5 stars). One may note that the social network need not be

a bipartite graph. When the relation is defined between actors (e.g., Carol likes Douglas), the actor-object matrix becomes simply a square matrix. The situation becomes slightly more complex in case of multirelational social networks, where multiple different relations, of possibly varying semantics, may exist between actors in the network. A typical example is a network where actors may express both fondness of and rejection of certain objects (e.g. Eve likes to watch comedy movies but she hates horror movies). If the storage of relation values is permitted by a given data model, multirelational networks may be modeled by assigning distinct values (or sets of values) to particular relations, but for a binary actor-object matrix it is necessary to represent each relation by a separate matrix and to include processing of multiple matrices by the recommendation algorithm.

In this paper we argue that actor-object matrix is not the optimal data model for recommendation algorithms. Our experiments conclusively show that transformation from the actor-object to the actor-fact matrix improves recommendation quality significantly, as measured by the popular “area under receiver-operator characteristic curve” (AUROC) measure. We perform extensive experiments on a large real-world dataset to support our claims. Given the fact that the vast majority of link recommendation algorithms for social networks compute actor-object, actor-actor, or object-object similarities by applying linear algebra on data representation matrices, the superiority of actor-fact matrix representation becomes quite obvious (in particular for methods which are generally based on singular value decomposition paradigm). The original contribution of this paper consists in the introduction of two elements:

- (i) a data representation method based on a binary actor-fact matrix,
- (ii) a similarity quasimeasure based on the 1-norm length of the Hadamard product of the given tuple of vectors.

Our key finding is that the proposed data representation and the new similarity measure, when combined with reflexive matrix processing, significantly outperform state-of-the-art collaborative filtering methods based on the use of a standard actor-object matrix.

Our paper is organized as follows. In Section 2 we report on the related work on the subject and we present the referenced recommendation algorithms. Section 3 introduces the concept of the actor-fact matrix. In Section 4 we present the evaluation methodology of the actor-fact matrix representation and we report the results of conducted experiments in Section 5. The paper concludes in Section 6 with a brief summary.

2. Related Work

By far the most popular approach to link recommendation in social networks is collaborative filtering using an input matrix which represents each actor as a vector in the space of objects and each object as a vector in the space of actors. Many previous works consider building a model of collaborative similarity from a model of content-based interobject relations

to be the most promising hybrid link recommendation technique [5, 6]. As far as algebraic representations of graph data is concerned, the actor-fact matrix model is similar to the model described in [7]. Indeed, our model was inspired by the semantic data model of RDF triples. Also, as far as the algebraic transformation of the graph data is concerned, the model presented in this paper may be regarded as similar to RDF data search methods which are based on spreading activation realized by means of iterative matrix data processing [8] or single multiplication by a random projection matrix [7]. However, the latter method is limited to the RDF graph node search using a traditional bilateral similarity measure, whereas we extend the model by using a vector-space quasisimilarity measure which allows to efficiently compute the likelihood of an unknown relationship.

In our evaluation we use three main types of collaborative filtering recommender algorithms. The baseline is established by a simple popularity-based algorithm favoring objects having the highest number of positive relationships in the train set [9]. Next, we have employed several different approaches to the input matrix decomposition. Firstly, we have used the algorithm based on reflexive random indexing [10]. Secondly, we have used two types of algorithms that are based on the singular value decomposition: a traditional implementation of the method (PureSVD), in which actor vectors are represented as combinations of object vectors without any specific parameterization, and an implementation of the randomized singular value decomposition (RSVD) [11], which is a combination of the reflexive random indexing and SVD. We have chosen so since SVD-based methods have been long considered to be the most efficient recommender engines in real world settings [12–15].

Section 5 presents the results of conducted experiments. Since our data have the form of binary prepositions (i.e. our social network is a signed network), the evaluation of the proposed method is oriented on the task of finding relevant links [16] rather than on the minimization of recommendation rating error. Classification metrics, such as area under ROC (AUROC), measure the probability of making correct or incorrect decisions by the recommender algorithm about whether an object is relevant. Moreover, classification metrics tolerate the differences between actual and predicted values, as long as they do not lead to wrong decisions. Thus, these metrics are appropriate to examine binary relevance relationships. In particular, while using AUROC it is assumed that the ordering among relevant items does not matter. According to [17], AUROC is equivalent to the probability of the system being able to choose properly between two objects, one randomly selected from the set of relevant objects and one randomly selected from the set of nonrelevant objects. For this reason, the results of the theoretical research are evaluated by means of experiments based on quality measures that are probabilistically interpretable such as AUROC.

3. Actor-Fact Matrix

Let us recall that our model is influenced by the semantic model of RDF triples. Each RDF triple combines information about the predicate that relates a subject to an object. We

consider a generic social network (for simplicity we constrain ourselves to nonvalued relations, but the proposed method may be easily extended to valued relations) which conceptually consists of a set of actors $A = \{a_1, \dots, a_n\}$, a set of objects $O = \{o_1, \dots, o_m\}$, and a set of relations $R = \{R_1, \dots, R_l\}$, where each relation R_i represents a function $R_i : A \times O \rightarrow \{0, 1\}$. Let us now combine all actors, objects, and possible predicates into a single set $E = A \cup O \cup R$. Furthermore, let $|A| = n$, $|O| = m$, and $|R| = l$. Of course, there is no requirement to have the set of actors be separate from the set of objects; that is, in general it is possible that $A \subseteq O$. It should be noted though that if sets A and O would overlap, that is, if they would be represented by the same vectors, it would not be possible to take advantage of the semantics of actors constituting relationships. In other words, putting actors and objects together into a single set would make it impossible to distinguish between semantically correct relationships, such as “Alice likes apples,” and semantically incorrect relationships, such as “apples like Alice.” Being able to encode such semantics directly in social network matrix representation is obviously a very desirable property, but this issue is out of the scope of this paper.

We refer to the set of actual instances of relations as the set of *facts* denoted by F , and let $|F| = f$. The binary actor-fact matrix is defined as $X = [x_{i,j}]_{(n+m+l) \times f}$, where each column of the matrix X represents a single fact (i.e., an existing dyad connected in the social network by a relation), each row of the matrix X represents an entity (actor, object, or relation), and each column contains exactly three nonzero entries, that is, for each j there exist exactly three nonzero entries $x_{i_1,j}$, $x_{i_2,j}$, and $x_{i_3,j}$, such that $1 \leq i_1 \leq n$, $n+1 \leq i_2 \leq n+m$, and $n+m+1 \leq i_3 \leq n+m+l$ (the rows containing these three nonzero entries correspond to the actor, object, and relation of a given dyad, or, in the RDF parlance, to the subject, predicate, and the object of a triple). At the same time the number of nonzero entries in each row represents the number of dyads in which a given actor/object participates, or the number of dyads of a given relation.

Let us consider a simple social network depicted in Figure 1. It represents two different relationships between actors *Alice*, *Bob*, *Titanic*, and *Star Wars*. The relationships between these actors include *liking* and *being a friend of*. Implicitly, we understand that *liking* is a relationship between an actor representing a person and an actor representing a movie, whereas *being a friend of* is a relationship between two actors representing persons. This network can be easily transformed into the actor-fact model. There are three facts that exist in this network:

- (i) $fact_1$: Alice is a friend of Bob,
- (ii) $fact_2$: Alice likes Titanic,
- (iii) $fact_3$: Bob likes Star Wars.

In our actor-fact matrix representation $fact_1$ will constitute a column in the matrix X and this column will have nonzero entries for cells $X[Alice, fact_1]$, $X[is\ a\ friend\ of, fact_1]$, and $X[Bob, fact_1]$. The entire social network from Figure 1 is presented in the actor-fact matrix representation in Table 1.

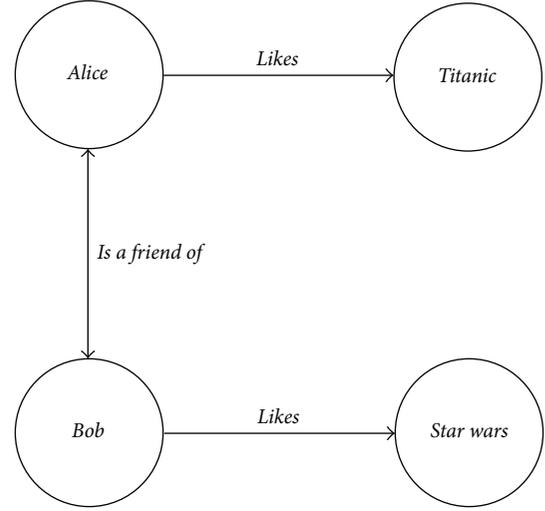


FIGURE 1: Example of a social network.

TABLE 1: Actor-fact matrix representing the network from Figure 1.

	$fact_1$	$fact_2$	$fact_3$
Alice	1	1	0
Bob	1	0	1
Is a friend of	1	0	0
Likes	0	1	1
Titanic	0	1	0
Star Wars	0	0	1

When using the actor-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 values of the input matrix are estimated. In order to achieve this, the input matrix X is processed into its reconstructed form \tilde{X} using one of the evaluated recommendation algorithms. Afterwards, each of the predictions is calculated as the 1-norm length of the Hadamard product of row vectors. The Hadamard product (also known as the Schur product or the entrywise product) of two vectors $X = [x_1, \dots, x_n]$ and $Y = [y_1, \dots, y_n]$ of the same length n is defined as

$$X \circ Y = [x_1 \cdot y_1, \dots, x_n \cdot y_n]. \quad (1)$$

Each dyad forms the proposition which is the subject of the likelihood estimation. More formally, the prediction value $P_{i,j,k}$ is calculated according to the formula

$$P_{i,j,k} = \left\| \widehat{x}_i \circ \widehat{x}_j \circ \widehat{x}_k \right\|_1, \quad (2)$$

where \widehat{x}_i , \widehat{x}_j , and \widehat{x}_k are the row vectors of the reconstructed matrix $\tilde{X} = [\widehat{x}_{i,j}]_{(n+m+l) \times f}$ corresponding to the elements of the given dyad, and the symbol \circ represents the Hadamard product.

For instance, using the proposed measure on the example shown in Table 1 one may predict the likelihood of Alice liking the movie Titanic (i.e., the likelihood of the joint

incidence of actors *Alice*, *likes*, and *Titanic* represented by row vectors $X[\textit{Alice},] = [1, 1, 0]$, $X[\textit{likes},] = [0, 1, 1]$, and $X[\textit{Titanic},] = [0, 1, 0]$, resp.). This likelihood equals $\|[1, 1, 0] \circ [0, 1, 1] \circ [0, 1, 1]\| = 1$. Conversely, the likelihood of any nonexistent fact, such as *Bob likes Titanic*, equals 0. Naturally, the practical value of such a measure is to estimate the likelihood of missing links after the application of appropriate collaborative filtering algorithms.

The proposed formula may be seen as a generalization of the dot product formula, as in the hypothetical case of measuring quasisimilarity of two (rather than three) vectors, the formula is equivalent to the dot product of the two vectors. It should be also noted that the measure may be easily extended to larger number of vectors. The interpretation of the proposed formula as the likelihood of the joint incidence of two or more facts represented as vectors is based on the quantum information retrieval model [18]. 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 information retrieval model of probability calculation.

4. Evaluation Methodology

Let us now present the evaluation methodology for the experiments. Our goal is to quantitatively compare two matrix-based methods of social network representation: the classical actor-object matrix and the new actor-fact matrix from the point of view of the link recommendation task. Taking into consideration that link recommendation tasks may vary, we have additionally considered two subproblems: a one-class link recommendation, where the aim is to discover only the missing links of a single relation (e.g., for a given actor recommend to her a set of possible new friends), and the biclass link recommendation, where the aim is to discover missing links of one particular relation while not recommending any of the links of another relation (e.g., for a given actor, show him possible friends who share theatrical preferences, but do not recommend any new movies). The combination of the two results in the following four scenarios:

- (i) S1: using single relation and an actor-object matrix $B = [b_{i,j}]_{n \times m}$, where n is the number of actors and m is the number of objects, a scenario which corresponds to friend recommendations using only information on friendship between actors,
- (ii) S2: using two antagonistic relations and an actor-object matrix $B = [b_{i,j}]_{n \times m}$, where n is the number of actors and m is the number of objects, a scenario which corresponds to friend recommendations using information on friendship and dislike between actors,
- (iii) S3: using single relation and an actor-fact matrix $X = [x_{i,j}]_{(n+m+l) \times f}$, where n is the number of actors, m is the number of objects, and $l = 1$ is the number of predicates (in this case a single predicate),

- (iv) S4: using two antagonistic relations and an actor-fact matrix $X = [x_{i,j}]_{(n+m+l) \times f}$, where n is the number of actors, m is the number of objects, and $l = 2$ is the number of predicates.

In order to evaluate the effect of data representation model on collaborative filtering methods, we have decided to use one of the most widely referenced datasets in the recommender systems area. We have deliberately chosen to turn a typical recommender system dataset into an artificial social network, instead of using a genuine network (e.g., Facebook friend graph or Twitter followers graph), because we also wanted to compare our results with previous results; thus we needed a well-established benchmark dataset. MovieLens ML100k set was collected over various periods of time from Internet users who expressed their opinions on different movies in order to receive personalized recommendations. It contains 100 000 ratings of 1682 movies given by 943 unique users. Each rating which is above the average for a given movie has been treated as an indication that a user likes the movie. Analogically, each rating below the average has been used as an indication that a user dislikes the movie. Finally, train and test data sets were generated by randomly dividing the set of all known facts into two subsets. The data were divided according to the specified training ratio, denoted by tr . To compensate for the impact that the randomness in the dataset partitioning has on the results of the presented methods, each plot in this section shows a series of values that represent averaged results of individual experiments.

As we have previously stated, four recommendation algorithms are used: a simple popularity-based method, a traditional SVD (PureSVD), a randomized version of SVD (RSVD), and a reflexive random indexing (RRI). To clarify, the actual algorithm being used is the collaborative filtering (CF), but it works on a matrix decomposed using the above algorithms. The decomposition of the original matrix is of course necessary to make collaborative filtering computation feasible in practice. In real social networks the size of the matrix (actor-object, and actor-fact in particular) is so huge that vector similarity computation in original dimensions is impossible. Each of the methods has been tested using the following parameters (where applicable):

- (i) vector dimension: 256, 512, 768, 1024, 1536, 2048;
- (ii) seed length: 2, 4, 8;
- (iii) 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 the experiments, may appear as quite small when compared to a typical LSI application scenario. This choice has been made in order to avoid overfitting, in accordance with the assumptions concerning the dimensionality reduction sensitivity presented in [14]. Moreover, it has been 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. We have also varied the number of reflections used in RRI and RSVD between 3 and 15.

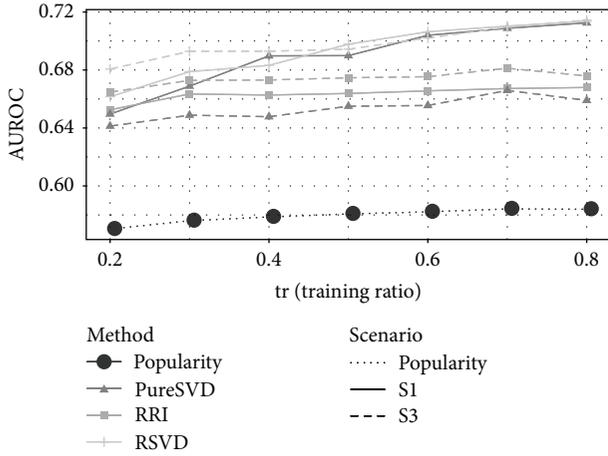


FIGURE 2: AUROC results for S1 and S3 scenarios.

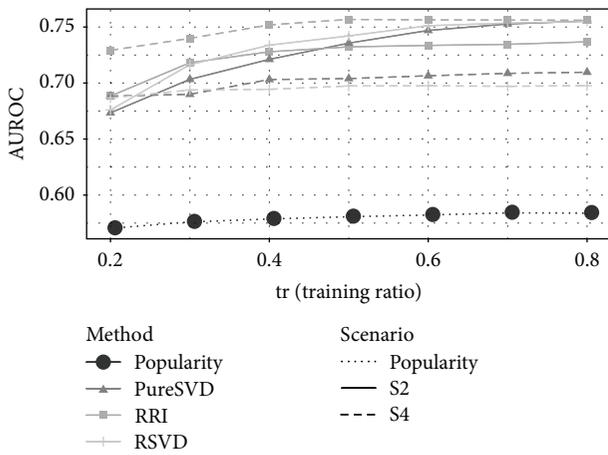


FIGURE 3: AUROC results for S2 and S4 scenarios.

5. Experiments

Figures 2 and 3 show a comparison of the investigated recommendation algorithms, each using either the classical actor-object or the actor-factor 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 2 presents AUROC evaluation results obtained for the case of using the network consisting of a single relation (i.e., only positive ratings), whereas Figure 3 presents analogical results obtained for the case of using the full network, that is, the one containing both positive and negative relations.

As it has been confirmed experimentally, the actor-factor data representation matrix obtains recommendation quality which is higher than the analogical results obtained with the use of the classical actor-object matrix representation. It can be observed that the advantage of the proposed model is especially visible in the case of employing the full network containing both positive and negative relations, and the RRI method. Such behavior is the result of the more native ability

to represent multiple relations provided by the actor-factor model.

One may realize that the popularity-based algorithm, instead of modeling actors' preference profiles, simply reflects the ratio between the number of positive relation instances (hits) and negative relation instances (misses) for the most popular objects in a given network. Since a random procedure is used to divide the dataset 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 $tr = 0.2$ and $tr = 0.8$, which additionally confirms the reliability of the AUROC measurement.

In Figures 4, 5, and 6 the impact of the data representation method on the performance evaluation results is presented. As can be seen, the application of the new fact-based data representation method, accompanied with the Hadamard-based reconstruction technique, improves the results of using RRI for both single and multiple relations networks (see Figures 4(a) and 4(b)). Moreover, for the case of using the network with multiple relations, RRI outperforms any other presented method. It may be concluded 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 by the reconstruction technique based on the Hadamard product, decreases the quality of results of using PureSVD for both single and multiple relation networks (see Figures 5(a) and 5(b)). The reason of such 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 decomposition. In the case of using the SVD dimensionality reduction, an input matrix reconstruction result should rather be used directly as the set of prediction values. The comparatively low quality of the method based on PureSVD and Hadamard product may be explained by the nonprobabilistic nature of SVD results: it is especially evident in cases when the vectors multiplied together (by means of the Hadamard product) have negative coordinates, which indicates that they obviously have no probabilistic interpretation.

Furthermore, in the case of using RSVD (see Figures 6(a) and 6(b)), 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 single relation network is concerned (especially for small numbers of the ratio tr). On the other hand, the application of the new data representation method decreases the system performance for the multiple relations network scenario for the same reasons as in the case of using PureSVD. It may be additionally concluded that when the methods based on 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 the ratio tr , that is, for sparser datasets for which the recommendation task is harder.

Figure 7 presents the performance of the recommender algorithms as compared in the investigated scenarios (i.e., in scenarios S1–4). It may be concluded that, as it was already

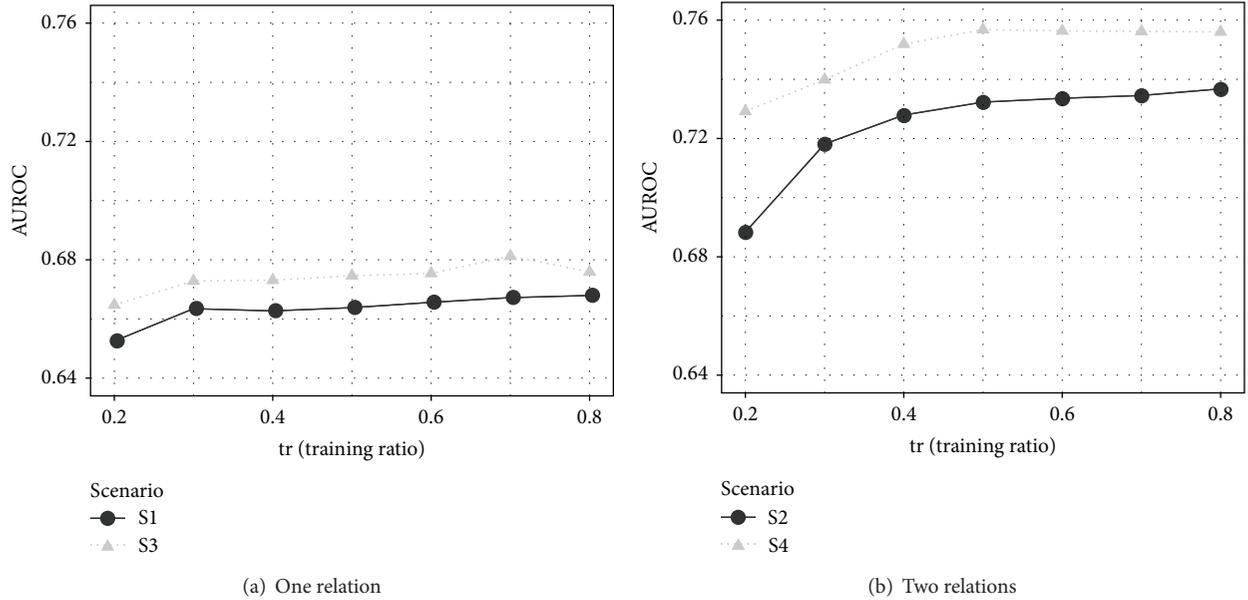


FIGURE 4: Impact of the input data representation method on AUROC results achieved using the RRI method.

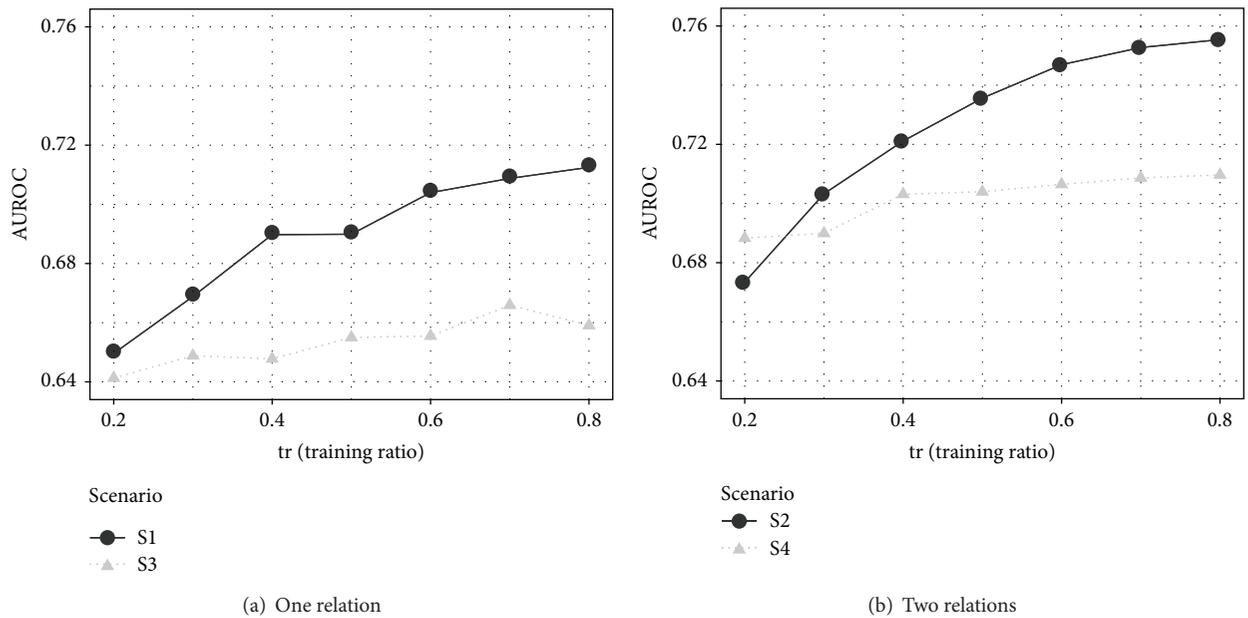


FIGURE 5: Impact of the input data representation method on AUROC results achieved using the PureSVD method.

shown in [11], the RSVD method outperforms other methods (i.e., PureSVD and RRI) when the standard input data representation is used. As far as the S1 scenario is concerned, that is, the one with the standard data representation based on the actor-object coincidence matrix single relation, 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 the training ratio). It may also be seen that the decomposition-based methods behave

quite differently in the S3 scenario, in which the novel, fact-based data representation is used: in such case, RSVD is the method which not only outperforms all the other methods compared in the scenario (including PureSVD), but also provides a high recommendation quality for various values of the training ratio. When analyzed together, S1 and S3 scenarios show the superiority of RSVD in cases when single relation network is used. Moreover, as long as RSVD is combined with the fact-based data representation, it provides recommendation quality that is the most reliable, which is

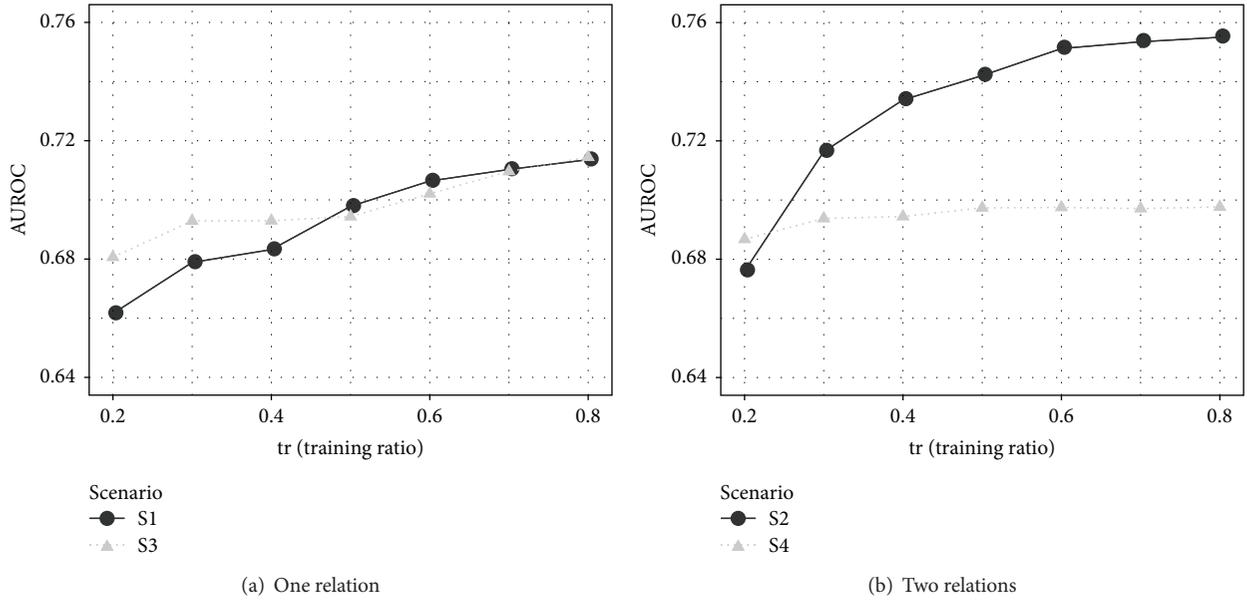


FIGURE 6: Impact of the input data representation method on AUROC results achieved using the RSVD method.

higher than the quality observed when any other method is used for the majority of investigated values of the training ratio. In the case of scenario S4 (fact-based data representation with multiple relations) RRI method outperforms both decomposition-based methods, which shows the compatibility of the Hadamard-based reconstruction technique with the reflective processing of multirelational data. Such combination, that is, the application of RRI together with the fact-based multirelational data representation, provides the highest recommendation quality among all the combinations presented in this paper. As the RRI method does not involve any computationally expensive spectral decomposition, this result may be very valuable from the perspective of the practical applicability of the RRI-based link recommendation systems in real-world scenarios.

The results of the experiments presented herein clearly indicate that the presence of the additional information about the negative relation improves the recommendation quality. The results for S2 and S4 scenarios (see Figure 3) are significantly better than the results obtained in S1 and S3 scenarios (see Figure 2). However, the main conclusion from the experiments is that the best quality is observed in scenario S4 (in which the proposed data representation and prediction method has been applied) for the case of the RRI-based data processing application.

The results of the comparison show that, in general, as long as the proposed multirelational actor-fact matrix data representation is used, the reflective processing methods (in particular RRI) outperform the well-known SVD-based dimensionality reduction methods. While trying to explain this observation, one may note that the typical actor-object matrix (representing only positive relations between actors and objects) is equivalent to a part of another much bigger matrix. This bigger matrix may be obtained as the result of multiplying the actor-fact matrix (with both actors and

objects represented as the rows) by its transposition. The “submatrix” of the bigger matrix (together with its transposed “clone”) is just a typical collaborative filtering matrix—it represents the “magnitudes” of the actor-object positive preference relation. Demonstrating this correspondence between the object-fact matrix format and widely used actor-object matrix format (typically used together with the SVD-based dimensionality reduction) requires an additional matrix multiplication (i.e., an additional reflection). Therefore, it may be expected that, as long as the proposed data representation is used, only reflective data processing methods can take full advantage of using it by applying appropriately many reflections. To put this observation (confirmed by the results of the experiments presented herein) in other words, while using the fact-based data representation, SVD-based collaborative filtering methods need at least one more matrix multiplication to provide the recommendation quality comparable to the quality achieved by means of the optimized reflective matrix processing.

6. Conclusions

The new framework proposed in this paper consists of two core elements: the new data representation method based on the actor-fact matrix and the new prediction calculation technique based on the Hadamard product of vectors. The 1-norm length of the Hadamard product vector may be seen as a natural extension of the vector dot product (in this case as a kind of group inner product of the three vectors representing the actor, the object, and the relation) whereas the dot product may be seen as an elemental step of the matrix multiplication, that is, the basic operation used in reflective matrix processing. Therefore, the calculation of the 1-norm length of the Hadamard product vector may be regarded as an operation compatible with the reflective matrix processing,

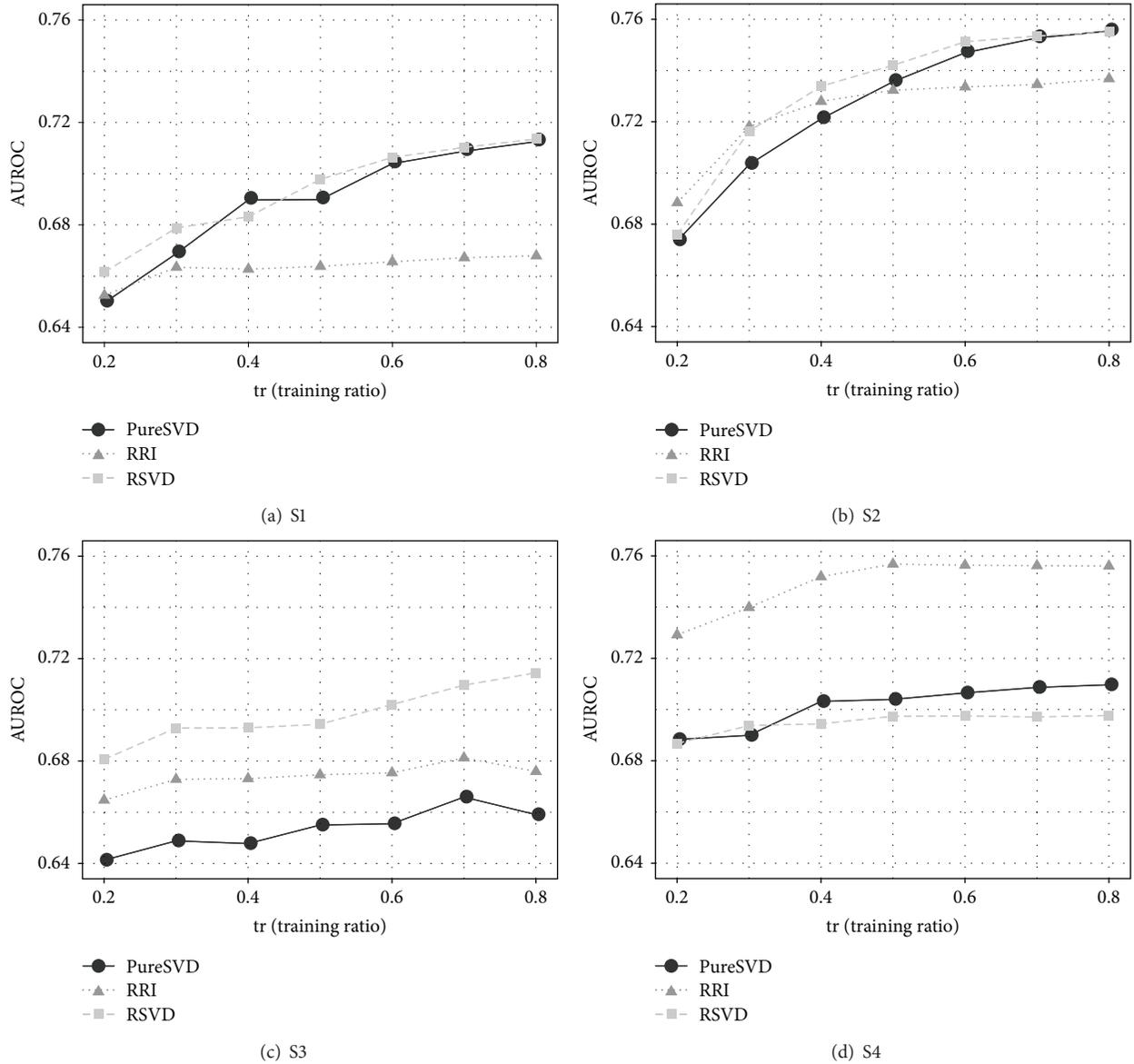


FIGURE 7: Comparison of all recommender algorithms for various scenarios.

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). 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 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.

We have shown that the proposed fact-based approach to social network representation allows to improve the quality

of collaborative filtering. The application of the proposed actor-factor matrix in systems featuring the most widely known methods for input data processing, such as the SVD-based dimensionality reduction and the reflective matrix processing, has been investigated. We have also shown that using the actor-factor matrix together with reflective data processing enables us to design a collaborative system outperforming systems based on the application of the dimensionality reduction techniques.

We have demonstrated the superiority of multiple matrix data reflections by realizing a new kind of spreading activation. However, the purpose of the spreading activation mechanism introduced herein is to realize the probabilistic reasoning about any fact that may be composed of actors, relations, and objects appearing in the network. To state it more precisely, in order to estimate the probability that

a given fact represents a true statement, the three constituents of the fact are independently primed. On the basis of the three independently generated vectors, each one representing levels of node activation obtained as the result of priming the node represented by the vector, the 1-norm length of a Hadamard product is applied to measure holistically (i.e., by taking into account the state of all nodes) the amount of joint similarity of the three fact constituents or, more precisely, their representations that have been obtained as the result of the spreading activation procedure.

While taking the perspective of related areas of research (such as Web scale reasoning), one may find it particularly interesting to investigate our proposal of using the 1-norm length of the Hadamard product as the measure of an unknown dyad likelihood. The authors believe that, due to probabilistic reasoning as a vector-space technique, the introduced solution provides basic means for extending the capacity for reasoning on social networks beyond the boundaries provided by currently used nonstatistical methods. In our opinion, the application of introduced methods (in particular, the new actor-fact data representation and the new Hadamard product likelihood calculation) leads to a significant link recommendation quality improvement, at least for the case of using the reflective matrix processing. Although this paper provided an evaluation using only the two relations scenario, one may also find the proposed approach to matrix-based propositional data representation to be promising from the perspective of its extendability to truly multirelational applications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

Mikołaj Morzy has been supported by the National Science Centre Grant 2011/03/B/ST6/01563. Michał Ciesielczyk and Andrzej Szwabe have been supported by the National Science Centre Grant DEC-2011/01/D/ST6/06788.

References

- [1] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.
- [2] L. L. Linyuan and T. Zhou, "Link prediction in complex networks: a survey," *Physica A: Statistical Mechanics and Its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.
- [3] R. Lichtenwalter and N. V. Chawla, "Link prediction: fair and effective evaluation," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '12)*, pp. 376–383, August 2012.
- [4] C. Lee, B. Nick, U. Brandes, and P. Cunningham, "Link prediction with social vector clocks," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2013.
- [5] B. Mobasher, X. Jin, and Y. Zhou, "Semantically enhanced collaborative filtering on the Web," in *Proceedings of the 1st European Web Mining Forum (EWMF '03)*, pp. 57–76, Springer, Cavtat-Dubrovnik, Croatia, September 2003.
- [6] J. Salter and N. Antonopoulos, "CinemaScreen recommender agent: combining collaborative and content-based filtering," *IEEE Intelligent Systems*, vol. 21, no. 1, pp. 35–41, 2006.
- [7] D. Damjanovic, J. Petrak, M. Lupu et al., "Random indexing for finding similar nodes within large RDF graphs," in *The Semantic Web: ESWC 2011 Workshops*, vol. 7117 of *Lecture Notes in Computer Science*, pp. 156–171, Springer, Berlin, Germany, 2012.
- [8] P. Todorova, A. Kiryakov, D. Ogniano, I. Peikov, R. Velkov, and Z. Tashev, "Conclusions from experimental data and combinatorics analysis," Tech. Rep., The Large Knowledge Collider (LarKC), 2009.
- [9] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-N recommendation tasks," in *Proceedings of the 4th ACM Recommender Systems Conference (RecSys '10)*, pp. 39–46, ACM, September 2010.
- [10] T. Cohen, R. Schvaneveldt, and D. Widdows, "Reflective Random Indexing and indirect inference: a scalable method for discovery of implicit connections," *Journal of Biomedical Informatics*, vol. 43, no. 2, pp. 240–256, 2010.
- [11] M. Ciesielczyk and A. Szwabe, "RSVD-based dimensionality reduction for recommender systems," *International Journal of Machine Learning and Computing*, vol. 1, no. 2, pp. 170–175, 2011.
- [12] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [13] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC '00)*, pp. 158–167, 2000.
- [15] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," in *Applications of Data Mining to Electronic Commerce*, pp. 115–153, Springer, New York, NY, USA, 2001.
- [16] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.
- [17] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [18] I. Pitowsky, "Quantum mechanics as a theory of probability," in *Physical Theory and Its Interpretation*, vol. 72 of *The Western Ontario Series in Philosophy of Science*, pp. 213–240, Springer, Dordrecht, The Netherlands, 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

