

HOLISTIC ENTROPY REDUCTION FOR COLLABORATIVE FILTERING

Andrzej SZWABE *, Pawel MISIOREK *, Tadeusz JANASIEWICZ *,
Przemyslaw WALKOWIAK *

Abstract. We propose a collaborative filtering (CF) method that uses behavioral data provided as propositions having the RDF-compliant form of $(user_X, likes, item_Y)$ triples. The method involves the application of a novel self-configuration technique for the generation of vector-space representations optimized from the information-theoretic perspective. The method, referred to as Holistic Probabilistic Modus Ponendo Ponens (HPMPP), enables reasoning about the likelihood of unknown facts. The proposed vector-space graph representation model is based on the probabilistic apparatus of quantum Information Retrieval and on the compatibility of all operators representing subjects, predicates, objects and facts. The dual graph-vector representation of the available propositional data enables the entropy-reducing transformation and supports the compositionality of mutually compatible representations. As shown in the experiments presented in the paper, the compositionality of the vector-space representations allows an HPMPP-based recommendation system to identify which of the unknown facts having the triple form $(user_X, likes, item_Y)$ are the most likely to be true in a way that is both effective and, in contrast to methods proposed so far, fully automatic.

Keywords: collaborative filtering, self-configuration, propositional RDF-compliant data representation, quantum IR, information theory

1 Introduction

The integration of probabilistic reasoning with machine learning and the use of relational representations – which is the subject of studies on Statistical Relational Learning (SRL) – is ranked among the main challenges of research on Artificial Intelligence (AI) [14]. Due to the universal nature of the fact-based data representation, any significant progress in investigation of vector-space propositional data modeling and processing methods is very likely

*Institute of Control and Information Engineering, Poznan University of Technology, M. Skłodowskiej-Curie Square 5, 60-965 Poznan, Poland, e-mail: pawel.misiorek@put.poznan.pl

to be valuable for research on SRL and recommendation systems, as well as for Information Retrieval (IR), Machine Learning (ML), and Natural Language Processing (NLP).

Algebraic approaches to SRL deserve a serious interest of the IR community, in particular as far as the progress in research on the vector-space data representation and transformation methods is concerned. It should be also pointed out that typical ‘IR-like’ recommendation methods and SRL-based recommendation methods share the same evaluation methodology originally proposed by authors working in the field of IR [18]. Moreover, the task of user action prediction (naturally based on the use of a propositional data represented as a set of facts) is strongly related to the ‘find good items’ task that defines one of the most important recommendation system application scenarios [7].

In contrast to the application of a typical (i.e., non-SRL) recommendation system, the application of an SRL-based recommendation system is usually based of assumption of using a probabilistic inference system. In turn, the application of theoretically grounded probabilistic inference may eliminate the need for experimental tuning of the inference system parameters. Surprisingly, despite the importance of the recommender system self-configurability (which is especially evident in the context of real-world applications), the authors of solutions presented in the relevant literature have not seriously undertaken the self-configurability challenge [1, 5, 12]. Self-configurability is essential for on-line applications which have to take into account the most recent information when generating recommendations (e.g., the information about the user session or her/his last actions). In such a scenario the data necessary to refine the parameters of collaborative filtering system are not available at the stage of system configuration. Additionally, advanced collaborative filtering systems usually use some additional data in order to augment the information about users and items. These data is usually of heterogeneous structure what makes the task of the manual parameter configuration even harder.

The recommendation system proposed in the paper may be seen as an application of a new method for reasoning about unknown facts, that we refer to as Holistic Probabilistic Modus Ponendo Ponens (HPMPP). To our knowledge, the vector-space graph representation method, being one of the key elements of HPMPP (exemplifying a group of methods that in the graph theory literature are referred to as graph embedding methods [20]), is the first propositional data representation method that is based on the probabilistic apparatus of quantum Information Retrieval [2, 19]. Moreover, the properties of the proposed model enable a synergetic combination of the core elements of the HPMPP system: the use of a propositional data graph of a carefully designed probabilistic edge weighting scheme and the use of a vector-space data transformation that enables the fully automatic operation of the HPMPP system.

The paper presents the very first application of the proposed framework. Evaluation scenarios investigated here are limited to the single-relational case. Such a choice is a consequence of the assumption that one of the core issues of the paper is the analysis of the information-theoretical aspects of the HPMPP data processing algorithm and their influence on the ability of making the algorithm auto-configurable. The choice of the simplified experimental scenario has enabled us to observe the spectral features of the processing results, which may be conveniently analyzed and illustrated in small-scale experiments. Nevertheless, the authors regard the further experimental evaluation of the proposed method, in particular in heterogeneous multi-relational domains, as one of the key points of the future work.

2 Related work

The approach presented in the paper is not the first application of an algebraic SRL method to a personalized recommendation system [15, 21, 22]. However, rather than being based on the widely-used dimensionality reduction technique or, in general, on some factorization heuristics [9, 15], our method follows the quantum approach to IR [2, 19] and is the first SRL method that applies an information-theoretic model for the optimality of the vector-space representation generation.

Recently algebraic SRL solutions methods based on the 3rd order tensor decomposition [9, 15] have emerged as an important alternative to graphical models [14]. Our approach may be regarded as an algebraic method capable of representing propositions of various arity (including RDF triples), which makes it an alternative to tensor-based approaches. Moreover, HPMPP provides the possibility of modeling unconstrained structures (complete weighted graphs). To our knowledge, the framework of HPMPP is the first that makes it possible to model probabilistically all the elements (i.e., subjects, objects and predicates) and all the facts as compatible and composable vector-space representations.

The proposed method may also be regarded as a contribution to research on information-theoretic optimization of CF systems [24]. HPMPP is the first method that explicitly applies a model of gradual holistic entropy reduction, without compromising entropies of individual representations [12, 24]. The majority of the papers on the information-theoretical optimization of collaborative filtering focus on the application of the proven maximum entropy rule (MaxEnt) to the process of individual representations generation [23, 24]. Although it has been additionally observed (e.g., in [6]) that adding relevance feedback information reduces the system holistic entropy, so far neither a model nor an algorithm has been proposed that explicitly targets the decrease of the holistic dataset representation entropy achieved in parallel with the increase of the item representation entropy. In this paper, we provide such an algorithm and analyze it from the information-theoretic perspective.

To our knowledge, none of the existing approaches to collaborative filtering (see [12] for a comprehensive survey) is oriented toward automatic self-configuration. Although there are many collaborative filtering systems that are able to provide a high quality recommendation for particular application scenarios, they require purely experimental parameter optimization (e.g., involving searches for the optimal number of components used to represent data [5, 13] or for the optimal number of reflections [4]). In the present paper, we propose a solution that is fully automatic due to the application of the probabilistically-grounded relational data representation and processing model of HPMPP. Our system is able to achieve a high recommendation quality for datasets of different sparsity and spectral characteristics (i.e., different heavy-tailness [16]).

3 Model

In contrast to the well-known collaborative filtering methods [5, 7], HPMPP is founded on a probabilistic model and avoids both the application of reflective random indexing [4] and the application of dimensionality reduction heuristics. The model proposed in the paper enables our algorithm to iteratively produce increasingly optimized vector-space representations of

subjects, predicates and objects (optimized with respect to the quality of recommendation provided on the basis of these representations), but also to effectively determine at which iteration the procedure should be terminated.

3.1 Basic Assumptions

We assume that the input data of the system are given as a set of propositions (i.e., facts) F describing the relationships between elements modeled in the system, where $(|F| = m)$.

The proposed model is not limited only to the case of RDF propositions – it enables to represent propositions of any arity greater or equal to 2. In general, we define $Y = \{y_1, \dots, y_n\}$ (such that $|Y| = n$) as the set of all possible elements which may be used to define facts from the set F . In the case of RDF propositions, which are propositions of arity 3, the set of elements Y is equal to $S \cup P \cup O$, where S is the set of subjects, P is the set of predicates, and O is the set of objects. The application scenario used in the tests presented in this paper involves modeling of propositions of arity 2, what is equivalent to the presence of the single relation between users and movies. Thus, in the investigated case, the set Y is just a sum of the set of users and the set of items not including any predicate. In a case of processing more than one relation, the model would involve the use of propositions of arity 3 (i.e., propositions equivalent to the RDF triples), since we assume to model predicates exactly in the same way as other elements. For the general case of propositions of arity l the set Y may be modeled as $Y = Y_1 \cup Y_2 \cup \dots \cup Y_l$, where Y_i denotes the domain of i -th element in the proposition ($i = 1, \dots, l$).

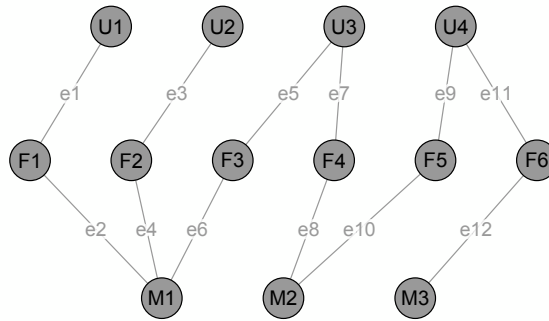


Figure 1: The example of graph G .

The structure of the input data may be represented as a bipartite graph $G = (V(G), E(G))$ specifying coincidences of (i.e., ‘links’ between) propositions (i.e. facts) and elements, where $V(G) = Y \cup F$ is a set of graph vertices and $E(G) \subseteq Y \times F$ is a set of graph edges. Each edge of this graph $\{v_i, v_j\}$ represents the information that the element modeled by the vertex v_i is used to define the fact modeled by the vertex v_j .

Moreover, we introduce the notion of events for the purposes of the probability estimation. We associate each element and each fact with the event of the ‘observation’ of the element’s or the fact’s presence in the input data [19]. In particular, B_i denotes the event associated with a given entity (i.e., with a given element or fact), which is modeled by the vertex $v_i \in V(G)$ where $1 \leq i \leq n + m$. In Figure 1 an example of graph G for the case of modeling propositions of arity 2 is presented. The example corresponds to the application scenario in which the propositions modeled in the system represent a set of facts constituting a single relation (i.e., the likes relation). Such a scenario has been also assumed in the experiments reported in this paper. In particular, for the illustrated example we have:

- $V(G) = Y \cup F$, where $Y = \{U1, \dots, U4\} \cup \{M1, \dots, M3\}$ is a set of elements including four users and three movies, and $F = \{F1, \dots, F6\}$ is a set of six propositions – each representing the fact that a given user likes a given movie,
- and $E(G) = \{e1, \dots, e12\}$.

3.2 Input Data Representation

Our method enables the generation of compatible representations [19] for all elements and facts modeled in the system (vertices of G). The coincidences of elements and facts (edges of G) are the atomic units of the modeled information. According to these two assumptions, the incidence matrix $M = [m_{i,j}]_{(n+m) \times r}$ of graph G (where $|E(G)| = r$) is used to represent the structure of the input data. The matrix M is constructed as follows:

1. The incidences of edge $e_j \in E(G)$, ($1 \leq j \leq r$), and the vertices constituting the ends of the edge are encoded by setting the corresponding matrix entries to 1 and -1 :

$$m_{i,j} = \begin{cases} 1 & \text{if there exists } k \text{ such that edge } e_j = \{v_i, v_k\} \text{ and } i < k, \\ -1 & \text{if there exists } k \text{ such that edge } e_j = \{v_i, v_k\} \text{ and } i > k, \\ 0 & \text{otherwise.} \end{cases}$$

2. Each row of matrix M is normalized using norm 2. The row normalization step is an application of the Principle of Indifference with respect to prior probabilities of events represented in the system. We use norm 2 in order to model all the atomic events as equally probable quantum states represented by unit vectors in a Hilbert space of r dimensions [19].

A form of the incidence matrix obtained as a result of this procedure is known as the normalized oriented incidence matrix of a given graph. Matrix M for the graph presented in

Figure 1 is equal to:

$$M = \begin{matrix} & e1 & e2 & e3 & e4 & e5 & e6 & e7 & e8 & e9 & e10 & e11 & e12 \\ \begin{matrix} U1 \\ U2 \\ U3 \\ U4 \\ M1 \\ M2 \\ M3 \\ F1 \\ F2 \\ F3 \\ F4 \\ F5 \\ F6 \end{matrix} & \left[\begin{array}{cccccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{array} \right] \end{matrix}.$$

At the next step, a normalized Laplacian matrix [20] of G is constructed according to formula:

$$L(G) = MM^T = [l_{i,j}]_{(n+m) \times (n+m)},$$

where

$$l_{i,j} = \begin{cases} 1 & \text{if } i = j, \\ -\frac{1}{\sqrt{d_i(G)d_j(G)}} & \text{if } \{v_i, v_j\} \in E(G), \\ 0 & \text{otherwise,} \end{cases}$$

where $d_i(G)$ is the degree of vertex v_i of graph G . The Laplacian matrix is the operator on Hilbert space which may be regarded as representing the observable of the quantum system [10]. The smallest eigenvalue of $L(G)$ ($\lambda_0(L(G))$) is equal to 0 and it corresponds to the eigenvector of coordinates proportional to square roots of degrees of graph G vertices [8]. The central role in the data transformation process is played by matrix $A = I - L(G) = [a_{i,j}]_{(n+m) \times (n+m)}$ (where $I = I_{n+m}$ is the $(n+m)$ -dimensional identity matrix), for which

$$a_{i,j} = \begin{cases} \frac{1}{\sqrt{d_i(G)d_j(G)}} & \text{if } \{v_i, v_j\} \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

A is the adjacency matrix of \overline{G} , which is the weighted version of graph G with edge weighting $w(\{v_i, v_j\}) = a_{i,j}$. In the case of the example presented in Figure 1, the matrix A takes a form:

$$A = \begin{matrix} & \begin{matrix} U1 & U2 & U3 & U4 & M1 & M2 & M3 & F1 & F2 & F3 & F4 & F5 & F6 \end{matrix} \\ \begin{matrix} U1 \\ U2 \\ U3 \\ U4 \\ M1 \\ M2 \\ M3 \\ F1 \\ F2 \\ F3 \\ F4 \\ F5 \\ F6 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & \frac{1}{\sqrt{6}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{6}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{\sqrt{6}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}.$$

The normalized Laplacian matrix has all eigenvalues within interval $[0, 2]$, where $0 = \lambda_1(L(G)) \leq \lambda_2(L(G)) \leq \dots \leq \lambda_{n+m}(L(G)) = 2$. Since $A = I - L(G)$, the eigenvalues of A are equal to $\lambda_i(A) = 1 - \lambda_i(L(G))$ and share their eigenvectors set with $L(G)$. Therefore, $\lambda_i(A) \in [-1, 1]$.

3.3 Holistic Entropy Reduction

The data transformation realized by the HPMPP system is based on an iterative transformation of the Hermitian operator describing the quantum system (represented by A). The transformation results in a gradual reduction of the von Neumann entropy [10]. The process may also be seen as the exploration of \bar{G} resulting in the increase of the similarities between event representations.

As our approach is based on the quantum IR theory of Hermitian operators [19], we model the holistic entropy as von Neumann entropy [10]. To compute it, we normalize the trace of a given operator in order to obtain a statistical quantum operator complying with the definition of the density matrix [10]. The von Neumann entropy is calculated on the basis of the density matrix eigencomposition according to the formula $S(\rho) = -\sum_i \lambda_i \ln \lambda_i$, where λ_i denotes the eigenvalues of the density matrix.

The representation generation process starts with the operator having the maximum entropy (equal to $\log_2(n + m)$) and, if not terminated by an optimization algorithm, ends by producing an operator of the minimal entropy. The process outcome is the sequence of operators $A^{(t)}$ ($t \geq 0$) defined in the following way:

- $t = 0$: $A^{(0)} = [a_{i,j}^{(0)}]_{(n+m) \times (n+m)} = I_{(n+m) \times (n+m)}$,
- $t \geq 1$: $A^{(t)} = A^{(t-1)} \cdot A = [a_{i,j}^{(t)}]_{(n+m) \times (n+m)}$, where $a_{i,j}^{(t)} = \sum_{k=1}^{n+m} a_{i,k}^{(t-1)} a_{k,j}$.

The entries of $A^{(t)} = [a_{i,j}^{(t)}]$ correspond to probabilities of events' coincidences $P(B_i \cap B_j)$ (events B_i are defined in Subsection 3.1) based on a reasoning chain of length t (i.e., the estimated number of t -step walks on the weighted graph \overline{G} starting at vertex i and finishing at vertex j).

The final result of the representation generation process is an arithmetic average of operators $A^{(t)}$, i.e., the matrix $\overline{A}^{(k)} = [\overline{a}_{i,j}^{(k)}]_{(n+m) \times (m+n)}$, where $\overline{a}_{i,j}^{(k)} = \frac{1}{k+1} \sum_{t=0}^k a_{i,j}^{(t)}$ and $k \geq 0$. The use of the arithmetic average in the construction of operator $\overline{A}^{(k)}$ may be regarded as an application of the Principle of Indifference with respect to the length of the graph exploration process. $\overline{A}^{(k)}$ may be seen as a partial (i.e. approximate) inverse of $L(G)$ realized by means of Neumann series $\sum_{i=0}^k (I - L(G))^i = \sum_{i=0}^k A^i$. Due to the fact that the Laplacian matrix has an eigenvalue equal to zero, it is formally not invertible and the Neumann series does not converge, but approaches a zero-entropy state. This state corresponds to the projector on the eigenvector of $L(G)$ associated to the eigenvalue that is equal to zero (the vector representing the stationary distribution of a random walk on graph \overline{G}) [8].

Operators $A^{(i)}$ and operators $\overline{A}^{(k)}$ have the same eigenbasis as operator A . Therefore, the eigenvalues of $\overline{A}^{(k)}$ ($k \geq 0$) may be calculated according to the formula:

$$\lambda_i(\overline{A}^{(k)}) = \begin{cases} 1 & \text{if } \lambda_i(A) = 1, \\ \frac{1}{k+1} \frac{1 - (\lambda_i(A))^{k+1}}{1 - \lambda_i(A)} & \text{otherwise,} \end{cases} \quad (1)$$

for $1 \leq i \leq n + m$.

Due to the fact that $\lambda_i(A) \in [-1, 1]$, formula (1) implies that $\lambda_i(\overline{A}^{(k)}) \in [0, 1]$ and that the entropy of operators $\overline{A}^{(k)}$ gradually decreases with the increase of k , i.e., $S(\overline{A}^{(k+1)}) \leq S(\overline{A}^{(k)})$ for each $k \geq 0$.

Due to the fact that the biggest eigenvalue of operator A is equal to 1, the biggest eigenvalue of each operator $A^{(t)}$ as well as of each operator $\overline{A}^{(k)}$ is also equal to 1. As a result of this property, the entries of these matrices (i.e., the values $a_{i,j}^{(t)}$, as well as values $\overline{a}_{i,j}^{(k)}$) may be regarded as mutually comparable probability estimates $P(B_i \cap B_j)$.

The entropy reduction process described above may be alternatively seen as a result of the exploration of input data (given as operator A) aimed at obtaining new estimations of probabilities of events coincidences. At the same time, such a process may be illustrated as the procedure of counting the expected number of walks on \overline{G} (i.e., the graph, for which A is its adjacency matrix) of, at most, k steps – the walks of the length less or equal to k .

For the purposes of the probability estimation, our model assumes the use of element representations taken from matrix $\overline{A}^{(k)}$ for the optimal value of k . The HPMPP model proposes two techniques for the optimization of k , which are derived from graph theory. The first one assumes using the *diameter* of graph G as the optimal number of k , whereas the other one uses the *mixing time* [8] of \overline{G} for this purpose. The diameter of the graph presented in the Figure 1 is equal to 10 whereas its mixing time is equal to 25. The block of the matrix $\overline{A}^{(25)}$ limited only to the rows and columns that correspond to the modeled elements (i.e., users and movies) is as follows:

$$\bar{A}_{7 \times 7}^{(25)} = \begin{matrix} & \begin{matrix} U1 & U2 & U3 & U4 & M1 & M2 & M3 \end{matrix} \\ \begin{matrix} U1 \\ U2 \\ U3 \\ U4 \\ M1 \\ M2 \\ M3 \end{matrix} & \begin{bmatrix} 0.133 & 0.056 & 0.047 & 0.010 & 0.113 & 0.023 & 0.004 \\ 0.056 & 0.133 & 0.047 & 0.010 & 0.113 & 0.023 & 0.004 \\ 0.047 & 0.047 & 0.148 & 0.054 & 0.099 & 0.089 & 0.030 \\ 0.010 & 0.010 & 0.054 & 0.202 & 0.024 & 0.111 & 0.127 \\ 0.113 & 0.113 & 0.099 & 0.024 & 0.222 & 0.049 & 0.012 \\ 0.022 & 0.022 & 0.089 & 0.111 & 0.049 & 0.162 & 0.066 \\ 0.004 & 0.004 & 0.030 & 0.127 & 0.012 & 0.066 & 0.155 \end{bmatrix} \end{matrix}.$$

3.4 Probability Estimation

The method used to compute the probability of any coincidence of events (of any arity) is based on the 1-norm length of the Hadamard product of vectors from matrix $\bar{A}^{(k)}$. For a given subset of events $Z = \{B_{i_1}, B_{i_2}, \dots, B_{i_s}\}$ such that $|Z| = s$ and i_k ($1 \leq k \leq s$) are distinct indices taken from set $\{1, \dots, n+m\}$, we calculate the probability estimation p_{i_1, i_2, \dots, i_s} as the 1-norm length of the Hadamard product of row vectors corresponding to these events, according to the formula:

$$p_{i_1, i_2, \dots, i_s} = \|\bar{a}_{i_1}^{(k)} \circ \bar{a}_{i_2}^{(k)} \circ \dots \circ \bar{a}_{i_s}^{(k)}\|_1,$$

where $\bar{a}_{i_1}^{(k)}$, $\bar{a}_{i_2}^{(k)}$, and $\bar{a}_{i_s}^{(k)}$ are the row vectors of matrix $\bar{A}^{(k)}$ that correspond to elements forming the given proposition, and the symbol $\bar{a}_{i_1}^{(k)} \circ \bar{a}_{i_2}^{(k)} \circ \dots \circ \bar{a}_{i_s}^{(k)}$ denotes the Hadamard product of vectors $\bar{a}_{i_1}^{(k)}$, $\bar{a}_{i_2}^{(k)}$, and $\bar{a}_{i_s}^{(k)}$. In the case of propositions of arity 2 the formula simplifies to the dot product of two vectors. Therefore, the proposed formula may be seen as a generalization of the dot product formula.

The model of computing the probability of the joint incidence of two or more events follows the approach of quantum probability modeling presented in [11] and [19]. 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 the Gleason's theorem [11], [19], which explains the correspondence between probability measure on the lattice of closed subspaces of the Hilbert space and the density operator on this space [11]. The procedure of the probability calculation may be described in terms of quantum observables (represented by Hermitian operators) corresponding both to the elements and to the facts modeled in the system.

As a result of such modeling assumptions (in particular, the operators' compatibility assumption), each vector representation may be used to build a diagonal operator exemplifying a quantum observable of the probability of the given entity occurrence in the dataset. Probability of entities' coincidences (i.e., observables of propositions) are obtained as products of its constituents observables [19], [10]. The probability calculation is done as a result of quantum measurement procedure, i.e., as an expectation value of the respective observables [11].

In the case of the graph presented in Figure 1 the final probabilities are calculated as dot products of rows of matrix $\bar{A}^{(10)}$ (for the optimization parameter k equal to the diameter of G)

or $\bar{A}^{(25)}$ (for the optimization parameter k equal to the mixing time of G). The probabilities corresponding to pairs (*user*, *movie*) are equal to:

$$p_{U_i, M_j} = \begin{array}{c} U1 \\ U2 \\ U3 \\ U4 \end{array} \begin{array}{ccc} M1 & M2 & M3 \\ \left[\begin{array}{ccc} 0.126 & 0.019 & 0.002 \\ 0.126 & 0.019 & 0.002 \\ 0.106 & 0.099 & 0.027 \\ 0.019 & 0.122 & 0.145 \end{array} \right] \end{array}$$

for the case of the HPMPP diameter-based method, and

$$p_{U_i, M_j} = \begin{array}{c} U1 \\ U2 \\ U3 \\ U4 \end{array} \begin{array}{ccc} M1 & M2 & M3 \\ \left[\begin{array}{ccc} 0.104 & 0.039 & 0.015 \\ 0.104 & 0.039 & 0.015 \\ 0.107 & 0.085 & 0.048 \\ 0.052 & 0.110 & 0.108 \end{array} \right] \end{array}.$$

for the case of the HPMPP mixing-time-based method.

4 Experiments

The experiments presented in this section have been conducted with the aim of comparing HPMPP with state-of-the-art collaborative filtering solutions from the perspective of the recommendation quality. Moreover, all the methods have been analyzed from the information-theoretic perspective. In order to provide reliable and realistic results, the experiments involve using datasets of various sparsity (various training-set/testing-set ratios) [13] and various heavy-tailness (i.e., various spectral features of input data) [16].

4.1 Datasets

We have selected MovieLens [25] as the experimentation dataset, as it is undoubtedly the most popular collaborative filtering dataset that is widely referred to in the relevant literature. The MovieLens (ML100k) dataset (available at [25]) consists of 100000 ratings (provided by means of a five-star scale) concerning 1682 movies given by 943 users. The majority of the ML100k ratings is condensed in a small set of the most popular items (called the short head) [5]. For this reason, we have made our experiments more realistic by reflecting the observation made by the authors of [5] that recommending the most popular items is rarely beneficial for the users. Following [16], we have removed a specified number of the most popular items to produce a long-tailed dataset. The percentage of items remaining in the dataset after removing items from the short head has been defined as lt .

Furthermore, we have randomly partitioned each dataset into pairs of a training set and a test set, according to the ratio x as shown in [13]. A training set constructed for a given x and

Table 1: Numbers of triples in $ML(lt)$ datasets.

Dataset (lt):	ML100k (0.6)	ML100k (0.8)	ML100k (0.945)
Likes	6735	18341	34669
Dislikes	6268	16915	31051

a given lt is denoted as $ML(x, lt)$. In our experiments, we have used $x \in \{0.2, 0.5, 0.8\}$ and $lt \in \{0.6, 0.8, 0.945\}$.

Using a heavy-tailed dataset for the analysis of the representation entropy reduction process implies a dataset spectral characteristic that allows for slower entropy changes, and, in turn, for a more precise analysis. Setting $lt = 0.6$ has been chosen as the smallest value of parameter lt for which training set $ML(0.5, lt)$ (obtained after removing ratings from the short head) still contains a single dominating connected graph component; choosing a smaller value of lt would lead to decomposing the training set into several disconnected ‘datasets’.

The main assumption with regard to the dataset preparation was that extracted RDF triples should represent data of both the *likes* and *dislikes* relations. Following the approach presented in [16] (which was based on the MovieLens dataset analysis presented in [5]), we have subtracted the arithmetic average of the user average rate and the item average rate from each known rate. All the users’ ratings below zero have been interpreted (both for the training set and the test set) as facts: $(user_X, dislikes, movie_Y)$ and those above zero as facts: $(user_X, likes, movie_Y)$. The numbers of propositions remaining in the MovieLens dataset after removing items from the short head (for different values of lt) and after the transformation of ratings into the RDF triples of *likes* and *dislikes* relations are presented in Table 1.

In this paper, we present the first application of the proposed framework for the scenario of one-relational training sets (consisting of propositions of the *likes* relation only). As a consequence of using one-relational training sets for each compared method, the observed recommendation quality for the experiments presented in this paper is lower than the quality observed in experiments on the ML100k dataset including all the ratings (i.e., in [5] or [16]).

4.2 Evaluation Methodology

HPMPP has been compared with three widely-referenced methods, namely, pure Singular Value Decomposition (SVD) [13], Reflective Random Indexing (RRI) [4], Randomized SVD-RI (RSVD-RI) [3]. For both RRI and RSVD-RI, the same configuration of the random indexing function has been used, i.e., a configuration of the seed number $s = 2$ and the dimension number $d = 500$, which was chosen as the optimized settings for the MovieLens dataset according to the test results described in [17]. Additionally, for RRI and for each of the investigated recommendation quality measures, an experimentally optimized number of reflections has been used. It has to be stressed that SVD, RRI, and RSVD-RI methods have been applied in the classical way, i.e., the one that assumes using the standard $user \times item$ matrix (i.e., neither the normalized Laplacian matrix nor the matrix A), which is much smaller matrix than matrix A used for the HPMPP method. As a consequence, the runtime of these methods is considerably smaller than the runtime of the HPMPP method (see Table 5).

In this paper, we investigate a mono-relational propositional data representation method.

Table 2: Results of the evaluation performed using AUROC.

Dataset (lt):		ML100k (0.6)			ML100k (0.8)			ML100k (0.945)			Avg
Set partition (x):		0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8	
Method	k	—	—	—	—	—	—	—	—	—	
RRI	—	0.511	0.498	0.459	0.525	0.523	0.502	0.556	0.560	0.539	0.519
RSVD-RI	2	0.501	0.521	0.498	0.542	0.554	0.544	0.574	0.591	0.573	0.544
	5	0.519	0.540	0.521	0.544	0.571	0.570	0.571	0.611	0.601	0.561
	15	0.519	0.537	0.530	0.534	0.557	0.574	0.550	0.600	0.613	0.557
	50	0.498	0.527	0.532	0.520	0.536	0.567	0.528	0.564	0.595	0.541
SVD	2	0.501	0.520	0.498	0.542	0.553	0.543	0.573	0.594	0.573	0.544
	5	0.518	0.539	0.520	0.543	0.570	0.570	0.571	0.610	0.600	0.560
	15	0.518	0.537	0.529	0.533	0.556	0.574	0.550	0.599	0.613	0.557
	50	0.498	0.526	0.532	0.520	0.535	0.567	0.528	0.564	0.594	0.540
hpmpp-diameter		0.512	0.544	0.521	0.544	0.565	0.568	0.574	0.607	0.601	0.559
hpmpp-mix-time		0.508	0.540	0.512	0.541	0.563	0.571	0.574	0.611	0.601	0.557

Therefore, the input data have been limited to the triples of the *likes* relation (for each of the compared methods, facts have their representations as binary entries in the input matrix).

We have evaluated the methods by means of the most widely used recommendation quality measures: F1 and AUROC [18]. The recommendations have been generated according to the user-item likelihood values obtained by means of (i) the procedure based on the Hadamard product (for the case of HPMPP), (ii) the procedure based on the dot product (for the case of RRI), or (iii) the procedure based on low dimensional approximation of the input $user \times item$ matrix (for the case of SVD and RSVD). The F1 results have been calculated based on the recommendation lists of a given length, whereas the AUROC results have been obtained based on an ordered list of all possible recommendations. Obviously, in order to calculate the AUROC values, for each compared method we have used both the *likes* and *dislikes* propositions of the corresponding testing set.

4.3 Results

The evaluation results of the algorithms have been collected in Tables 2, 3, and 4. The cells of the tables have been shaded according to their values. The shades indicate a few of the highest values in each column. The highest values in the column have been marked by the dark gray color, while the rest have been made lighter, proportionally to their values.

As far as the AUROC results are concerned (see Table 2), the highest overall performance has been achieved by RSVD-RI and SVD (at $k = 5$). However, in order to outperform the fully automatic system based on HPMPP, RSVD-RI and SVD algorithms have to be carefully configured, what requires the execution of many experiments. Therefore, the applicability of these algorithms to a real-world system is severely limited.

Tables 3 and 4 present the results of recommendation scenarios in which, respectively two and twenty items are included in a single recommendation list [12]. In the first of these scenarios, the standard methods have been proven to be less effective in experiments involving the use of realistically sparse datasets. In scenarios in which 20 items for each user are recommended (see Table 4), on average HPMPP outperforms the other algorithms. On the other hand, in the case of using dense datasets, the RSVD-RI method achieves the highest quality. The overall results from both the tables show the superiority of the standard methods

Table 3: Results of the evaluation performed using F1@2.

Dataset (lt):		ML100k (0.6)			ML100k (0.8)			ML100k (0.945)			Sum
Set partition (x):		0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8	
Method	k	—	—	—	—	—	—	—	—	—	
RRI	—	0.005	0.004	0.008	0.010	0.011	0.008	0.016	0.017	0.013	0.093
RSVD-RI	2	0.009	0.016	0.016	0.011	0.013	0.011	0.015	0.024	0.020	0.135
	5	0.008	0.014	0.020	0.011	0.023	0.028	0.015	0.027	0.055	0.221
	15	0.013	0.017	0.029	0.013	0.029	0.047	0.016	0.055	0.064	0.274
	50	0.011	0.017	0.029	0.014	0.035	0.050	0.017	0.051	0.065	0.289
SVD	2	0.008	0.016	0.016	0.011	0.013	0.011	0.015	0.024	0.020	0.134
	5	0.008	0.014	0.020	0.011	0.023	0.028	0.015	0.027	0.055	0.201
	15	0.013	0.017	0.029	0.013	0.029	0.047	0.016	0.055	0.064	0.283
	50	0.011	0.017	0.029	0.014	0.035	0.034	0.017	0.051	0.065	0.273
hpmpp-diameter	—	0.015	0.020	0.038	0.013	0.027	0.037	0.012	0.040	0.062	0.264
hpmpp-mix-time	—	0.017	0.023	0.042	0.016	0.028	0.037	0.014	0.036	0.062	0.275

Table 4: Results of the evaluation performed using F1@20.

Dataset (lt):		ML100k (0.6)			ML100k (0.8)			ML100k (0.945)			Sum
Set partition (x):		0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8	
Method	k	—	—	—	—	—	—	—	—	—	
RRI	—	0.022	0.017	0.014	0.027	0.033	0.026	0.069	0.062	0.043	0.312
RSVD-RI	2	0.019	0.027	0.029	0.049	0.048	0.034	0.080	0.078	0.053	0.417
	5	0.025	0.037	0.034	0.050	0.071	0.056	0.080	0.128	0.096	0.577
	15	0.032	0.041	0.039	0.054	0.069	0.067	0.073	0.132	0.112	0.639
	50	0.032	0.036	0.034	0.051	0.066	0.061	0.075	0.121	0.098	0.574
SVD	2	0.020	0.034	0.029	0.049	0.048	0.028	0.080	0.078	0.053	0.419
	5	0.025	0.037	0.034	0.050	0.071	0.056	0.080	0.128	0.096	0.577
	15	0.032	0.041	0.039	0.054	0.069	0.067	0.073	0.132	0.112	0.619
	50	0.032	0.036	0.034	0.051	0.066	0.062	0.075	0.116	0.098	0.569
hpmpp-diameter	—	0.033	0.045	0.048	0.046	0.079	0.073	0.072	0.137	0.109	0.642
hpmpp-mix-time	—	0.036	0.050	0.049	0.054	0.080	0.072	0.083	0.131	0.109	0.664

Table 5: Runtime comparison for the case of $lt = 0.8$ and for various training set ratios x (the results are provided in seconds).

Dataset (lt):		ML100k (0.8)		
Set partition (x):		0.2	0.5	0.8
Method	k	—	—	—
RRI	—	20.55	21.65	20.51
RSVD-RI	2	21.17	20.83	20.51
	5	20.41	21.09	21.22
	15	20.63	21.02	21.02
	50	20.86	20.53	21.03
SVD	2	5.21	5.54	5.79
	5	5.23	5.54	5.82
	15	5.12	5.49	5.80
	50	5.34	5.54	5.84
HPMPP - diameter		261.84	467.97	1294.502
HPMPP - mixing time		146.67	912, 97	1139.89

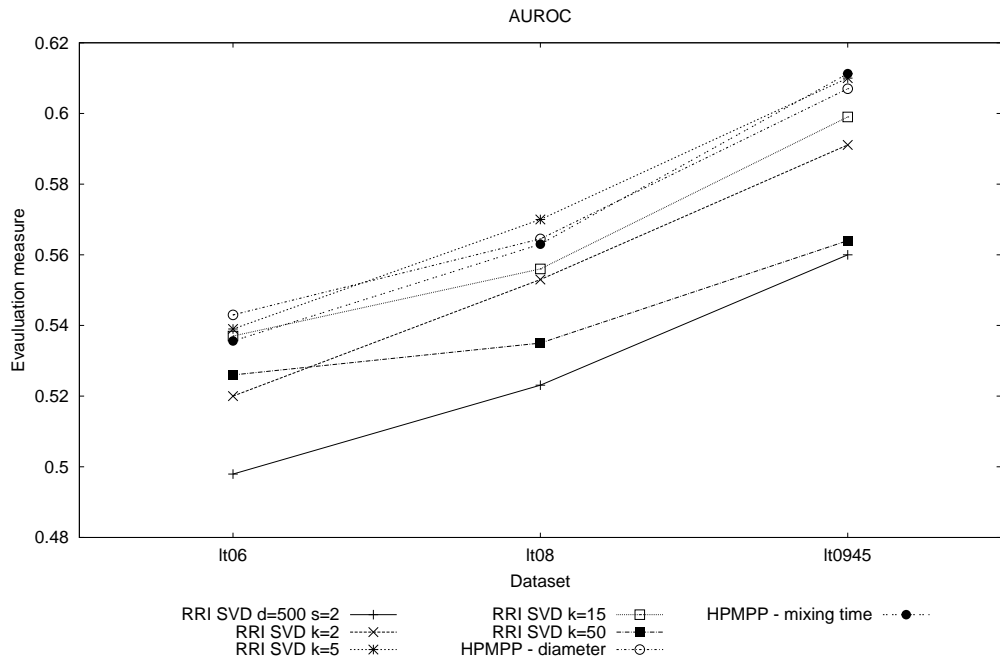


Figure 2: The AUROC results for the $ML(0.5, lt)$ dataset.

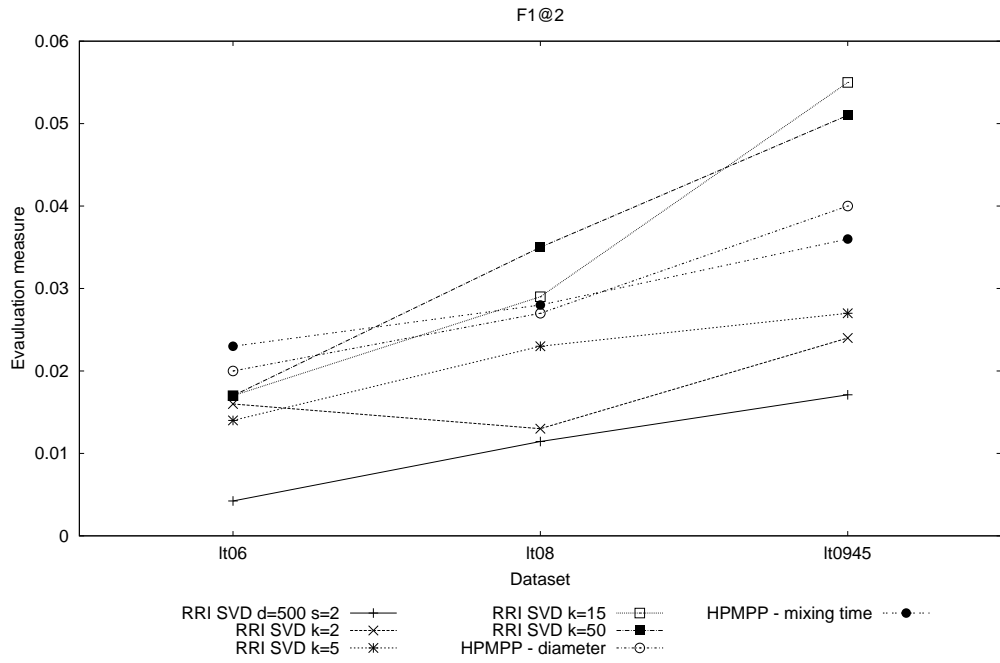


Figure 3: The F1@2 results for the $ML(0.5, lt)$ dataset.

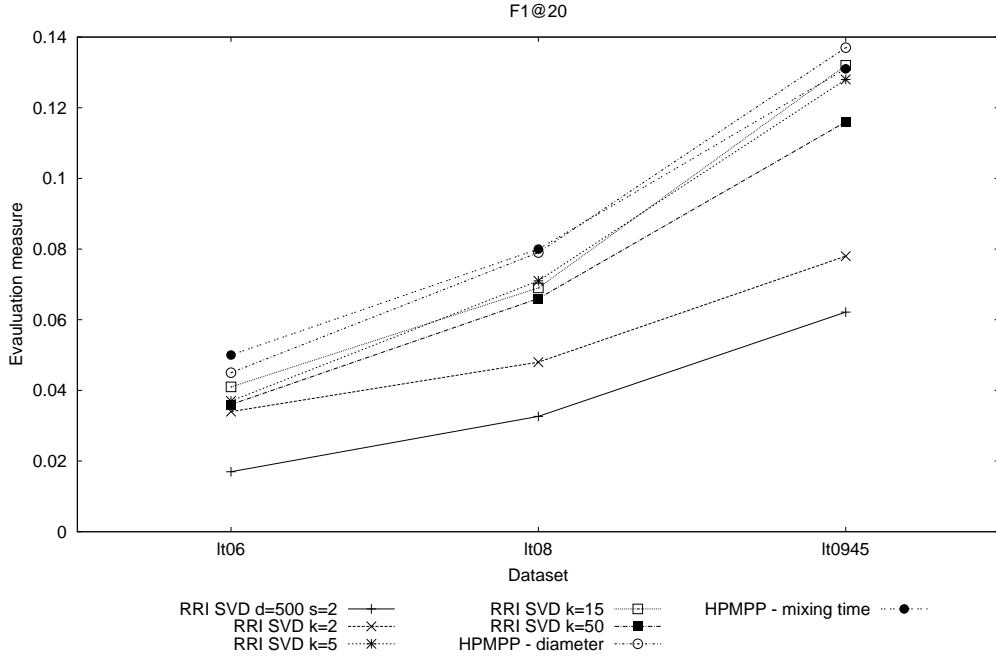


Figure 4: The F1@20 results for the $ML(0.5, lt)$ dataset.

in the right parts of the tables. Therefore, one may conclude that the standard algorithms can be accurate, but only when an unrealistically high number of ratings is available. On the other hand, the proposed method provides a relatively high quality of recommendation in all the cases and scenarios.

It may be observed that the quality of the standard methods depends on the particular measure used. For example, RSVD-RI achieves the highest AUROC-measure quality at $k = 15$, but achieves the highest F1 value at $k = 50$. Thanks to the self-configuration capability, the HPMPP method allows us to achieve a high recommendation quality in all these cases. Figures 2-4 provide the additional visual qualitative results' analysis of experiments executed for datasets of various heavy-tailness and x ratio equal to 0.5.

4.3.1 Computational Complexity

All the presented methods have been evaluated using server of 2 CPUs 6 cores with the clock of 2.67GHz and 48GB of RAM each.

It has to be admitted that the HPMPP method is more time consuming than the other techniques under the comparison. The most time consuming operation is the full eigenvalue decomposition of matrix A which is a square matrix of the size equal to the summarized number of elements and facts $n + m$. It has to be stressed, that A is a much bigger matrix than the user-item matrices being decomposed in the case of using the SVD method. Thus, the

computational complexity of the HPMPP method depends mostly on the complexity of eigendecomposition of matrix A , which is in the worst case equal to $O((n+m)^3)$. Despite the fact that we use the optimized matrix eigendecomposition algorithm from Intel Math Kernel Library (Intel MKL), complexity of this operation is the main reason why the proposed method requires much more time than other compared methods. However, the HPMPP method does not require execution of iterative matrix multiplications described in section 3.3 – the operator $\overline{A}^{(k)}$, which is an arithmetic average of operators $A^{(t)}$ is reconstructed using the eigenvectors of A and eigenvalues calculated once (according to the formula (1)). The operators A and $A^{(t)}$ of size $(n+m) \times (n+m)$ are also the biggest dense matrices that have to be stored during the algorithm operation indicating the method demands in terms of memory.

Table 5 presents the runtime comparison for the case of $ML(x, 0.8)$ dataset for $x \in \{0.2, 0.5, 0.8\}$. It may appear as surprising that the SVD method is faster than the RRI method. Such a result is caused by the fact that in the case of the RRI application (similarly as in the case of the RSVD-RI method) the computation time involves the dataset preparation time (including the random vector generation phase).

4.4 Information-Theoretic Analysis

Each entity representation is a vector of probabilities that may be expressed as a conditional probability distribution, which allows us to calculate the Shannon entropy of each entity representation.

In order to measure the overall representations similarity in information-theoretic terms, i.e., in order to calculate the von Neumann entropy of such a set, the vector-space representations have to be represented collectively as a density matrix. It may be noted that each of the compared well-known methods for the generation of vector-space representations, be it RRI or SVD-based dimensionality reduction, may be used to produce a set of vector-space representations that are arbitrarily similar to one another, i.e., a set of the von Neumann entropy equal to zero. It should be pointed out that, from the a formal point of view, the calculation of the von Neumann entropy based on the results of SVD (matrices which are not operators) requires some additional modeling assumptions, which are out of the scope of this paper.

Figure 5 shows plots of several types of entropies that have been measured in two different domains - one being the number of matrix multiplications (for RRI and HPMPP) and the other being the number of dimensions used to represent data after decomposition (for SVD). As shown in Figure 5, the iterative procedure of the HPMPP method results in a gradual reduction of the von Neumann entropy of the whole set of representations. The entropy reduction, which in the case of HPMPP and RRI is a result of the iterative matrix multiplication (i.e. ‘reflections’), in the case of using SVD may be achieved by a gradual dimensionality reduction.

As may be deduced from the definitions of the SVD and HPMPP methods, either an extreme SVD-based dimensionality reduction or the application of many HPMPP procedure steps leads to an extreme decrease of the von Neumann entropy. Moreover, as RRI is also based on the multiple use of the multiplication of the input matrix by itself, it is natural that RRI is another method that behaves in the above-described way.

A question that naturally arises is as follows: is the same correspondence among the

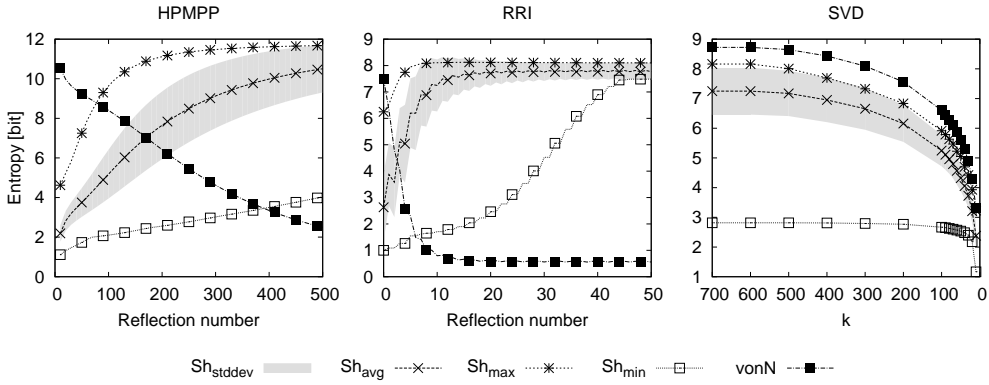


Figure 5: Entropies of whole representations sets (each marked as $vonN$) and entropies of individual representations (each marked as Sh_{min} , Sh_{avg} , Sh_{stddev} or Sh_{max}) shown as functions of the parameters of the analyzed methods, used to control the holistically measured entropy. The greyed area Sh_{stddev} visualizes standard deviations of the individual representations entropies. The three plots represent the results obtained for HPMPP, RRI, and SVD (visualized from left to right, respectively).

analyzed methods observable for less extreme configurations? As may be seen in Figure 5, the answer is affirmative. However, it should be noted that there are some differences between the impact that the number of dimensions has on the analyzed entropies and the analogical impact of the number of steps on the entropies. What is especially evident is that HPMPP reduces the von Neumann entropy in a much more asymptotic way than either RRI or SVD. This is a direct consequence of the basic property of the Neumann series: each component of the series ‘fine-tunes’ the sum of all the previous components by introducing a weaker modification than the one introduced at the previous step. From a functional perspective, this is a key property of the HPMPP-based system as it allows us to make it relatively robust to the suboptimal operation of the procedure termination mechanism.

It is worth stressing that, among the analyzed methods, HPMPP is the only one that provides recommendation results based on representations for which the holistic von Neumann entropy is comparatively high and significantly higher than the Shannon entropies (for $ML(0.5, 0.6)$ the optimal number of reflections is equal to 22 and 26, for *HPMPP-diameter* and *HPMPP-mixing time*, respectively). On the other hand, the optimal results for RRI (for the reflection number equal to 3) and for SVD (for $k = 15$) are obtained when the process of holistic entropy reduction is much more difficult to control (i.e., for ranges of parameter values for which the entropy changes very rapidly).

The basic explanation of the key information-theoretic properties of HPMPP may be given by showing that HPMPP applies the MaxEnt rule to the whole set of representations, rather than to individual vectors. Obviously, the realization of such an objective has to be constrained by the necessary increase of the Shannon entropies of the representations. Moreover, as shown in Figure 5, in the case of using SVD and, to some extent, also in the case of using RRI, the average value and the variance of entropies of the individual representations closely depend on the overall entropy (approximating the von Neumann entropy with the use

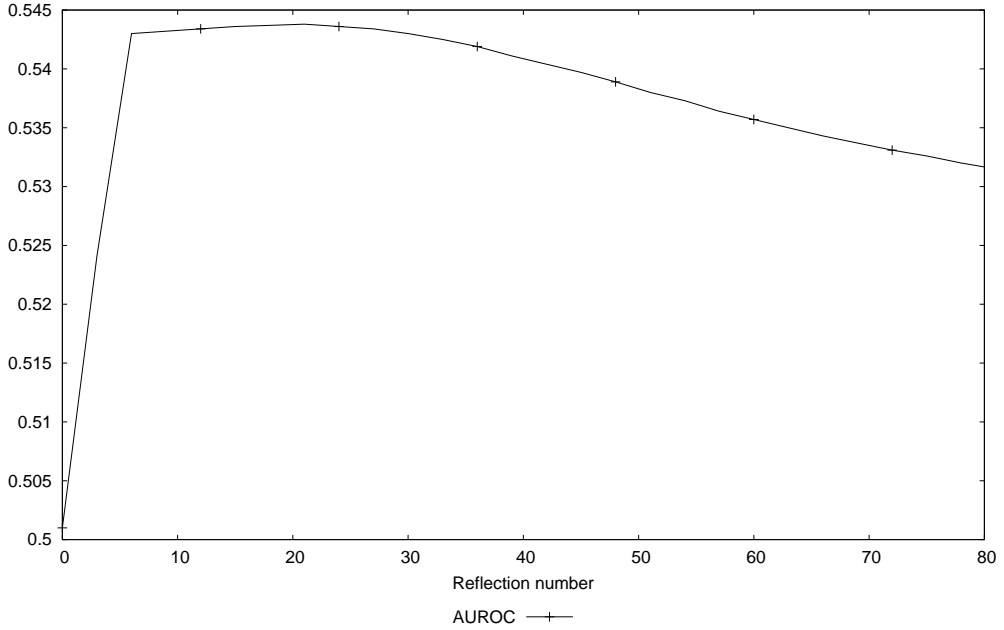


Figure 6: The changes in the AUROC results across the HPMPP iterations.

of XX^T operator, where X is a matrix containing individual representations). Therefore, despite differences between structures of coincidences that ‘link’ entities with other entities (i.e., ‘different views’ on the input data graph that different nodes have), vectors produced as a result of a dimensionality reduction (appropriately reducing the overall entropy) are severely homogenized. In other words, as far as the vector-space graph nodes representations are concerned, the application of SVD imposes entropy constraints that are avoidable when the HPMPP method is used.

Additionally, we have observed that the performance of the HPMPP method on the CF task changes smoothly across computing iterations. Figures 6-7 present the changes of performance measures for the test on the $ML(0.5, 0.6)$ dataset, in which the optimal number of iterations is equal to 22 and 39 for the cases of using the graph diameter and the graph mixing time as the ‘optimality indicator’, respectively. It can be concluded that, in contrast to methods based on SVD or RRI, the HPMPP method shows relatively strong performance for sparse datasets, since it enables to explore indirect relationships in data (i.e., to explore paths of various length in graph G) without the rapid decline in von Neumann entropy and, in consequence, without rapid changes of performance measures.

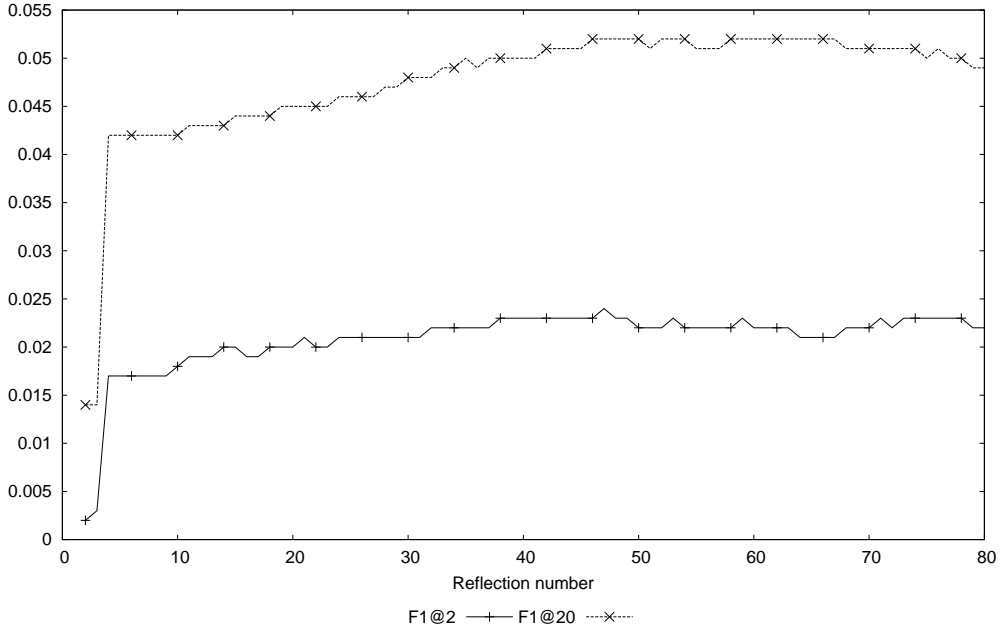


Figure 7: The changes in the F1 results across the HPMPP iterations.

5 Conclusions

The matrix data processing that is one of the core elements of HPMPP may appear similar to the well-known SVD-based dimensionality reduction. On the other hand, from an algorithmic point of view, the proposed method for the generation of vector-space representations may resemble popular reflective matrix data processing methods, such as RRI [4]. However, it should be stressed that the proposed method, thanks to being based on the rigorous probabilistic data processing model, avoids the application of the above-mentioned data processing heuristics, typically dominating the collaborative filtering solutions presented in the literature.

As we have demonstrated in the experiments, our model of the vector-space representations transformation constitutes a valuable basis for the iterative procedure of producing useful vector-space representations of subjects, predicates, and objects. We have shown that the iterative ‘gradual unification’ of HPMPP representations enables the generation of representations that may be used to effectively estimate the likelihood of unknown facts. Moreover, we have shown that the quality provided by a particular HPMPP implementation does not strongly depend on the number of the executed HPMPP procedure steps. In contrast to such a behavior of HPMPP systems, algorithms based on RRI or SVD dimensionality reduction are highly susceptible to a suboptimal configuration, such as too low or too high a number of dimensions used to represent the users and items (in the case of applying SVD) or too low or too high a number of reflections (in the case of applying RRI). Naturally, a low susceptibil-

ity to a suboptimal configuration makes the effective self-configuration of the HPMPP-based recommendation system much easier to achieve.

From an information-theoretic perspective, the key feature of the proposed dual, graph-vector propositional data representation and optimization scheme is the ability to perform the entropy-reducing transformation of mutually compatible vector-space representations in a way that leads to high recommendation quality. In a clear contrast to methods proposed so far, this property allows the HPMPP-based system to effectively optimize the number of the executed HPMPP procedure steps and, as a result, to operate in a fully automatic way.

Acknowledgments.

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

References

- [1] Adomavicius G., Tuzhilin A., Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Trans. on Knowl. and Data Eng.*, **17**, 2005, pp. 734–749.
- [2] Bruza P., Widdows D., Woods J.W., A quantum logic of down below, in: K. Engesser, D. Gabbay, and D. Lehmann (eds.), *Handbook of Quantum Logic, Quantum Structure and Quantum Computation*, Elsevier, 2009, pp. 625–660.
- [3] Ciesielczyk M., Szwabe A., RI-based dimensionality reduction for recommender systems, in: *Proc. of 3rd International Conference on Machine Learning and Computing*, IEEE Press, Singapore, 2011.
- [4] 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, pp. 240–256.
- [5] 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.
- [6] de Vries A. P., Roelleke T., Relevance information: A loss of entropy but a gain for idf?, in: *Proc. of SIGIR*, 2005, pp. 282–289.
- [7] 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.
- [8] Levin D. A., Peres Y., Wilmer E. L., *Markov Chains and Mixing Times*, Amer. Math. Soc. Press, Providence, Rhode Island, 2008.
- [9] 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.
- [10] Nielsen M. A., Chuang I.L., *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2010.

- [11] Pitowsky I., Quantum probability, quantum logic, in: *Lecture Notes in Physics*, **321**, Heidelberg, Springer, 1989.
- [12] Ricci F., Rokach L., Shapira B., Kantor P.B., *Recommender Systems Handbook*, Springer, 2011.
- [13] 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.
- [14] Struyf J., Blockeel H., Relational Learning, in: C. Sammut, G. Webb (eds.), *Encyclopedia of Machine Learning*, Springer, 2010, pp. 851–857.
- [15] Sutskever I., Salakhutdinov R., Tenenbaum J. B., Modelling relational data using Bayesian clustered tensor factorization, *Advances in Neural Information Processing Systems*, **22**, 2009.
- [16] 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, pp. 142–151.
- [17] Szwabe A., Misiorek P., Ciesielczyk M., Jedrzejek C., Collaborative filtering based on bi-relational data representation, *Foundations of Computing and Decision Sciences*, **38**, 4, 2013, pp. 67–83.
- [18] van Rijsbergen C. J., *Information Retrieval (2nd ed.)*, Butterworth-Heinemann, USA, 1979.
- [19] van Rijsbergen C. J., *The Geometry of Information Retrieval*, Cambridge University Press, New York, USA, 2004, pp. 73–101.
- [20] Wilson R.C., Hancock E.R., Luo B., Pattern vectors from algebraic graph theory, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 7, 2005, pp. 1112–1124.
- [21] Wermser H., Rettinger A., Tresp V., Modeling and learning context-aware recommendation scenarios using tensor decomposition, in: *Proceedings of 2011 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2011, pp. 137–144.
- [22] Xu Z., Kersting K., Tresp V., Multi-relational learning with Gaussian processes, in: *Proc. of the 21st International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, 2009, pp. 1309–1314.
- [23] Yahyaei S., Monz Ch., Applying maximum entropy to known-item email retrieval, in: C. MacDonald, I. Ounis, V. Plachouras, I. Ruthven, R.W. White (eds.), *Proc. of European Conference on Information Retrieval (ECIR'08)*, Springer-Verlag, Berlin Heidelberg, 2008, pp. 406–413.
- [24] Zitnick C.L., Kanade T., Maximum entropy for collaborative filtering. In: *Proc. of the 20th Conf. on Uncertainty in Artificial Intelligence*, AUAI Press, Arlington, VA, USA, 2004, pp. 636–643.
- [25] <http://www.grouplens.org>

Received November, 2012