Reducing the Rank of Relational Factorization Models by Including Observable Patterns

Maximilian Nickel[1,2] **Xueyan Jiang**[3,4] **Volker Tresp**[3,4]
[1]LCSL, Poggio Lab, Massachusetts Institute of Technology, Cambridge, MA, USA
[2]Istituto Italiano di Tecnologia, Genova, Italy
[3]Ludwig Maximilian University, Munich, Germany
[4]Siemens AG, Corporate Technology, Munich, Germany
mnick@mit.edu, {xueyan.jiang.ext,volker.tresp}@siemens.com

Abstract

Tensor factorization has become a popular method for learning from multi-relational data. In this context, the rank of the factorization is an important parameter that determines runtime as well as generalization ability. To identify conditions under which factorization is an efficient approach for learning from relational data, we derive upper and lower bounds on the rank required to recover adjacency tensors. Based on our findings, we propose a novel additive tensor factorization model to learn from latent and observable patterns on multi-relational data and present a scalable algorithm for computing the factorization. We show experimentally both that the proposed additive model does improve the predictive performance over pure latent variable methods and that it also reduces the required rank — and therefore runtime and memory complexity — significantly.

1 Introduction

Relational and graph-structured data has become ubiquitous in many fields of application such as social network analysis, bioinformatics, and artificial intelligence. Moreover, relational data is generated in unprecedented amounts in projects like the Semantic Web, YAGO [26], NELL [4], and Google's Knowledge Graph [5] such that learning from relational data, and in particular learning from large-scale relational data, has become an important subfield of machine learning. Existing approaches to relational learning can approximately be divided into two groups: First, methods that explain relationships via observable variables, i.e. via the observed relationships and attributes of entities, and second, methods that explain relationships via a set of latent variables. The objective of latent variable models is to infer the states of these hidden variables which, once known, permit the prediction of unknown relationships. Methods for learning from observable variables cover a wide range of approaches, e.g. inductive logic programming methods such as FOIL [22], statistical relational learning methods such as Probabilistic Relational Models [6] and Markov Logic Networks [23], and link prediction heuristics based on the Jaccard's Coefficient and the Katz Centrality [16]. Important examples of latent variable models for relational data include the IHRM and the IRM [28, 10], the Mixed Membership Stochastic Blockmodel [1] and low-rank matrix factorizations [16, 25, 7]. More recently, tensor factorization, a generalization of matrix factorization to higher-order data, has shown state-of-the-art results for relationship prediction on *multi-relational* data [20, 8, 2, 13]. The number of latent variables in tensor factorization is determined via the number of latent components used in the factorization, which in turn is bounded by the factorization rank. While tensor and matrix factorization algorithms scale typically well with the size of the data — which is one reason for their appeal — they often do not scale well with respect to the rank of the factorization. For instance, RESCAL is a state-of-the art relational learning method based on tensor factorization which can be applied to large knowledge bases consisting of millions of entities and billions of known facts [21].

However, while the runtime of the most scalable known algorithm to compute RESCAL scales linearly with the number of entities, linearly with the number of relations, and linearly with the number of known facts, it scales *cubical* with regard to the rank of the factorization [21].[1] Moreover, the memory requirements of tensor factorizations like RESCAL become quickly infeasible on large data sets if the factorization rank is large and no additional sparsity of the factors is enforced. Hence, tensor (and matrix) rank is a central parameter of factorization methods that determines generalization ability as well as scalability. In this paper we study therefore how the rank of factorization methods can be reduced while maintaining their predictive performance and scalability. We first analyze under which conditions tensor and matrix factorization requires high or low rank on relational data. Based on our findings, we then propose an additive tensor decomposition approach to reduce the required rank of the factorization by combining latent and observable variable approaches.

This paper is organized as follows: In section 2 we develop the main theoretical results of this paper, where we show that the rank of an adjacency tensor is lower bounded by the maximum number of strongly connected components of a single relation and upper bounded by the sum of diclique partition numbers of all relations. Based on our theoretical results, we propose in section 3 a novel tensor decomposition approach for multi-relational data and present a scalable algorithm to compute the decomposition. In section 4 we evaluate our model on various multi-relational datasets.

Preliminaries We will model relational data as a directed graph (digraph), i.e. as an ordered pair $\Gamma = (\mathcal{V}, \mathcal{E})$ of a nonempty set of vertices $\mathcal{V}$ and a set of directed edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. An existing edge between node v_i and v_j will be denoted by $v_i \rightsquigarrow v_j$. By a slight abuse of notation, $\Gamma(Y)$ will indicate the digraph Γ associated with an adjacency matrix $Y \in \{0,1\}^{N \times N}$. Next, we will briefly review further concepts of tensor and graph theory that are important for the course of this paper.

Definition 1. *A strongly connected component of a digraph Γ is a maximal subgraph Ψ for which every vertex is reachable from any other vertex in Ψ by following the directional edges in the subgraph. A strongly connected component is* trivial *if it consists only of a single element, i.e. if it is of the form $\Psi = (\{v_i\}, \varnothing)$, and* nontrivial *otherwise.*

We will denote the number of strongly connected components in a digraph Γ by $\mathrm{scc}(\Gamma)$. The number of nontrivially connected components will be denoted by $\mathrm{scc}_+(\Gamma)$.

Definition 2. *A digraph $\Gamma = (\mathcal{V}, \mathcal{E})$ is a diclique if it is an orientation of a complete undirected bipartite graph with bipartition $(\mathcal{V}_1, \mathcal{V}_2)$ such that $v_1 \in \mathcal{V}_1$ and $v_2 \in \mathcal{V}_2$ for every edge $v_1 \rightsquigarrow v_2 \in \mathcal{E}$.*

Figure 3 in supplementary material A shows an example of a diclique. Please note that dicliques consist only of trivially strongly connected components, as there cannot exist any cycles in a diclique. Given the concept of a diclique, the diclique partitioning number of a digraph is defined as:

Definition 3. *The diclique partition number* $\mathrm{dp}(\Gamma)$ *of a digraph $\Gamma = (\mathcal{V}, \mathcal{E})$ is the minimum number of dicliques such that each edge $e \in \mathcal{E}$ is contained in* exactly one *diclique.*

Tensors can be regarded as higher-order generalizations of vectors and matrices. In the following, we will only consider third-order tensors of the form $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$, although many concepts generalize to higher-order tensors. The mode-n unfolding (or matricization) of $\mathbf{X}$ arranges the mode-n fibers of $\mathbf{X}$ as the columns of a newly formed matrix and will be denoted by $X_{(n)}$. The tensor-matrix product $\mathbf{A} = \mathbf{X} \times_n B$ multiplies the tensor $\mathbf{X}$ with the matrix B along the n-th mode of $\mathbf{X}$ such that $A_{(k)} = BX_{(k)}$. For a detailed introduction to tensors and these operations we refer the reader to Kolda et al. [12]. The k-th frontal slice of a third-order tensor $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ will be denoted by $X_k \in \mathbb{R}^{I \times J}$. The outer product of vectors will be denoted by $\boldsymbol{a} \circ \boldsymbol{b}$. In contrast to matrices, there exist two non-equivalent notions of the rank of a tensor:

Definition 4. *Let $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ be a third-order tensor. The tensor rank* $\text{t-rank}(\mathbf{X})$ *of $\mathbf{X}$ is defined as* $\text{t-rank}(\mathbf{X}) = \min\{r \mid \mathbf{X} = \sum_{i=1}^{r} \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i\}$ *where $\boldsymbol{a}_i \in \mathbb{R}^I$, $\boldsymbol{b}_i \in \mathbb{R}^J$, and $\boldsymbol{c}_i \in \mathbb{R}^K$. The multilinear rank* $\text{n-rank}(\mathbf{X})$ *of $\mathbf{X}$ is defined as the tuple (r_1, r_2, r_3), where $r_i = \mathrm{rank}\left(X_{(i)}\right)$.*

To model multi-relational data as tensors, we use the following concept of an adjacency tensor:

Definition 5. *Let $\mathcal{G} = \{(\mathcal{V}, \mathcal{E}_k)\}_{k=1}^{K}$ be a set of digraphs over the same set of vertices $\mathcal{V}$, where $|\mathcal{V}| = N$. The adjacency tensor of $\mathcal{G}$ is a third-order tensor $\mathbf{X} \in \{0,1\}^{N \times N \times K}$ with entries $x_{ijk} = 1$ if $v_i \rightsquigarrow v_j \in \mathcal{E}_k$ and $x_{ijk} = 0$ otherwise.*

[1] Similar results can be obtained for state-of-the-art algorithms to compute the well-known CP and Tucker decompositions. Please see the supplementary material A.3 for the respective derivations.

For a single digraph, an adjacency tensor is equivalent to the digraph's adjacency matrix. Note that K would correspond to the number of relation types in a domain.

2 On the Algebraic Complexity of Graph-Structured Data

In this section, we want to identify conditions under which tensor factorization can be considered efficient for relational learning. Let $\mathbf{X}$ denote an observed adjacency tensor with missing or noisy entries from which we seek to recover the true adjacency tensor $\mathbf{Y}$. Rank affects both the predictive as well as the runtime performance of a factorization: A high factorization rank will lead to poor runtime performance while a low factorization rank might not be sufficient to model $\mathbf{Y}$. We are therefore interested in identifying upper and lower bounds on the minimal rank — either tensor rank or multilinear rank — that is required such that a factorization can model the true adjacency tensor $\mathbf{Y}$. Please note that we are not concerned with bounds on the generalization error or the sample complexity that is needed to *learn* a good model, but on bounds on the algebraic complexity that is needed to *express* the true underlying data via factorizations. For sign-matrices $Y \in \{\pm 1\}^{N \times N}$, this question has been discussed in combinatorics and communication complexity via their *sign-rank* $\operatorname{rank}_{\pm}(Y)$, which is the minimal rank needed to recover the sign-pattern of Y:

$$\operatorname{rank}_{\pm}(Y) = \min_{M \in \mathbb{R}^{N \times N}} \left\{ \operatorname{rank}(M) \;\middle|\; \forall i,j : \operatorname{sgn}(m_{ij}) = y_{ij} \right\}. \tag{1}$$

Although the concept of sign-rank can be extended to adjacency tensors, bounds based on the sign-rank would have only limited significance for our purpose, as no practical algorithms exist to find the solution to equation (1). Instead, we provide upper and lower bounds on tensor and multilinear rank, i.e. bounds on the *exact* recovery of $\mathbf{Y}$, for the following reasons: It follows immediately from (1) that any upper-bound on $\operatorname{rank}(\mathbf{Y})$ will also hold for $\operatorname{rank}_{\pm}(\mathbf{Y})$ since it has to hold that $\operatorname{rank}_{\pm}(\mathbf{Y}) \leqslant \operatorname{rank}(\mathbf{Y})$. Upper bounds on $\operatorname{rank}(\mathbf{Y})$ can therefore provide insight under what conditions factorizations can be efficient on relational data — regardless whether we seek to recover exact values or sign patterns. Lower bounds on $\operatorname{rank}(\mathbf{Y})$ provide insight under what conditions the exact recovery of $\mathbf{Y}$ can be inefficient. Therefore, such bounds provide also insight under which conditions the recovery of the sign patterns in $\mathbf{Y}$ can potentially be inefficient.

Based on these considerations, we state now the main theorem of this paper, which bounds the different notions of the rank of an adjacency tensor by the diclique partition number and the number of strongly connected components of the involved relations:

Theorem 1. *Tensor rank* $\text{t-rank}(\mathbf{Y})$ *and multilinear rank* $\text{n-rank}(\mathbf{Y}) = (r_1, r_2, r_3)$ *of any adjacency tensor* $\mathbf{Y} \in \{0,1\}^{N \times N \times K}$ *representing* K *relations* $\{\Gamma_k(Y_k)\}_{k=1}^{K}$ *are bounded as*

$$\sum_{k=1}^{K} \operatorname{dp}(\Gamma_k) \geqslant \theta \geqslant \max_k \operatorname{scc}_+(\Gamma_k),$$

where θ *is any of the quantities* $\text{t-rank}(\mathbf{Y})$, r_1, *or* r_2.

To prove theorem 1 we will first derive upper and lower bounds on adjacency *matrices* and then show how these bounds generalize to adjacency *tensors*.

Lemma 1. *For any adjacency matrix* $Y \in \{0,1\}^{N \times N}$ *it holds that* $\operatorname{dp}(\Gamma) \geqslant \operatorname{rank}(Y) \geqslant \operatorname{scc}_+(\Gamma)$.

Proof. The upper bound of lemma 1 follows directly from the fact that $\operatorname{dp}(\Gamma(Y)) = \operatorname{rank}_{\mathbb{N}}(Y)$ and the fact that $\operatorname{rank}_{\mathbb{N}}(Y) \geqslant \operatorname{rank}(Y)$, where $\operatorname{rank}_{\mathbb{N}}(Y)$ denotes the *non-negative integer rank* of the binary matrix Y [18, see eq. 1.6.5 and eq. 1.7.1]. □

Next we will prove the lower bound of lemma 1. Let $\lambda_i(Y)$ denote the i-th (complex) *eigenvalue* of Y and let $\Lambda(Y)$ denote the *spectrum* of $Y \in \mathbb{R}^{N \times N}$, i.e. the multiset of (complex) eigenvalues of Y. Furthermore, let $\rho(Y) = \max_i |\lambda_i(Y)|$ be the *spectral radius* of Y. Now, recall the celebrated Perron-Frobenius theorem:

Theorem 2 ([24, Theorem 8.2])**.** *Let* $Y \in \mathbb{R}^{N \times N}$ *with* $y_{ij} \geqslant 0$ *be a non-negative irreducible matrix. Then* $\rho(Y) > 0$ *is a simple eigenvalue of* Y *associated with a positive eigenvector.*

Please note that a nontrivial digraph is strongly connected iff its adjacency matrix is irreducible [3, Theorem 3.2.1]. Furthermore, an adjacency matrix is nilpotent iff the associated digraph is acyclic [3, Section 9.8]. Hence, the adjacency matrix of a strongly connected component Ψ is nilpotent iff Ψ is trivial. Given these considerations, we can now prove the lower bound of lemma 1:

Lemma 2. *For any non-negative adjacency matrix* $Y \in \mathbb{R}^{N \times N}$ *with* $y_{ij} \geqslant 0$ *of a weighted digraph* Γ *it holds that* $\text{rank}(Y) \geqslant \text{scc}_+(\Gamma)$.

Proof. Let Γ consist of k nontrivial strongly connected components. The Frobenius normal form B of its associated adjacency matrix Y consists then of k irreducible matrices B_i on its block diagonal. It follows from theorem 2 that each irreducible B_i has at least one nonzero eigenvalue. Since B is block upper triangular, it holds also that $\Lambda(B) = \bigcup_{i=1}^{k} \Lambda(B_i)$. As the rank of a square matrix is larger or equal to the number of its nonzero eigenvalues, it follows that $\text{rank}(B) \geqslant k$. Lemma 2 follows from the fact that B is similar to Y and that matrix similarity preserves rank. □

So far, we have shown that $\text{rank}(Y)$ of an adjacency *matrix* Y is bounded by the diclique covering number and the number of nontrivial strongly connected components of the associated digraph. To complete the proof of theorem 1 we will now show that these bounds for unirelational data translate directly to multi-relational data and to the different notions of the rank of an adjacency *tensor*. In particular we will show that both notions of tensor rank are lower bounded by the maximum rank of a single frontal slice in the tensor and upper bounded by the sum of the ranks of all frontal slices:

Lemma 3. *The tensor rank* $\text{t-rank}(\mathbf{Y})$ *and multilinear rank* $\text{n-rank}(\mathbf{Y}) = (r_1, r_2, r_3)$ *of any third-order tensor* $\mathbf{Y} \in \mathbb{R}^{I \times J \times K}$ *with frontal slices* Y_k *are bounded as*

$$\sum_{k=1}^{K} \text{rank}(Y_k) \geqslant \theta \geqslant \max_k \text{rank}(Y_k),$$

where θ *is any of the quantities* $\text{t-rank}(\mathbf{Y})$, r_1, *or* r_2.

Proof. Due to space constraints, we will include only the proof for tensor rank. The proof for multilinear rank can be found in supplementary material A.1. Let $\text{t-rank}(\mathbf{Y}) = r$ and $\text{rank}(Y_k) = r_{\max}$. It can be seen from the definition of tensor rank that $Y_k = \sum_{i=1}^{r} c_{kr} (\boldsymbol{a}_r \boldsymbol{b}_r^\top)$. Consequently, it follows from the subadditivity of matrix rank, i.e. $\text{rank}(A + B) \leqslant \text{rank}(A) + \text{rank}(B)$, that

$$r_{\max} = \text{rank}\left(\sum_{i=1}^{r} c_{kr} \boldsymbol{a}_r \boldsymbol{b}_r^\top\right) \leqslant \sum_{i=1}^{r} \text{rank}\left(c_{kr} \boldsymbol{a}_r \boldsymbol{b}_r^\top\right) \leqslant r$$

where the last inequality follows from $\text{rank}\left(c_{kr} \boldsymbol{a}_r \boldsymbol{b}_r^\top\right) \leqslant 1$. Now we will derive the upper bound of lemma 3 by providing a decomposition of $\mathbf{Y}$ with rank $r = \sum_k \text{rank}(Y_k)$ that recovers $\mathbf{Y}$ exactly. Let $Y_k = U_k S_k V_k^\top$ be the SVD of Y_k with $S_k = \text{diag}(\boldsymbol{s}_k)$. Furthermore, let $U = [U_1\ U_2\ \cdots\ U_K]$, $V = [V_1\ V_2\ \cdots\ V_K]$, and let S be a block-diagonal matrix where the i-th block on the diagonal is equal to $\boldsymbol{s}_i^\top$ and all other entries are 0. It can be easily verified that $\sum_{i=1}^{r} \hat{\boldsymbol{u}}_i \circ \hat{\boldsymbol{v}}_i \circ \hat{\boldsymbol{s}}_i$ provides an exact decomposition of $\mathbf{Y}$, where $r = \sum_k \text{rank}(Y_k)$ and $\hat{\boldsymbol{u}}_i$, $\hat{\boldsymbol{v}}_i$, and $\hat{\boldsymbol{s}}_i$ are the i-th columns of the matrices U, V, and S. The inequality in lemma 3 follows since r is not necessarily minimal. □

Theorem 1 can now be derived by combining lemmas 1 and 3 what concludes the proof.

Discussion It can be seen from theorem 1 that factorizations can be computationally efficient when $\sum_k \text{dp}(\Gamma_k)$ is small. However, factorizations can potentially be inefficient when $\text{scc}_+(\Gamma_k)$ is large for *any* Γ_k in the data. For instance, consider an idealized *marriedTo* relation, where each person is married to exactly one person. Evidently, for m marriages, the associated digraph would consist of m strongly connected components, i.e. one component for each marriage. According to lemma 2, a factorization model would at least require m latent components to recover this adjacency matrix exactly. Consequently, an algorithm with cubic runtime complexity in the rank would only be able to recover $\mathbf{Y}$ for this relation when the number of marriages is small, what limits its applicability to these relations. A second important observation for multi-relational learning is that the lower bound in theorem 1 depends only on the largest rank of a single frontal slice (i.e. a single adjacency matrix) in $\mathbf{Y}$. For multi-relational learning this means that regularities between different relations can not decrease tensor or multilinear rank below the largest matrix rank of a single relation. For instance, consider an $N \times N \times 2$ tensor $\mathbf{Y}$ where $Y_1 = Y_2$. Clearly it holds that $\text{rank}(Y_{(3)}) = 1$, such that Y_1 could easily be predicted from Y_2 when Y_2 is known. However, theorem 1 states that the rank of the factorization must be at least $\text{rank}(Y_1)$ — which can be arbitrarily large up to N — when the first two modes of $\mathbf{Y}$ are also factorized. Please note that this is not a statement about sample complexity or generalization error which *can be* reduced when factorizing all modes of a tensor, but a statement about the minimal rank that is required to *express* the data. A last observation from the previous discussion is that factorizations and observable variable methods excel at different aspects of relationship prediction. For instance, predicting relationships in the idealized *marriedTo* relation

can be done easily with Horn clauses and link predication heuristics as listed in supplementary material A.2. In contrast, factorization methods would be inefficient in predicting links in this relation as they would require at least one latent component for each marriage. At the same time, links in a diclique of any size can trivially be modeled with a rank-2 factorization that indicates the partition memberships, while standard neighborhood-based methods will fail on dicliques since — by the definition of a diclique — there do not exist links within one partition yet the only vertices that share neighbors are located in the same partition.

3 An Additive Relational Effects Model

RESCAL is a state-of-the-art relational learning method that is based on a constrained Tucker-decomposition and as such is subject to bounds as in theorem 1. Motivated by the results of section 2, we propose an additive tensor decomposition approach to combine the strengths of latent and observable variable methods to reduce the rank requirements of RESCAL on multi-relational data. To include the information of observable pattern methods in the factorization, we augment the RESCAL model with an additive term that holds the predictions of observable pattern methods. In particular, let $\mathbf{X} \in \{0,1\}^{N \times N \times K}$ be a third-order adjacency tensor and $\mathbf{M} \in \mathbb{R}^{N \times N \times P}$ be a third-order tensor that holds the predictions of an arbitrary number of relational learning methods. The proposed *additive relational effects* model (ARE) decomposes $\mathbf{X}$ into

$$\mathbf{X} \approx \mathbf{R} \times_1 A \times_2 A + \mathbf{M} \times_3 W, \tag{2}$$

where $A \in \mathbb{R}^{N \times r}$, $\mathbf{R} \in \mathbb{R}^{r \times r \times K}$ and $W \in \mathbb{R}^{K \times P}$. The first term of equation (2) corresponds to the RESCAL model which can be interpreted as following: The matrix A holds the latent variable representations of the entities, while each frontal slice R_k of $\mathbf{R}$ is an asymmetric $r \times r$ matrix that models the interactions of the latent components for the k-th relation. The variable r denotes the number of latent components of the factorization. An important aspect of RESCAL for relational learning is that entities have a unique latent representation via the matrix A. This enables a relational learning effect via the propagation of information over different relations and the occurrences of entities as a subject or objects in relationships. For a detailed description of RESCAL we refer the reader to Nickel et al. [20, 21]. After computing the factorization (2), the score for the existence of a single relationship is calculated in ARE via $\widehat{x}_{ijk} = \boldsymbol{a}_i^T R_k \boldsymbol{a}_j + \sum_{p=1}^{P} w_{kp} m_{ijp}$.

The construction of the tensor $\mathbf{M}$ is of the following: Let $\mathcal{F} = \{f_p\}_{p=1}^{P}$ be a set of given real-valued functions $f_p : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ which assign scores to each pair of entities in $\mathcal{V}$. Examples of such score functions include link prediction heuristics such as Common Neighbors, Katz Centrality, or Horn clauses. Depending on the underlying model these scores can be interpreted as confidences value or as probabilities that a relationship exists between two entities. We collect these real-valued predictions of P score functions in the tensor $\mathbf{M} \in \mathbb{R}^{N \times N \times P}$ by setting $m_{ijp} = f_p(v_i, v_j)$. Supplementary material A.2 provides a detailed description of the construction of $\mathbf{M}$ for typical score functions. The tensor $\mathbf{M}$ acts in the factorization as an independent source of information that predicts the existence of relationships. The term $\mathbf{M} \times_3 W$ can be interpreted as learning a set of weights w_{kp} which indicate how much the p-th score function in $\mathbf{M}$ correlates with the k-th relation in $\mathbf{X}$. For this reason we refer to $\mathbf{M}$ also as the *oracle tensor*. If $\mathbf{M}$ is composed of relation path features as proposed by Lao et al. [15], the term $\mathbf{M}W$ is closely related to the Path Ranking Algorithm (PRA) [15].

The main idea of equation (2) is the following: The term $\mathbf{R} \times_1 A \times_2 A$ is equivalent to the RESCAL model and provides an efficient approach to learn from latent patterns on relational data. The oracle tensor $\mathbf{M}$ on the other hand is not factorized, such that it can hold information that is difficult to predict via latent variable methods. As it is not clear a priori which score functions are good predictors for which relations, the term $\mathbf{M} \times_3 W$ learns a weighting of how predictive any score function is for any relation. By integrating both terms in an additive model, the term $\mathbf{M} \times_3 W$ can potentially reduce the required rank for the RESCAL term by explaining links that, for instance, reduce the diclique partition number of a digraph. Rules and operations that are likely to reduce the diclique partition number of slices in $\mathbf{X}$ are therefore good candidates to be included in $\mathbf{M}$. For instance, by including a copy of the observed adjacency tensor $\mathbf{X}$ in $\mathbf{M}$ (or some selected frontal slices X_k), the term $\mathbf{M} \times_3 W$ can easily model common multi-relational patterns where the existence of a relationship in one relation correlates with the existence of a relationship between the same entities in another relation via $x_{ijk} = \sum_{p \neq k} w_{kp} x_{ijp}$. Since w_{kp} is allowed to be negative, anti-correlations can be modeled

efficiently. ARE is similar in spirit to the model of Koren [14], which extends SVD with additive terms to include local neighborhood information in an uni-relational recommendation setting and Jiang et al. [9] which uses an additive matrix factorization model for link prediction. Furthermore, the recently proposed Google Knowledge Vault (KV) [5] considers a combination of PRA and a neural network model related to RESCAL for learning from large multi-relational datasets. However, in KV both models are trained separately and combined only later in a separate fusion step, whereas ARE learns both models jointly what leads to the desired rank-reduction effect.

To compute ARE, we pursue a similar optimization scheme as used for RESCAL which has been shown to scale to large datasets [21]. In particular, we solve the regularized optimization problem

$$\min_{A,\mathbf{R},W} \|\mathbf{X} - (\mathbf{R} \times_1 A \times_2 A + \mathbf{M} \times_3 W)\|_F^2 + \lambda_A \|A\|_F^2 + \lambda_R \|\mathbf{R}\|_F^2 + \lambda_W \|W\|_F^2. \tag{3}$$

via alternating least-squares, which is a block-coordinate optimization method in which blocks of variables are updated alternatingly until convergence. For equation (3) the variable blocks are given naturally by the factors A, $\mathbf{R}$, and W.

Updates for W Let $\mathbf{E} = (\mathbf{X} - \mathbf{R} \times_1 A \times_2 A)$ and I be the identity matrix. We rewrite equation (2) as $E_{(3)} \approx W M_{(3)}$ such that equation (3) becomes a regularized least-squares problem when solving for W. It follows that updates for W can be computed via $W \leftarrow (M_{(3)} M_{(3)}^\top + \lambda_W I)^{-1} M_{(3)} E_{(3)}^\top$. However, performing the updates in this way would be very inefficient as it involves the computation of the *dense* $N \times N \times K$ tensor $\mathbf{R} \times_1 A \times_2 A$. This would quickly lead to scalability issues with regard to runtime and memory requirements. To overcome this issue, we rewrite $M_{(3)} E_{(3)}^\top$ using the equality $(\mathbf{R} \times_1 A \times_2 A)_{(3)} M_{(3)}^\top = R_{(3)} (\mathbf{M} \times_1 A^\top \times_2 A^\top)_{(3)}^\top$. Updates for W can then be computed efficiently as

$$W^\top \leftarrow \left[X_{(3)} M_{(3)}^\top - R_{(3)} (\mathbf{M} \times_1 A^\top \times_2 A^\top)_{(3)}^\top \right] (M_{(3)} M_{(3)}^\top + \lambda_W I)^{-1}. \tag{4}$$

In equation (4) the dense tensor $\mathbf{R} \times_1 A \times_2 A$ is never computed explicitly and the computational complexity with regard to the parameters N, K, and r is reduced from $O(N^2 K r)$ to $O(NKr^3)$. Furthermore, all terms in equation (4) except $R_{(3)} (\mathbf{M} \times_1 A^\top \times_2 A^\top)_{(3)}^\top$ are constant and have only to be computed once at the beginning of the algorithm. Finally, $X_{(3)} M_{(3)}^\top$ and $M_{(3)} M_{(3)}^\top$ are the products of *sparse* matrices such that their computational complexity depends only on the number of nonzeros in $\mathbf{X}$ or $\mathbf{M}$. A full derivation of equation (4) can be found in the supplementary material A.4.

Updates for A and $\mathbf{R}$ The updates for A and $\mathbf{R}$ can be derived directly from the RESCAL-ALS algorithm by setting $\mathbf{E} = \mathbf{X} - \mathbf{M} \times_3 W$ and computing the RESCAL factorization of $\mathbf{E}$. The updates for A can therefore be computed by:

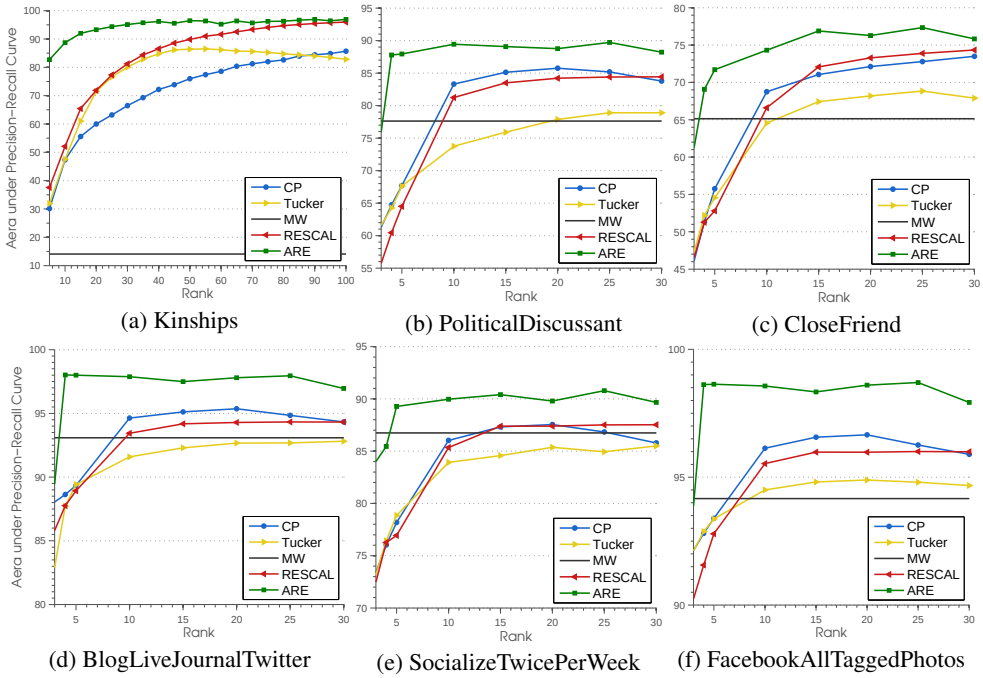
$$A \leftarrow \left(\sum_{k=1}^K E_k A R_k^\top + E_k^\top A R_k \right) \left(\sum_{k=1}^K R_k A^\top A R_k^\top + R_k^\top A^\top A R_k + \lambda I \right)^{-1}$$

where $E_k = X_k - \mathbf{M} \times_3 \boldsymbol{w}_k$ and $\boldsymbol{w}_k$ denotes the k-th row of W.

The updates of $\mathbf{R}$ can be computed in the following way: Let $A = U \Sigma V^\top$ be the SVD of A, where σ_i is the i-th singular value of A. Furthemore, let S be a matrix with entries $s_{ij} = \sigma_i \sigma_j / (\sigma_i^2 \sigma_j^2 + \lambda_R)$. An update of R_k can then be computed via $R_k \leftarrow V \left(S * (U^\top (X_k - \mathbf{M} \times_3 \boldsymbol{w}_k) U) \right) V^\top$, where "$*$" denotes the Hadamard product. For a full derivation of these updates please see [19].

4 Evaluation

We evaluated ARE on various multi-relational datasets where we were in particular interested in its generalization ability relative to the factorization rank. For comparison, we included the well-known CP and Tucker tensor factorizations in the evaluation, as well as RESCAL and the non-latent model $\mathbf{X} \approx \mathbf{M} \times_3 W$ (in the following denoted by $\mathbf{M}W$). In all experiments, the oracle tensor $\mathbf{M}$ used in $\mathbf{M}W$ and ARE is identical, such that the results of $\mathbf{M}W$ can be regarded as a baseline for the contribution of the heuristic methods to ARE. Following [10, 11, 27, 20] we used k-fold cross-validation for the evaluation, partitioning the entries of the adjacency tensor into training, validation, and test sets. In



(a) Kinships (b) PoliticalDiscussant (c) CloseFriend

(d) BlogLiveJournalTwitter (e) SocializeTwicePerWeek (f) FacebookAllTaggedPhotos

Figure 1: Evaluation results for AUC-PR on the Kinships (1a) and Social Evolution data sets (1b-1f).

the test and validation folds all entries are set to 0. Due to the large imbalance of true and false relationships, we used the area under the precision-recall curve (AUC-PR) to measure predictive performance, which is known to behave better with imbalanced classes then AUC-ROC. All AUC-PR results are averaged over the different test-folds. Links and references for the datasets used in the evaluation are provided in the supplementary material A.5.

Social Evolution First, we evaluated ARE on a dataset consisting of multiple relations of persons living in an undergraduate dormitory. From the relational data, we constructed a $84 \times 84 \times 5$ adjacency tensor where two modes correspond to persons and the third mode represents the relations between these persons such as friendship (*CloseFriend*), social media interaction (*BlogLivejournalTwitter* and *FacebookAllTaggedPhotos*), political discussion (*PoliticalDiscussant*), and social interaction (*SocializeTwicePerWeek*). For each relation, we performed link prediction via 5-fold cross validation. The oracle tensor $\mathbf{M}$ consisted only of a copy of the observed tensor $\mathbf{X}$. Including $\mathbf{X}$ in $\mathbf{M}$ allows ARE to efficiently exploit patterns where the existence of a social relationship for a particular pair of persons is predictive for other social interactions between exactly this pair of persons (e.g. close friends are more likely to socialize twice per week). It can be seen from the results in figure $1(b-f)$ that ARE achieves better performance than all competing approaches and already achieves excellent performance at a very low rank, what supports our theoretical considerations.

Kinship The Kinship dataset describes the kinship relations in the Australian Alyawarra tribe in terms of 26 kinship relations between 104 persons. The task in the experiment was to predict unknown kinship relations via 10-fold cross validation in the same manner as in [20]. Table 1 shows the improvement of ARE over state-of-the-art relational learning methods. Figure 1a shows the predictive performance compared to the rank of multiple factorization methods. It can be seen that ARE outperforms all other methods significantly for lower rank. Moreover, starting from rank 40 ARE gives already comparable results to the best results in table 1. As in the previous experiments, $\mathbf{M}$ consisted only of a copy of $\mathbf{X}$. On this dataset, the copy of $\mathbf{X}$ allows ARE to model efficiently that the relations in the data are mutually exclusive by setting $w_{ii} > 0$ and $w_{ij} < 0$ for all $i \neq j$. This also explains the large improvement of ARE over RESCAL for small ranks.

Link Prediction on Semantic Web Data The SWRC ontology models a research group in terms of people, publications, projects, and research interests. The task in our experiments was to predict the affiliation relation, i.e. to map persons to research groups. We followed the experimental setting in [17]: From the raw data, we created a $12058 \times 12058 \times 85$ tensor by considering all directly

connected entities of persons and research groups. In total, 168 persons and 5 research groups are considered in the evaluation data. The oracle tensor $\mathbf{M}$ consisted again of a copy of $\mathbf{X}$ and of the common neighbor heuristics $X_i X_i$ and $X_i^\top X_i^\top$. These heuristics were included to model patterns like *people who share the same research interest are likely in the same affiliation* or *a person is related to a department if the person belongs to a group in the department*. We also imposed a sparsity penalty on W to prune away inactive heuristics during iterations. Table 2 shows that ARE improved the results significantly over three state-of-the-art link prediction methods for Semantic Web data. Moreover, whereas RESCAL required a rank of 45, ARE required only a small rank of 15.

Figure 2: Runtime on Cora

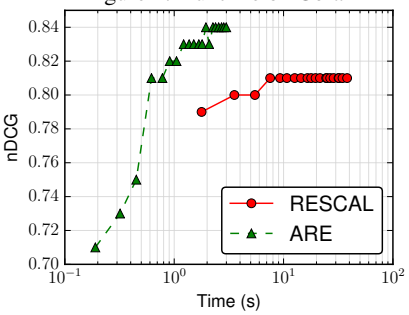


Table 1: Evaluation Results on Kinships.

	MRC [11]	BCTF [27]	LFM [8]	RESCAL	ARE
AUC	86	90	94.6	96	**96.9**
Rank	-	-	(50,50,500)	100	**90**

Table 2: Evaluation results on SWRC.

	SVD	Subtrees [17]	RESCAL	MW	ARE
nDCG	0.8	0.95	0.96	0.59	**0.99**

Runtime Performance To evaluate the trade-off between runtime and predictive performance we recorded the nDCG values of RESCAL and ARE after each iteration of the respective ALS algorithms on the Cora citation database. We used the variant of Cora in which all publications are organized in a hierarchy of topics with two to three levels and 68 leaves. The relational data consists of information about paper citations, authors and topics from which a tensor of size $28073 \times 28073 \times 3$ is constructed. The oracle tensor consisted of a copy of $\mathbf{X}$ and the common neighbor patterns $X_i X_j$ and $X_i^\top X_j^\top$ to model patterns such that a cited paper shares the same topic, a cited paper shares the same author etc. The task of the experiment was to predict the leaf topic of papers by 5-fold cross-validation on a moderate PC with Intel(R) Core i5 @3.1GHz, 4G RAM. The optimal rank 220 for RESCAL was determined out of the range $[10, 300]$ via parameter selection. For ARE we used a significantly smaller rank 20. Figure 2 shows the runtime of RESCAL and ARE compared to their predictive performance. It is evident that ARE outperforms RESCAL after a few iterations although the rank of the factorization is decreased by an order of magnitude. Moreover, ARE surpasses the best prediction results of RESCAL in terms of total runtime even before the first iteration of RESCAL-ALS has terminated.

5 Concluding Remarks

In this paper we considered learning from latent and observable patterns on multi-relational data. We showed analytically that the rank of adjacency tensors is upper bounded by the sum of diclique partition numbers and lower bounded by the maximum number of strongly connected components of any relation in the data. Based on our theoretical results, we proposed an additive tensor factorization approach for learning from multi-relational data which combines strengths from latent and observable variable methods. Furthermore we presented an efficient and scalable algorithm to compute the factorization. Experimentally we showed that the proposed approach does not only increase the predictive performance but is also very successful in reducing the required rank — and therefore also the required runtime — of the factorization. The proposed additive model is one option to overcome the rank-scalability problem outlined in section 2, however not the only one. In future work we intend to investigate to what extent sparse or hierarchical models can be used to the same effect.

Acknowledgements Maximilian Nickel acknowledges support by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216. We thank Youssef Mroueh and Lorenzo Rosasco for clarifying discussions on the theoretical part of this paper.

References

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. "Mixed Membership Stochastic Blockmodels". In: *Journal of Machine Learning Research* 9 (2008), pp. 1981–2014.

[2] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. "Learning Structured Embeddings of Knowledge Bases". In: *Proceedings of the 25th Conference on Artificial Intelligence*. 2011.

[3] R. A. Brualdi and H. J. Ryser. *Combinatorial Matrix Theory*. 1991.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, Jr, and T. Mitchell. "Toward an Architecture for Never-Ending Language Learning". In: *AAAI*. 2010, pp. 1306–1313.

[5] X. L. Dong, K. Murphy, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, T. Strohmann, S. Sun, and W. Zhang. "Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion". In: *Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2014.

[6] L. Getoor, N. Friedman, D. Koller, A. Pfeffer, and B. Taskar. "Probabilistic Relational Models". In: *Introduction to statistical relational learning*. 2007, pp. 129–174.

[7] P. D. Hoff. "Modeling homophily and stochastic equivalence in symmetric relational data". In: *Advances in Neural Information Processing Systems*. Vol. 20. 2008, pp. 657–664.

[8] R. Jenatton, N. Le Roux, A. Bordes, and G. Obozinski. "A latent factor model for highly multi-relational data". In: *Advances in Neural Information Processing Systems*. Vol. 25. 2012, pp. 3176–3184.

[9] X. Jiang, V. Tresp, Y. Huang, and M. Nickel. "Link Prediction in Multi-relational Graphs using Additive Models." In: *Proceedings of International Workshop on Semantic Technologies meet Recommender Systems & Big Data at the ISWC*. Vol. 919. 2012, pp. 1–12.

[10] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. "Learning systems of concepts with an infinite relational model". In: *AAAI*. Vol. 3. 2006, p. 5.

[11] S. Kok and P. Domingos. "Statistical Predicate Invention". In: *Proceedings of the 24th International Conference on Machine Learning*. 2007, pp. 433–440.

[12] T. G. Kolda and B. W. Bader. "Tensor Decompositions and Applications". In: *SIAM Review* 51.3 (2009), pp. 455–500.

[13] T. G. Kolda, B. W. Bader, and J. P. Kenny. "Higher-order web link analysis using multilinear algebra". In: *Proceedings of the Fifth International Conference on Data Mining*. 2005, pp. 242–249.

[14] Y. Koren. "Factorization meets the neighborhood: a multifaceted collaborative filtering model". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2008, pp. 426–434.

[15] N. Lao and W. W. Cohen. "Relational retrieval using a combination of path-constrained random walks". In: *Machine learning* 81.1 (2010), pp. 53–67.

[16] D. Liben-Nowell and J. Kleinberg. "The link-prediction problem for social networks". In: *Journal of the American society for information science and technology* 58.7 (2007), pp. 1019–1031.

[17] U. Lösch, S. Bloehdorn, and A. Rettinger. "Graph Kernels for RDF Data". In: *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012*. Vol. 7295. 2012, pp. 134–148.

[18] S. D. Monson, N. J. Pullman, and R. Rees. "A survey of clique and biclique coverings and factorizations of (0,1)-matrices". In: *Bulletin of the ICA* 14 (1995), pp. 17–86.

[19] M. Nickel. "Tensor factorization for relational learning". PhD thesis. LMU München, 2013.

[20] M. Nickel, V. Tresp, and H.-P. Kriegel. "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.

[21] M. Nickel, V. Tresp, and H.-P. Kriegel. "Factorizing YAGO: scalable machine learning for linked data". In: *Proceedings of the 21st international conference on World Wide Web*. 2012, pp. 271–280.

[22] J. R. Quinlan. "Learning logical definitions from relations". In: *Machine Learning* 5 (1990), pp. 239–266.

[23] M. Richardson and P. Domingos. "Markov logic networks". In: *Machine Learning* 62.1 (2006), pp. 107–136.

[24] D. Serre. *Matrices: Theory and applications*. Vol. 216. 2010.

[25] A. P. Singh and G. J. Gordon. "Relational learning via collective matrix factorization". In: *Proc. of the 14th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*. 2008, pp. 650–658.

[26] F. M. Suchanek, G. Kasneci, and G. Weikum. "Yago: A Core of Semantic Knowledge". In: *Proceedings of the 16th international conference on World Wide Web*. 2007, pp. 697–706.

[27] I. Sutskever, R. Salakhutdinov, and J. Tenenbaum. "Modelling Relational Data using Bayesian Clustered Tensor Factorization". In: *Advances in Neural Information Processing Systems 22*. 2009, pp. 1821–1828.

[28] Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. "Infinite Hidden Relational Models". In: *Proc. of the Twenty-Second Conference Annual Conference on Uncertainty in Artificial Intelligence*. 2006, pp. 544–551.

A Supplementary Material

A.1 Proof for Upper Bound on Multilinear Rank

In the following we will proof the upper bound on multilinear rank for lemma 3

Proof. Let $\text{rank}(X_m) = r_{\max}$. It follows from the basic properties of matrix rank that X_m has $r_{\max}$ linearly independent rows. Since the unfolding of $\mathbf{X}$ in the first and second mode is a block matrix where the k-th block corresponds to X_k or its transpose, i.e.

$$X_{(1)} = \begin{bmatrix} X_1 & X_2 & \dots & X_K \end{bmatrix}$$
$$X_{(2)} = \begin{bmatrix} X_1^\top & X_2^\top & \dots & X_K^\top \end{bmatrix},$$

it follows that $X_{(1)}$ and $X_{(2)}$ have also at least $r_{\max}$ linearly independent rows and at most $\sum_k \text{rank}(X_k)$ independent rows, such that $\sum_k \text{rank}(X_k) \geqslant r_1, r_2 \geqslant r_{\max}$ □

A.2 Link Prediction Methods

Table 3 lists typical examples for relational learning functions that we will consider for the construction of $\mathbf{M}$. In table 3 $\text{score}(v_1, v_2)$ denotes the score that a function assigns to a link, while $N(v_1)$ denotes

Table 3: Link Prediction Heuristics

Method	$\text{score}(v_1, v_2)$
Common Neighbors	$\lvert N(v_1) \cap N(v_2)\rvert$
Jaccard Coefficient	$\dfrac{\lvert N(v_1) \cap N(v_2)\rvert}{\lvert N(v_1) \cup N(v_2)\rvert}$
Adamic/Adar	$\sum_{z \in N(v_1) \cap N(v_2)} (\log \lvert N(z)\rvert)^{-1}$
Katz	$\sum_k^\infty \beta^k \lvert \text{paths}(v_1, v_2, k)\rvert$
Horn Clause	$\begin{cases} 1, & \text{if } P_1 \wedge P_2 \wedge \dots \wedge P_n \\ 0, & \text{else.} \end{cases}$

the set of neighbors of vertex v_1, and $\text{path}(v_1, v_2, k)$ denotes the set of all paths between vertices v_1 and v_2 of length k. For the definition of neighborhood in a digraph see definition 6:

Definition 6 (Neighorhood)**.** *Let $\Gamma = (\mathcal{V}, \mathcal{E})$ be a digraph. The* in-neighborhood *of a vertex v is defined as $N^-(v) = \{u | u \rightsquigarrow v \in \mathcal{E}\}$, the* out-neighborhood *is defined as $N^+(v) = \{u | v \rightsquigarrow u \in \mathcal{E}\}$, and the* neighborhood *of v is defined as $N^-(v) \cup N^+(v)$.*

A.3 Computational Complexity of Tensor Factorizations

Here, we review the computational complexity of standard algorithms to compute tensor factorizations with regard to the rank of an adjacency tensor. Alternating least-squares algorithms (ALS) are the "workhorse" algorithms to compute the CP decomposition and the RESCAL factorization, while the higher-order orthogonal iterations (HOOI) algorithm is commonly used to computed the Tucker decomposition [2, 6].

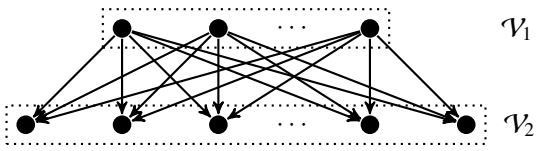


Figure 3: Illustration of a diclique.

It can be verified easily that one iteration of CP-ALS scales quadratic with the number of entities and cubic with the number of latent components r when factorizing an adjacency tensor $\mathbf{X} \in \{0,1\}^{N \times N \times K}$. To compute a single update of C (and analogously for A and B), the following term has to be computed

$$C \leftarrow X_{(1)}(B \odot A)(B^\top B * A^\top A)^\dagger. \tag{5}$$

Since $B^\top B * A^\top A$ is a $r \times r$ matrix, the computational complexity of $(B^\top B * A^\top A)^\dagger$ is $O(r^3)$. Furthermore, $B \odot A$ is an $N^2 \times r$ matrix, such that its computation needs $O(N^2 r)$ operations.

To compute the Tucker decomposition using HOOI, it is necessary to compute the r_1 largest eigenvectors of $Y_{(1)}Y_{(1)}^\top$ where $Y_{(1)} = G_{(1)}(C \otimes B)^\top$. Since $Y_{(1)}$ is an $N \times r_2 r_3$ matrix, the matrix product $Y_{(1)}Y_{(1)}^\top$ alone already needs $O(N^2 r_2 r_3)$ operations. Furthermore, to compute the r_1 largest eigenvectors of a $N \times N$ matrix the implicitly restarted Arnoldi method (IRAM) is used which has a computational complexity of $O(N r_1^2)$ [3]. Derivations for r_2 and r_3 are analogous.

RESCAL-ALS scales linearly with the data size, i.e. linearly with the number of entities, number of relations, and the number of known facts. The computational complexity with regard to the number of latent components r is $O(r^3)$. For a full derivation of the runtime complexity see Nickel [5].

A.4 Derivation of Updates for R_k and W

The improve updates for W can be derive from the following equality

$$\begin{aligned}(R \times_1 A \times_2 A)_{(3)} M_{(3)}^\top &= R_{(3)}(A \otimes A)^\top M_{(3)}^\top \\ &= R_{(3)} \left(M_{(3)}(A \otimes A)\right)^\top \\ &= R_{(3)} \left(\mathbf{M} \times_1 A^\top \times_2 A^\top\right)_{(3)}^\top\end{aligned}$$

Please note that A is not required to be orthonormal.

The runtime complexity of computing $(R \times_1 A \times_2 A)_{(3)} M_{(3)}^\top$ is $O(N^2 K r + \text{nnz}(M) N)$, while the computational complexity of $R_{(3)}(\mathbf{M} \times_1 A^\top \times_2 A^\top)_{(3)}$ is only $O(NKr^3 + \text{nnz}(M) r)$.

A.5 Datasets

The datasets used in the evaluation are available from the following locations:

Social Evolution [4]	`http://realitycommons.media.mit.edu/socialevolution.html`
Kinships Denham [1]	`http://alchemy.cs.washington.edu/data/kinships/`
SWRC [7]	`http://ontoware.org/swrc/`
Cora	`https://people.cs.umass.edu/~mccallum/data.html`

References

[1] W. Denham. "The detection of patterns in Alyawarra nonverbal behavior". PhD thesis. University of Washington, 1973.

[2] T. G. Kolda and B. W. Bader. "Tensor Decompositions and Applications". In: *SIAM Review* 51.3 (2009), pp. 455–500.

[3] R. R. B. Lehoucq, D. D. C. Sorensen, and C.-C. Yang. *Arpack User's Guide: Solution of Large-Scale Eigenvalue Problems With Implicityly Restarted Arnoldi Methods*. Vol. 6. 1998.

[4] A. Madan, M. Cebrian, S. Moturu, K. Farrahi, and S. Pentland. "Sensing the 'Health State'of a Community". In: (2012).

[5] M. Nickel. "Tensor factorization for relational learning". PhD thesis. LMU München, 2013.

[6] M. Nickel, V. Tresp, and H.-P. Kriegel. "Factorizing YAGO: scalable machine learning for linked data". In: *Proceedings of the 21st international conference on World Wide Web*. 2012, pp. 271–280.

[7] Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, and D. Oberle. "The SWRC Ontology – Semantic Web for Research Communities". In: *Progress in Artificial Intelligence*. 3808. 2005, pp. 218–231.