Tensor Factorization for Multi-Relational Learning

Maximilian Nickel[1] and Volker Tresp[2]

[1] Ludwig Maximilian University, Oettingenstr. 67, Munich, Germany
nickel@dbs.ifi.lmu.de
[2] Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, Munich, Germany
volker.tresp@siemens.com

Abstract. Tensor factorization has emerged as a promising approach for solving relational learning tasks. Here we review recent results on a particular tensor factorization approach, i.e. RESCAL, which has demonstrated state-of-the-art relational learning results, while scaling to knowledge bases with millions of entities and billions of known facts.

1 Introduction

Exploiting the information contained in the relationships between entities has been essential for solving a number of important machine learning tasks. For instance, social network analysis, bioinformatics, and artificial intelligence all make extensive use of relational information, as do large knowledge bases such as Google's Knowledge Graph or the Semantic Web. It is well-known that, in these and similar domains, statistical relational learning (SRL) can improve learning results significantly over non-relational methods. However, despite the success of SRL in specific applications, wider adoption has been hindered by multiple factors: without extensive prior knowledge about a domain, existing SRL methods often have to resort to structure learning for their functioning; a process that is both time consuming and error prone. Moreover, inference is often based on methods such as MCMC and variational inference which introduce additional scalability issues. Recently, tensor factorization has been explored as an approach that overcomes some of these problems and that leads to highly scalable solutions. Tensor factorizations realize multi-linear latent factor models and contain commonly used matrix factorizations as the special case of bilinear models. We will discuss tensor factorization for relational learning by the means of RESCAL [6,7,5], which is based on the factorization of a third-order tensor and which has shown excellent learning results; outperforming state-of-the-art SRL methods and related tensor-based approaches on benchmark data sets. Moreover, RESCAL is highly scalable such that large knowledge bases can be factorized, which is currently out of scope for most SRL methods. In our review of this model, we will also exemplify the general benefits of tensor factorization for relational learning, as considered recently in approaches like [10,8,1,4,2]. In the following, we will mostly follow the notation outlined in [3]. We will also assume that all relationships are of dyadic form.

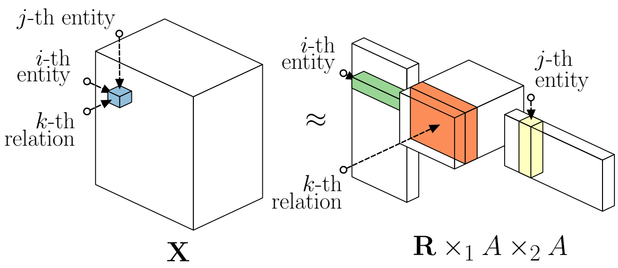

Fig. 1. Factorization of an adjacency tensor $\mathbf{X}$ using the RESCAL model.

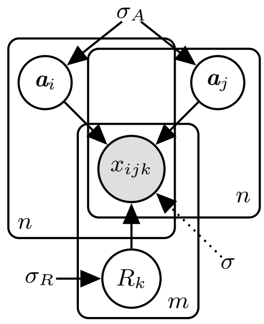

Fig. 2. Graphical plate model of RESCAL.

2 Relational Learning via Tensor Factorization

Dyadic relational data has a natural representation as an adjacency tensor $\mathbf{X} \in \mathbb{R}^{n \times n \times m}$ whose entries x_{ijk} correspond to all possible relationships between n entities over m different relations. The entries of $\mathbf{X}$ are set to

$$x_{ijk} = \begin{cases} 1, & \text{if the relationship } \mathit{relation}_k(\mathit{entity}_i, \mathit{entity}_j) \text{ exists} \\ 0, & \text{otherwise.} \end{cases}$$

RESCAL [6] is a latent factor model for relational learning, which factorizes an adjacency tensor $\mathbf{X}$ into a core tensor $\mathbf{R} \in \mathbb{R}^{r \times r \times m}$ and a factor matrix $A \in \mathbb{R}^{n \times r}$ such that

$$\mathbf{X} \approx \mathbf{R} \times_1 A \times_2 A. \tag{1}$$

Equation (1) can be equivalently specified as $x_{ijk} \approx \boldsymbol{a}_i^T R_k \boldsymbol{a}_j$, where the column vector $\boldsymbol{a}_i \in \mathbb{R}^r$ denotes the i-th row of A and the matrix $R_k \in \mathbb{R}^{r \times r}$ denotes the k-th frontal slice of $\mathbf{R}$. Consequently, $\boldsymbol{a}_i$ corresponds to the latent representation of entity_i, while R_k models the interactions of the latent variables for $\mathit{relation}_k$. The dimensionality r of the latent space A is a user-given parameter which specifies the complexity of the model. The symbol "≈" denotes the approximation under a given loss function. Figure 1 shows a visualization of the factorization. Probabilistically, eq. (1) can be interpreted as estimating the joint distribution over *all possible* relationships

$$\mathrm{P}(X|A, \mathbf{R}) = \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{k=1}^{m} \mathrm{P}(x_{ijk}|\, \boldsymbol{a}_i^T R_k \boldsymbol{a}_j). \tag{2}$$

Hence, a RESCAL factorization of an adjacency tensor $\mathbf{X}$ computes a complete model of a relational domain where the state of a relationship x_{ijk} depends on the matrix-vector product $\boldsymbol{a}_i^T R_k \boldsymbol{a}_j$. Here, a Gaussian likelihood model would imply a least squares loss function, while a Bernoulli likelihood model would imply a logistic loss function [6,5]. To model attributes of entities efficiently, coupled tensor factorization can be employed [7,11], where simultaneously to eq. (1) an attribute matrix $F \in \mathbb{R}^{n \times \ell}$ is factorized such that $F \approx AW$ and where the latent factor A is shared between the factorization of $\mathbf{X}$ and F. RESCAL and other tensor factorizations feature a number of important properties that can be exploited for tasks like link prediction, entity resolution or link-based clustering:

Efficient Inference The latent variable structure of RESCAL decouples inference such that global dependencies are captured during learning, whereas prediction relies only on a typically small number of latent variables. It can be seen from eq. (2) that a variable x_{ijk} is conditionally independent from all other variables given the expression $\boldsymbol{a}_i^T R_k \boldsymbol{a}_j$. The computational complexity of these matrix-vector multiplications depends only on the dimensionality of the latent space A, what enables, for instance, fast query answering on knowledge bases. It is important to note that this locality of computation does not imply that the likelihood of a relationship is only influenced by local information. On the contrary, the conditional independence assumptions depicted in fig. 2 show that information is propagated globally when computing the factorization. Due to the colliders in fig. 2, latent variables ($\boldsymbol{a}_i$, $\boldsymbol{a}_j$, R_k) are not d-separated from other variables such that they are possibly dependent on all remaining variables. Therefore, as the variable x_{ijk} depends on its associated latent variables $\{\boldsymbol{a}_i, \boldsymbol{a}_j, R_k\}$, it depends *indirectly* on the state of any other variable such that global dependencies between relationships can be captured. Similar arguments apply to tensor factorizations such as the TUCKER decomposition and CP, which explains the strong relational learning results of RESCAL and CP compared to state-of-the-art methods such as MLN or IRM [6,7,2,5].

Unique Representation A distinctive feature of RESCAL is the unique representation of entities via the latent space A. Standard tensor factorization models such as CP and TUCKER compute a bipartite model of relational data, meaning that entities have different latent representations as subjects or objects in a relationship. For instance, a TUCKER-2 model would factorize the frontal slices of an adjacency tensor $\mathbf{X}$ as $X_k \approx AR_kB^T$ such that entities are represented as subjects via the latent factor A and as objects via the latent factor B. However, relations are usually not bipartite and in these cases a bipartite modeling would effectively break the flow of information from subjects to objects, as it does not account for the fact that the latent variables $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ refer to the identical entity. In contrast, RESCAL uses a unique latent representation $\boldsymbol{a}_i$ for each entity in the data set, what enables efficient information propagation via the dependency structure shown in fig. 2 and what has been demonstrated to be critical for capturing correlations over relational chains. For instance, consider the task to predict the party membership of presidents of the United States of America. When the party membership of a president's vice president is known, this can be done with high accuracy, as both persons have usually been members of the same party, meaning that the formula $\mathit{vicePresident}(x, y) \wedge \mathit{party}(y, z) \Rightarrow \mathit{party}(x, z)$ holds with high probability. For this and similar examples, it has been shown that bipartite models such as CP and TUCKER fail to capture the necessary correlations, as, for instance, the object representation $\boldsymbol{b}_y$ does not reflect that person y is in a relationship to party z as a subject. RESCAL, on the other hand, is able to propagate the required information, e.g. the party membership of y, via the unique latent representations of the involved entities [6,7].

Latent Representation In relational data, the similarity of entities is determined by the similarity of their relationships, following the intuition that "if two objects are in the same relation to the same object, this is evidence that they may be the same object" [9]. This notion of similarity is reflected in RESCAL via the latent space A. For the i-th entity, all possible occurrences as a subject are grouped in the slice $X_{i,:,:}$ of an adjacency tensor, while all possible occurrences as an object are grouped in the slice $X_{:,i,:}$ (see figs. 3 and 4). According to the RESCAL model, these slices are computed by $\text{vec}\,(X_{i,:,:}) \approx \boldsymbol{a}_i R_{(1)}(I \otimes A)^T$ and $\text{vec}\,(X_{:,i,:}) \approx \boldsymbol{a}_i R_{(2)}(I \otimes A)^T$. Since the terms $R_{(1)}(I \otimes A)^T$ and $R_{(2)}(I \otimes A)^T$ are constant for different values of i, it is sufficient to consider only the similarity of $\boldsymbol{a}_p$ and $\boldsymbol{a}_q$ to determine the *relational* similarity of $entity_p$ and $entity_q$. As this measure of similarity is based on the latent representations of entities, it is not only based on counting identical relationships of identical entities, but it also considers the similarity of the entities that are involved in a relationship. The previous intuition could therefore be restated as *if two objects are in similar relations to similar objects, this is evidence that they may be the same object*. Latent representations of entities have been exploited very successfully for entity resolution and also enabled large-scale hierarchical clustering on relational data [6,7]. Moreover, since the matrix A is a vector space representation of entities, non-relational machine learning algorithms such as k-means or kernel methods can be conveniently applied to any of these tasks.

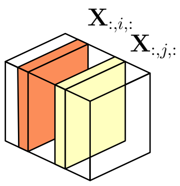

Fig. 3. Incoming Links

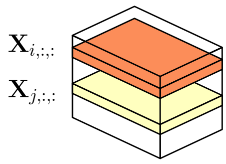

Fig. 4. Outgoing Links

High Scalability The scalability of algorithms has become of utmost importance as relational data is generated in an unprecedented amount and the size of knowledge bases grows rapidly. RESCAL-ALS is a highly scalable algorithm to compute the RESCAL model under a least-squares loss. It has been shown that it can efficiently exploit the sparsity of relational data as well as the structure of the factorization such that it features *linear* runtime complexity with regard to the number of entities n, the number of relations m, and the number of known relationships $\text{nnz}(\mathbf{X})$, while being cubic in the model complexity r. This property allowed, for instance, to predict various high-level classes of entities in the YAGO 2 ontology, which consists of over three million entities, over 80 relations or attributes, and over 33 million existing relationships, by computing low-rank factorizations of its adjacency tensor on a single desktop computer [7].

3 Conclusion and Outlook

Rescal has shown state-of-the-art relational learning results, while scaling up to the size of complete knowledge bases. Due to its latent variable structure, Rescal does not require deep domain knowledge and therefore can be easily applied to most domains. Its latent representation of entities enables the application of non-relational algorithms to relational data for a wide range of tasks such as cluster analysis or entity resolution. Rescal is applicable if latent factors are suitable for capturing the essential information in a domain. In ongoing research, we explore situations where plain latent factor models are not a very efficient approach to relational learning and examine how to overcome the underlying causes of these situations.

References

1. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Proc. of the 25th Conference on Artificial Intelligence. SF, USA (2011)
2. Jenatton, R., Le Roux, N., Bordes, A., Obozinski, G.: A latent factor model for highly multi-relational data. In: Advances in Neural Information Processing Systems. vol. 25 (2012)
3. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Review 51(3), 455—500 (2009)
4. Kolda, T.G., Bader, B.W., Kenny, J.P.: Higher-order web link analysis using multilinear algebra. In: Proc. of the Fifth International Conference on Data Mining. pp. 242—249 (2005)
5. Nickel, M., Tresp, V.: Logistic tensor-factorization for multi-relational data. In: ICML Workshop - Structured Learning: Inferring Graphs from Structured and Unstructured Inputs. Atlanta, GA, USA (2013)
6. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proc. of the 28th International Conference on Machine Learning. pp. 809—816. Bellevue, WA, USA (2011)
7. Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing YAGO: scalable machine learning for linked data. In: Proc. of the 21st International World Wide Web Conference. Lyon, France (2012)
8. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proc. of the 19th International Conference on World Wide Web. pp. 811—820 (2010)
9. Singla, P., Domingos, P.: Entity resolution with markov logic. In: Proc. of the Sixth International Conference on Data Mining. p. 572582. Washington, DC, USA (2006)
10. Sutskever, I., Salakhutdinov, R., Tenenbaum, J.B.: Modelling relational data using bayesian clustered tensor factorization. Advances in Neural Information Processing Systems 22 (2009)
11. Ylmaz, Y.K., Cemgil, A.T., Simsekli, U.: Generalised coupled tensor factorisation. In: Advances in Neural Information Processing Systems (2011)