

Tensor Factorization for Multi-Relational Learning

Maximilian Nickel¹ and Volker Tresp²

¹ Ludwig Maximilian University, Oettingenstr. 67, Munich, Germany
`nickel@dbis.lmu.de`

² Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, Munich, Germany
`volker.tresp@siemens.com`

Abstract. Relational learning has become ubiquitous in many fields of application. Here, we review tensor factorization for relational learning on the basis of RESCAL, which has shown state-of-the-art relational learning results, while scaling to knowledge bases with millions of entities and possibly billions of known facts.

1 Introduction

Modeling information via relationships of entities is a powerful method of knowledge representation and has become ubiquitous in many fields of application. For instance, social networks, 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 statistical relational learning (SRL) can improve learning results significantly over non-relational methods in such domains. 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 what introduces additional scalability issues and renders large-scale relational learning difficult to realize. Tensor factorizations are multi-linear latent factor models, which allow for a significantly different approach to SRL, in order to overcome aforementioned problems and to create new applications for relational learning. 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 has shown excellent learning results, outperforming state-of-the-art SRL methods as well as related tensor-based approaches on benchmark data sets. Moreover, the factorization is highly scalable, such that large knowledge bases can be factorized, what is currently out of reach for most SRL methods. During the review of the RESCAL 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 remainder, we will mostly follow the notation outlined in [3] and also assume that all relationships are of dyadic form.

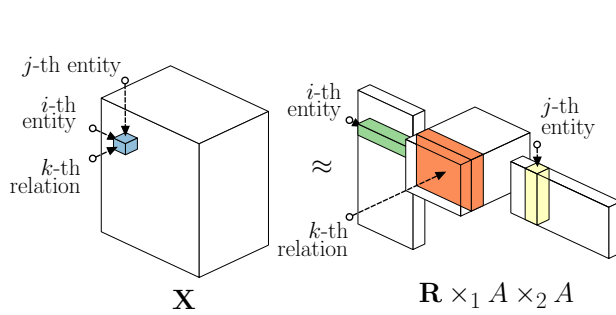


Fig. 1. Factorization of an adjacency tensor \mathbf{X} using the RESCAL model.

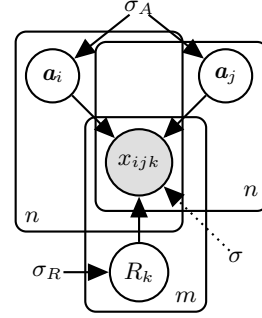


Fig. 2. Graphical model of RESCAL in plate notation.

2 Relational Learning via Tensor Factorization

Dyadic relational data consisting of n entities and m relations has a natural representation as an adjacency tensor $\mathbf{X} \in \mathbb{R}^{n \times n \times m}$ with entries $x_{ijk} = 1$ if the relationship $relation_k(entity_i, entity_j)$ exists and 0 otherwise. RESCAL [6] is a latent factor model for relational learning, which factorizes an adjacency tensor \mathbf{X} as

$$\mathbf{X} \approx \mathbf{R} \times_1 A \times_2 A, \quad (1)$$

what can be equivalently stated as $x_{ijk} \approx \mathbf{a}_i^T R_k \mathbf{a}_j$. In this model, the row $\mathbf{a}_i \in \mathbb{R}^r$ of $A \in \mathbb{R}^{n \times r}$ corresponds to the latent representation of $entity_i$, while the frontal slice $R_k \in \mathbb{R}^{r \times r}$ of the tensor $\mathbf{R} \in \mathbb{R}^{r \times r \times m}$ models the interactions of the latent variables for $relation_k$. The dimensionality r of the latent space is a user-given parameter which specifies the complexity of the model. The symbol “ \approx ” 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

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

For the least squares loss, the probability of a relationship is proportional to the entry in the reconstructed tensor, meaning $P(x_{ijk} = 1) \propto \mathbf{a}_i^T R_k \mathbf{a}_j$ [6]. Using the logistic loss, exact probabilities can be obtained [5]. To handle 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 as $F \approx AW$ such that $A \in \mathbb{R}^{n \times r}$ 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 few latent variables. It can be seen from eq. (2) that a variable x_{ijk} is conditionally independent from all other variables given $\mathbf{a}_i^T R_k \mathbf{a}_j$, what enables fast query answering independently of the size of a knowledge base and what can essentially be computed in real-time. 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 repeated colliders in fig. 2, latent variables $(\mathbf{a}_i, \mathbf{a}_j, R_k)$ can not be d -separated from any other variable and thus are possibly dependent on all these variables. Since x_{ijk} depends only on $\{\mathbf{a}_i, \mathbf{a}_j, R_k\}$, it depends *indirectly* on the state of any other variable, such that global dependencies between relationships can be captured. Similar arguments apply to factorizations such as TUCKER and CP what explains the generally 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, such that entities have different latent representations whether they occur as subjects or objects in a relationship. For instance, factorizing a frontal slice as $X_k \approx AR_k B^T$ would represent entities as subjects in A and as objects in B . However, relations are usually not bipartite and this modeling would effectively break the flow of information from subjects to objects in these cases, as it does not account for the fact that latent variables \mathbf{a}_i and \mathbf{b}_i refer to identical entities. In contrast, RESCAL uses exactly one latent representation \mathbf{a}_i for the i -th entity, what enables an efficient propagation of information via the dependency structure shown in fig. 2. It has been demonstrated that this property is critical to propagate information over relational chains such as $vicePresident(x, y) \wedge party(y, z) \Rightarrow party(x, z)$, what allows RESCAL to outperform CP and TUCKER significantly on such prediction tasks [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 slice $X_{i,:}$ of an adjacency tensor, while all possible occurrences as an object are grouped in $X_{:,i}$ (see figs. 3 and 4). According to the RESCAL model, these slices are computed by $\text{vec}(X_{i,:} \approx \mathbf{a}_i R_{(1)} (I \otimes A)^T)$ and $X_{:,i} \approx \mathbf{a}_i R_{(2)} (I \otimes A)^T$. Since $R_{(1)} (I \otimes A)^T$ and $R_{(2)} (I \otimes A)^T$ are constant for different values of i , it is sufficient to consider the similarity of \mathbf{a}_p and \mathbf{a}_q to compute the *relational* similarity of *entity_p* and *entity_q*. As this measure of similarity is based on the latent representations of entities, it reflects that *if two objects are in similar relations to similar objects, this is evidence that they may be the same object*, what is an important extension to the previous definition. Latent representations of entities have been exploited very

successfully for entity resolution and also enabled the first large-scale hierarchical clustering approach 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.

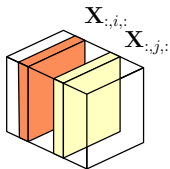


Fig. 3. Incoming Links

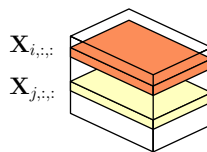


Fig. 4. Outgoing Links

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 and being straightforward to use without any need for prior knowledge. Moreover, its latent representations of entities enable the application of non-relational algorithms to relational data for a wide range of tasks, such that proven methods can be reused. This set of features opens up new applications for relational learning and can be an important step towards large-scale relational learning.

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)