Logistic Tensor Factorization for Multi-Relational Data

Maximilian Nickel NICKEL@DBS.IFI.LMU.DE
Ludwig Maximilian University, Oettingenstr. 67, Munich, Germany

Volker Tresp VOLKER.TRESP@SIEMENS.COM
Siemens AG, Corporate Technology Otto-Hahn-Ring 6, Munich, Germany

Abstract

Tensor factorizations have become increasingly popular approaches for various learning tasks on structured data. In this work, we extend the RESCAL tensor factorization, which has shown state-of-the-art results for multi-relational learning, to account for the binary nature of adjacency tensors. We study the improvements that can be gained via this approach on various benchmark datasets and show that the logistic extension can improve the prediction results significantly.

1. Introduction

Tensor factorizations have become increasingly popular for learning on various forms of structured data such as large-scale knowledge bases, time-varying networks or recommendation data (Nickel et al., 2012; Bordes et al., 2011; Bader et al., 2007; Rendle et al., 2010). The success of tensor methods in these fields is strongly related to their ability to efficiently model, analyze and predict data with multiple modalities. Due to their multilinear nature, tensor models overcome limitations of linear models, such as their limited expressiveness, but at the same time remain more scalable and easier to handle then general non-linear approaches.

RESCAL (Nickel et al., 2011; 2012) is a tensor factorization for dyadic multi-relational data which has been shown to achieve state-of-the-art results for various relational learning tasks such as link prediction, entity resolution or link-based clustering. Briefly, the RESCAL model can be summarized as following: For relational data with K different dyadic relations and N entities, a third-order *adjacency tensor* $\mathcal{X}$ of size $N \times N \times K$ is created, where

$$x_{ijk} = \begin{cases} 1, & \text{if } Rel_k(Entity_i, Entity_j) \text{ is true} \\ 0, & \text{otherwise.} \end{cases}$$

This adjacency tensor $\mathcal{X}$ is then factorized into latent representations of entities and relations, such that

$$X_k \approx A R_k A^T$$

where X_k is the k-th frontal slice of $\mathcal{X}$. After computing the factorization, the matrix $A \in \mathbb{R}^{N \times r}$ then holds the latent representations for the entities in the data, i.e. the row $\boldsymbol{a}_i$ holds the latent representation of the i-th entity. Furthermore, $R_k \in \mathbb{R}^{r \times r}$ can be regarded as the latent representation of the k-th predicate, whose entries encode how the latent components interact for a specific relation. Since R_k is a *full, asymmetric* matrix, the factorization can also handle directed relations. When learning the latent representation of an entity, unique global representation allows the model to efficiently access information that is more distant in the relational graph via information propagation through the latent variables. For instance, it has been shown that RESCAL can propagate information about party membership of presidents and vice presidents over multiple relations, such that the correct latent representations are learned even when the party membership is unknown (Nickel et al., 2011). Moreover, since the entries of $\mathcal{X}$ are mutually independent given the latent factors A and $\mathcal{R}_k$, prediction is very fast, as it reduces to simple vector-matrix-vector products.

In its original form, the RESCAL factorization is computed by minimizing the least-squares error between the observed and the predicted entries; in a probabilistic interpretation this implies that the random variation of the data follows a Gaussian distribution, i.e. that

$$x_{ijk} \sim \mathcal{N}(\theta, \sigma^2)$$

ICML workshop on Structured Learning: Inferring Graphs from Structured and Unstructured Inputs (SLG 2013).

where θ are the parameters of the factorization. However, a Bernoulli is more appropriate for binary variables with

$$x_{ijk} \sim Bernoulli(\theta)$$

where the parameter θ is again computed via the factorization of the corresponding adjacency tensor. In the following, we will present a learning algorithm based on logistic regression[1] using the Bernoulli likelihood model and evaluate on benchmark data what gains can be expected from this updated model on relational data.

2. Methods

In the following, we interpret RESCAL from a probabilistic point of view. Each entry x_{ijk} in $\mathcal{X}$ is regarded as a random variable and we seek to compute the MAP estimates of A and $\mathcal{R}$ for the joint distribution

$$p(\mathcal{X}|A, \mathcal{R}) = \prod_{ijk} p(x_{ijk}|\boldsymbol{a}_i^T R_k \boldsymbol{a}_j). \tag{1}$$

Figure 1 also shows the graphical model in plate notation for the factorization. We will also fix the prior distributions of the latent factors to the Normal distribution, i.e. we set

$$\boldsymbol{a}_i \sim \mathcal{N}(0, \lambda_A I)$$
$$R_k \sim \mathcal{N}(0, \lambda_R I)$$

Furthermore, we will maximize the log-likelihood of Equation 1, such that the general form of the objective function that we seek to optimize is

$$\arg\min_{A,\mathcal{R}} \text{loss}(X; A, \mathcal{R}) + \lambda_A \|A\|_F^2 + \sum_k \lambda_R \|R_k\|_F^2 \tag{2}$$

The nature of the loss function depends on the distribution that we assume for x_{ijk}. In the following we consider the least-squares and the logistic loss function.

2.1. Least-Squares Regression

In its original form, RESCAL sets the loss function to

$$\text{loss}(X; A, \mathcal{R}) := \sum_k \|X_k - AR_kA^T\|_F^2. \tag{3}$$

In this case, Equation 2 and Equation 3 maximize the log-likelihood of Equation 1 when

$$x_{ijk} \sim \mathcal{N}(\boldsymbol{a}_i^T R_k \boldsymbol{a}_j, \sigma^2)$$

[1]In the theory of the exponential family, the logistic function describes the inverse parameter mapping for the Bernoulli distribution

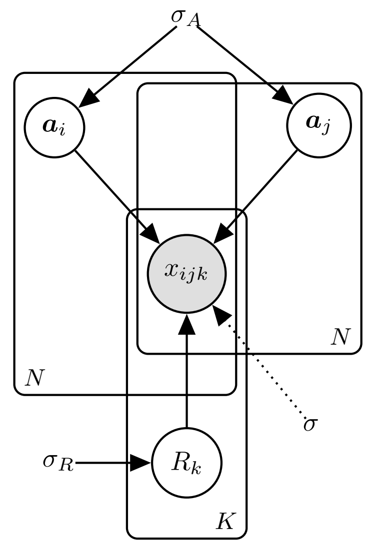

Figure 1. Graphical model in plate notation of the RESCAL factorization. The parameter σ is only present when the random variable x_{ijk} follows a Normal distribution.

It should be noted that although the least-squares error does not imply the correct error model, it has the appealing property that it enables a very efficient and scalable implementation. An algorithm based on alternating least-squares updates of the factor matrices, has been shown to scale up to large knowledge bases via exploiting the sparsity of relational data. For instance, it has been used to factorize YAGO, an ontology which consists of around 3 million entities, 40 relations, and 70 million known facts on a single desktop computer (Nickel et al., 2012). In the following we will refer to this implementation as RESCAL-ALS.

2.2. Logistic Regression

To describe the random variation in the data via a Bernoulli distribution, we set

$$\text{loss}(\mathcal{X}; A, \mathcal{R}) := -\sum_{ijk} x_{ijk} \log \sigma(\theta_{ijk}) + (1 - x_{ijk}) \log (1 - \sigma(\theta_{ijk})) \tag{4}$$

where

$$\sigma(\theta_{ijk}) = \frac{1}{1 + \exp(-\boldsymbol{a}_i^T R_k \boldsymbol{a}_j)}$$

Now, Equation 2 and Equation 3 maximize the log-likelihood of Equation 1 when

$$x_{ijk} \sim Bernoulli(\sigma(\theta_{ijk}))$$

Since, there exists no closed form solution to compute Equation 4, we use a gradient based approach to

Table 1. Evaluation results of the area under the precision-recall curve on the Kinships, Nations, Presidents, and Bacteriome datasets.

	Kinships	Nations	Pres.	Bact.
RESCAL-ALS	0.966	0.848	0.805	0.927
RESCAL-Logit	**0.981**	0.851	0.800	**0.938**
MLN	0.85	0.75	-	-
IRM	0.66	0.75	-	-

compute Equation 4 via quasi-Newton optimization, i.e. via the L-BFGS algorithm. The partial gradients for A and R_k are

$$\frac{\partial}{\partial A} = \sum_k \left[\sigma(AR_kA^T) - X_k)\right] AR_k^T + \left[\sigma(AR_kA^T) - X_k)\right]^T AR_k + 2\lambda_A A$$

$$\frac{\partial}{\partial R_k} = A^T \left[\sigma(AR_kA^T) - X_k\right] A + 2\lambda_R R_k$$

where $\sigma(AR_kA^T)$ denotes the elementwise application of $\sigma(\cdot)$ to AR_kA^T. Unfortunately, terms of the form

$$\left[\sigma\left(AR_kA^T\right) - X_k\right] A$$

can not be reduced to a significantly simpler form, due to the logistic function. Hence, this approach currently requires to compute the dense matrix AR_kA^T, what limits its scalability compared to the alternating least-squares approach. In the following, we will refer to this approach as RESCAL-Logit

3. Experiments

To evaluate the logistic extension of RESCAL, we conducted link-prediction experiments on the following datasets:

Presidents Multi-relational data, consisting of presidents of the United States, their vice-presidents as well as the parties of presidents and vice presidents.

Kinships Multi-relational data, consisting of several kinship relations within the Alwayarra tribe.

Nations Multi-relational, data consisting of relations between nations such as treaties, military actions, immigration etc.

Bacteriome Uni-Relational data, consisting of protein-protein and functional interactions within the context of an E. coli knowledgebase

For all datasets we performed 10-fold cross-validation and evaluated the results using the area under the precision-recall curve. In case of the presidents data, the task was to predict the party membership for presidents only based on the party memberships of their vice-presidents (and vice-versa). For all other datasets, cross-validation has been applied over all existing relations. It can be seen from the results in Table 1 that the logistic extension of RESCAL can considerably improve the prediction results. Especially the improvements for Kinships and Bacteriome are noteworthy, considering the already very good results of RESCAL-ALS.

4. Conclusion

To improve the modeling of multi-relational data, we have presented an extension for RESCAL based on logistic regression. We have shown on several benchmark datasets that the logistic extension can improve the prediction results significantly. While the evaluation results are very encouraging, future work will have to address the scalability of the presented approach, as the scalability of its current implementation is too limited for practical use on larger datasets.

NOTE ADDED IN PROOF

Independently, a similar logistic extension of the RESCAL factorization has been proposed in (London et al., 2013).

References

Bader, B. W., Harshman, R. A., and Kolda, T. G. Temporal analysis of semantic graphs using ASALSAN. In *7th IEEE International Conference on Data Mining*, pp. 33—42, 2007.

Bordes, A., Weston, J., Collobert, R., and Bengio, Y. Learning structured embeddings of knowledge bases. In *Proc. of the 25th Conference on Artificial Intelligence*, 2011.

London, B., Rekatsinas, T., Huang, B., and Getoor, L. Multi-relational learning using weighted tensor decomposition with modular loss. *arXiv preprint arXiv:1303.1733*, 2013.

Nickel, M., Tresp, V., and Kriegel, H.-P. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 809—816, 2011.

Nickel, M., Tresp, V., and Kriegel, H.-P. Factorizing YAGO: scalable machine learning for linked data. In *Proc. of the 21st Int. World Wide Web Conference*, 2012.

Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. Factorizing personalized markov chains for next-basket recommendation. In *Proc. of the 19th Int. World Wide Web Conference*, pp. 811—820, 2010.