A Three-Way Model for Collective Learning on Multi-Relational Data

28th International Conference on Machine Learning

Maximilian Nickel[1] Volker Tresp[2] Hans-Peter Kriegel[1]

[1]Ludwig-Maximilians Universität, Munich

[2]Siemens AG, Corporate Technology, Munich

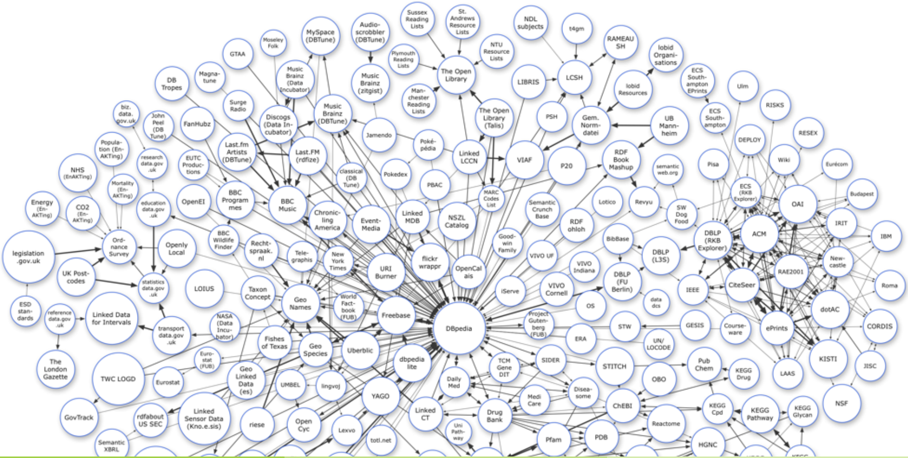
June 30th, 2011

Outline

Introduction

Multi-Relational Data

- Multi-relational data is a part of many different important fields of application, such as *Computational Biology, Social Networks*, the *Semantic Web*, the *Linked Data* cloud (shown below) and many more



Introduction

Motivation to use Tensors for Relational Learning

Why Tensors?

- **Modelling simplicity**: Multiple binary relations can be expressed straightforwardly as a three-way tensor
- **No structure learning**: Not necessary to have information about independent variables, knowledge bases, etc. or to infer it from data
- **Expected performance**: Relational domains are high-dimensional and sparse, a setting where factorization methods have shown very good results

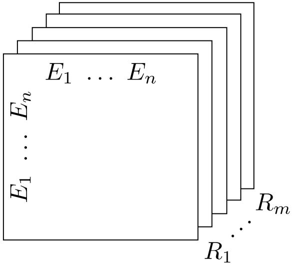
Problem: Tensor factorizations like CANDECOMP/PARAFAC (CP) or Tucker can not perform collective learning or in the case of DEDICOM have unreasonable constraints for relational learning.

(For an excellent review on tensors see (Kolda and Bader, 2009))

Introduction

Modelling and Terminology

- Modelling binary relations as a tensor: Two modes of a tensor refer to the entities, one mode to the relations.
- The entries of the tensor are 1 when a relation between two entities exists and 0 otherwise
- We use the RDF formalism to model relations as `(subject, predicate, object)` triples



Outline

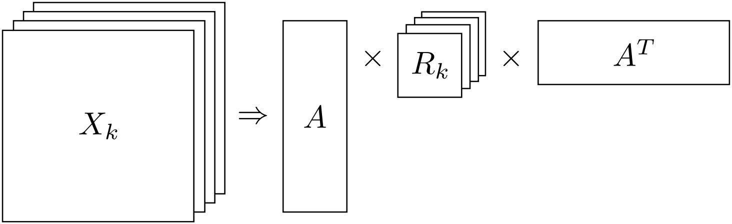
RESCAL

Tensor Factorizaion

- RESCAL takes the inherent structure of dyadic relational data into account, by employing the tensor factorization

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

- A is a $n \times r$ matrix, representing the *global* entity-latent-component space
- R_k is an *asymmetric* $r \times r$ matrix that specifies the interaction of the latent components per predicate



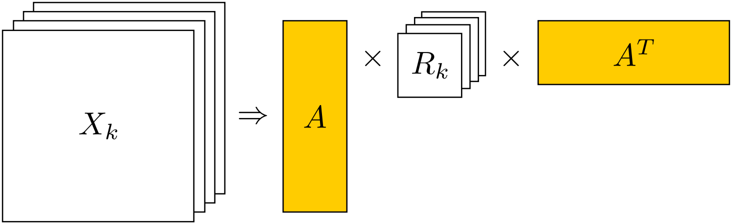
RESCAL

Tensor Factorizaion

- RESCAL takes the inherent structure of dyadic relational data into account, by employing the tensor factorization

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

- A is a $n \times r$ matrix, representing the *global* entity-latent-component space
- R_k is an *asymmetric* $r \times r$ matrix that specifies the interaction of the latent components per predicate



RESCAL

Solving canonical relational learning tasks

- **Link Prediction**: To predict the existence of a relation between two entities, it is sufficient to look at the rank-reduced reconstruction of the appropriate slice AR_kA^T
- **Collective Classification**: Can be cast as a link prediction problem by including the classes as entities and adding a `classOf` relation. Alternatively, standard classification algorithms could be applied to the entites' latent-component representation A
- **Link-based Clustering**: Since the entities latent-component representation is computed considering all relations, Link-based clustering can be done by clustering the entities in the latent-component space A

RESCAL

Computing the factorization

- To compute the factorization, we solve the optimization problem

$$\min_{A,R_k} loss(A, R_k) + reg(A, R_k)$$

where $loss$ is the loss function

$$loss(A, R_k) = \frac{1}{2} \sum_k \|\mathcal{X}_k - AR_kA^T\|_F^2$$

and reg is the regularization term

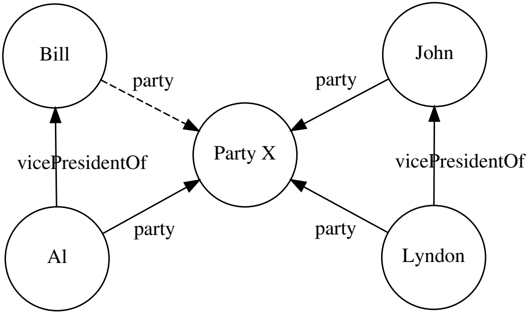
$$reg(A, R_k) = \frac{1}{2}\lambda \left(\|A\|_F^2 + \sum_k \|R_k\|_F^2 \right)$$

- Efficient alternating-least squares algorithm based on ASALSAN (Bader et al., 2007)

RESCAL

Collective Learning Example

- Predict party membership of US (vice) presidents



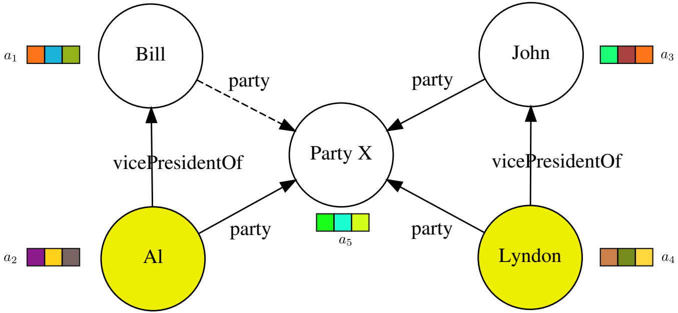
- Helpful to consider element-wise version of the loss function f

$$f(A, R_k) = \frac{1}{2} \sum_{i,j,k} \left(\mathcal{X}_{ijk} - \boldsymbol{a}_i^T R_k \boldsymbol{a}_j \right)^2$$

RESCAL

Collective Learning Example

- Predict party membership of US (vice) presidents



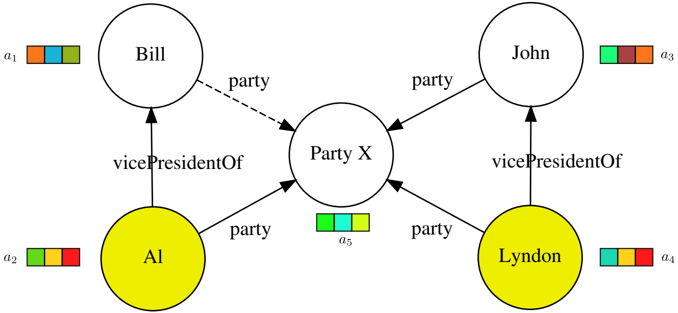
- Helpful to consider element-wise version of the loss function f

$$f(A, R_k) = \frac{1}{2} \sum_{i,j,k} \left(\mathcal{X}_{ijk} - \boldsymbol{a}_i^T R_k \boldsymbol{a}_j \right)^2$$

RESCAL

Collective Learning Example

- Predict party membership of US (vice) presidents



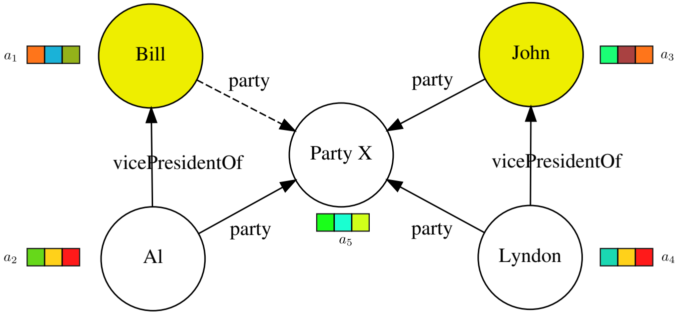
- Helpful to consider element-wise version of the loss function f

$$f(A, R_k) = \frac{1}{2} \sum_{i,j,k} \left(\mathcal{X}_{ijk} - \boldsymbol{a}_i^T R_k \boldsymbol{a}_j \right)^2$$

RESCAL

Collective Learning Example

- Predict party membership of US (vice) presidents



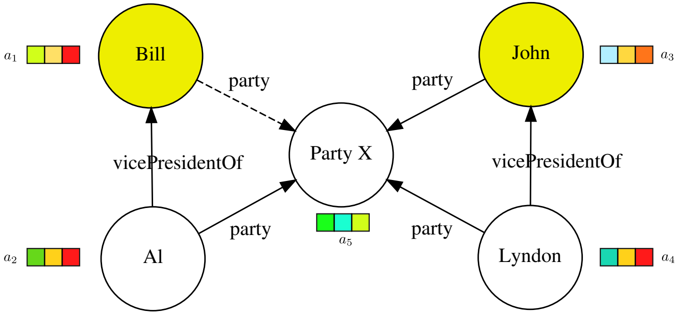
- Helpful to consider element-wise version of the loss function f

$$f(A, R_k) = \frac{1}{2} \sum_{i,j,k} \left(\mathcal{X}_{ijk} - \boldsymbol{a}_i^T R_k \boldsymbol{a}_j \right)^2$$

RESCAL

Collective Learning Example

- Predict party membership of US (vice) presidents



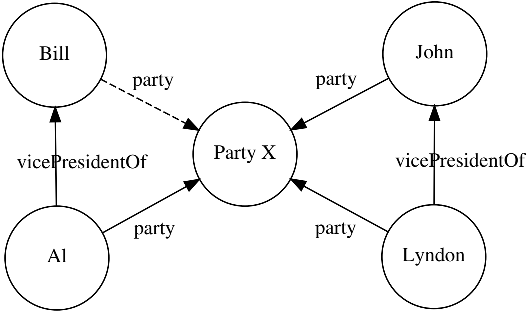
- Helpful to consider element-wise version of the loss function f

$$f(A, R_k) = \frac{1}{2} \sum_{i,j,k} \left(\mathcal{X}_{ijk} - \boldsymbol{a}_i^T R_k \boldsymbol{a}_j \right)^2$$

RESCAL

Collective Learning Example

- Predict party membership of US (vice) presidents



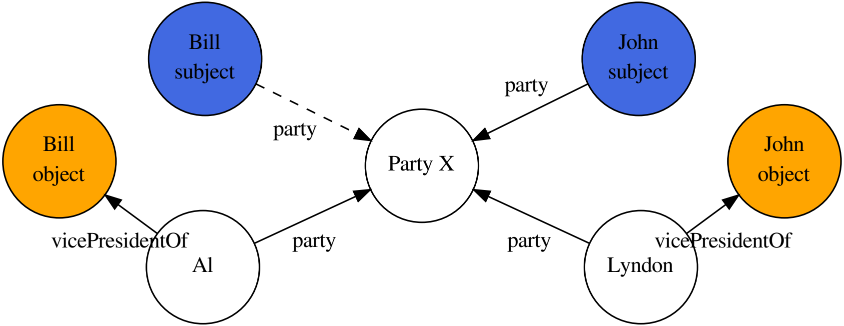
- Helpful to consider element-wise version of the loss function f

$$f(A, R_k) = \frac{1}{2} \sum_{i,j,k} \left(\mathcal{X}_{ijk} - \boldsymbol{a}_i^T R_k \boldsymbol{b}_j \right)^2$$

RESCAL

Collective Learning Example

- Predict party membership of US (vice) presidents



- Helpful to consider element-wise version of the loss function f

$$f(A, R_k) = \frac{1}{2} \sum_{i,j,k} \left(\mathcal{X}_{ijk} - \boldsymbol{a}_i^T R_k \boldsymbol{b}_j \right)^2$$

RESCAL

Collective Learning with RESCAL

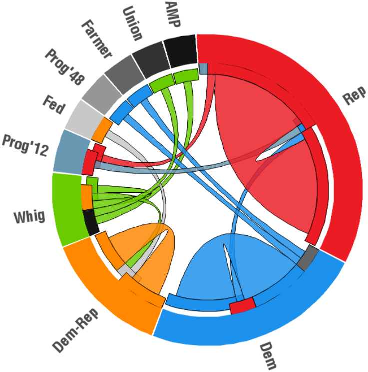
- Collective learning is performed via the entities' latent-component representation
- Important aspect of the model: Entities have a unique latent-component representation, regardless of their occurrence as subjects or objects

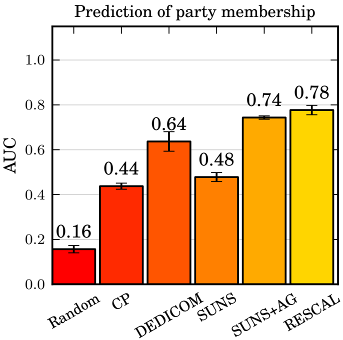
Outline

Experiments

Predicting the party membership of US (vice) presidents

- **Task**: Predict party membership of US (vice) presidents
- No other information included in the data other than the party membership and who is (vice) president of whom

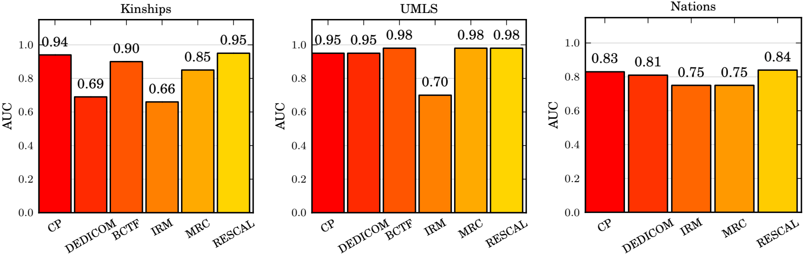




Experiments

Comparison to state-of-the-art approaches

- **Task**: Perform link prediction on the IRM datasets Kinships, UMLS and Nations
- Comparison to MRC (Kok & Domingos, 2007), IRM (Kemp et al., 2007) and BCTF (Sutskever et al., 2009) as well as CP and DEDICOM



Experiments

Runtime and Implementation

- RESCAL-ALS algorithm features very fast training times

Dataset	**Entities**	**Relations**	**Total Runtime** in seconds			
			Rank	10	20	40
Kinships	104	26		1.1	3.7	51.2
Nations	125	57		1.7	5.3	54.4
UMLS	135	49		2.6	4.9	72.3
Cora	2497	7		364	348	680

Table: Average runtime to compute a rank-r factorization in RESCAL

- Implementation uses only standard matrix operations

Outline

Summary

- RESCAL is an tensor-based relational learning approach capable of collective learning
- Collective learning mechanism works through information propagation via the entities' latent-component representations
- Good performance compared to current state-of-the-art relational learning approaches
- Fast training times and simple Implementation
- Code available at `http://www.cip.ifi.lmu.de/~nickel`

Thank you!