Machine Learning in Networked Data

Volker Tresp

Siemens Research and Technology Center
Ludwig Maximilian University of Munich

(with Maximilian Nickel, Xueyan Jiang, Yi Huang)

Big Data

- Why do we want to do Big Data?

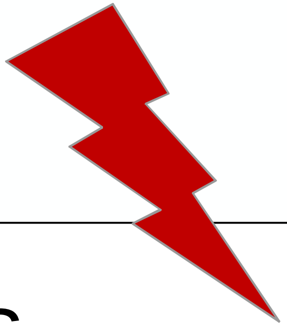
Big Data

- Because we can!

Big Data

- Why do we have to do Big Data?

Big Data



- Because something explodes

#data sources

#entitities, sensors, data generators

Consumers, users, customers,
Text documents
WWW
Log-Files generators
OMICS
Mobile devices
Images
Cars
Smart meters

Data complexity

#attributes

Detailed user / customer profiles
Genomics
Proteomics
…

#relationships

Social networks
Patient networks

Multipliers

time

location

- Let’s consider complexity

Patient in a Complex Environment with all Sorts of Networks

Patient Modell

A patient in multiple social and other networks with relationships to

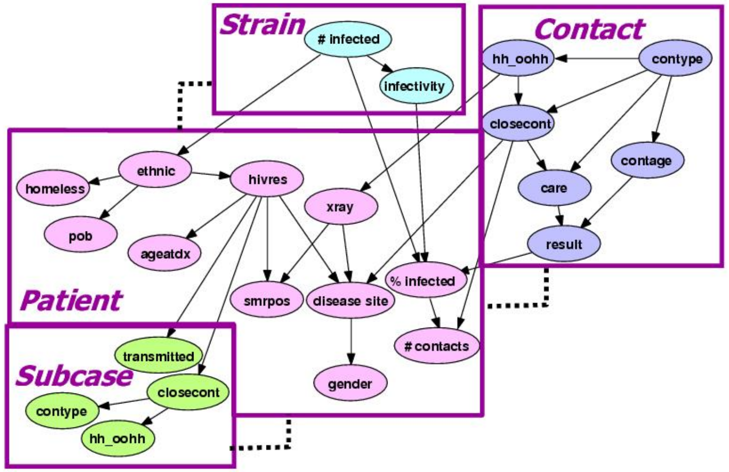
- Physicians
- Patient with similar complaints
- Orders, medications
- Diagnosis
- Treatments

Increasing relevance of –omics data

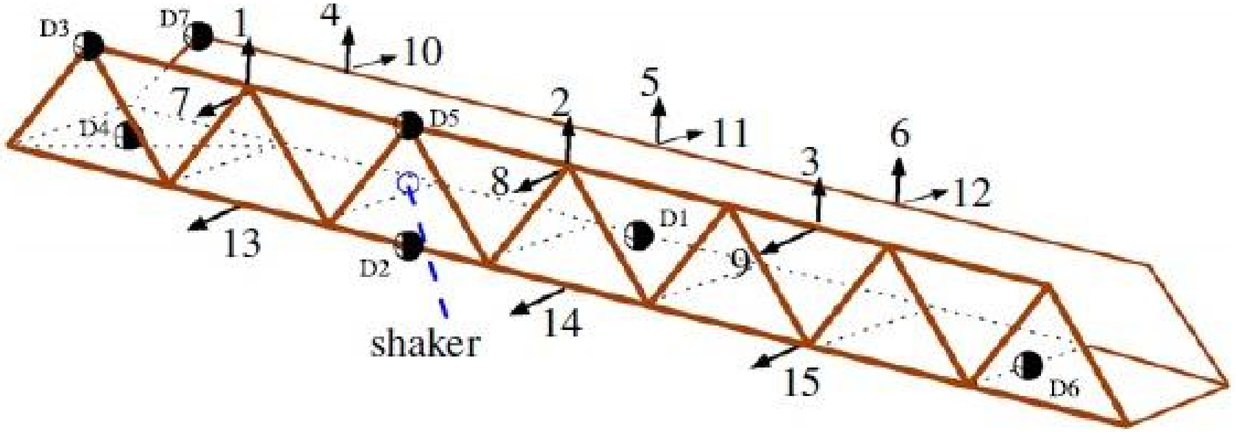
- genomics, proteomics, metabolomics, …

The new view

A patient in a clinic as a socal being with multiple complex relationships and attributes and part of severeral networks



Network of Sensors

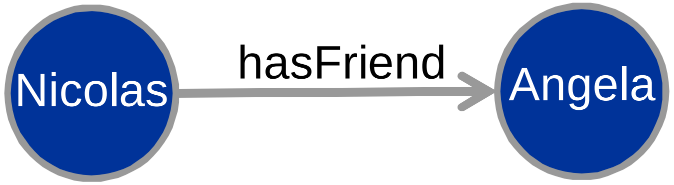


Let’s address complexity

- How to describe the complexity of the world?

Basic Information Element

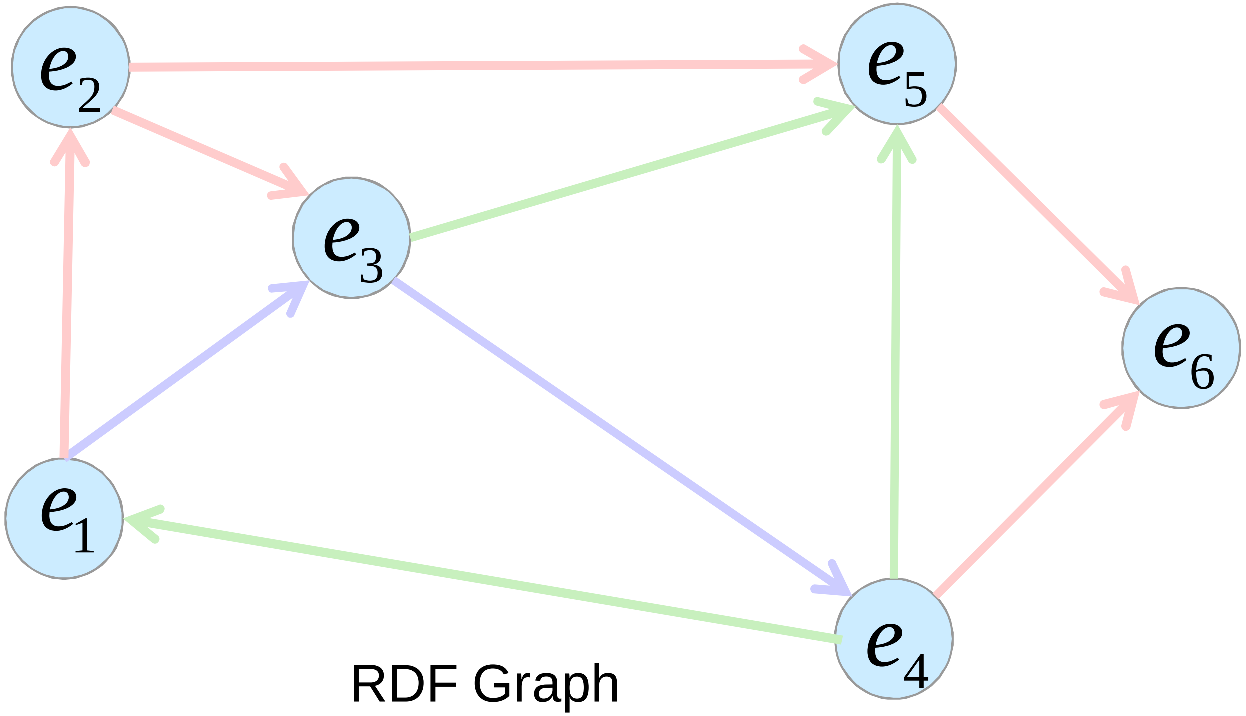
- A triples as a representation of a binary relation
- RDF triple (resource description framework)



Subject Predicate Object

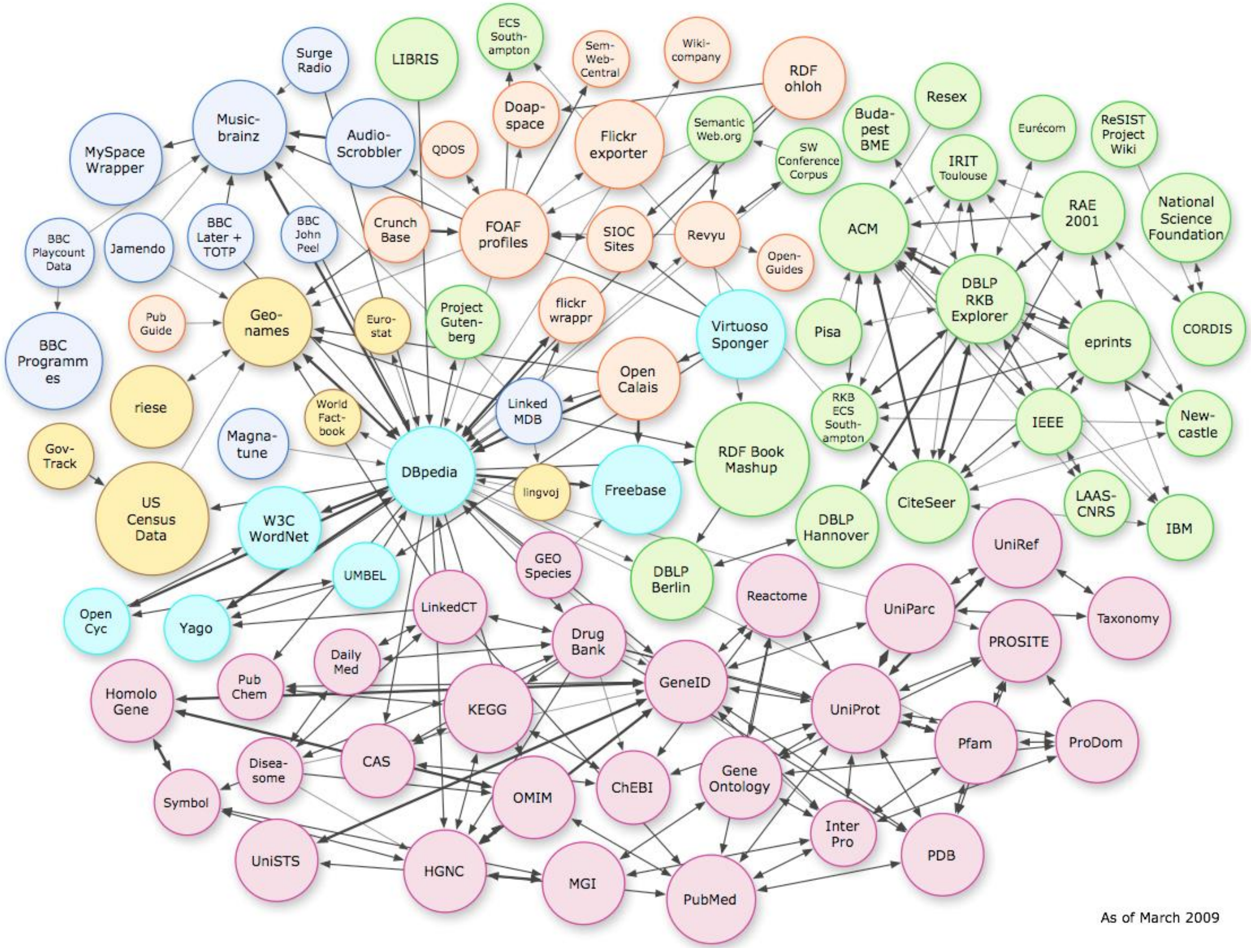
- *The world is just a bunch of triples*

The World is Just a Directed Labeled Graph



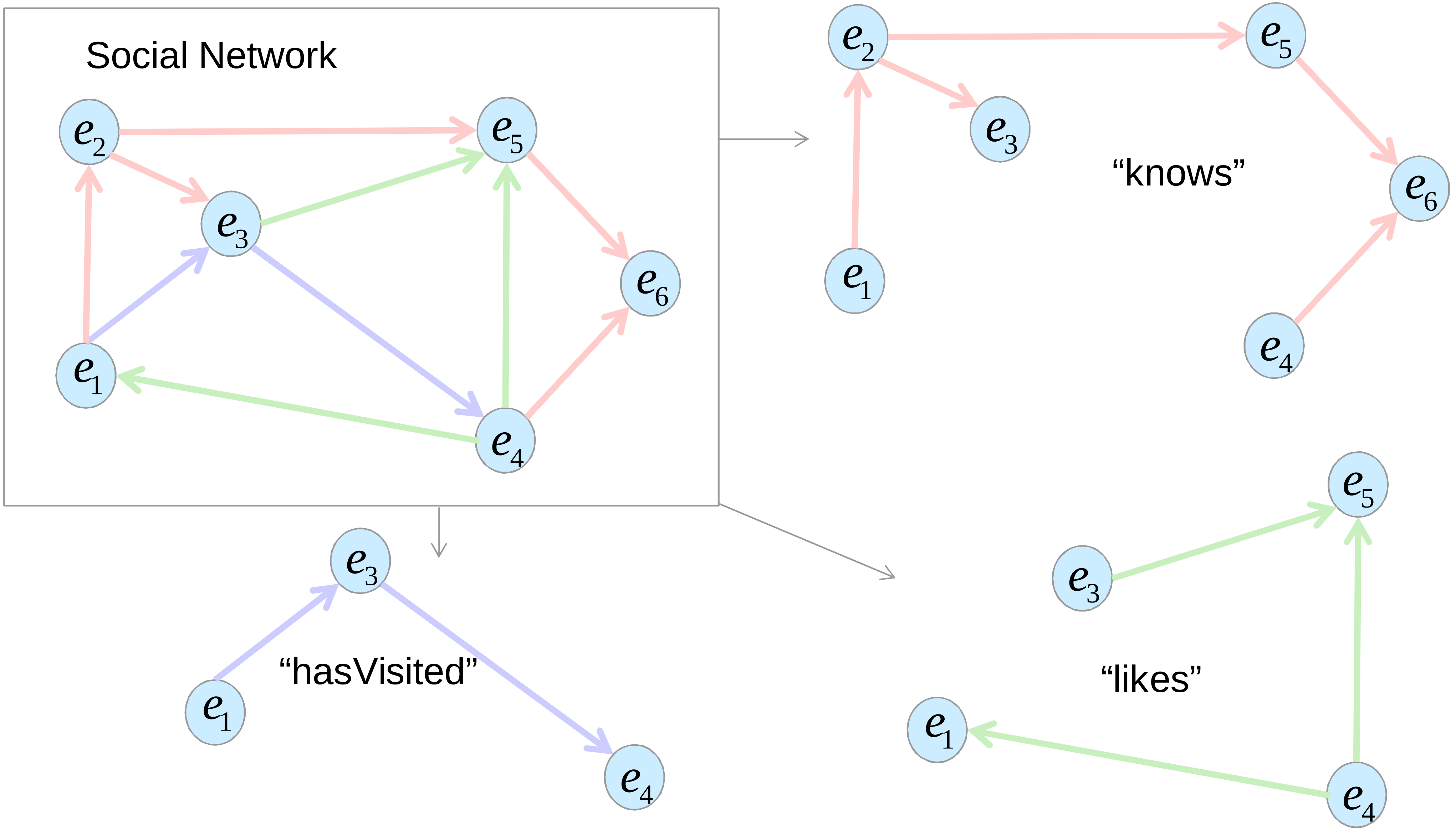
RDF Graph

Linked Open Data (LOD) und YAGO



As of March 2009

The World is Just a Bunch of Networks



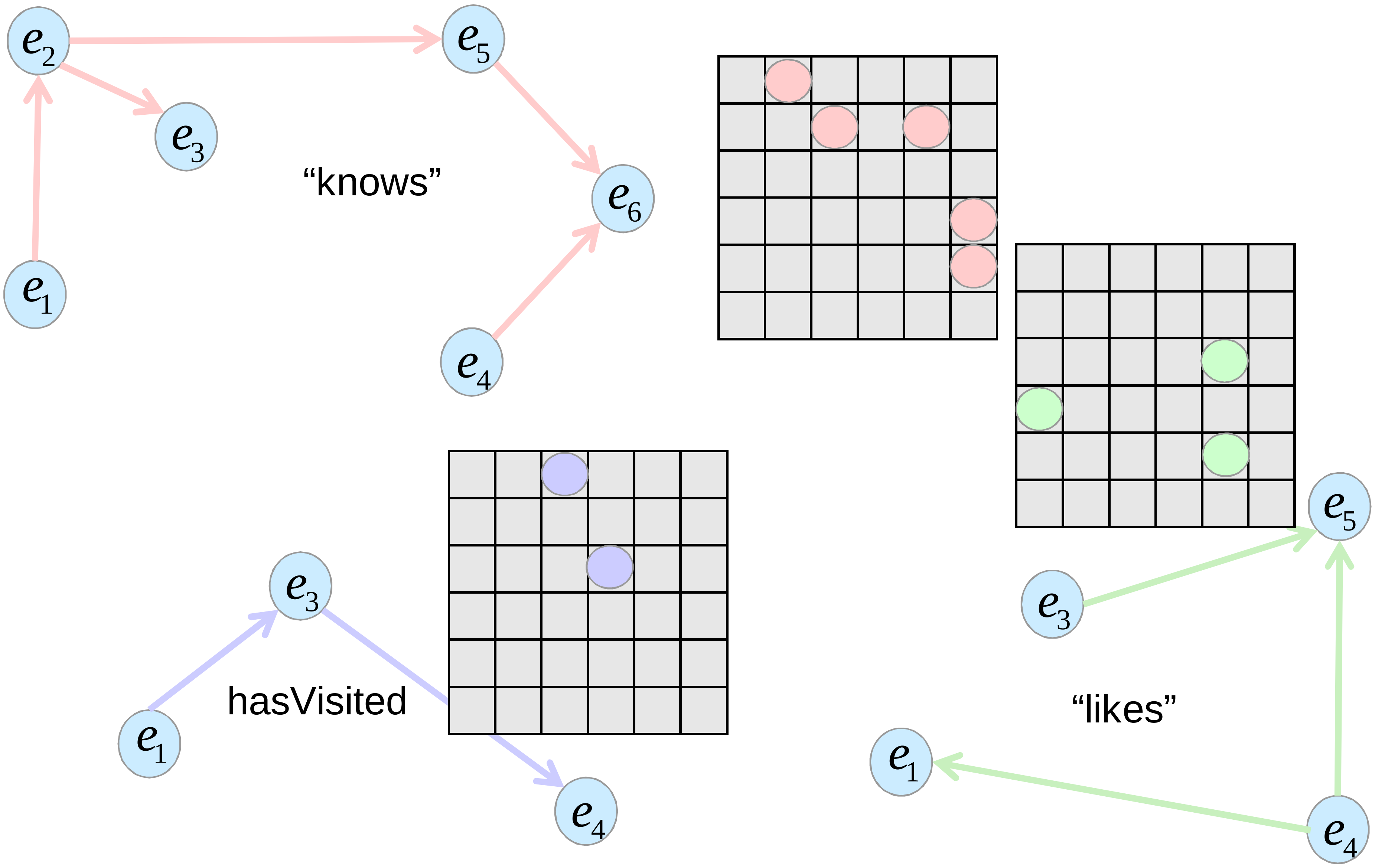
Social Network

“knows”

“hasVisited”

“likes”

The Associated Adjacency Matrices

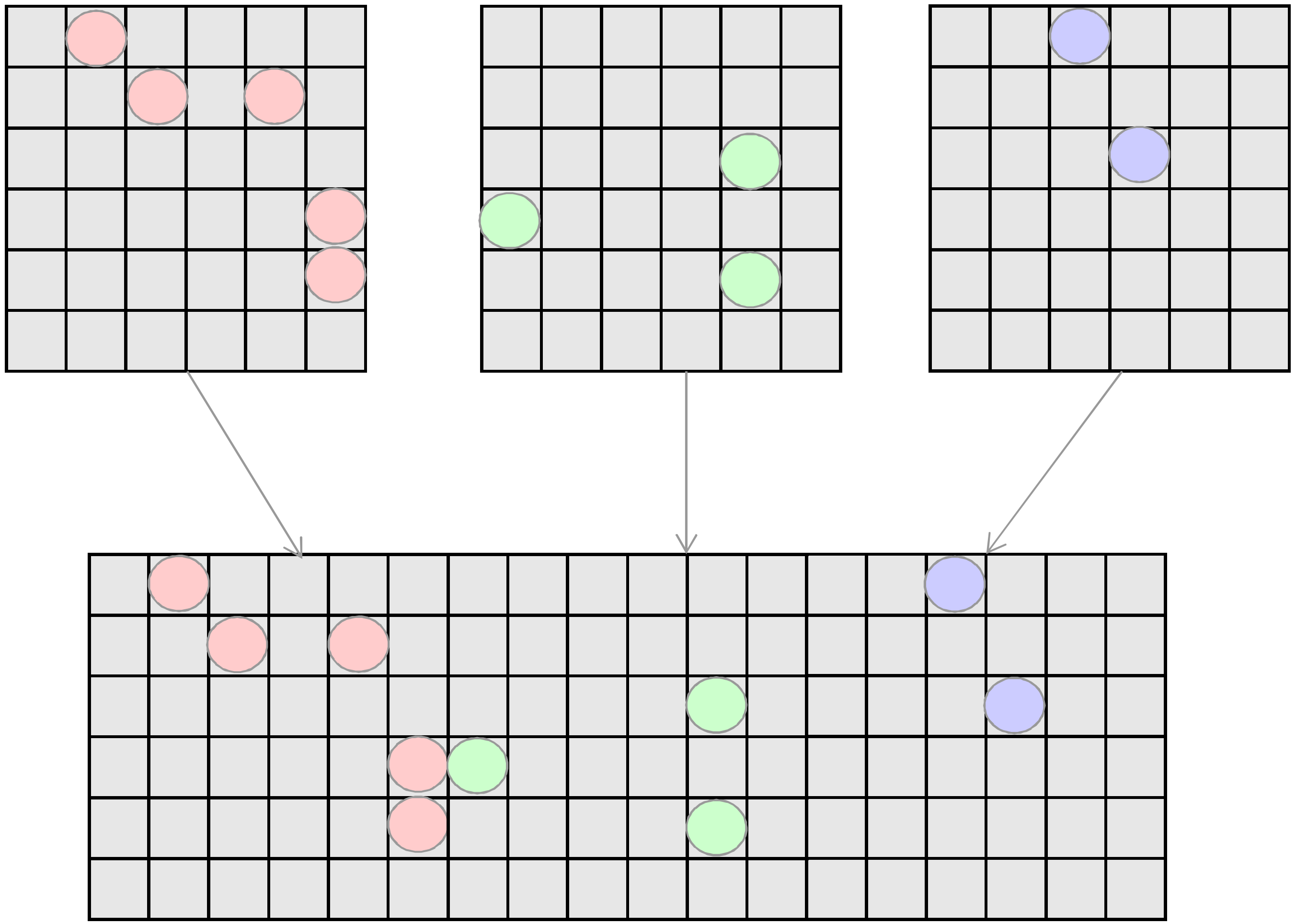


	●				
		●		●	
					●
					●

				●	
●					
				●	

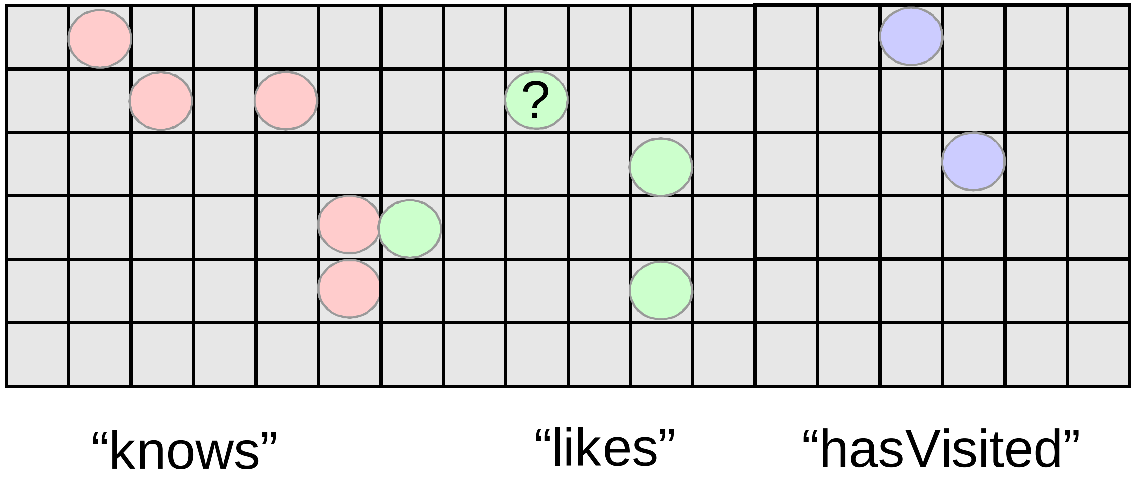
		●			
			●		

The World is Just a Big Adjacency Matrix



“knows” “likes” “hasVisited”

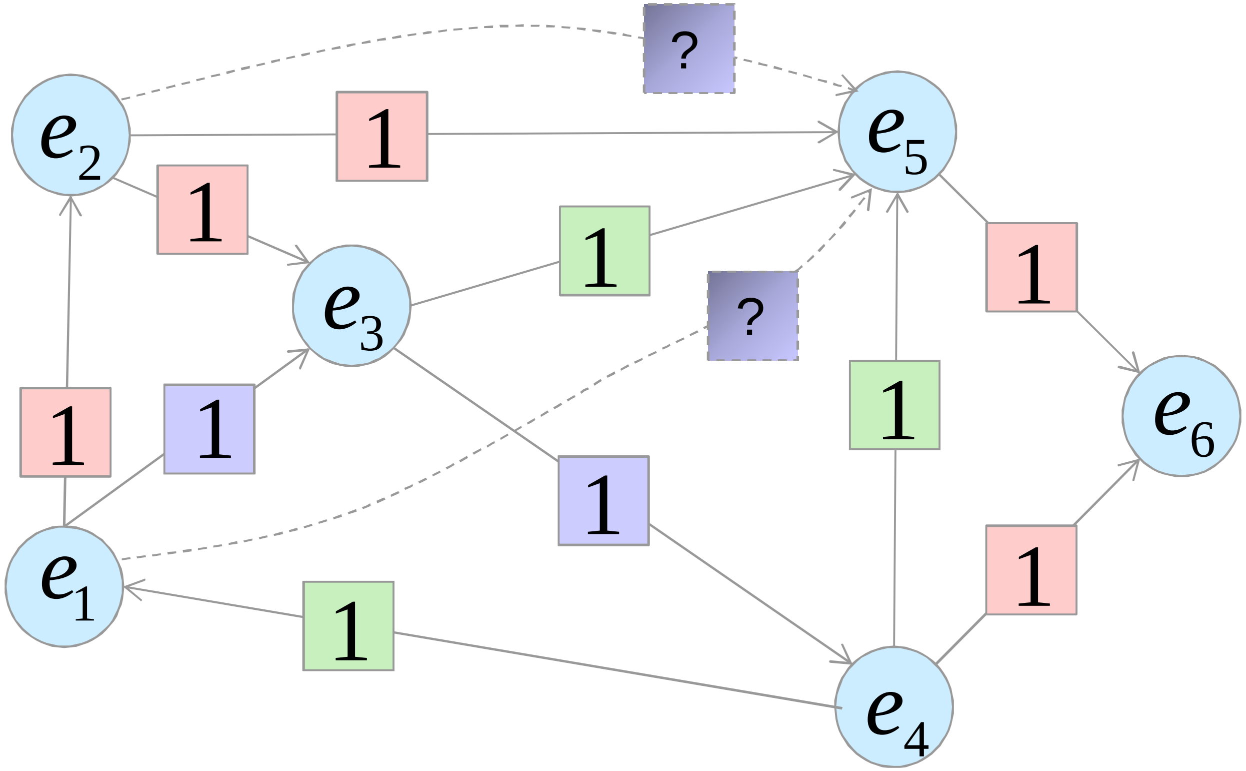
Why Machine Learning?



A lot of Machine Learning tasks can be reduced to the task of predicting the existence of triples

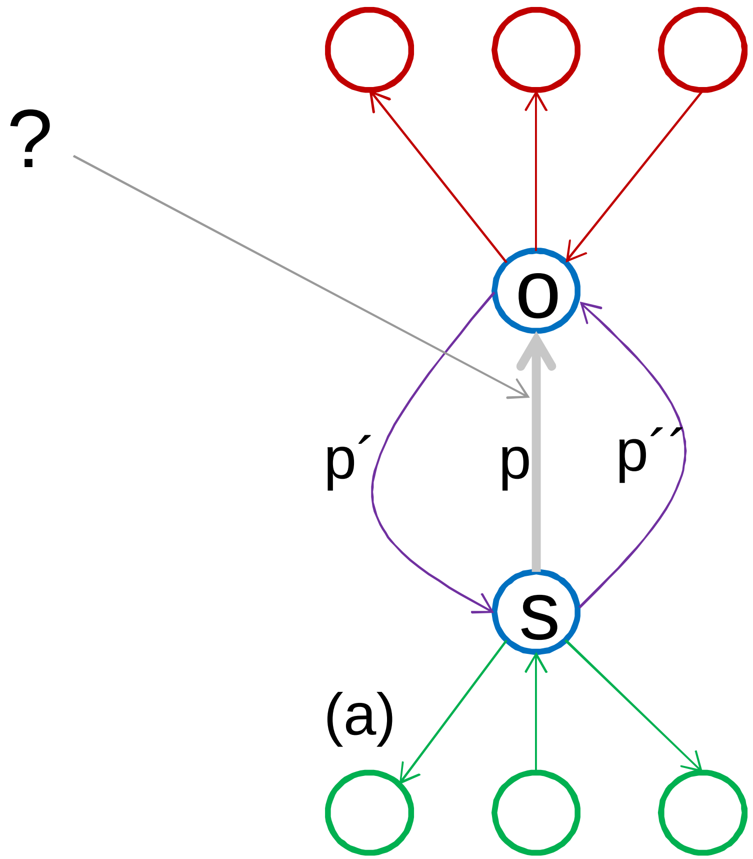
- Classification
- Attribute Prediction
- Relationship Prediction
- Clustering
- …

A Dual Graph of Random Variables



- Goal of machine learning: predict triples not known to be true (dashed links)
- We introduce a random variable for each possible link; the random variables then form a *dual graph* to the RDF graph

Predicting a Triple from Its Immediate Neighborhood



Machine Learning in Terms of Adjacency Matrices

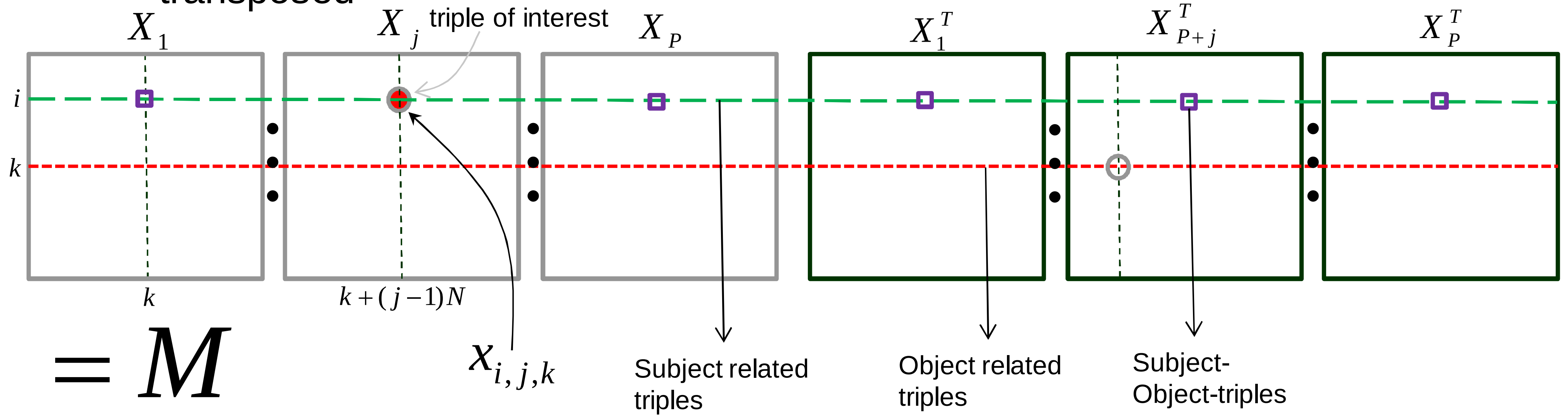
- If $(s=i, p=j, o=k)$ is known to be true

  then the variable $x_{i,j,k}=1$ otherwise $x_{i,j,k}=0$

- For each predicate *p=j* we form an adjacency matrix X_j

  with $(X_j)_{i,k} = x_{i,j,k}$

- We form the Matrix *M* by concatenating the adjacency matrices and their transposed



$= M$

A Model

$$\hat{x}_{i,j,k} = \boxed{\sum_{l=1}^{2PN} w_{l,k+(j-1)N}\ m_{i,l}} + \boxed{\sum_{l=1}^{2PN} r_{l,i+(j-1)N}\ m_{k,l}} + \boxed{\sum_{l=1}^{2P} h_{l,j}\ m_{i,k+N(l-1)}}$$

- *N*: number of entities *P*: number of predicates
- Here, $w_{l,k+(l-1)N}$ is the weight for predicting $\hat{x}_{i,j,k}$ from $m_{i,l}$
- Note that the weight is independent of subject *i:* implied exchangeability

$r_{l,i+(l-1)N}$ weights for object triples (independent of object *j*)

$h_{l,j}$ weights for subject-object triples (independent of both subject *i* and object *j*)

Weight Optimization (Overview)

- We use a least squares cost function
- We include weight regularizers to avoid overfitting (ridge regression)

$$\|X - \hat{X}\|_F^2 + \lambda_W \|W\|_F^2 + \lambda_R \|R\|_F^2 + \lambda_H \|H\|_F^2$$

- We first perform an SVD smoothing on the input representation (improves generalization and reduces the number of free parameters)
- Parameter optimization is performed efficiently using alternating least square

$$MW = U_r \,\mathrm{diag}\left\{ \frac{d_i^2}{d_i^2 + \lambda_W} \right\}_{i=1}^{r} U_r^T \hat{X}^{(-W)}$$

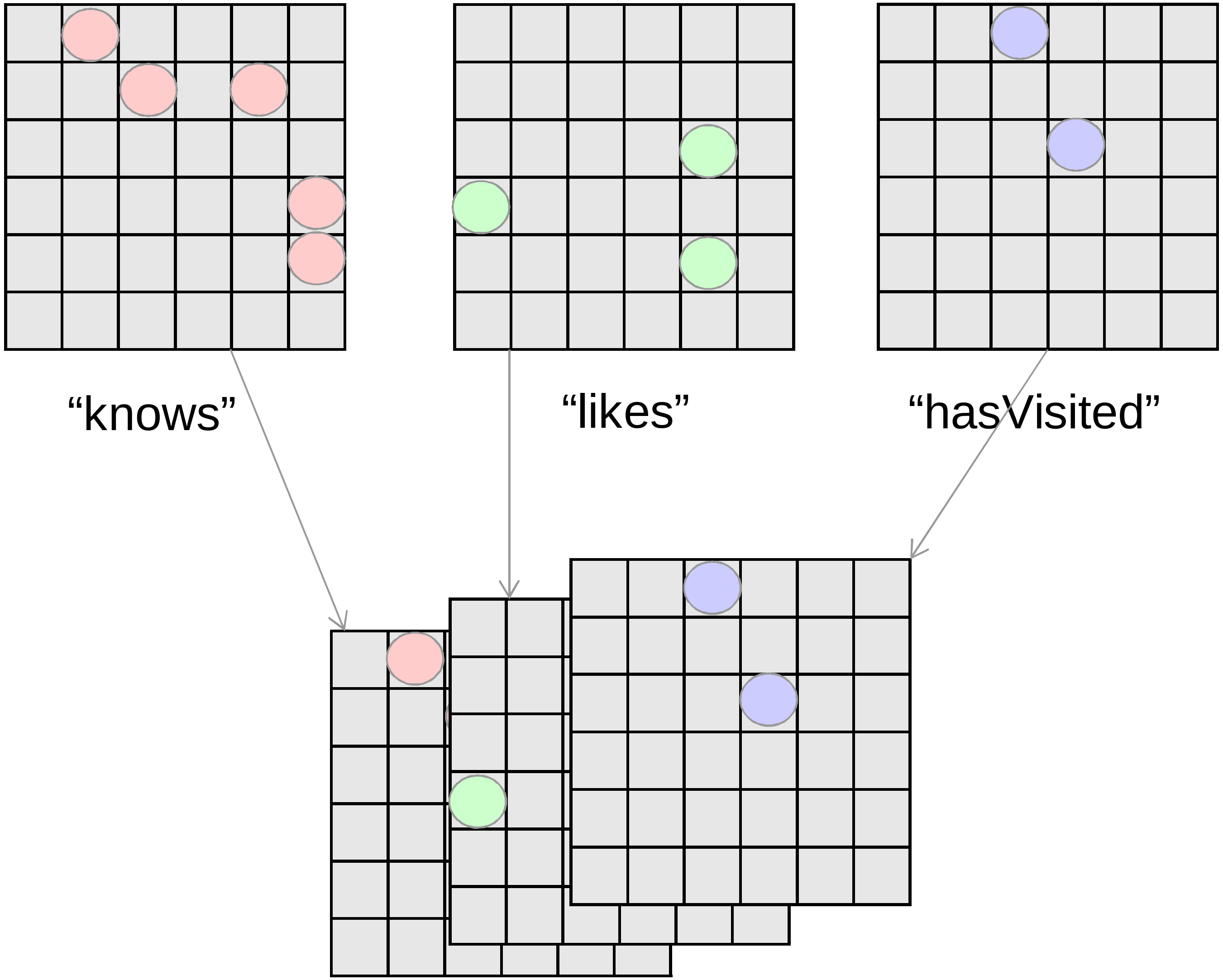
$$MR = U_r \,\mathrm{diag}\left\{ \frac{d_i^2}{d_i^2 + \lambda_R} \right\}_{i=1}^{r} U_r^T \left(\hat{X}^{(-R)} \right)^\dagger$$

$$\tilde{M}H = \tilde{U}_{\tilde{r}} \,\mathrm{diag}\left\{ \frac{\tilde{d}_i^2}{\tilde{d}_i^2 + \tilde{\lambda}_H} \right\}_{i=1}^{\tilde{r}} \tilde{U}_{\tilde{r}}^T \tilde{X}^{(-H)}$$

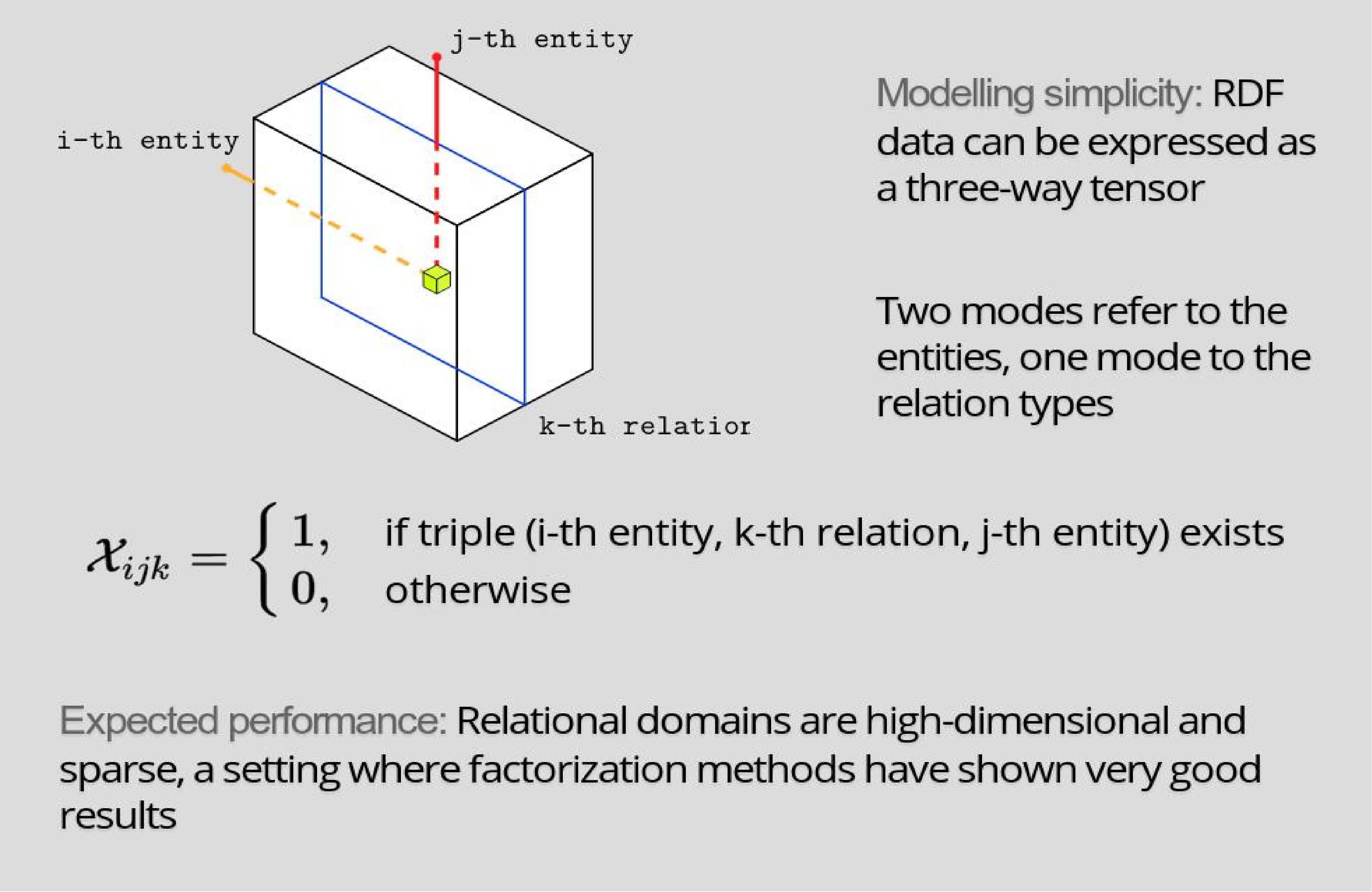
Sparse matrix algebra!!!

[Jiang et al., ISWC 2012]

Another Representation: The World as a Tensor



Modeling an RDF Triple Store as a Three-Way Tensor


j-th entity
i-th entity
k-th relatior

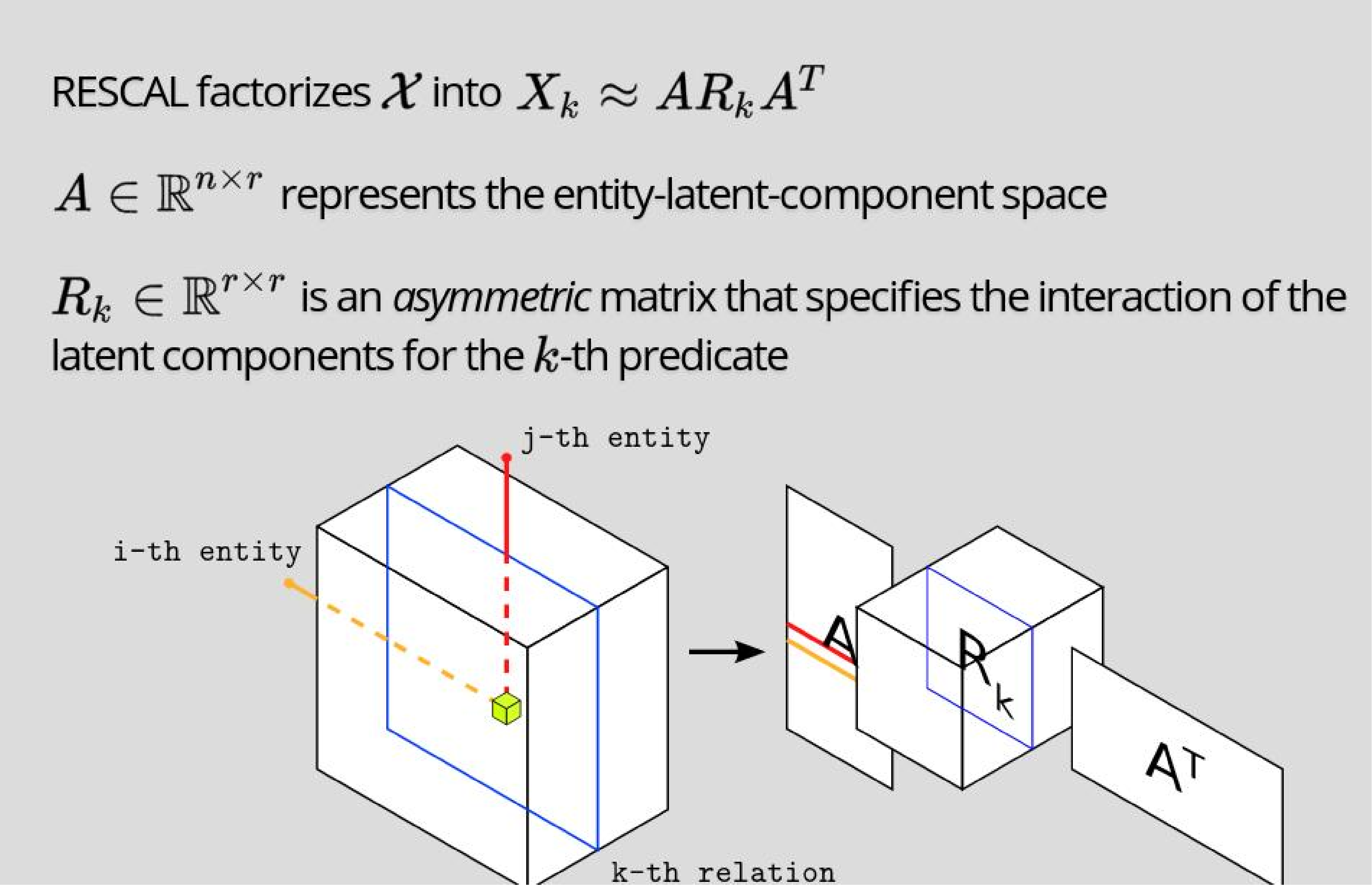
Modelling simplicity: RDF data can be expressed as a three-way tensor

Two modes refer to the entities, one mode to the relation types

$$\mathcal{X}_{ijk} = \begin{cases} 1, & \text{if triple (i-th entity, k-th relation, j-th entity) exists} \\ 0, & \text{otherwise} \end{cases}$$

Expected performance: Relational domains are high-dimensional and sparse, a setting where factorization methods have shown very good results

RESCAL Factorization



RESCAL factorizes $\mathcal{X}$ into $X_k \approx A R_k A^T$

$A \in \mathbb{R}^{n \times r}$ represents the entity-latent-component space

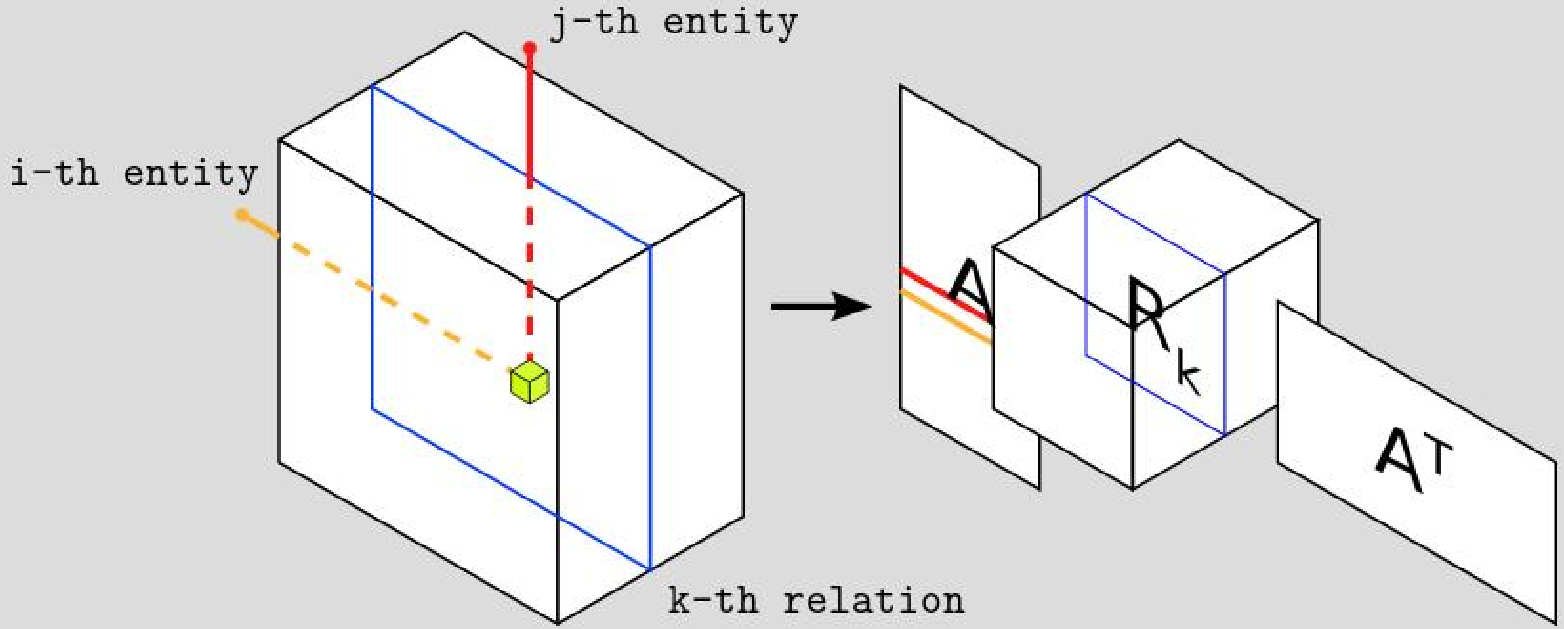
$R_k \in \mathbb{R}^{r \times r}$ is an *asymmetric* matrix that specifies the interaction of the latent components for the k-th predicate

Tasks

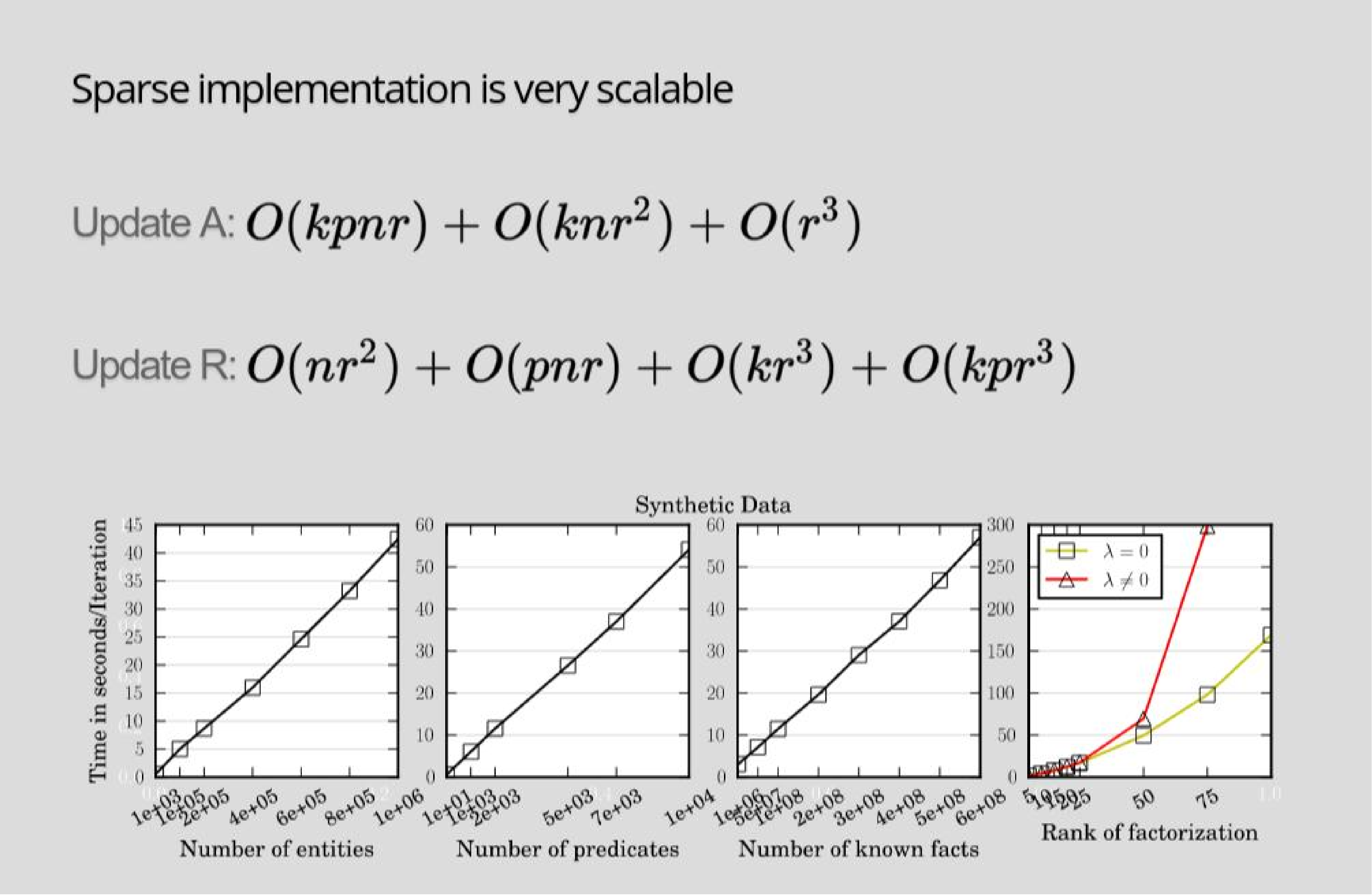
Link-Prediction: Rank entries in reconstructed tensor by their values

Collective Classification: Cast as link-prediction or classify entities on A

Entity Resolution: Exploit similarity of entities on A



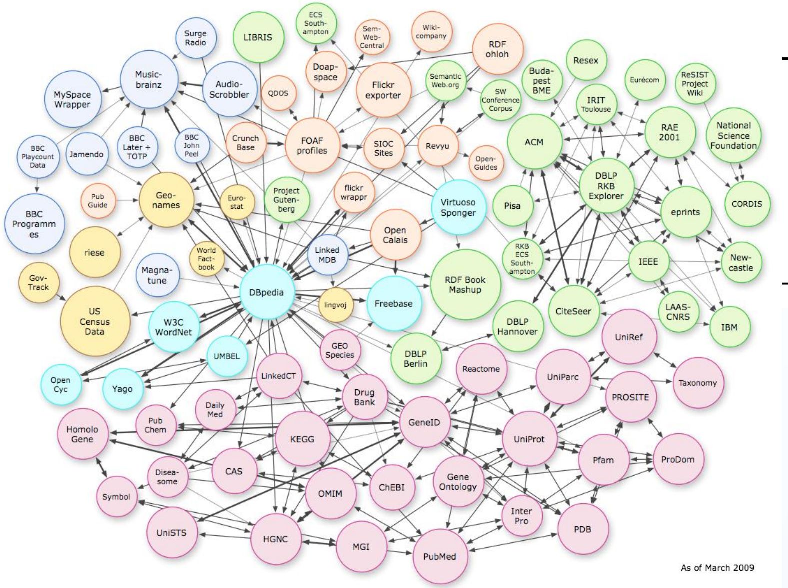
Scalability



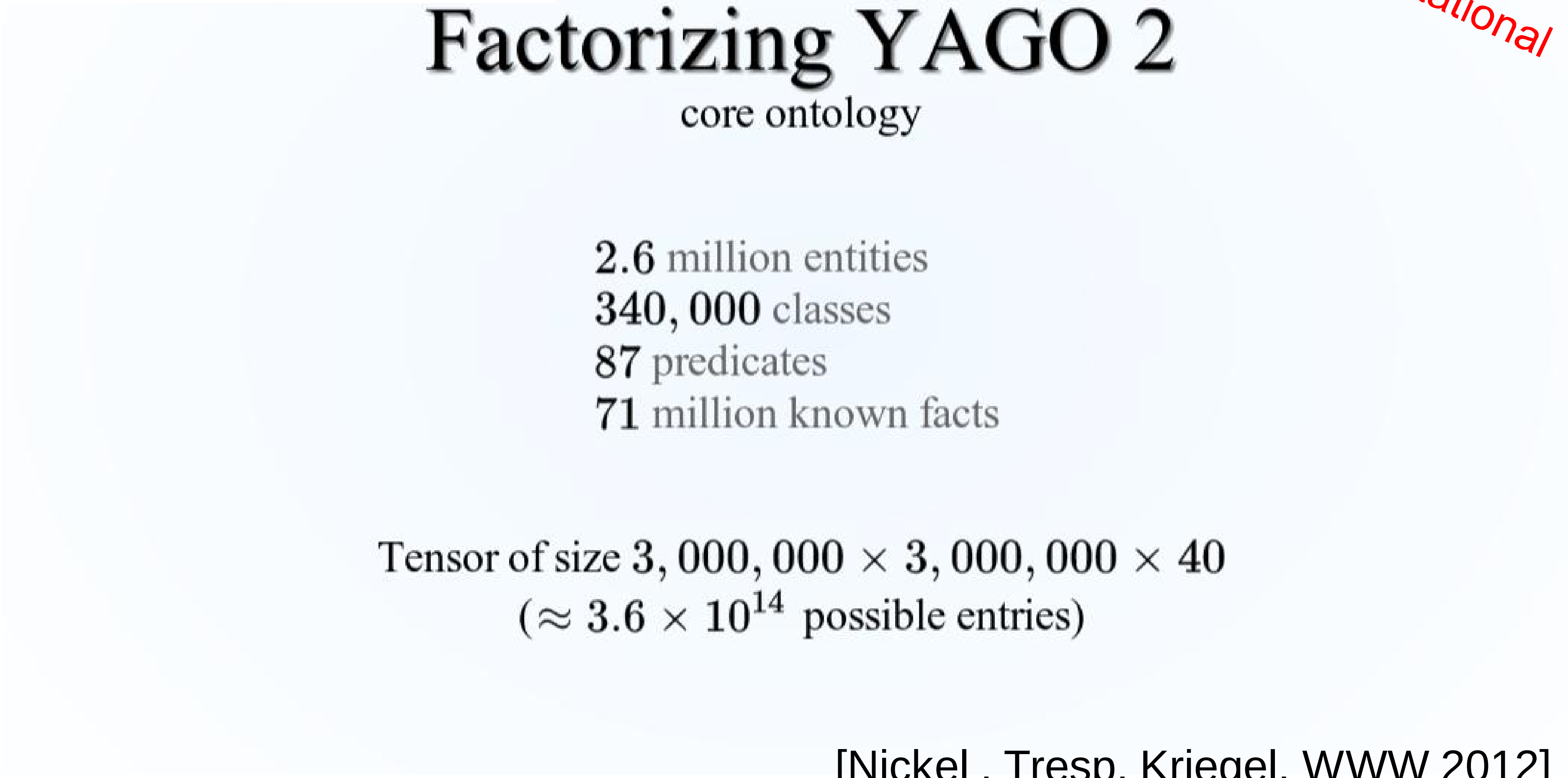
Sparse implementation is very scalable

Update A: $O(kpnr) + O(knr^2) + O(r^3)$

Update R: $O(nr^2) + O(pnr) + O(kr^3) + O(kpr^3)$



Prediction of ALL 3.6x10^14 triples in one computational step!!!!



Factorizing YAGO 2

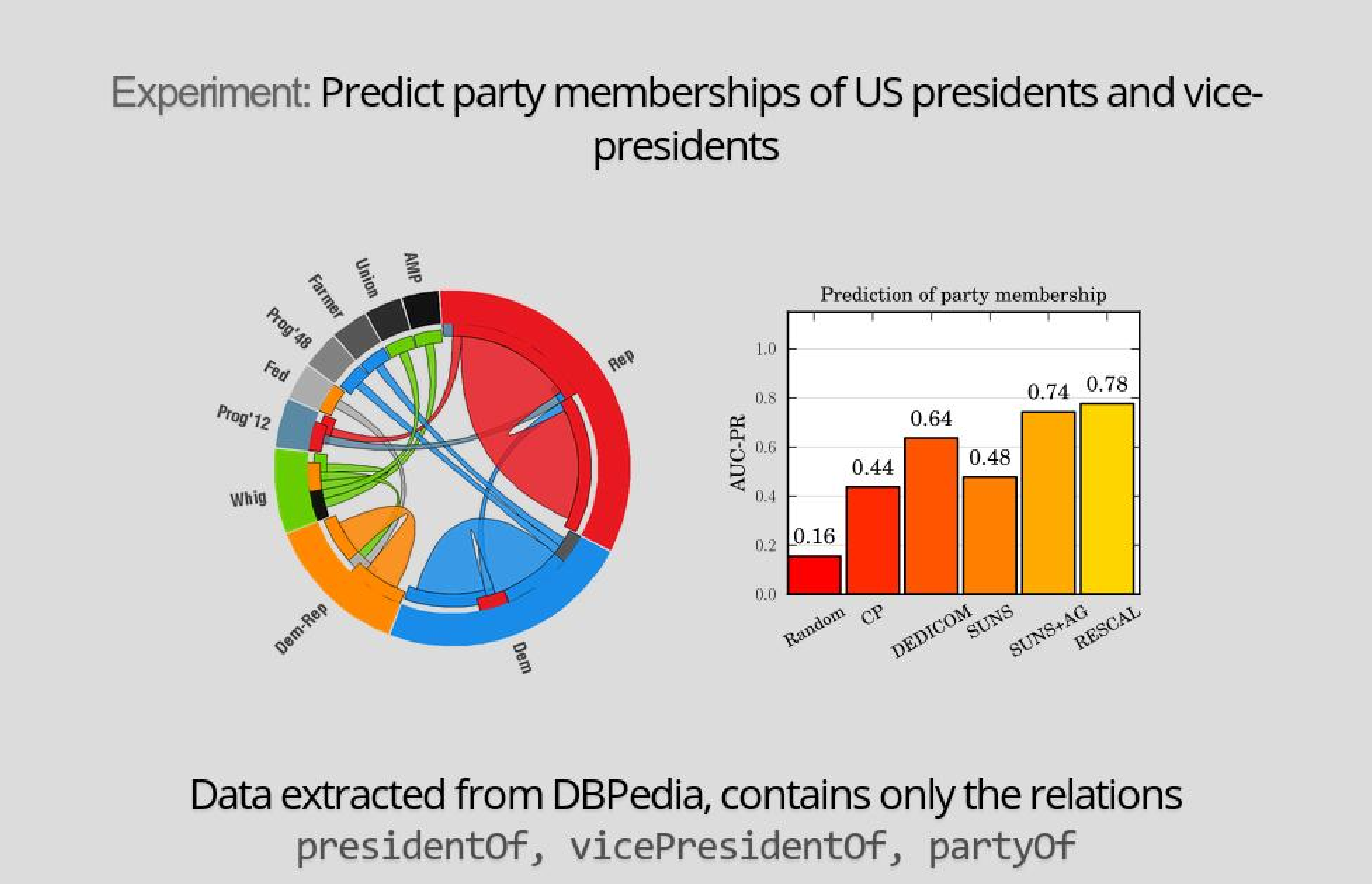
core ontology

2.6 million entities
340, 000 classes
87 predicates
71 million known facts

Tensor of size $3,000,000 \times 3,000,000 \times 40$
$(\approx 3.6 \times 10^{14}$ possible entries)

[Nickel , Tresp, Kriegel, WWW 2012]

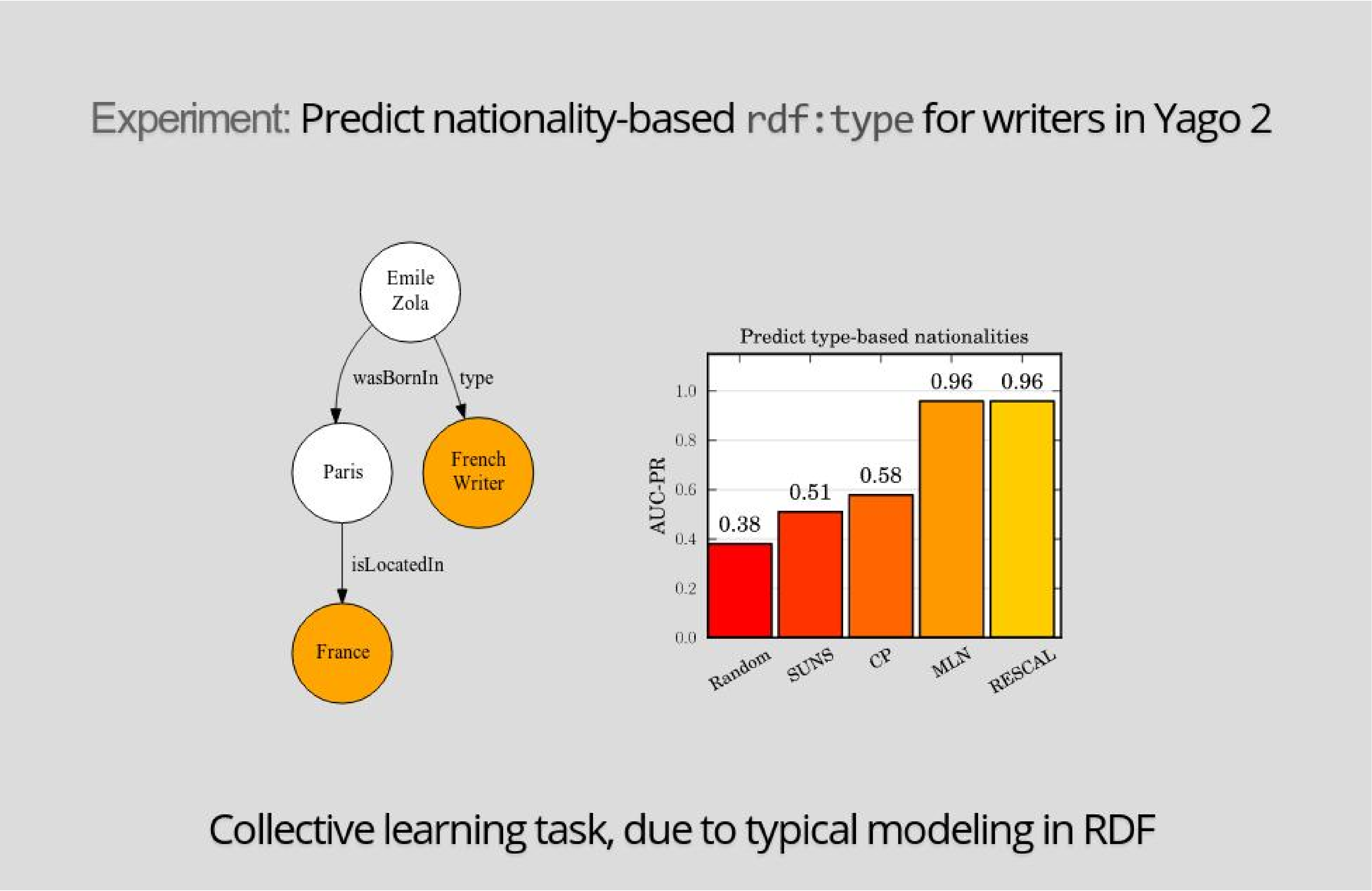
US-Presidents Example



Experiment: Predict party memberships of US presidents and vice-presidents

Data extracted from DBPedia, contains only the relations `presidentOf, vicePresidentOf, partyOf`

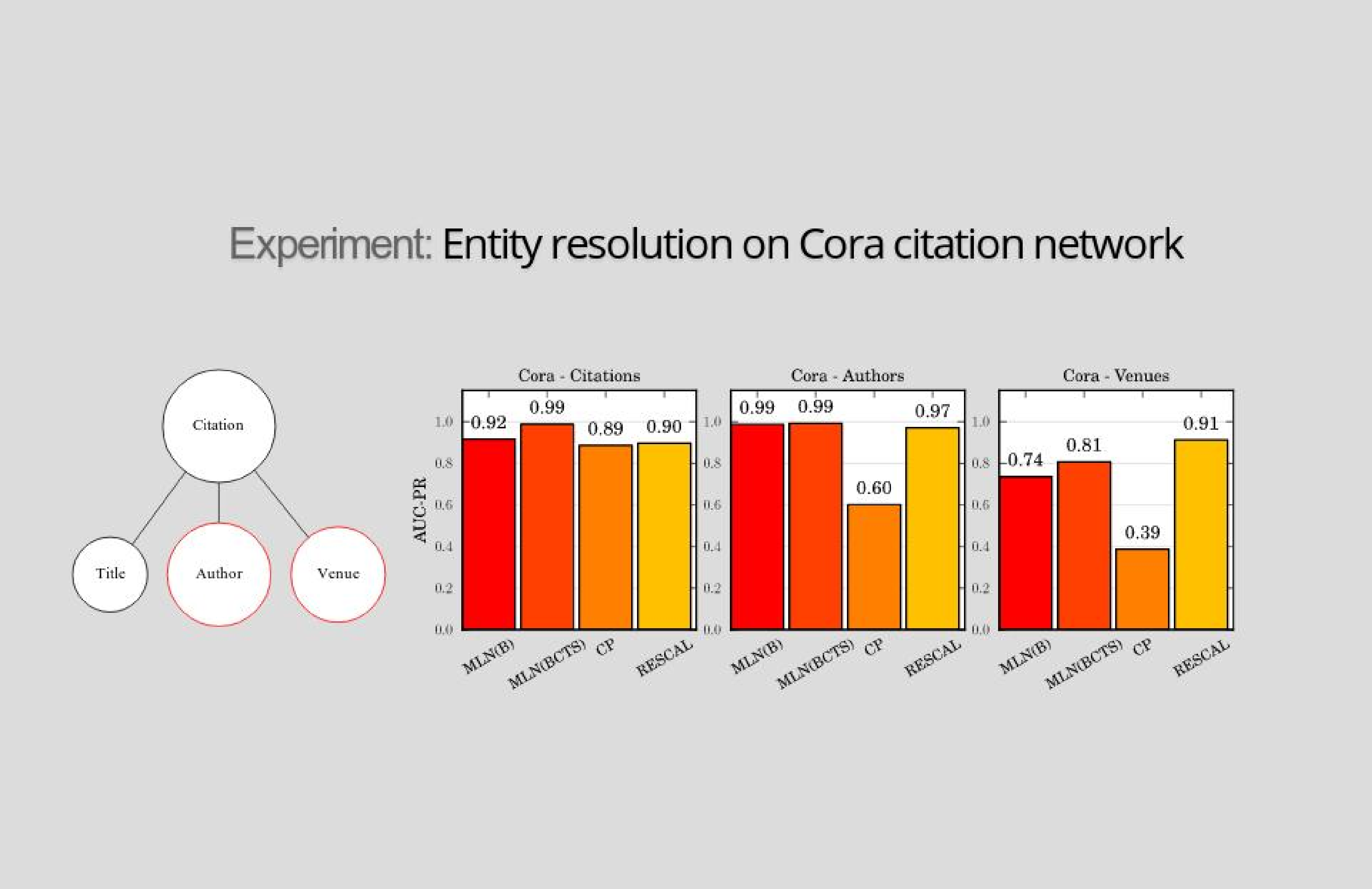
Writer‘s Nationality



Experiment: Predict nationality-based `rdf:type` for writers in Yago 2

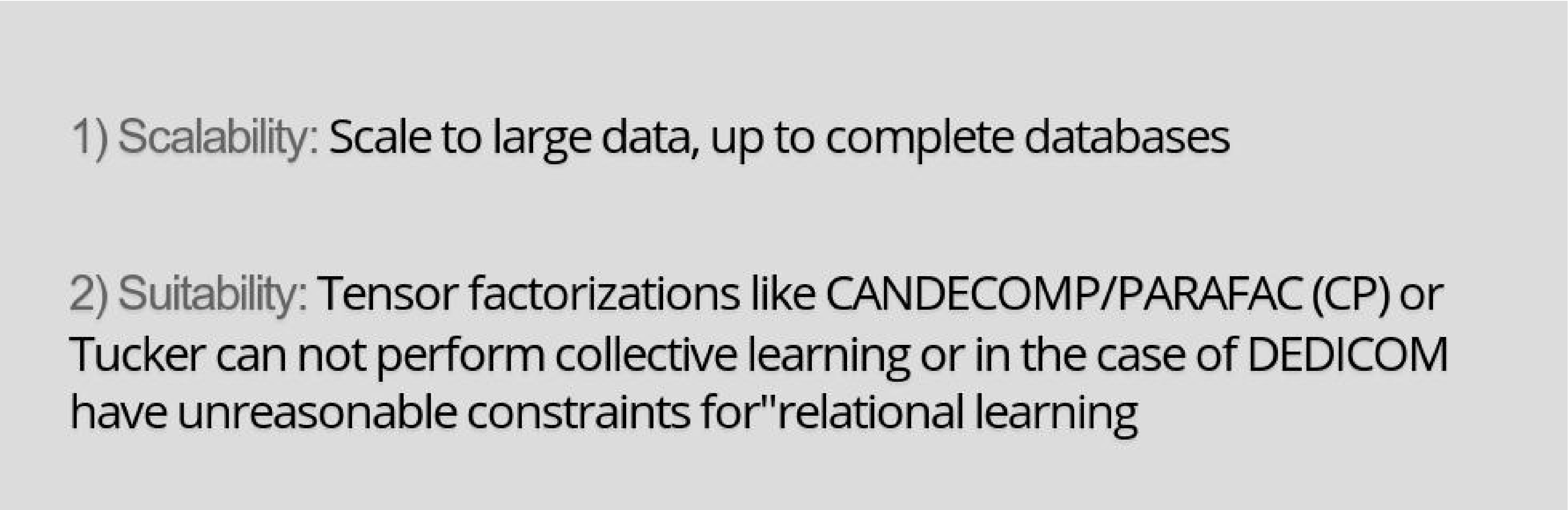
Collective learning task, due to typical modeling in RDF

Cora Citation Network



Experiment: Entity resolution on Cora citation network

Scalability



1) Scalability: Scale to large data, up to complete databases

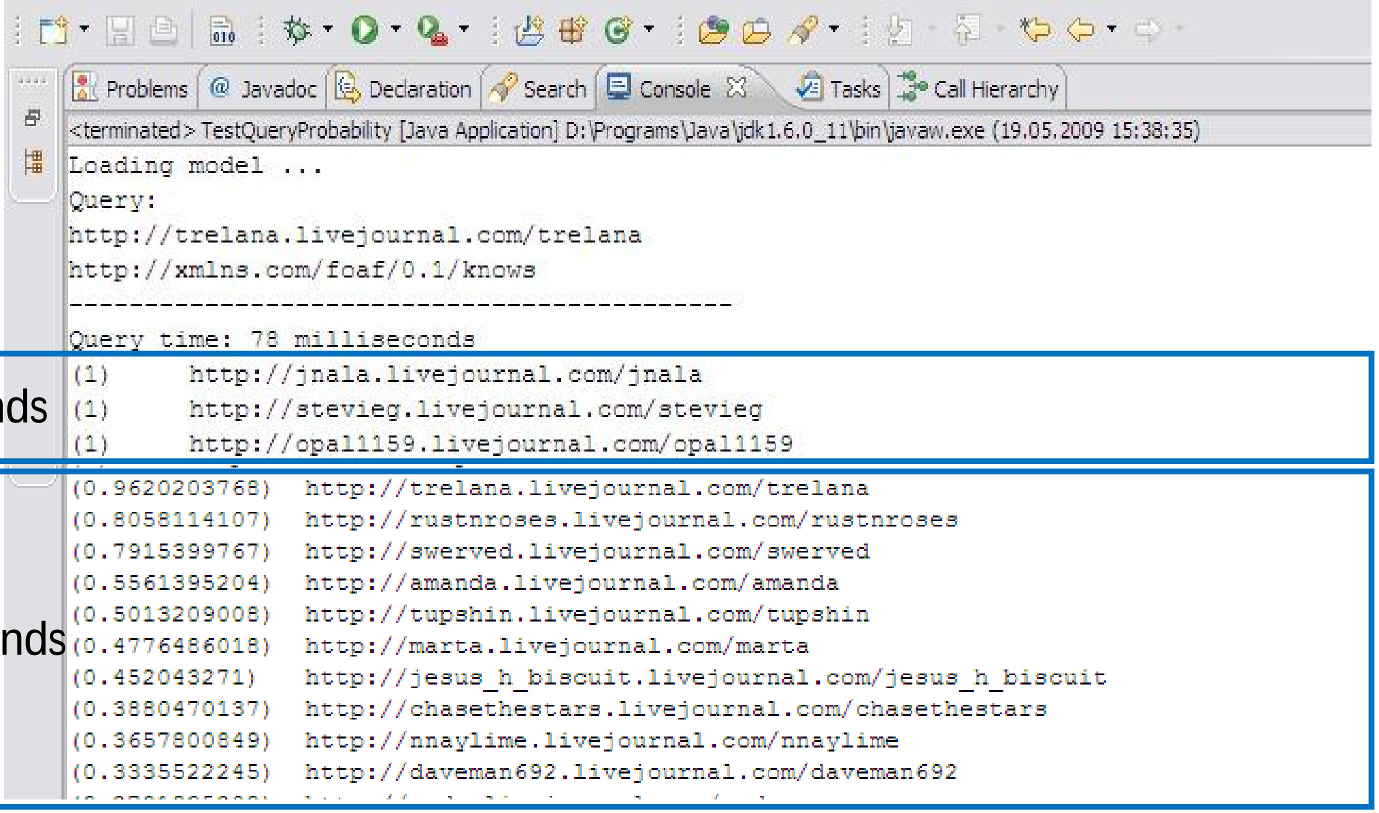
2) Suitability: Tensor factorizations like CANDECOMP/PARAFAC (CP) or Tucker can not perform collective learning or in the case of DEDICOM have unreasonable constraints for"relational learning

Querying with Statistical Machine Learning:

Find all persons, that live in Munich and who want to be Trelena‘s friends

```
PREFIX ya:   http://blogs.yandex.ru/schema/foaf/
PREFIX foaf: http://xmlns.com/foaf/0.1/
PREFIX dc:   http://purl.org/dc/elements/1.1/
SELECT DISTINCT ?person
WHERE
 { ?person ya:located ?city .
   ?person foaf:knows <http://trelana.livejournal.com/trelana>
          WITH PROB ?prob .
   FILTER REGEX(?city, "Munich") .
 }
ORDER BY DESC(?prob)
```

learn



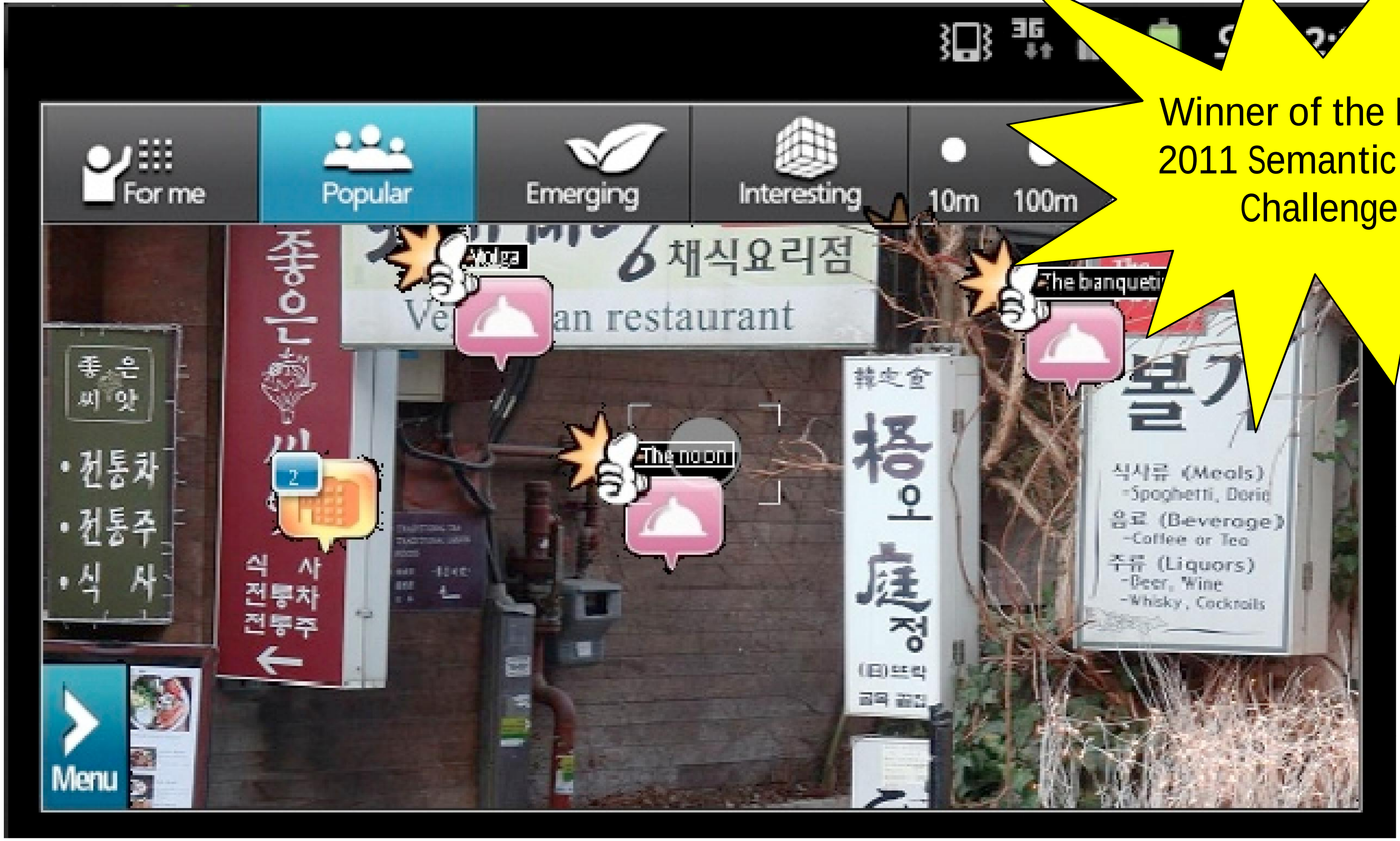
```
<terminated> TestQueryProbability [Java Application] D:\Programs\Java\jdk1.6.0_11\bin\javaw.exe (19.05.2009 15:38:35)
Loading model ...
Query:
http://trelana.livejournal.com/trelana
http://xmlns.com/foaf/0.1/knows
--------------------------------------------
Query time: 78 milliseconds
(1)      http://jnala.livejournal.com/jnala
(1)      http://stevieg.livejournal.com/stevieg
(1)      http://opal1159.livejournal.com/opal1159
(0.9620203768)  http://trelana.livejournal.com/trelana
(0.8058114107)  http://rustnroses.livejournal.com/rustnroses
(0.7915399767)  http://swerved.livejournal.com/swerved
(0.5561395204)  http://amanda.livejournal.com/amanda
(0.5013209008)  http://tupshin.livejournal.com/tupshin
(0.4776486018)  http://marta.livejournal.com/marta
(0.452043271)   http://jesus_h_biscuit.livejournal.com/jesus_h_biscuit
(0.3880470137)  http://chasethestars.livejournal.com/chasethestars
(0.3657800849)  http://nnaylime.livejournal.com/nnaylime
(0.3335522245)  http://daveman692.livejournal.com/daveman692
```

Known friends

Recom. Friends

Bottari: Deductive and Inductive Stream Reasoning for Semantic Social Media Analytics

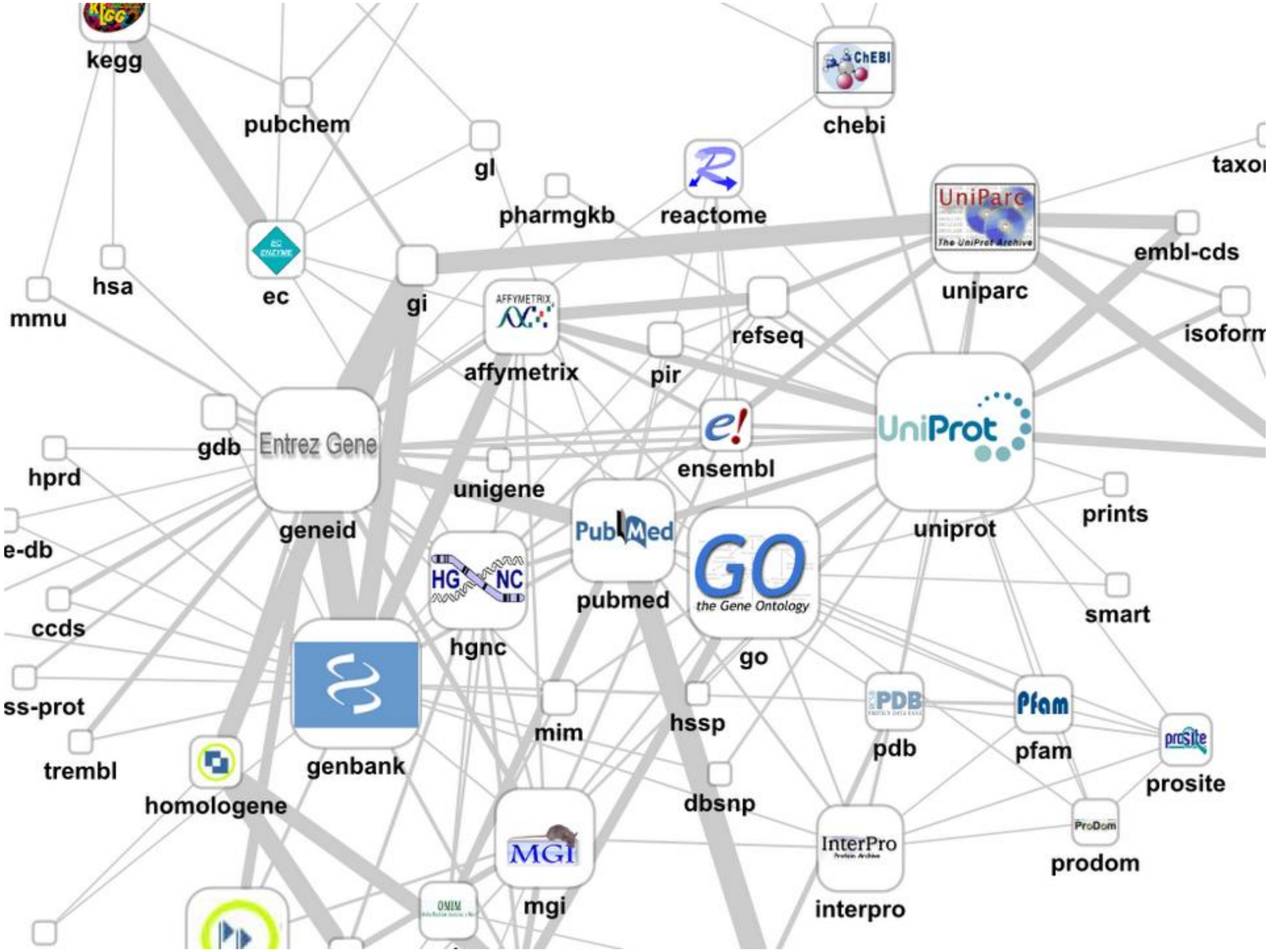
An augmented reality application for personalized **recommendation of restaurants** in Seoul



Winner of the ISWC 2011 Semantic Web Challenge

Balduini et al., JWS, 2012

Predicting Relationships between Genes and Diseases



[Huang et al., 2012] [Jiang et al., 2012]

Genes and Diseases

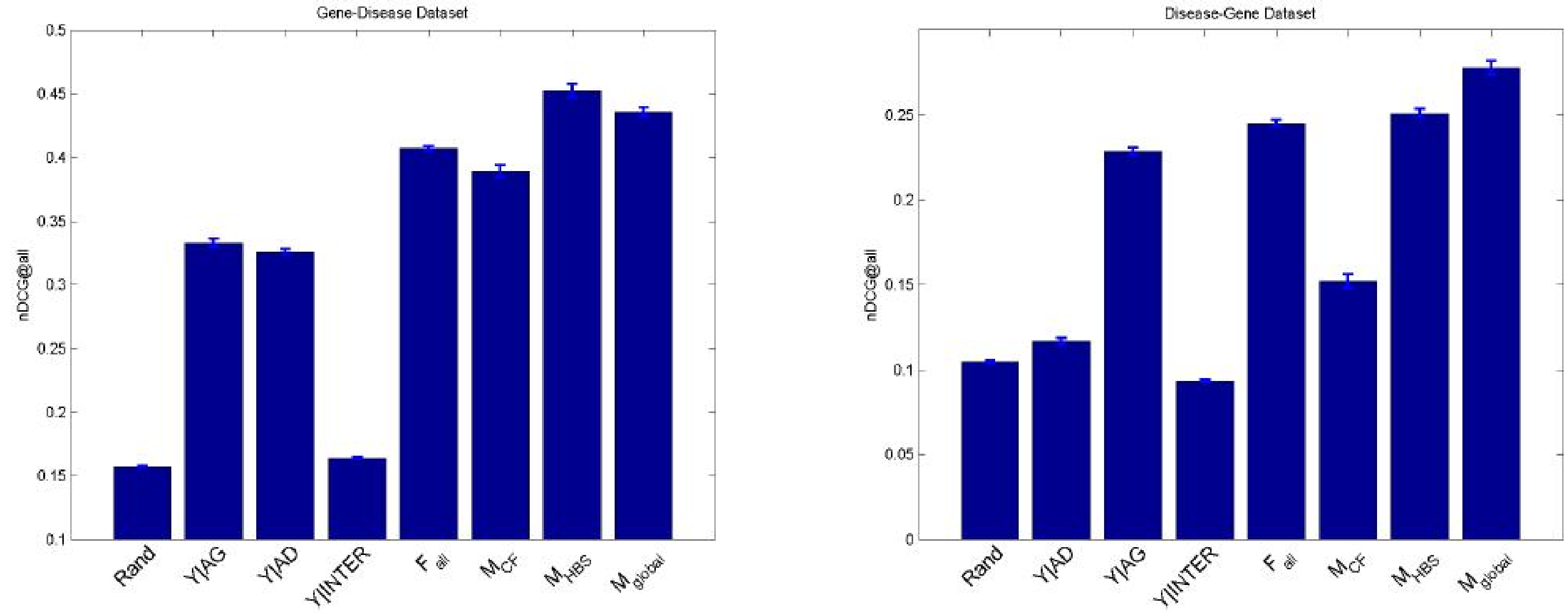


Fig. 4. The goal is to predict the relationship between genes and diseases. On the left we ranked recommended diseases for genes and on the right we ranked genes for diseases. In the left experiment, the subject attributes of the genes $F_{Y|AG}$, and of the object attributes of the diseases $F_{Y|AD}$ are comparable in strength. F_{all} that uses gene attributes, disease attributes and interaction terms in combination gives strong results. Our proposed model (M_{global}) can exploit both contextual information and intrarelational correlations. The reference model (M_{HBS}) is slightly stronger than our proposed model. The right plot shows results from the second experiment where we rank genes for diseases. This task is more difficult due to the large number of genes and our proposed system gives best results.

Patient in a Complex Environment with all Sorts of Networks

Patient Modell

A patient in multiple social and other networks with relationships to

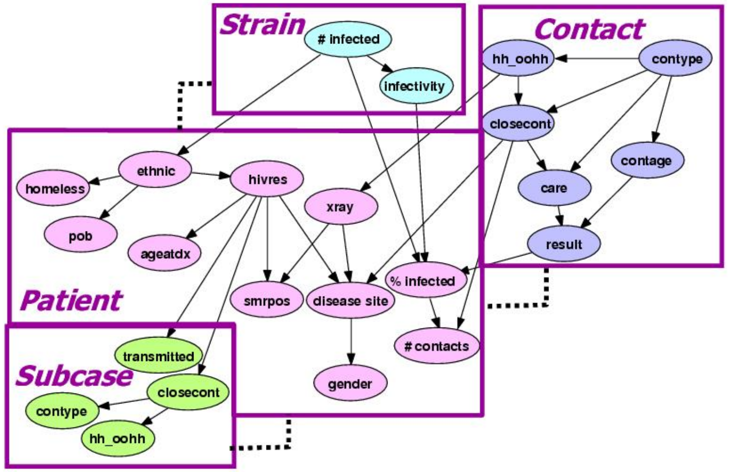
- physicians
- Patient with similar complaints
- Orders, medications
- Diagnosis
- Treatments

Increasing relevance of –omics data

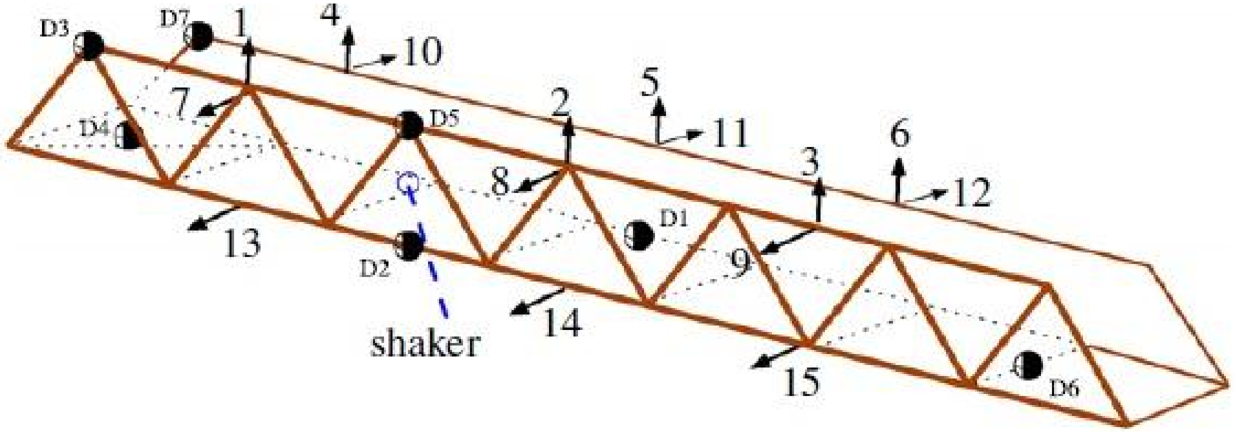
- genomics, proteomics, metabolomics, …

The new view

A patient in a clinic as a socal being with multiple complex relationships and attributes and part of severeral networks



Network of Sensors



Conclusions

- We have addressed the complexity aspect of Big Data
- Interesting data structures: graphs, matrices, tensors
- Machine learning as triple prediction (>10^14 in one step)
    - Efficient solutions exploiting sparse matrix algebra!

- We are extending our approach in several directions
    - Inclusion of textual documents and logical background
    - Sequential and temporal information
    - Real numbers