Accessing the Semantic Web via Statistical Machine Learning

Version 1.0

Yi Huang and Volker Tresp

Siemens Corporate Technology
Ludwig Maximilian University of Munich

(Joint work with Maximilian Nickel, Xueyan Jiang, Achim Rettinger)

Overview

I. Accessing the Semantic Web via Statistical Machine Learning
II. Linked Data and Statistical Machine Learning
III. Querying with Statistical Machine Learning
   - Triple Prediction with Machine Learning
   - SUNS Models
   - Extensions and Variations
IV. Three-Way Models (Tensor Models)
   - Tensor Models for Higher Order Relations
   - Tensor Models for Collective Learning
   - RESCAL
V. Conclusions

Overview

I. Accessing the Semantic Web via Statistical Machine Learning

II. Linked Data and Statistical Machine Learning

III. Querying with Statistical Machine Learning

- Triple Prediction with Machine Learning
- SUNS Models
- Extensions and Variations

IV. Three-Way Models (Tensor Models)

- Tensor Models for Higher Order Relations
- Tensor Models for Collective Learning
- RESCAL

V. Conclusions

Web Information Access: Keyword Search and Browsing

- The traditional means of extracting information from the Web are keyword search and browsing

- By using search, the user enters query terms and, if lucky, can read off the required information from the returned pages
  - Thus one can easily find out
    - What is the name of the eldest daughter of the president of the United States?
    - What is the best way to prepare a turkey?

- In browsing, the user follows hyperlinks to gain a deeper information on an issue

Web Information Access: Semantic Web

- The Semantic Web adds structured information (i.e., semantic annotations and references) with new search and browsing capabilities

Web Information Access: LOD

- One of the most interesting recent developments here is Linked Open Data (LOD)
  - where information is presented in form of facts -often originating from published domain-specific data bases- that can be accessed by both a human and a machine via specific query endpoints

- Thus, one can query for the
  - 10 largest German companies whose CEOs were born in the US
  - A list of: genes associated with a given disease

- LOD does not just *reference* information, it *represents* information in form of simple subject-predicate-object triples

Web Information Access: Reasoning

- With this novel representation of information, new opportunities for accessing information emerge that explore and exploit regularities in the represented data

- In recent years mostly deterministic regularities, which can be formulated as logical expressions, have been explored

- Thus, deductive reasoning might conclude that an author born in Landshut would also be recognized as an author born in Bavaria

- Deterministic regularities originate, for example:
  - from natural laws (e.g., law of gravity)
  - from human definitions and conventions (e.g.,”dogs are mammals”)
  - from design(e.g.,”the car only starts when the key is turned”)
  - and from laws and regulations (e.g.,”work begins at 9 am”)

Web Information Access: Uncertainty

- In addition to deterministic or close-to-deterministic regularities, the world also contains statistical regularities
- One might debate if the world is inherently deterministic or probabilistic, but at the abstract level of representation that is typically available for decision making, the world certainly appears uncertain
- Although the world might be governed by scientific laws and logical constraints in general, at the level of abstraction that we and our applications have to function, the world partially appears to be governed by probabilities and statistical patterns

Uncertain Expert Knowledge?

- Difficult to get
  - Querying of experts
- Modelling
  - + Can brush over details in the model by using probabilities
    - Truly a printer or a car is deterministic but at a more abstract level one can work with probabilities
  - + Truthful if the world is really probabilistic
  - - Conditional probabilities might be difficult to get
- Difficult to do inference
  - Problems with loops

Web Information Access: Include Machine Learning

- *Young males typically like action movies but whether young Jack will buy "Iron Man 2" might depend more on the availability of illegal downloads, the opinions of his peers and Jack's financial situation*
  - A system recommending a movie to Jack must work with the available information, maybe a list of movies that Jack has bought before, and can only make statistical recommendations

- Similarly, the interactions between genes and diseases might or might not be inherently deterministic in nature; at the level of current knowledge the relationships are only partially known

Integration into Querying

- Machine learning is a basis for extracting statistical patterns and in this tutorial we will describe our work on statistical machine learning for the Web as pursued in the German THESEUS project and in the EU FP7 project LarKC

- In this work we have proposed that statistical patterns extracted from the Web via machine learning should be integrated into queries

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

Problems | Javadoc | Declaration | Search | Console | Tasks | Call Hierarchy

```
<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

Machine Learning Examples:

- Thus a search for diseases associated with a particular gene can be done in three ways:
  - First, one can study the documents returned via keyword search
  - Second, one can obtain a list of diseases known or suspected to be associated via a structured query on LOD
  - Finally, one can use machine learning to extract diseases likely to be related based on disease and gene attributes and established gene-disease patterns
- Note that machine learning depends on repeatable statistical patterns: thus machine learning cannot help to give you the first name of the wife of the US-president (a one-to-one relationship), but it can predict the probability of re-election, his income, the party of the vice president and the number of expected grand children

Challenges: Unclear statistical setting

- Web data does not represent an i.i.d. (independent and identically distributed) statistical sample of some sort but might have been collected and published for any reason, often not following a particular systematic
- For similar reasons, the data, in general, is incomplete, e.g., from the fact that a social network lists no friends of Jack one cannot conclude that Jack does not have friends
- In general, negation is very rare on Web data, thus one might find information that two persons are friends but rarely that two persons are not friends
  - This needs to be considered to avoid biased predictions

Challenges: Relationships are important

- Another interesting property of Web data is that relationships between entities are often more informative than entity attributes, an effect exploited in collective learning:
  - It might be easier to predict Jane's wealth from the wealth of her friends than from known properties of Jane. As in this example, nonlocal aggregated information is often informative for a prediction task and machine learning needs to take this into account.
- Collective learning / collective classification

Challenges: Relationships are of Interest

- Sometimes, as in the examples mentioned in the introduction, relationships themselves are of interest, e.g.,
  - item preferences
  - friendships
  - relationships between genes and diseases

- Since the number of potential relationships is generally very large, the output of a learning system will often be a ranked list of candidate relationships, e.g., a ranked list of recommended items, instead of a single answer

Challenges: Should be easy to use and easy to apply

- Learning as easy as querying
- Answer any question on the likelihood of a triples at any time

Challenges: A Lot of Contextual Information

- After all a triple lives in a graph

Challenges: Textual Information Available

- Often there is textual information available describing entities
    - Wikipedia pages
    - Text as literals

- (basis for the success of IBM Watson)

Challenges: Scale, dynamics, noisiness

- Information is incomplete
- Information is large scale
- Information might change continuously

Challenges: Ontological Background Knowledge

- Ontological background is available and should support machine learning

Goal

Predictions of <u>all</u> (or a large subset of) possible triples in a domain
with Machine Learning?

Overview

What’s the Problem

Attitude of a naive Machine Learner

Of course I should know what my data means

- In the Semantic Web / LOD this is what is the challenge!

Linked Open Data is about Presenting and Communicating Information

Linked data describes a method of

- publishing structured data so that it can be
- interlinked

and becomes more useful

Information Elements

Subject Predicate Object

Information Elements

Subject Predicate Object

- Elementary statements providing information on the subject
  - RDF triples; (Nikolas, hasFriend, Angela)

What a Computer Understands: How can we get meaning?

尼古拉·萨科齐 有朋友 安格拉·默克尔

Communicating Meaning (“Semantics”): URIs

<http://siemens.com/world-knowledge/Angela>

<http://siemens.com/world-knowledge/hasFriend>

<http://siemens.com/world-knowledge/Nikolas> .

- Use URIs (Uniform Resource Identifiers)
- The URI might give away meaning (Angela likely is a human being)

Communicating Meaning (“Semantics”): Specifying Types

<http://siemens.com/world-knowledge/Angela>
<rdf:type>
<Foaf:Person> .

Communicating Meaning ("Semantics"): Specifying Links

<http://siemens.com/world-knowledge/Angela>

<Owl:sameAs>

<http://yago-knowledge.org/resource/Angela_Merkel> .

Communicating Meaning (“Semantics”): Meta Data

Meta data (when a triple is not enough):

- Who provided the information (provenance)
- When is the triple valid (“Lincoln is the president of the US”)
- Where is the triple true (“The weather is nice”)

Communicating Meaning (“Semantics”): Ontologies

Ontologies:

- subclass hierarchies
- type constraints
- partOf
- geo-reasoning
- sameAs, …

LOD by Tim Berners-Lee

Tim Berners-Lee outlined four principles of linked data in his Design Issues:

- Use URIs to identify things
- Use HTTP URIs so that these things can be referred to and looked up ("dereferenced") by people and user agents
- Provide useful information about the thing when its URI is dereferenced, using standard formats such as RDF/XML
- Include links to other, related URIs in the exposed data to improve discovery of other related information on the Web

Roles of Machine Learning

- **Supporting LOD**
  - Supports mapping of entities (e.g., are two persons the same) and classes (is “student” the same as “pupil”)
  - (Discovers order: ontology learning)

Roles of Machine Learning

- **Supporting Information Retrieval**
  - Retrieval of similar entities

Roles of Machine Learning

Enhancing LOD by triple prediction (focus in this tutorial)

- Predict links (triples) that are not explicitly in the data base
  - Classes (classical ML problems)
  - Attributes (classical ML problems)
  - Relationships (classical SRL problems)

Roles of Machine Learning

Exploiting LOD for ML (focus in this tutorial)

- LOD as a great source of data for ML

Overview

I. Accessing the Semantic Web via Statistical Machine Learning
II. Linked Data and Statistical Machine Learning
III. Querying with Statistical Machine Learning
- Triple Prediction with Machine Learning
- SUNS Models
- Extensions and Variations

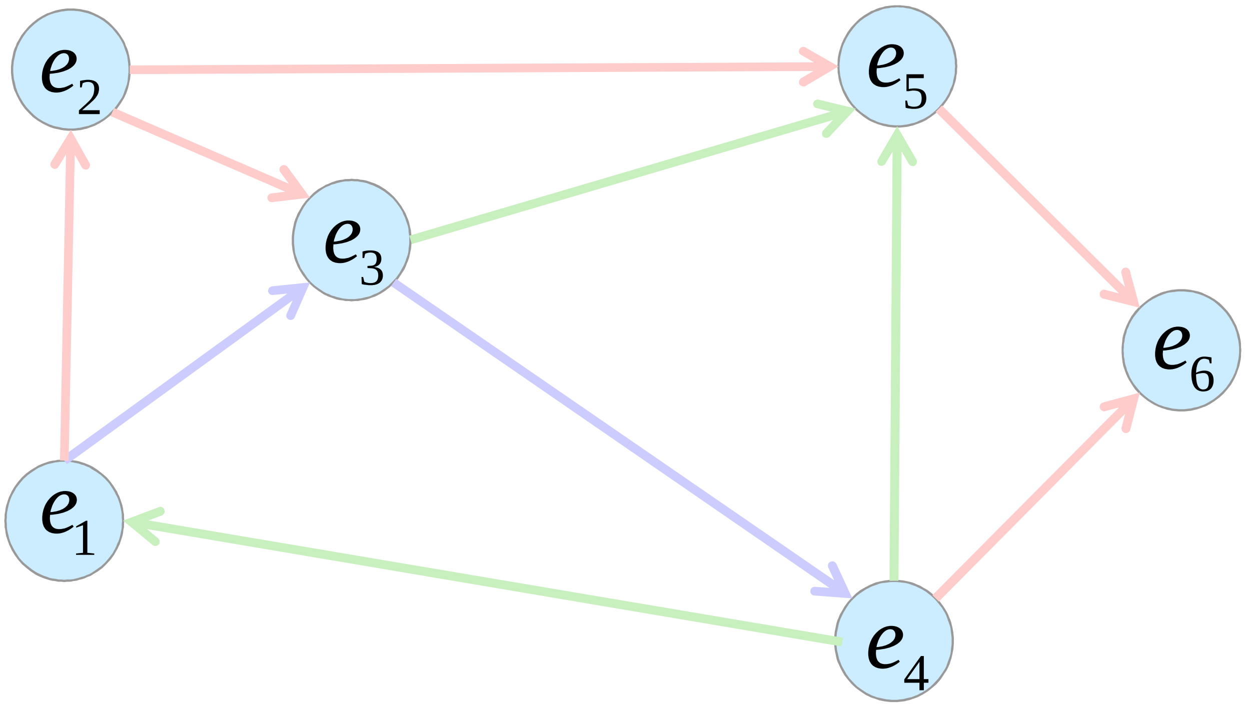
IV. Three-Way Models (Tensor Models)
- Tensor Models for Higher Order Relations
- Tensor Models for Collective Learning
- RESCAL

V. Conclusions

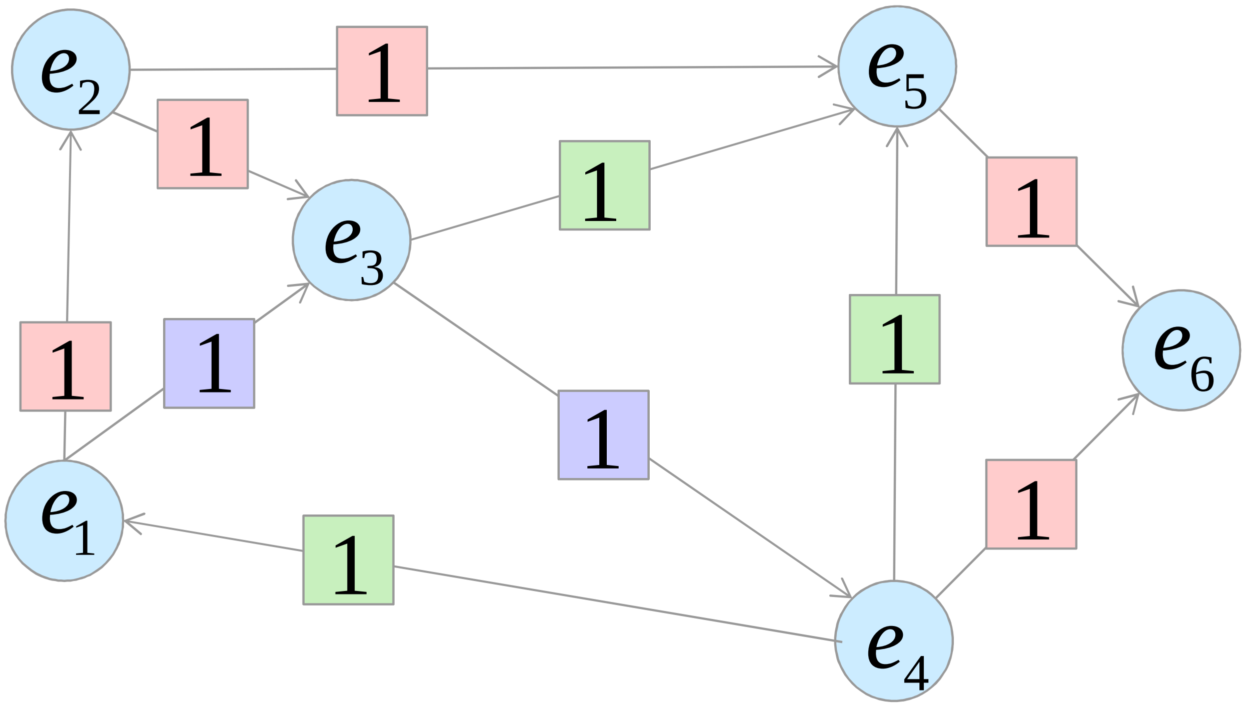
Overview

I. Accessing the Semantic Web via Statistical Machine Learning
II. Linked Data and Statistical Machine Learning
III. Querying with Statistical Machine Learning
   - **Triple Prediction with Machine Learning**
   - SUNS Models
   - Extensions and Variations
IV. Three-Way Models (Tensor Models)
   - Tensor Models for Higher Order Relations
   - Tensor Models for Collective Learning
   - RESCAL
V. Conclusions

RDF Graph

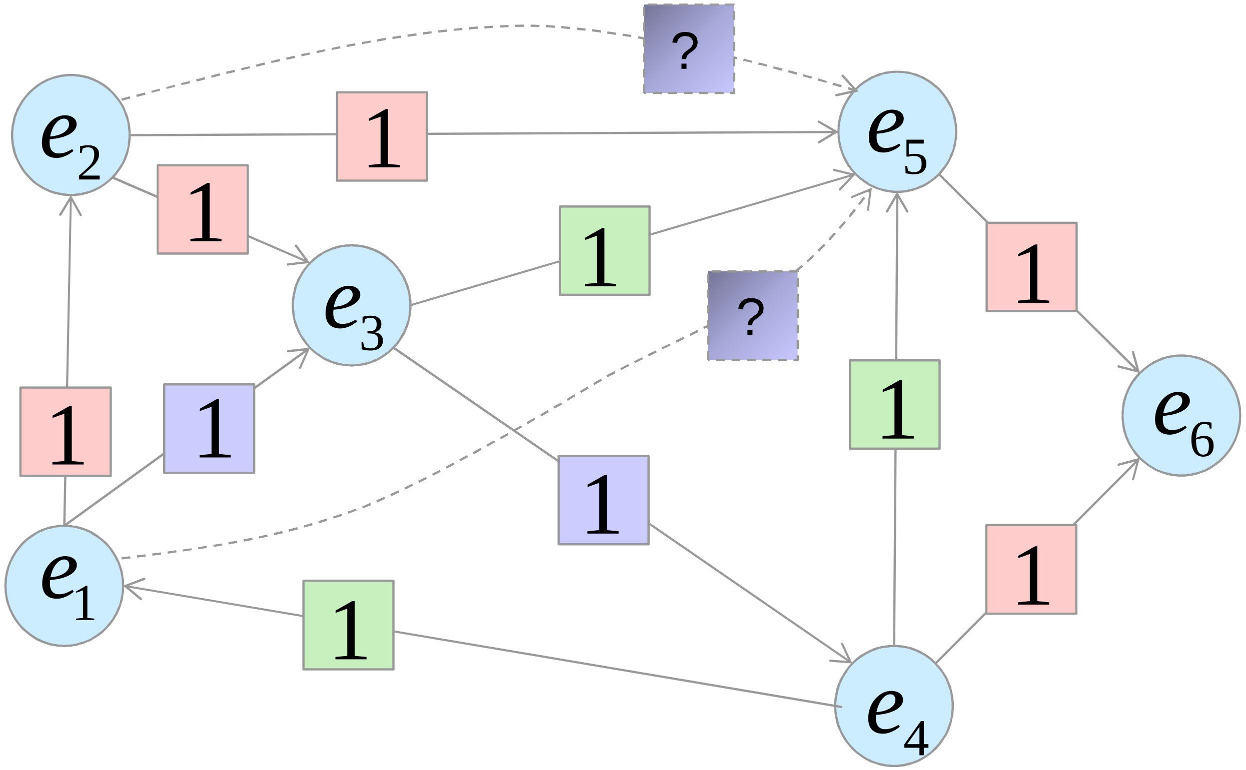


RDF Graph with Random Variables



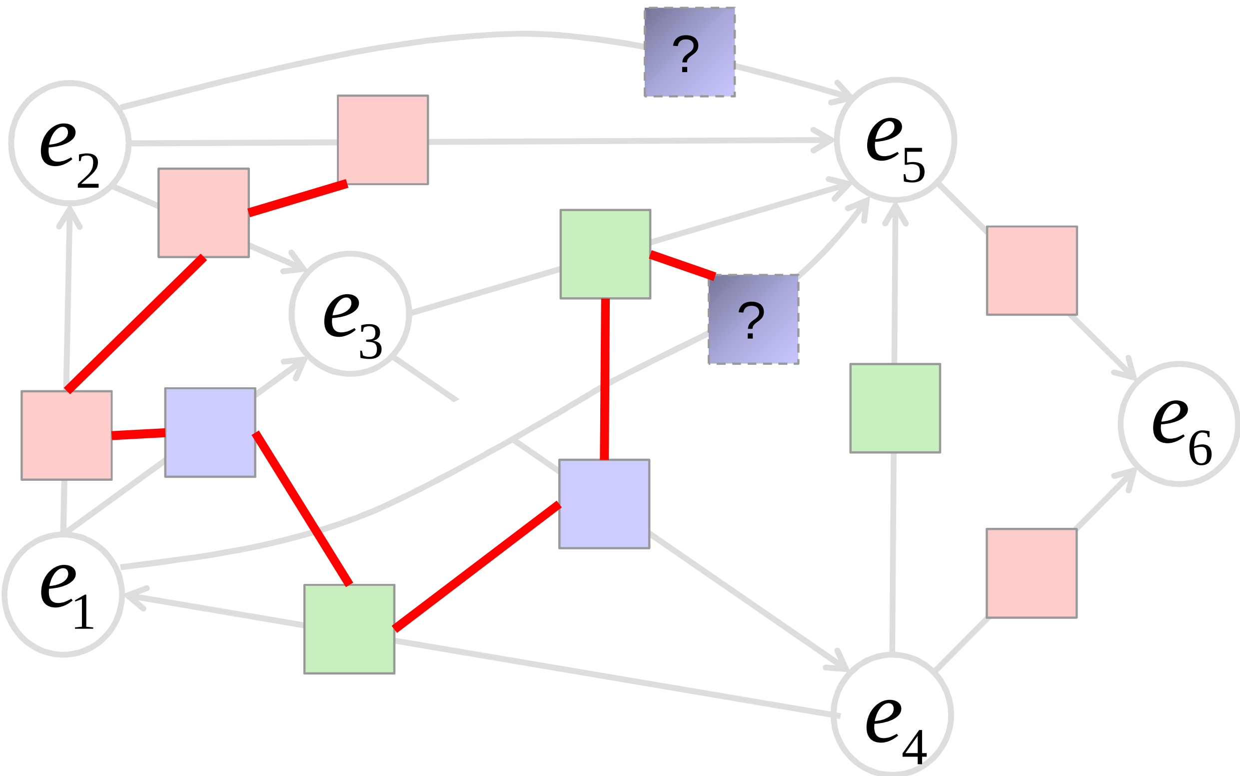
- Variables (rectangles) representing the truth values of triples
- A one indicates that the triple is known to be true; a zero indicates that the triple is untrue
- Color: relation type

Goal: Predict new Triples



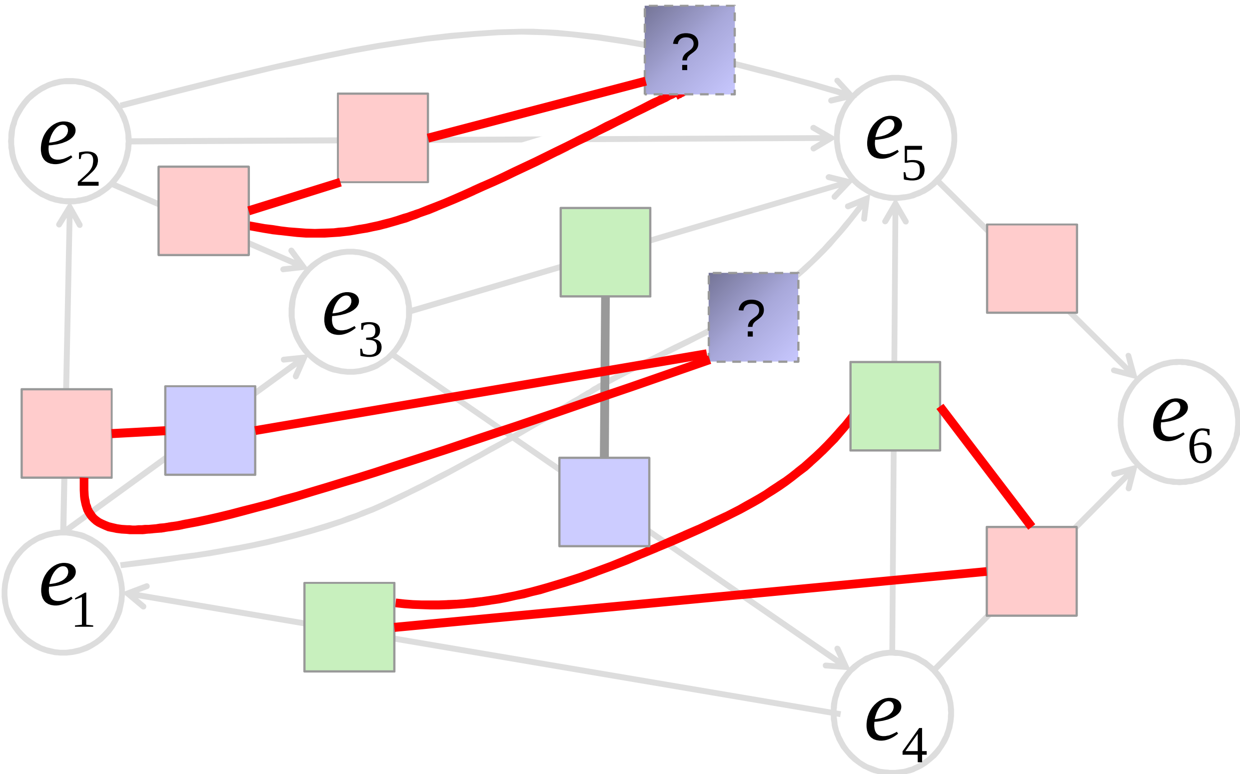
- Goal of machine learning: predict triples not known to be true (dashed links)

Dual Graphs displaying Dependencies and Independencies



- Links in between the variables indicate assumed dependencies and independencies (we consider different options)
- Parameterization of dependencies

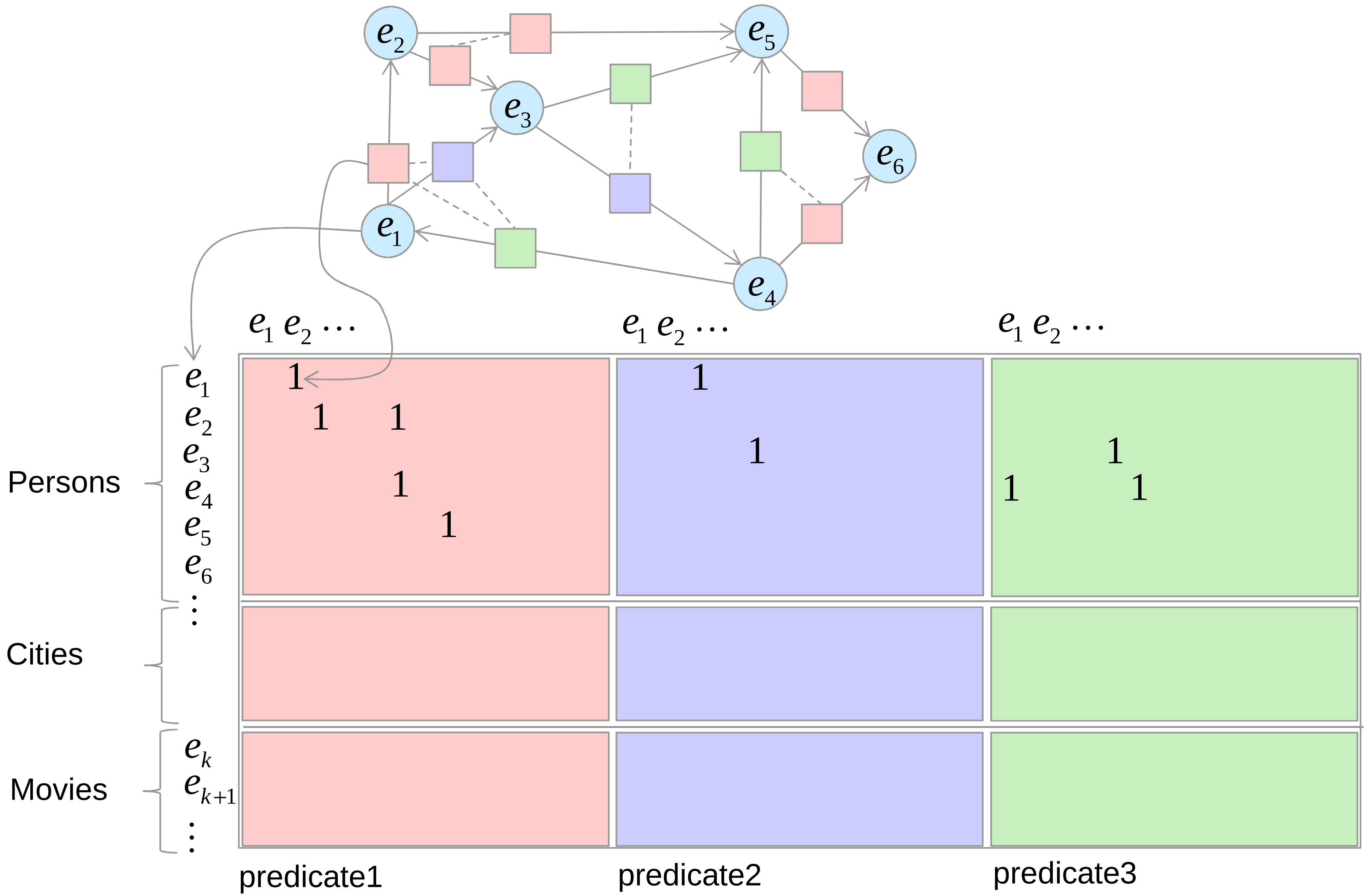
Dual Graphs in the SUNS Model



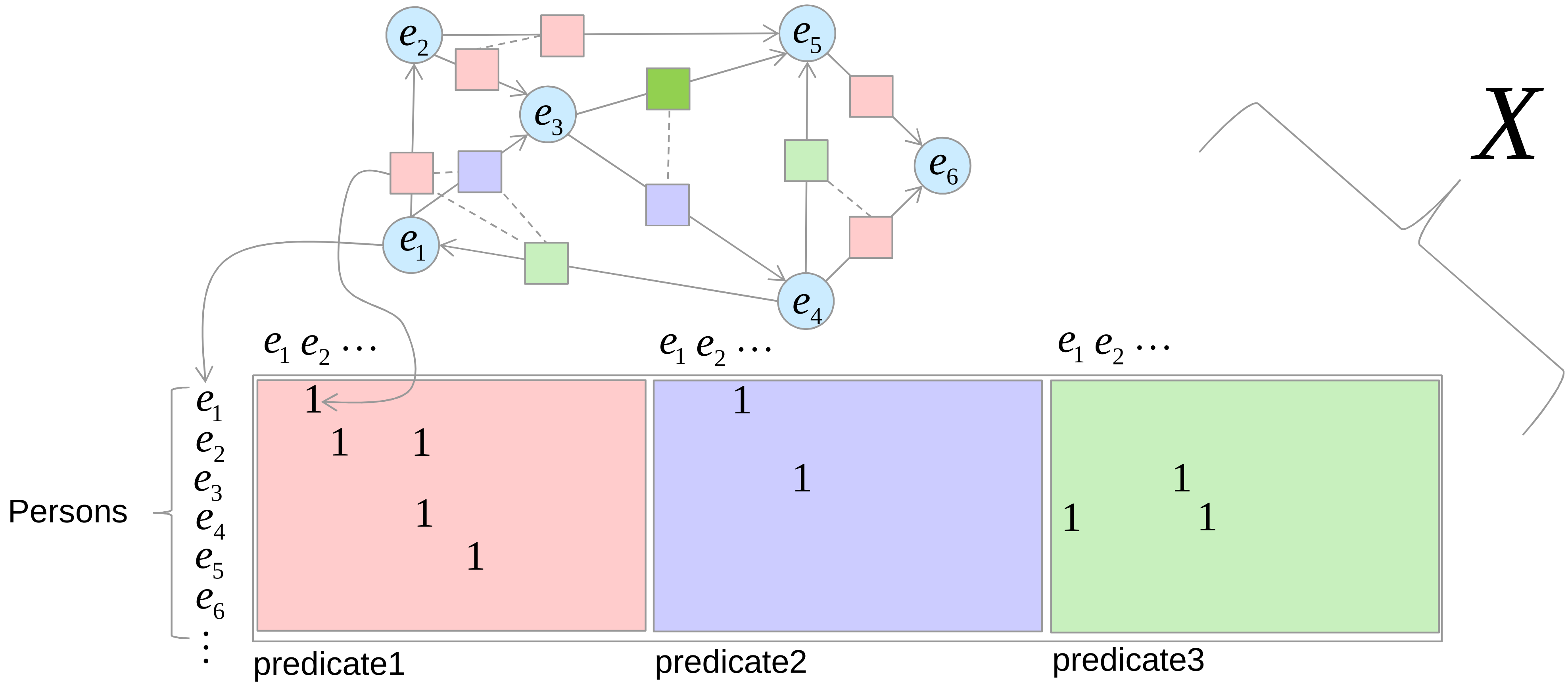
- Links are defined between variables with the same subject entity

Overview

From RDF to a Data Matrix



SUNS



$$\hat{X} = U_r \ \mathrm{diag}_r\left(\frac{d_i^3}{d_i^2+\lambda}\right) V_r^T = U_r \ \mathrm{diag}_r\left(\frac{d_i^2}{d_i^2+\lambda}\right) U_r^T X = X V_r \,\mathrm{diag}_r\left(\frac{d_i^2}{d_i^2+\lambda}\right) V_r^T$$

Mathematical Background

Consider the singular value decomposition (SVD) $X = UDV^T$

The columns of *U* are mutually orthonormal, the columns of *V* are mutually orthonormal and *D* is a diagonal matrix with diagonal entries greater or equal to zero and ordered according to magnitude

An optimal rank-*r* reconstruction in the Frobenius norm $\left\| \hat{X} - X \right\|_F$

is $$\hat{X} = U_r \,\mathrm{diag}_\mathrm{r}(d_i)\, V_r^T$$

U_r, V_r contain the first *r* columns of *U*, resp. *V*

$\mathrm{diag}_\mathrm{r}(d_i)$ contains the *r* leading entries of *D*

Consists of the first *r* columns of *U* and *V* and the leading *r* entries of *D*

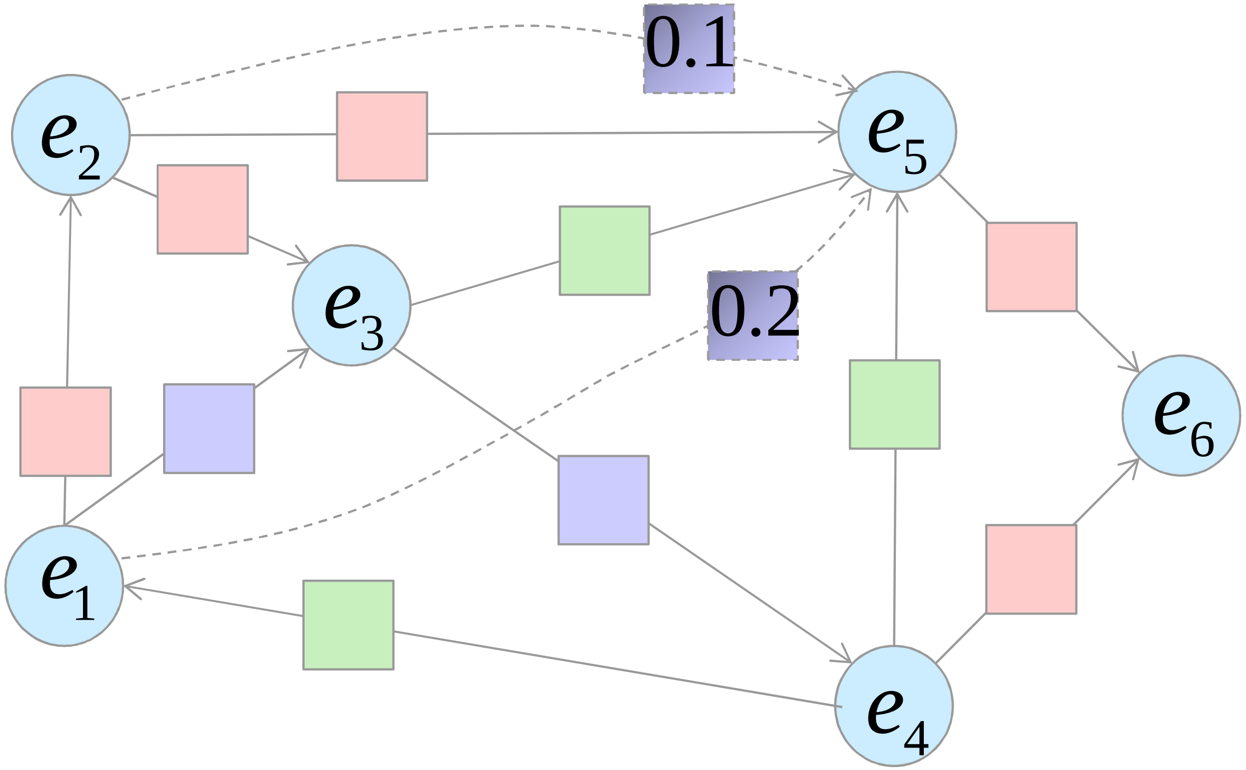
Better performance can be obtained by the regularized version

$$\hat{X} = U_r \ \mathrm{diag}_\mathrm{r}\left(\frac{d_i^3}{d_i^2+\lambda}\right) V_r^T = U_r \ \mathrm{diag}_\mathrm{r}\left(\frac{d_i^2}{d_i^2+\lambda}\right) U_r^T X = X V_r \,\mathrm{diag}_\mathrm{r}\left(\frac{d_i^2}{d_i^2+\lambda}\right) V_r^T$$

where $\lambda \geq 0$ is a regularization parameter

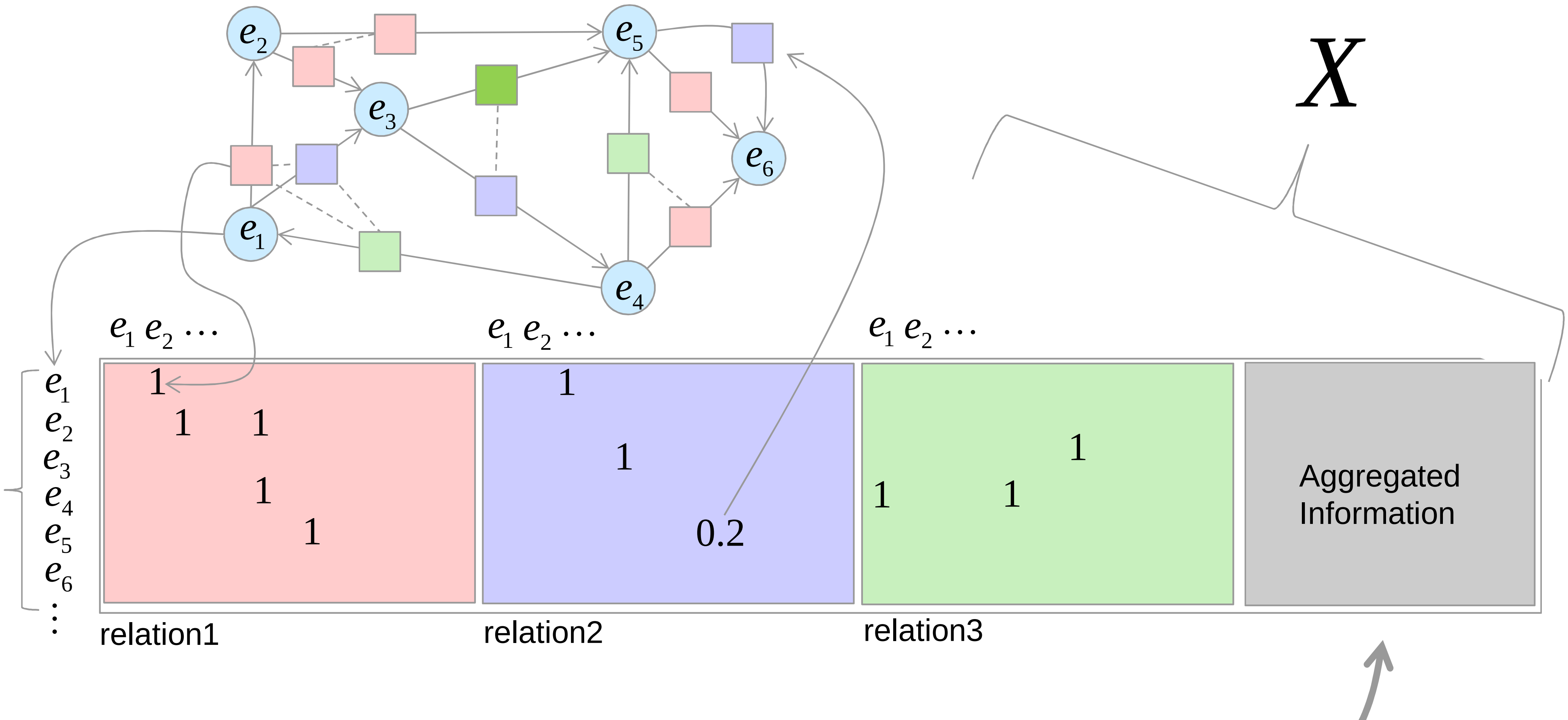
[Tresp, Huang, Bundschus, Rettinger, 2009] [Huang et al., 2010].

Confidence Values for New Triples



- SUNS prediction for unknown triples

SUNS with Aggregation



Example: average income of friends

Properties

Unkown triples

- Unknown triple as negative evidence to be overwritten by machine learning
    - Very likely non existing

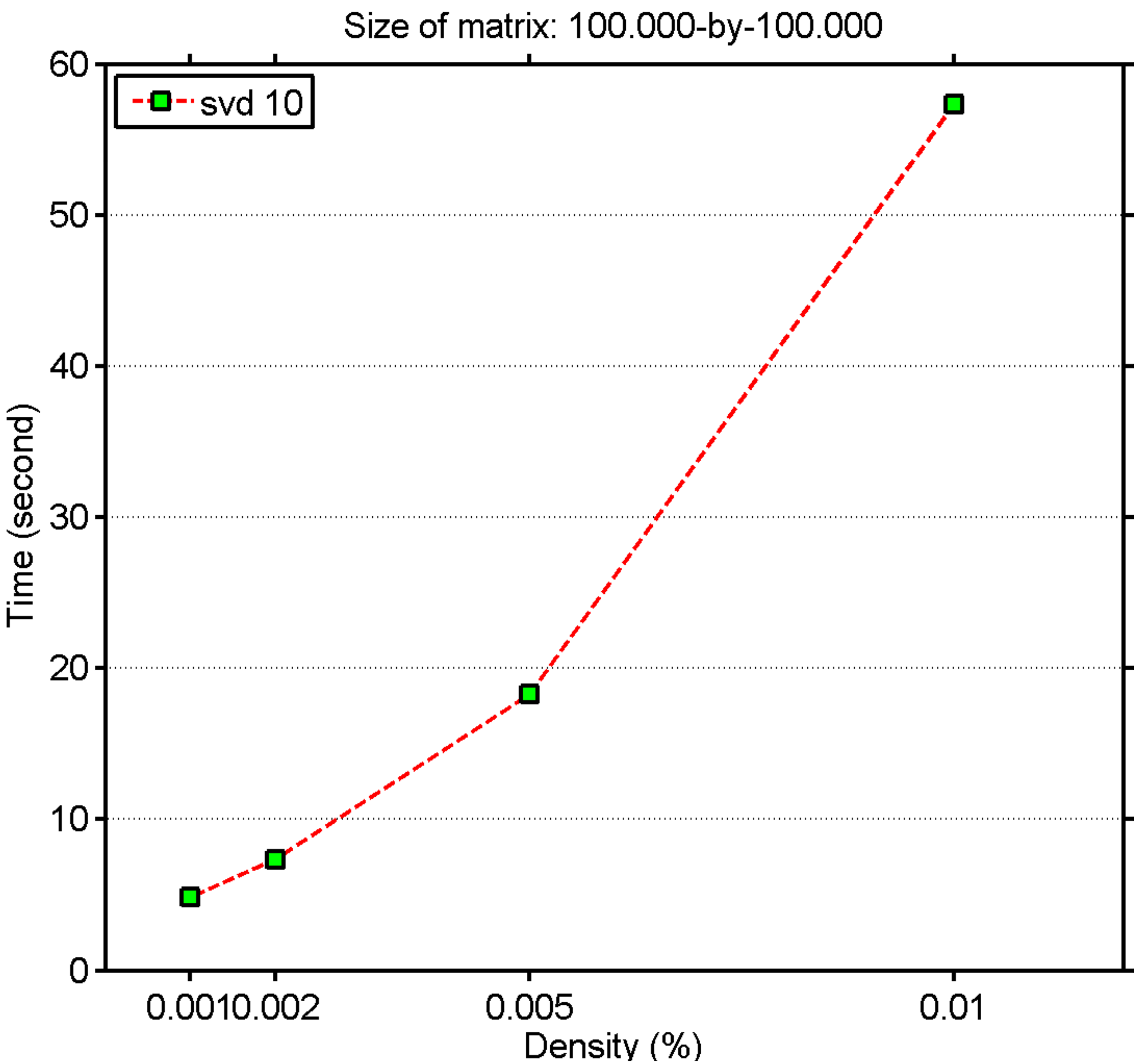
Scalability

- Exploiting sparse matrix algebra:
    - Computation scales as
      (number of columns) X (number of nonzero elements) X (rank *r*)

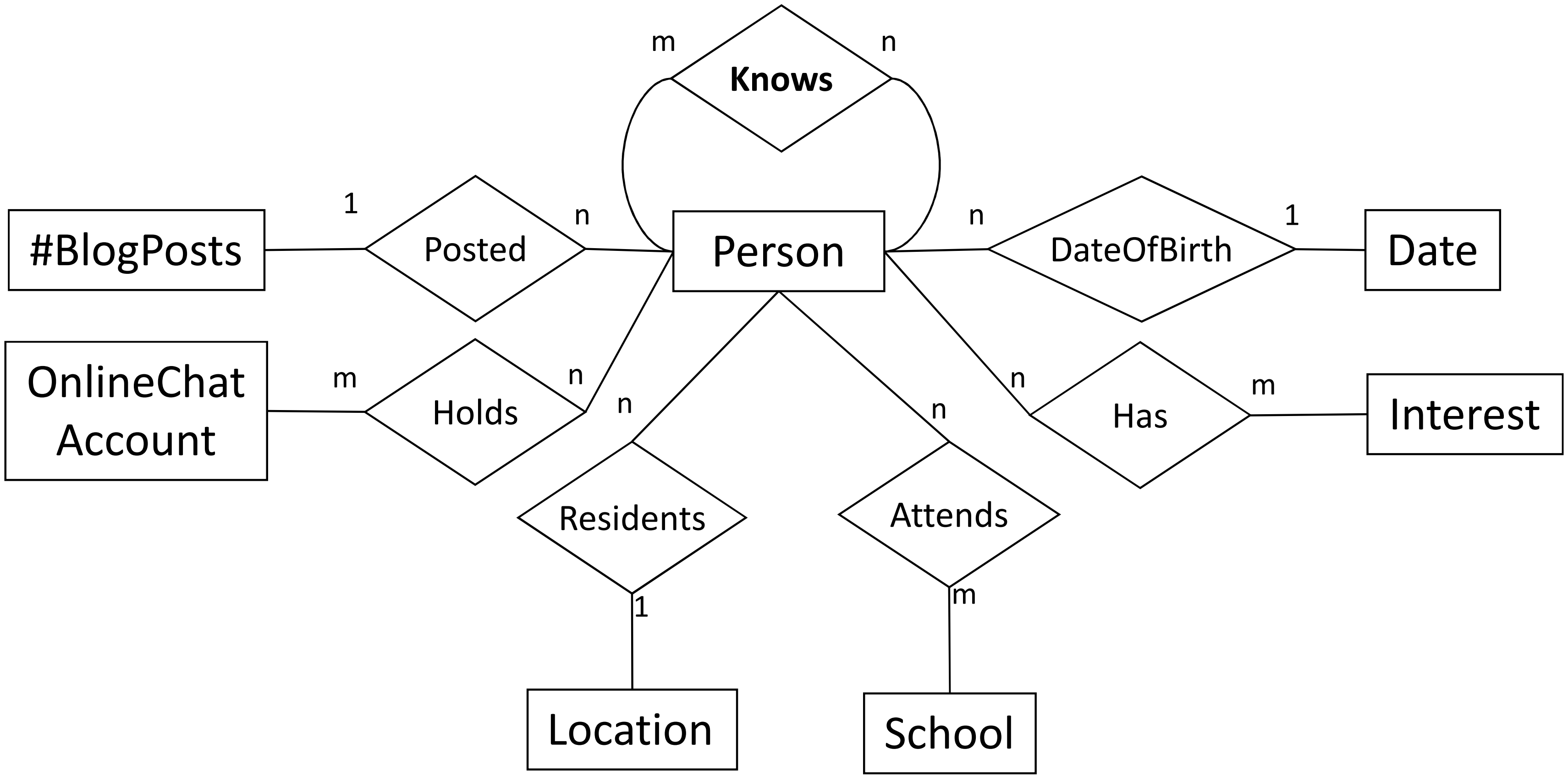
Ontologies and Deduction

- sameAs is resolved
- Materialization (deductive closure) prior to learning

Scalability

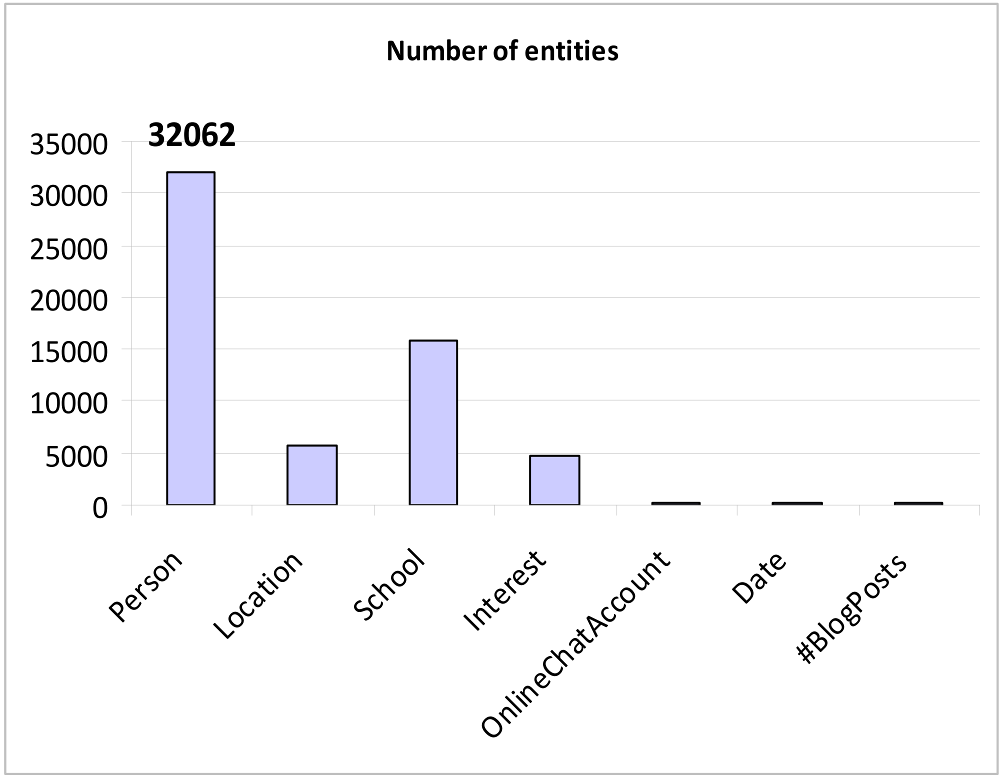


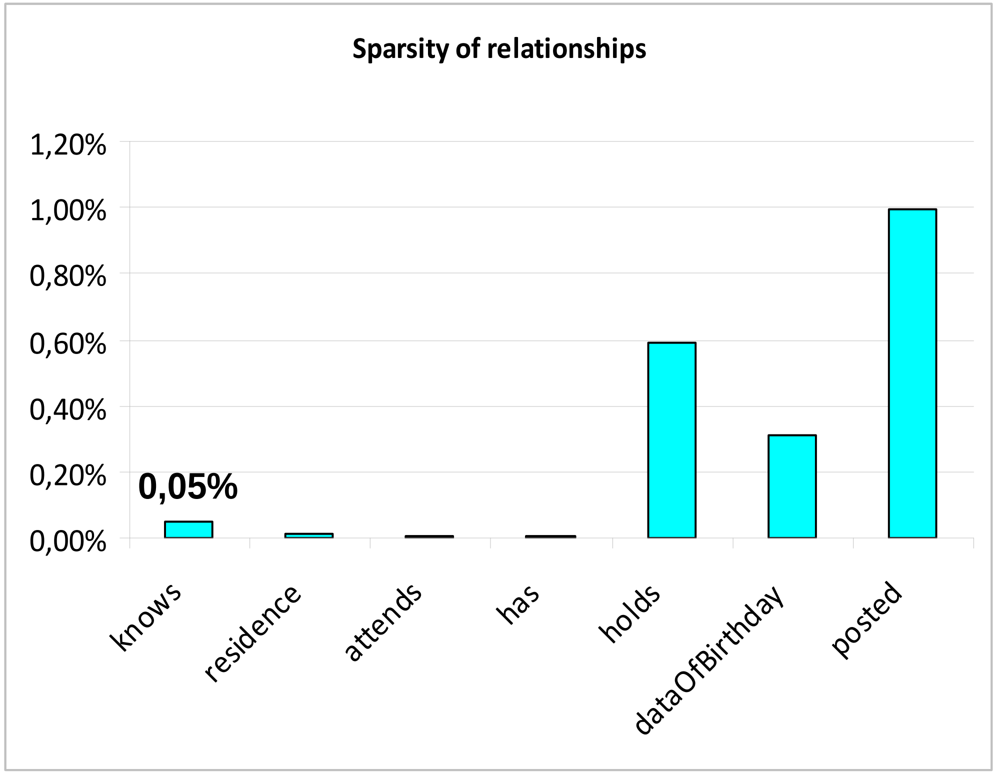
Experiments Using the Basic Model on the FOAF Data Set



Entity-relationship diagram of LJ-FOAF domain

Data Statistics





Query Example

```
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

Problems | Javadoc | Declaration | Search | Console | Tasks | Call Hierarchy

<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

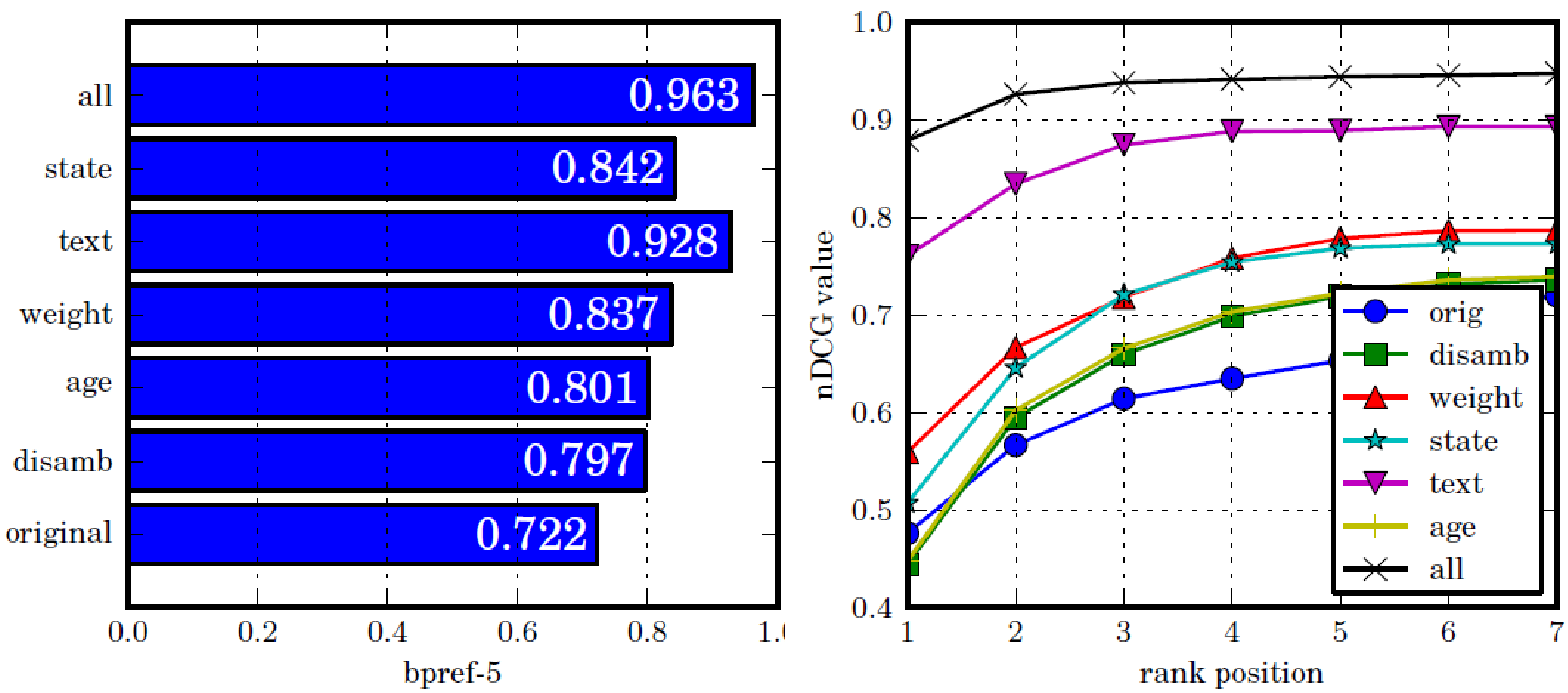
Recom. Friends

Experiments with DBpedia Data

- DBpedia is part of LOD and contains structured information extracted from Wikipedia. It serves as a "nucleus for the web of data”
- Even though DBpedia already provides a great value, it is still limited in the information it provides and in terms of quality
  - Although there are many cities covered in DBpedia, most information, like its most famous citizens and its most spectacular sights, is not very useful for machine learning purposes
- Here we report results using a population consisting of all ***members of the German Bundestag***
- Challenges:
  - **Incomplete data:** Only 101 of 293 members of the German Bundestag represented in DBpedia have an entry for the predicate *party*
  - **Noisy predicates:** For predicates it is often the case that there are semantical duplicates, e.g. *dbp-prop:party* and *dbp-ont:party*. While duplicate predicates aren't a big problem by default, they can become a challenge when they are used inconsistently
  - **Noisy objects:** E.g. the Christian Democratic Union of Germany was represented by the literals "CDU" and "Christian Democratic Union" or the different resources
- In the following experiments the learning challenge was to correctly predict the political party for each subject

Preprocessing

- **ORIG:** The original data that has been obtained from DBpedia
- **DISAMB:** After the disambiguation exactly one resource for each party (CDU, CSU, SPD, FDP, Alliance '90/The Greens, The Left, Centre Party) was present in the data set
- **AGE:** In this experiment a continuous feature for the age of each politician has been added, by subtracting the birth year (when present) from the year 2010
- **WEIGHT:** In this experimental setting the attributes and relations in the dataset have been weighted differently. Attributes have been weighted down by multiplying their values with 0.4
- **STATE:** Naturally, the birthplace is not a useful attribute for our task. Filling in the state information from the birthplace information can easily be done by exploiting geographical part-of-relationships with OWL reasoning by using Ontotext's Linked Data Semantic Repository http://ldsr.ontotext.com (geo-reasoning)
- **TEXT:** Finally a text mining approach has been added, by tokenizing the objects of the predicates *comment* and *abstract* and adding one feature for each occurring token. When a token is present for a particular statistical unit, the entry is set to 1, else to 0
- **ALL:** In this experiment all previously described approaches have been combined. Since the number of attributes changed, we also changed the weighting factor to 0.2



- Worst results with the raw data
- Disambiguation improved results by 7 percent
- Small improvement by adding age
- Weighting improved results
- Textual description, often containing hints on party memberships, improved the results to 0.928
- State made a significant contribution
- Best results by the combination of all methods (0.963)

[Huang et al., 2012]

Application (3)
Bottari: Deductive and Inductive Stream Reasoning for Semantic Social Media Analytics

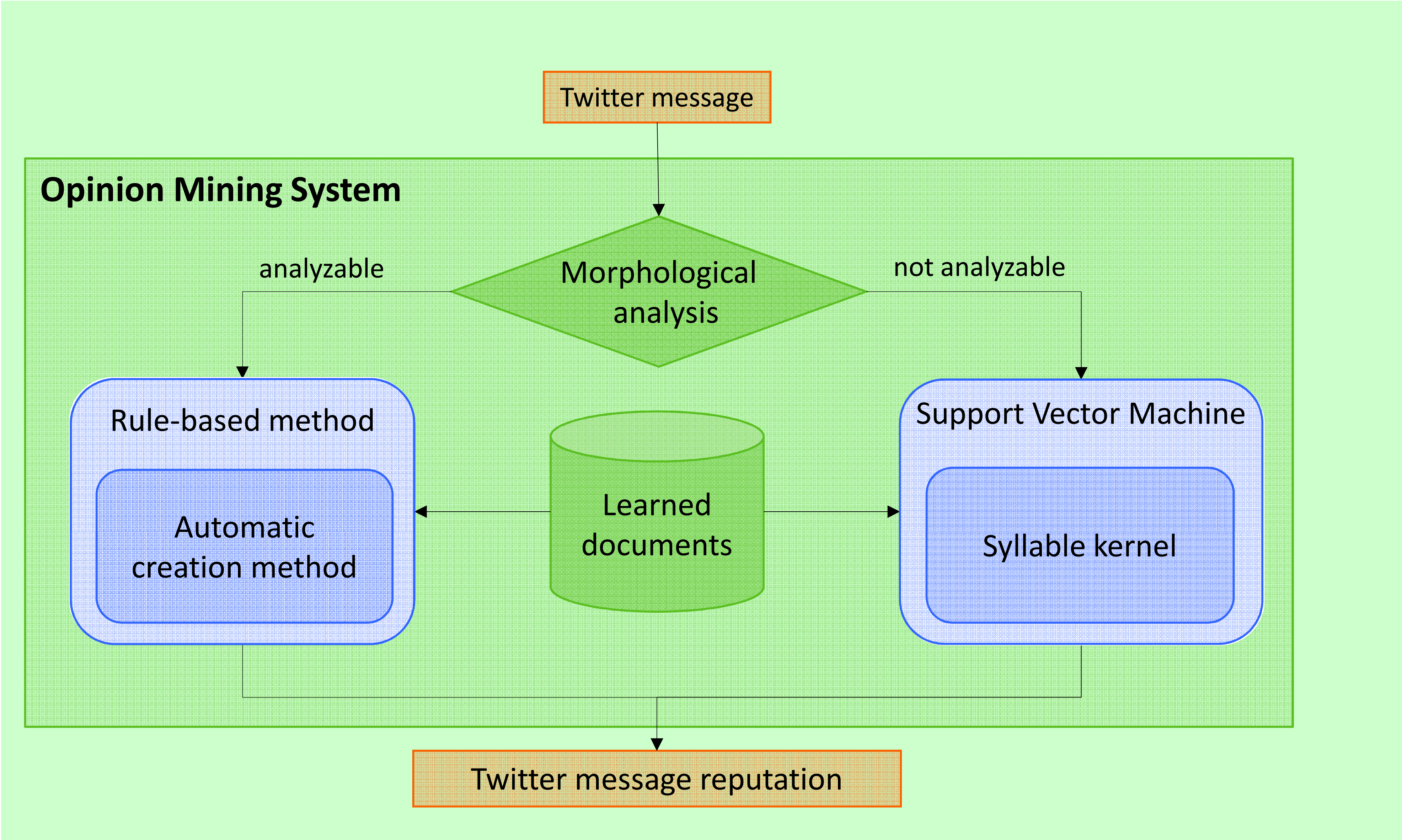
An augmented reality application for personalized **recommendation of restaurants** in Seoul

Winner of the ISWC 2011 Semantic Web Challenge

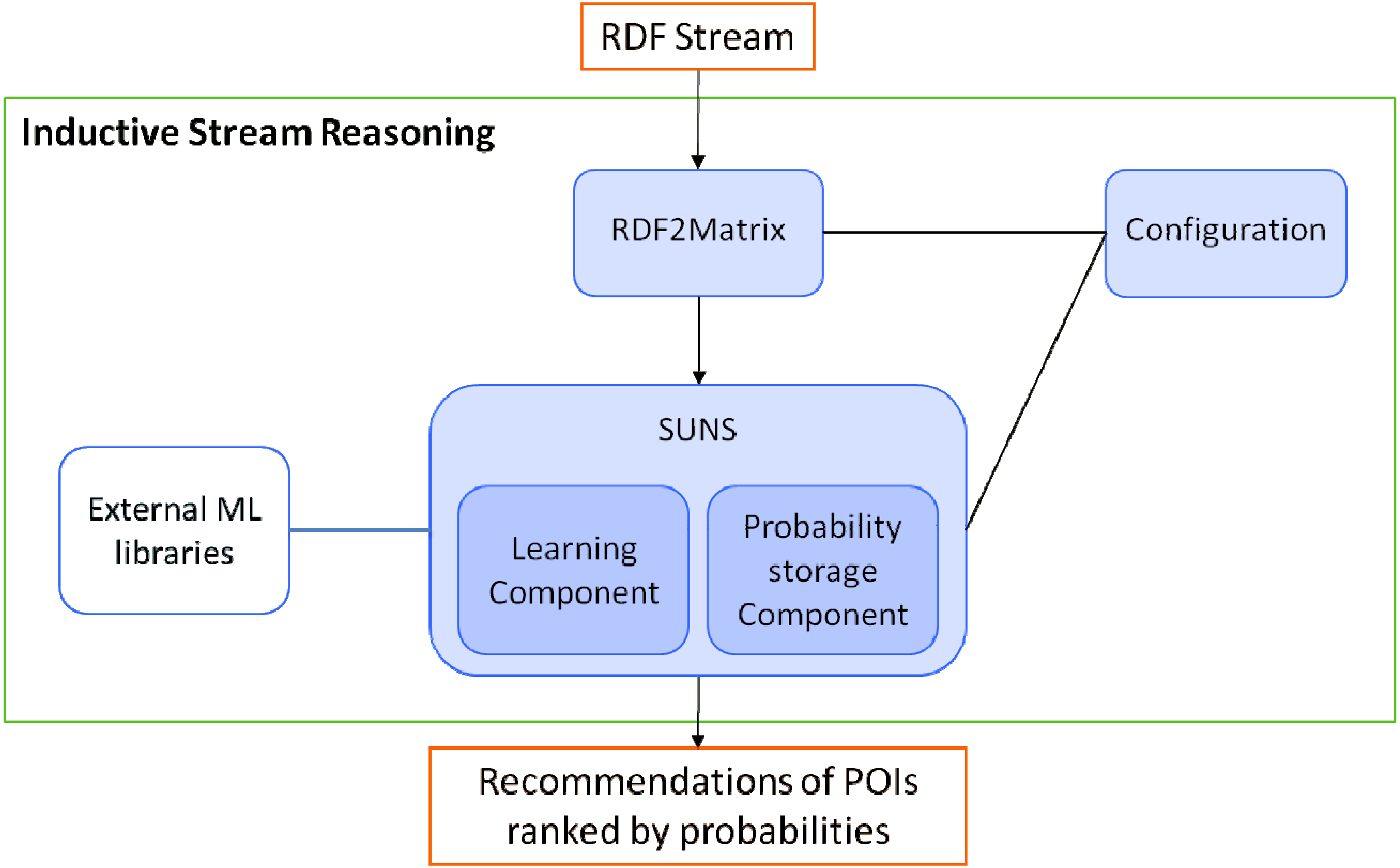
Let's watch the movie

[Celino et al., 2011]

Opinion Mining



SUNS (Statistical Unit Node Sets)



Overview

I. Accessing the Semantic Web via Statistical Machine Learning
II. Linked Data and Statistical Machine Learning
III. Querying with Statistical Machine Learning
   - Triple Prediction with Machine Learning
   - SUNS Models
   - **Extensions and Variations**
IV. Three-Way Models (Tensor Models)
   - Tensor Models for Higher Order Relations
   - Tensor Models for Collective Learning
   - RESCAL
V. Conclusions

(1) Covariance Matrices and Kernels Matrices

Recall:

$$\hat{X} = U_r \ \mathrm{diag}_\mathrm{r}\left(\frac{d_i^3}{d_i^3+\lambda}\right) V_r^T = U_r \ \mathrm{diag}_\mathrm{r}\left(\frac{d_i^2}{d_i^2+\lambda}\right) U_r^T X = X V_r \ \mathrm{diag}_\mathrm{r}\left(\frac{d_i^2}{d_i^2+\lambda}\right) \ V_r^T$$

- The second and third form can easily be applied to new entities (new *rows* in X) without the recalculation of the decomposition (Nystroem approximation)

The terms can be calculated from the decompositions:

SVD: $X = UDV^T$

Covariance decomposition: $X^T X = V \ (D^T D) \ V^T$

Kernel decomposition: $XX^T = U \ (DD^T) \ U^T$

Kernel Formulation

We can start with any appropriate kernel K and do a rank r approximation as

$$K_r = U_r \ \mathrm{diag}_\mathrm{r}\left(d_i^2\right) U_r^T$$

- (Of course we can also use our kernel $K = XX^T$)

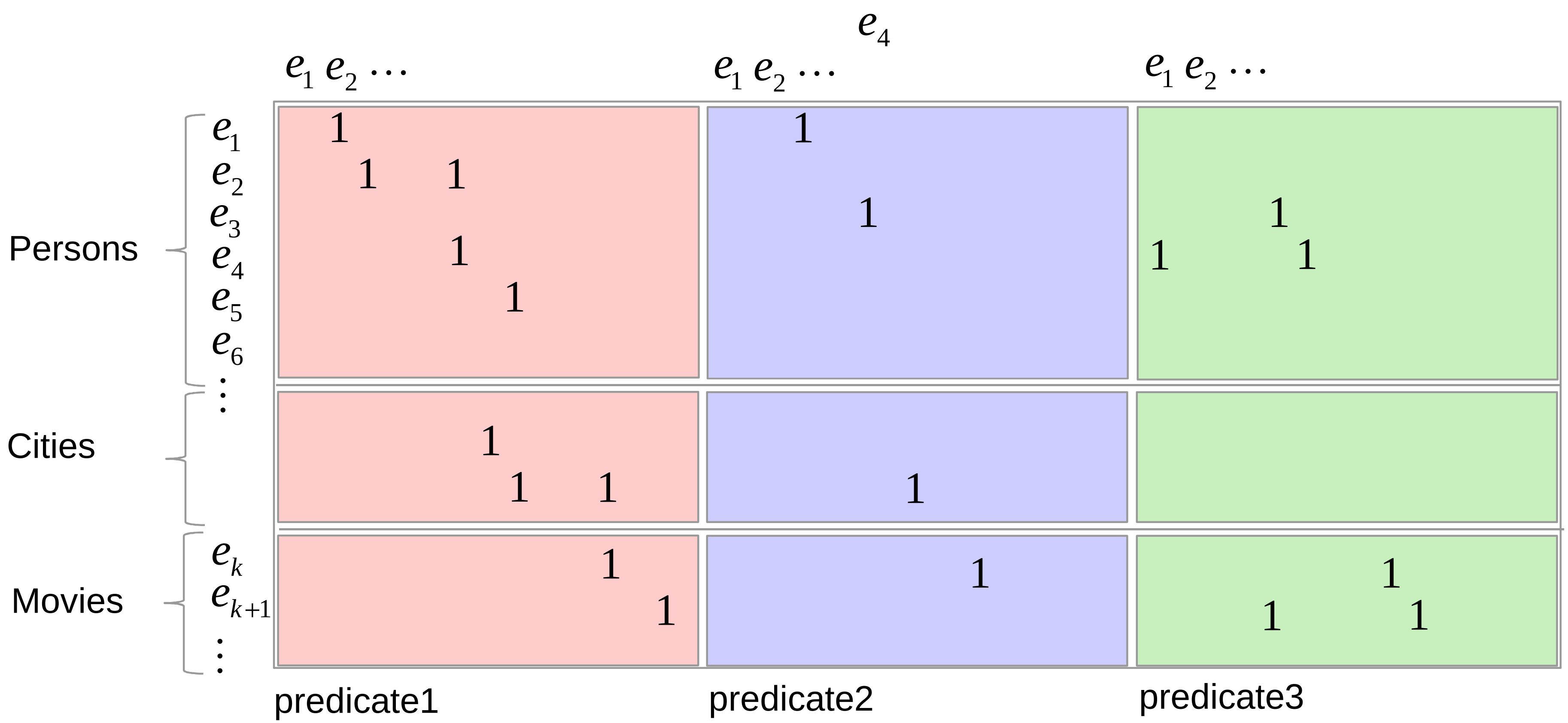
The solution then becomes $\hat{X} = K W_{LS}$

With $$W_{LS} = U \,\mathrm{diag}\left(\frac{1}{d_i^2 + \lambda}\right) U^T X$$

We can generalize to new entities using the Nystroem approximation

[Huang et al., 2012]

(2) A Global SUNS Model

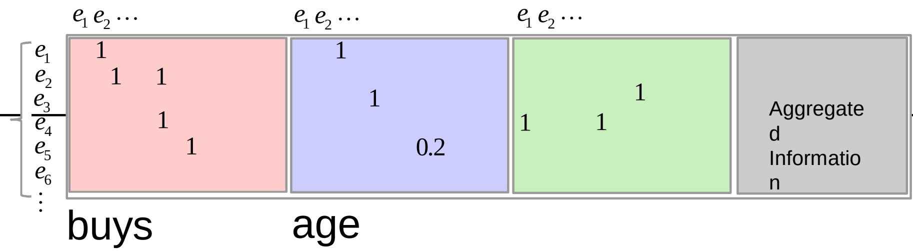


Options:

- One SUNS model for each set of entities (defined appropriately)
- Or: One global SUNS model (scalability problems)

[Jiang et al., 2012]

(3) Limitations of the SUNS model



Consider triples of the form (user, buys, movie)

The SUNS model can learn that

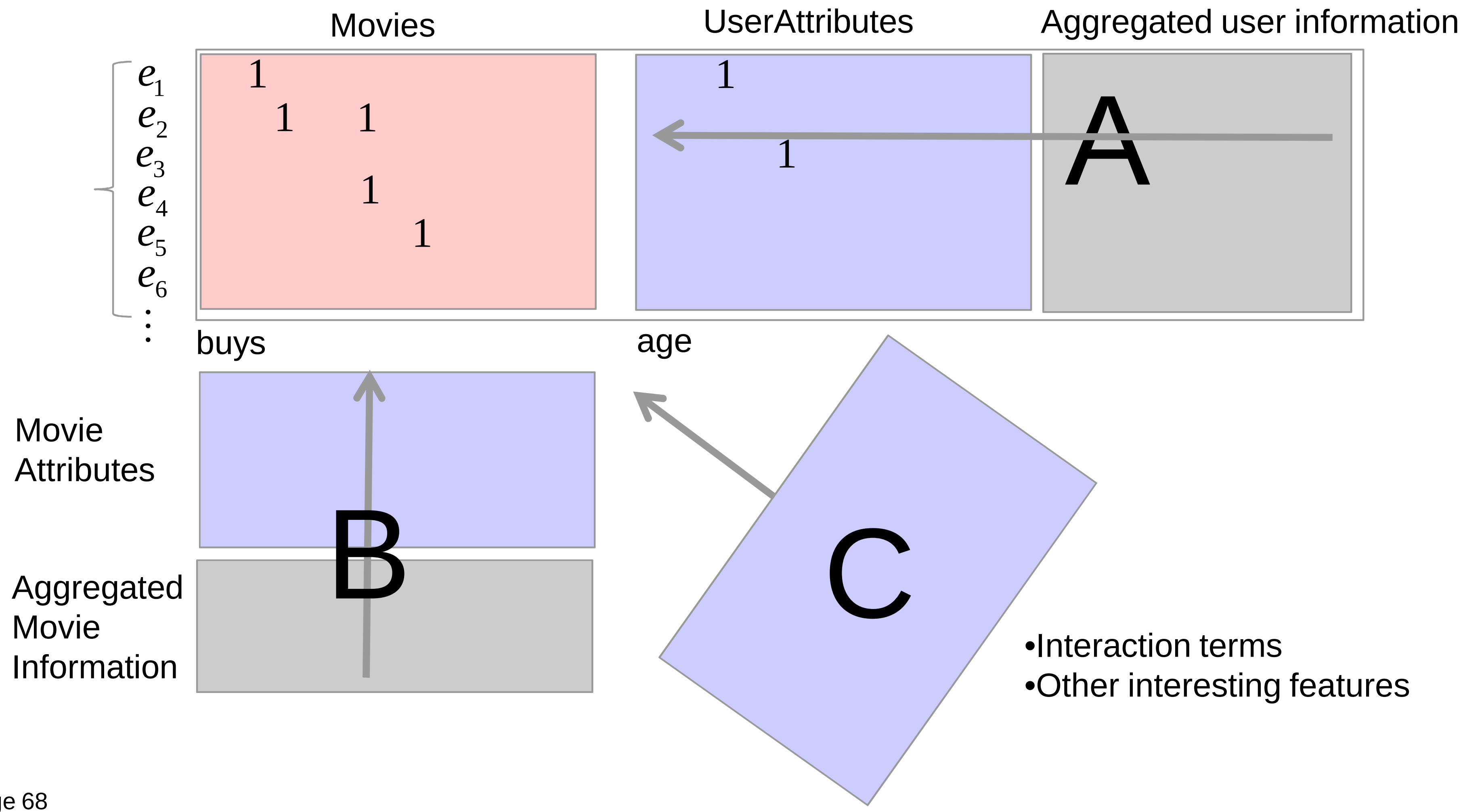
- If a user buys movie A this user tends to also buy movie B
- Movie A is often bought by young people

But it has problems learning that

- John typically buys action movies
- Young people tend to buy action movies
- People tend to buy movies produced in their country of origin
- If someone watches the trailer of a movie this user tends to buy this movie

Fixing Some of the Limitations

- Consider triples (User, buys, Movie) that is predicted based on
  - Triples and aggregated triples describing user attributes: (A)
  - Triples and aggregated triples describing movie attributes: (B)
  - Interaction terms between (A) and (B) : (C)



Solution: Global Additive Model

$$\hat{x}_{i,j} =$$

$$\sum_k w_{j,k} x_{i,k}$$ SUNS term

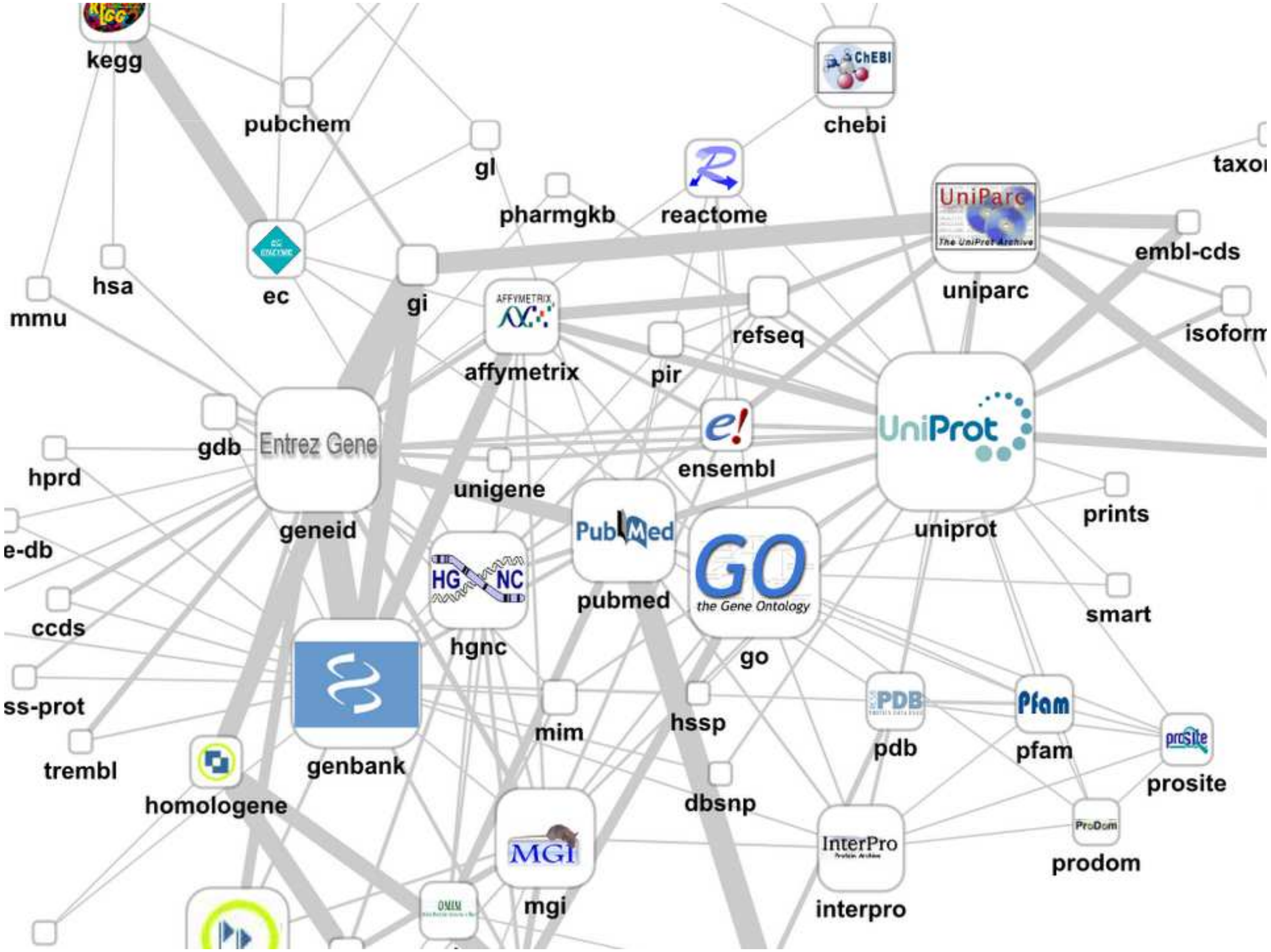
$$+\sum_{k'} w^{(A)}_{j,k'} x^{(A)}{}_{i,k'}$$ „Young people like ActionHero3“ term (A)

$$+\sum_{k} w^{(B)}_{i,k} x^{(B)}{}_{j,k}$$ Jack likes action movies term (B)

$$+\sum_{k,k'} w^{(C)}{}_{i,j,k,k'} x^{(A)}{}_{i,k} x^{(B)}{}_{j,k'}$$ Young people like action movie term (C)

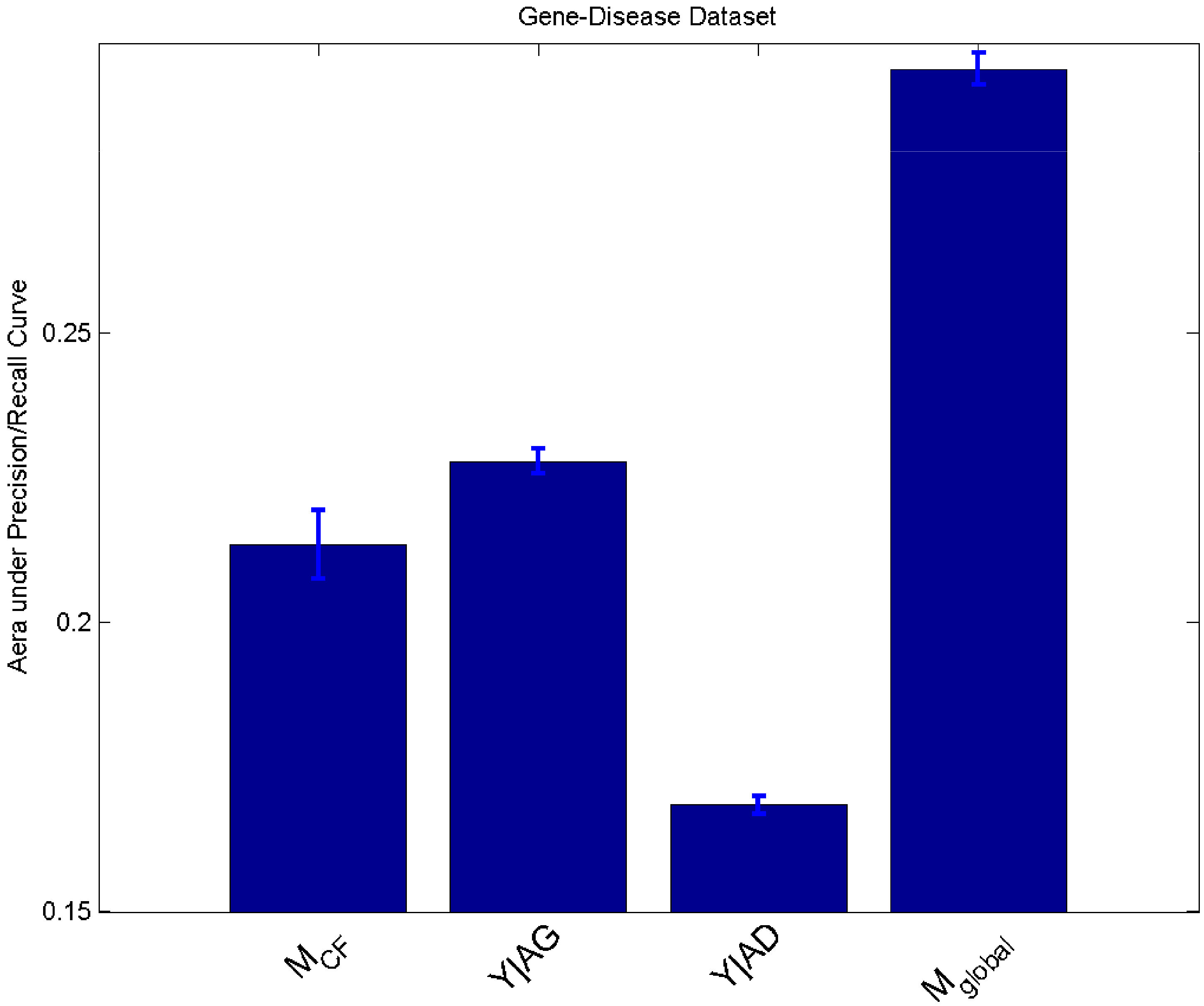
- Can be optimized via alternating least squares (including regularization and low-rank approximations)
- Same complexity as before
- The features can include keywords from textual descriptions
- Feature selection should be applied to (C)
- Many more interesting terms can be added

Predicting Relationships between Genes and Diseases



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

Gene Disease Prediction (State of the Art Quality)



(4) Including Textual and Sensory Information

Consider again triples of the form (user, buys, movie)

- What we have just obtained is a general additive model where one term corresponds to the SUNS model
- We can easily include a term that predicts a triple based in textual or other sensory information
- Often textual is available that can be useful for predicting a triple
  - Wikipedia pages describing an author
  - Literal with a brief description of a person
  - This information should be made useful for triple prediction
- Simply include a term

$$IE_{i,j}(text, v)$$

- where v could be parameters that are optimized

(5) Predicting triples: Deductive Inference, IE and Machine Learning

We are now able to combine three approaches for predicting triples

1: Ontologies and Deductive Reasoning

- Materialization (deductive closure) prior to learning

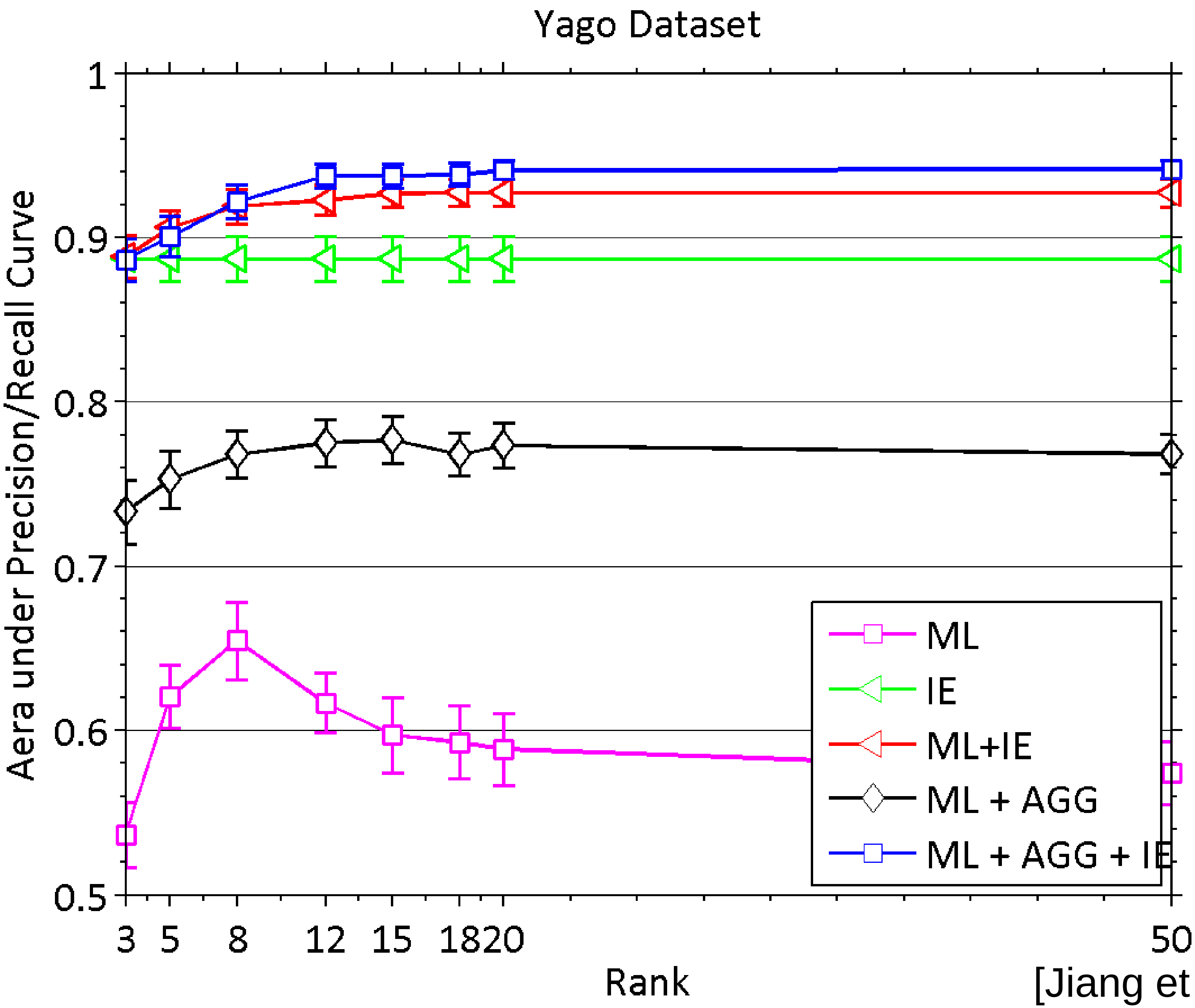
2: Information Extraction

3: **Statistical Relational Learning**

- Exploiting correlations in the target relation

[Jiang et al., 2012]

Predicting Writer's Nationalities (Yago Data)



[Jiang et al., 2012]

(6) A Probabilistic Interpretation

- A factor model as presented has a probabilistic interpretation
- To each entity, we associate a random variable which is distributed as

$$\alpha_i \propto N(0; AA^T)$$

- *A* has *r* columns and as many rows as there are columns in *X*

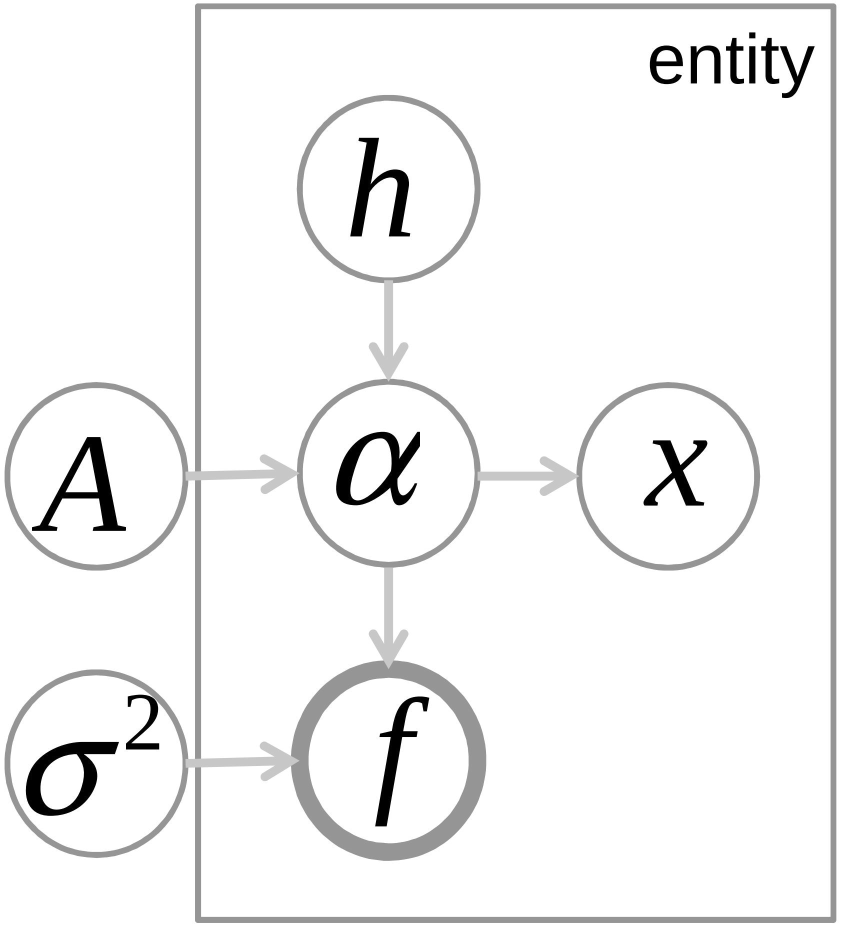
- Then $P(x_{i,j}=1)=\sigma(\alpha_{i,j})$

- Given that we have probability estimates $\hat{P}(x_{i,j}=1)$
  (from the triple store, from an expert, from IE)
  we calculate $f_{i,j}=\sigma^{-1}\left(\hat{P}(x_{i,j}=1)\right)$

And interpret these as noisy estimates $f_{i,j}=\alpha_{i,j}+\varepsilon_{i,j}$

$\varepsilon_{i,j}$ is independent Gaussian noise

Plate Model and Solution



$$\hat{\alpha}_i = U_r \,\mathrm{diag}_\mathrm{r}\left(\frac{d_i^2 - \sigma^2}{d_i^2}\right) U_r^T f_i$$

- Again, the solution is obtained via low rank matrix reconstruction
- Noise variance: σ^2

[Jiang et al., 2012]

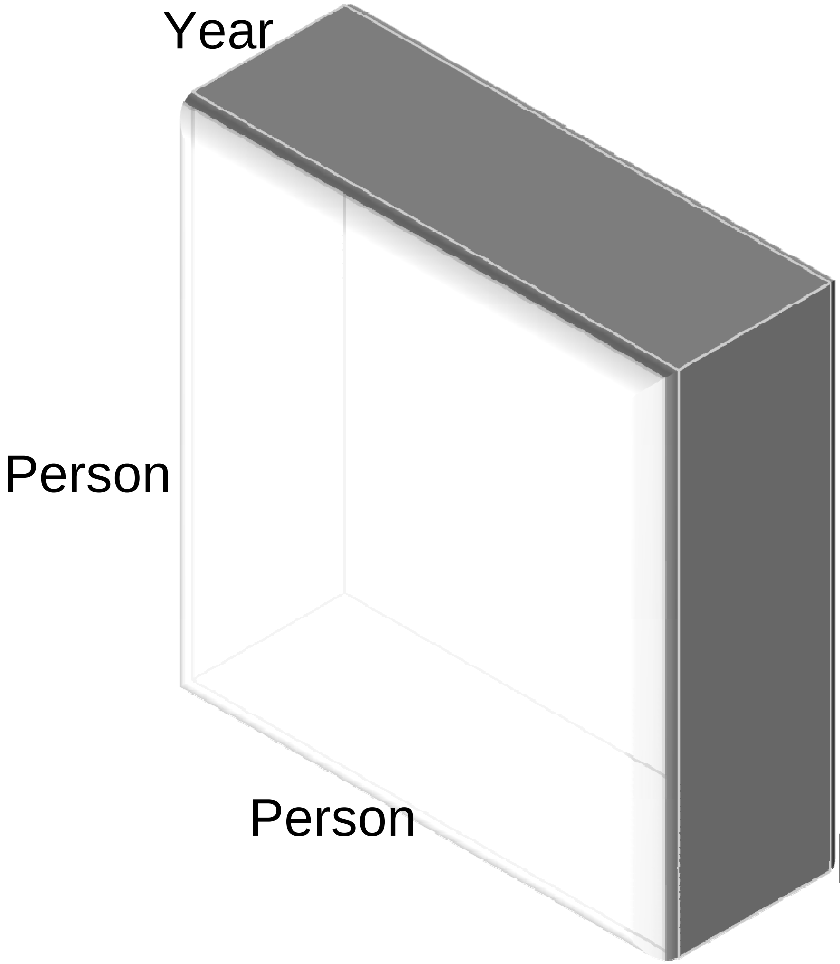
Overview

Overview

I. Accessing the Semantic Web via Statistical Machine Learning
II. Linked Data and Statistical Machine Learning
III. Querying with Statistical Machine Learning
   - Triple Prediction with Machine Learning
   - SUNS Models
   - Extensions and Variations
IV. Three-Way Models (Tensor Models)
   - **Tensor Models for Higher Order Relations**
   - Tensor Models for Collective Learning
   - RESCAL
V. Conclusions

When Binary Relations are not Enough: Ternary Relations

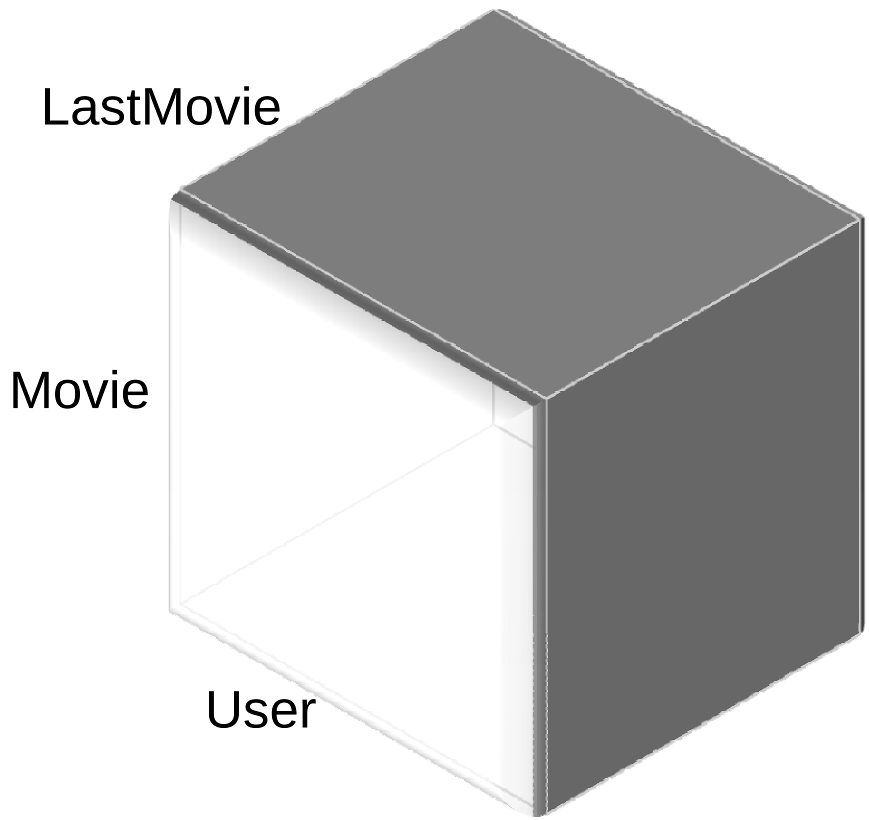
- Classical examples:
    - isFriendOf(Person, Person, Year)
- Can be described by a three-way model (tensor) and approximated using tensor factorization



[Bader et al., 2007]

Approximating a Tensor by Coupled Matrices

- watches(User, Movie, LastMovie)
- Approximated by coupled matrices [Rendle et al., 2010]
- Extended to include contextual information: [Rettinger et al., 2012]



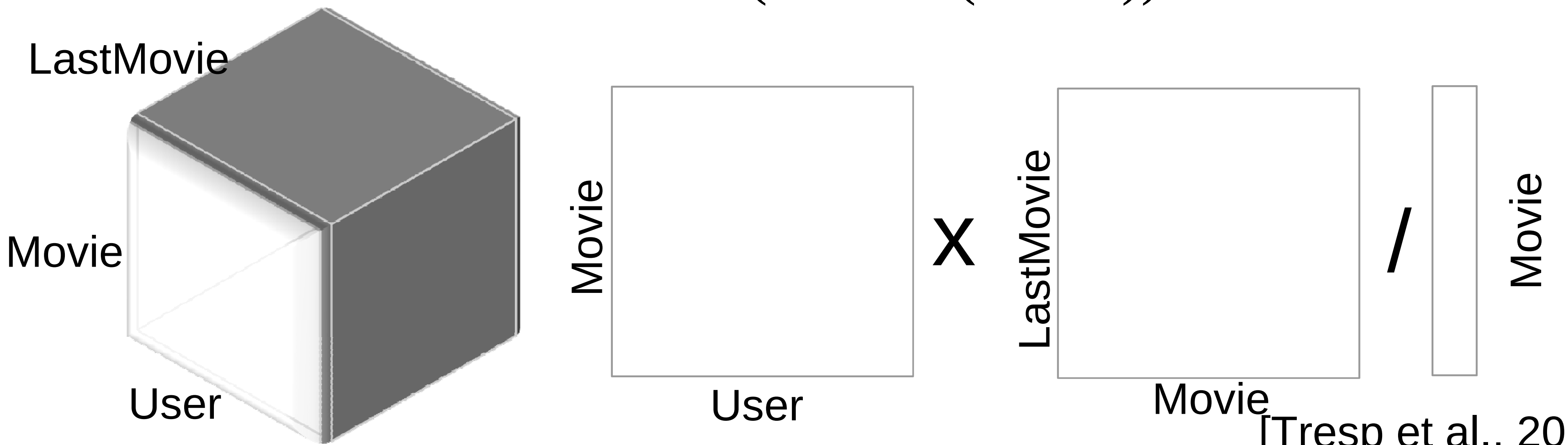
Approximating a Tensor by Coupled Matrices: Probabilistic Model

- watches(Person, Movie, LastMovie)

- Form a SUNS model for watches(User, Movie)
- Form a SUNS model for last(Movie, LastMovie)

- Then $P(watches(User, Movie, LastMovie))$

$$= \frac{P(watches(User, Movie))P(last(Movie, LastMovie))}{P(watches(Movie))}$$



[Tresp et al., 2011]

Overview

I. Accessing the Semantic Web via Statistical Machine Learning

II. Linked Data and Statistical Machine Learning

III. Querying with Statistical Machine Learning
   - Triple Prediction with Machine Learning
   - SUNS Models
   - Extensions and Variations

IV. Three-Way Models (Tensor Models)
   - Tensor Models for Higher Order Relations
   - **Tensor Models for Collective Learning**
   - RESCAL

V. Conclusions

Challenge: Collective Learning

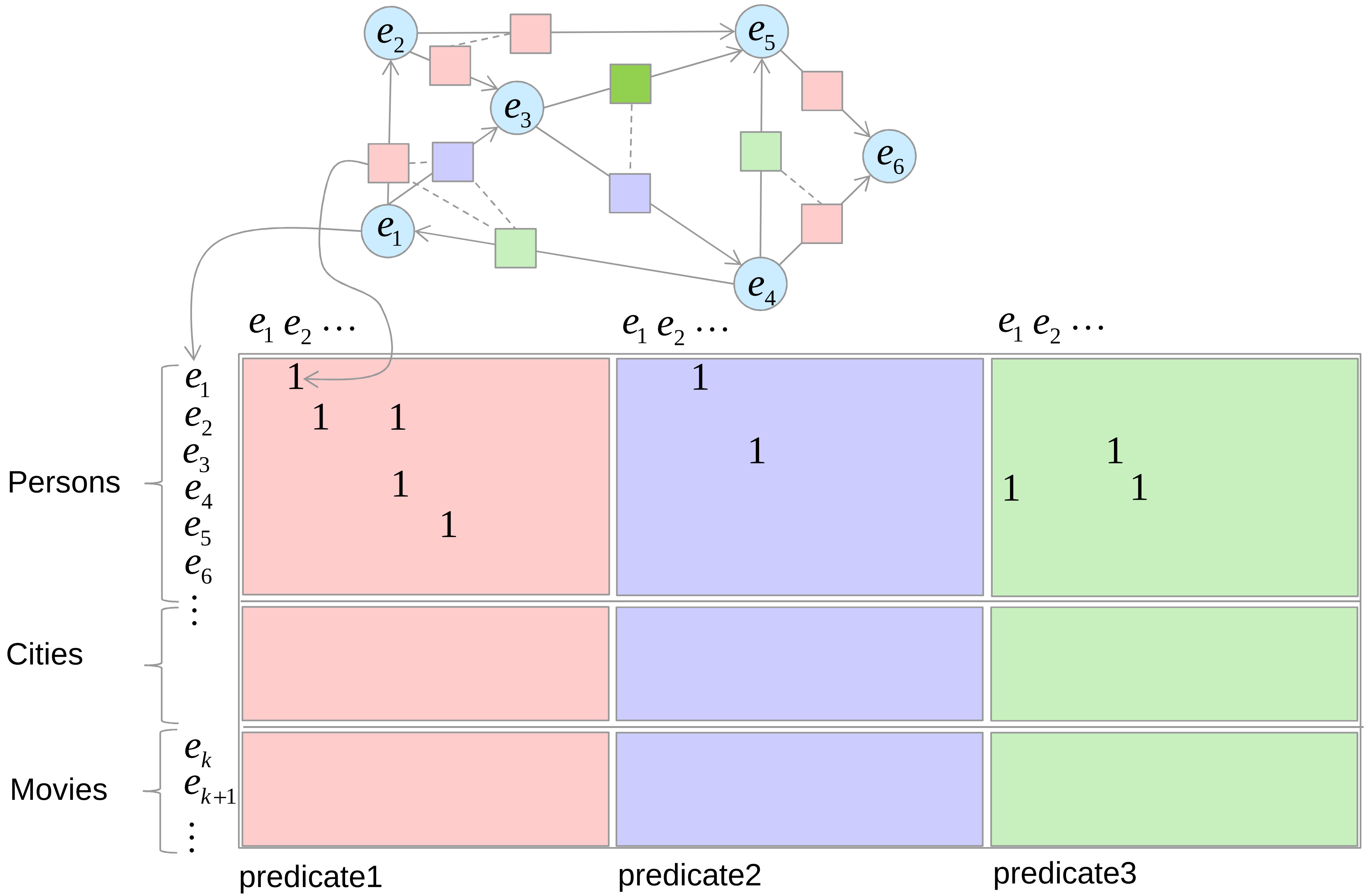
- In the SUNS model the dependencies were between all triples with the same subject entity
- Additional information was supplied by aggregation and by sensory input

- Collective learning is a form of relational learning where information distant in the graph can be made useful
    - Classical examples:
        - *If my friends are rich, I am likely also* rich
        - *If any of my ancestors was rich, I am likely also rich ….*

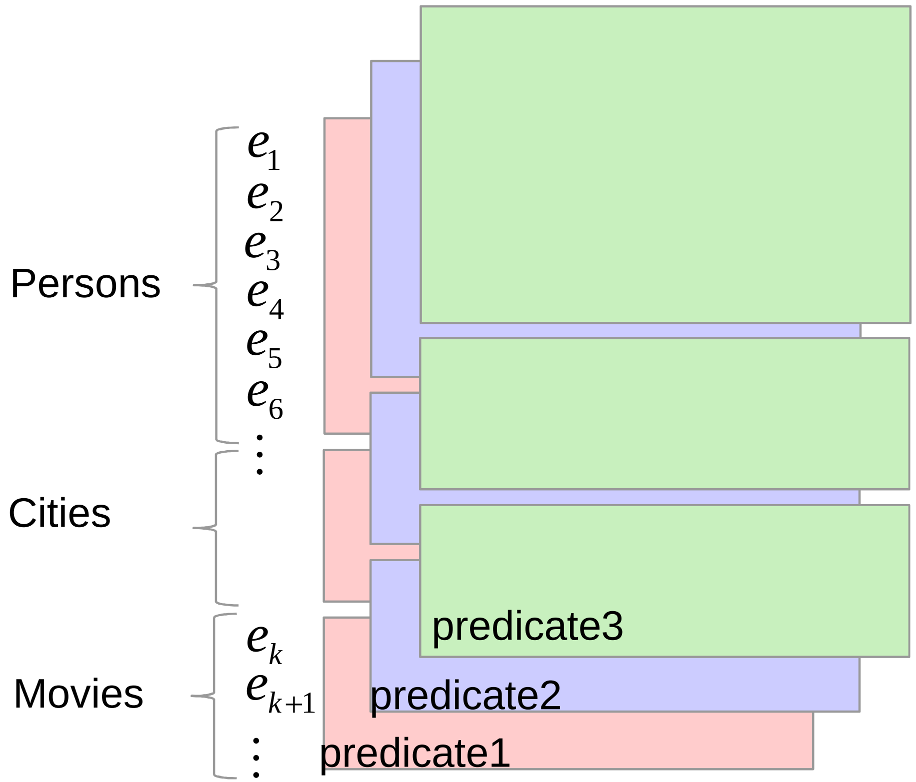
- To address these types of issues we have developed the RESCAL tensor model

[Nickel et al. 2011] [Nickel et al. 2012]

Recall: From RDF to a Data Matrix

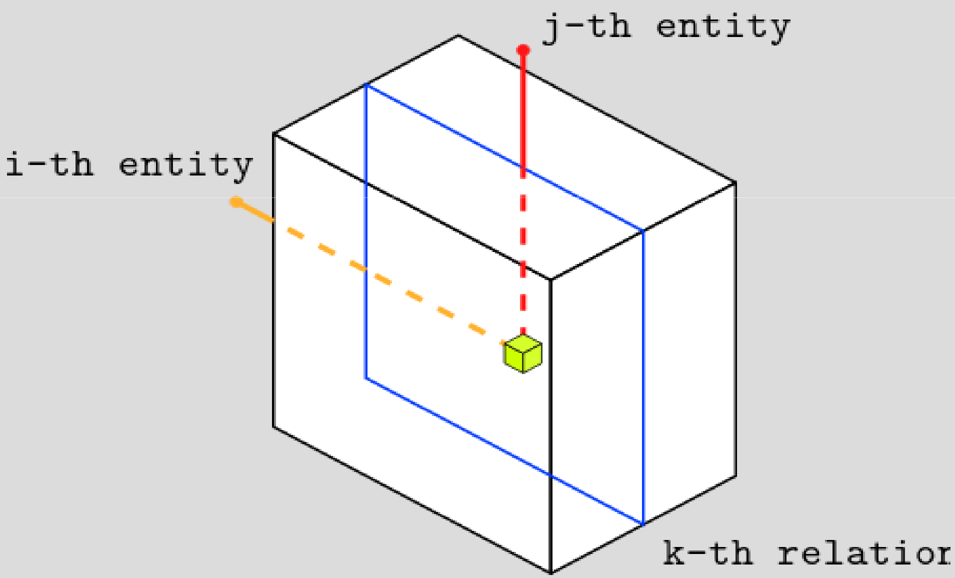


Now We Stack Relational Slices on Top of Each Other



- We obtain a three-way tensor where
  - The subject mode is indexed by all entities in a domain
  - The object mode is also indexed by all entities in a domain
  - The relation mode is indexed by the predicates in a domain

Modeling an RDF Triple Store as a Three-Way Tensor



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

Overview

I. Accessing the Semantic Web via Statistical Machine Learning

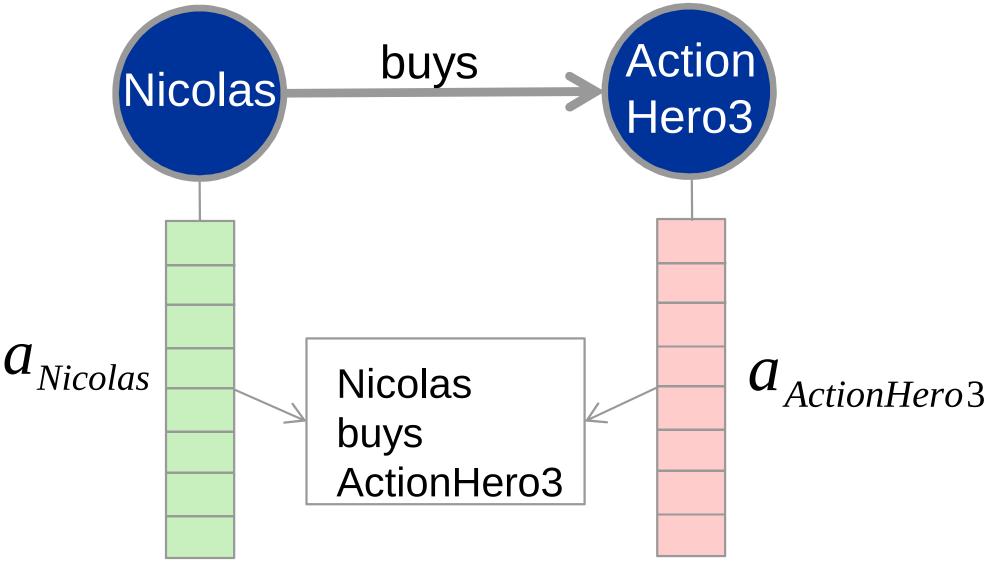
II. Linked Data and Statistical Machine Learning

III. Querying with Statistical Machine Learning
   - Triple Prediction with Machine Learning
   - SUNS Models
   - Extensions and Variations

IV. Three-Way Models (Tensor Models)
   - Tensor Models for Higher Order Relations
   - Tensor Models for Collective Learning
   - **RESCAL**

V. Conclusions

Another Look at Factorization



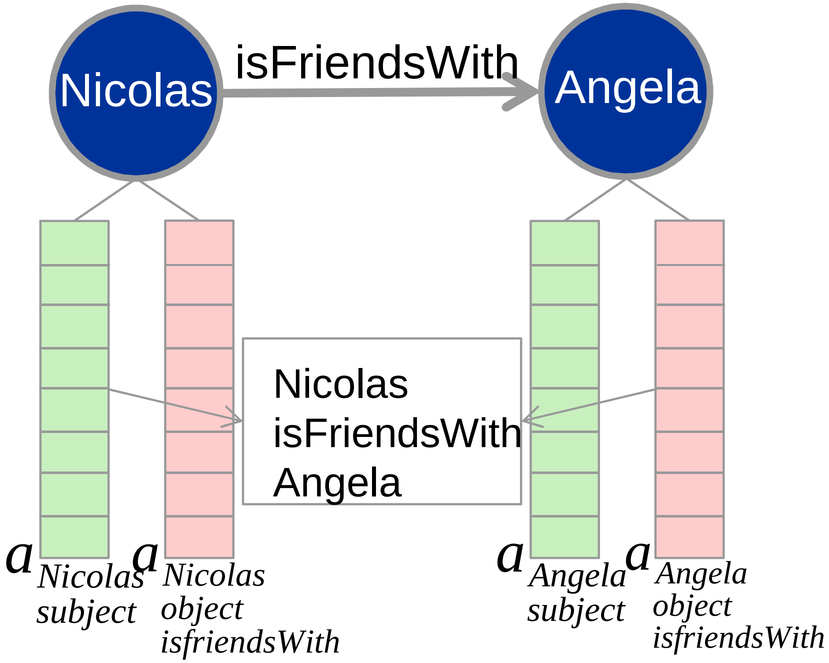
$$\hat{x}_{Nicolas,ActionHero3} = a^T_{Nicolas}\ a_{ActionHero3}$$

One solution is based on the SVD: $X = UDV^T$

$a^T_{Nicolas}$ is corresponding row of $U\sqrt{D}$

$a^T_{ActionHero3}$ is corresponding row of $V\sqrt{D}$

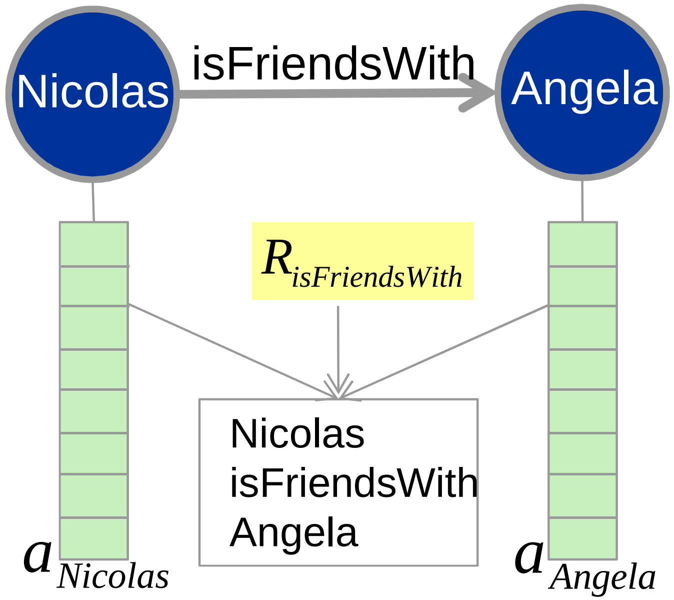
SUNS: Different Representations in Different Roles



$$\hat{x}_{\substack{Nicolas\\isFriendsWith\\Angela}} = a^T_{\substack{Nicolas\\subject}} \; a_{\substack{Angela\\object\\isFriendsWith}}$$

Information flow is limited

RESCAL Solution



- Each entity has only one internal representation

$$a_{Entity}$$

- Each predicate has its own interaction matrix

$$R_{predicate}$$

$$\hat{x}_{\substack{Nicolas\\ isFriendsWith\\ Angela}} = a^T_{Nicolas}\, R_{isFriendsWith}\, a_{Angela}$$

$$\hat{x}_{\substack{Angela\\ isFriendsWith\\ Nicolas}} = a^T_{Angela}\, R_{isFriendsWith}\, a_{Nicolas}$$

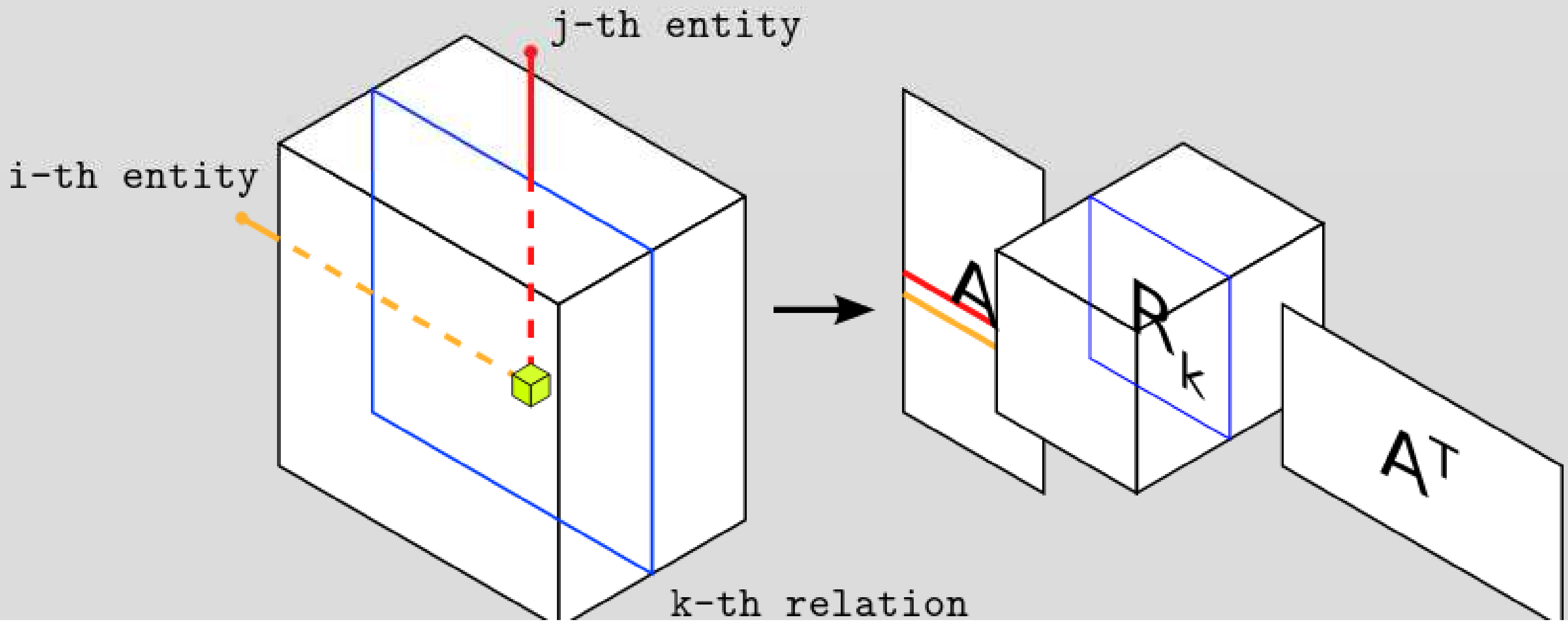
$$\hat{x}_{\substack{Nicolas\\ buys\\ ActionHero3}} = a^T_{Angela}\, R_{buys}\, a_{ActionHero3}$$

RESCAL Factorization

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

$A \in \mathbb{R}^{n \times r}$ 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

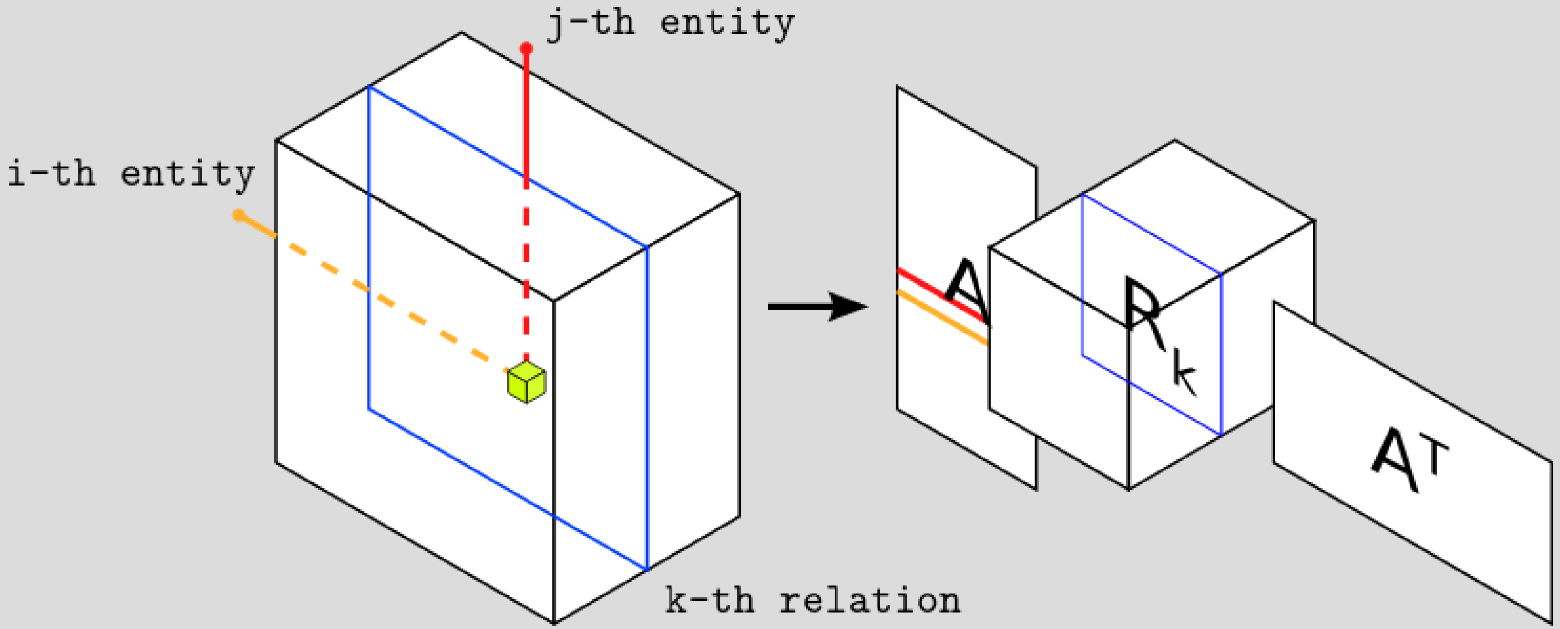


Tasks

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

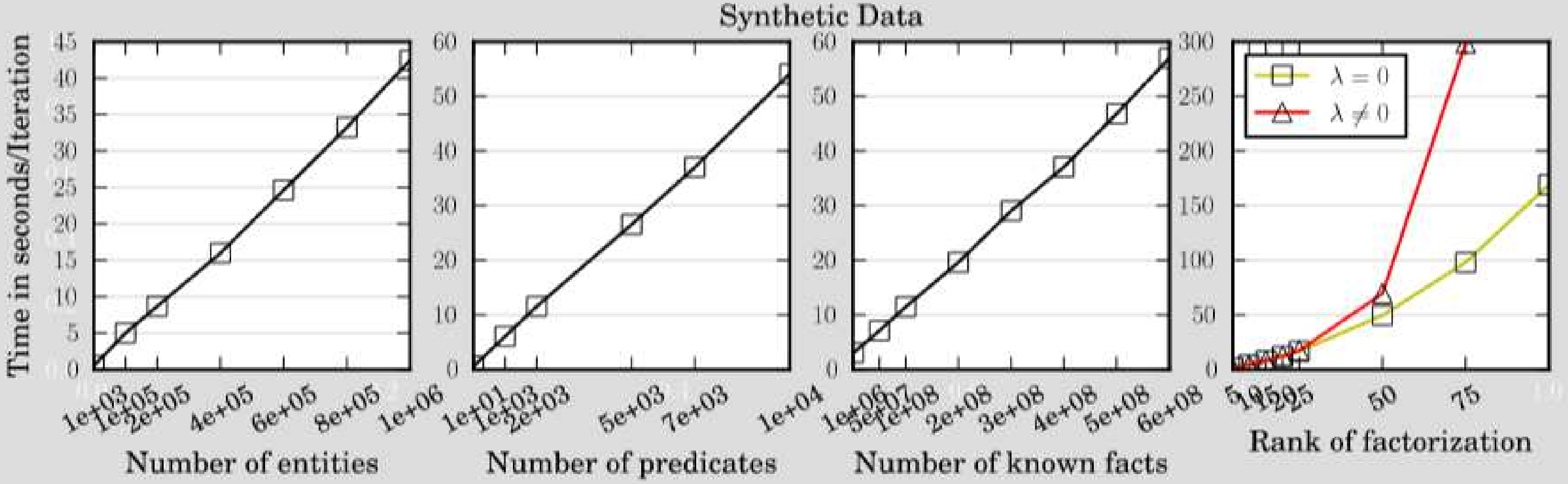


Scalability

Sparse implementation is very scalable

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

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



Factorizing YAGO 2

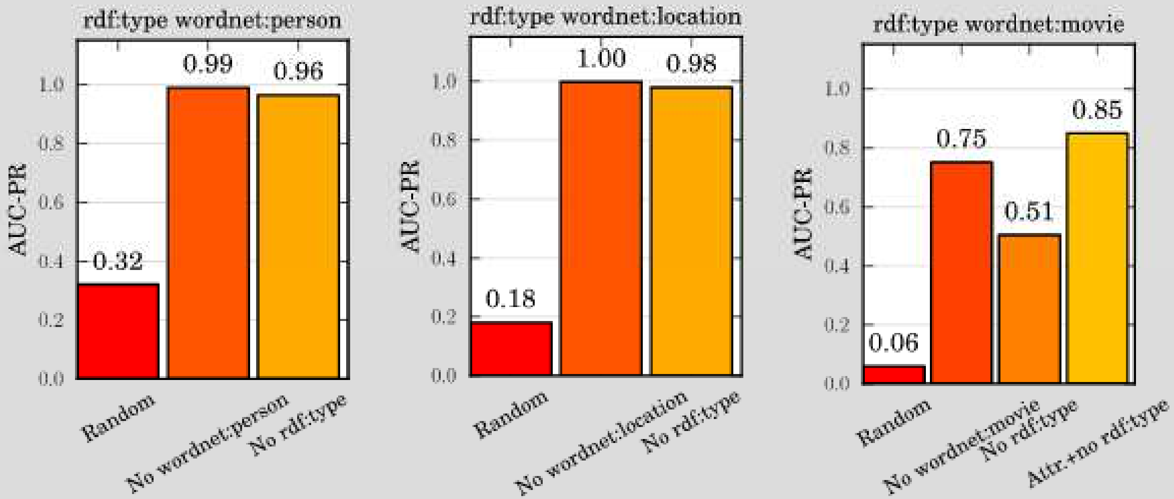
core ontology

$\mathbf{2.6}$ million entities
$\mathbf{340,000}$ classes
$\mathbf{87}$ predicates
$\mathbf{71}$ million known facts

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

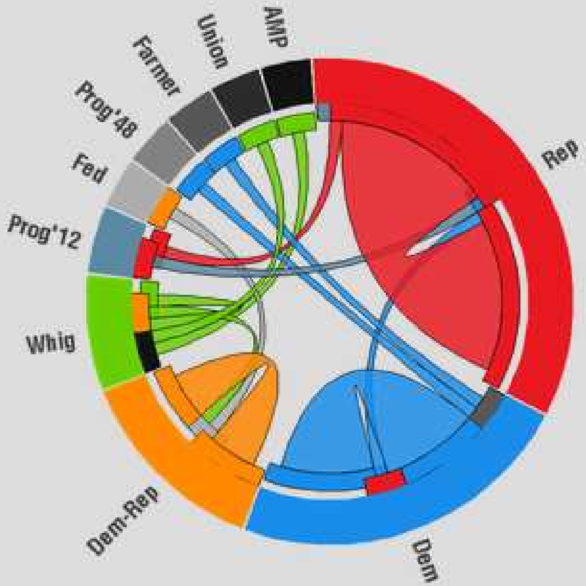
Type Prediction

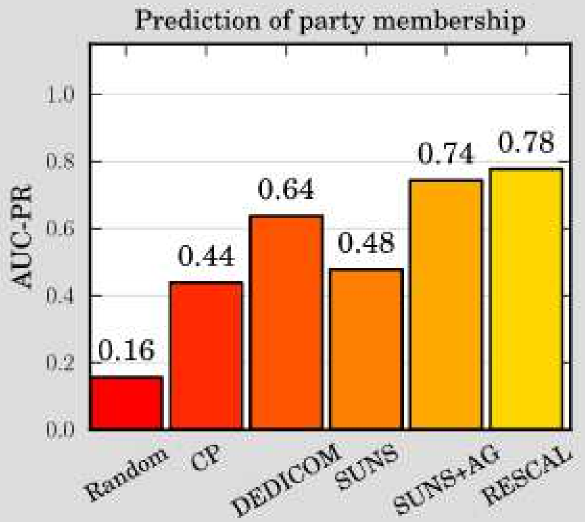
Experiment: Predict *globally* `rdf:type` triples for various classes



US-Presidents Example

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

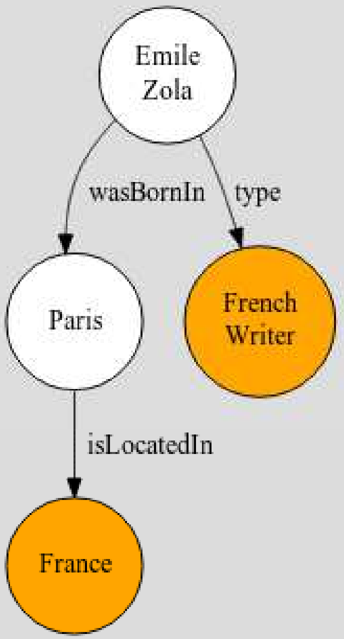


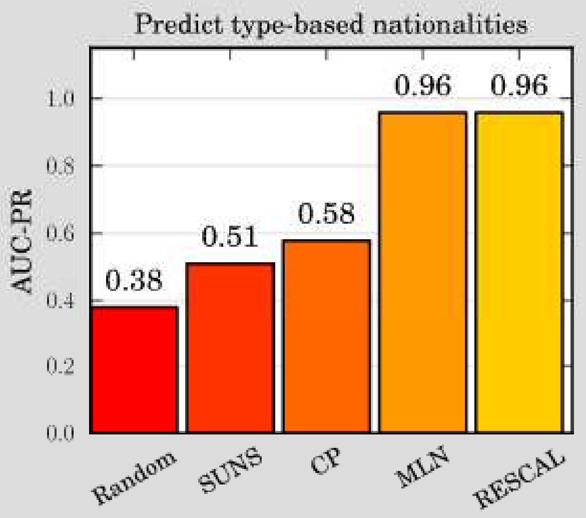


Data extracted from DBPedia, contains only the relations
`presidentOf, vicePresidentOf, partyOf`

Writer's Nationality

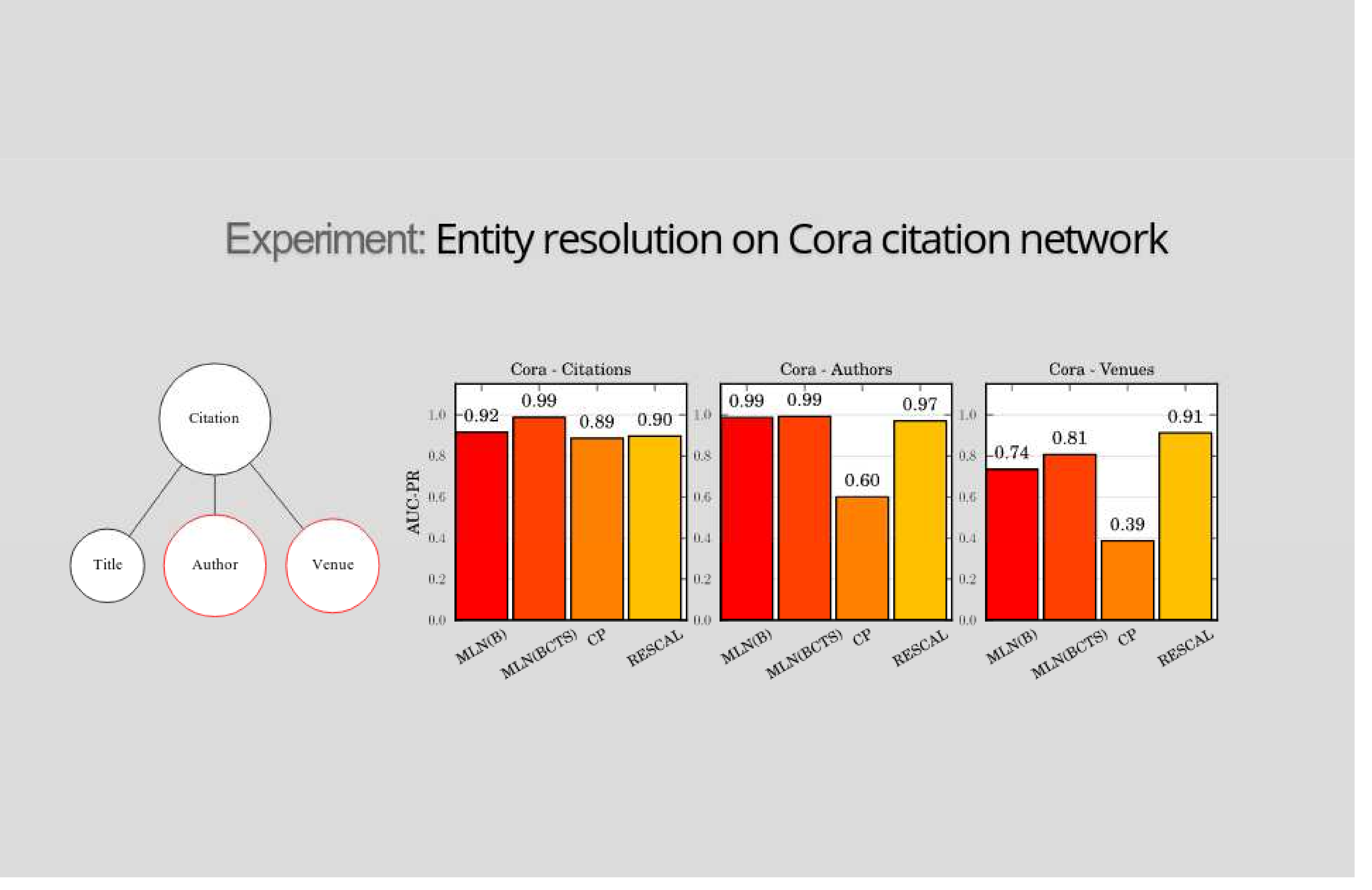
Experiment: Predict nationality-based `rdf:type` for writers in Yago 2





Collective learning task, due to typical modeling in RDF

Cora Citation Network



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

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

V. Conclusions

- We discussed Statistical Machine Learning for Linked Data
- The perspective is: querying Linked Data with Machine Learning
- We presented a number or approaches based on matrix factorization and tensor factorization to predict triples not explicitly in the data base
- We demonstrate how to include sensor information and how to combine it with deductive reasoning
- The tensor models are particularly suitable to exploit collective learning
- We demonstrated how the learned triples can be integrated into an extended SPARQL query

Conclusions (part 2)

- We believe that Statistical Machine learning ultimately should support decisions (and should not as much be about learning some ground truth)
- Vladimir Vapnik's prinicple:
  - "When solving a problem of interest, do not solve a more general [more difficult] problem as an intermediate step. Try to get the answer that you really need but not a more general one."
- In so many ways, reality is ambiguous
- It's all about making the right decisions!

If You Want More

- Tuesday 17:00 at Machine Learning I:
  - Xueyan Jiang, Yi Huang, Maximilian Nickel, and Volker Tresp. Combining Information Extraction, Deductive Reasoning and Machine Learning for Relation Prediction.

- ISWC Tutorial
  - Learning on Linked Data: Tensors and their Applications in Graph-Structured Domains

- RESCAL available at: http://www.cip.ifi.lmu.de/~nickel/
- SUNS plugins: http://wiki.larkc.eu/LarkcPlugins

References

Reviews and Overviews

- Achim Rettinger, Uta Lösch, Volker Tresp, Claudia d’Amato, and Nicola Fanizzi. Mining the Semantic Web - Statistical learning for next generation knowledge bases. Data Mining and Knowledge Discovery (DATAMINE), Springer, 2012
- Volker Tresp, Markus Bundschus, Achim Rettinger, and Yi Huang. Towards machine learning on the semantic web. Uncertainty Reasoning for the Semantic Web I Lecture Notes in AI, Springer, 2008.

References

SUNS and Extensions

- Volker Tresp, Yi Huang, Markus Bundschus, and Achim Rettinger. Materializing and querying learned knowledge. In Proceedings of the First ESWC Workshop on Inductive Reasoning and Machine Learning on the Semantic Web (IRMLeS 2009), 2009.
- Yi Huang, Volker Tresp, Markus Bundschus, Achim Rettinger and Hans-Peter Kriegel. Multivariate structured prediction for learning on the semantic web. In Proceedings of the 20th International Conference on Inductive Logic Programming (ILP), 2010.
- Yi Huang, Maximilian Nickel, Volker Tresp and Hans-Peter Kriegel. A Scalable Kernel Approach to Learning in Semantic Graphs with Applications to Linked Data. In Proc. of the 1st Workshop on Mining the Future Internet, 2010.
- Volker Tresp, Yi Huang, Xueyan Jiang, and Achim Rettinger. Graphical Models for Relations - Modeling Relational Context. International Conference on Knowledge Discovery and Information Retrieval, 2011
- Xueyan Jiang, Yi Huang, Maximilian Nickel, and Volker Tresp. Combining Information Extraction, Deductive Reasoning and Machine Learning for Relation Prediction. In *Proceedings of the ESWC,* 2012.
- Yi Huang, Volker Tresp, Maxmilian Nickel, Achim Rettinger and Hans-Peter Kriegel A Scalable Approach for Statistical Learning in Semantic Graphs. Accepted for Publication, 2012.

References

SPARQL-ML:

- Kiefer, C., Bernstein, A., Locher, A.: Adding data mining support to sparql via statistical relational learning methods. ESWC 2008

Kernel Approaches:

- C. Cumby and D. Roth. On kernel methods for relational learning. ICML 2003
- S. Bloehdorn and Y. Sure. Kernel methods for mining instance data in ontologies. ISWC/ASWC 2007
- Claudia d'Amato, Nicola Fanizzi, Floriana Esposito: Non-parametric Statistical Learning Methods for Inductive Classifiers in Semantic Knowledge Bases. ICSC 2008
- V. Bicer, T. Tran, A. Gossen. Relational Kernel Machines for Learning from Graph-structured RDF Data. Submitted 2011
- Lösch, U., Bloehdorn, S., Rettinger, A.: Graph kernels for RDF data. In: Proceedings of the 9th Extended Semantic Web Conference (ESWC’12), Springer (2012)

IRM/IHRM:

- Achim Rettinger, Matthias Nickles, and Volker Tresp. Statistical relational learning with formal ontologies. In Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2009.

References

Tensors

- B. W. Bader, R. A. Harshman and T. G. Kolda. Temporal Analysis of Semantic Graphs using ASALSAN, ICDM 2007.
- Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing YAGO: Scalable Machine Learning for Linked Data. In Proceedings of the 21st International World Wide Web Conference (WWW1012), 2012.
- Achim Rettinger, Hendrik Wermser, Yi Huang and Volker Tresp. Context-aware Tensor Decomposition for Relation Prediction in Social Networks. Social Network Analysis and Mining (SNAM), Springer, 2012
- Maximilian Nickel, Volker Tresp and Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. In Proceedings of the 28th International Conference on Machine Learning, 2011.
- Franz, T., Schultz, A., Sizov, S., Staab, S.: Triplerank: Ranking Semantic Web data by tensor decomposition. In: 8th International Semantic Web Conference(ISWC2009). (2009)

Applications

- Irene Celino, Daniele Dell'Aglio, Emanuele Della Valle, Yi Huang, Tony Lee, Seon-Ho Kim, and Volker Tresp. Towards BOTTARI: Using Stream Reasoning to Make Sense of Location-Based Micro-Posts. In: R. Garcia-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, Springer, 2011.

References

Learning of Ontologies and Learning to serve the SW:

- Alexander Maedche, Steffen Staab: Ontology Learning. Handbook on Ontologies 2004
- Grobelnik, M., Mladenic, D.: Automated knowledge discovery in advanced knowledge management. Library Hi Tech News: Online and CD Notes 9(5) (2005)
- Staab, S., Hotho, A.: Machine learning and the semantic web. In: ICML 2005 tutorial.
- Grobelnik, M., Mladenic, D.: Knowledge discovery for ontology construction. Semantic Web Technologies 2006
- J. Lehmann. DL-Learner: Learning Concepts in Description Logics, JMLR 2009.
- Fanizzi, N., d'Amato, C., Esposito, F.: DL-FOIL: Concept learning in description logics. In: Proceedings of the International Conference on Inductive Logic Programming. (2008)

ILP

- Francesca A. Lisi, Floriana Esposito: An ILP Perspective on the Semantic Web. SWAP 2005
- N. Fanizzi, C. d'Amato, and F. Esposito. DL-FOIL: Concept learning in Description Logics. ILP2008

Datamining

- B. Berendt, A. Hotho, and G. Stumme. Towards semantic web mining. ISWC 2002
- Peter Mika: Social Networks and the Semantic Web Springer 2007