Relation Extraction with Matrix Factorization and Universal Schemas

José Luis Licón Saláiz

December 18, 2013

Albert-Ludwigs-Universität Freiburg

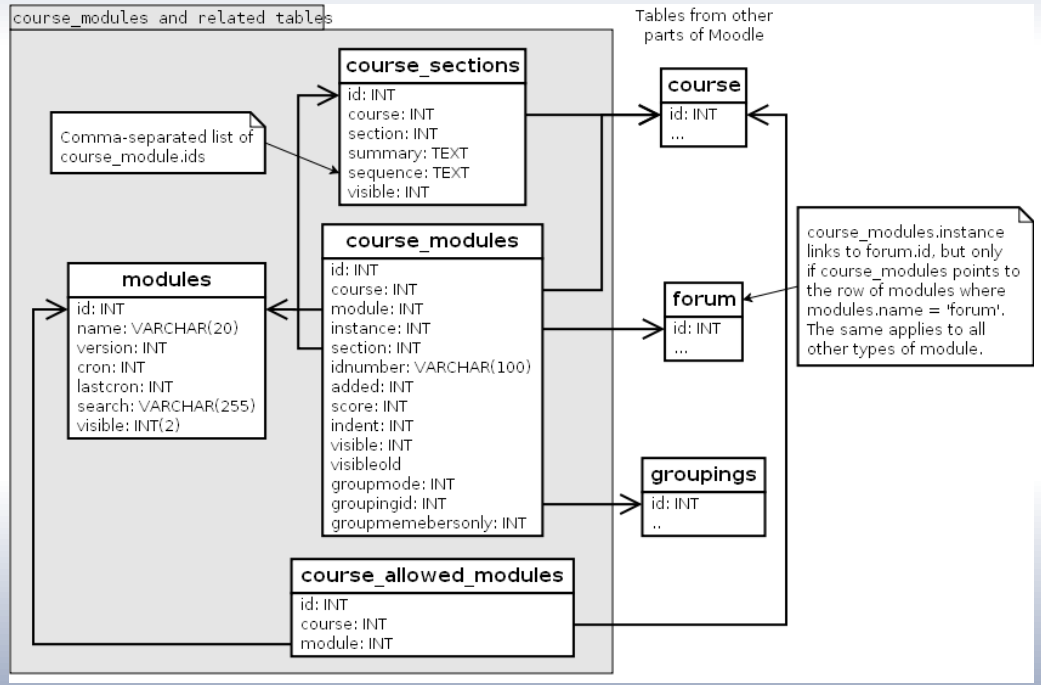
liconj@informatik.uni-freiburg.de

Overview

1. **Introduction**
   General Problem
2. **Probabilistic Databases of Universal Schema**
   Mathematical Preamble
3. **Empirical findings**
   Data
   Prediction evaluation
4. **Conclusion**

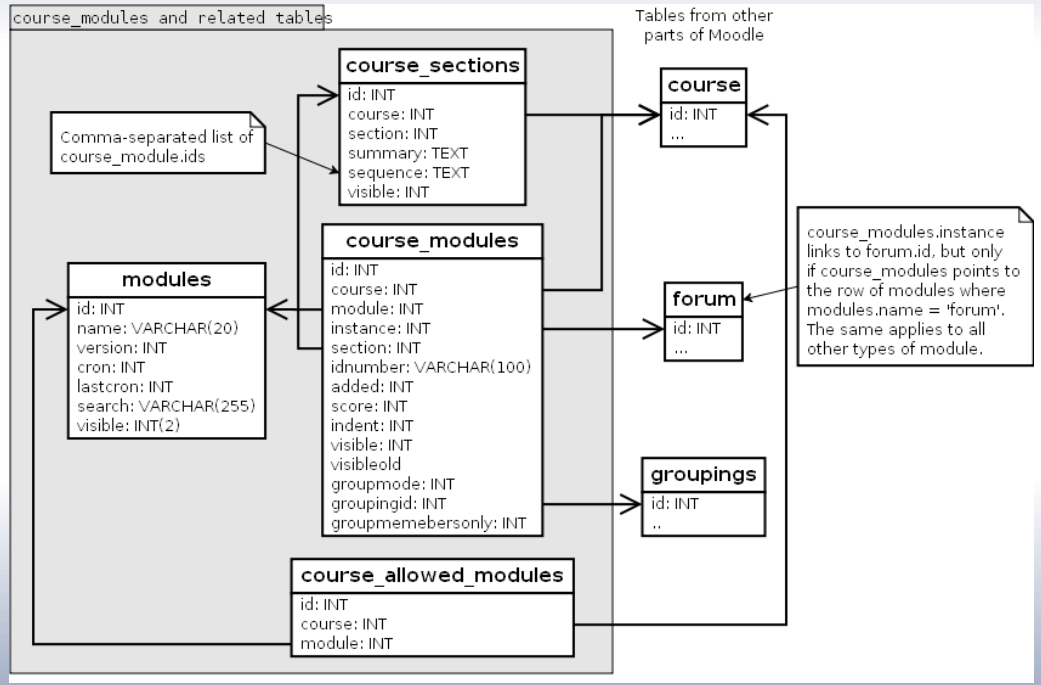
Information Extraction

- We want to transform information from a source into a target schema.
- If the source is natural language, it might contain more information than is possible to store into a given database schema.
- It can also contain knowledge that can't be reasonably put into any schema at all.



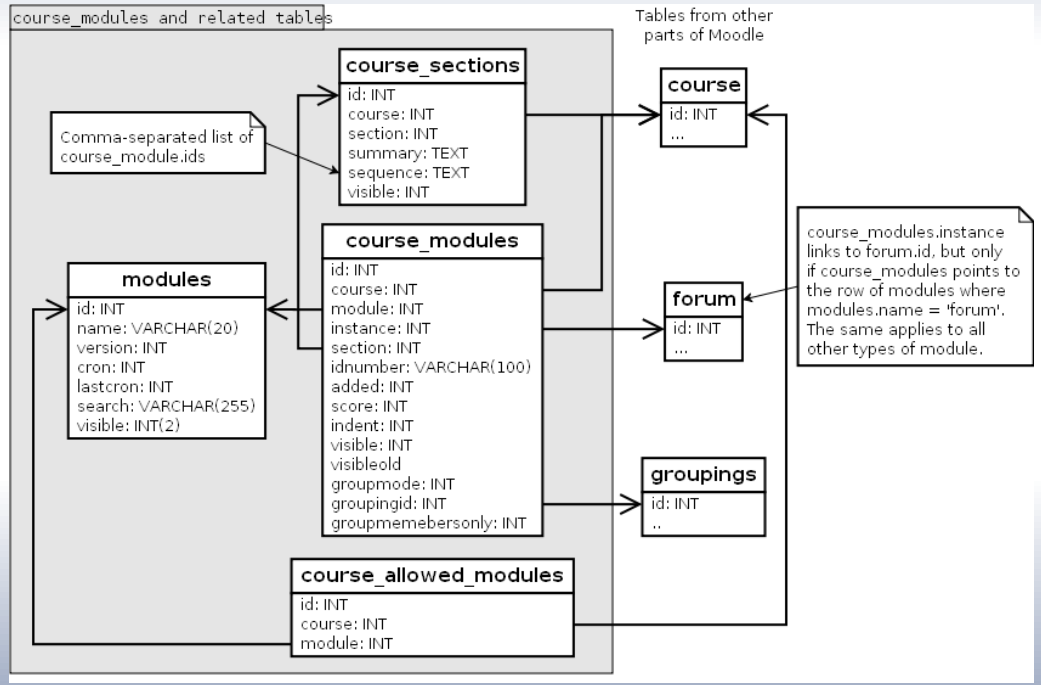
Information Extraction

- We want to transform information from a source into a target schema.
- If the source is natural language, it might contain more information than is possible to store into a given database schema.
- It can also contain knowledge that can't be reasonably put into any schema at all.



Information Extraction

- We want to transform information from a source into a target schema.
- If the source is natural language, it might contain more information than is possible to store into a given database schema.
- It can also contain knowledge that can't be reasonably put into any schema at all.



Possible Solutions

- It is possible to automatically learn a **latent** schema from the natural language source, as done in OpenIE. This has no ability to generalise - cannot predict potential facts mentioned in text. E.g. it might accurately extract the relation **Gödel-professor-at-Princeton**, but provides no information about the relation **Gödel-historian-at-Princeton**.
- Distant supervision aligns relations from structured sources with surface patterns in text to train a relation extractor. Thus it cannot predict surface patterns.
- Store the information in a probabilistic database with universal schema.

Possible Solutions

- It is possible to automatically learn a **latent** schema from the natural language source, as done in OpenIE. This has no ability to generalise - cannot predict potential facts mentioned in text. E.g. it might accurately extract the relation **Gödel-professor-at-Princeton**, but provides no information about the relation **Gödel-historian-at-Princeton**.
- Distant supervision aligns relations from structured sources with surface patterns in text to train a relation extractor. Thus it cannot predict surface patterns.
- Store the information in a probabilistic database with universal schema.

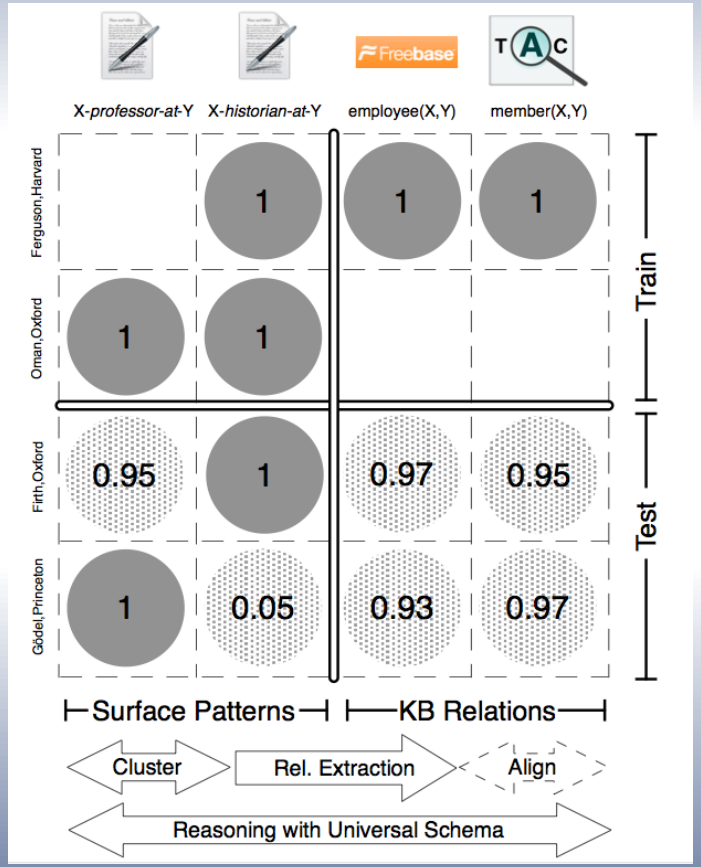
Possible Solutions

- It is possible to automatically learn a **latent** schema from the natural language source, as done in OpenIE. This has no ability to generalise - cannot predict potential facts mentioned in text. E.g. it might accurately extract the relation **Gödel-professor-at-Princeton**, but provides no information about the relation **Gödel-historian-at-Princeton**.
- Distant supervision aligns relations from structured sources with surface patterns in text to train a relation extractor. Thus it cannot predict surface patterns.
- Store the information in a probabilistic database with universal schema.

Relation Extraction With Universal Schema

Hypothesis

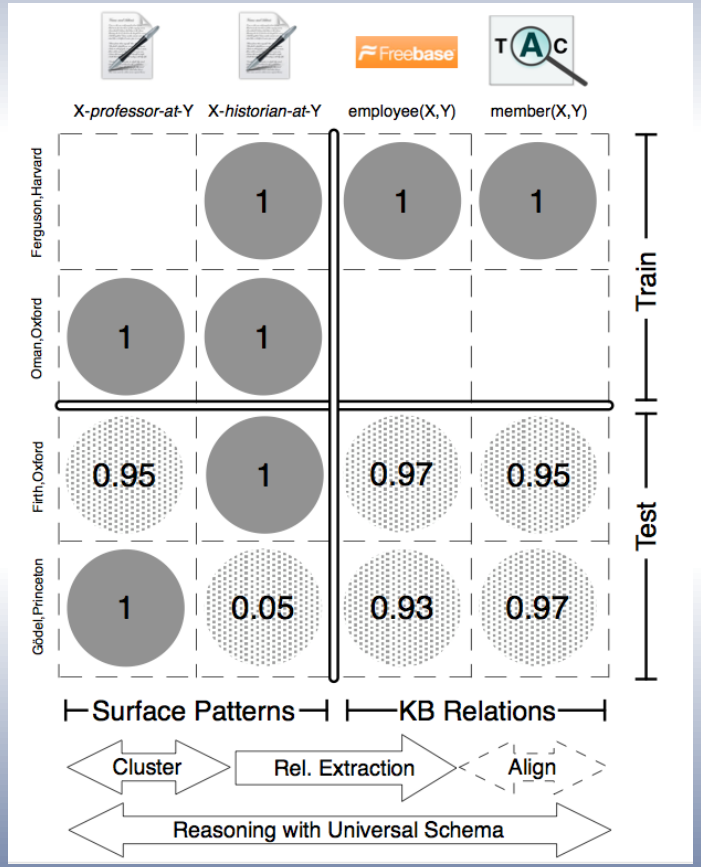
- The schema is now the **union** of all source schemas (structured and non-structured).
- Asymmetric implicature among relations must be accounted for, with help of all available information.
- Surface patterns are mapped to structured relations.
- Final objective: prediction of source data, not modelling semantic equivalence.



Relation Extraction With Universal Schema

Hypothesis

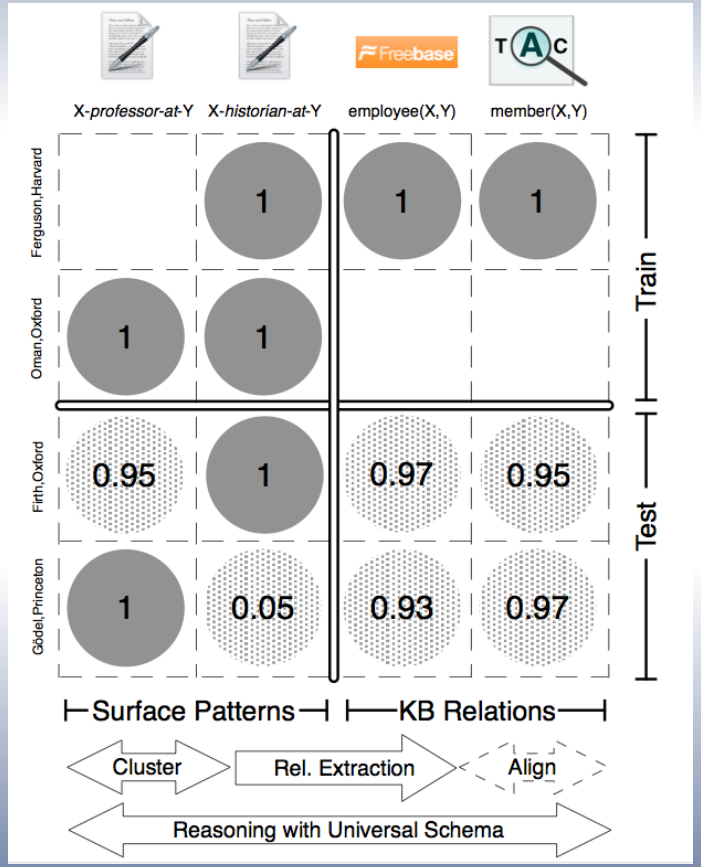
- The schema is now the **union** of all source schemas (structured and non-structured).
- Asymmetric implicature among relations must be accounted for, with help of all available information.
- Surface patterns are mapped to structured relations.
- Final objective: prediction of source data, not modelling semantic equivalence.



Relation Extraction With Universal Schema

Hypothesis

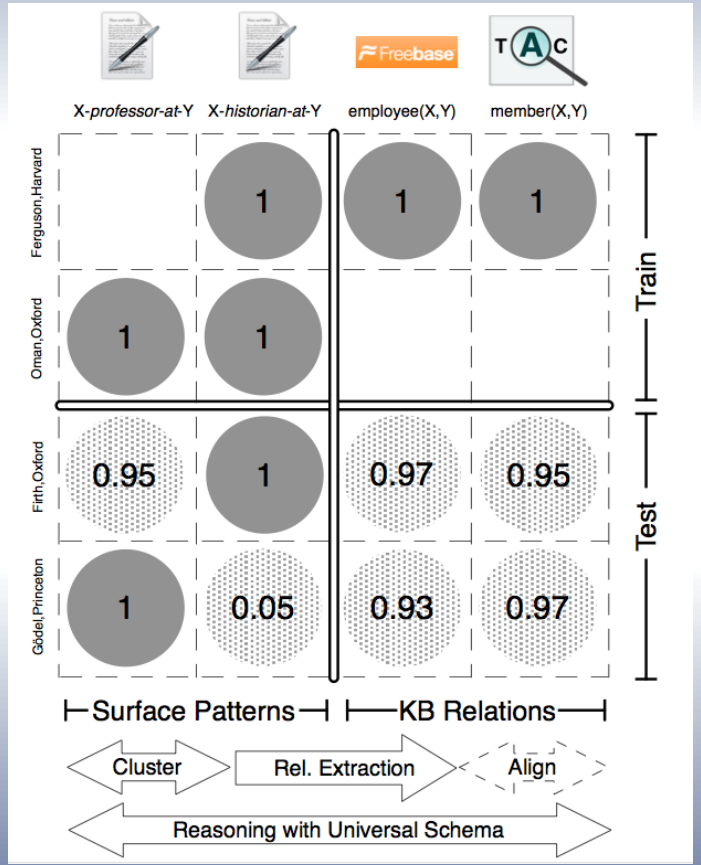
- The schema is now the **union** of all source schemas (structured and non-structured).
- Asymmetric implicature among relations must be accounted for, with help of all available information.
- Surface patterns are mapped to structured relations.
- Final objective: prediction of source data, not modelling semantic equivalence.



Relation Extraction With Universal Schema

Hypothesis

- The schema is now the **union** of all source schemas (structured and non-structured).
- Asymmetric implicature among relations must be accounted for, with help of all available information.
- Surface patterns are mapped to structured relations.
- Final objective: prediction of source data, not modelling semantic equivalence.



Probabilistic model

Let $\mathcal{R}$ be the set of binary relations present in the information sources, both structured and non-structured. E.g. **X-historian-at-Y**.

Let $\mathcal{T}$ be the set of input tuples, e.g. **<Ferguson,Harvard>**.

If $r \in \mathcal{R}$ and $t \in \mathcal{T}$, then the pair (r, t) is a **fact** or a **relation instance**. The model input is the set of all the observed facts $\mathcal{O}$. The set of observed facts for a given tuple is $\mathcal{O}_t$.

Probabilistic model

We then construct the following matrix:

$$\mathbf{Y} = y_{r,t},$$

where each entry is a random variable defined as follows:

$$y_{r,t} = \begin{cases} 1 \text{ if } (r,t) \text{ is a true relation} \\ 0 \text{ otherwise.} \end{cases}$$

It is relatively easy to fill up the matrix entries given the initial data...

Probabilistic model

We then construct the following matrix:

$$\mathbf{Y} = y_{r,t},$$

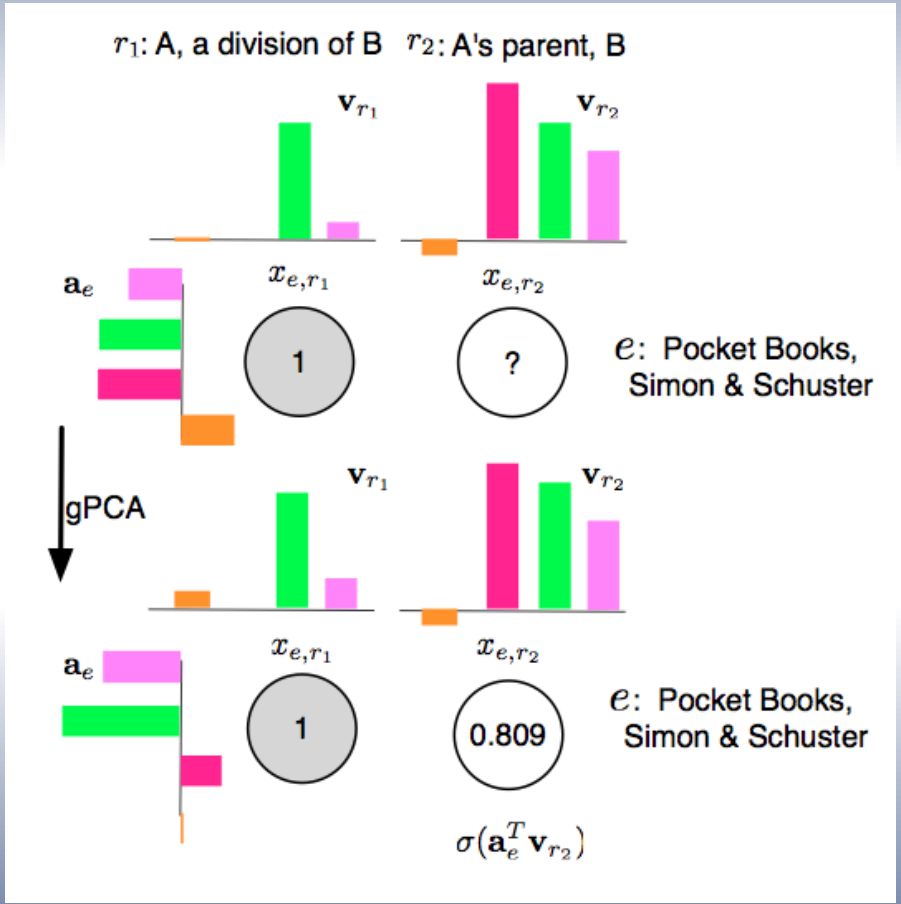
where each entry is a random variable defined as follows:

$$y_{r,t} = \begin{cases} 1 \text{ if } (r,t) \text{ is a true relation} \\ 0 \text{ otherwise.} \end{cases}$$

It is relatively easy to fill up the matrix entries given the initial data...

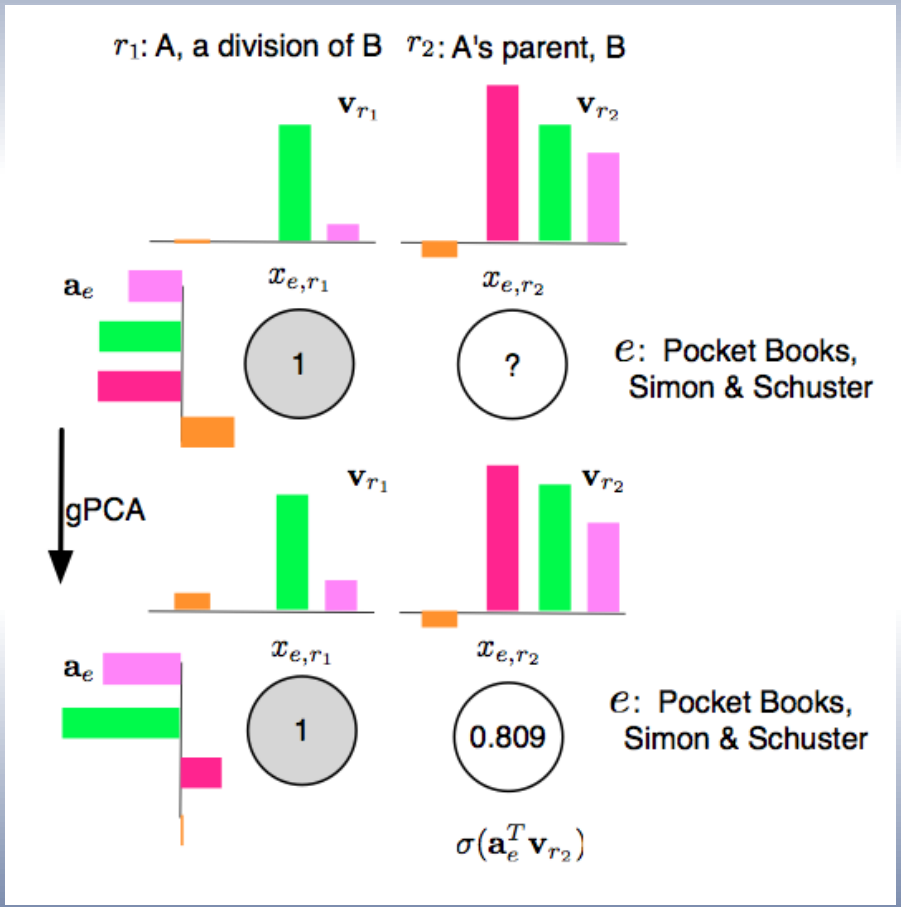
Probabilistic model

Each row will correspond to all the relations a tuple could potentially appear in, while each column will correspond to all the tuples that could potentially satisfy a given relation. The next question is: how can the empty entries be filled?



Probabilistic model

Each row will correspond to all the relations a tuple could potentially appear in, while each column will correspond to all the tuples that could potentially satisfy a given relation. The next question is: how can the empty entries be filled?



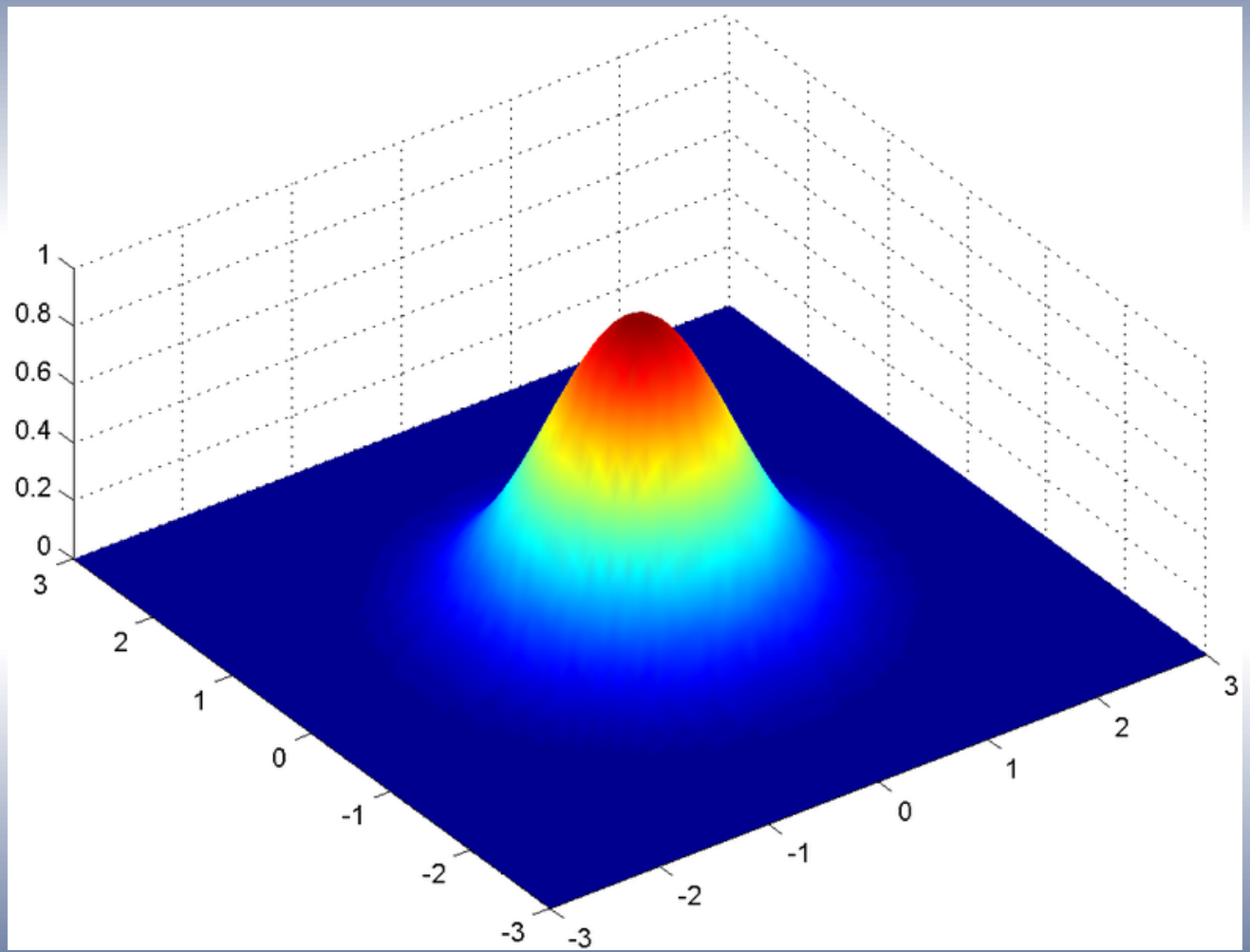
Principal Component Analysis

- PCA: for data $x_i \in \mathbb{R}^n$, find a lower-dimensional subspace such that, if θ_i are the projections of x_i on this subspace, the quantity $\sum_i \|x_i - \theta_i\|^2$ is minimised.
- Equivalently: if each x_i is seen as a random sample from a multivariate normal distribution with unit variance and mean $\mu = \theta \in \mathbb{R}^n$, PCA amounts to maximising the likelihood of the data, with the restriction that the θ_i all lie in a low-dimensional subspace.

Principal Component Analysis

- PCA: for data $x_i \in \mathbb{R}^n$, find a lower-dimensional subspace such that, if θ_i are the projections of x_i on this subspace, the quantity $\sum_i \|x_i - \theta_i\|^2$ is minimised.
- Equivalently: if each x_i is seen as a random sample from a multivariate normal distribution with unit variance and mean $\mu = \theta \in \mathbb{R}^n$, PCA amounts to maximising the likelihood of the data, with the restriction that the θ_i all lie in a low-dimensional subspace.

Principal Component Analysis



But what can be done if the data is not normally distributed?

Generalised PCA

In general, if the distribution of x_i is a member of the exponential family, it is also possible to find θ_i vectors that approximate our data within a given low-dimensional subspace. (cf. Collins et al. 2001)

- Define a basis $\{v_i \mid i = 1, \cdots, l\}$ such that $\theta_i = \sum_k \alpha_{ik} v_k$ also minimise the distances to the x_i vectors.
- We can then obtain the matrix $\Theta = AV$, with the θ_i vectors as its rows, and which represents the natural parameters for the exponential family distribution corresponding to the x_i's.

Put simply: we express the mapping of the data to the new parameter space as a matrix factorization, and maximise log-likelihood as before in this new space.

Generalised PCA

In general, if the distribution of x_i is a member of the exponential family, it is also possible to find θ_i vectors that approximate our data within a given low-dimensional subspace. (cf. Collins et al. 2001)

- Define a basis $\{v_i \mid i = 1, \cdots, l\}$ such that $\theta_i = \sum_k \alpha_{ik} v_k$ also minimise the distances to the x_i vectors.
- We can then obtain the matrix $\Theta = AV$, with the θ_i vectors as its rows, and which represents the natural parameters for the exponential family distribution corresponding to the x_i's.

Put simply: we express the mapping of the data to the new parameter space as a matrix factorization, and maximise log-likelihood as before in this new space.

Generalised PCA

In general, if the distribution of x_i is a member of the exponential family, it is also possible to find θ_i vectors that approximate our data within a given low-dimensional subspace. (cf. Collins et al. 2001)

- Define a basis $\{v_i \mid i = 1, \cdots, l\}$ such that $\theta_i = \sum_k \alpha_{ik} v_k$ also minimise the distances to the x_i vectors.
- We can then obtain the matrix $\Theta = AV$, with the θ_i vectors as its rows, and which represents the natural parameters for the exponential family distribution corresponding to the x_i's.

Put simply: we express the mapping of the data to the new parameter space as a matrix factorization, and maximise log-likelihood as before in this new space.

Generalised PCA

In general, if the distribution of x_i is a member of the exponential family, it is also possible to find θ_i vectors that approximate our data within a given low-dimensional subspace. (cf. Collins et al. 2001)

- Define a basis $\{v_i \mid i = 1, \cdots, l\}$ such that $\theta_i = \sum_k \alpha_{ik} v_k$ also minimise the distances to the x_i vectors.
- We can then obtain the matrix $\Theta = AV$, with the θ_i vectors as its rows, and which represents the natural parameters for the exponential family distribution corresponding to the x_i's.

Put simply: we express the mapping of the data to the new parameter space as a matrix factorization, and maximise log-likelihood as before in this new space.

Generalised PCA

This is important because the **Bernoulli distribution,** $B(p)$, is a member of the exponential family, its natural parameter is given by the logit function:

$$\eta = \log\left(\frac{p}{1-p}\right).$$

The inverse mapping is the logistic function:

$$p = \frac{1}{1+\exp(-\eta)}.$$

This means, in particular, that gPCA can be applied to a matrix with entries coming from a Bernoulli distribution, via the logistic function.

Natural parameter

Let (r, t) be a **fact**, for any $r \in \mathcal{R}$ and $t \in \mathcal{T}$.

Then, given a relation like **X-historian-at** and a tuple like **<Ferguson,Harvard>** , the probabilistic model should estimate

$$P(y_{r,t} = 1)$$

for this relation. To this end the logistic function is used:

$$\sigma(\theta_{r,t}) = p(y_{r,t} \mid \theta_{r,t}) = \frac{1}{1 + \exp(-\theta_{r,t})}$$

Latent Feature Model

In this model the compatibility between the relation r and tuple t is expressed as the inner product of two latent feature representations, a_r and v_t:

$$\theta^F_{r,t} = \sum_k^{K^F} a_{r,k}\, v_{t,k}.$$

In other words, this is the matrix $\Theta = AV$ that we encountered a few slides ago.

Neighborhood Model

It is also possible to define a local model as follows:

$$\theta^N_{r,t} = \sum_{(r',t)\in\mathcal{O}\setminus\{(r,t)\}} w_{r,r'}.$$

Each $w_{r,r'}$ is a measure of the **affinity** or **association strength** between two relations.

Thus it is not possible for this model to generalise beyond the preexisting knowledge base relations using the surface patterns of text data.

Entity Model

The argument slots of a relation are typed, i.e. they don't allow arbitrary types of entities.

The relation **scientist-at** might admit the tuple **<Hilbert,Göttingen>**, but of course cannot admit the tuple **<Gänselisel, Göttingen>**.

For each entity e a latent feature vector t_e is introduced. For each relation r and argument slot i a feature vector d_i is also introduced. The natural parameter definition is a measure of compatibility between the entity tuple and the relation expressed as:

$$\theta^E_{r,t} = \sum_{i=1}^{2} \sum_{k}^{K_E} d_{i,k}\, t_{t_i,k}.$$

Entity Model

The argument slots of a relation are typed, i.e. they don't allow arbitrary types of entities.

The relation **scientist-at** might admit the tuple <**Hilbert,Göttingen**>, but of course cannot admit the tuple <**Gänselisel, Göttingen**>.

For each entity e a latent feature vector t_e is introduced. For each relation r and argument slot i a feature vector d_i is also introduced. The natural parameter definition is a measure of compatibility between the entity tuple and the relation expressed as:

$$\theta^{E}_{r,t} = \sum_{i=1}^{2} \sum_{k}^{K_{E}} d_{i,k}\, t_{t_i,k}.$$

Combined Model

With all these models at our disposal, it is of course natural to combine them like so:

$$\theta^{NFE}_{r,t} = \theta^{N}_{r,t} + \theta^{F}_{r,t} + \theta^{E}_{r,t}$$

Parameter Estimation

Important observation: there are no negative facts available. Thus, finding the maximum likelihood estimates for all the model parameters could lead to the model predicting all facts to be true.

Again using an analogy to collaborative filtering, the authors employ Bayesian Personalised Ranking (cf. Rendle et al., 2009). In other words: the task is not so much prediction, as it is merely ranking.

Objective Function

Given a relation $r \in \mathcal{R}$ and the set of facts $f^+ = (r, t^+) \in \mathcal{O}$, choose all tuples t^- such that $f^- = (r, t^-) \notin \mathcal{O}$.

For each pair of facts f^+ and f^- we want $P(f^+) > P(f^-)$, which is equivalent to $\theta_{f^+} > \theta_{f^-}$.

The idea behind the objective function to be maximised is that it should be a linear combination of terms with the form $\log(\sigma(\theta_{f^+} - \theta_{f^-}))$...

Data preprocessing

Non-structured Data

- NYT annotated corpus used as source
- Training set: articles after 2000
- Test set: articles between 1990 and 1999

Structured Data

- Freebase was used as source
- 50/50 split for training and test sets

Both corpuses must be **aligned**: NER mentions in text are linked (via string-matching heuristic) to entities in FB $\Rightarrow 8k$ tuples training/test with facts mentioned in the aligned text, $\sim 200k$ tuples for which both arguments appear in Freebase, but there is no fact linking both. A sample of $10k$ of these is taken for the test set.

Relations with < 10 tuples mentioned in text are filtered out.

Data preprocessing

Non-structured Data

- NYT annotated corpus used as source
- Training set: articles after 2000
- Test set: articles between 1990 and 1999

Structured Data

- Freebase was used as source
- 50/50 split for training and test sets

Both corpuses must be **aligned**: NER mentions in text are linked (via string-matching heuristic) to entities in FB $\Rightarrow 8k$ tuples training/test with facts mentioned in the aligned text, $\sim 200k$ tuples for which both arguments appear in Freebase, but there is no fact linking both. A sample of $10k$ of these is taken for the test set.

Relations with < 10 tuples mentioned in text are filtered out.

Observed facts

Recall that the model input is a set $\mathcal{O} = \cup_t \mathcal{O}_t$ of observed facts.

We can see that

$$\mathcal{O}_t = \mathcal{O}_t^{FB} \cup \mathcal{O}_t^{PAT}.$$

Let $t = (t_1, t_2)$ be a tuple, $m = (m_1, m_2)$ a mention of this tuple in text.

The lexicalized dependency path p is extracted, and (p, t) is then added to $\mathcal{O}_t$.

For instance, the text **M1 heads M2** yields the following fact: **<M1-subj<-head->M2-obj>**.

Evaluation

For predicting Freebase relations, other methods are used as baseline:

- MI09: distantly supervised classifier (Mintz, et al. (2009)).
- YA11: a newer version of MI09 that uses preprocessed cluster features (Yao et al. (2011)).
- SU12: state of the art Multi-Instance Multi-Label system (surdeanu et al. (2012)).
- All the latent feature models are faster to train: 45 minutes at most. YA11 takes 4 hours, SU12 2 hours (on less data).

Evaluation

For predicting Freebase relations, other methods are used as baseline:

- MI09: distantly supervised classifier (Mintz, et al. (2009)).
- YA11: a newer version of MI09 that uses preprocessed cluster features (Yao et al. (2011)).
- SU12: state of the art Multi-Instance Multi-Label system (surdeanu et al. (2012)).
- All the latent feature models are faster to train: 45 minutes at most. YA11 takes 4 hours, SU12 2 hours (on less data).

Evaluation

For predicting Freebase relations, other methods are used as baseline:

- MI09: distantly supervised classifier (Mintz, et al. (2009)).
- YA11: a newer version of MI09 that uses preprocessed cluster features (Yao et al. (2011)).
- SU12: state of the art Multi-Instance Multi-Label system (surdeanu et al. (2012)).
- All the latent feature models are faster to train: 45 minutes at most. YA11 takes 4 hours, SU12 2 hours (on less data).

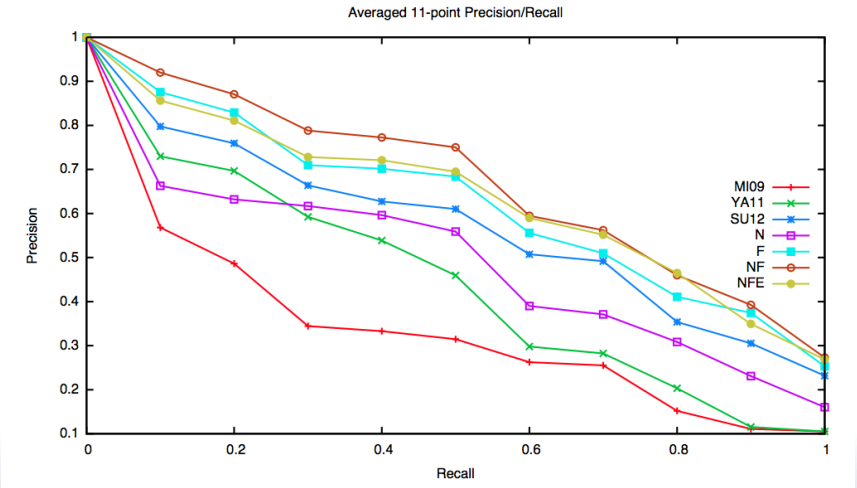
Evaluation

For predicting Freebase relations, other methods are used as baseline:

- MI09: distantly supervised classifier (Mintz, et al. (2009)).
- YA11: a newer version of MI09 that uses preprocessed cluster features (Yao et al. (2011)).
- SU12: state of the art Multi-Instance Multi-Label system (surdeanu et al. (2012)).
- All the latent feature models are faster to train: 45 minutes at most. YA11 takes 4 hours, SU12 2 hours (on less data).

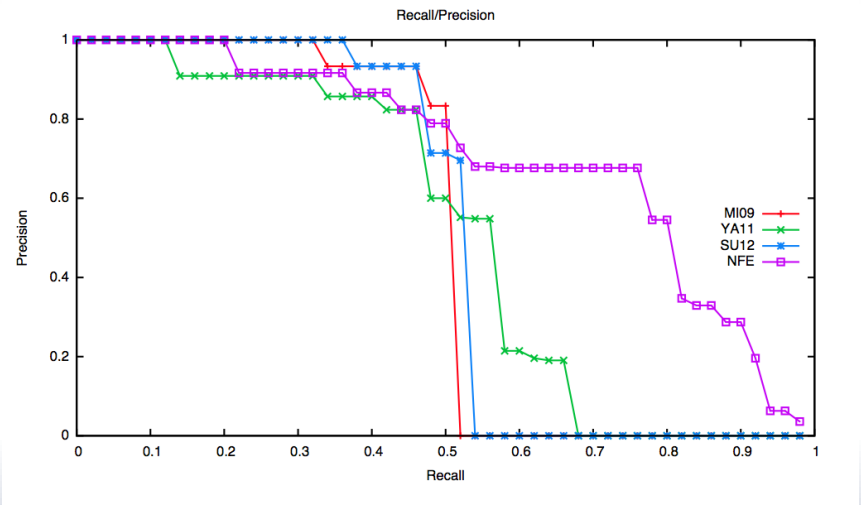
Prediction of Freebase relations

- The latent feature models outperform all others across all recall levels.



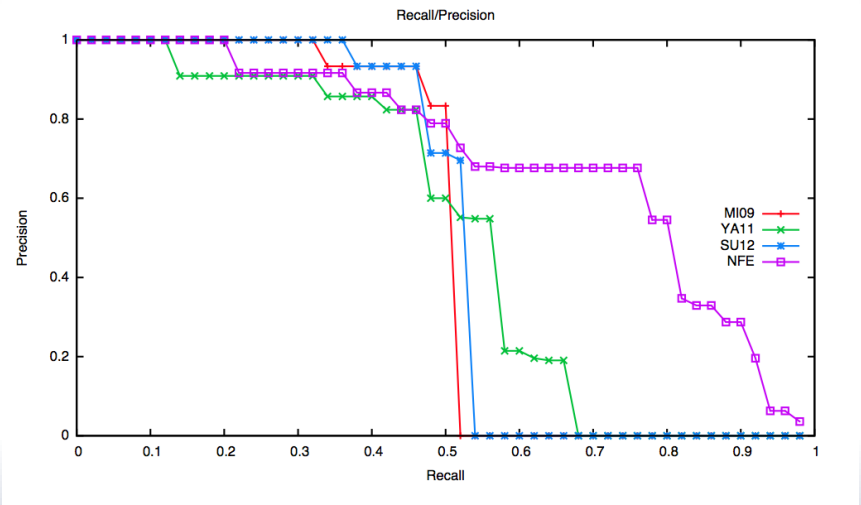
Prediction of Freebase relations

- Precision and recall for the relation **works_written(X,Y)**.
- The drop for MI09 and SU12 is due to the unretrieved facts with patterns not seen with this relation in the training set.
- YA11 overcomes this using clustered features.



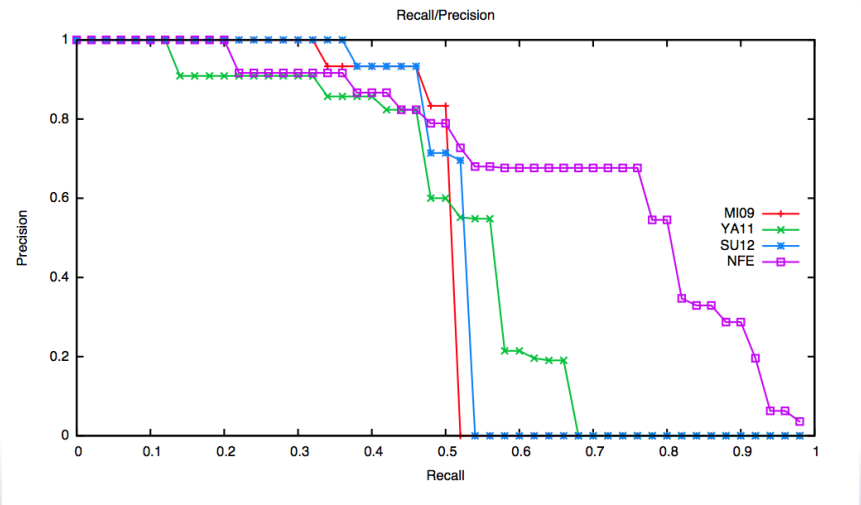
Prediction of Freebase relations

- Precision and recall for the relation **works_written(X,Y)**.
- The drop for MI09 and SU12 is due to the unretrieved facts with patterns not seen with this relation in the training set.
- YA11 overcomes this using clustered features.



Prediction of Freebase relations

- Precision and recall for the relation **works_written(X,Y)**.
- The drop for MI09 and SU12 is due to the unretrieved facts with patterns not seen with this relation in the training set.
- YA11 overcomes this using clustered features.



Prediction of surface patterns

- These surface pattern relations were not captured in the Freebase snapshot used.
- The model can capture asymmetry: it can predict the relation X-scientist-at-Y as well as the more general X-employed-by-Y (cf. Riedel et al., 2013).

Relation	#	N	F	NF	NFE
visit	80	0.19	**0.68**	0.49	0.42
attend	69	**0.23**	0.10	0.07	0.10
base	61	0.46	**0.87**	0.81	0.68
head	38	0.47	0.67	**0.70**	0.68
scientist	36	0.25	**0.84**	0.79	0.73
support	18	0.16	0.29	0.32	**0.38**
adviser	11	0.19	0.15	0.19	**0.28**
criticize	9	0.09	0.60	**0.67**	0.64
praise	4	0.01	0.03	0.05	**0.10**
vote	3	0.18	0.18	**0.34**	0.34
MAP		0.22	0.44	0.44	0.43
Weighted MAP		0.28	0.56	0.50	0.46

Prediction of surface patterns

- These surface pattern relations were not captured in the Freebase snapshot used.
- The model can capture **asymmetry**: it can predict the relation **X-scientist-at-Y** as well as the more general **X-employed-by-Y** (cf. Riedel et al., 2013).

Relation	#	N	F	NF	NFE
visit	80	0.19	**0.68**	0.49	0.42
attend	69	**0.23**	0.10	0.07	0.10
base	61	0.46	**0.87**	0.81	0.68
head	38	0.47	0.67	**0.70**	0.68
scientist	36	0.25	**0.84**	0.79	0.73
support	18	0.16	0.29	0.32	**0.38**
adviser	11	0.19	0.15	0.19	**0.28**
criticize	9	0.09	0.60	**0.67**	0.64
praise	4	0.01	0.03	0.05	**0.10**
vote	3	0.18	0.18	**0.34**	0.34
MAP		0.22	0.44	0.44	0.43
Weighted MAP		0.28	0.56	0.50	0.46

Brief overview

The method in a nutshell:

- Input data: a structured knowledge base (Freebase in this case) and annotated text data (the NYT corpus).
- Both sources must be aligned - this can be done by a simple string matching heuristic.
- The lexicalized dependency paths in the aligned text data must be extracted, since they will end up as elements of the set of observed facts, $\mathcal{O}$.
- The matrix Y is populated with the information from the set $\mathcal{O}$.
- The BPR objective function is constructed with the information from the Y matrix.
- Maximise the objective function using, for example, stochastic gradient descent.
- Use the parameter estimates to obtain the desired (posterior) probabilites for the unobserved relations.

Brief overview

The method in a nutshell:

- Input data: a structured knowledge base (Freebase in this case) and annotated text data (the NYT corpus).
- Both sources must be aligned - this can be done by a simple string matching heuristic.
- The lexicalized dependency paths in the aligned text data must be extracted, since they will end up as elements of the set of observed facts, $\mathcal{O}$.
- The matrix Y is populated with the information from the set $\mathcal{O}$.
- The BPR objective function is constructed with the information from the Y matrix.
- Maximise the objective function using, for example, stochastic gradient descent.
- Use the parameter estimates to obtain the desired (posterior) probabilites for the unobserved relations.

Brief overview

The method in a nutshell:

- Input data: a structured knowledge base (Freebase in this case) and annotated text data (the NYT corpus).
- Both sources must be aligned - this can be done by a simple string matching heuristic.
- The lexicalized dependency paths in the aligned text data must be extracted, since they will end up as elements of the set of observed facts, $\mathcal{O}$.
- The matrix Y is populated with the information from the set $\mathcal{O}$.
- The BPR objective function is constructed with the information from the Y matrix.
- Maximise the objective function using, for example, stochastic gradient descent.
- Use the parameter estimates to obtain the desired (posterior) probabilites for the unobserved relations.

Brief overview

The method in a nutshell:

- Input data: a structured knowledge base (Freebase in this case) and annotated text data (the NYT corpus).
- Both sources must be aligned - this can be done by a simple string matching heuristic.
- The lexicalized dependency paths in the aligned text data must be extracted, since they will end up as elements of the set of observed facts, $\mathcal{O}$.
- The matrix Y is populated with the information from the set $\mathcal{O}$.
- The BPR objective function is constructed with the information from the Y matrix.
- Maximise the objective function using, for example, stochastic gradient descent.
- Use the parameter estimates to obtain the desired (posterior) probabilites for the unobserved relations.

Brief overview

The method in a nutshell:

- Input data: a structured knowledge base (Freebase in this case) and annotated text data (the NYT corpus).
- Both sources must be aligned - this can be done by a simple string matching heuristic.
- The lexicalized dependency paths in the aligned text data must be extracted, since they will end up as elements of the set of observed facts, $\mathcal{O}$.
- The matrix Y is populated with the information from the set $\mathcal{O}$.
- The BPR objective function is constructed with the information from the Y matrix.
- Maximise the objective function using, for example, stochastic gradient descent.
- Use the parameter estimates to obtain the desired (posterior) probabilites for the unobserved relations.

Brief overview

The method in a nutshell:

- Input data: a structured knowledge base (Freebase in this case) and annotated text data (the NYT corpus).
- Both sources must be aligned - this can be done by a simple string matching heuristic.
- The lexicalized dependency paths in the aligned text data must be extracted, since they will end up as elements of the set of observed facts, $\mathcal{O}$.
- The matrix Y is populated with the information from the set $\mathcal{O}$.
- The BPR objective function is constructed with the information from the Y matrix.
- Maximise the objective function using, for example, stochastic gradient descent.
- Use the parameter estimates to obtain the desired (posterior) probabilites for the unobserved relations.

Brief overview

The method in a nutshell:

- Input data: a structured knowledge base (Freebase in this case) and annotated text data (the NYT corpus).
- Both sources must be aligned - this can be done by a simple string matching heuristic.
- The lexicalized dependency paths in the aligned text data must be extracted, since they will end up as elements of the set of observed facts, $\mathcal{O}$.
- The matrix Y is populated with the information from the set $\mathcal{O}$.
- The BPR objective function is constructed with the information from the Y matrix.
- Maximise the objective function using, for example, stochastic gradient descent.
- Use the parameter estimates to obtain the desired (posterior) probabilites for the unobserved relations.

Conclusions

- The general idea is good.
- The results presented in the paper seem to support that notion.
- Not a parsimonious method at all.
- Unclear what might happen if arbitrary text sources (instead of or in addition to NYT corpus) are used.
- How stable are the parameter estimates? No cross validation was reported.

Conclusions

- The general idea is good.
- The results presented in the paper seem to support that notion.
- Not a parsimonious method at all.
- Unclear what might happen if arbitrary text sources (instead of or in addition to NYT corpus) are used.
- How stable are the parameter estimates? No cross validation was reported.

Conclusions

- The general idea is good.
- The results presented in the paper seem to support that notion.
- Not a parsimonious method at all.
- Unclear what might happen if arbitrary text sources (instead of or in addition to NYT corpus) are used.
- How stable are the parameter estimates? No cross validation was reported.

Conclusions

- The general idea is good.
- The results presented in the paper seem to support that notion.
- Not a parsimonious method at all.
- Unclear what might happen if arbitrary text sources (instead of or in addition to NYT corpus) are used.
- How stable are the parameter estimates? No cross validation was reported.

Conclusions

- The general idea is good.
- The results presented in the paper seem to support that notion.
- Not a parsimonious method at all.
- Unclear what might happen if arbitrary text sources (instead of or in addition to NYT corpus) are used.
- How stable are the parameter estimates? No cross validation was reported.

References

Sebastian Riedel, Limin Yao, Andrew McCallum, Benjamin M. Marlin (2013)

Relation Extraction with Matrix Factorization and Universal Schemas

Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13).

Limin Yao, Sebastian Riedel, Andrew McCallum (2012)

Probabilistic Databases of Universal Schema

Proceedings of the AKBC-WEKEX Workshop at NAACL 2012

Sebastian Riedel, Limin Yao, Andrew McCallum (2010)

Modelling relations and their mentions without labeled text

Proceedings of the European Conference on Machine Learning and Knowledge Discovery on Databases (ECML PKDD '10)

Michael Collins, Sanjoy Dasgupta, Robert E. Schapire (2001)

A generalisation of principal component analysis to the exponential family.

Proceedings of NIPS

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, Lars Schmidt-Thieme (2009)

BPR: Bayesian Personalized Ranking from implicit feedback.

Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)

The End