



Statistical Relational Learning

Volker Tresp

Siemens Corporate Technology
Ludwig Maximilian University of Munich

Statistical Relational Learning: Generalization of Multi-variate Learning

- I. Bayesian Networks for Relational Learning
- II. Markov Networks for Relational Learning
- III. Statistical Mixture Models for Relational Learning
- IV. Latent Factor Models for Relational Learning
- V. Machine Learning with Knowledge Graphs



I. Bayesian Networks for Relational Learning

Volker Tresp

Siemens Corporate Technology
Ludwig Maximilian University of Munich



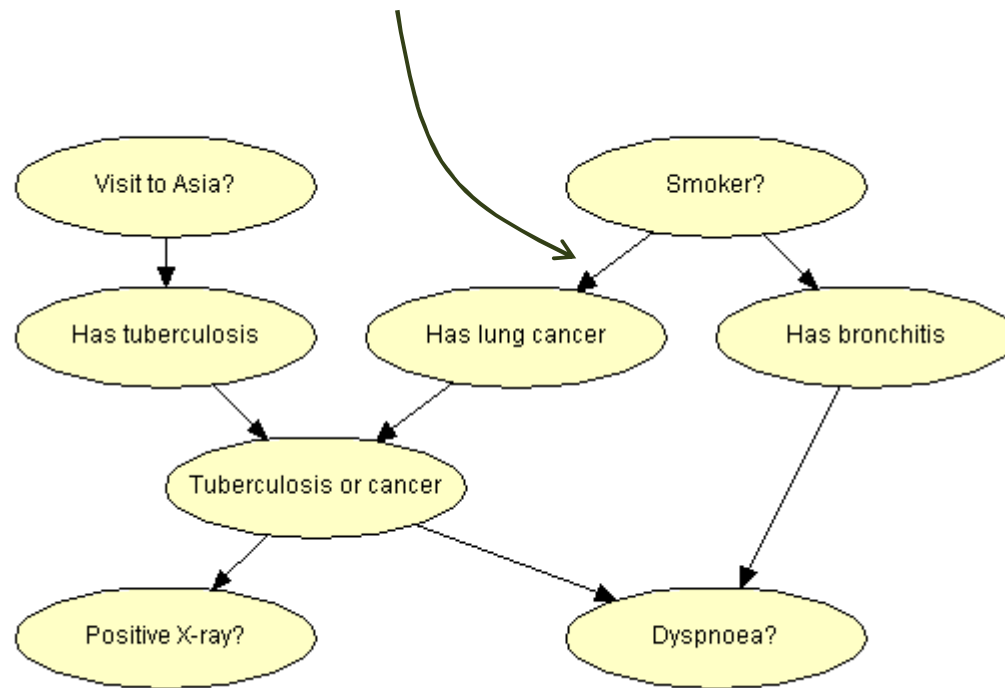
I. Bayesian Networks for Relational Learning

- a: Introduction

Classical Bayes Net

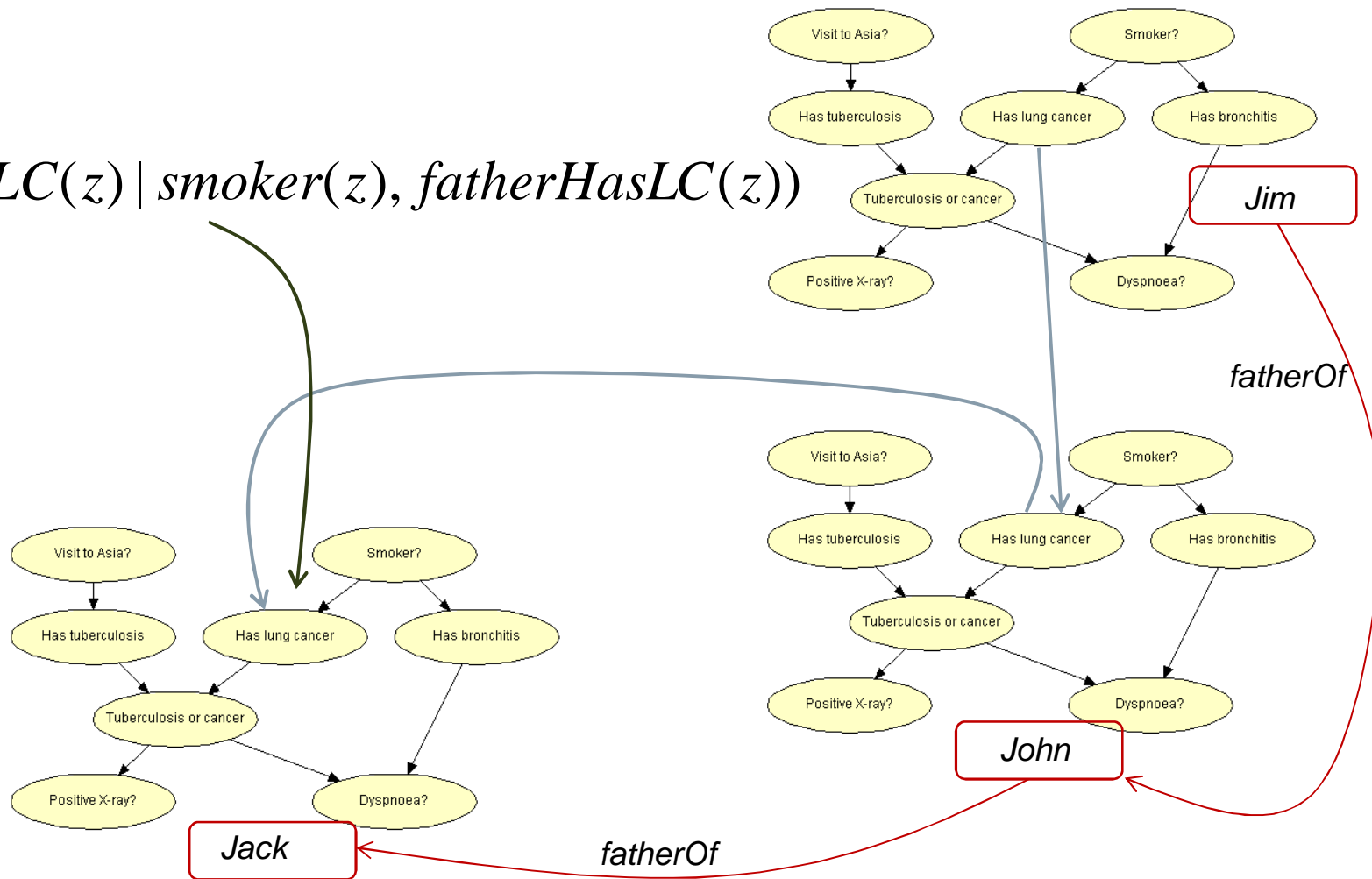
- Chest Clinic

$$P(\text{hasLC}(z) \mid \text{smoker}(z))$$



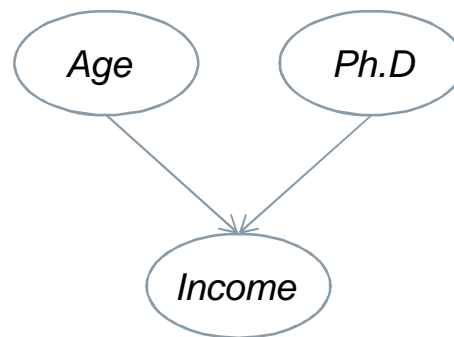
Including Father and Grandfather

$$P(\text{hasLC}(z) \mid \text{smoker}(z), \text{fatherHasLC}(z))$$



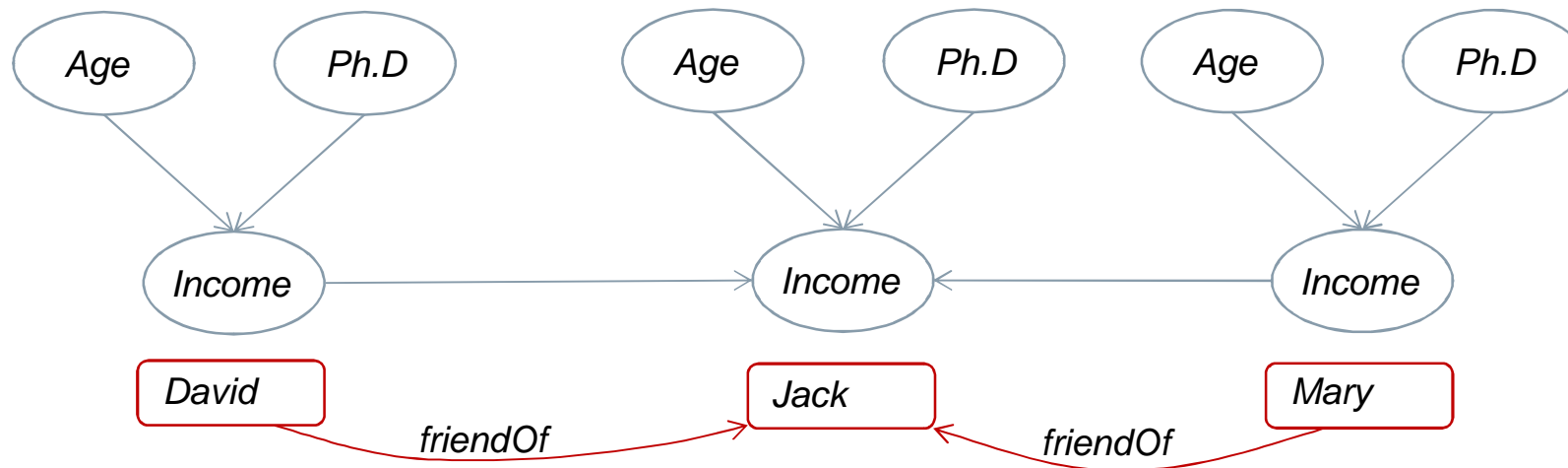
Income

$$P(\text{hasHighIncome}(z) \mid \text{middleAge}(z), \text{hasPhD}(z))$$



Income (cont'd)

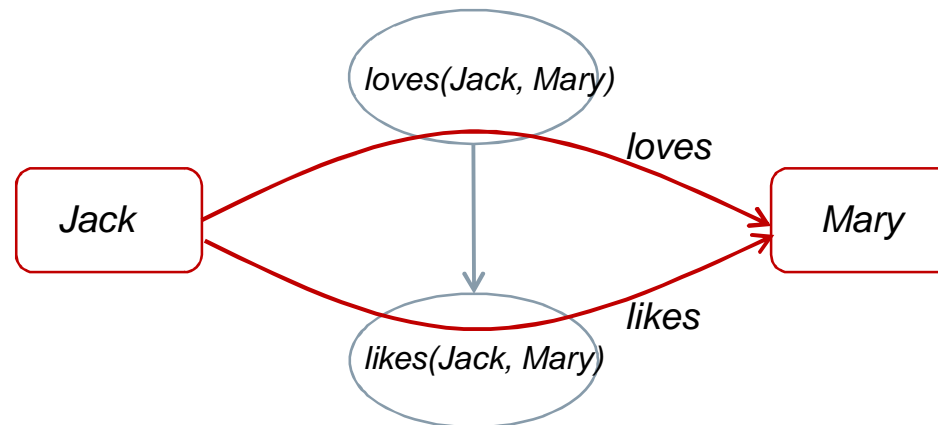
$P(\text{hasHighIncome}(z) \mid$
 $\text{middleAge}(z), \text{hasPhD}(z), \text{highIncomeFriends}(z))$



Relationships

- We can also have probabilistic dependencies involving relationships

$$P(\text{likes}(z, y) \mid \text{loves}(z, y))$$



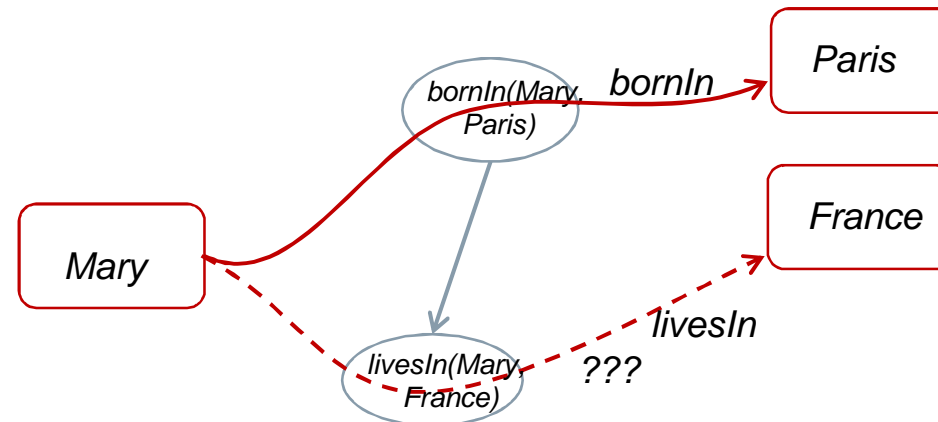
- Note that there is the dependency graph (grey) and the triple graph (ontology, knowledge graph) (red) (dual!)
 - Not to be confused!

Relationships (cont'd)

- “Born in Paris” can predict “Lives in France”

$$P(\text{livesIn}(z, \text{France}) \mid \text{bornIn}(z, \text{Paris}))$$

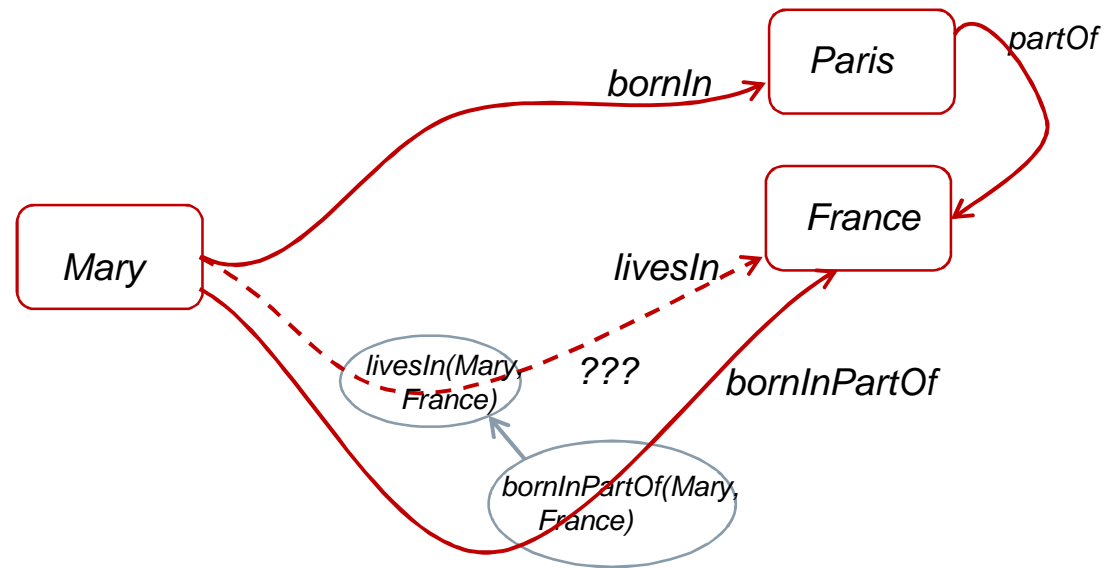
- But do we need to learn this for all cities and all countries?



Relationships (cont'd)

$$P(\text{livesIn}(z, y) \mid \text{bornInPartOf}(z, y))$$

$$\text{bornInPartOf}(z, y) := \exists t. \text{bornIn}(z, t) \wedge \text{partOf}(t, y)$$



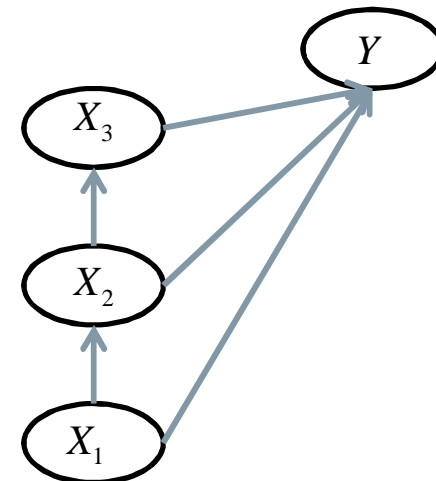
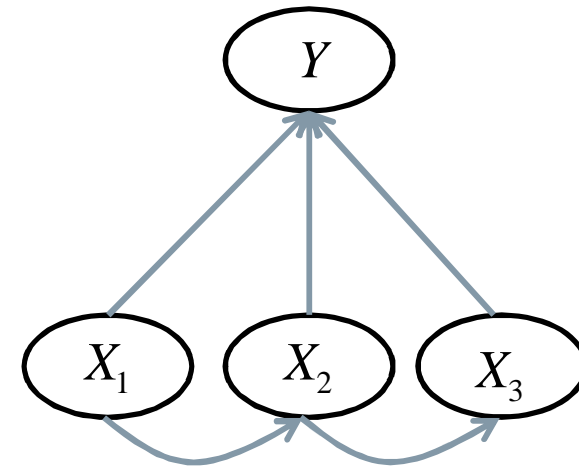


I: Bayesian Networks for Relational Learning

- b: Review of Bayes Nets

Why Bayes Nets

- There are cases where supervised learning is not applicable: when there is not one target variable of interest but many, or when in each data point different variables might be available or missing
- Typical example: medical domain with many kinds of diseases, symptoms, and context information: for a given patient little is known and one is interested in the prediction of many possible diseases and procedures



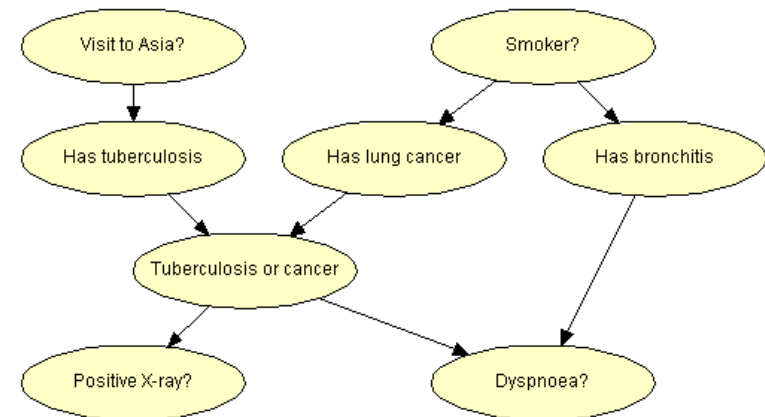
Definition of a Bayes Net

- The random variables in a domain are displayed as nodes (vertices)
- Directed links (arcs, edges) represent direct (causal) dependencies between parent node and child node
- Quantification of the dependency:
 - For nodes without parents one specifies a priori probabilities

$$P(A = i) \quad \forall i$$

- For nodes with parents, one specifies conditional probabilities. E.g., for two parents

$$P(A = i | B = j, C = k) \quad \forall i, j, k$$

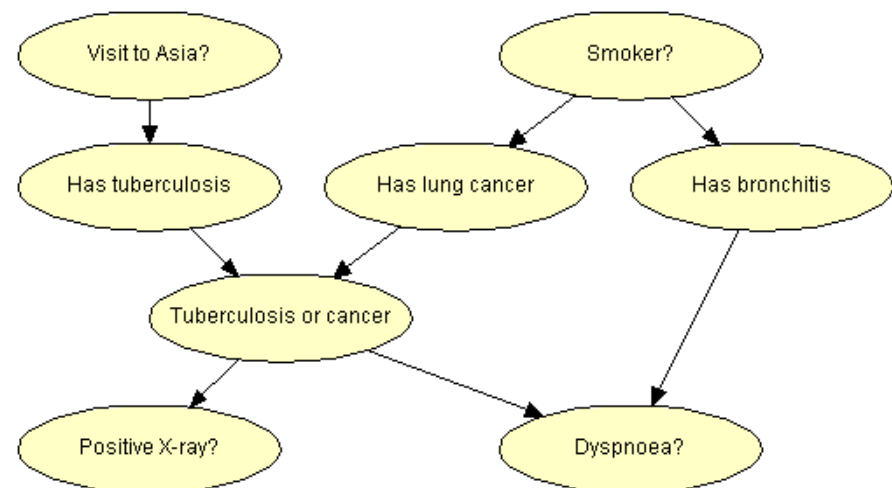


Joint Probability Distribution

- A Bayes net specifies a probability distribution in the form

$$P(X_1, \dots, X_M) = \prod_{i=1}^M P(X_i | \text{par}(X_i))$$

where $\text{par}(X_i)$ is the set of parent nodes. This set can be zero



Mathematical Foundation for Bayes Nets

- Let's start with the factorization of a probability distribution

$$P(X_1, \dots, X_M) = \prod_{i=1}^M P(X_i | X_1, \dots, X_{i-1})$$

- This decomposition can be done with an arbitrary ordering of variables; each variable is conditioned on all predecessor variables
- The dependencies can be simplified if a variable does not depend on all of its predecessors

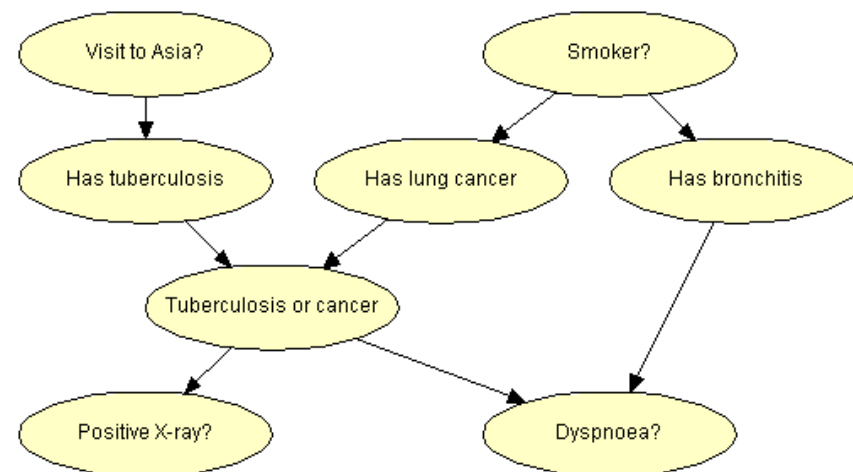
$$P(X_i | X_1, \dots, X_{i-1}) = P(X_i | \text{par}(X_i))$$

with

$$\text{par}(X_i) \subseteq X_1, \dots, X_{i-1}$$

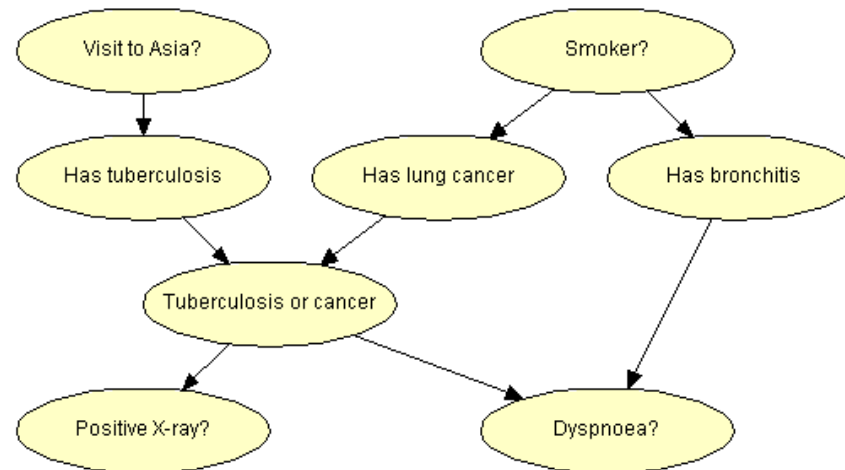
Causal Probabilistic Networks

- When the ordering of the variables corresponds to a causal ordering, we obtain a causal probabilistic network
- A decomposition obeying the causal ordering typically yields a representation with the smallest number of parent variables, i.e., the smallest number of links
- For causal probabilistic networks, the assumption is that the un-modeled factors should only significantly influence individual nodes (and thus appear as noise), but NOT pairs or larger sets of variables (which would induce dependencies)!



Design of a Bayes Net

- The expert needs to be clear about the important variables in the domain
- The expert must indicate direct causal dependencies by specifying the directed links in the net
- The expert needs to quantify the causal dependencies: define the conditional probability tables



Inference

- The most important operation is inference: given that the state a set of random variables is known, what is the probability distribution of one or several of the remaining variables
- Let \mathcal{X} be the set of random variables. Let $\mathcal{X}^m \subseteq \mathcal{X}$ be the set of known (measured) variables and let $X^q \in \mathcal{X} \setminus \mathcal{X}^m$ be the variable of interest and let $\mathcal{X}^r = \mathcal{X} \setminus (\mathcal{X}^m \cup X^q)$ be the set of remaining variables

Inference: Marginalization and Conditioning

Inference consists of the following steps:

- We calculate the probability distribution of the known variables and the query variable via marginalization

$$P(X^q, \mathcal{X}^m) = \sum_{\mathcal{X}^r} P(X_1, \dots, X_M)$$

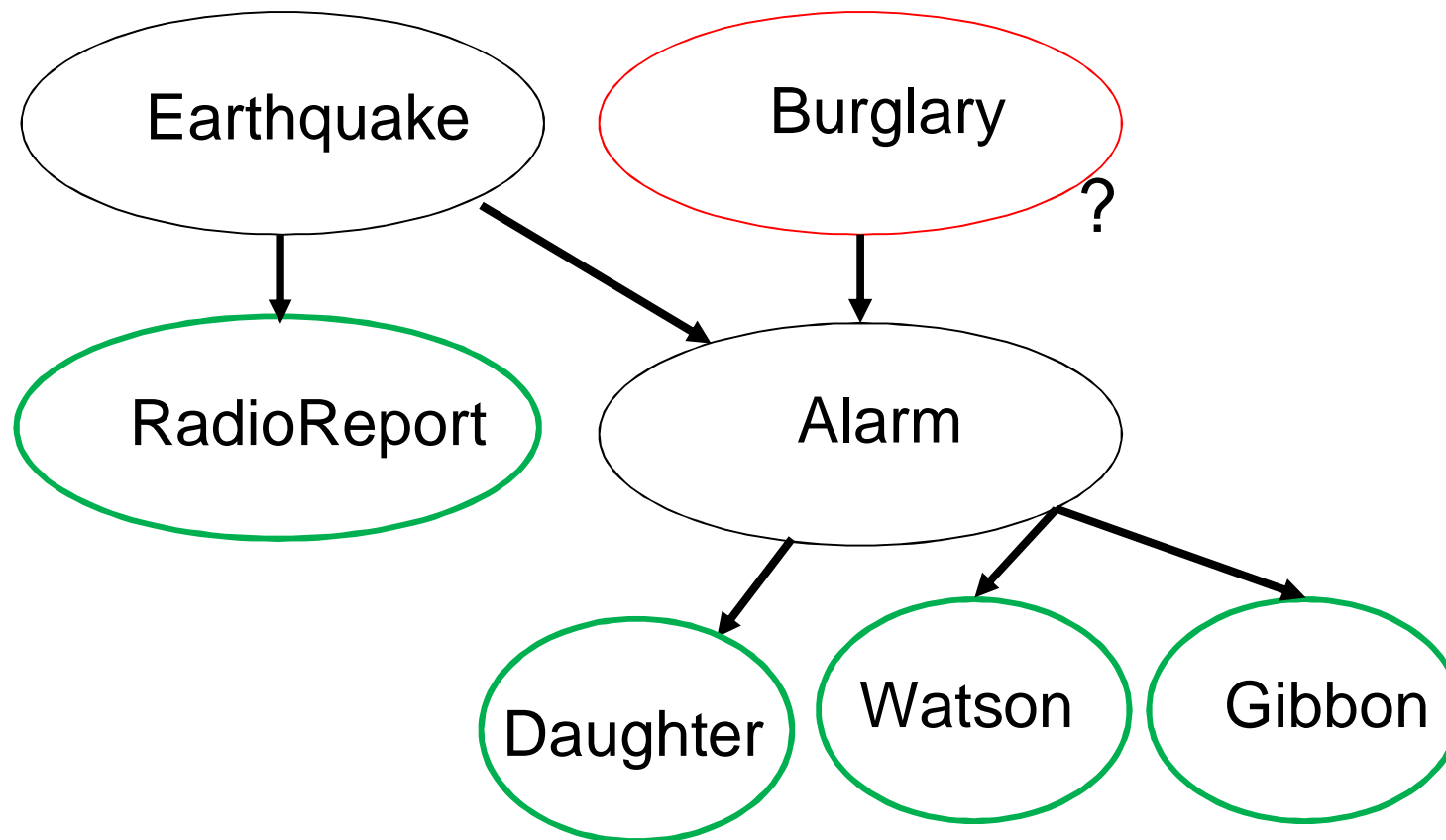
- The normalization is calculated as

$$P(\mathcal{X}^m) = \sum_{\mathcal{X}^q} P(X^q, \mathcal{X}^m)$$

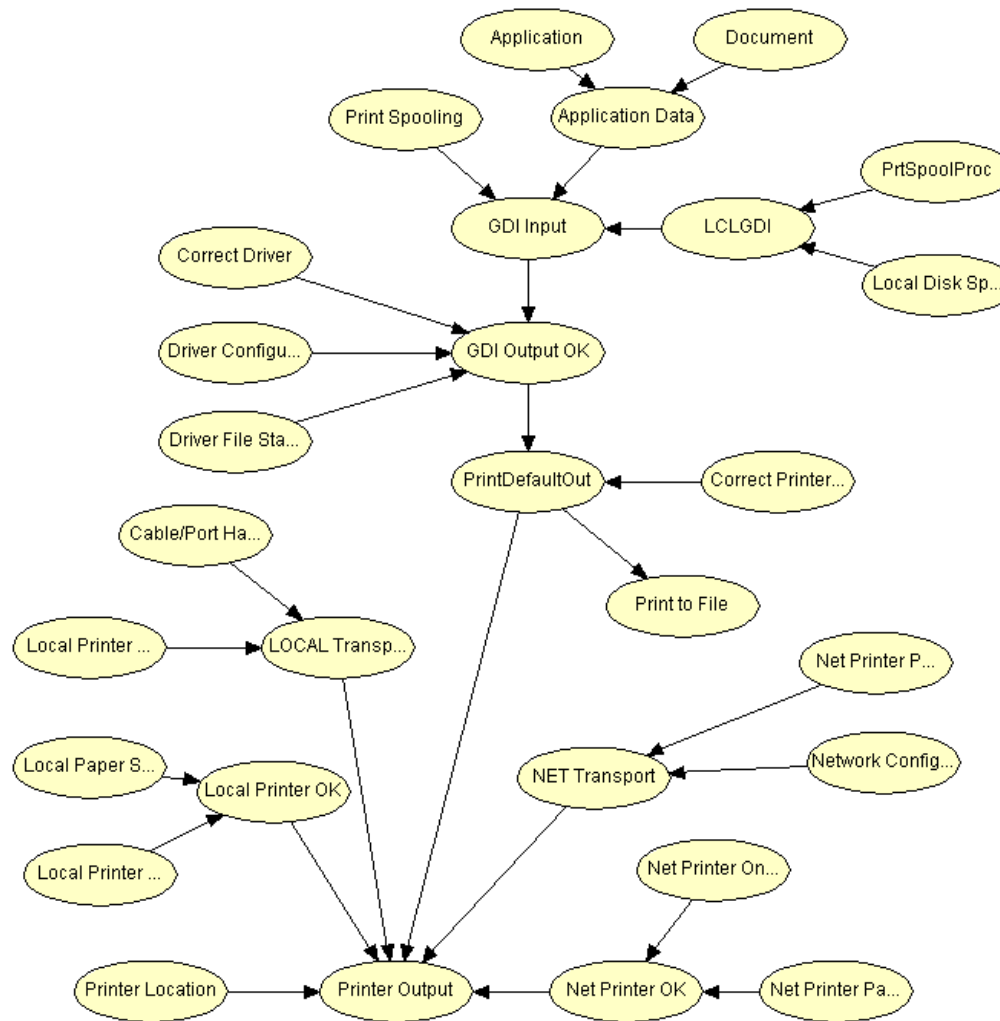
- Calculation of the conditional probability distributions

$$P(X^q | \mathcal{X}^m) = \frac{P(X^q, \mathcal{X}^m)}{P(\mathcal{X}^m)}$$

Holmes Net



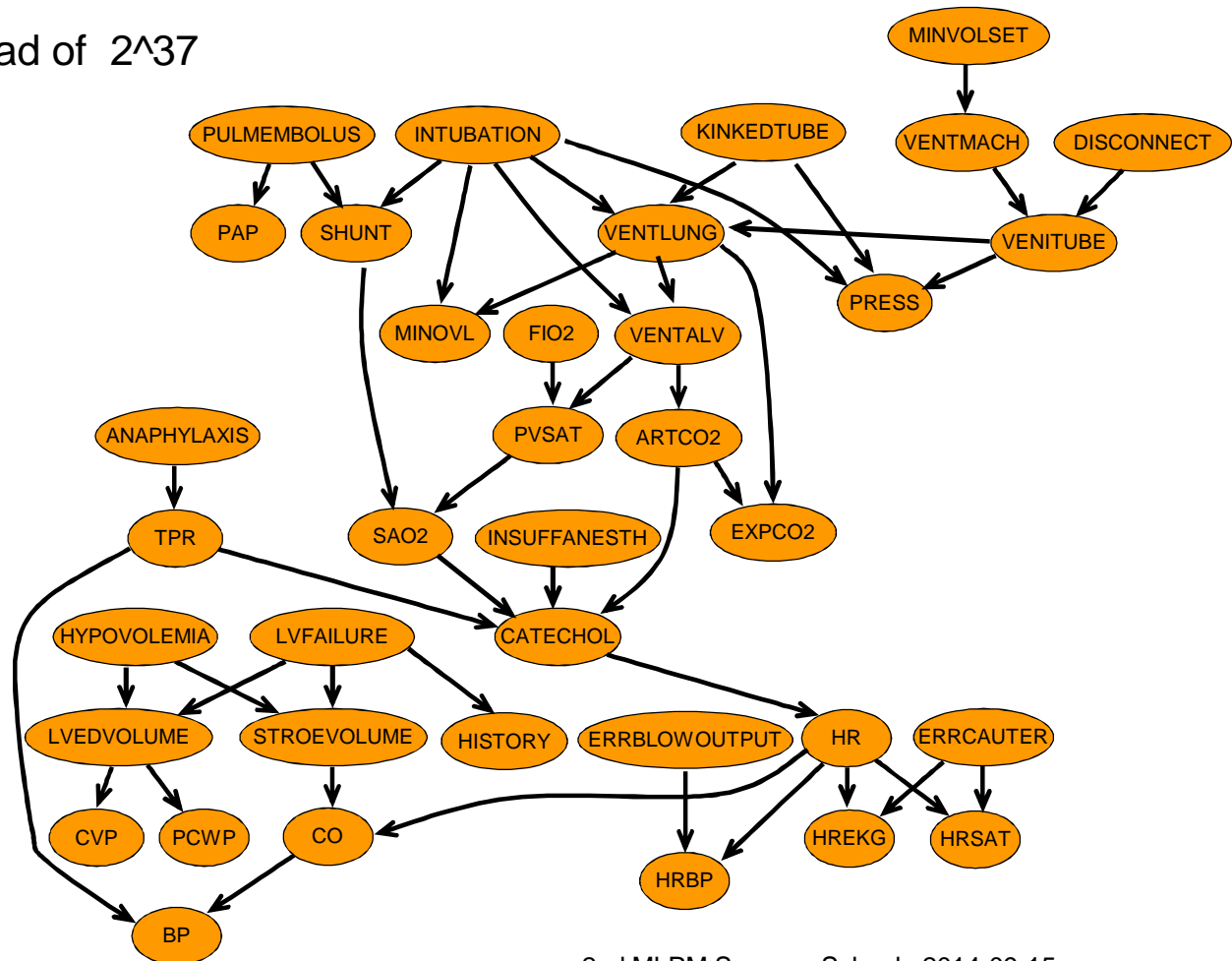
Microsoft's Printer Trouble Shooter



Alarm Net

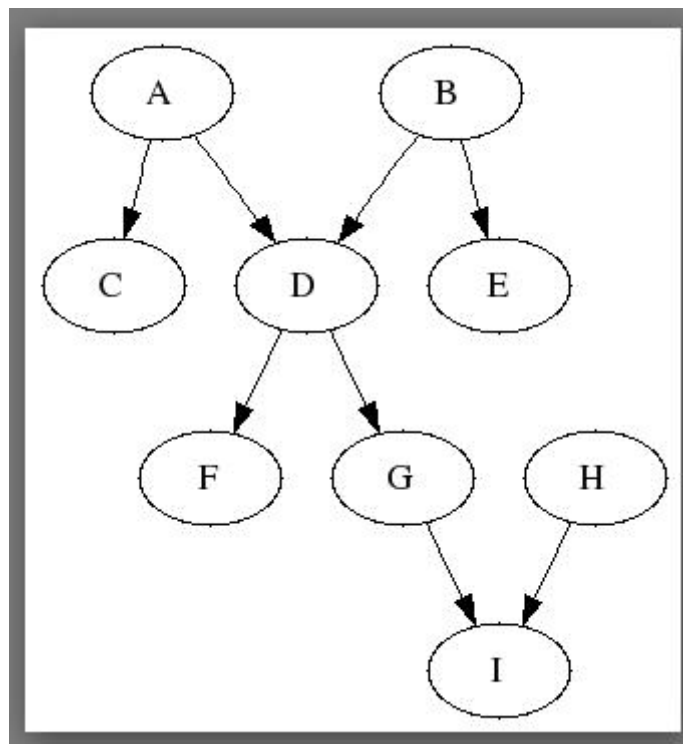
Application: monitoring intensive-care patients

- 37 variables
- 509 parameters ...instead of 2^{37}

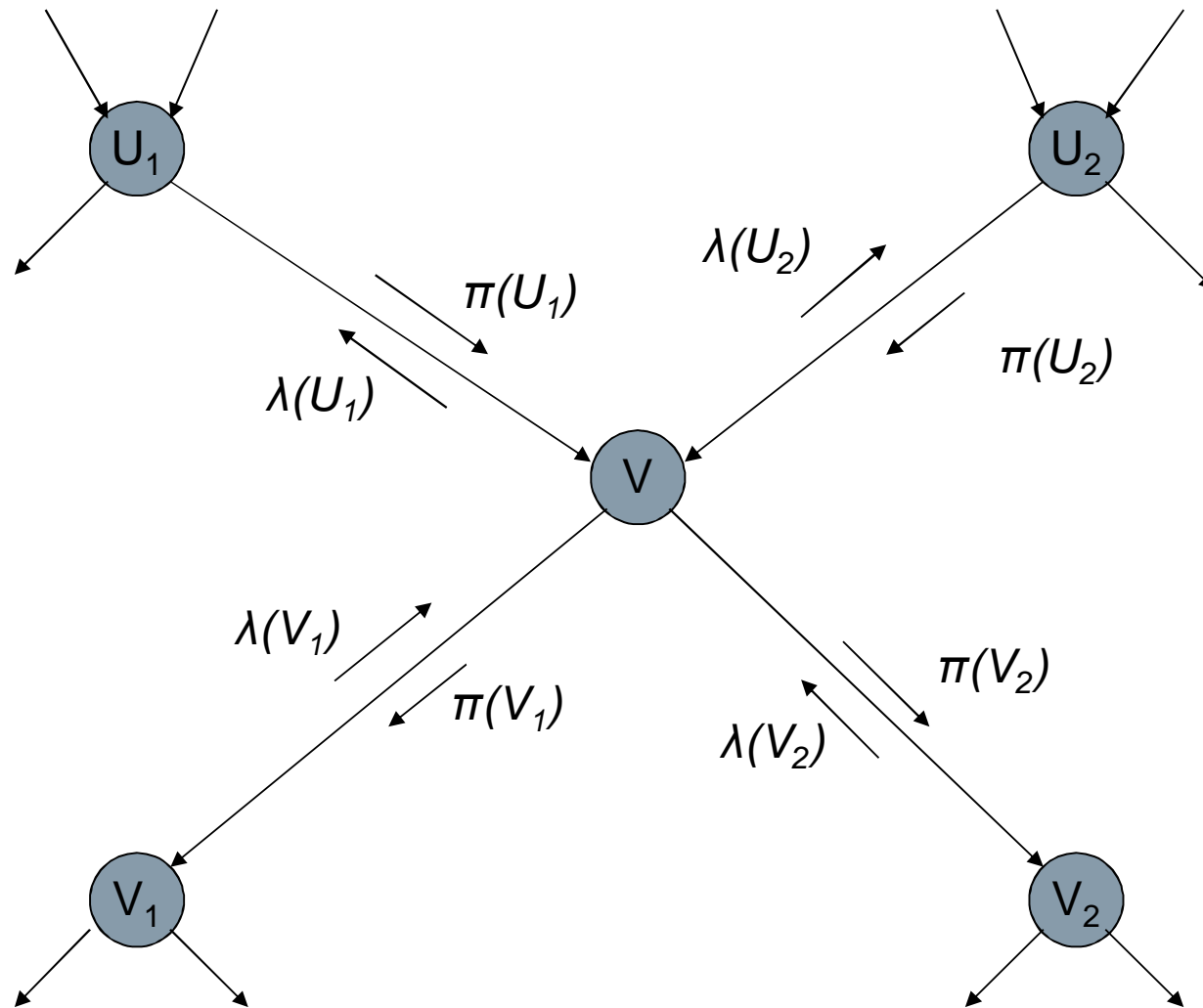


Inference in Bayes Nets without Cycles when the Link Directions are Removed

- By construction there are no cycles in the directed net; the structure of a Bayesian net is a directed acyclic graph (DAG)
- But there might be cycles when one ignores the directions
- Let's first consider the simpler case without cycles in the undirected graph; the structure of the Bayes net is a poly-tree: there is at most one directed path between two nodes



Pearl's Belief Propagation



Approximate Inference

- The **junction tree algorithm** performs correct inference also in Bayes nets with cycles; but with large loops the algorithm can be inefficient
- Approximate methods
 - Sampling-based methods
 - Markov Chain Monte Carlo (**MCMC**)
 - **Gibbs sampling**
 - **Mean-field inference**
 - **Loopy belief propagation**
 - Loopy belief propagation: the application of belief propagation to Bayes nets with cycles (although strictly not correct)
 - The local update rules are applied until convergence is achieved (convergence is not guaranteed)

Design of a Bayes Net (cont'd)

- The expert needs to be clear about the important variables in the domain
- The expert must indicate direct causal dependencies by specifying the directed links in the net
- The expert needs to quantify the causal dependencies: define the conditional probability tables
- This can be challenging if a node has many parents: if a binary node has n binary parents, then the expert needs to specify 2^{n-1} numbers!
- One often makes simplifying assumptions; the best-known one is the *noisy-or assumption* and the expert only needs to specify n parameters

Maximum Likelihood Learning

- We assume that all nodes in the Bayesian net have been observed for N instances (e.g., N patients)
- Let $\theta_{i,j,k}$ be defines as

$$\theta_{i,j,k} = P(X_i = j | \text{par}(X_i) = k)$$

- This means that $\theta_{i,j,k}$ is the probability that X_i is in state j , when its parents are in the state k (we assume that the states of the parents can be enumerated in a systematic way)
- Let $N_{i,j,k}$ be the number of samples in which $X_i = j$ and $\text{par}(X_i) = k$
- The maximum likelihood (ML) estimate is simply

$$\hat{\theta}_{i,j,k} = \frac{N_{i,j,k}}{\sum_j N_{i,j,k}}$$

MAP-estimate for Integrating Prior Knowledge

- Often counts are very small and a ML-estimate has high variance
- One simply specifies efficient counts (counts from virtual data) which then can be treated as real counts Let $\alpha_{i,j,k} \geq 0$ be virtual counts for $N_{i,j,k}$
- One obtains the *maximum a posteriori* (MAP) estimate as

$$\hat{\theta}_{i,j,k} = \frac{\alpha_{i,j,k} + N_{i,j,k}}{\sum_j (\alpha_{i,j,k} + N_{i,j,k})}$$

Missing Data: EM

- The problem of missing data is an important issue in statistical modeling
- In the simplest case, one can assume that data is missing at random
- Data is not missing at random, if, for example, one analyses the wealth distribution in a city and rich people tend to refuse to report their income
- For some models the EM (Expectation Maximization)-algorithm can be applied to calculate ML or MAP estimates
- Consider a particular data point l . In the E-step we calculate the probability for marginal probabilities of interest given the known information \mathcal{X}_l^m in that data point and given the current estimates of the parameters $\hat{\theta}$, using e.g. belief propagation. Then we get for expected counts

$$E(N_{i,j,k}) = \sum_{l=1}^N P(X_i = j, \text{par}(X_i) = k | \mathcal{X}_l^m, \hat{\theta})$$

Missing Data: M-Step

- Based on the E-step, we get in the M-step

$$\hat{\theta}_{i,j,k} = \frac{E(N_{i,j,k})}{\sum_k E(N_{i,j,k})}$$

- E-Step and M-Step are iterated until convergence. One can show that EM does not decrease the likelihood in each step; EM might converge to local optima
- The E-step is really an inference step
- Here, also approximate inference can be used (loopy-belief propagation, MCMC, Gibbs, mean-field)

Structural Learning in Bayes Nets

- One can also consider learning the structure of a Bayes net and maybe even discover causality
- In structural learning, several points need to be considered
- There are models that are structural equivalent. For example in a net with only two variables A and B one might show that there is statistical correlation between the two variables, but it is impossible to decide if $A \rightarrow B$ or $A \leftarrow B$. Colliders (nodes where arrow-head meet) can make directions identifiable
- If C is highly correlated with A and B and A and B are also highly correlated, it might be clear from the data that C depends on both A and B but difficult to decide if it only depends on A or only depends on B

Greedy Search

- In the most common approaches one defines a cost function and looks for the structure that is optimal under the cost function. One has to deal with many local optima
- Greedy Search: One starts with an initial network (fully connected, empty) and makes local changes (removal of directed link, adding a link, reversing direction of a link, ...) and accepts the change, when the cost function improves
- Greedy search can be started from different initial conditions
- Alternatives: Simulated Annealing, Genetic Algorithms

Cost Function

- As a cost function one might use a cross-validation set
- Alternatively, BIC (Bayesian Information Criterion) is used: Maximize

$$\frac{1}{N} \log L - \frac{M}{2N} \log N$$

(M is the number of parameters; N is the number of data points)

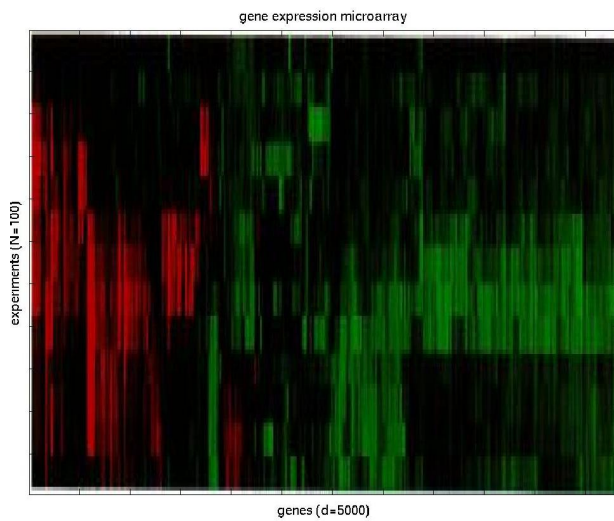
- The first term is the average log-likelihood and increases with model fit; the second term penalizes models with many parameters M and becomes less important with $N \rightarrow \infty$

Constrained-Based Methods for Structural Learning

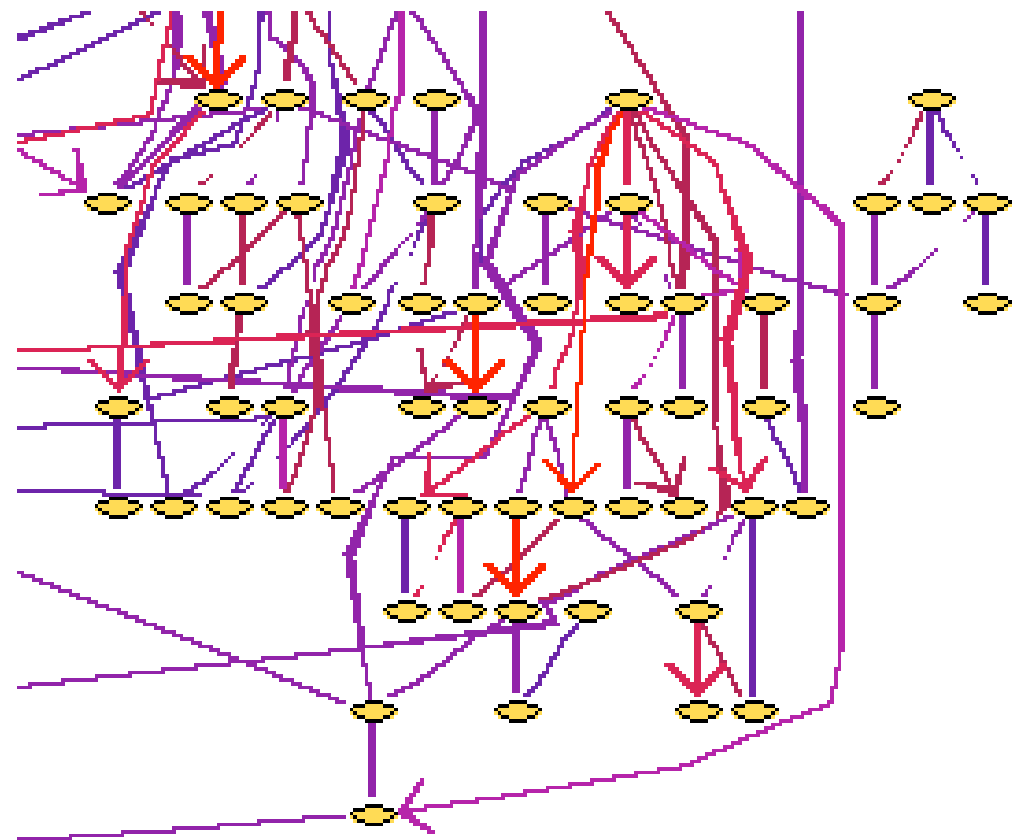
- One performs statistical independence tests and uses those to decide on network structure

Causal Structure in Gene Expression Data

Gene expression data



Genetic pathway



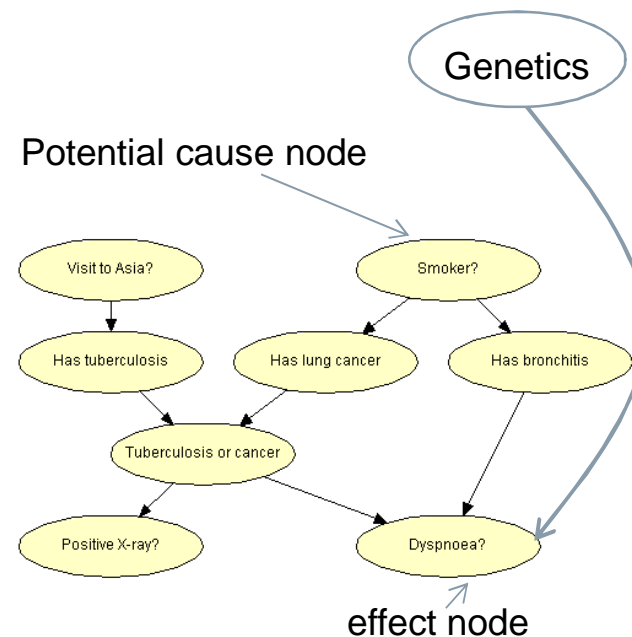
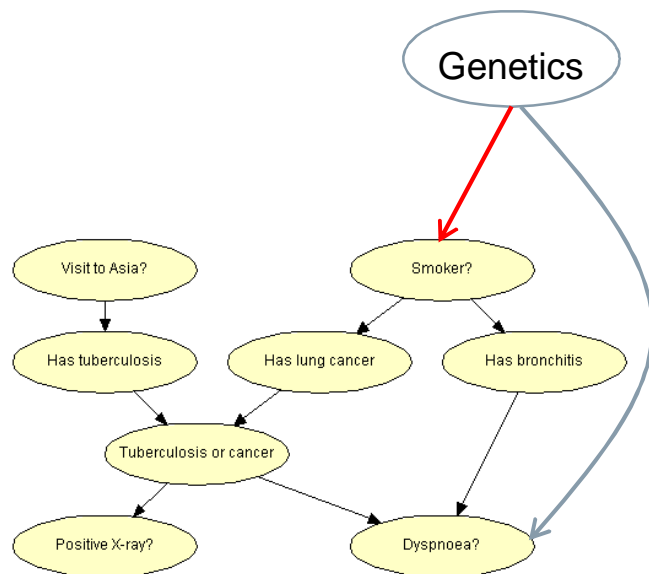
Some Comments on Causality

- In some cases the causal net has been designed by an expert
 - Causal world assumption
 - All relevant variables need to be included
 - Un-modeled variables must appear as *local* “noise” on nodes (otherwise they would induce correlations)


- Same conditions apply for the learning causal structure
- An additional aspect: structural uncertainty with finite training data

Quantifying the Causal Effect

- In a causal Bayes net the causal effect can be evaluated easily
 - Remove all links into the possible cause node
 - In case you want to evaluate the causal effect in a specific context: condition on context variables, e.g., gender, age
 - Do inference in this network and calculate the conditional probability of the possible effect node when the cause node is true and false
 - Evaluate the difference in the conditional probabilities of the effect node in both cases



- Is there a causal effect of smoking on dyspnea?
- Is there a causal effect of smoking on dyspnea given that the person has tuberculosis?



I: Bayes Nets for Relational Learning

- c: Bayes Nets for Relational Learning

Relational Complexities

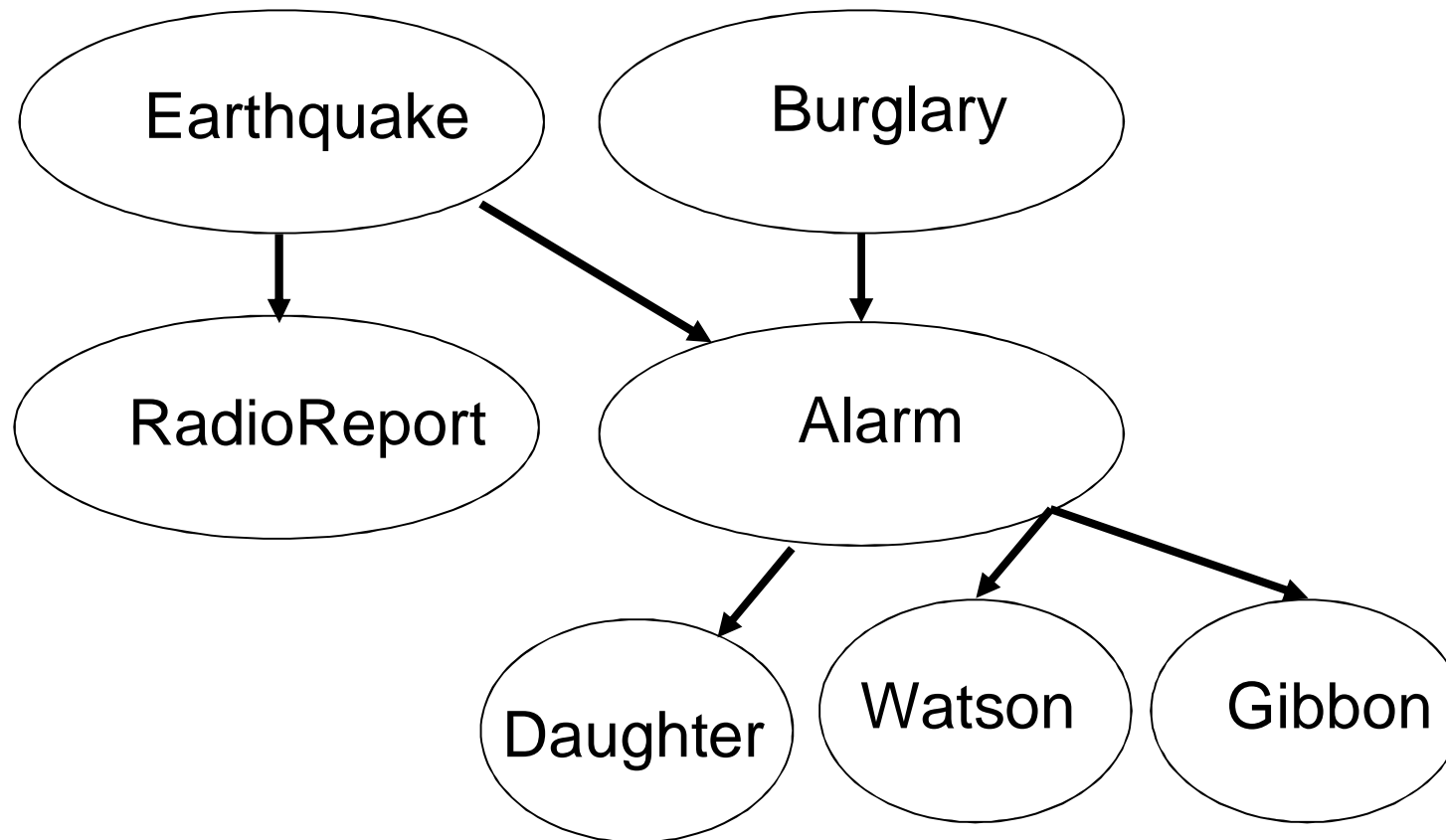
- As discussed, if we consider relationships between patients, there can be some nonlocal propagation of information (e.g., correlation between father and child concerning, smoking, bronchitis, or cancer)
- We will see that by considering binary relations such as *fatherOf* or *friendOf* we can get dependencies between nodes of potentially all patients
 - We cannot treat each patient independently of the others!
 - Technically, the whole observed world becomes one point

Relational Hierarchy

- Relationships are naturally being considered in first-order logic (FOL) and in relational databases (RDBs)
- Let's look at three relational complexities:
 - A propositional Bayes net
 - A Bayes net with unary relations/predicates (the classical learning scenario)
 - A Bayes net with also binary relations/predicates

I. Propositional Bayes Net

- Nodes in a propositional Bayesian network represent atomic propositions as *Alarm, WatsonCalls, DaughterCalls*



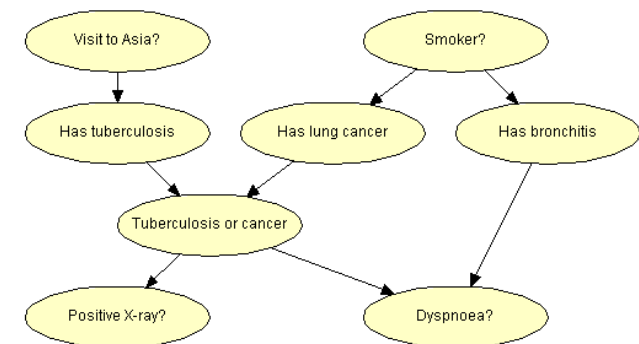
II. Template Bayes Net with Unary Predicates/Relations

- This is a Bayes typically used in machine learning
- A Bayesian network does not only make statements about a single person (*Jack*) but about a whole set of entities (all patients); a Bayesian network is a template
- Example. If we have no evidence for any descendent:

$$\forall z. P(\text{bronchitis}(z) = j | \text{smoker}(z) = k) = \theta_{\text{bron},j,k}$$

where $j \in \{0, 1\}$

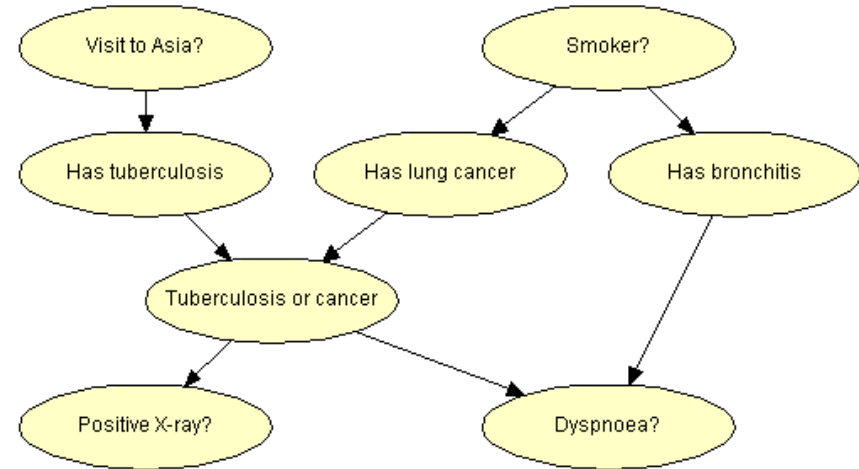
- *Jack, John, ...* are constants (e.g., objects, entities)
- z is a variable that represents constants
- \forall (for all) is a quantifier; the other one is \exists (there is)



Predicates, Atoms, Ground Atoms, Relations and Tuples

- $hasTBC()$, $hasCancer()$, $positiveXray()$ are examples of **predicates** that map the arguments to true or false
- $hasTBC(z)$, $hasCancer(z)$, $positiveXray(z)$ are **atoms** (predicates applied to arguments). They correspond to nodes (random variables) in the template Bayes net; thus, $smoker(x)$ would be the node X_{smoker} in the Bayes template net with variables \mathcal{X}
- $hasTBC(Jack)$, $hasCancer(Jack)$, $positiveXray(Jack)$ are **ground atoms** (predicates with only constants (objects) as arguments). They correspond to nodes in the ground Bayes net, which is a propositional Bayes net. Thus, $smoker(Jack)$ would be the node $X_{smoker(Jack)}$ in the ground Bayes net
- Predicate and ground atom versus relation and tuple: A **relation** is a table that contains all true ground atoms as **tuples** for a given predicate in a domain (in a (possible) world); typically predicates and relations have the same name

Database: Set of Relations (Tables)



- Here is where the Bayes net is:
 - On the schema level

$$P(\text{hasLC}(z) \mid \text{smoker}(z))$$

visitA	smoker	hasT	hasLC	hasB	posX	hasD
John	Jack	Mary	Jack	Jack	John	John
Mary	John			Mary	Mary	Jack
	Mary					

Another Representation: Matrix

- We can display the data as a matrix
- **Possible World:** truth assignments to all possible ground atoms (as defined by the relation tables or the design matrix) :

All possible ground atoms
(all possible matrix
entries)

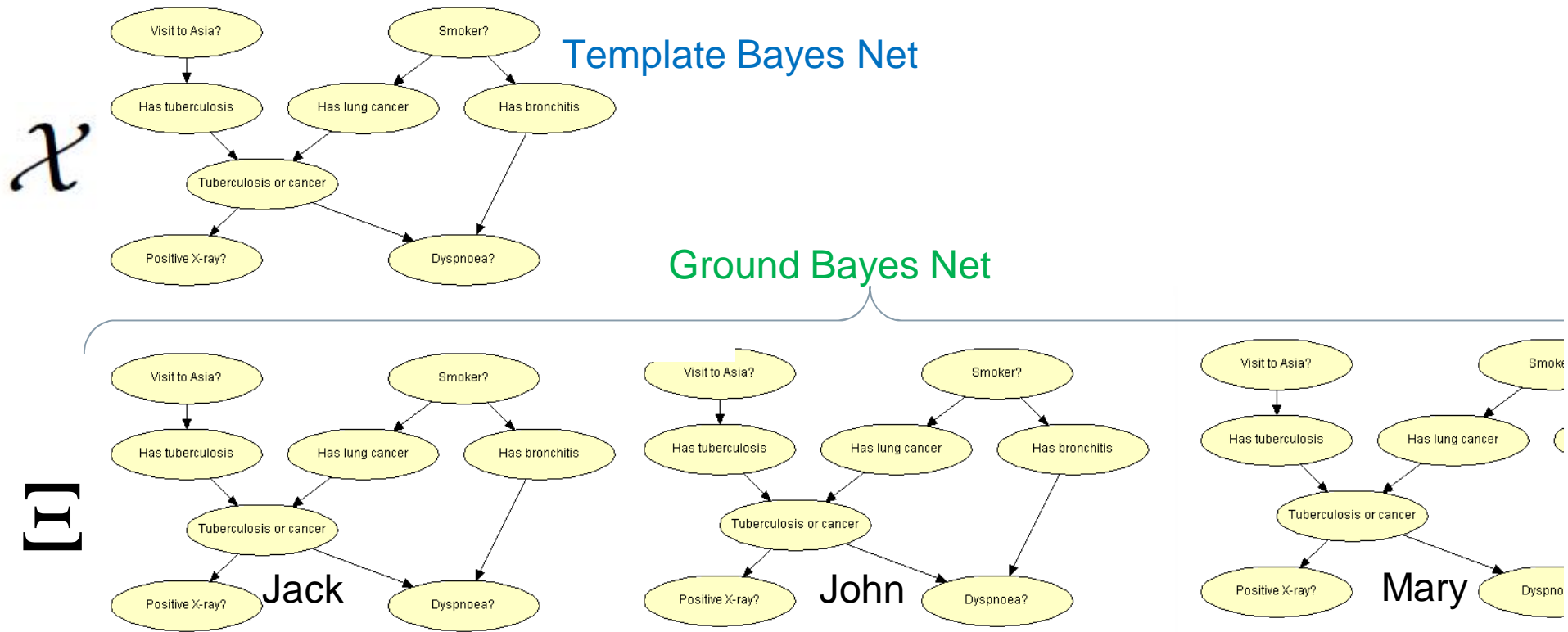
$$\mathbb{E} = \mathcal{X}$$

The truth values of those
ground atoms
(here the actual matrix
entries (0/1))

	visitA	smoker	hatT	hasLC	hasB	posX	hasD
John	1	1	0	0	0	1	1
Mary	1	1	1	0	1	1	0
Jack	0	1	0	1	1	0	1
Jim	0	0	0	0	0	0	0
Jane	0	0	0	0	0	0	0

Template and Ground Bayes Net

- The real thing is the ground Bayes net
- The template Bayes net is a template which specifies the probabilistic dependencies in the ground Bayes net:

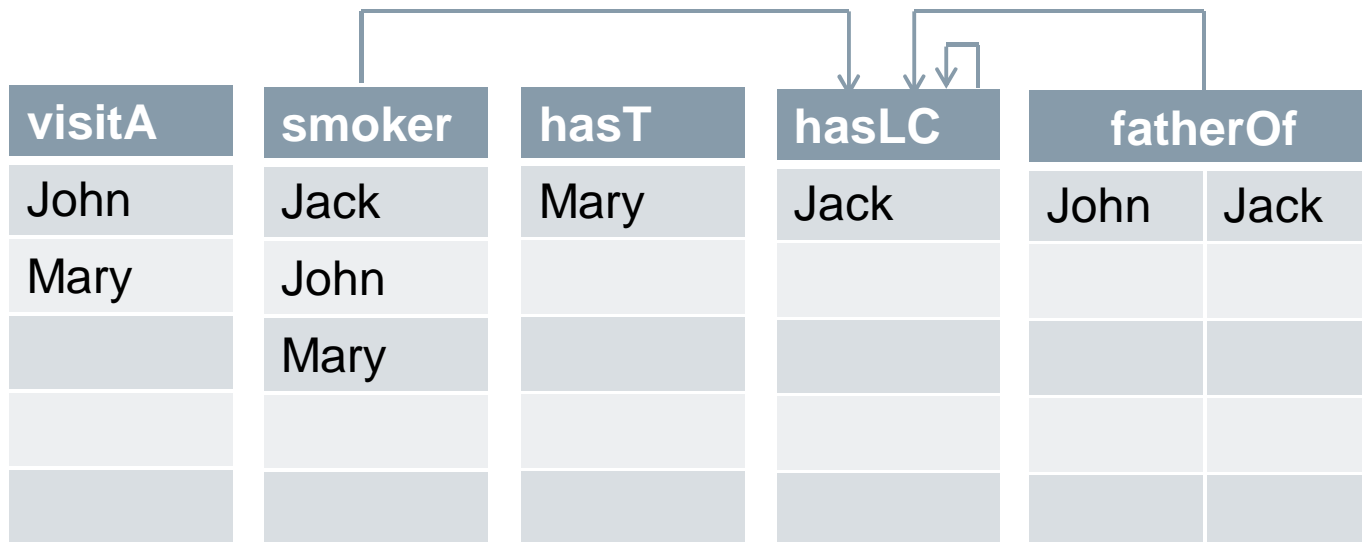


III. Template Bayes Net with Binary Predicates/Relations

- Binary predicates represent relationships between entities:
likes(z, y), knows(z,y), fatherOf(z, y), ...
In the convention we are using the first argument is the subject and the second one the object
- Again: Ξ is the set of all ground atoms that can be formed by all known constants and all predicates. But note that atoms can now have two arguments, e.g., *fatherOf(John, Jack)*

**Database:
Set of Relations (Tables)**

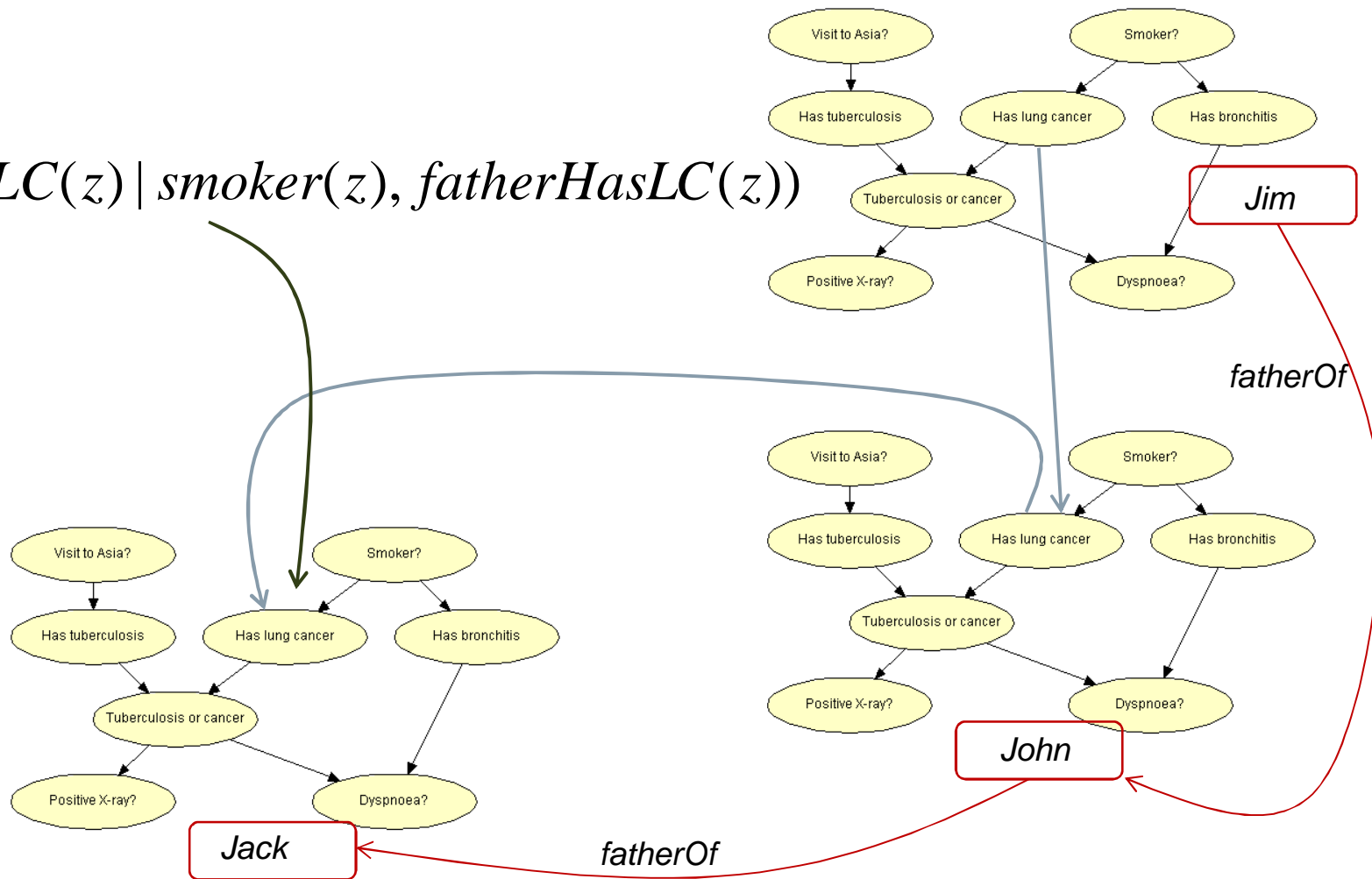
$$P(\text{hasLC}(z) \mid \text{smoker}(z), \exists y. \text{fatherOf}(y, z) \wedge \text{hasLC}(y))$$



- Note that there is a loop on the template level but not on the level of the ground atoms/ground Bayes net, since Jack cannot be his own father

Including Father and Grandfather

$$P(\text{hasLC}(z) \mid \text{smoker}(z), \text{fatherHasLC}(z))$$



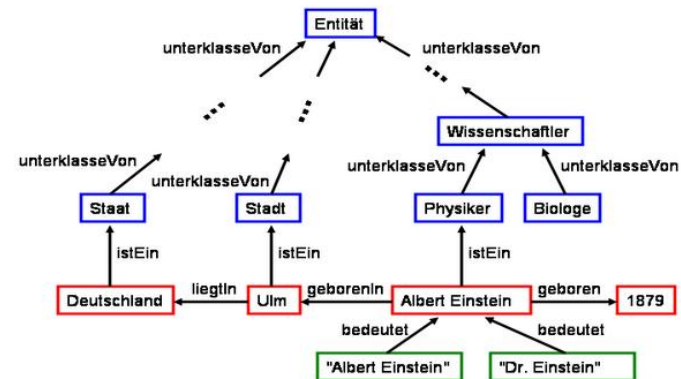
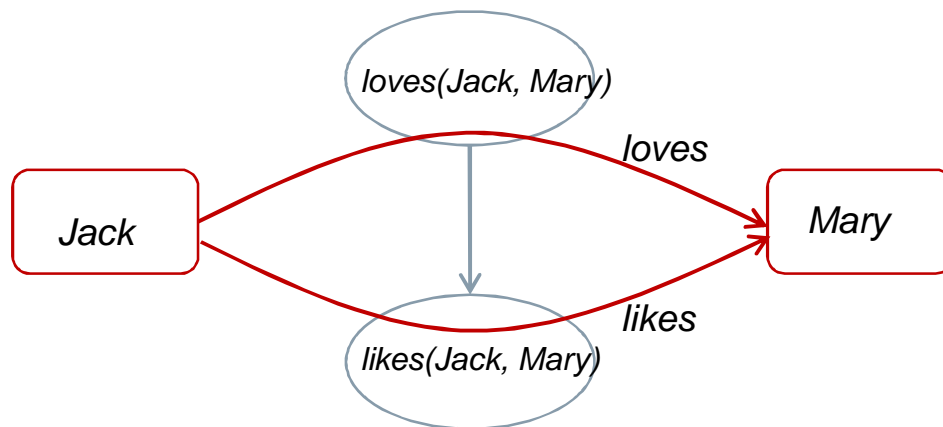
Tensor

- Recall that the **unary ground atoms** can nicely be represented as a **matrix**
- The **binary ground atoms** can be represented as a **3-way tensor** (set of matrices)

	likes			John	Jack	Mary
	John	0	1	0	0	0
	loves			John	Jack	Mary
	John	0	1	0	0	0
	fatherOf			John	Jack	Mary
John	0	1	0	0	0	0
Jack	0	0	0	0	0	0
Mary	0	0	0	0	0	0

Triple Graph

- Alternatively we write a ground atom as a $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ triple, for example $\text{fatherOf}(\text{John}, \text{Jack})$ becomes $\langle \text{Jack}, \text{fatherOf}, \text{John} \rangle$ and form a graph where entities are nodes and a triple is represented as a directed link between subject and object. Known true ground atoms are entered as links in the graph, where the link is labelled by the predicate
- The resulting graph is called a triple graph. Knowledge graphs (DBpedia, Yago, Freebase, Google Knowledge Graph) are special triple graphs. Another example is the RDF (resource description framework) graph used in the linked open data (LOD) cloud



Specifying Dependencies

- Again: Each ground atom must appear exactly once on the left side of a conditional probability (or unconditional, in the case of no parents)
- (subsets of) FOL provides powerful means to derive meaningful views to be used as parent nodes:
 - Datalog (a subset of FOL) can efficiently be executed in relational databases

$$P(\text{hasLC}(z) \mid \text{smoker}(z), \exists y. \text{fatherOf}(y, z) \wedge \text{hasLC}(y))$$

Unary Heads

- What should be the parents? Examples:

$$\forall z. P(\text{smoker}(z) | ???)$$

- $\text{youngAge}(z)$
- $\text{fatherOf}(\text{John}, z)$
- $\exists y. \text{friendOf}(y, z)$
- $\exists y. \text{father}(y, z) \wedge \text{smoker}(y)$

Binary Heads

- What should be the parent nodes?

$$\forall z. \forall y. P(\text{likes}(z, y) | ???)$$

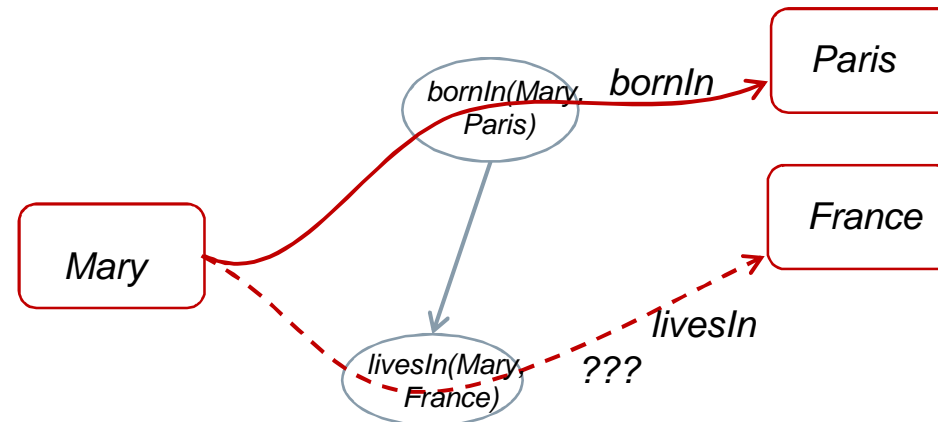
- $\text{youngAge}(z), \text{youngAge}(y) \dots$
- $\text{knows}(z, y)$
- $\exists. \text{like}(z, t) \text{like}(y, t)$

Binary Heads (cont'd)

- “Born in Paris” can predict “Lives in France”

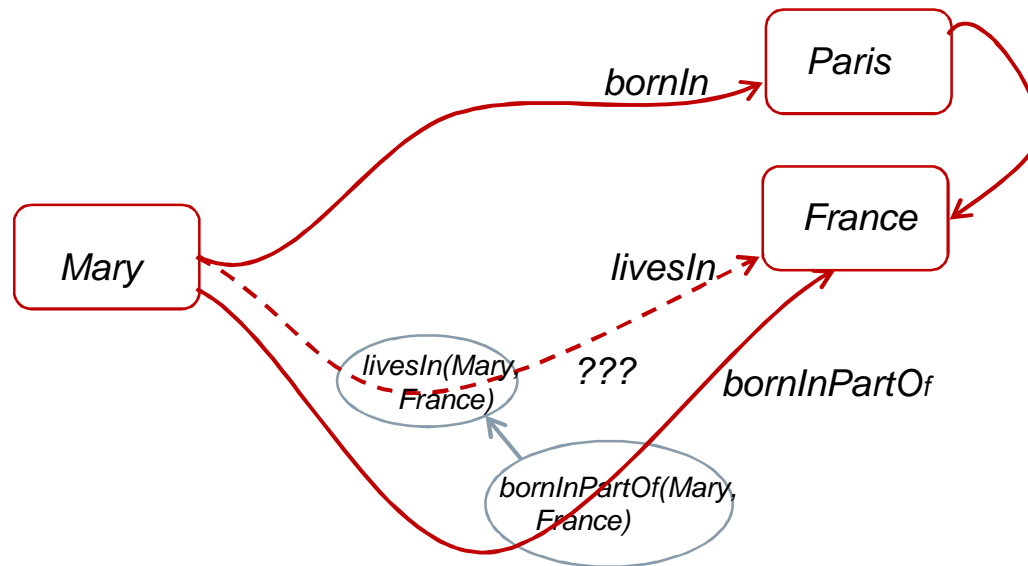
$$P(\text{livesIn}(z, \text{France}) \mid \text{bornIn}(z, \text{Paris}))$$

- But do we need to learn this for all cities and all countries?



Binary Heads (cont'd)

$$P(\text{livesIn}(z, y) \mid \exists t. \text{bornIn}(z, t) \wedge \text{partOf}(t, y))$$



Likelihood (Still Just Counting)

- If all dependencies can be defined on a template level, then the joint probability distribution or likelihood is

$$L = P(\Xi = x) = \prod_{X \in \Xi} P_{\text{predicate}(X)}(X | \text{par}(X))$$

where $\text{predicate}(X)$ returns the predicate of the ground atom represented by X . The important fact is that the conditional probability only depends on the predicate

- Let $\theta_{\text{pred},j,k}$ again be defined as

$$\theta_{\text{pred}=\text{predicate}(X),j,k} = P(X = j | \text{par}(X) = k)$$

and the likelihood function can be written as

$$L = \prod_{\text{pred},j,k} \theta_{\text{pred},j,k}^{N_{\text{pred},j,k}}$$

with the constraint that $\forall \text{pred}, i, k : \sum_j \theta_{\text{pred},j,k} = 1, 0 \leq \theta_{\text{pred},j,k} \leq 1$

- $N_{\text{pred},j,k}$ is the number of times that ground atoms with predicate pred are in state j and their parent nodes are in state k

Inference

- Recall that inference might propagate information in the whole ground Bayes net
- Exact inference is typically not feasible
- Usually, some form of approximate inference is used (loopy belief propagation, MCMC, mean field inference)

Missing Information: Ground Atoms / Tuples

- The world should be complete (that's why it is called the world)
- In reality: The world is part of a greater world
- **States of ground atoms are missing**
 - If missing at random, this can be handled by some form of EM.
 - Often missing is treated as negative evidence (0) in training (closed-world assumption) and one can learn with a complete data set

	visitA	smoker	hatT	hasLC	hasB	posX	hasD
John	1	1	0	0	0	1	1
Mary	1	1	1	0	1	1	0
Jack	?	1	0	1	?	0	1
	0	0	?	0	0	0	0
	0	0	0	0	0	0	0

Missing Information: Entities

- A test entity (e.g., new patient) belonging to the same world introduces new ground atoms.
 - In principal one would need to retrain the whole model but there are efficient approximations by applying learned templates also to the new entity
- A more severe problem is how the training data was generated. Example: if I want to study social interactions between students using a relational model, it makes more sense to study all 10000 students from one university than it would be to study 10000 randomly chosen students from the whole US
 - Ideally one would want to have information on all entities in an isolated community
 - Sampling issues (random sampling, link following sampling, ...) are an important issue in social network analysis, in general

Missing Information: Predicates / Relations

- The logical expressions on the right of the conditioning side can be thought of as defining new predicates/relations

$$P(\text{hasLC}(z) \mid \text{smoker}(z), \exists y. \text{fatherOf}(y, z) \wedge \text{hasLC}(y))$$

$$P(\text{hasLC}(z) \mid \text{smoker}(z), \text{fatherOfHasLC}(z))$$

$$\text{fatherOfHasLC}(z) := \exists y. \text{fatherOf}(y, z) \wedge \text{hasLC}(y)$$

new predicate!

- A cluster analysis or a factor analysis can derive unary predicates (predicate invention)

The World is a Little Bit More Complex

1. Type constraints are typically reliable and can reduce the number of ground atoms dramatically
 - E.g.: only persons can legally get married
2. Subclass hierarchies are typically reliable (a dog is a mammal is a vertebrate is an animal ...)
 - Similarly: *sameAs*, *partOf*
 - One solution: materialization (add ground atoms that can be derived from background knowledge to the database)
3. A large mouse is different from a large planet. An easy solution: instead of just having the predicate *large*, one might want to introduce the predicates *largeRodent* and *largePlanet*
4. Sometimes a good option is to learn several conditional probabilities for a predicate and then use a combination scheme (e.g., noisy-or)

Can We Transfer Knowledge to a New World?

- A model is trained in a hospital in Paris
- We want to apply it to a hospital in Nantes

- Besides the usual problems (different population, ..) this should be possible if the Bayesian templates do not refer to entities (constants) which only make sense in Paris (e.g., a particular physician)

Does the Model have Default Knowledge?

- If I only know that the new object “*dksdjf*” is a dog, the model can infer default knowledge about “*dksdjf*” (typical dog properties)

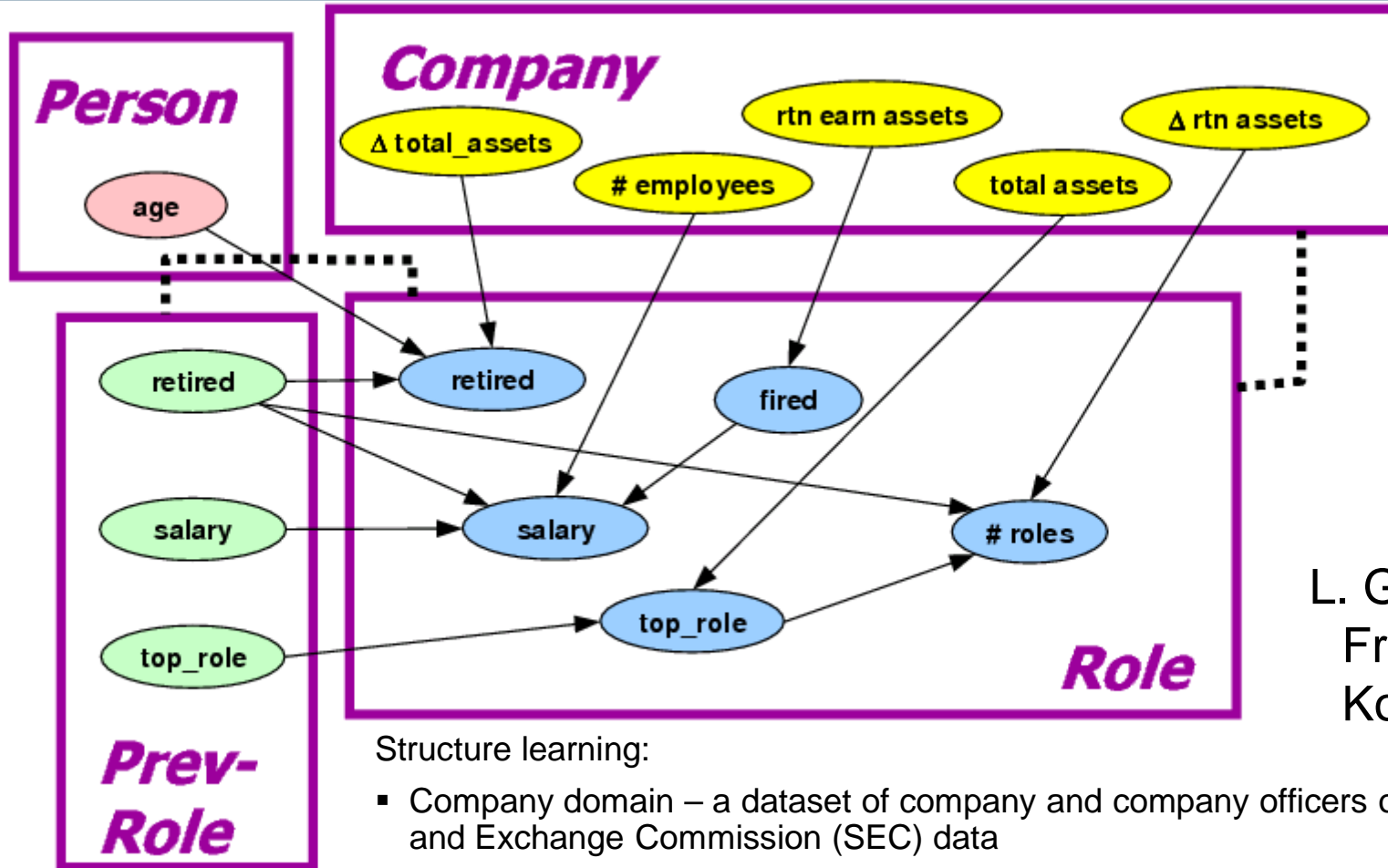
Parameter Learning

- With complete data, maximum likelihood learning can be straightforward
- With missing information, we need to rely on some form of an EM algorithm which is typically based on approximate inference (loopy belief, MCMC, Gibbs, mean field); note that for the E-step, we might need to estimate high-dimensional distributions!

Structural Learning

- In addition to the usual issues in structural learning, we are faced with the problem of searching for interesting views (aggregates, logical expressions)
- The ILP (Inductive Logic Programming) community has developed a number of interesting techniques (e.g., FOIL, Claudian) for deriving interesting views
- The ground Bayes net is not allowed to have directed loops. A sufficient condition is that the Bayes net on the template level does not have directed loops

Company Domain



L. Getoor, N. Friedman, D. Koller, A. Pfeffer

Structure learning:

- Company domain – a dataset of company and company officers obtained from Security and Exchange Commission (SEC) data
- The dataset includes information, gathered over a five year period, about companies, corporate officers in the companies, and the role that the person plays in the company
- For testing, the following classes and table sizes were used: **Company** (20,000), **Person** (40,000), and **Role** (120,000)

TB Domain

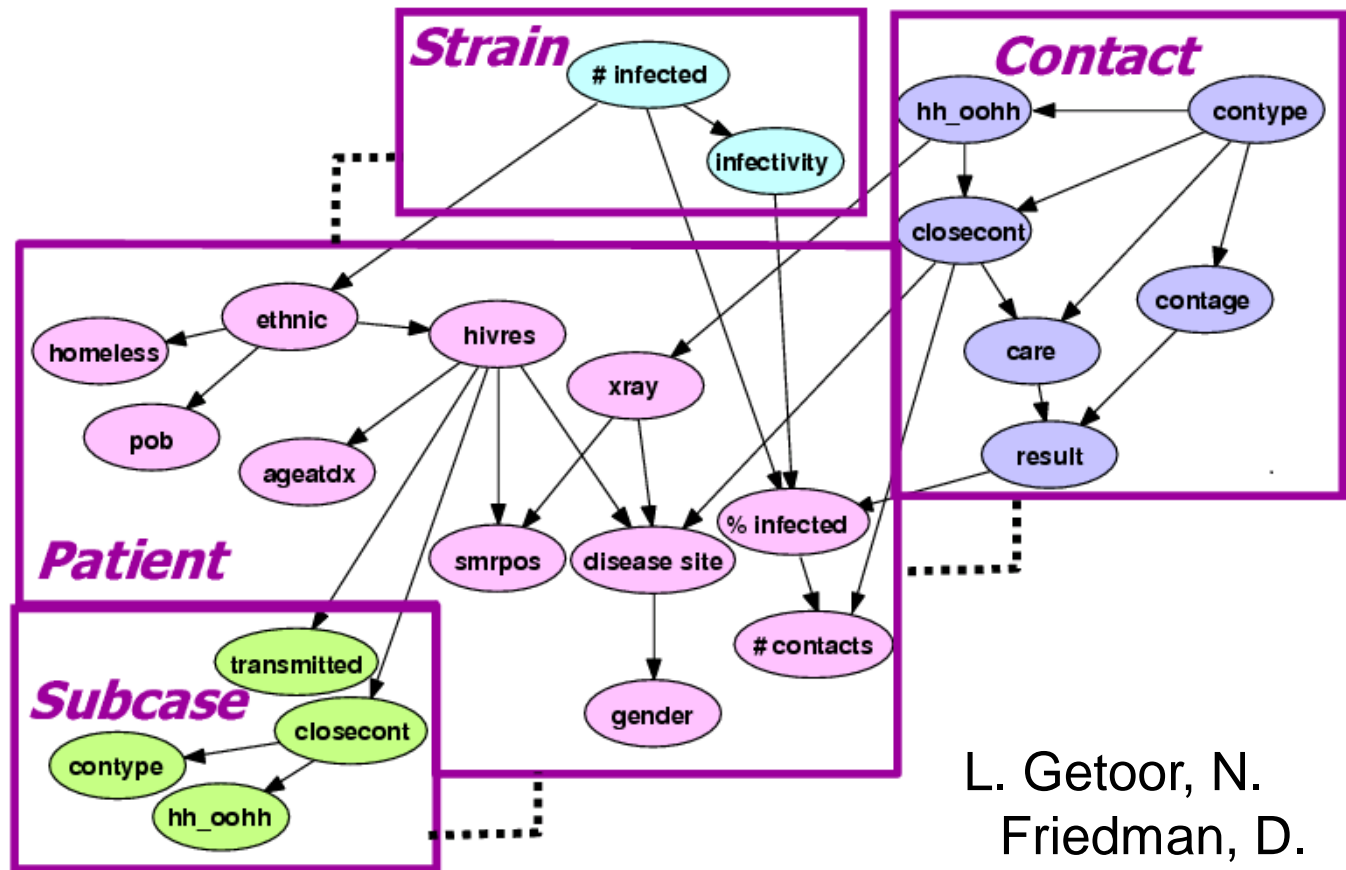


Fig. 1.7. The PRM structure for the TB domain.

L. Getoor, N. Friedman, D. Koller, A. Pfeffer

Structure learning:

- Tuberculosis patient domain – drawn from a database of epidemiological data for 1300 patients from the SF tuberculosis (TB) clinic, and their 2300 contacts
- Relational dependencies, along with other interesting dependencies, were discovered: there is a dependence between the patient’s HIV result and whether he transmits the disease to a contact; there is a correlation between the ethnicity of the patient and the number of patients infected by the strain

Literature

ILP (Inductive Logic Programming) [Dependencies are deterministic or close to deterministic]

- Stephen Muggleton. Inductive logic programming. New Generation Comput., 1991
- J. Ross Quinlan. Learning logical definitions from relations. Machine Learning, 1990
- Saso Dzeroski. Inductive logic programming in a nutshell. In: Introduction to Statistical Relational Learning, 2007

PRMs (Probabilistic Relational Models) [Dependencies defined on a database schema]

- Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. IJCAI, 1999
- Lise Getoor, Nir Friedman, Daphne Koller, Avi Pferrer, and Benjamin Taskar. Probabilistic relational models. In: Introduction to Statistical Relational Learning, 2007

PER (Probabilistic Entity-Relationship Model) [Dependencies defined in an ER Model]

- David Heckerman, Christopher Meek, and Daphne Koller. Probabilistic entity-relationship models, prms, and plate models. In: Introduction to Statistical Relational Learning, 2007

Probabilistic ILP

- Manfred Jaeger. Relational bayesian networks. In UAI, 1997
- Kristian Kersting and Luc De Raedt. Bayesian logic programs. CoRR, cs.AI/0111058, 2001

Conclusions

- Bayes Nets work well in relational learning if there is prior knowledge about possible candidates for relational dependencies
- Problem: it is difficult to avoid loops in the ground Bayes net with symmetric relations such as *friendOf*
- Problem: as always in Bayes nets, one needs to define *a complete system*, i.e., conditional probabilities for each node in the ground Bayes net. This can be very demanding
- Both problems can be solved by Markov logic networks (MLNs)!
- Both Bayes nets and Markov nets are not “off-the-shelf” methods. Off the shelf methods can be used if there is no or only little prior knowledge about relational interactions; relational mixture models (e.g., the IHRM) and relational factor models (e.g., RESCAL) are better models in this situation



II. Markov Networks for Relational Learning

Volker Tresp

Siemens Corporate Technology
Ludwig Maximilian University of Munich

Introduction

- Markov nets are another way of modeling multivariate distributions
- Typically they are better suited for modeling symmetric interactions (e.g., *friendOf*); no concern about directed loops!
- Another advantage is that they do not need to be complete: “Just model what you know about, the rest is filled in with maximum entropy”
- Disadvantages:
 - Maximum likelihood learning with complete data is already non-trivial
 - No causal interpretation

From a Bayes Net to a Markov Net

- A ground relational Bayes net specifies a probability distribution in the form

$$L = P(\Xi = x) = \prod_{X \in \Xi} P(X \mid \text{par}(X)) = \prod_{pred, j, k} \theta_{pred, j, k}^{N_{pred, j, k}} \quad j \in \{0, 1\}$$

- Here X stands for a ground atom and $\theta_{pred, j, k}$ is the probability that a ground atom with $pred = \text{predicate}(X)$ is in state j given that its parents are in state k (see *previous lecture*); $\text{predicate}(X)$ returns the predicate of the ground atom X
- Note that this is also the definition of the complete data likelihood
- We can write the complete data likelihood in exponential form

$$L = \exp \sum_{X \in \Xi} \log P(X \mid \text{par}(X)) = \exp \sum_{pred, j, k} N_{pred, j, k} \log \theta_{pred, j, k}$$

From a Bayes Net to a Markov Net (cont'd)

- In learning, one would need to enforce $0 \leq \theta_{pred,j,k} \leq 1$ $\sum_j \theta_{pred,j,k} = 1$
- One can parameterize

$$\theta_{pred,j,k} = \exp w_{pred,j,k}$$

$$L = \frac{1}{\prod_{pred,k} Z_{pred,k}} \exp \sum_{pred,j,k} w_{pred,j,k} N_{pred,j,k}$$

$$Z_{pred,k} = \left(\exp w_{pred,j=1,k} + \exp w_{pred,j=0,k} \right)^N$$

- Now the model is properly normalized for any parameter values

Markov Logic Network (MLN)

- Recall that $N_{pred,j,k}$

is the number of times that in the data the logical formula

$$(\text{predicate}(X) = pred) \wedge (X = j) \wedge (\text{par}(X) = k)$$

is true

- In MLN one can use *any* FOL formula F_i (not just the ones derived from Bayes nets)
- MLN does not require local normalization, i.e. an interpretation of the terms as local conditional probabilities; it requires **global normalization**
- MLN does not require a specific number of formulae

Markov Logic Network (MLN) (cont'd)

- For an MLN model, we get

$$P(\Xi = x) = L = \frac{1}{Z(w)} \exp \sum_i w_i N_i(x)$$

where $N_i(x)$ is the number of true groundings of formula F_i

Formulae with only Unary Predicates

▪ Example: $F_i : \forall z. \text{smoker}(z) \wedge \text{hasB}(z)$

▪ Then $N_i(x)$ is the number of times that in the set of patients, a patient is a smoker and has bronchitis:

$$\begin{aligned} & \text{smoker}(Jack) \wedge \text{hasB}(Jack), \\ & \text{smoker}(Mary) \wedge \text{hasB}(Mary), \dots \end{aligned}$$

▪ Note that, in contrast to deterministic logic, there is no problem in MLN if a formula is sometimes false. w_i corresponds to the degree that formula F_i is supported in the training data

Formulae with Binary Predicates

- Nothing really new
- Example:

$$F_i : \forall z. hasLC(z) \wedge \exists y. fatherOf(y, z) \wedge hasLC(y)$$

- $N_i(x)$ is the number of times that in the set of patients, a patient has lung cancer and her/his father has lung cancer

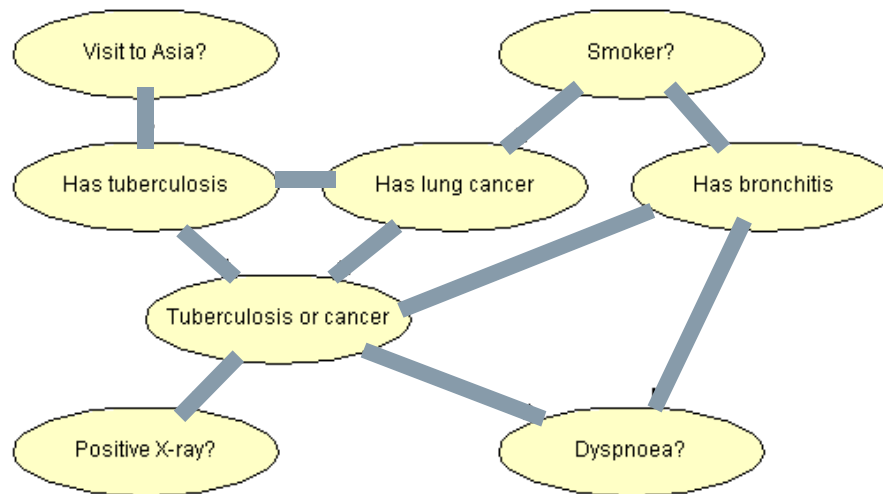
$$hasLC(Jack) \wedge fatherOf(John, Jack) \wedge hasLC(John), \\ hasLC(Mary) \wedge fatherOf(Jim, Mary) \wedge hasLC(Jim), \dots$$

Comments

- + We do not have to worry any more about directed loops!
- + We do not have to worry anymore about completeness! There can be any number of formulae (from none to more than there are ground atoms)
 - Comment: All distributions with the same exponent have the same probability (maximum entropy principle). Example: with no formula *each configuration of the node states* has the same probability! The *marginal probability* of each node being in state 1 is then 0.5
- - There is no interpretation of a local conditional distribution (no causal interpretation)
- - The partition function is problematic in learning

MLN Network for the Chest Clinic

One can generate a graphical representation where the random variables are represented as nodes and all nodes in a ground formula are cliques (fully connected subnets; links are undirected)



Inference

Same techniques as in Bayesian models (MCMC, Gibbs, loopy belief, mean field)

Parameter Learning

- Recall that in Bayesian nets, parameter learning with complete data was trivial
- In MLNs parameter learning is difficult due to the global normalization constant which is a sum over all states
- Typical approach: closed-world assumption and optimization of a pseudo-likelihood

Structural Learning

- We do not have to worry about directed loops or parameter constraints
- Formulae are derived from ILP techniques (FOIL, Claudian)

Conclusions

- Highly recommended reading:
 - Matthew Richardson and Pedro Domingos. Markov logic networks. Machine Learning, 2006
- Great software support: Alchemy
- Highly popular in logics community
- Success sort of depends on good hand-crafted features
- Sometimes structural learning finds good features automatically



III. Mixture Models for Relational Learning

Volker Tresp

Siemens Corporate Technology
Ludwig Maximilian University of Munich

Introduction

- We have discussed generalizations of **Bayes nets** towards relational domains
- We have discussed generalizations of **Markov nets** towards relational domains
- Both perform well if there is good prior knowledge available about relational dependencies, but both approaches are not as suitable *as off-the-shelf methods*
- Here we discuss generalizations of **statistical mixture models** towards relational domains
 - Suitable as off-the-shelf method

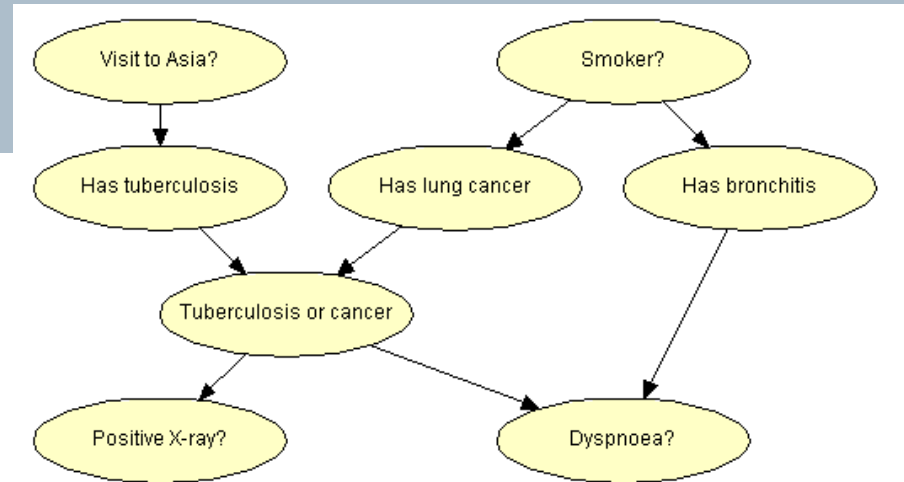
Classical Mixture Model

- With each entity (data point) a latent discrete variable H is associated. For a given data point, H is in a particular state, but this state is unknown; it is not recorded in the data
- All observable random variables are children of the latent variable. The probability distribution for a random variable X_i is then

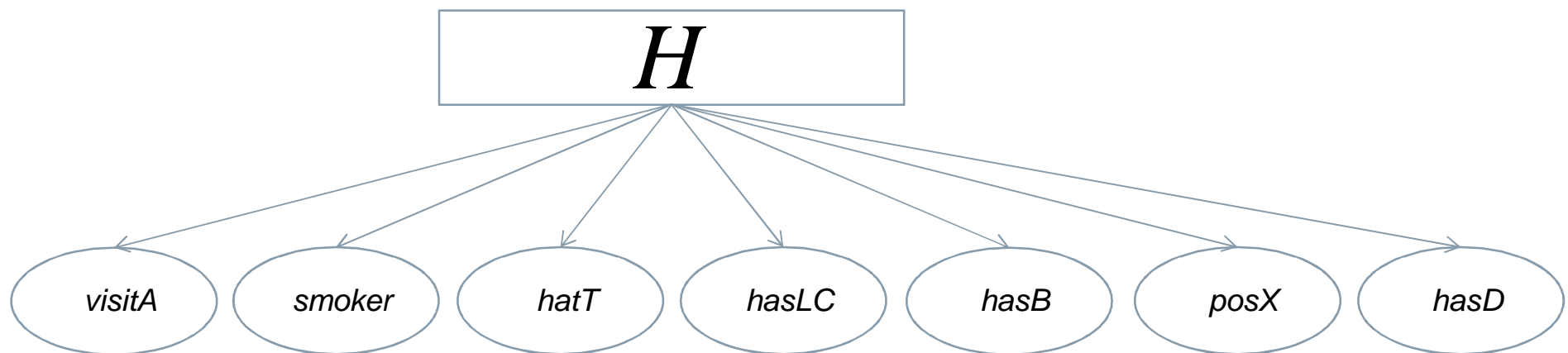
$$P(X_i = j) = \sum_h P(H = h)P(X_i = j|H = h)$$

- Here $j \in \{0, 1\}$. An advantage is the great simplicity of the model: no need to think about rules, conditional independencies, loops, or global partition functions

Graphical Model



- Assumption: H represents latent information that can explain all visible attributes
- This is a Bayes net (only that H is unknown)



Maximum Likelihood Learning

- Let's assume a multinomial model for the latent variable

$$P(H = h) = \kappa_h \quad \text{with } 0 \leq \kappa_h \leq 1 \text{ and } \sum_h \kappa_h = 1$$

- We are interested in binary observable variables with a Bernoulli distribution, with

$$P(X_i = 1|H = h) = \theta_{i,h} \quad \text{with } 0 \leq \theta_{i,h} \leq 1$$

- Since this is a Bayes net, it is easy to write the complete-data maximum likelihood solution as

$$\hat{\kappa}_h = \frac{N_h}{N} \quad \hat{\theta}_{i,h} = \frac{N_{i,j=1,h}}{N_{i,j=1,h} + N_{i,j=0,h}}$$

Here, N is the number of data points, N_h is the number of times that in the data the latent variable is in state h . $N_{i,j=1,h}$ is the number of times that the latent variable is in state h and $X_i = 1$. $N_{i,j=0,h}$ is the number of times that the latent variable is in state h and $X_i = 0$

Expected Counts

- Due to the latent variables, we have to use expected counts, i.e.,

$$E(N_h) = \sum_{l=1}^N P(H = h | \{X_{i'} = j_{l,i'}\}_{i'=1}^M)$$

$$E(N_{i,j=1,h}) = \sum_{l:j_{l,i}=1} P(H = h | \{X_{i'} = j_{l,i'}\}_{i'=1}^M)$$

$$E(N_{i,j=0,h}) = \sum_{l:j_{l,i}=0} P(H = h | \{X_{i'} = j_{l,i'}\}_{i'=1}^M)$$

Mixture Models with Unary Variables

- Nothing changes, except for the interpretation. Let $upred$ be a generic unary predicate
- With $P(upred(z) = 1|H = h) = \theta_{upred,h}$ we get

$$P(upred(z) = 1) = \sum_h P(H = h)P(upred(z) = 1|H = h)$$

Mixture Models with Binary Predicates

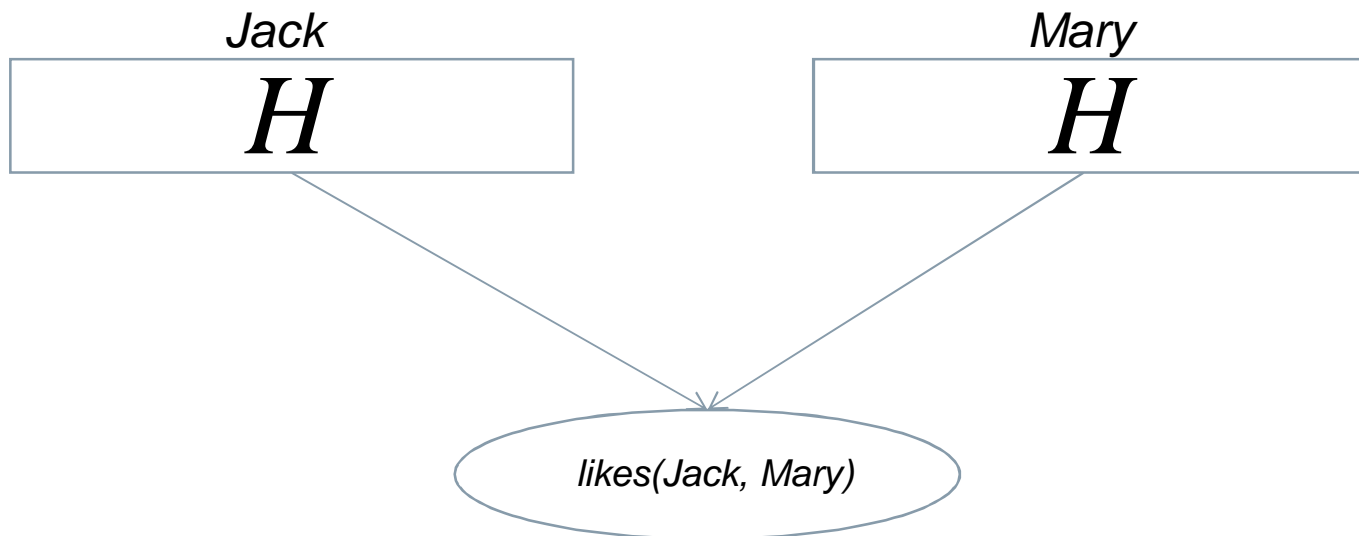
- We maintain that each entity has an associated latent discrete variable
- But now a ground atom depends on the state of the latent variables of **both involved entities**:

$$P(\text{bpred}(z, y) = 1 | H_z = h_z, H_y = h_y) = \theta_{\text{bpred}, h_z, h_y}$$

- *bpred* stands for a binary predicate

Graphical Model

- Assumption: H represents latent information that can explain the binary ground atoms



Likelihood

- Since this is a Bayes net, it is easy to write the complete-data maximum likelihood solution as

$$\hat{\kappa}_h = \frac{N_h}{N}$$

$$\hat{\theta}_{upred,h} = \frac{N_{upred,j=1,h}}{N_{upred,j=1,h} + N_{upred,j=0,h}}$$

$$\hat{\theta}_{bpred,h_z,h_y} = \frac{N_{bpred,j=1,h_z,h_y}}{N_{bpred,j=1,h_z,h_y} + N_{bpred,j=0,h_z,h_y}}$$

- Here, N is the number of entities under consideration, N_h is the number of times that the latent variable is in state h , in all entities. $N_{upred,j=1,h}$ is the number of times that $upred(z)$ is true when the latent variable associated with z is in state h . $N_{bpred,j=1,h_z,h_y}$ is the number of times that $bpred(z, y)$ is true when the latent variable associated with z is in state h_z and the latent variable associated with y is in state h_y .

EM Learning

- Since H is latent, we have to use expected counts, i.e.,

$$E(N_h) = \sum_z P(H_z = h | \Xi)$$

$$E(N_{upred, j=1, h}) = \sum_{z: upred(z)=1} P(H_z = h | \Xi)$$

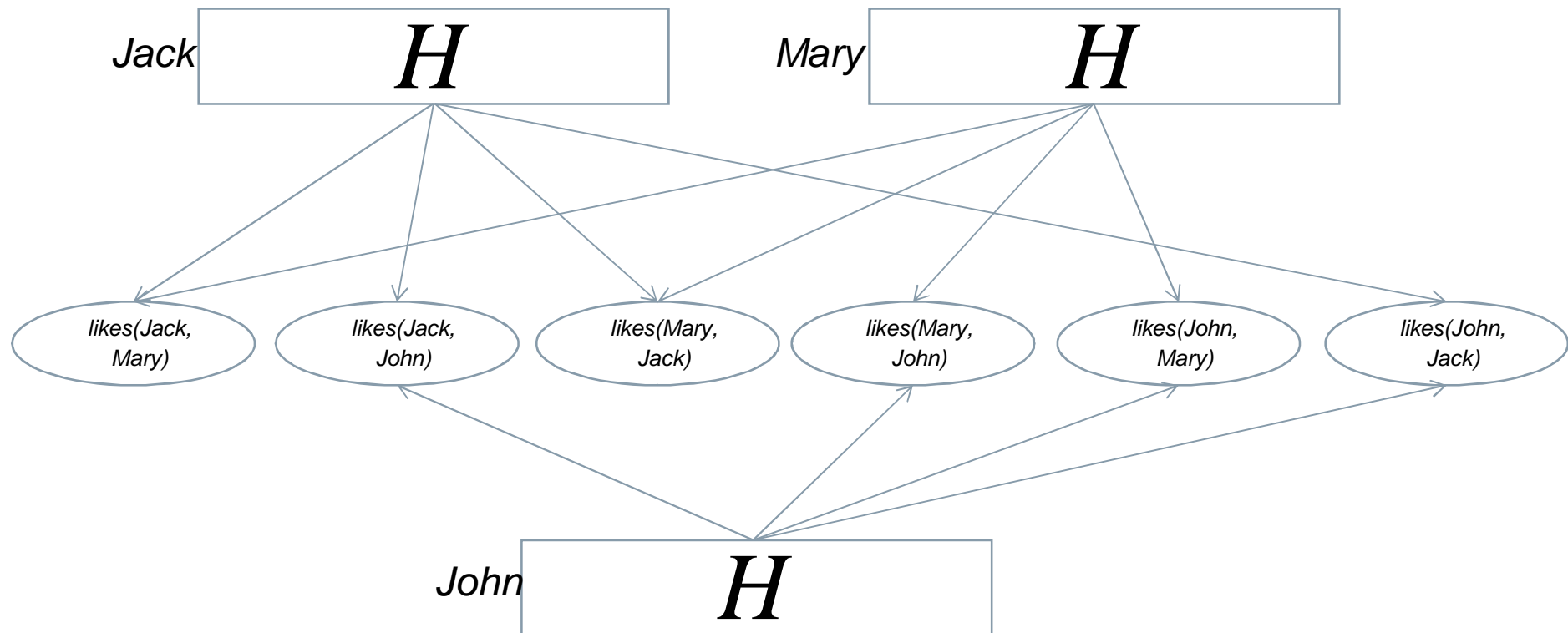
$$E(N_{bpred, j=1, h_z, h_y}) = \sum_{z, y: bpred(z, y)=1} P(H_z = h_z, H_y = h_y | \Xi)$$

- The technical difficulty is that the latent variables are conditioned on the whole world $\Xi = x$, i.e., on all training data

Global Propagation of Information

Free flow of information:

- The parent nodes (here, the *H*-nodes) block information when they are known but permit the flow of information when they are unknown (the case here)
- Collider nodes (here, the *binary ground atoms*) block information when they are unknown but permit the flow of information when they are known (the case here with a closed-world assumption)



Infinite Hidden Relational Model / Infinite Relational Model

- Exact EM is not suitable due to the expensive E-step
- Typically, Gibbs sampling or mean field is employed to approximate the E-step
- In a fully Bayesian model we apply a Dirichlet prior on K
- We can make the transition to infinitely many states and obtain *Dirichlet process mixture models* in form of the IHRM/IRM models: these are infinite models (nonparametric Bayesian models) in which the number of hidden states is determined in the sampling process!
- In Gibbs sampling, the parameters θ can be integrated out

Inference

- Note that the latent variables are all unknown
- As just discussed, there is global propagation of information in the network
- Thus, the model can use information about the context of an atom, e.g., about a friend's friend, without explicit aggregation (collective learning)
- Approximate inference is used (Gibbs sampling, mean field)

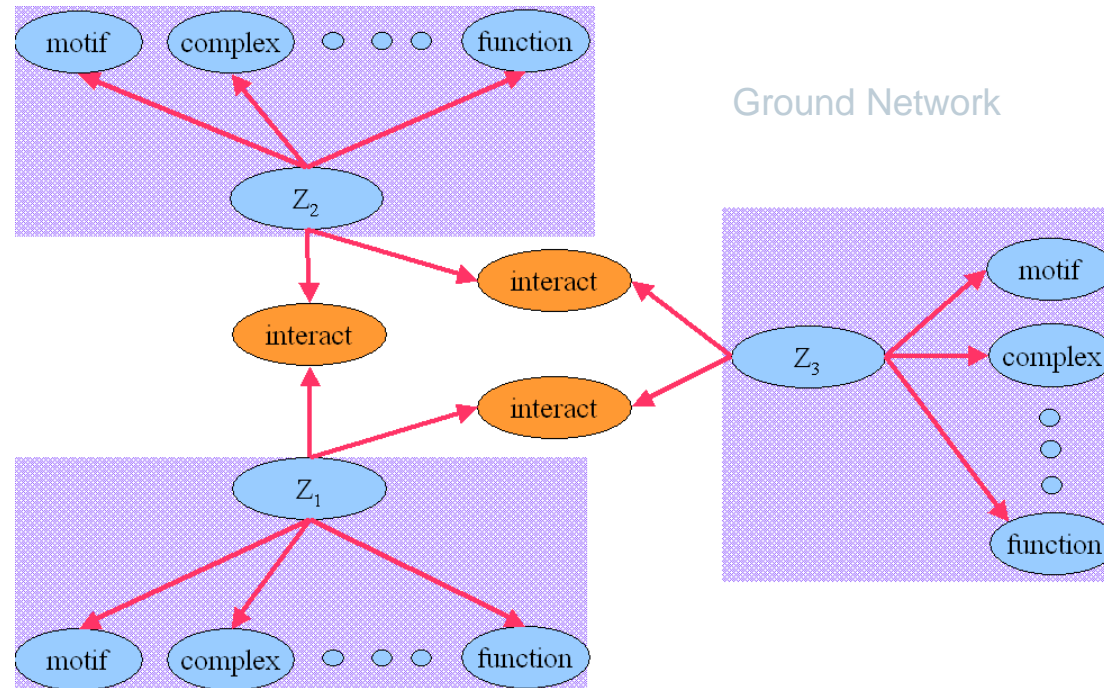
Structural Learning

- There is no need for structural learning!
- The structure is defined by the relational model

Gene Interaction and Gene Function

- **Tasks**
 - Cluster analysis
 - Prediction of gene functions given information on the gene level and the protein level, as well as information on interactions between the genes
- **Attribute data: CYGD** (Comprehensive Yeast Genome Database) from MIPS (Munich Information Center for Protein Sequences)
 - 1000 Genes
 - Attributes: Chromosome, Motif, Essential, Class, Phenotype, Complex, Function
- **Interaction data: DIP** (data base of interacting proteins)

IHRM Model



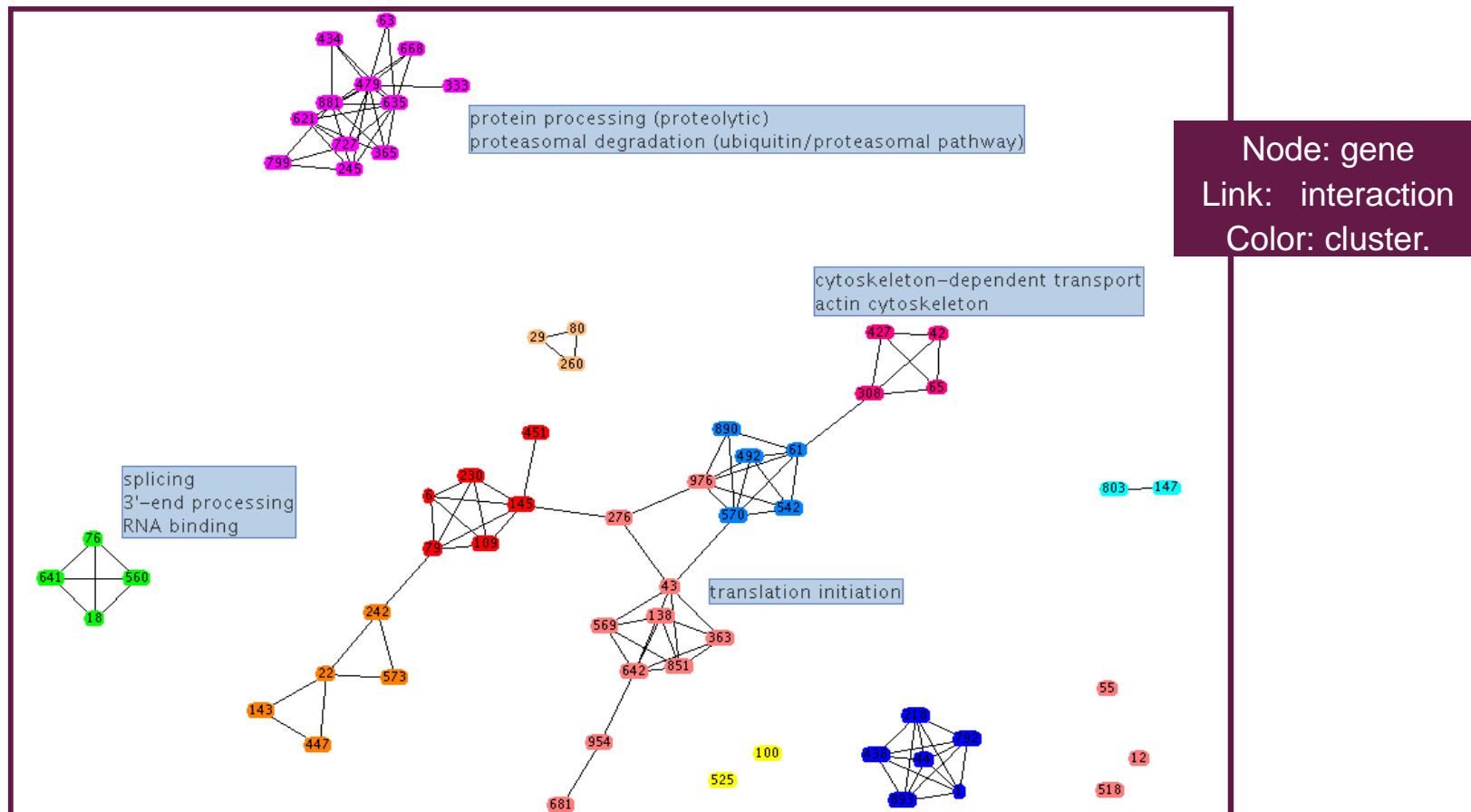
Task: Genes (1243) have one or more functions (14)[1-4] (cell growth, cell organization, transport, ...) to be predicted; 862 for genes for training, 381 for testing
Genes might interact with one another

For a gene one or more phenotypes (11)[1-6] are observed in the organism
How the expression of the gene can complex with others to form a larger protein (56)[1-3]
The protein coded by the gene might belong to one or more structural categories (24) [1-2]

A gene might contain one or more characteristic motifs (351) [1-6] (information about the amino acid sequence of the protein)
Gene attributes are: essential (an organism with a mutation can survive?), which chromosome

Cluster Structure

- Some gene clusters: the genes in the same cluster have dense interactions; but the genes in the different clusters have rare interactions



Relevance of Attributes and Relationships

The importance of a variety of relationships in function prediction of genes

Relationships	Prediction Accuracy (%) (without the relationship)	Importance
Complex	91.13	197
Interaction	92.14	100
Structural Category	92.61	55
Phenotype	92.71	45
Attributes of Gene	93.08	10
Motif	93.12	6

References

Stochastic Block Model

- Krzysztof Nowicki and Tom A. B. Snijder. Estimation and Prediction for Stochastic Blockstructures. *Journal of the American Statistical Association*, 2001

Infinite Models (nonparametric models)

- Zhao Xu, Volker Tresp, Kai Yu, and Hans-Peter Kriegel. Infinite hidden relational models. *UAI*, 2006
- Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. *AAAI*, 2006

Extensions

- Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 2008

Conclusions

- Relational mixture models have many attractive properties
- They are useful off-the shelf approaches
- Good results on some problems
- Obtaining convergence can be tricky
- In our opinion, the best off-the-shelf approaches with great scalability and great predictive results are based on factorization approaches, e.g., the RESCAL model described in the following lecture



IV. Factor Models for Relational Learning

Volker Tresp

Siemens Corporate Technology
Ludwig Maximilian University of Munich

Introduction

- Here we discuss generalizations of **statistical factor models** towards relational domains
 - Suitable as off-the-shelf methods
 - Highly scalable and excellent predictive performance

Classical Factor Model

- We assume for each random variable a model of the form

$$P(X_{l,i} | f_{l,i})$$

$X_{l,i}$ random variable with index i in data point with index l

- The most important special cases are a Bernoulli model and a Gaussian model

$$\text{Bernoulli: } P(X_{l,i} | f_{l,i}) = f_{l,i} \quad 0 \leq f_{l,i} \leq 1$$

$$\text{Gauss: } P(X_{l,i} | f_{l,i}) \propto N(f_{l,i}, \sigma^2)$$

Classical Factor Model

- We assume for each random variable a model of the form

$$f_{l,i} = b_i^T a_l$$

a_l : r – dimensional vector of latent factors specific to data point l

w_i : r – dimensional latent vector specific to random variable i

- Note that both the dimension-specific and the data point specific factors are unknown and have to be learned from data
- The solution is not unique; one possible solution can be computed via singular value decomposition (SVD)

Factor Model for Unary Relations

- We assume for each random variable a model of the form

$$P(\textit{upred}(z) | f_{\textit{upred},z}) \quad f_{\textit{upred},z} = v_{\textit{upred}}^T a_z$$

a_z : r – dimensional vector of latent factors that describe entity z

$v_{\textit{upred}}$: r – dimensional latent vector specific to \textit{upred}

Matrix Algebra

- Recall that we can write the possible world unary ground atoms as a matrix

M is a matrix where

$$(M)_{z,upred} = upred(z)$$

F is a matrix with the same dimension

$P(M | F)$ is an element-wise conditional probability

$F = V^T A$ describes the matrix decomposition

V is a matrix with

$$(V)_{upred,k} = v_{upred,k}$$

A is a matrix that contains the latent entity factors with

$$(A)_{z,k} = a_{z,k}$$

Factor Model for Binary Relations

- We assume for each random variable a model of the form

$$P(bpred(z, y) | f_{bpred,z,y})$$

- We can now assume that we should consider all interactions between the latent representations of the two involved entities, and one models

$$f_{bpred,z,y} = \sum_j \sum_k w_{bpred,j,k} a_{z,j} a_{y,k}$$

Tensor Algebra

- Recall that we can write the possible world binary ground atoms as a tensor

X

is a three-way tensor where

$$(\mathbf{X})_{z,y,bpred} = bpred(z, y)$$

F

is a tensor with the same dimensions

$$P(\mathbf{X} | \mathbf{F})$$

is an element-wise conditional probability

$$\mathbf{F} = \mathbf{R} \times_1 A \times_2 A$$

Describes the tensor decomposition

R

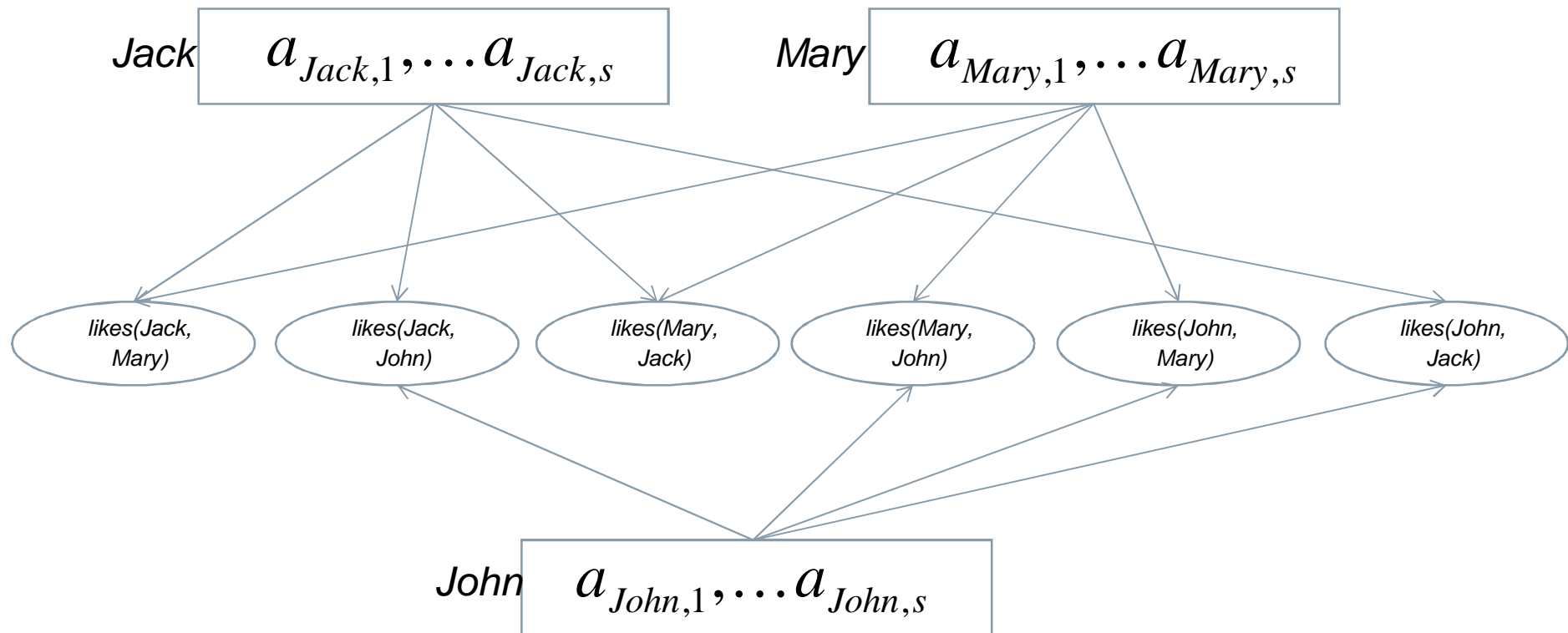
is the core tensor with

$$(\mathbf{R})_{j,k,bpred} = w_{bpred,j,k}$$

The Latent Structure is Similar to the IHRM/IRM

Free flow of information:

- The parent nodes (here, the a -nodes) block information when they are known but permit the flow of information when they are unknown (the case here)
- Collider nodes (here, the *binary ground atoms*) block information when they are unknown but permit the flow of information when they are known (the case here with a closed-world assumption)



RESCAL: Cost Functions and Parameter Learning

- The model I have just described is known as the RESCAL model
 - 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
- The cost function and the parameter optimization are described in the next lecture on „Machine Learning with Knowledge Graphs”
- Note that the A matrix is shared between the tensor model for the binary relations and the matrix model for the unary relations

Inference

- After learning the ground atoms are all independent

Structure Learning

- As in the IHRM/IRM, there is no structure learning

Conclusions

- The RESCAL model has excellent performance and scales well to large data sets
- For more details see also the following lecture on *Machine Learning with Knowledge Graphs*



V. Machine Learning with Knowledge Graphs

Volker Tresp

Siemens Corporate Technology
Ludwig Maximilian University of Munich

Joint work with Maximilian Nickel

With contributions from Xueyan Jiang and Denis Krompass

Prelude

- My background is in Machine Learning and I got involved in Semantic Web projects maybe 6 years ago
- Learning about the Semantic Web clarified my thinking about many things dramatically

- Immediate love affair with RDF
 - Nothing is ever wrong
 - No contradictions

IRMLeS 2009
IRMLeS 2009: 1st ESWC Workshop on Inductive Reasoning and Machine Learning on the Semantic Web

Sections

- Welcome!
- Organization**
- Submission Details
- Important Dates
- Workshop Program
- Invited speakers
- Accepted papers
- Photos
- News
- Sitemap

1823
days since
Workshop day

Organization

Organizing Committee

- [Claudia d'Amato](#), University of Bari, Italy
- [Nicola Fanizzi](#), University of Bari, Italy
- [Marko Grobelnik](#), Jožef Stefan Institute, Slovenia
- [Agnieszka Ławrynowicz](#), Poznan University of Technology, Poland
- [Vojtěch Svátek](#), University of Economics, Prague, Czech Republic

Program Committee

Prelude

- My background is in Machine Learning and I got involved in Semantic Web projects maybe 6 years ago
- Learning about the Semantic Web clarified my thinking about many things dramatically

- Immediate love affair with RDF
 - Nothing is ever wrong
 - No contradictions



IRMLeS 2009
IRMLeS 2009: 1st ESWC Workshop on Inductive Reasoning and Machine Learning on the Semantic Web

Sections

- Welcome!
- Organization
- Submission Details
- Important Dates
- Workshop Program
- Invited speakers
- Accepted papers
- Photos
- News
- Sitemap

Organization

Organizing Committee

- [Claudia d'Amato](#), University of Bari, Italy
- [Nicola Fanizzi](#), University of Bari, Italy
- [Marko Grobelnik](#), Jožef Stefan Institute, Slovenia
- [Agnieszka Ławrynowicz](#), Poznan University of Technology, Poland
- [Vojtěch Svátek](#), University of Economics, Prague, Czech Republic

1823
days since
Workshop day

Program Committee

Overview

- **Why Machine Learning needs Knowledge Graphs**
- Statistical Relational Learning
- Learning with the YAGO Knowledge Graph
- Towards Relevant Use Cases

What is Machine Learning?

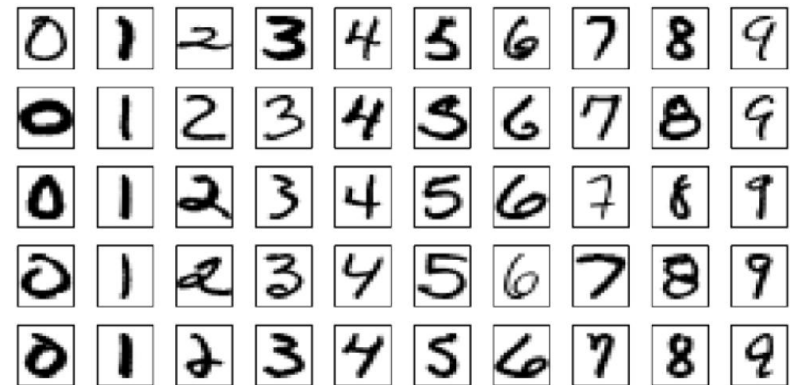
Machine Learning versus Statistics versus Data Mining

- Statistics focuses on interpretable parameters
- Data mining focuses on the discovery of meaningful patterns
- Machine Learning focuses on prediction accuracy

Classification

Classification is the work horse of machine learning

- Predict class memberships for many objects
- Very powerful
- Surprisingly general



Typical Classifiers

Predicting class k for input z_l $P(x^k(z_l) = 1) \leftarrow f^k(z_l)$

Fixed basis functions

$$f^k(z_l) = \sum_{m=1}^M w_m^k b_m(z_l)$$

Kernels

$$f^k(z_l) = \sum_{n=1}^N v_m^k k(z_l, z_n)$$

Neural Networks

$$f^k(z_l) = NN_{deep}(z_l)$$

Really the same things; deep learners would call the shallow

- 10 layers with 1000 neurons per layer
- Currently the hottest thing!

Deep Learning Neural Networks

Scientists See Promise in Deep-Learning Programs



A voice recognition program translated a speech given by Richard F. Rashid, Mi Chinese.

By JOHN MARKOFF
Published: November 23, 2012

Using an artificial intelligence technique inspired by theories of how the brain recognizes patterns, technology companies are reporting startling gains in fields as diverse as computer vision, speech recognition and the identification of promising new drugs for designing drugs.

[点击查看本文中文版。](#)

Connect With Us on Social Media

@nytimescience on Twitter.

Science Reporters and Editors on Twitter

Like the science desk on Facebook.



The advances have led to widespread enthusiasm among researchers in design software to perform human activities like seeing, listening and thinking. They offer the promise of machines that converse with humans and perform tasks like driving cars and working in factories, [raising the specter of automated robots that could replace human workers.](#)



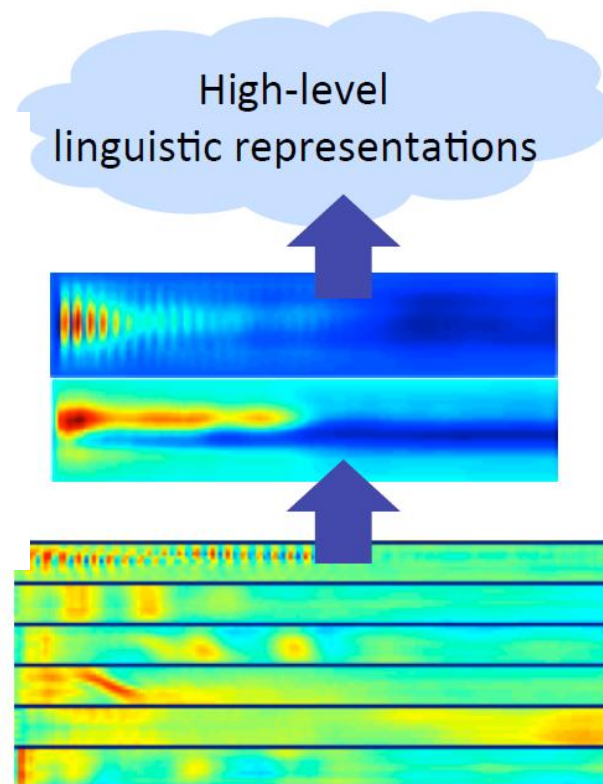
PRINT

SINGLE PAGE

REPRINTS

THE WAY WAY BACK
WATCH TRAILER

- Google, Microsoft, Facebook, Baidu are all investing heavily in deep learning



Detecting Cats in Images

- Best performing in detecting cats in images and videos (Andrew Ng)



Where from here?

- A deep learning network sees more cats than any child but is not as good at this task
- Deep Learning community: we need better unsupervised learning to pre-structure the network

Image of cats

- Maybe we would say: we need background knowledge
- Also: we do not just want to *detect* cats!

Challenges

Predict all classes: „This is a cat!“ „This is a dog!“
„This is a house!“ ...

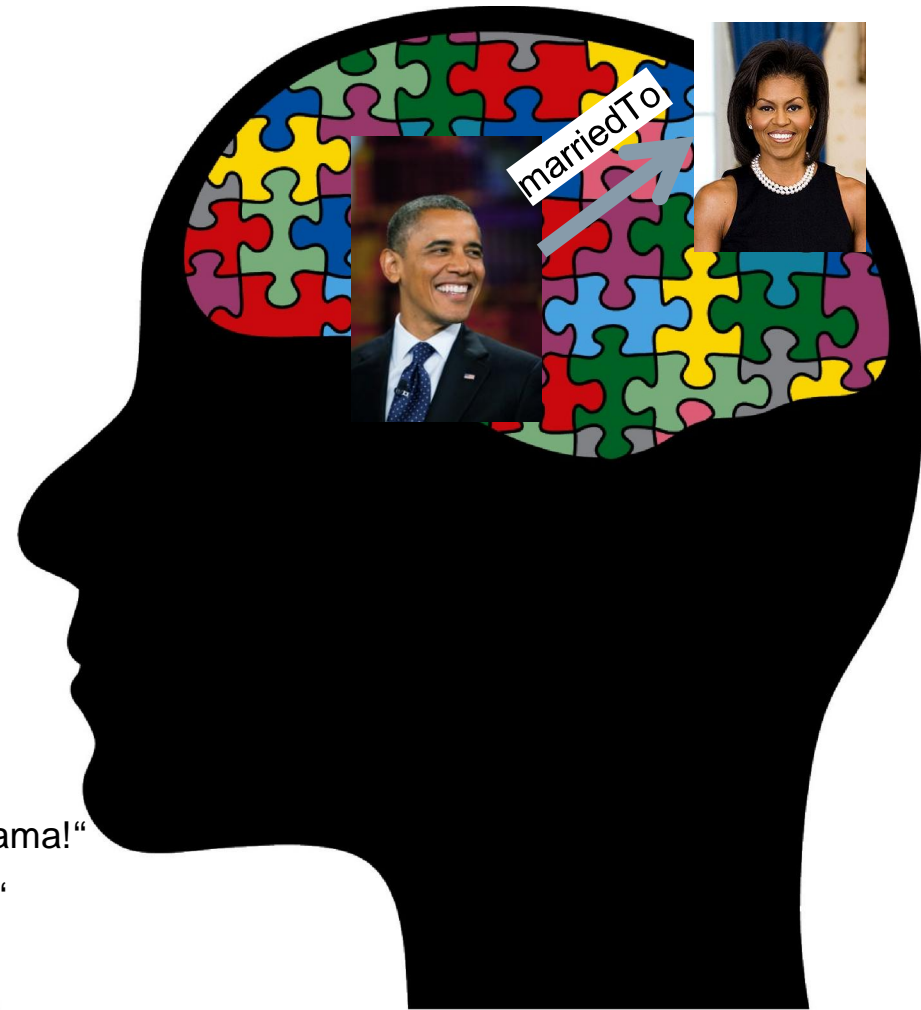
Recognize specific entities: „This my cat Max!“
[In our experiments 10^7]

Images of cats

Predict all attributes: „Max is evil!“

Predict all relationships: „Max likes Mary!“
[In our experiments 10^{14}] [#of synapses]

Vision



„You must be president Obama!“
„How is your wife Michelle?“

γλαῦκας εἰς Ἀθήνας κομίζειν



Requirement: Understanding of the World

- We need to know about the entities, attributes and classes in the world, and the various relationships that do or might exist between those
- We need ontologies!

Biomedical Ontologies

International Statistical Classification of Diseases and Related Health Problems (ICD)

- Used extensively in billing

SNOMED Clinical Terms (SNOMED CT)

- A systematically organized computer processable collection of medical terms providing codes, terms, synonyms and definitions used in clinical documentation and reporting.
- Application: *EHR*

RadLex

- Unified language of radiology terms for standardized indexing and retrieval of radiology information resources

Open Biomedical Ontologies (OBO)

- Controlled vocabularies for shared use across different biological and medical domains
- Gene Ontology (GO) is a part (genes and gene products)



Example GO term [\[edit\]](#)

```
id:          GO:0000016
name:        lactase activity
namespace:   molecular_function
def:         "Catalysis of the reaction: lactose + H2O = D-glucose + D-galactose." [EC:3.2.1.108]
synonym:     "lactase-phlorizin hydrolase activity" BROAD [EC:3.2.1.108]
synonym:     "lactose galactohydrolase activity" EXACT [EC:3.2.1.108]
xref:        EC:3.2.1.108
xref:        MetaCyc:LACTASE-RXN
xref:        Reactome:20536
is_a:        GO:0004553 ! hydrolase activity, hydrolyzing O-glycosyl compounds
```

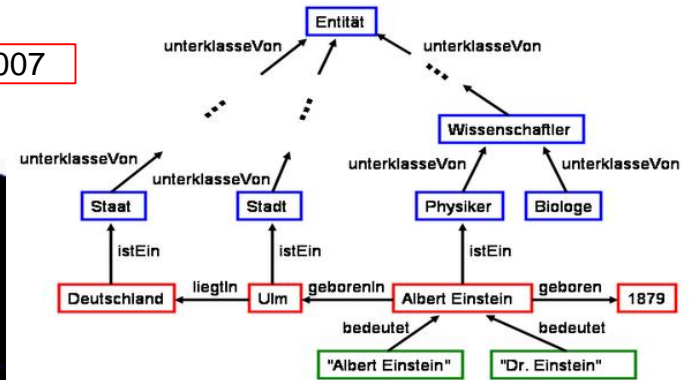
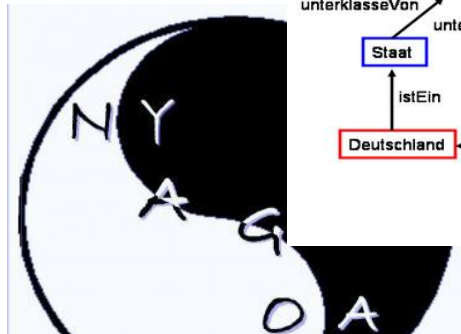
For the First Time there Exist Sizable General Ontologies: DBpedia, YAGO, Freebase, Knowledge Graph



Auer, Bizer, Kobilarov, Lehmann, Cyganiak, Ives: 2007

About / News Datasets

DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link other data sets on the Web to Wikipedia data.



Suchanek, Kasneci, Weikum: 2007

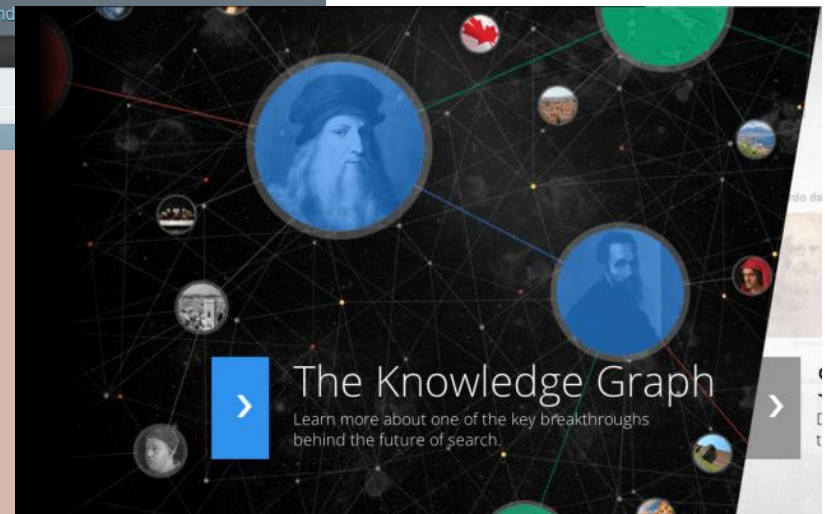


A community-curated database of well-known people, places, and...

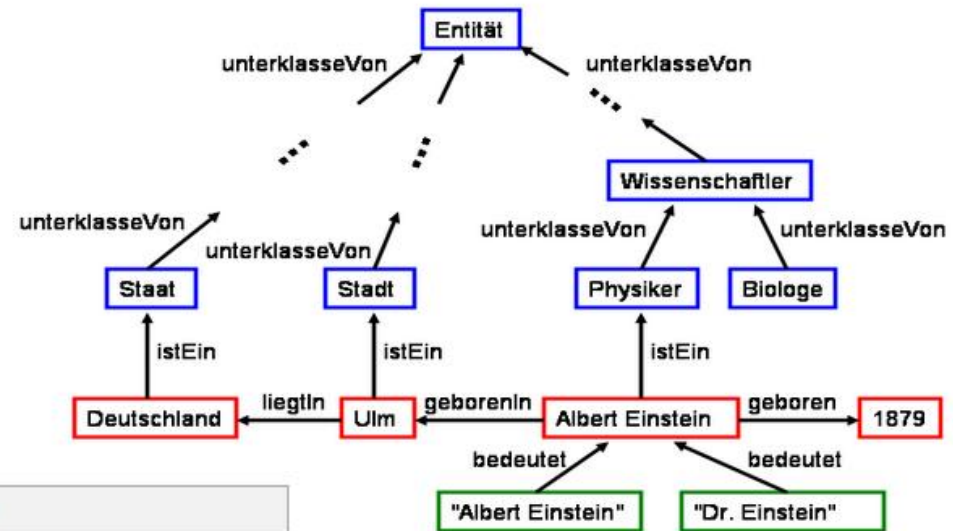
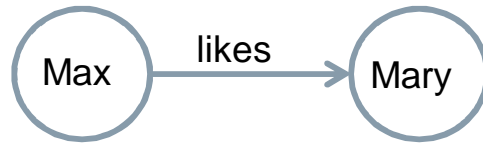
Domain	ID	Topics	Facts
Music	/music	27M	107M
Books	/book	6M	15M
Media	/media_common	5M	16M
People	/people	3M	17M
Film		2M	18M
TV		2M	17M
Location	/location	1M	18M
Business	/business	1M	3M
Fictional Universes	/fictional_universe	923K	1M
Organization	/organization	812K	4M
Biology	/biology	639K	4M
Sports		459K	4M
Awards		340K	5M
Education		244K	3M
Government		148K	921K
Soccer		143K	912K

2,462,493,693 Facts (and counting)

Bollacker, Evans, Paritosh, Sturge, Taylor, 2008



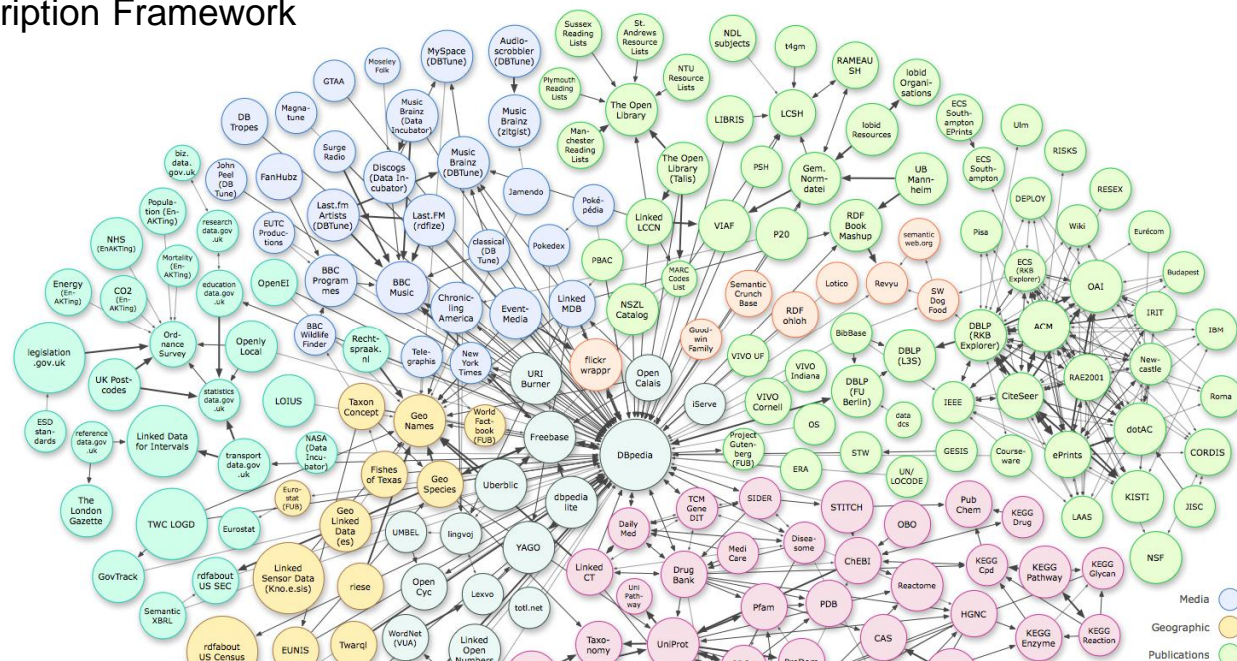
Triple Graphs



In english	The graph
<ul style="list-style-type: none"> • Dog1 is an animal • Cat1 is a cat • Cats are animals • Zoos host animals • Zoo1 hosts the Cat2 	<p style="text-align: center;"> RDF special terms RDFS special terms </p>
RDF/turtle	
<pre> @PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> . @PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> . @PREFIX ex: <http://example.org/> . @PREFIX zoo: <http://example.org/zoo/> . ex:dog1 rdf:type ex:animal . ex:cat1 rdf:type ex:cat . ex:cat rdfs:subClassOf ex:animal . zoo:host rdfs:range ex:animal . ex:zoo1 zoo:host ex:cat2 . </pre>	

Knowledge Bases are Triple Graphs

- Linked Open Data (LOD) and large ontologies like DBpedia, Yago, Knowledge Graph are graph-based knowledge representations using light-weight ontologies, and are **accessible to machine learners**
- They are all triple oriented and more or less follow the RDF standard
 - RDF: Resource Description Framework



Overview

- Why Machine Learning needs Knowledge Graphs
- **Statistical Relational Learning**
- Learning with the YAGO Knowledge Graph
- Towards Relevant Use Cases

Canonical Relational Machine Learning Task

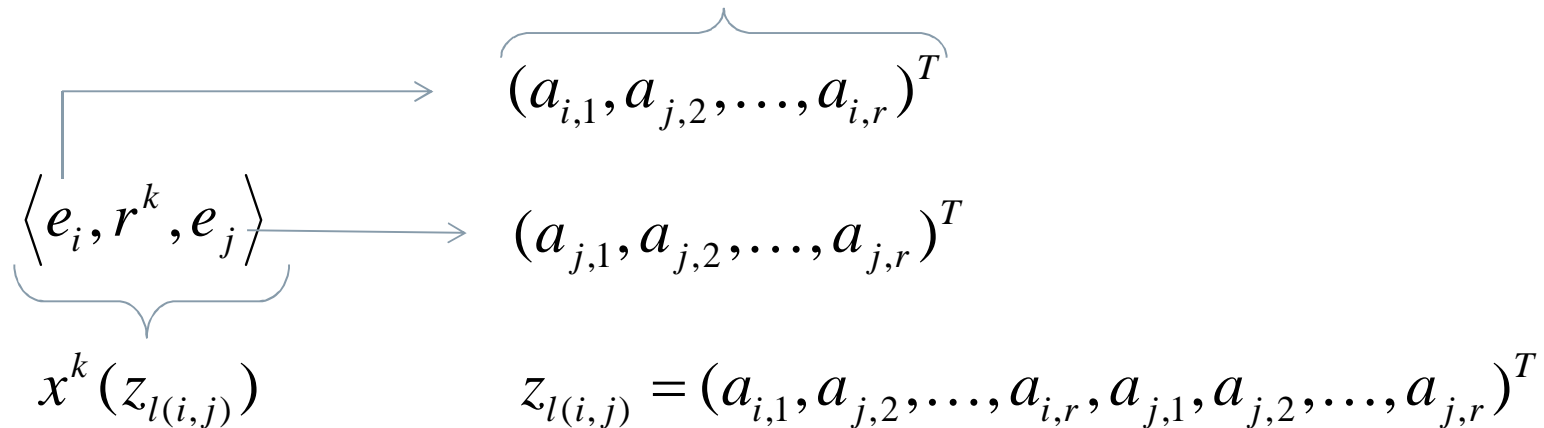
$\langle e_i, r^k, e_j \rangle$ true or false?

$$P(\langle e_i, r^k, e_j \rangle = 1) \leftarrow f^k(z_l)$$

- So, very simple, we build one classifier for each relation type k and we are done
- But what is the input z_l ?

I. Relational Learning with *Known* Features

features (age, sex, *features derived from a neighborhood of the entity in the environment of the RDF-graph*)



$$f^k(z_l) = \sum_{m=1}^M w_m^k b_m(z_l)$$

$$f^k(z_l) = \sum_{n=1}^N v_n^k k(z_l, z_n)$$

Popular in learning from the Semantic Web

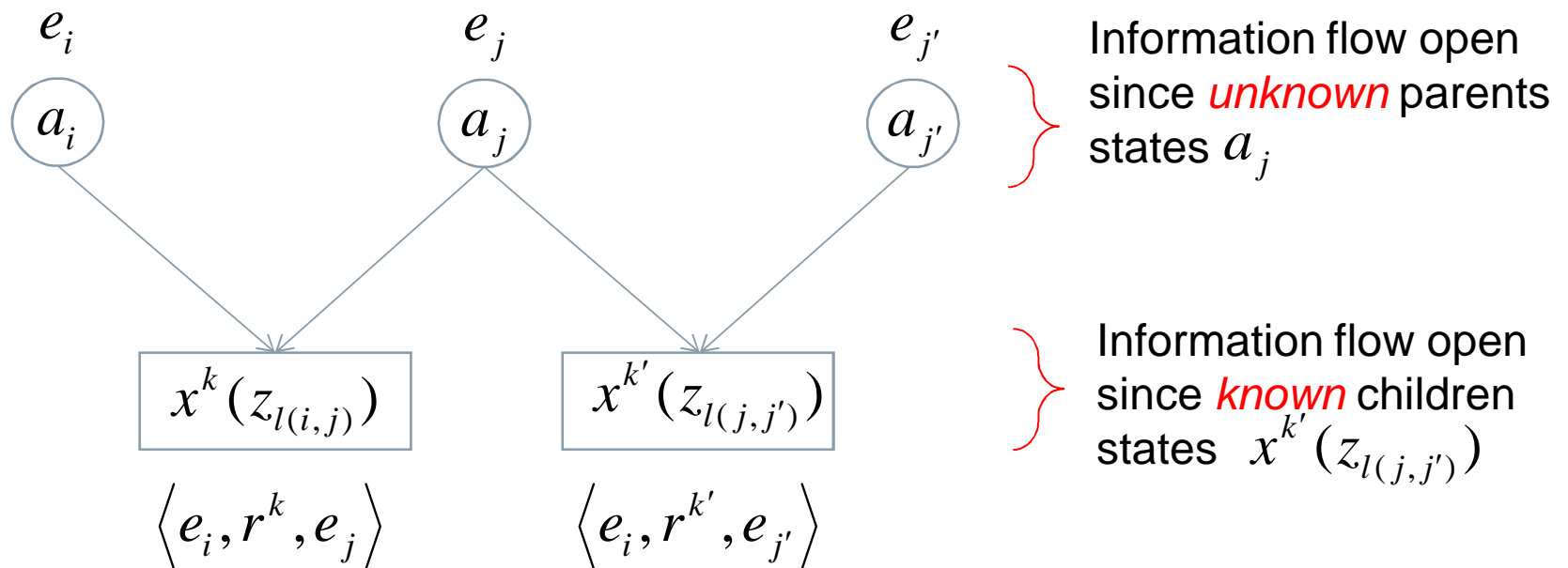
$$f^k(z_l) = NN_{deep}(z_l)$$

II. Relational Learning with *Latent* Features

Same, but features are treated as *latent (unknown) variables*

$$\begin{array}{l}
 \left. \begin{array}{l} \langle e_i, r^k, e_j \rangle \\ x^k(z_{l(i,j)}) \end{array} \right\} \begin{array}{l} \longrightarrow (a_{i,1}, a_{j,2}, \dots, a_{i,r})^T \\ \longrightarrow (a_{j,1}, a_{j,2}, \dots, a_{j,r})^T \end{array} \\
 \\
 z_{l(i,j)} = (a_{i,1}, a_{j,2}, \dots, a_{i,r}, a_{j,1}, a_{j,2}, \dots, a_{j,r})^T \\
 \underbrace{\hspace{15em}}_{\text{unknowns!}} \\
 f^k(z) = \sum_{m=1}^M w_m^k b_m(z_l)
 \end{array}$$

With Latent Features We Get *Collective Learning*




- Information can globally propagate in the network of random variables
- Thus one can learn that: *Jack is rich since the father of his father is rich*

Model with Polynomial Basis Functions

- But what are good basis functions?
- We need to represent the interactions between all feature components
- Binary interactions

$$f^k(z_l) = \sum_{s=1}^r \sum_{t=1}^r w_{s,t}^k b_{s,t}(z_l)$$


$$b_{s,t}(z_l) = a_{i,s} a_{j,t}$$

Mapping to a Tensor Factorization Problem

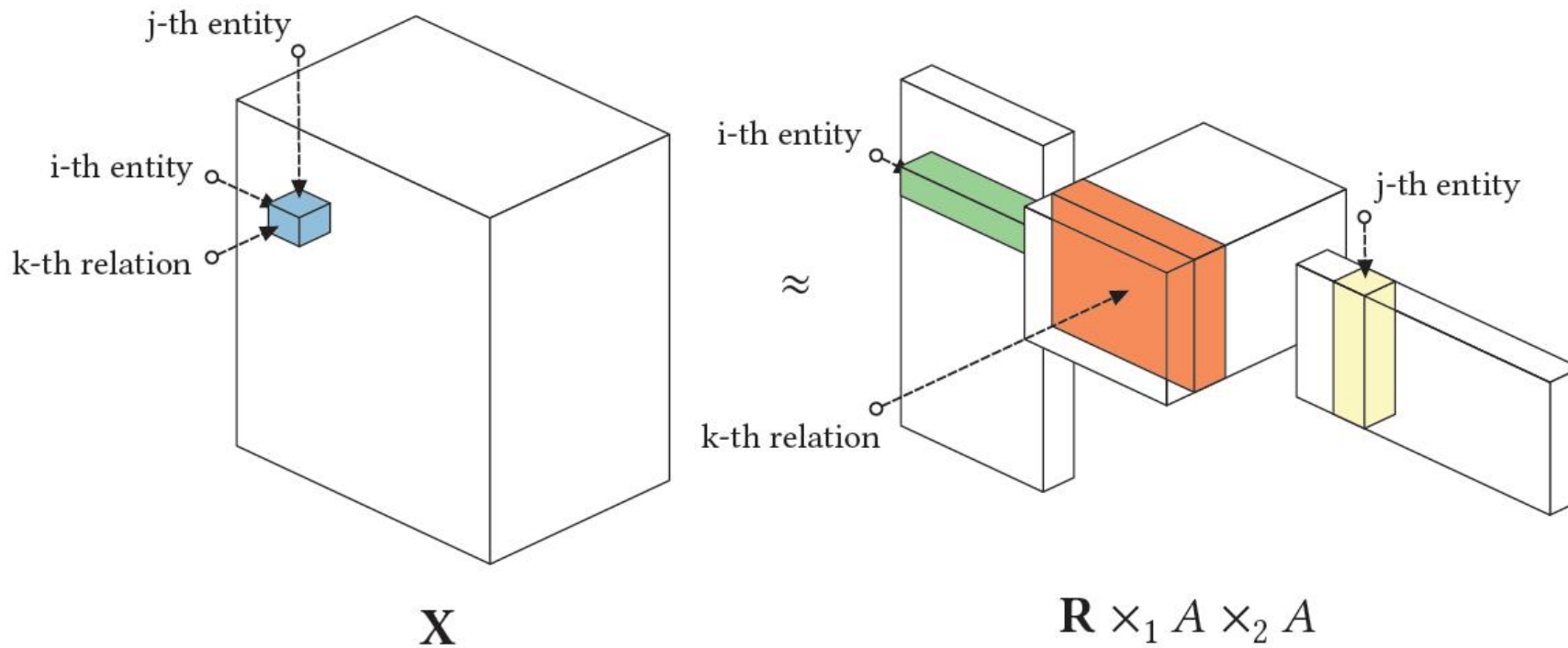
$$f^k(z_l) = \sum_{s=1}^r \sum_{t=1}^r w_{s,t}^k a_{i,s} a_{j,t} = a_i^T R_k a_j \quad (R_k)_{s,t} = w_{s,t}^k$$

- Here, R_k is a $r \times r$ matrix
- We can take the matrices for the different relations R_1, R_2, R_3, \dots on to of each other and obtain the core tensor R
- In tensor notation: We factorize the tensor X

$$X \leftarrow R \times_1 A \times_2 A$$

$$(X)_{i,j,k} = x^k(z_{l(i,j)})$$

RESCAL Factorization



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

Cost Functions

Frobenius norm

$$\arg \min_{A, \mathbf{R}} \|\mathbf{X} - \mathbf{R} \times_1 A \times_2 A\|^2 + \lambda_A \|A\|^2 + \lambda_{\mathbf{R}} \|\mathbf{R}\|^2$$

Probabilistic View

$$P(\mathbf{X} | A, \mathbf{R}) = \prod_{i=1}^n \prod_{j=1}^n \prod_{k=1}^m P(x_{ijk} | \mathbf{a}_i^T R_k \mathbf{a}_j)$$

$$\mathbf{a}_i \sim \mathcal{N}(0, \sigma_A^2 I)$$

$$R_k \sim \mathcal{N}(0, \sigma_R^2 I)$$

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

$$\text{Bernoulli } x_{ijk} \sim \text{Bernoulli}(\mathbf{a}_i^T R_k \mathbf{a}_j)$$

Iterative Update

- Most efficient: Alternating Least Squares (ALS)
 - Can exploit data sparsity
- (stochastic gradient descent, ...)

$$A \leftarrow \left(\sum_{k=1}^m X_k A R_k^T + X_k^T A R_k \right) \left(\sum_{k=1}^m B_k + C_k + \lambda_A I \right)^{-1}$$

$$B_k = R_k A^T A R_k^T, \quad C_k = R_k^T A^T A R_k$$

$$\text{vec}(R_k) \leftarrow \left(Z^T Z + \lambda_R I \right)^{-1} Z^T \text{vec}(X_k)$$

$$Z = A \otimes A$$

RESCAL for Different -arities

Unary Relations

$$P(r_k(e_i)) \leftarrow v_k^T a_i = \sum_{n=1}^r v_{k,n} a_{i,n}$$

Binary Relations

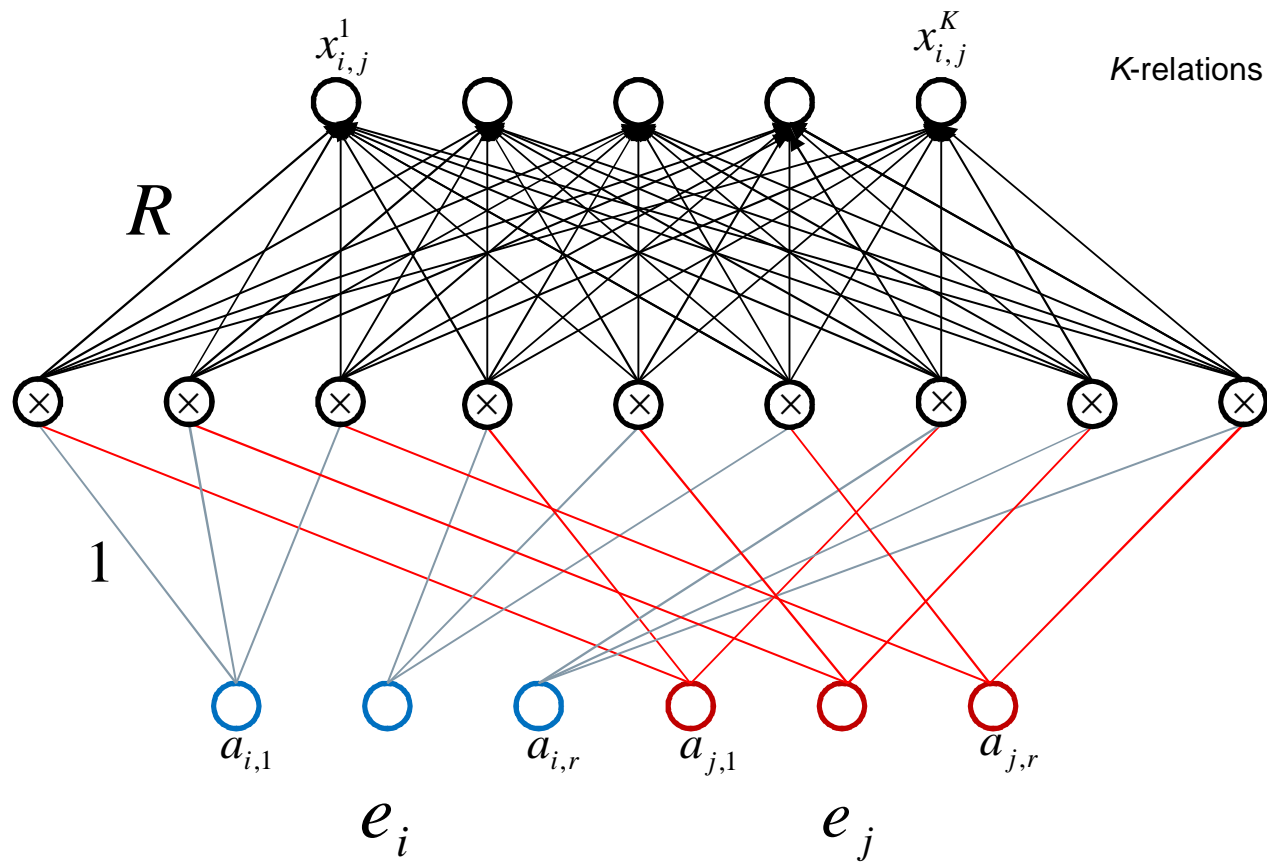
$$P(r_k(e_i, e_j)) \leftarrow a_i^T R_k a_j = \sum_{n_1=1}^r \sum_{n_2=1}^r R_{k,n_1,n_2} a_{i,n_1} a_{j,n_2}$$

Ternary Relations

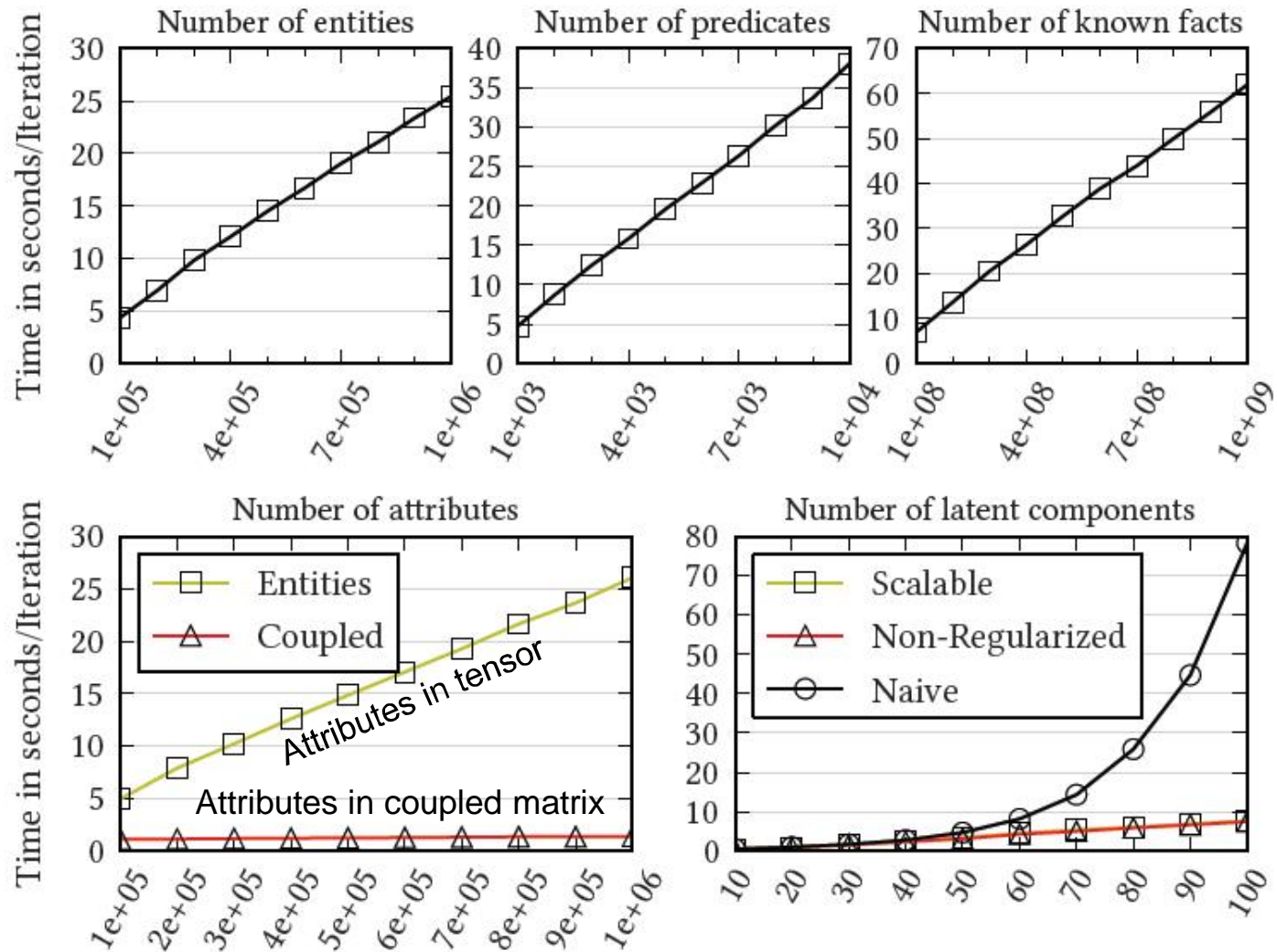
$$P(r_k(e_i, e_j, e_l)) \leftarrow \sum_{n_1=1}^r \sum_{n_2=1}^r \sum_{n_3=1}^r R_{k,n_1,n_2,n_3} a_{i,n_1} a_{j,n_2} a_{l,n_3}$$

- In our applications only unary and binary relations are used
- The latent entity representation (a -vector) for a given entity is identical in all relations and thus information can be shared between all relations, as well!

RESCAL for Binary Relations



Scalability



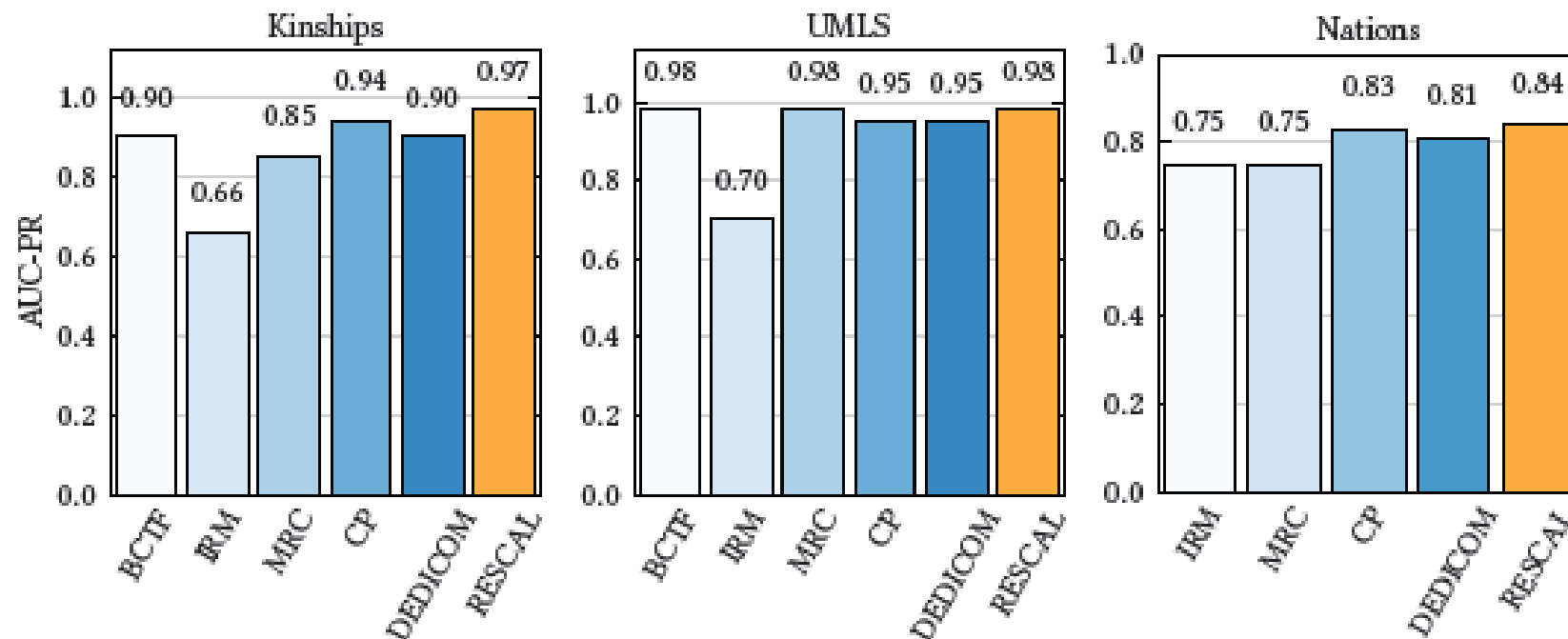
Leading Performance in Link prediction on benchmark data sets

Predicting relationships:
„Max likes Mary“

Kinship: multiple kinship relations between members of the Alyawarra tribe in central Australia (10,790 kinship relationships (facts) between 104 persons over 26 relations)

UMLS: The UMLS data set consists of a small semantic network which is part of the Unified Medical Language System (UMLS) ontology. 6,752 relationships (facts) between 135 concepts over 49 relations

Nations: The Nations data set describes political interactions of countries between 1950 and 1965 . It contains information such as military alliances, trade relationships or whether a country maintains an embassy in a particular country. 2,024 relationships between 14 countries over 56 dyadic relations

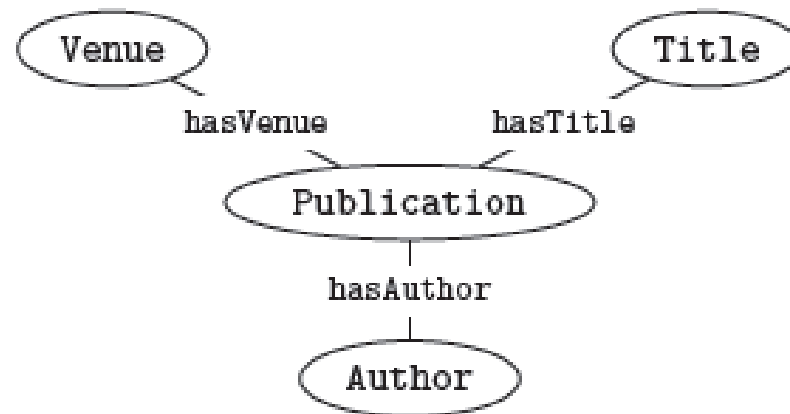


BCTF: Bayesian clustered tensor factorization; MRC: Multi-View Relational Classification

Cora Data: Entity Resolution

Recognizing specific entities:
„This my cat Max!“

- 1295 publication records, where each publication is the subject of a relationship to its first author, a relationship to its title, and a relationship to its publication venue
- Task: identify which authors, entities and venues refer to identical entities

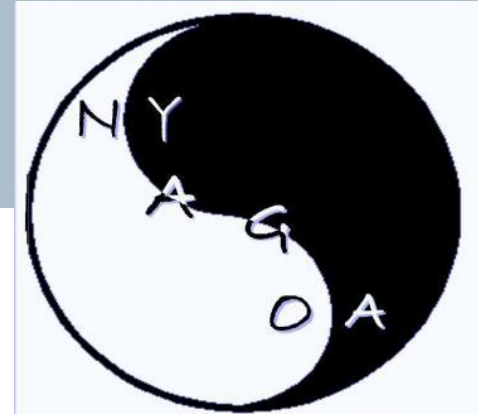


Entity Type	AUC-PR				
	Naive Bayes	MLN (B) (basic rules)	MLN (BCTS) (complex rules)	CP	RESCAL
Publications	0.913	0.915	0.988	0.991	0.991
Authors	0.986	0.987	0.992	0.984	0.997
Venue	0.738	0.736	0.807	0.746	0.810

Overview

- Why Machine Learning needs Knowledge Graphs
- Statistical Relational Learning
- **Learning with the YAGO Knowledge Graph**
- Towards Relevant Use Cases

Yago2 Core Ontology



YAGO2 core ontology

Number of Resources	2.6 million
Number of Classes	340,000
Number of Predicates	87
Number of Known Facts	33 million

Now also DBpedia

The tensor has 10^{14} entries!

Siemens – MPII cooperation

Classification: Type Prediction

Type	Number of entities
wordnet:person	884,261
wordnet:location	429,828
wordnet:movie	62,296

Table 3.9.: Link-prediction experiments on YAGO2.

Predicting concepts:
„This is a cat“

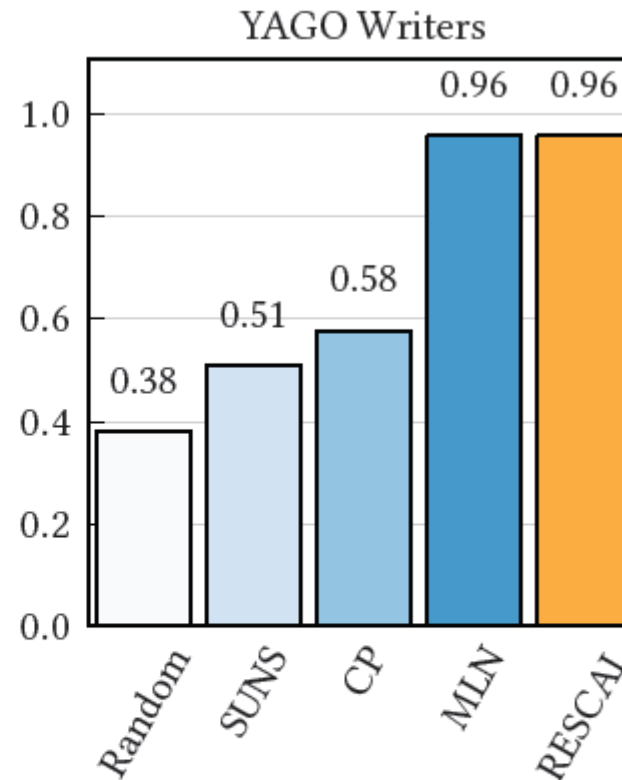
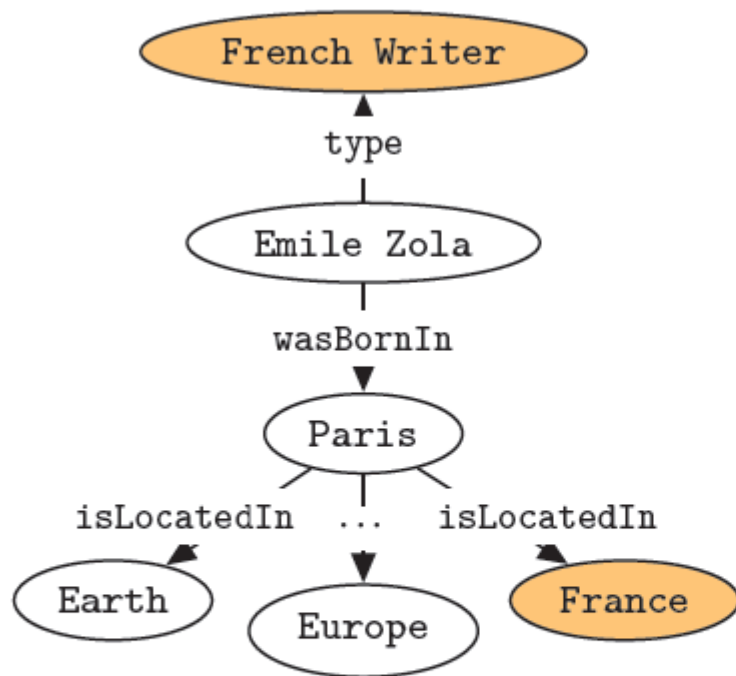
	AUC-PR		
	wordnet:person	wordnet:location	wordnet:movie
Random	0.32	0.18	0.06
Setting a)	0.99	1.0	0.75
Setting b)	0.96	0.98	0.51
With attributes	-	-	0.85

(text attributes)

- a) Only those `rdf:type` triples that include the class C that should be predicted were removed from the test fold. All other type triples, including subclasses of C , are still present in the data.
- b) *All* `rdf:type` triples were deleted in the test fold.

Writer's Nationality: Demonstrating Collective Learning

Predicting concepts/attributes:
„Max is evil“



(a) Collective learning example on YAGO. The objective is to learn the correlation between France and French Writer from examples like Emile Zola.

(b) Results for link prediction on YAGO2 writers data set over ten-fold cross-validation.

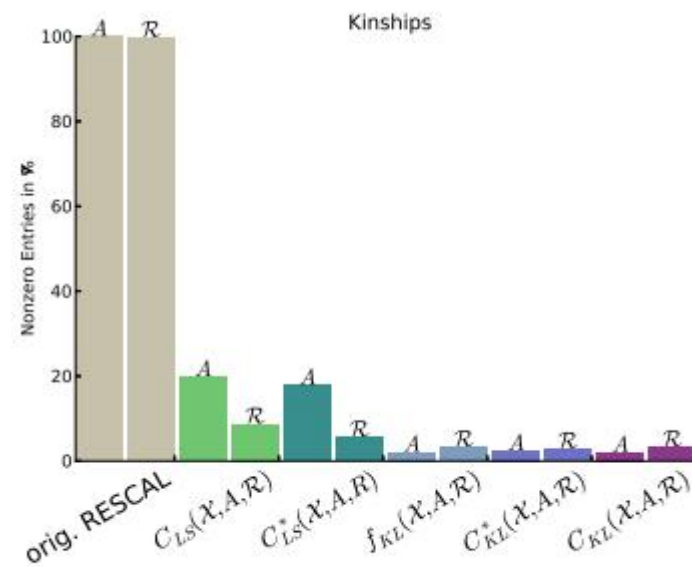
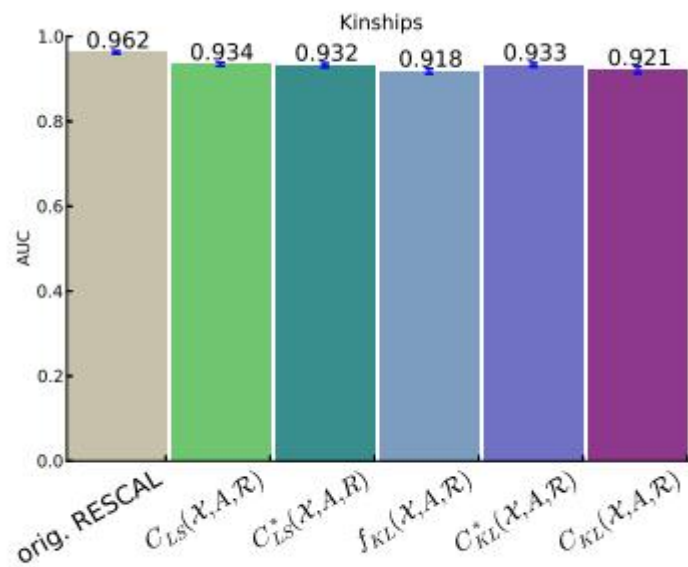
Learning a Taxonomy (-> Ontology)

- IIMB 2010 benchmark provided by the Ontology Alignment Evaluation
- Around 1400 entities of a movie domain
- 5 distinct top-level concepts
- On the top level: every concept is represented by a sufficient number of entities, while e.g. some level 2 movie concepts only include two or three entities and therefore are hard to recognize.

Table 3.10.: F-measure for selected concepts and weighted F-measure for all concepts per subclass-level

Level 1		Level 2		Level 3	
Locations	0.95	City	0.99	Capital	0.99
Films	1.0	Anime	0.67	Director	0.78
Creature	1.0	Character	0.73	Character Creator	0.53
Budget	1.0	Person	1.0	Actor	0.98
Language	1.0	Country	0.80		
All	0.982	All	0.852	All	0.947

Extensions: Nonnegative RESCAL



Kinships

Nonnegative RESCAL (Krompass, Nickel, Tresp)

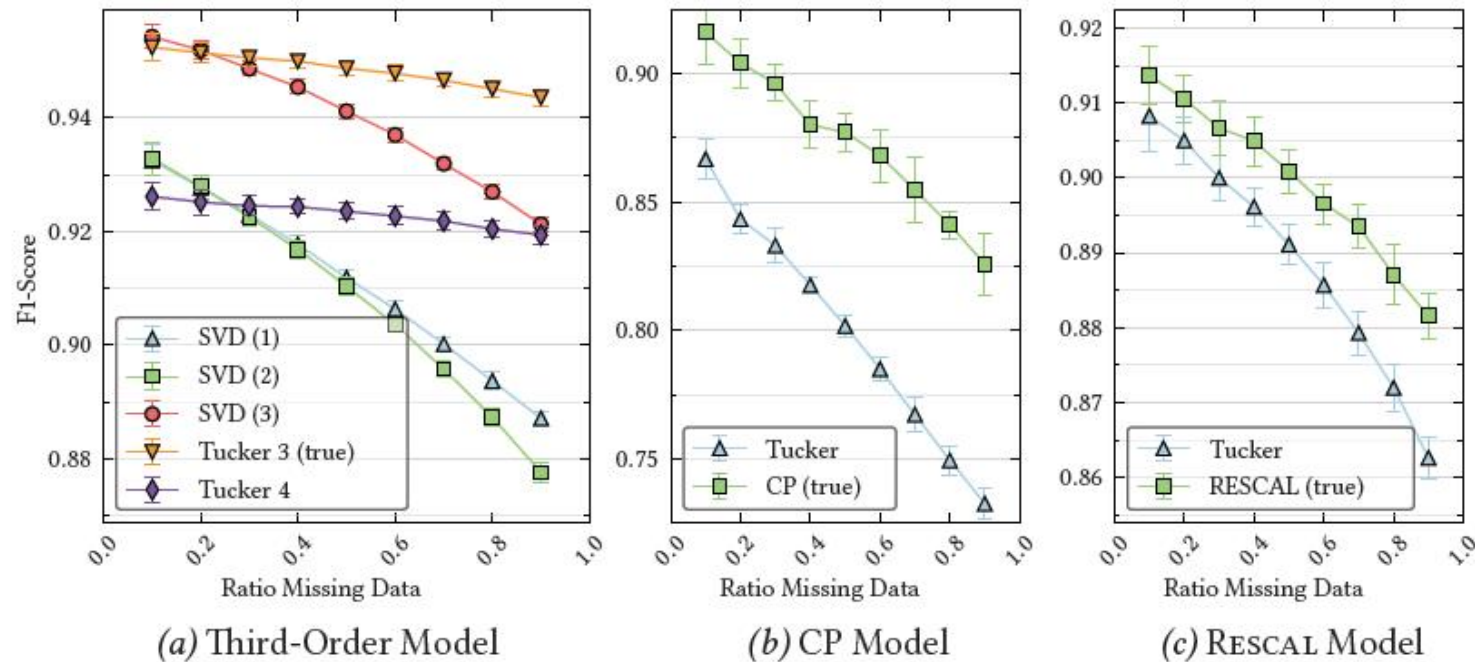
- sparse solutions with clustering properties

Extensions: Proofs and Bounds



- Analysis of generalization bounds when order of the tensor match or do not match
- Matricization results in a loss of generalization performance

Maximilian Nickel and Volker Tresp. *An Analysis of Tensor Models for Learning on Structured Data. Proceedings of the ECML/PKDD, 2013*



Overview

- Why Machine Learning needs Knowledge Graphs
- Statistical Relational Learning
- Learning with the YAGO Knowledge Graph
- **Towards Relevant Use Cases**

Machine Learning with Structured Data and Ontologies

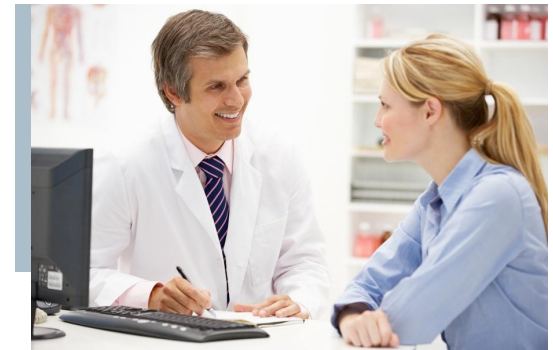
Within the domain:

- Prediction of triples
- Classification (defining type)
- Clustering
- Taxonomy Learning
- Entity Resolution
- Visualization
- Querying
 - Who wants to be Trelenas friends
 - Can be generalized towards more complex probabilistic queries
(*Krompass, Nickel, Tresp, ISWC 2014*)

Outside of the domain (new entities):

- Calculate the latent factors for the new entity
 - Can do all of the tasks above
 - Object recognition becomes entity resolution
- Formulate the new object as a query
 - Object recognition as a query
 - Queries can become complex

Clinical Data Intelligence



Goals

- Personalized medicine: modeling the patient in her/his full complexity -> patient specific recommendations
- Global modeling of the clinical data / clinical decision processes: clinical ontology (concepts and instances)

Use Cases

- All data from all patients
- Breast cancer
- Nephrology
- Data from clinical studies

Challenges

- Ontologies
- Complex relational data (patient in a clinic)
- Representing time; sequential data
- Decision modeling: decision optimization (confounders, causality)
- Including unstructured data (reports, images)
- Including OMICS data

SIEMENS

CHARITÉ
UNIVERSITÄTSMEDIZIN BERLIN

FAU
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Fraunhofer

IIS
Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

averbis
medical language technology

Institut
für
FrauenGesundheit
(IFG®)

Predicting Diagnoses and Procedures

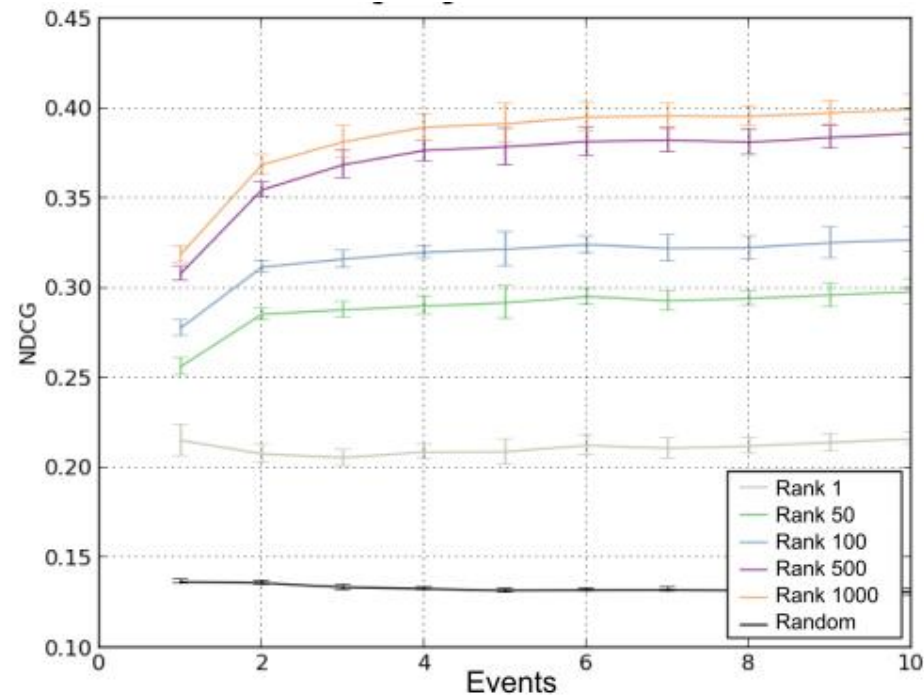


Figure 1: Data from 10000 patients were used. We considered 2331 possible diagnoses, 1634 possible procedures, 2721 possible lab results, 209 possible therapies and 281 general patient data. In total the data contained 5.9 million facts. We predicted the next decision (diagnosis, procedure) as a function of the information available for each patient. Plotted is the NDCG score (a popular score for evaluating ranking results [11]) as a function of the information available for each patient (a large number is desirable). An event corresponds to an instance in time where patient data is recorded. With increasing information, the prediction improves. We see plots for different approximation ranks: the highest rank gives best scores which reflects the high degree of data complexity.

Machine Learning with Images and Ontologies



Linking textual descriptions in radiology reports to medical images

References

RESCAL

- 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
- 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
- Maximilian Nickel and Volker Tresp. An Analysis of Tensor Models for Learning on Structured Data. *Proceedings of the ECML/PKDD*, 2013
- Denis Krompaß, Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Non-Negative Tensor Factorization with RESCAL. *ECML/PKDD 2013 Workshop on Tensor Methods for Machine Learning*, 2013
- Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Learning from Latent and Observable Patterns in Multi-Relational Data. In *Advances in Neural Information Processing Systems (NIPS*2014)*, 2014
- Denis Krompaß, Maximilian Nickel, and Volker Tresp. Querying Factorized Probabilistic Triple Databases. *Proceedings of the ISWC*, 2014
- Denis Krompaß, Maximilian Nickel, and Volker Tresp. Factorizing Large Heterogeneous Multi-Relational-Data. *International Conference on Data Science and Advanced Analytics (DSAA'2014)*, 2014
- Denis Krompaß, Xueyan Jiang, Maximilian Nickel, and Volker Tresp. Probabilistic Latent-Factor Database Models. *Proceedings of the ECML workshop on Linked Data for Knowledge Discovery*, 2014

Related Work

Extensions from other groups

- R. Jenatton, N. Le Roux, A. Bordes, G. Obozinski (contributed equally). A latent factor model for highly multi-relational data. *Advances in Neural Information Processing Systems, NIPS, 2012*
- Richard Socher, Danqi Chen, Christopher D. Manning, Andrew Y. Ng. Reasoning With Neural Tensor Networks for Knowledge Base Completion, *NIPS, 2013*

SUNS (First application of factorization approaches to relational Semantic Web domains)

- Volker Tresp, Yi Huang, Markus Bundschuh, and Achim Rettinger. Materializing and querying learned knowledge. *IRMLeS, 2009*

Triplerank (Application of PARAFAC for ranking; no collective learning)

- T. Franz, A. Schultz, S. Sizov, and S. Staab. "Triplerank: Ranking semantic web data by tensor decomposition". *ISWC, 2009*

Factorization Machines

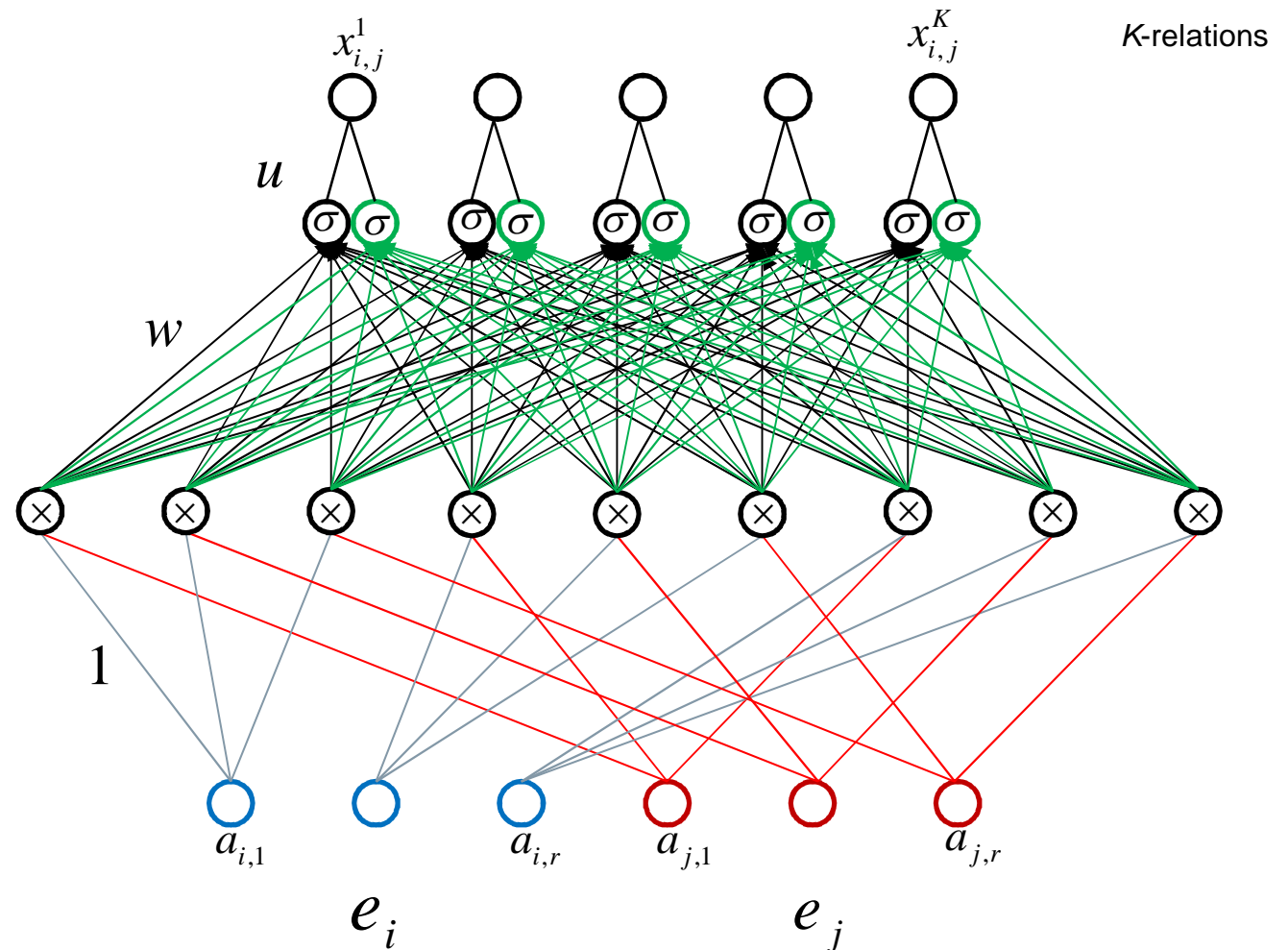
- S. Rendle et al.: Different factorization approaches for preference prediction and relational learning (2009 and later)

Knowledge Vault (Google Team)

- X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, **K. Murphy**, T. Strohmann, S. Sun, ND W. Zhang. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion *KDD 2014*.

Neural Tensor Model (Socher et al.)

- *Contrastive max-margin objective functions (similar to Collobert during pre-training)*
- *Use Batch SGD*
- *This means that per epoch, one does not adapt wrt all $M \times M \times K$ triples but only wrt $2 \times T \times C$ triples*
- *M : number of entities*
- *K : number of relation types*
- *T : number of true triples*
- *C : tuning parameter; often 10*

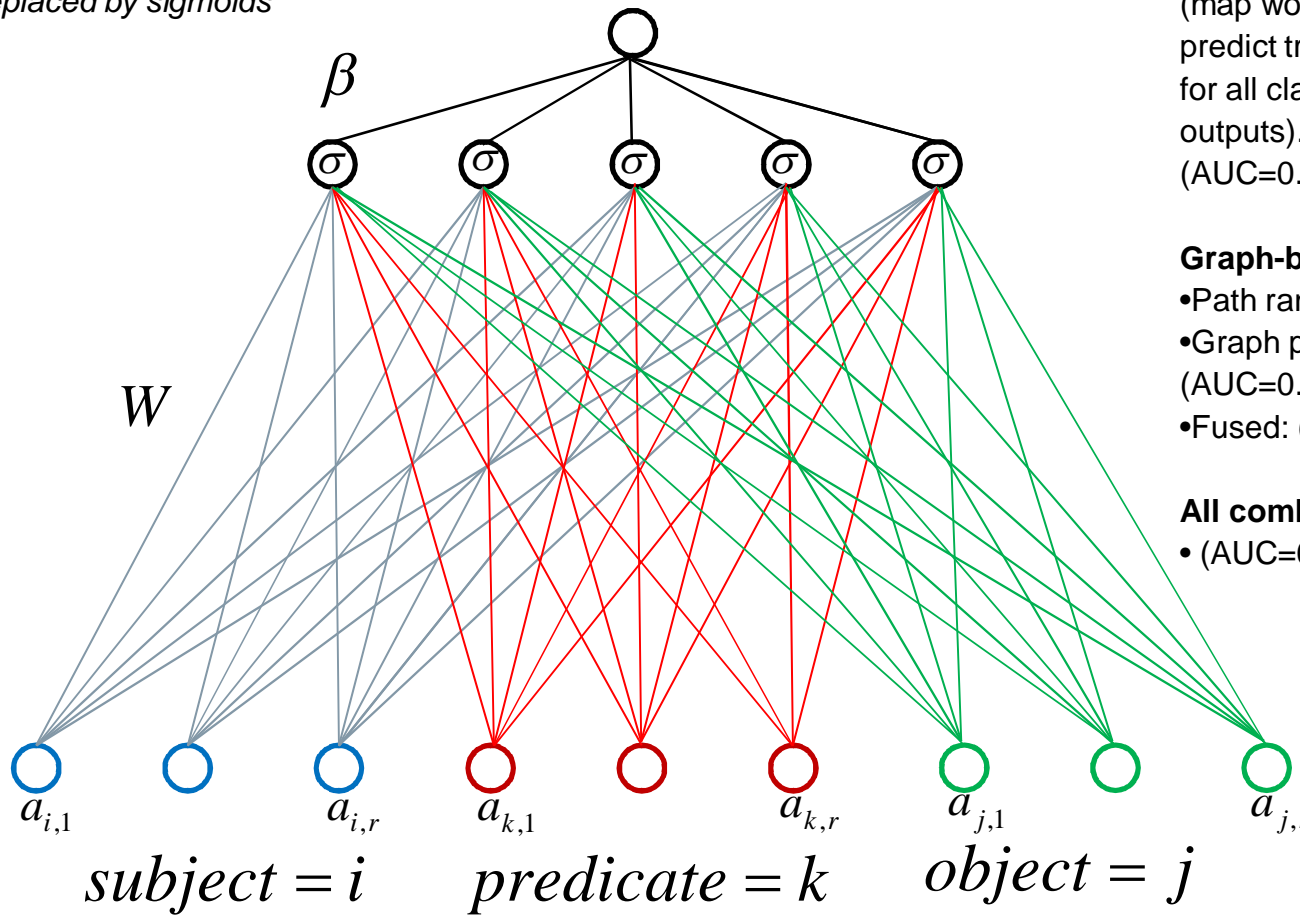


Google Vault (Murphy et al.)

RESCAL3

where the polynomials are replaced by sigmoids

$$P(x_{i,j}^k = 1 | A, W, \beta)$$



Fact extraction: NER, POS, entity linkage (map words to entities); complex features to predict triple from text; combination scheme for all classifiers (Platt scaling to normalize outputs). Output: probability for a triple (AUC=0.927)

Graph-based prior:

- Path ranking algorithm (AUC=0.884)
- Graph prior (“RESCAL” variant) (AUC=0.882)
- Fused: (AUC=0.911)

All combined:

- (AUC=0.947)

Conclusions

- **Knowledge Graphs**
 - First time: large general ontologies available
 - Useful for solving machine learning tasks
- **Relational Machine Learning with RESCAL**
 - Scalable relational learning with very competitive performance
 - Collective Learning
 - We are working on many improvements/extensions
- **RESCAL Learning with the YAGO Knowledge Graph**
 - Experimental results in a number of relational learning tasks
- **Towards Relevant Use Cases**
 - Text understanding
 - Image understanding
 - Clinical data