

Mining networked Data

Nicola Di Mauro & Donato Malerba

Department for Computer Science, LACAM Laboratory
University of Bari "Aldo Moro"

IEEE Symposium on Computational Intelligence and Data Mining
April 11-15, 2011 - Paris, France

Outline

Introduction

- Why SRL for networked data

Brief Intro to Probabilistic Graphical Models

- Bayesian Networks

- Markov Networks

Intro to SRL

Logic Programs with Annotated Disjunctions

- Intro to LPADs

- Information Bottleneck for LPADs

Markov Logic Networks

Sequence Learning

- SRL approaches to Sequence Learning

- Lynx

Conclusions and Open Issues

Statistical Relational Learning

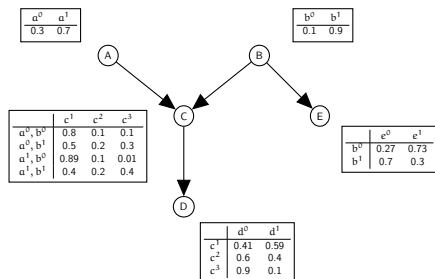
- ▶ Statistical Relational Learning (SRL) or Probabilistic Inductive Logic Programming (PILP)
(Getoor and Taskar, 2007) (De Raedt et al., 2008)
- ▶ Integration of probabilistic reasoning with logical or relational representations and machine learning
 - ▶ first-order logic provides formal tools to handle the complexity of the real world
 - ▶ probability mathematically deals with uncertainty

Trying to put together

- ▶ Probabilistic inference and learning
 - ▶ traditionally assume independent and identically distributed (i.i.d.) objects sampled from a single relation
- ▶ Logical valid inference and Inductive logic programming
 - ▶ consider objects from multiple relations, but usually no uncertainty in the data

Bayesian Network

Directed Graphical Models



- ▶ nodes: random variables
- ▶ edges: direct probabilistic influence

Local probability model

- ▶ the network structure encodes independence assumptions
 - ▶ each variable in the model is associated with a CPD
- ▶ *local independencies*: $(X_i \perp \text{NonDescendants}_{X_i} | \text{Pa}_{X_i}^G)$
 - ▶ X_i is conditionally independent of $\text{NonDescendants}_{X_i}$ given its parents $\text{Pa}_{X_i}^G$

Bayesian Networks

Inference and learning

Chain rule

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{Pa}_{X_i}^g)$$

- ▶ individual factors $\text{Pa}_{X_i}^g$ are the CPDs

Inference:

- ▶ the full joint distribution specifies answer to any query
- ▶ Inference (exact/approximate) algorithms
 - ▶ variable elimination, junction tree, loopy belief propagation, likelihood weighting, MCMC, variational methods, ...

Learning:

- ▶ BNs (parameters/structure) can be learned from empirical data
- ▶ Parameters w/ numerical optimization
 - ▶ counting (complete data), EM or gradient descent, IB, ...
- ▶ structure and parameters w/ combinatorial search
 - ▶ structure search (complete data), Structural EM, IB, ...

Learning BNs

Parameters Estimation and Structure Learning

a) The BN structure is assumed to be known

- ▶ Find the maximum likelihood estimates (MLEs) of the parameters of each CPD given a training set $\mathcal{D} = \{D_1, \dots, D_m\}$

$$L(\theta : \mathcal{D}) = \prod_i P(D_i : \theta | \mathcal{G}) = \prod_j \prod_i P(X_j | \text{Pa}_{X_j}^{\mathcal{G}}, D_i) = \prod_j \prod_i \hat{\theta}_{X_j | \text{Pa}_{X_j}^{\mathcal{G}}}$$

- ▶ Complete data: we observe all variables
 - ▶ computing $\hat{\theta}_{X_j | \text{Pa}_{X_j}^{\mathcal{G}}}$ is simple: counting
- ▶ Missing values
 - ▶ Expectation Maximization
 - ▶ start with arbitrary parameters values and then iterates the steps
 - ▶ **E**: compute probability of unobserved variables
 - ▶ **M**: calculate the new parameters knowing all data

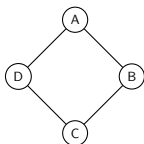
b) The BN structure is unknown

- ▶ Start with an empty network and add/remove/change edges guided by a scoring function
- ▶ Structural EM

Markov Networks

Undirected Graphical Models

- ▶ A BN requires that we ascribe a directionality to each influence



$\phi_1(A, B)$	
a^0, b^0	30
a^0, b^1	5
a^1, b^0	1
a^1, b^1	10

$\phi_2(B, C)$	
b^0, c^0	100
b^0, c^1	1
b^1, c^0	1
b^1, c^1	100

$\phi_3(C, D)$	
c^0, d^0	1
c^0, d^1	100
c^1, d^0	100
c^1, d^1	1

$\phi_4(D, A)$	
d^0, a^0	100
d^0, a^1	1
d^1, a^0	1
d^1, a^1	100

The MN structure encodes independence assumptions

- ▶ **nodes**: random variables; **edges**: direct probabilistic influence

Joint distribution

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}})$$

- ▶ ϕ_k potential functions for each clique
- ▶ $x_{\{k\}}$ is the state of the kth clique
- ▶ $Z = \sum_{x \in X} \prod_k \phi_k(x_{\{k\}})$

partition function

Markov Networks: Inference

log-linear model

- ▶ Compute marginals and conditionals of

$$P(X = \mathbf{x}) = \frac{1}{Z} \exp \left(\sum_i w_i f_i(\mathbf{x}) \right)$$

- ▶ Gibbs sampling: conditioning on Markov blanket
- ▶ MCMC
- ▶ Belief propagation
- ▶ Variational approximation
- ▶ Exact methods

MAP inference

$$\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$$

- ▶ simulated annealing, graph cuts, belief propagation, ...

Markov Networks: Learning

Generative weight learning

- ▶ Maximize likelihood $\frac{\partial}{\partial w_i} \log P_w(x) = n_i(x) - E_w[n_i(x)]$
- ▶ Pseudo-Likelihood $PL(x) = \prod_i P(x_i | \text{neighbours}(x_i))$
- ▶ Iterative scaling

Discriminative weight learning

- ▶ Maximize conditional likelihood
 $\frac{\partial}{\partial w_i} \log P_w(y|x) = n_i(x, y) - E_w[n_i(x, y)]$
- ▶ Max Margin

Structure Learning

- ▶ Start with atomic features
- ▶ Greedily conjoin features to improve score

SRL tasks

Object Classification: predicting the label of an object within the network using its observed/unobserved labels/attributes of the objects in its neighbourhood

- ▶ Collective classification (Jensen et al. 2004)
 - ▶ labelling of an object should depend on the labels of its neighbours
 - ▶ Chakrabarti et al. 1998, Taskar et al. 2001, Neville et al. 2003, Jensen et al. 2004

Entity Resolution: determining which reference in the data refer to the same underlying real-world entity

- ▶ Relational Probability Model (Pasula et al. 2003)
- ▶ Relational Markov Network (Singla and Domingos 2005)

Link Prediction

- ▶ identifying when two entities may be connected or more importantly whether they may be connected in the future
- ▶ see Getoor and Diehl, 2005 for a survey

SRL tasks /2

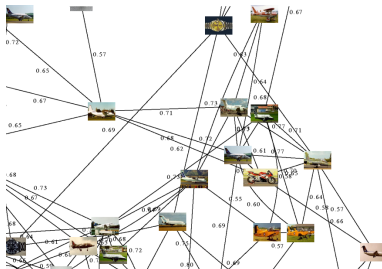
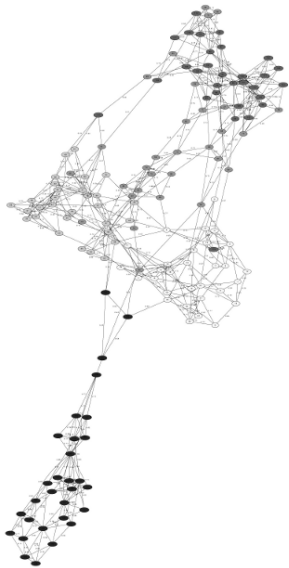
Community Detection: identifying communities (groups of entities) by studying the network structure and topology

Information Diffusion and Product recommendation: understanding the mechanism governing how information diffuses through the network

- ▶ Models of network diffusion may be used to study, for instance, product recommendation systems and viral marketing
(Richardson and Domingos, 2002), (Domingos, 2005),
(Leskovec et al., 2006), (Sharara et al., 2011)
- ▶ collaborative recommendation
 - ▶ Infinite Hidden Relational Model (Xu et al. 2008, 2010)
 - ▶ (Fouss et al. 2007)
- ▶ context sensitive trust
 - ▶ Infinite Hidden Relational Trust Model (Rettinger et al. 2011)

Inference over Image Networks

A ProbLog application to classification



- ▶ Improve k-NN classification with ProbLog
- ▶ nodes: images from Caltech 101
- ▶ edges: similarity between images

Logic

- ▶ Syntax
 - ▶ constants, variables, functions, predicates
nico, X, motherOf(X,Y), friends(X,Y)
 - ▶ A **literal** is a predicate or its negation
 - ▶ A **clause** is a disjunction of literals
 - ▶ A **grounding** replaces all variables by constants
friends(nico,donato)
 - ▶ a **world** (model or interpretation) is an assignment of truth values to all ground predicates
- ▶ Entailment or logical consequences, Herbrand models and proofs

Logic: adding probabilities

- ▶ in logic programming
 - ▶ `likes(A,B) :- friendOf(A,C), likes(C,B).`
- ▶ with categorical inference
 - ▶ if A is friend of C and C likes the book B then A likes the book B
- ▶ adding probability
 - ▶ `likes(A,B):0.6 :- friendOf(A,C), likes(C,B).`
 - ▶ if A is friend of C and C likes the book B then A likes the book B with probability 0.6
- ▶ adding rules
 - ▶ `likes(A,B):0.6 :- friendOf(A,C), likes(C,B).`
 - ▶ `likes(A,B):0.8 :- computerScientist(A), authorOf(C,B), computerScientist(C).`
 - ▶ we need combining rules

Knowledge Based Model Construction

- ▶ Based on some form of probabilistic Horn clauses

$$0.2 : p(X, Y) \leftarrow q(X, Y)$$

- ▶ In KBMC the probabilistic rules are interpreted as rules for the construction of BNs over ground atoms
 - ▶ Nodes correspond to ground atoms
 - ▶ Edges
 - ▶ the atom in the head of the clause is a child node
 - ▶ atoms in the body of the clause are parent nodes
 - ▶ Many combining functions for clauses with the same head
- ▶ Inference corresponds to BN inference after a partial grounding
- ▶ Learning obtained with Inductive Logic Programming and EM

Stochastic Logic Programs

- ▶ A generalization of stochastic grammars
- ▶ A SLP consists of a set of annotated Horn clauses

$$p : C \equiv p : A \leftarrow B_1, \dots, B_m$$

- ▶ $p \in [0, 1]$, and the sum of probabilities of clauses with same head is 1

0.5 : coin(0) \leftarrow

0.5 : coin(1) \leftarrow

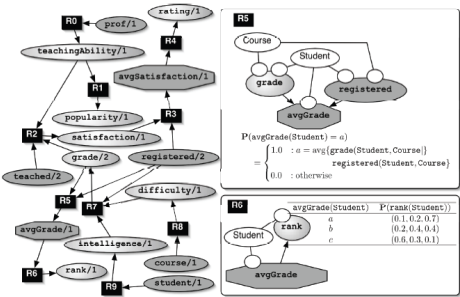
- ▶ Inference is calculated computing all proofs
- ▶ Parameter estimation: Failure Adjusted Maximization

Bayesian Logic

- ▶ Bayesian Logic Programs unify Bayesian networks with logic programming
- ▶ Bayesian clauses

$$bt(X) | mc(X), pc(X).$$

- ▶ Learning parameters with EM and learning the structure with ILP



Distribution Semantics

It is shared by many languages: ICL, PRISM, LPADs, ProbLog

- ▶ A program defines a probability distribution over normal logic programs called **worlds**
- ▶ The distribution is extended to queries
 - ▶ the probability of a query is obtained by marginalizing the joint distribution of the query and the programs

Let $P(W)$ be the distribution over worlds

- ▶ the probability of a query Q given a world w is $P(Q|w) = 1$ if $w \models Q$ and 0 otherwise

$$P(Q) = \sum_{w \in W} P(Q, w) = \sum_{w \in W} P(Q|w)P(w) = \sum_{w \in W: w \models Q} P(w)$$

LPAD

A *Logic Program with Annotated Disjunctions* (LPAD) L consists of a finite set of formulas of the form

$$(H_1 : \theta_1) \vee (H_2 : \theta_2) \vee \dots \vee (H_n : \theta_n) \leftarrow B_1, B_2, \dots, B_m$$

Annotated disjunctive clauses

- ▶ H_i logical atoms, B_i logical literals
- ▶ $\theta_i \in [0, 1]$ real numbers such that $\sum_{i=1}^n \theta_i \leq 1$
- ▶ The head of the clause implicitly contains an extra dummy atom `none` whose annotation is $1 - \sum_i \theta_i$
- ▶ $\text{head}(r) = \{(H_i : \theta_i) \mid 1 \leq i \leq n\}$
- ▶ $\text{body}(r) = \{B_i \mid 1 \leq i \leq m\}$
- ▶ P^g will denote the set of all ground LPADs

Semantics of LPAD

- ▶ Let $P \in P^g$. An *admissible probability distribution* π on \mathcal{J}_P is a mapping from \mathcal{J}_P to real numbers in $[0, 1]$, such that $\sum_{I \in \mathcal{J}_P} \pi(I) = 1$

Worlds in LPADs

A world is identified by means of a selection function

The *probability of a world* L_σ is the product of the probabilities of the individual choices made by the corresponding selection

$$P(L_\sigma) = \prod_{c \in g(L)} \sigma_{\text{prob}}(c)$$

The probability of a query Q is then given by

$$P(Q) = \sum_{L_\sigma \in W} P(Q, w)$$

Example

earthquake(X,strong):0.3 \vee earthquake(X,moderate):0.5 \leftarrow
 fault_rupture(X).

earthquake(X,strong):0.2 \vee earthquake(X,moderate):0.6 \leftarrow
 volcanic_eruption(X).

fault_rupture(stromboli).

volcanic_eruption(stromboli).

volcanic_eruption(eyjafjallajkull).

LPAD: Inference and Learning

Inference

- ▶ cplint: SLDNF resolution and BDD (Riguzzi 07)
- ▶ Contextual Variable Elimination: transforming CP-theory to BN (Meert et al. 09)
- ▶ SLGAD: SLG resolution for normal logic programs (Riguzzi 10)
- ▶ PITA: Tabling and Answer subsumption (Riguzzi & Swift 10)

Learning Parameters

- ▶ EM: reducing LPADs to BNs (Blockeel & Meert 07)

Learning Structure

- ▶ Structural EM: reducing LPADs to BNs (Blockeel & Meert 07, Meert et al. 08)
- ▶ ALLPAD: constraint optimization (Riguzzi 08)

Information Bottleneck

- ▶ IB developed in clustering (Tishby et al., 1999)
 - ▶ Given two variables X and Y and their joint distribution $Q(X, Y)$, group values of Y so that as much information as possible is preserved about X
 - ▶ *Y words appearing in a set of documents and X the documents' topics, we want to cluster words in a way that is most relevant to the documents' topics*
- ▶ The *information* that Y contains about X (and vice versa) is measured in terms of the *mutual information*

$$I_Q(X; Y) \triangleq \sum_{x, y} Q(x, y) \log \frac{Q(x, y)}{Q(x)Q(y)}$$

Information Bottleneck /2

- ▶ A bottleneck variable T whose values identify the various clusters is introduced
 - ▶ $Q(T|Y)$ degree of membership of the values of Y to the clusters
 - ▶ T must compress Y while capturing as much as possible the information about X
- ▶ **Clustering**: finding the parameters of the Q distribution such that the function

$$\mathcal{L}[Q] = \mathbf{I}_Q(Y; T) - \beta \mathbf{I}_Q(T; X)$$

is minimized

- ▶ β determines the trade-off between information compression and preservation

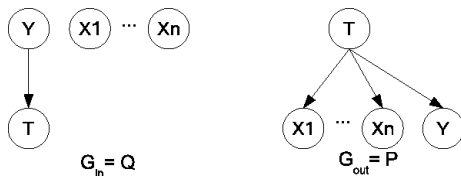
Information Bottleneck for Learning Bayesian Networks

- ▶ Learning BNs with hidden variables (Elidan & Friedman, 2005)
 - ▶ the hidden variables are treated as the bottleneck variable
- ▶ Data $\mathcal{D} = \{\mathbf{x}[1], \dots, \mathbf{x}[M]\}$ over the observed variables \mathbf{X}
- ▶ **Goal:** finding a generative model P over the observed variables \mathbf{X} and the hidden variable T that describes \mathcal{D}
 - ▶ Y represents the instance identity (domain $\{1, \dots, M\}$)
 - ▶ find the parameters (and possibly the structure) of P so that T explains the observed data

Information Bottleneck for Learning BNs

The networks \mathcal{G}_{in} and \mathcal{G}_{out}

- ▶ Two networks
 - ▶ \mathcal{G}_{in} represents the Q distribution
 - ▶ \mathcal{G}_{out} represents the P distribution



- ▶ In the general case, a vector \mathbf{T} of hidden variables.
- ▶ Any distribution for \mathcal{G}_{in} and \mathcal{G}_{out} can be chosen, provided that
 - ▶ \mathbf{T} is independent of \mathbf{X} in \mathcal{G}_{in} given Y
 - ▶ Y is a leaf in \mathcal{G}_{out} with \mathbf{T} as its only parents

Information Bottleneck EM Algorithm

- ▶ A factorized form for $Q(\mathbf{T}|Y)$ can be used
- ▶ For example we can use a naive Bayes assumption

$$Q(\mathbf{T}|Y) = \prod_i Q(T_i|Y)$$

- ▶ Different factorizations correspond to different choices for \mathcal{G}_{in}
- ▶ In this case, the objective function takes the following form

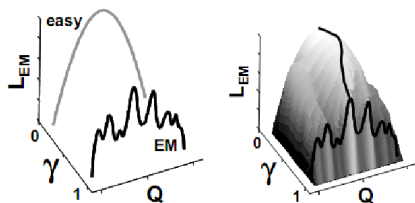
$$\mathcal{L}_{EM}^+ = \sum_i \mathbf{I}_Q(T_i; Y) - \gamma \left(\mathbf{E}_Q[\log P(\mathbf{X}, \mathbf{T})] - \sum_i \mathbf{E}_Q[\log Q(T_i)] \right)$$

Information Bottleneck EM Algorithm /2

- ▶ The Information Bottleneck EM algorithm (IB-EM) consists of the repetition of the following two steps:
 - ▶ **E-step:** minimize \mathcal{L}_{EM}^+ by varying $Q(\mathbf{T}|\mathbf{Y})$ while holding P fixed
 - ▶ **M-step:** minimize \mathcal{L}_{EM}^+ by varying P while holding Q fixed
- ▶ γ balances between compression of the data and the fit of parameters to \mathcal{G}_{out}
 - ▶ for $\gamma = 1$, \mathcal{L}_{EM}^+ is equivalent to the objective function of the EM algorithm
 - ▶ for $\gamma = 0$, \mathcal{L}_{EM}^+ is easy to solve

Information Bottleneck EM Algorithm /3

- ▶ The IB-EM can bypass local maxima of EM by varying γ in L_{EM}^+ using a deterministic annealing strategy:
 - ▶ γ is initially set to 0
 - ▶ γ is gradually incremented towards higher values, tracking the solution through various stages hopefully
- ▶ Aim: follow a smooth path from the trivial solution at $\gamma = 0$ to a good solution at $\gamma = 1$.



(Elidan & Friedman, 2005)

Relational Information Bottleneck (RIB)

- ▶ Application of IB to Statistical Relational Languages that can be converted to Bayesian networks
- ▶ RIB allows to learn parameters of SRL languages
 - ▶ that can be converted to Bayesian networks
 - ▶ that contain hidden variables
- ▶ In these networks, some parameters are tied
 - ▶ the formulas for updating parameters must take that into account
 - ▶ the counts must take into account all the instances of the same rule
 - ▶ the parameters computed for a rule must be applied to all of its instances
- ▶ RIB applied to LPAD
 - ▶ Each grounding of each clause corresponds to a hidden variable

RIB for Logic Programs with Annotated Disjunctions

$r1 = x1:0.2 ; x2:0.7.$

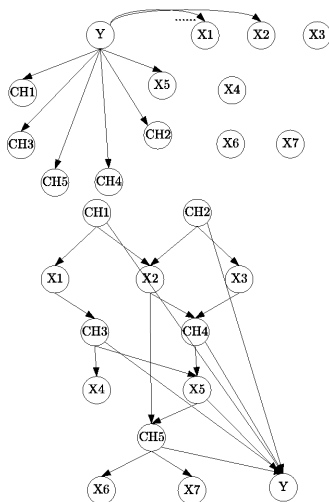
$r2 = x2:0.1 ; x3:0.8.$

$r3 = x4:0.7 ; x5:0.2 \text{ :- } x1.$

$r4 = x5:0.6 \text{ :- } x2, x3.$

$r5 = x6:0.4 ; x7:0.5 \text{ :- } x2, x5.$

- ▶ **X**: Boolean random variables for atoms
- ▶ Hidden variables:
 - ▶ **CH**: choice variables for each rule
 - ▶ plus variables for unobserved atoms in the data ($x5$)



IMDB Network: RIB evaluation

- ▶ Database of movies, actors, directors
 - ▶ director(a), actor/1, movie/2, gender/2, workedUnder/2, sameMovie/2, sameGenre/2, samePerson/2
- ▶ Predicting sameperson(A,B)

sameperson(X,Y):t :- movie(M,X),movie(M,Y).

sameperson(X,Y):t :- actor(X), actor(Y),
workedunder(X,Z), workedunder(Y,Z).

sameperson(X,Y):t :- gender(X,Z), gender(Y,Z).

sameperson(X,Y):t :- director(X), director(Y),
genre(X,Z), genre(Y,Z).

Results

- ▶ Average area under the precision recall curve
 - ▶ RIB: **0.199** (0.02 h)
 - ▶ LeProbLog: 0.096 (0.35 h)
 - ▶ Alchemy: 0.107 (1.54 h)

Cora Network: RIB evaluation

Database of bibliography citation

- ▶ 1295 different citations to 132 different research papers
- ▶ information about the title, the authors and the venue
- ▶ The task is to **deduplicate citations**
 - ▶ i.e., to predict the predicate `samebib(cit1,cit2)`
- ▶ `sameauthor(aut1,aut2)`, `sametitle(tit1,tit2)` and `samevenue(ven1,ven2)`, `haswordauthor(aut,wor)`, `haswordtitle(tit,wor)` and `haswordvenue(ven,wor)`.

Results

- ▶ Average area under the precision recall curve
 - ▶ RIB: **0.939** (2.49 h)
 - ▶ LeProbLog: 0.905 (13.25 h)
 - ▶ Alchemy: 0.469 (1.30 h)

Markov Logic

- ▶ Combining First-order Logic and Markov Networks
 - ▶ First-order formulas with weights are templates for Markov networks
- ▶ Learning
 - ▶ Generative or discriminative learning of parameters
 - ▶ Inductive Logic Programming and MAP score for structure learning
- ▶ Inference
 - ▶ MAP: Weighted satisfiability
 - ▶ Marginal: MCMC with moves proposed by SAT solver
 - ▶ Partial grounding and Lazy inference

Markov Logic: Intuition and Definition

- ▶ **Hard constraints** define a logical KB constraining the set of possible worlds
- ▶ **Soft constraints**: when a world violates a formula, it becomes less probable, not impossible
- ▶ **Formula with a weight**
 - ▶ Higher weight \Rightarrow Stronger constraint

$$P(\text{world}) \approx \exp\left(\sum \text{weights of formulas it satisfies}\right)$$

- ▶ **A Markov Logic Network (MLN)** is a set of pairs (F, w) where
 - ▶ F is a formula in first-order logic and w is a real number
- ▶ Given a set of constants, a MLN defines a Markov network with
 - ▶ a node for each grounding of each predicate in the MLN
 - ▶ a feature for each grounding of each formula F in the MLN, with the corresponding weight w

Markov Logic: an example

Smoking cause cancer.

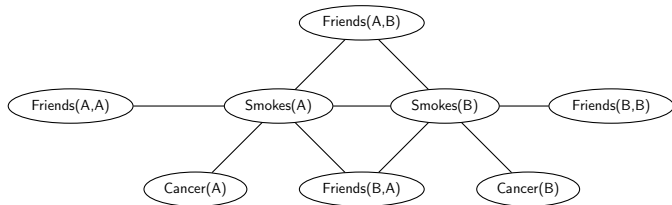
If two people are friends, either both smoke or neither does.



1.5: $\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)$

1.1: $\forall x \forall y \text{ Friends}(x,y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))$

Constants: Anna (A) and Bob (B)



Markov Logic Networks

- ▶ MLN is a template for ground Markov networks
- ▶ Probability of a world x

$$P(x) = \frac{1}{Z} \exp \left(\sum_i w_i n_i(x) \right)$$

- ▶ w_i : weight of formula i
- ▶ $n_i(x)$: number of true groundings of formula i in x

Inference

- ▶ $P(F|MLN, C)$
 - ▶ solved combining MCMC and WalkSAT \rightarrow MC-SAT algorithm
(Poon & Domingos, 2006)

MAP inference

- ▶ Find the most likely state of the world given evidence

$$\arg \max_y P(y|x) \rightarrow \arg \max_y \frac{1}{Z_x} \exp \left(\sum_i w_i n_i(x, y) \right)$$

- ▶ a weighted MaxSAT problem

MLNs Weight Learning

- ▶ Parameter tying: Groundings of same clause

$$\frac{\partial}{\partial w_i} \log P_w(x) = n_i(x) - E_w[n_i(x)]$$

- ▶ $n_i(x)$ number of times clause i is true in data
- ▶ $E_w[n_i(x)]$: expected number of times clause i is true according to MLN
- ▶ generative learning: Pseudo-likelihood
- ▶ discriminative learning: Conditional likelihood, use MC-SAT or MaxWalkSAT for inference

IRoTS and MC-IRoTS for MAP/MPE and conditional inference

(Biba et al., 2010)

- ▶ IRoTS is a MAX-SAT solver based on Iterated Local Search and Robust Tabu Search
- ▶ MC-IToTS combines IRoTS with MCMC

MLNs Structure Learning

- ▶ Start with a unit clauses or with a coded KB
- ▶ *Operators*: Add/remove literal, flip sign
- ▶ *Evaluation function*: Pseudo-likelihood and Structure prior
- ▶ *Search*: Beam, shortest-first, bottom-up
(Kok & Domingos, 2005; Mihalkova & Mooney, 2007)
 - ▶ Generative and Discriminative structure learning with iterated local search (Biba et al., 2008)

Trading expressivity in SRL

- ▶ Trade-off between the expressivity and computational efficiency
- ▶ Propositionalization (Kramer, 2001)
 - ▶ map the relational instance space to a propositional space
 - ▶ apply statistical machine learning techniques in this simpler space

Main Idea

- ▶ Learning relational features f_j from examples e_i
- ▶ $(i, j) = 1$ if $f_j \models e_i$, 0 otherwise

	e_1	e_2	e_3	e_4	e_5
f_1	0	1	0	1	1
f_2	1	1	0	0	0
f_3	0	0	0	1	1
f_4	0	0	1	1	1

Dynamic propositionalization

- ▶ integrate learning procedure
- ▶ nFOIL: Integrating Naive Bayes and FOIL (Landwehr et al., 2005)
 - ▶ change the FOIL algorithm to drive the feature search by the criterion of naïve Bayes
- ▶ kFOIL: Learning Simple Relational Kernels (Landwehr et al., 2006)
 - ▶ similar to nFOIL, but employs kernel methods

Relational Sequence Learning

Sequence Learning

- ▶ sequence prediction: predict elements based on preceding elements
- ▶ frequent sequence mining: sequences frequently occurring
- ▶ sequence classification: predicting a sequence class label
- ▶ sequence labeling: assign a label to each sequence element

Relational complex sequence

- ▶ opposite to flat sequence: ACCAGTAAGGACGT...
- ▶ sequence of logical atoms

*strand(sa,plus,short), helix(right,alpha,medium), strand(blb,plus,short),
helix(right,f3to10,short), strand(sa,minus,long), strand(blb,plus,short), ...*

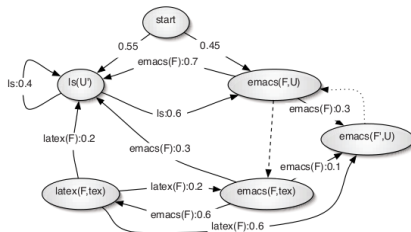
- ▶ helices and strands as protein secondary structure elements are defined in terms of their orientation, type, and length.

▶ Approaches

- ▶ Relational grams, Fisher Kernels for logical sequences, Logical Hidden Markov Models, Relational Conditional Random Fields, Lynx

Logical Hidden Markov Models

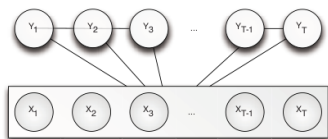
- ▶ Sequence of first-order atoms
- ▶ Abstract transitions: $p : H \xrightarrow{\circ} B$, where
 - ▶ $p \in [0, 1]$
 - ▶ H , B , and \circ are atoms
 - ▶ H and B are abstract states
 - ▶ \circ represents an abstract output symbol



- ▶ classification accuracy improved adopting *Fisher kernel*
(Kersting et al., 2004)

Relational Conditional Random Fields

- ▶ Training CRFs for logical sequences (TildeCRF)
 - ▶ using a relational regression tree
 - ▶ potential functions are represented as weighted sums of relational regression trees
 - ▶ learning via functional gradient ascent



$$P(Y|X) = \frac{1}{Z} \exp \sum_{t=1}^T \psi_t(y_t, X) + \psi_{t-1,t}(y_{t-1}, y_t, X)$$

$$\psi(y_t, X) = \sum \alpha_k g_k(y_t, X)$$

$$\psi(y_{t-1}, y_t, X) = \sum \beta_k f_k(y_{t-1}, y_t, X)$$

Lynx

- ▶ Relational feature construction
- ▶ Wrapper feature selection approach
 - ▶ Optimizing the model likelihood with SLS

Feature construction

- ▶ pattern mining
- ▶ learning relational patterns f_j from sequences e_i

For each sequence X_k we build a vector-valued $\mathbf{x} = (x_1, x_2, \dots, x_d)$ random variable

- ▶ each $x_i \in \mathbf{x}$ is 1 if the pattern p_i covers the sequence X_k , and 0 otherwise

	x_1	x_2	x_3	x_4	x_5
p_1	0	1	0	0	1
p_2	1	1	0	0	0
p_3	0	0	0	1	1
p_4	0	0	1	1	1

Lynx

- ▶ pattern based classification

$$p(Y_j|\mathbf{x}) = \frac{p(\mathbf{x}|Y_j)p(Y_j)}{\sum_{i=1}^Q p(\mathbf{x}|Y_i)p(Y_i)}$$

- ▶ discriminant function

$$g_j(\mathbf{x}) = \ln p(\mathbf{x}|Y_j) + \ln P(Y_j)$$

- ▶ probabilistic features: we define

$$p_{ij} = \text{Prob}(x_i = 1|Y_j)$$

- ▶ estimated by frequency counts on the training sequences
- ▶ $\hat{p}_{ij} = \text{support}_{Y_j}(p_i)$

$$g_j(\mathbf{x}) = \sum_{i=1}^d x_i \ln \frac{p_{ij}}{1 - p_{ij}} + \sum_{i=1}^d \ln(1 - p_{ij}) + \ln p(Y_j)$$

Lynx: feature selection

- ▶ \mathcal{P} : constructed original set of patterns
- ▶ $f : 2^{|\mathcal{P}|} \rightarrow \mathbb{R}$: function scoring a selected subset $X \subseteq \mathcal{P}$

Goal

- ▶ find a subset $\hat{X} \subseteq \mathcal{P}$ such that $f(\hat{X}) = \max_{Z \subseteq \mathcal{P}} f(Z)$
- ▶ Given $P \subseteq \mathcal{P}$, for each sequence X_j we want the MAP hypothesis

$$\hat{h}_P(X_j) = \arg \max_i g_i(\mathbf{x}_j)$$

- ▶ \mathbf{x}_j is the feature based representation of sequence X_j obtained using patterns in P
- ▶ the optimization problem corresponds to minimise the expectation $E[\mathbf{1}_{\hat{h}_P(X_i) \neq Y_i}]$ ($\mathbf{1}_{\hat{h}_P(X_i) \neq Y_i} = 1$ if $\hat{h}_P(X_i) \neq Y_i$, and 0 otherwise)
- ▶ D the training set, $|D| = m$, the number of classification errors made by the Bayesian model is

$$\text{err}_D(P) = mE[\mathbf{1}_{\hat{h}_P(X_i) \neq Y_i}]$$

Grasp

Require: D : the training set; \mathcal{P} : a set of patterns (features); *maxiter*: maximum number of iterations; $\text{err}_D(\mathcal{P})$: the evaluation function

Ensure: solution $\hat{S} \subseteq \mathcal{P}$

```
1:  $\hat{S} = \emptyset$ ,  $\text{err}_D(\hat{S}) = +\infty$ 
2: iter = 0
3: while iter < maxiter do
4:    $\alpha = \text{rand}(0,1)$ 
5:    $S = \emptyset$ ;  $i = 0$ 
6:   while  $i < n$  do
7:      $\mathcal{S} = \{S' | S' = \text{add}(S, A)\}$ 
8:      $\bar{s} = \max\{\text{err}_D(T) | T \in \mathcal{S}\}$ 
9:      $\underline{s} = \min\{\text{err}_D(T) | T \in \mathcal{S}\}$ 
10:     $\text{RCL} = \{S' \in \mathcal{S} | \text{err}_D(S') \leq \underline{s} + \alpha(\bar{s} - \underline{s})\}$ 
11:    select the new  $S$ , at random, from RCL
12:     $i \leftarrow i + 1$ 
13:     $\mathcal{N} = \{S' \in \text{neigh}(S) | \text{err}_D(S') < \text{err}_D(S)\}$ 
14:    while  $\mathcal{N} \neq \emptyset$  do
15:      select  $S \in \mathcal{N}$ 
16:       $\mathcal{N} \leftarrow \{S' \in \text{neigh}(S) | \text{err}_D(S') < \text{err}_D(S)\}$ 
17:    if  $\text{err}_D(S) < \text{err}_D(\hat{S})$  then
18:       $\hat{S} = S$ 
19:    iter = iter + 1
20: return  $\hat{S}$ 
```

Lynx: Results

- ▶ protein fold classification
- ▶ logical sequences of the secondary structure of protein domains
- ▶ **task**: predict one of the five most populated SCOP folds of alpha and beta proteins (a/b)

Cross validated accuracy w/ and w/o feature selection

Conf.	Lynx	Mean
0.95	w/o GRASP ^{FS}	0.826
	w GRASP ^{FS}	0.878
1.0	w/o GRASP ^{FS}	0.896
	w GRASP ^{FS}	0.942

Cross-validated accuracy of LoHMMs, Fisher kernels, TildeCRF and Lynx

System	Accuracy
LoHMMs	75%
Fisher kernels	84%
TildeCRF	92.96%
Lynx	94.15%

Conclusions and Open Issues

With SRL we have

- ▶ better predictive accuracy and understanding of real world domains

Issues

- ▶ Learning is much harder
- ▶ Scalability of the inference algorithms (inference is crucial)
- ▶ Model counting: weighted model counting
 - ▶ compilation to Binary Decision Diagrams (BDDs) does not scale
 - ▶ approximate compilation
- ▶ incremental learning (evolving networks)

Open problems

- ▶ Scalability
 - ▶ Use only information which affect the best action choice
 - ▶ Tighter connection with work done on probabilistic databases
- ▶ Killer applications
 - ▶ Social networks
 - ▶ Spatial data analysis
- ▶ Dynamic networks
 - ▶ Non-stationary distributions
 - ▶ Data streaming
- ▶ Learning structured internal representations
 - ▶ More compact representations
 - ▶ Recursive random fields (Lows & Domingos, 2007)

References

- ▶ M. Biba, S. Ferilli and F. Esposito, *Structure Learning of Markov Logic Networks through Iterated Local Search*, 2008.
- ▶ M. Biba, S. Ferilli and F. Esposito, *High Performing Algorithms for MAP and Conditional Inference in Markov Logic*, 2009.
- ▶ L. De Raedt, P. Frasconi, K. Kersting and S. Muggleton. *Probabilistic Inductive Logic Programming*, 2008.
- ▶ N. Di Mauro, T.M.A. Basile, S. Ferilli and F. Esposito, *Feature Construction for Relational Sequence Learning*, 2010.
- ▶ N. Di Mauro, T.M.A. Basile, S. Ferilli and F. Esposito, *Approximate relational reasoning by stochastic propositionalization*, 2010.
- ▶ N. Di Mauro, T.M.A. Basile, S. Ferilli and F. Esposito, *Optimizing Probabilistic Models for Relational Sequence Learning*, 2011.
- ▶ F. Esposito, S. Ferilli, T.M.A. Basile and N. Di Mauro, *Social Networks and Statistical Relational Learning: A Survey*, 2011.
- ▶ S. Ferilli, T.M.A. Basile and N. Di Mauro, *Markov Logic Networks for Document Layout Correction*, 2011
- ▶ L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning*, 2007.
- ▶ F. Riguzzi and N. Di Mauro, *Applying the Information Bottleneck to Statistical Relational Learning*, 2011.
- ▶ C. Taranto, N. Di Mauro and F. Esposito, *Probabilistic Inference over Image Networks*, 2011.