Scalable Collective Inference in Heterogeneous Networks using PSL

Prof. Lise Getoor
University of California, Santa Cruz
http://www.cs.umd.edu/~getoor

Stanford InfoSeminar
January 17, 2014



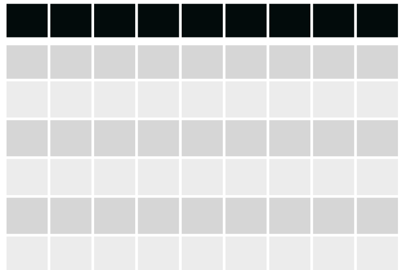




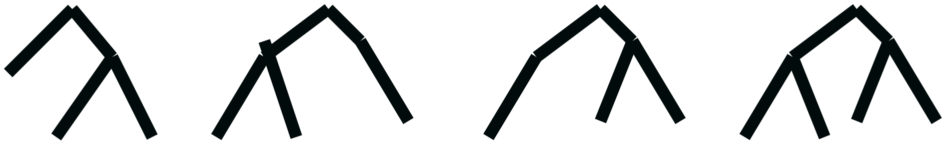


Big Data: What's Different?

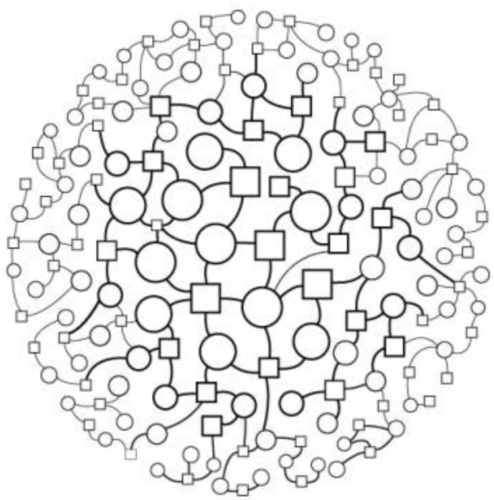
Most of the data does not look like this



Or even like this



It looks more like this



Statistical Relational Learning (SRL)

- AI/DB representations + statistics for multi-relational data
  - Entities can be of different types
  - Entities can participate in a variety of relationships
  - examples: Markov logic networks, relational dependency networks, Bayesian logic programs, probabilistic relational models, many others…..
- Key ideas
  - Relational feature construction
  - Collective reasoning
  - ‘Lifted’ representation, inference and learning

http://linqs.cs.umd.edu/projects//Tutorials/nips2012.pdf

http://psl.umiacs.umd.edu

Probabilistic Soft Logic

Stephen Bach

Matthias Broecheler

Alex Memory

Lily Mihalkova

Stanley Kok

Angelika Kimmig

Bert Huang

On job market
year!

Ben London

Arti Ramesh

Jay Pujara

Shobeir Fakhraei

Hui Miao

Probabilistic Soft Logic (PSL)

Declarative language based on logics to express collective probabilistic inference problems

- Predicate = relationship or property
- Atom = (continuous) random variable
- Rule = capture dependency or constraint
- Set = define aggregates

PSL Program = Rules + Input DB

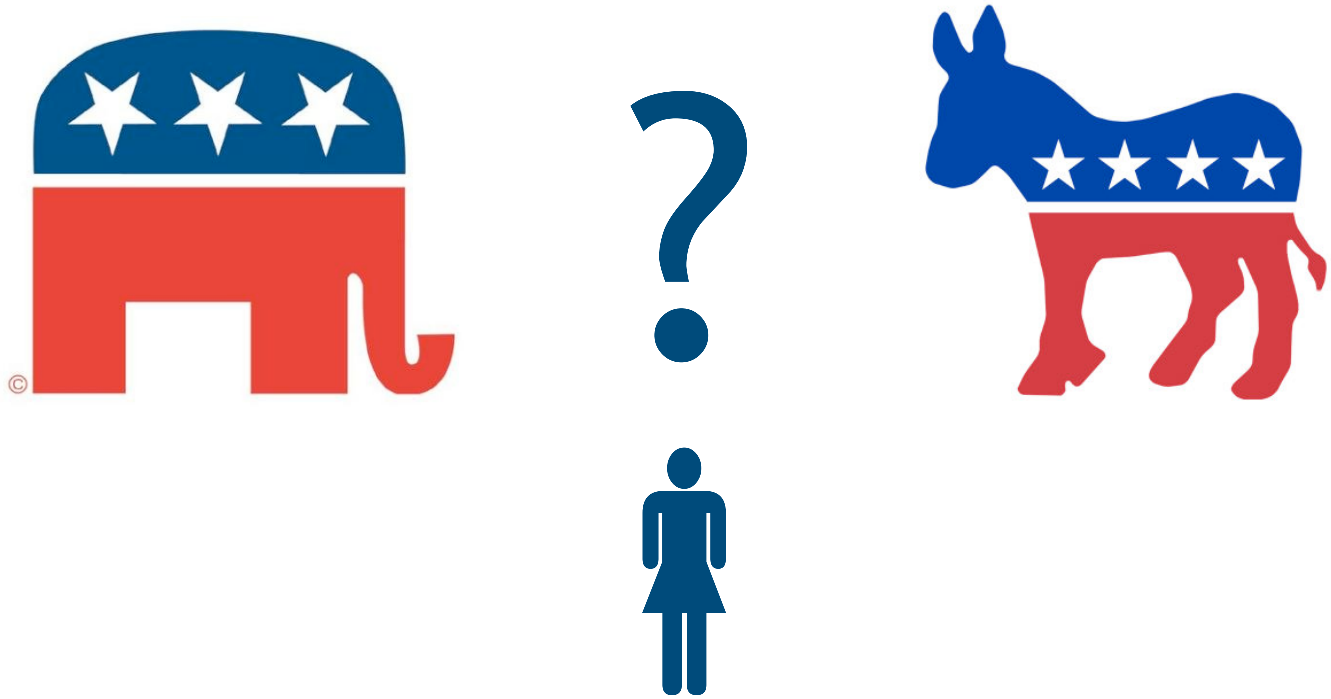
Probabilistic Soft Logic (PSL)

Declarative language based on logics to express collective probabilistic inference problems

- Predicate = relationship or property
- Atom = (continuous) random variable
- Rule = capture dependency or constraint
- Set = define aggregates

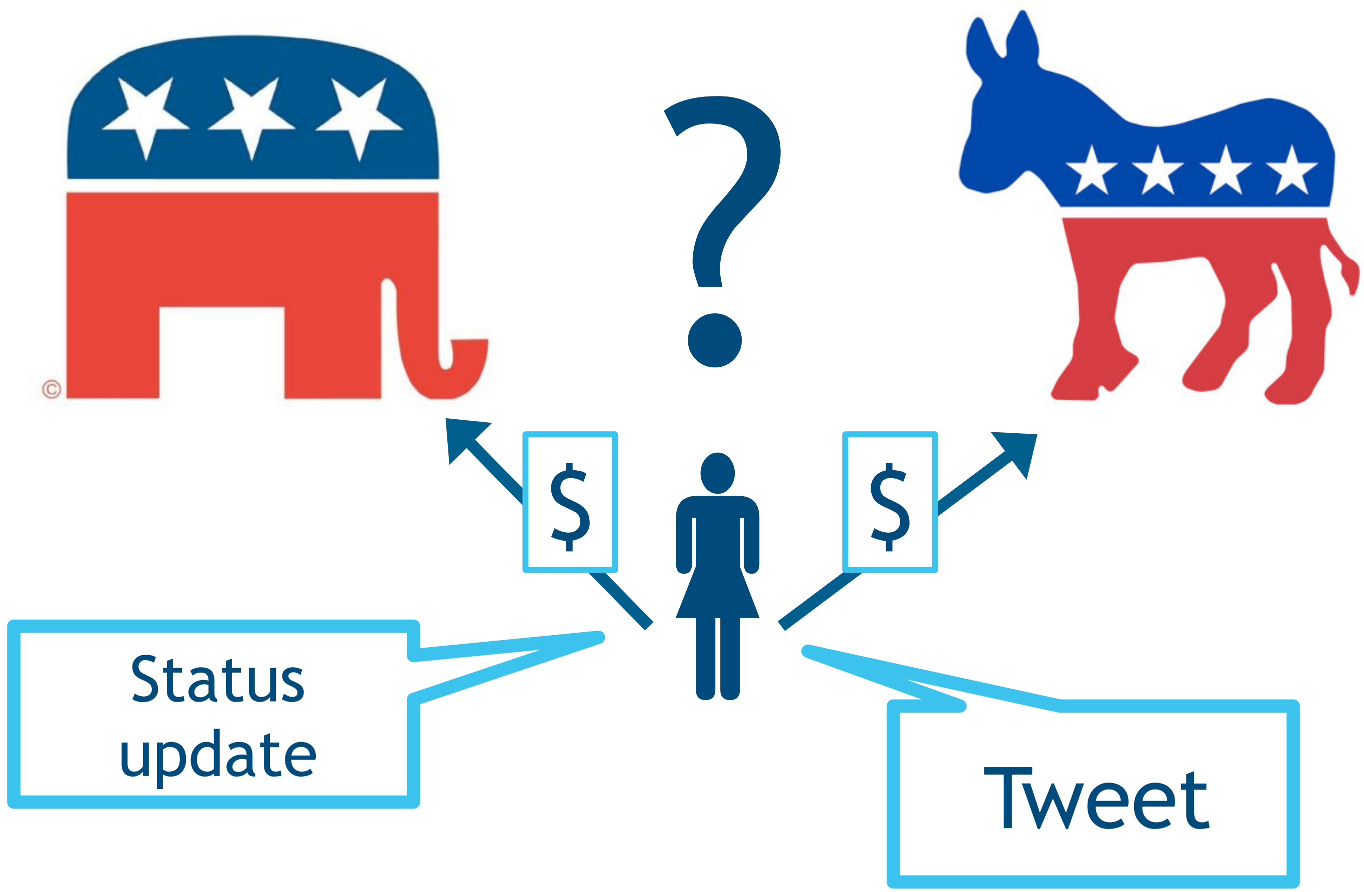
PSL Program = Rules + Input DB

Collective Classification



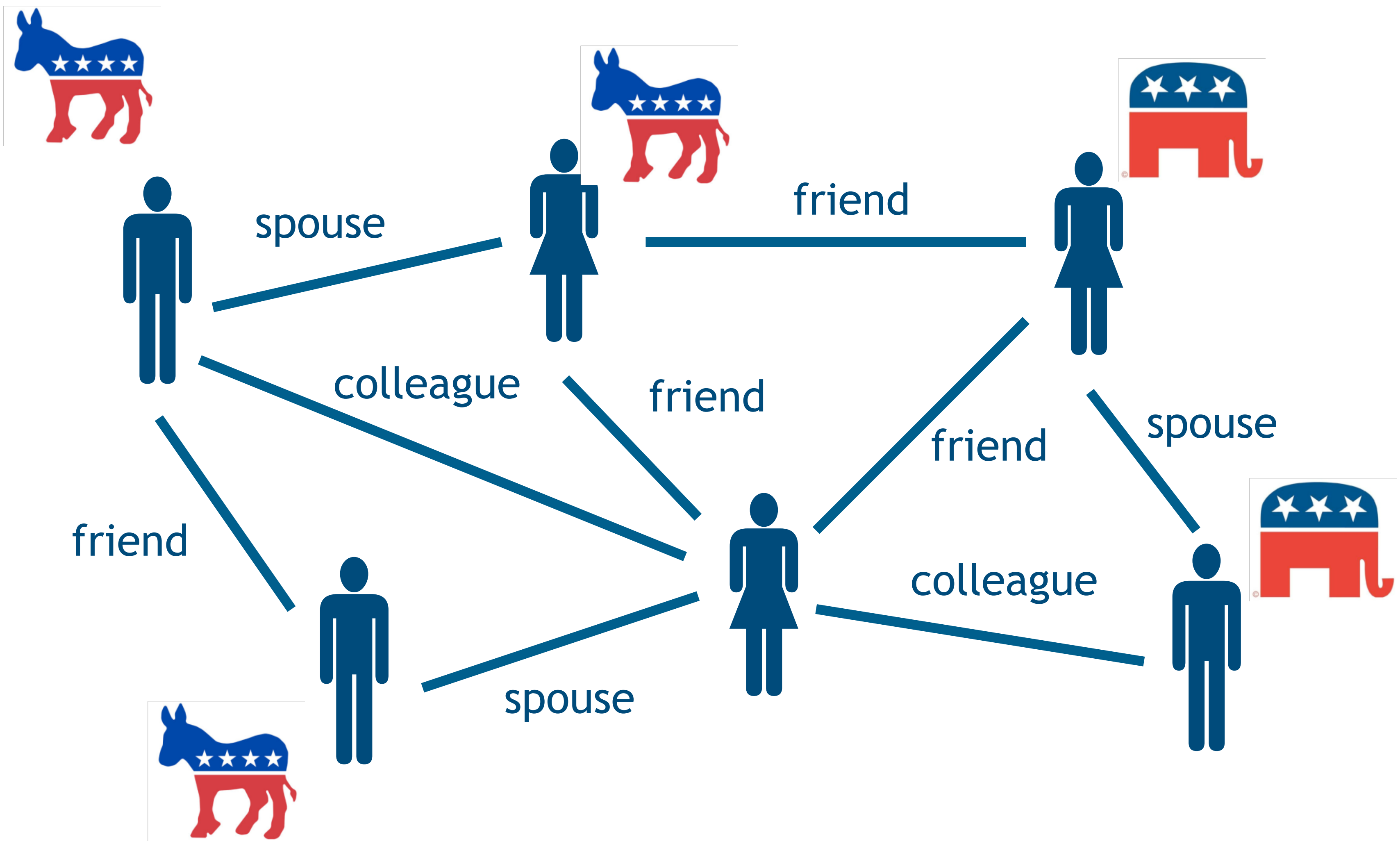
http://psl.umiacs.umd.edu

Voter Opinion Modeling



http://psl.umiacs.umd.edu

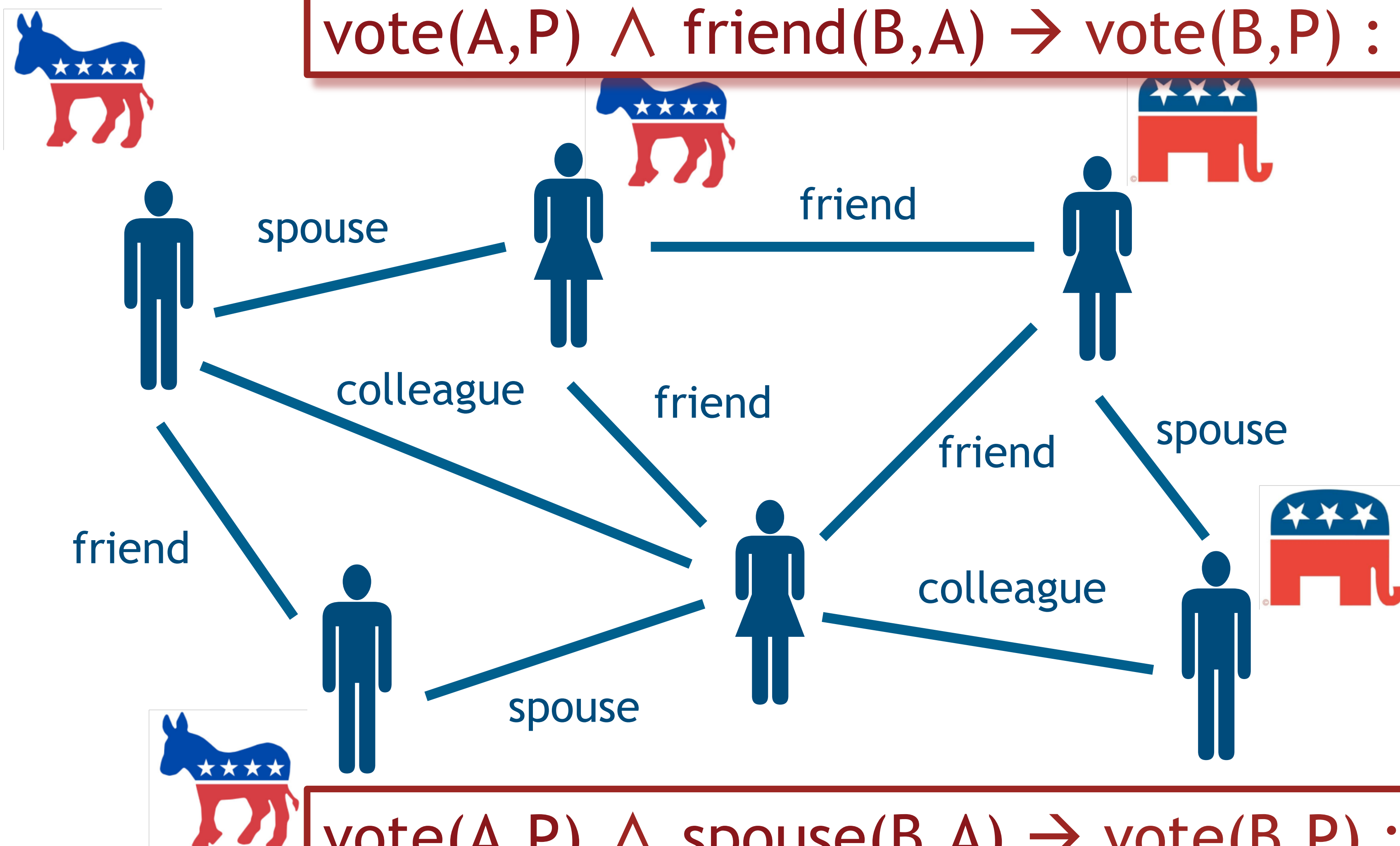
Voter Opinion Modeling



http://psl.umiacs.umd.edu

Voter Opinion Modeling

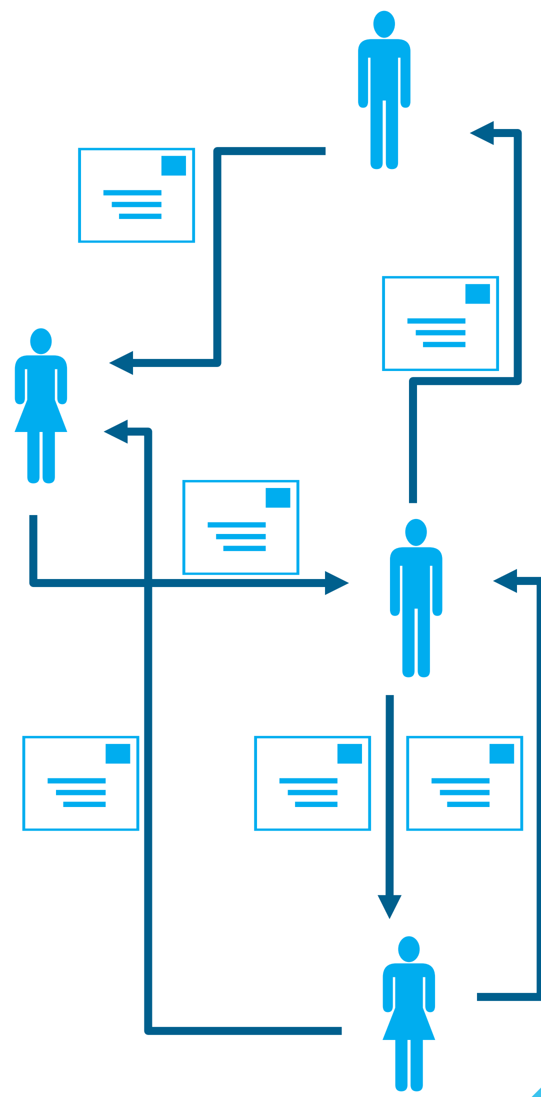
vote(A,P) ∧ friend(B,A) → vote(B,P) : 0.3



vote(A,P) ∧ spouse(B,A) → vote(B,P) : 0.8

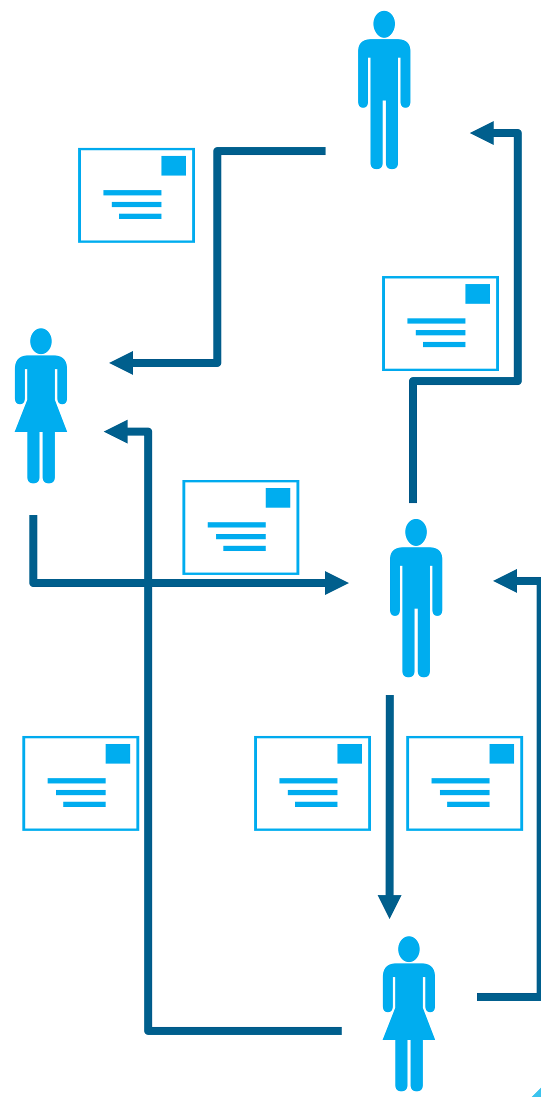
Link Prediction

- Entities
  - People, Emails
- Attributes
  - Words in emails
- Relationships
  - communication, work relationship
- Goal: Identify work relationships
  - Supervisor, subordinate, colleague



Link Prediction

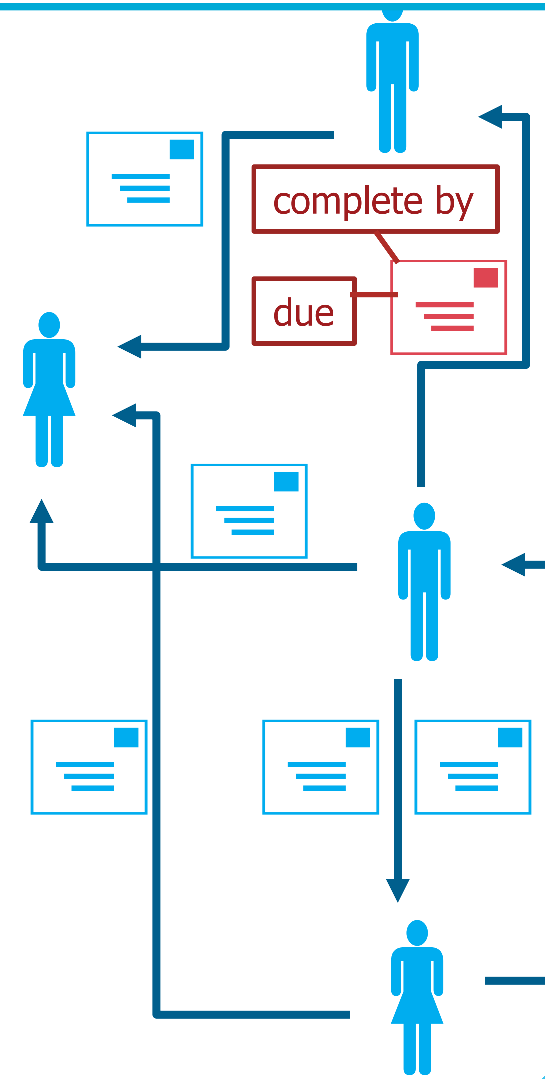
- People, emails, words, communication, relations
- Use rules to express evidence
  - "If email content suggests type X, it is of type X"
  - "If A sends deadline emails to B, then A is the supervisor of B"
  - "If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues"



Link Prediction

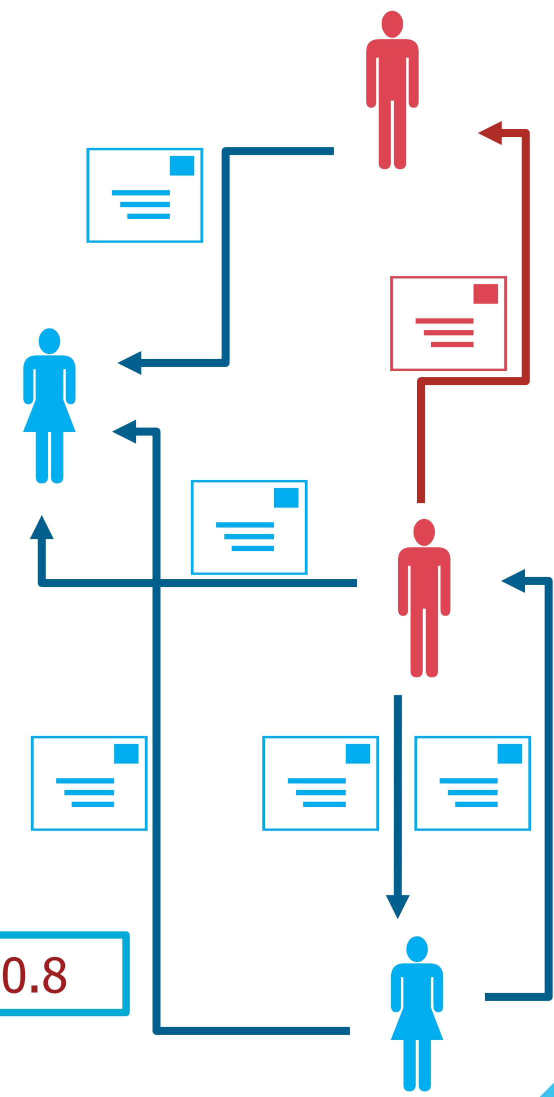
HasWord(E, "due") => Type(E, deadline) : 0.6

- People, emails, words, communication, relations
- Use rules to express evidence
  - "If email content suggests type X, it is of type X"
  - "If A sends deadline emails to B, then A is the supervisor of B"
  - "If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues"



Link Prediction

- People, emails, words, communication, relations
- Use rules to express evidence
  - "If email content suggests type X, it is of type X"
  - "If A sends deadline emails to B, then A is the supervisor of B"
  - "If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues"

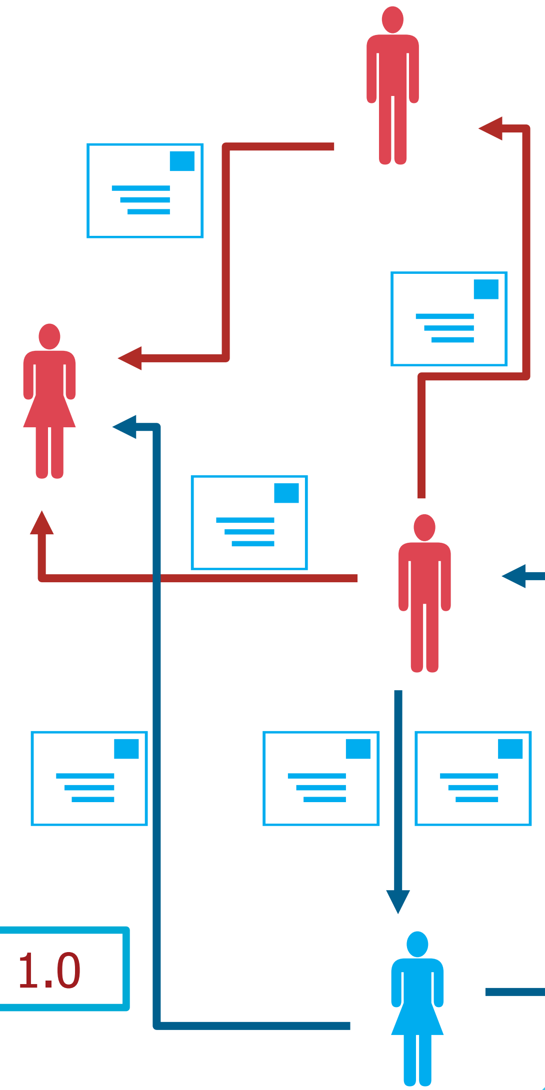


Sends(A,B,E) ^ Type(E,deadline) => Supervisor(A,B) : 0.8

Link Prediction

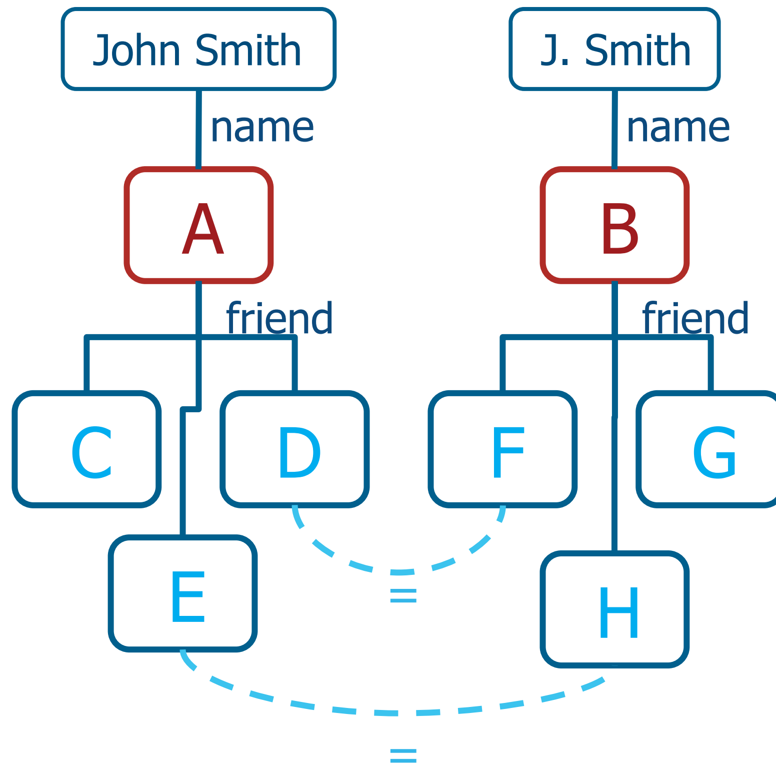
- People, emails, words, communication, relations
- Use rules to express evidence
  - "If email content suggests type X, it is of type X"
  - "If A sends deadline emails to B, then A is the supervisor of B"
  - "If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues"

Supervisor(A,B) ^ Supervisor(A,C) => Colleague(B,C) : 1.0



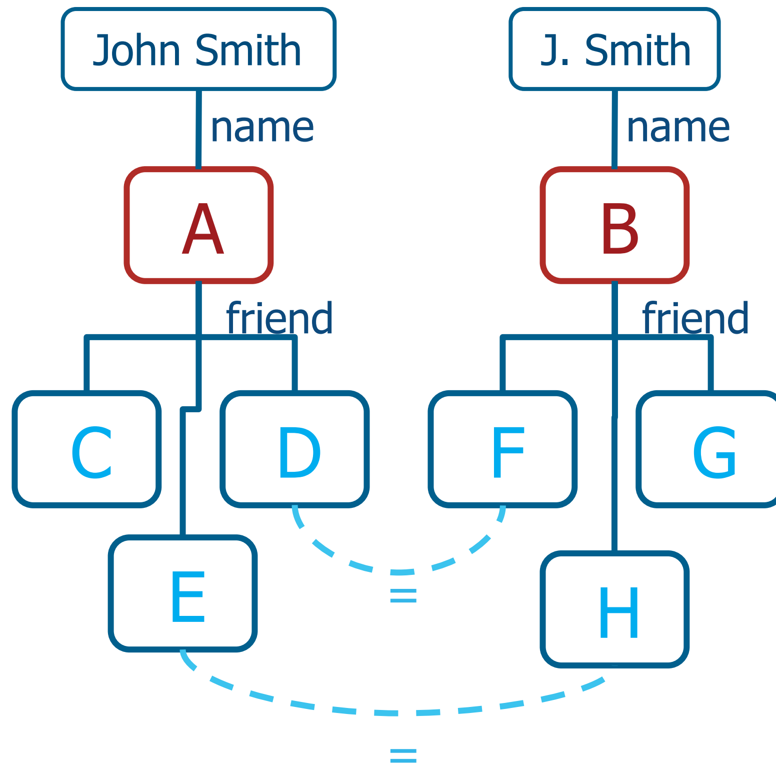
Entity Resolution

- Entities
  - People References
- Attributes
  - Name
- Relationships
  - Friendship
- Goal: Identify references that denote the same person



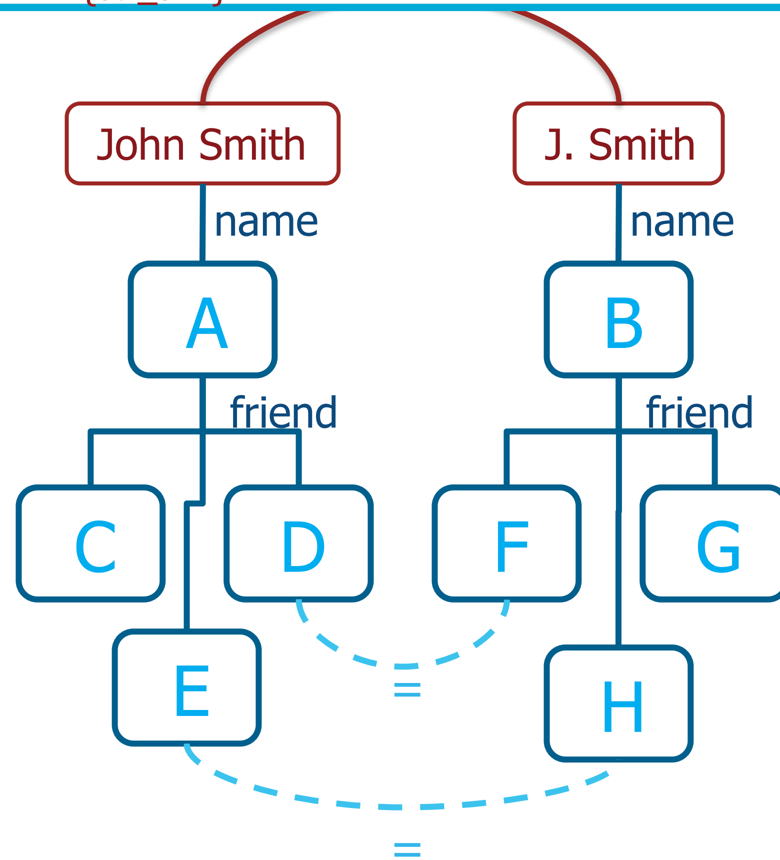
Entity Resolution

- References, names, friendships
- Use rules to express evidence
  - ‘’If two people have similar names, they are probably the same’’
  - ‘’If two people have similar friends, they are probably the same’’
  - ‘’If A=B and B=C, then A and C must also denote the same person’’



Entity Resolution

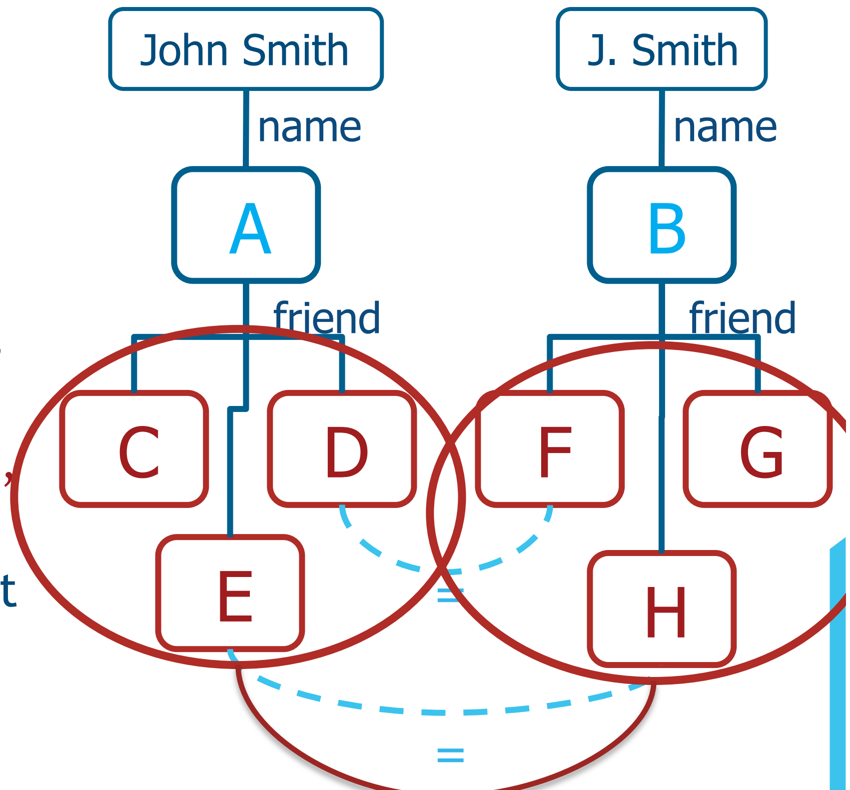
$A.name \approx_{\{str\_sim\}} B.name \Rightarrow A \approx B : 0.8$



- References, names, friendships
- Use rules to express evidence
  - ‘’ If two people have similar names, they are probably the same’’
  - ‘’ If two people have similar friends, they are probably the same’’
  - ‘’ If A=B and B=C, then A and C must also denote the same person’’

Entity Resolution

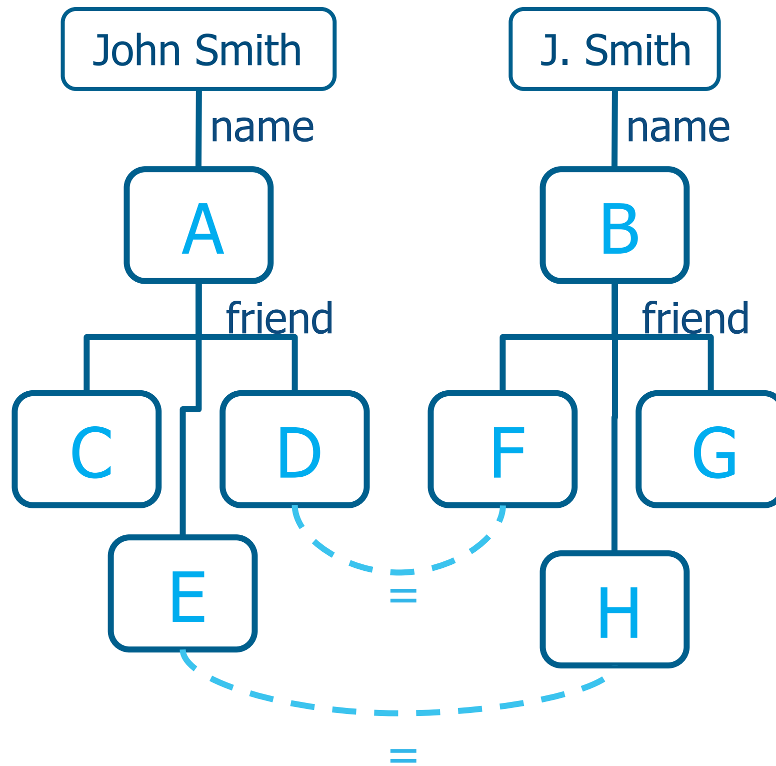
- References, names, friendships
- Use rules to express evidence
  - ‘’ If two people have similar names, they are probably the same’ ’
  - ‘’ If two people have similar friends, they are probably the same’ ’
  - ‘’ If A=B and B=C, then A and C must also denote the same person’ ’



$\{A.friends\} \approx_{\{\}} \{B.friends\} \Rightarrow A \approx B : 0.6$

Entity Resolution

- References, names, friendships
- Use rules to express evidence
  - ‘’ If two people have similar names, they are probably the same’’
  - ‘’ If two people have similar friends, they are probably the same’’
  - ‘’ If A=B and B=C, then A and C must also denote the same person’’



A≈B ^ B≈C => A≈C : ∞

Logic Foundation

http://psl.umiacs.umd.edu

[Broecheler, et al., UAI '10]

"Lifted" Rules

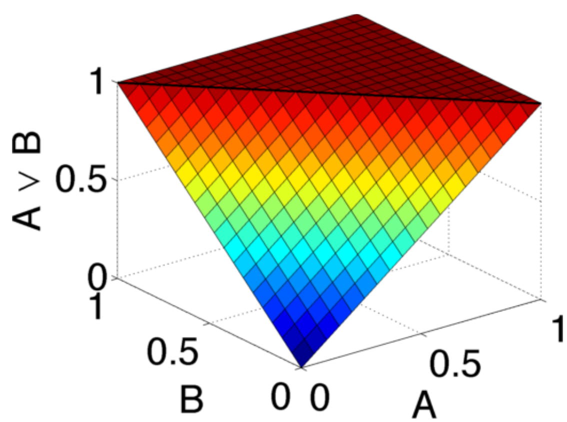
Atoms

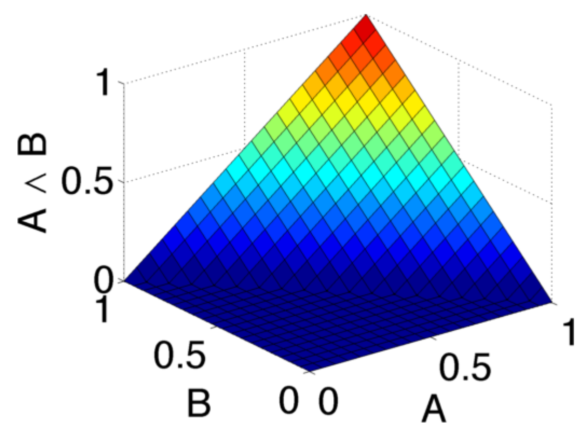
$$H_1(X) \vee \ldots \vee H_m(X) \leftarrow B_1(X) \wedge \ldots \wedge B_n(X)$$

- Will be instantiated for every $x \in X$ in the input
- Atoms are real valued
  - Interpretation I, atom A: $I(A) \in [0,1]$
  - We will omit the interpretation and write $A \in [0,1]$

Combination Functions

- $\vee$, $\wedge$: $[0,1]^n \rightarrow [0,1]$
- Here, we use Lukasiewicz T-norm
  - $A \vee B = \min(1, A + B)$
  - $A \wedge B = \max(0, A + B - 1)$





Rule Satisfaction

$$H_1(X) \leftarrow B_1(X) \wedge B_2(X)$$

- Establish Satisfaction

$$\boxed{\geq 0.5\ H_1(x) \leftarrow B_1(x){:}0.7 \wedge B_2(x){:}0.8}$$

Distance to Satisfaction

$$H_1(X) \vee \ldots \vee H_m(X) \leftarrow B_1(X) \wedge \ldots \wedge B_n(X)$$

- Distance to Satisfaction
  - $\mathrm{Max}(\wedge(B_1(X), \ldots, B_n(X)) - \vee(H_1(X), \ldots, H_m(X)), 0)$

$H_1(x){:}0.7 \leftarrow B_1(x){:}0.7 \wedge B_2(x){:}0.8$	0.0
$H_1(x){:}0.2 \leftarrow B_1(x){:}0.7 \wedge B_2(x){:}0.8$	0.3

Distance to Satisfaction

$$H_1(X) \vee \ldots \vee H_m(X) \leftarrow B_1(X) \wedge \ldots \wedge B_n(X)$$

- Distance to Satisfaction
  - $\mathrm{Max}(\wedge(B_1(X), \ldots, B_n(X)) - \vee(H_1(X), \ldots, H_m(X)), 0)$

- Weighted Rules

$$W_r: H_1(X) \vee \ldots H_m(X) \leftarrow B_1(X) \wedge \ldots \wedge B_n(X)$$

- Weighted Distance to Satisfaction

$$W_r \bullet \max(\wedge(B_1(x), \ldots, B_n(x)) - \vee(H_1(x), \ldots, H_m(x)), 0)$$

So far….

- Given a data set and a PSL program, we can construct a set of ground rules.
- Some of the atoms have fixed truth values and some have unknown truth values.
- For every assignment of truth values to the unknown atoms, we get a set of weighted distances from satisfaction.
- How to decide which is best?

Probabilistic Foundation

Probabilistic Model

Probability density over interpretation *I*

Ground rule's distance to satisfaction

$$d_r(I) = \max\{I_{r,\text{body}} - I_{r,\text{head}}, 0\}$$

$$P(I) = \frac{1}{Z} \exp\left[-\sum_{r \in R} w_r (d_r(I))^{p_r}\right]$$

Normalization constant

Ground rules

Rule weight

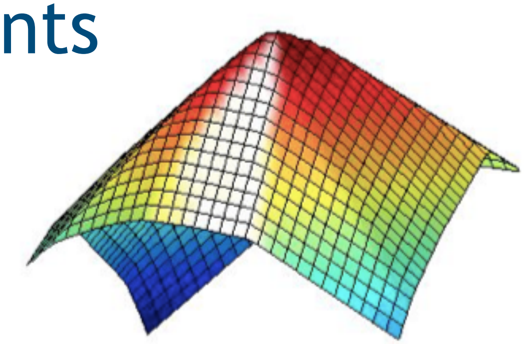
Distance exponent (in {1, 2})

Hinge-loss MRFs

Hinge-loss Markov Random Fields

$$P(\mathbf{Y} \mid \mathbf{X}) = \frac{1}{Z} \exp \left[-\sum_{j=1}^{m} w_j \max\{\ell_j(\mathbf{Y}, \mathbf{X}), 0\}^{p_j} \right]$$

- Continuous variables in [0,1]
- Potentials are hinge-loss functions
- Subject to arbitrary linear constraints
- Log-concave!



Inference as Convex Optimization

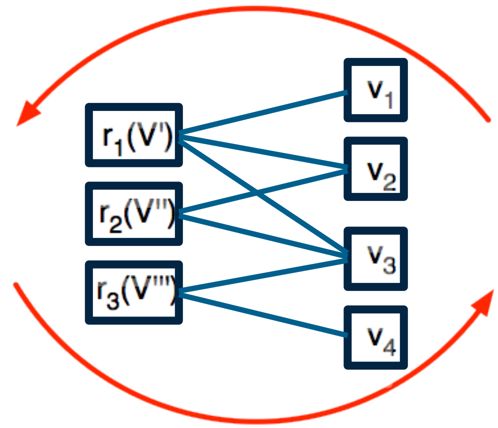
- Maximum Aposteriori Probability (MAP) Objective:

$$\arg\max_{\mathbf{Y}} P(\mathbf{Y} \mid \mathbf{X})$$
$$= \arg\min_{\mathbf{Y}} \sum_{j=1}^{m} w_j \max\{\ell_j(\mathbf{Y}, \mathbf{X}), 0\}^{p_j}$$

- This is convex!
- Can solve using off-the-shelf convex optimization packages
- … or custom solver

Consensus Optimization

- Idea: Decompose problem and solve sub-problems independently (in parallel), then merge results
  - Sub-problems are ground rules
  - Auxiliary variables enforce consensus across sub-problems



- Framework: ***Alternating direction method of multipliers*** (ADMM) [Boyd, 2011]

- Inference with ADMM is fast, scalable, and straightforward to implement [Bach et al., NIPS 2012, UAI 2013]

http://psl.umiacs.umd.edu

Bach et al., UAI 2013

Inference Algorithm

Initialize local copies of variables and Lagrange multipliers

Begin inference iterations

Simple updates solve subproblems for each potential...

...and each constraint

Average to update global variables and clip to [0,1]

Algorithm 1 MPE Inference for HL-MRFs

Input: HL-MRF$(\mathbf{Y}, \mathbf{X}, \phi, \lambda, C, \mathcal{E}, \mathcal{I})$, $\rho > 0$

Initialize $\mathbf{y}_j$ as copies of the variables $\mathbf{Y}_j$ that appear in ϕ_j, $j = 1, \ldots, m$

Initialize $\mathbf{y}_{k+m}$ as copies of the variables $\mathbf{Y}_{k+m}$ that appear in C_k, $k = 1, \ldots, r$

Initialize Lagrange multipliers $\boldsymbol{\alpha}_i$ corresponding to variable copies $\mathbf{y}_i$, $i = 1, \ldots, m + r$

while not converged **do**

  for $j = 1, \ldots, m$ **do**

    $\boldsymbol{\alpha}_j \leftarrow \boldsymbol{\alpha}_j + \rho(\mathbf{y}_j - \mathbf{Y}_j)$

    $\mathbf{y}_j \leftarrow \mathbf{Y}_j - \boldsymbol{\alpha}_j / \rho$

    if $\ell_j(\mathbf{y}_j, \mathbf{X}) > 0$ **then**

      $\mathbf{y}_j \leftarrow \arg\min_{\mathbf{y}_j} \left[\lambda_j [\ell_j(\mathbf{y}_j, \mathbf{X})]^{p_j} + \frac{\rho}{2} \|\mathbf{y}_j - \mathbf{Y}_j + \frac{1}{\rho}\boldsymbol{\alpha}_j\|_2^2 \right]$

      if $\ell_j(\mathbf{y}_j, \mathbf{X}) < 0$ **then**

        $\mathbf{y}_j \leftarrow \text{Proj}_{\ell_j = 0}(\mathbf{Y}_j)$

      end if

    end if

  end for

  for $k = 1, \ldots, r$ **do**

    $\boldsymbol{\alpha}_{k+m} \leftarrow \boldsymbol{\alpha}_{k+m} + \rho(\mathbf{y}_{k+m} - \mathbf{Y}_{k+m})$

    $\mathbf{y}_{k+m} \leftarrow \text{Proj}_{C_k}(\mathbf{Y}_{k+m})$

  end for

  for $i = 1, \ldots, n$ **do**

    $Y_i \leftarrow \frac{1}{|\texttt{copies}(Y_i)|} \sum_{y_c \in \texttt{copies}(Y_i)} \left(y_c + \frac{\alpha_c}{\rho} \right)$

    Clip Y_i to [0,1]

  end for

end while

Bach et al., UAI 2013; London et al., 2013

Speed

Average running time

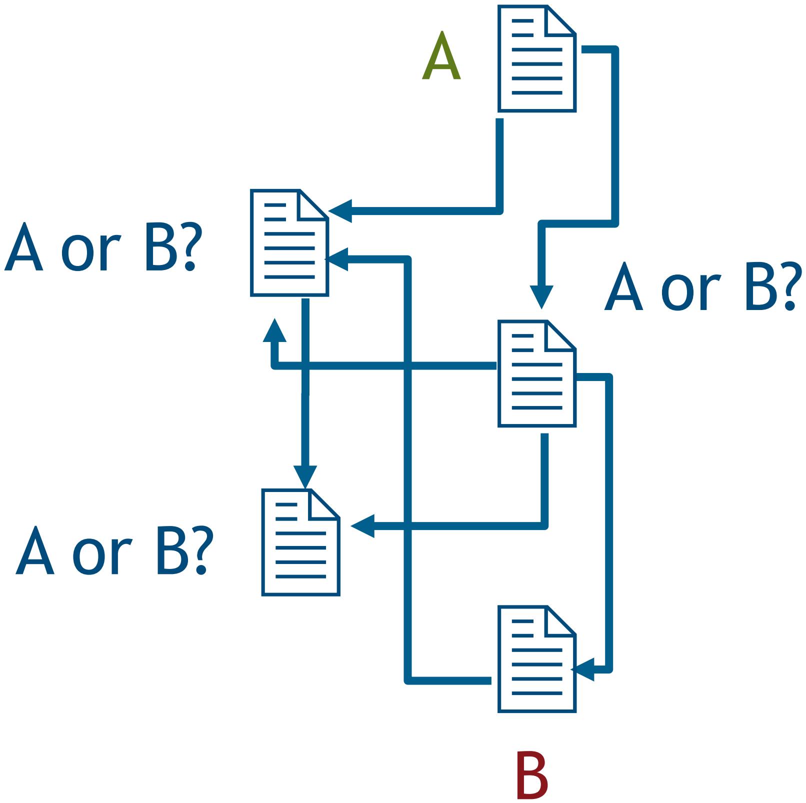
	Cora	Citeseer	Epinions	Activity
Discrete MRF	110.9 s	184.3 s	212.4 s	344.2 s
HL-MRF	**0.4 s**	**0.7 s**	**1.2 s**	**0.6 s**
Variables	**10k**	**10k**	**1k**	**8k**
Potentials	**14k**	**19k**	**18k**	**75k**

- Inference in HL-MRFs is orders of magnitude faster than in discrete MRFs which use MCMC approximate inference
- In practice, scales linearly with the number of potentials

Document Classification

- Given a networked collection of documents
- Observe some labels
- Predict remaining labels using
  - link direction
  - inferred class label

	Citeseer	Cora
HL-MRF-Q (MLE)	**0.729**	**0.816**
HL-MRF-Q (MPLE)	**0.729**	**0.818**
HL-MRF-Q (LME)	0.683	0.789
HL-MRF-L (MLE)	**0.724**	0.802
HL-MRF-L (MPLE)	**0.729**	**0.808**
HL-MRF-L (LME)	0.695	0.789
MLN (MLE)	0.686	0.756
MLN (MPLE)	0.715	0.797
MLN (LME)	0.687	0.783



Accuracy for collective classification. The label accuracy of the highest-scoring category for various HL-MRFs and MLNs. Scores statistically equivalent to the best scoring method are typed in bold.

Distributed MAP Inference

- ADMM consensus optimization problem can be implemented naturally in distributed setting
- For k+1 iteration, it consists three steps in which sub problems can run independently (1st and 2nd step):
  1. Update Lagrangian multiplier

$$y_j^{k+1} \leftarrow y_j^k + \rho(x_j^k - X_j^k)$$

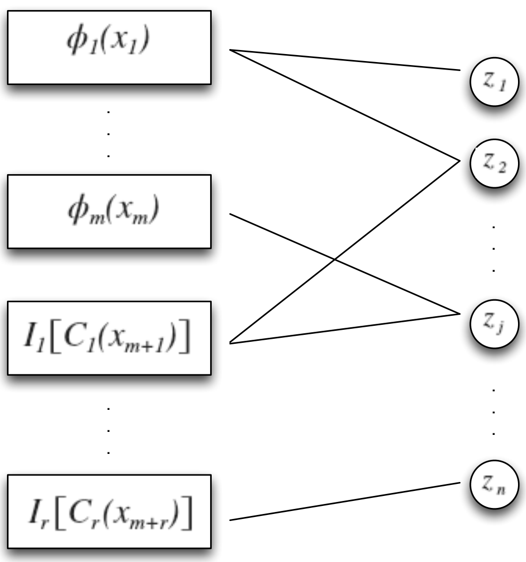
  2. Update each sub problem

$$x_j^{k+1} \leftarrow \arg\min_{x_j} \Lambda_j \phi_j(x_j) + \frac{\rho}{2} \left\| x_j - X_j^k + \frac{1}{\rho} y_j^{k+1} \right\|_2^2$$

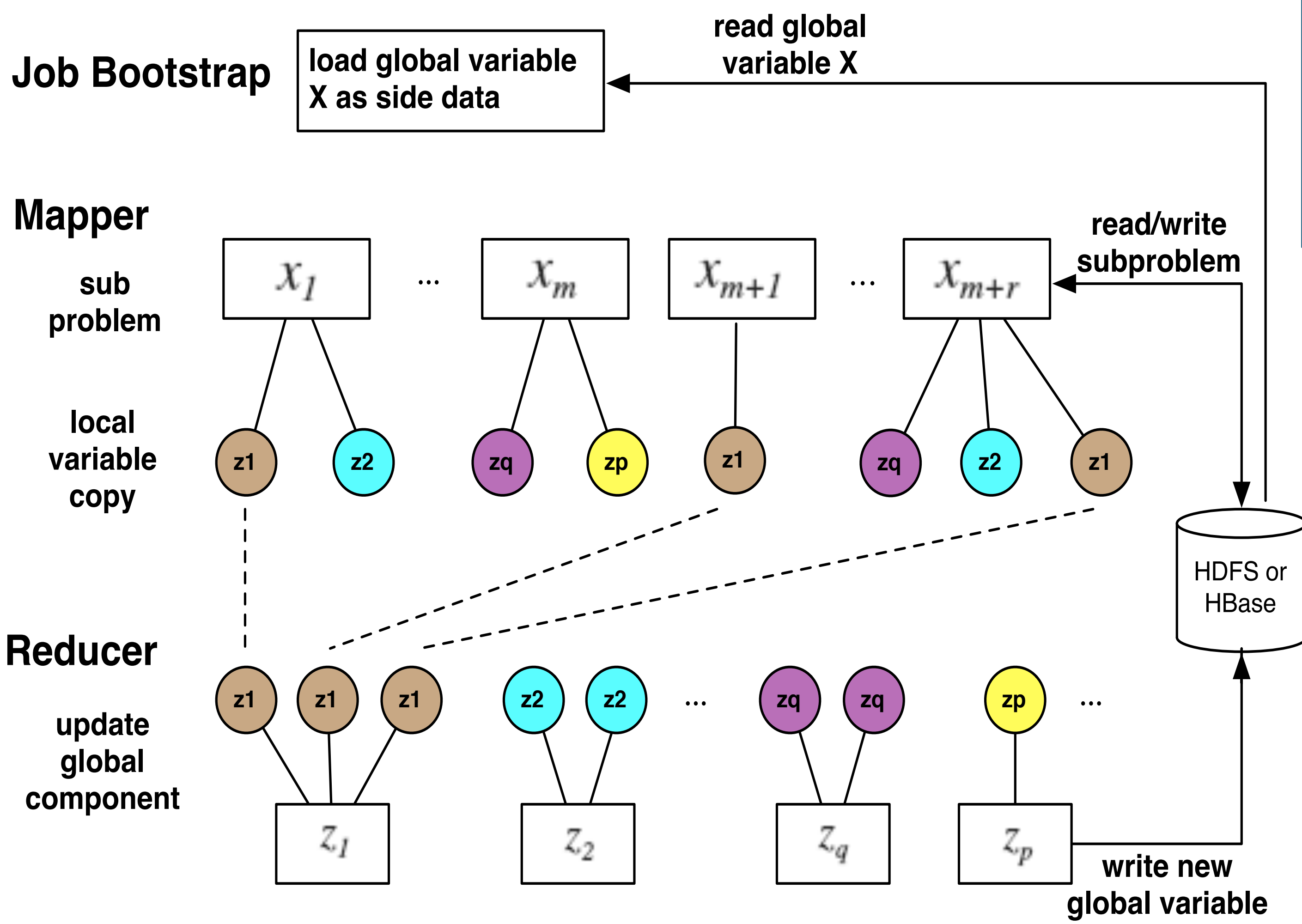
$$x_j^{k+1} \leftarrow \arg\min_{x_j} I_j[C_j(x_j)] + \frac{\rho}{2} \left\| x_j - X_j^k + \frac{1}{\rho} y_j^{k+1} \right\|_2^2$$

  3. Update the global variables

$$z_g^{k+1} \leftarrow \frac{1}{s_g} \sum_{G(i,j)=g} \left(x_i^{k+1} + \frac{y_i^{k+1}}{\rho} \right)_j$$



Distributed MAP: MapReduce



Pros:

- Straightforward Design

Cons:

- Job bootstrapping cost between iterations
- Difficult to schedule subset of nodes to run.

Distributed MAP: GraphLab

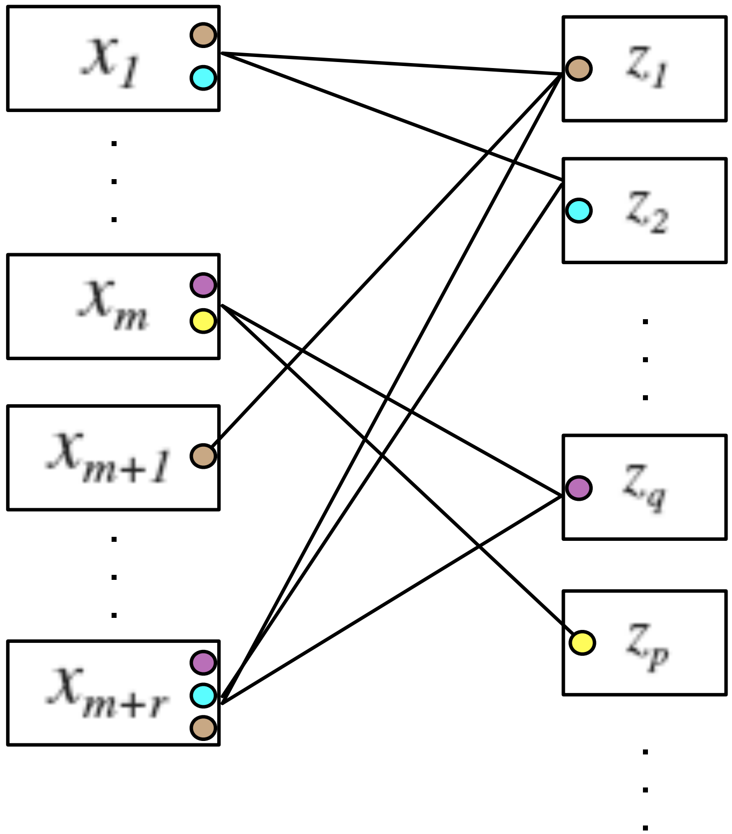
Advantages:
- No need to touch disk, no job bootstrapping cost
- Easy to express local convergence conditions to schedule only subset of nodes.

sub problem node

```
gather
   get z

apply
   update y
   update x

scatter
   notify z
```



global variable compo

```
gather
   get local z,y

apply
   update z

scatter
   unless converge
        notify X
```

update i

update i+1

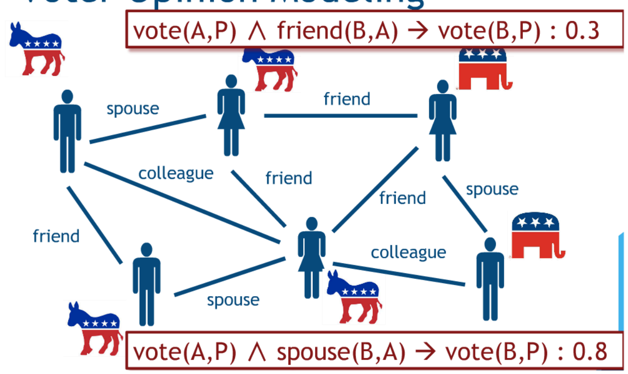
Experimental Results

- Using PSL for knowledge graph cleaning task
  - 16M+ vertices, 22M+ edges, for small running instances
  - Takes 100 minutes to finish in Java single machine implementation using 40G+ memory
  - Distributed GraphLab implementation takes less than 15 minutes using 4 smaller machines
  - Possible to use commodity machines on large models!

Experimental Results

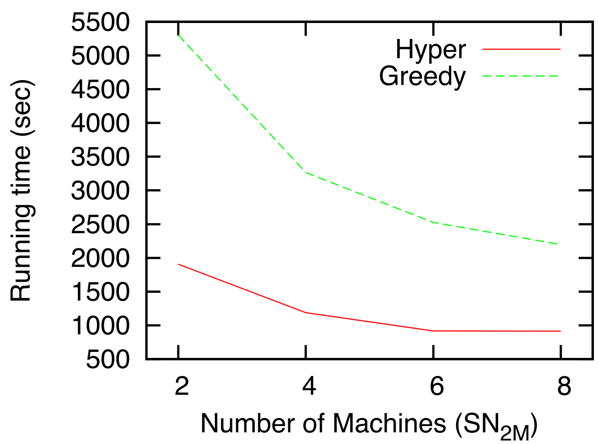
Voter model using commodity machines

Voter Opinion Modeling

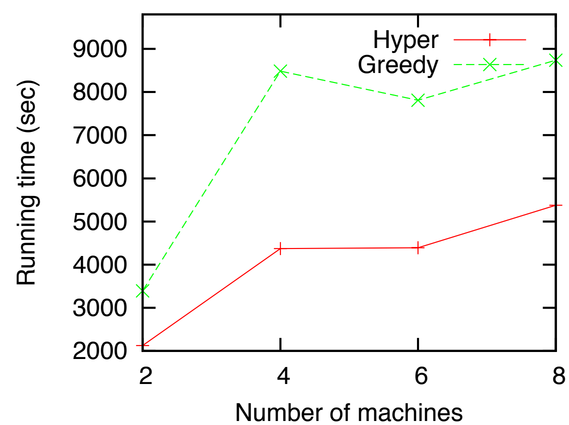


Name	|Subproblem|	|Consensus|	|Edge|	Fit in One Machine?	Run time (sec) |m| = 8
SN_{1M}	3.3M	1.1M	6M	Yes	2230
SN_{2M}	6.6M	2.1M	12M	No	3997
SN_{3M}	10M	3.1M	18M	No	4395
SN_{4M}	13M	4.2M	24M	No	5376

Machine: Intel Core2 Quad CPU 2.66GHz machines with 4GB RAM running Ubuntu 12.04 Linux



Strong scaling with fixed dataset



Weak scaling with increasing size

Weight Learning

Weight Learning

- Learn from training data
- No need to hand-code rule-weights
- Various methods:
  - approximate maximum likelihood

    Broecheler, Mihalkova, Getoor, UAI 2010
  - maximum pseudo-likelihood
  - large-margin estimation

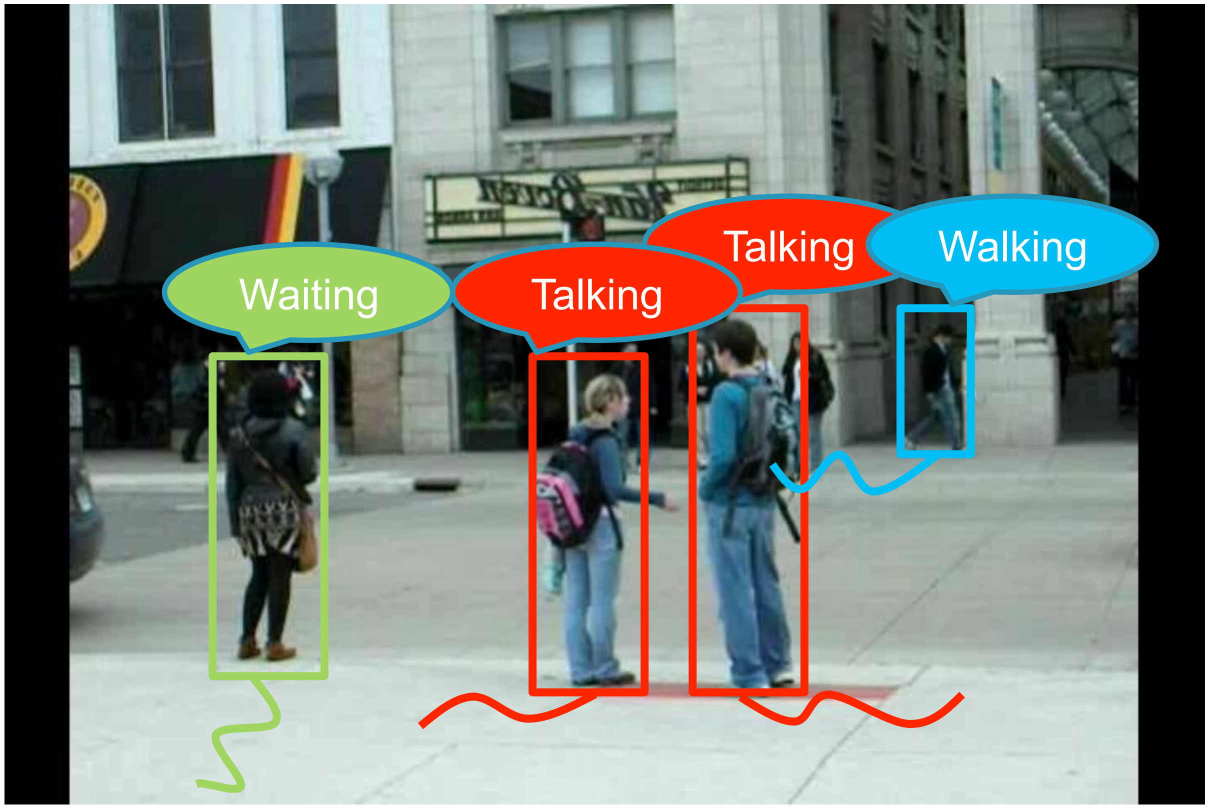
    Bach, Huang, London, UAI 2013

Weight Learning

- State-of-the-art supervised-learning performance on
  - Collective classification
  - Social-trust prediction
  - Preference prediction
  - Image reconstruction

Example PSL Program

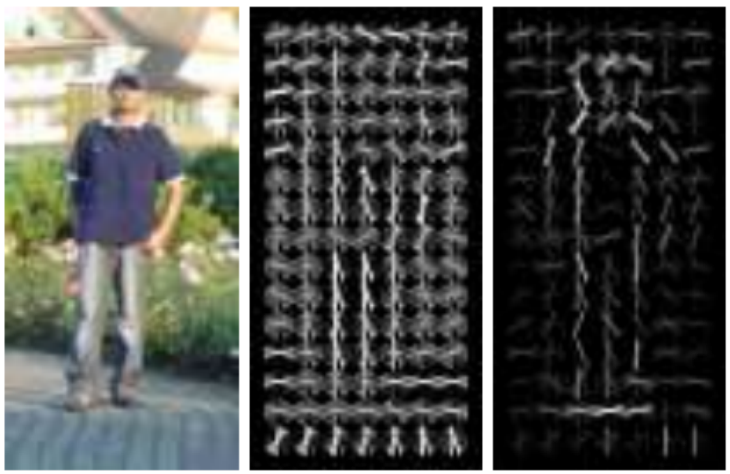
Collective Activity Detection



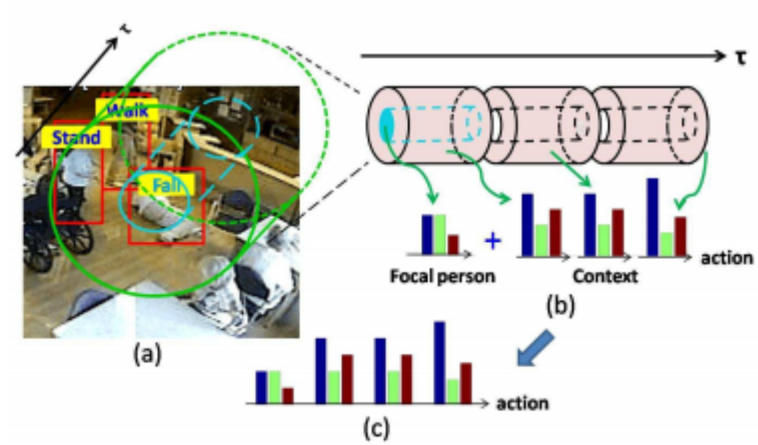
- Objective: Classify actions of individuals in a video sequence
  - Requires tracking the multiple targets, performing ID maintenance

Incorporate Low-level Detectors

Histogram of Oriented Gradients (HOG) [Dalal & Triggs, CVPR 2005]



Action Context Descriptors (ACD) [Lan et al., NIPS 2010]



For each action a, define PSL rule:

$$w_{local,a} : \mathrm{Doing}(X, a) \leftarrow \mathrm{Detector}(X, a)$$

e.g., $w_{local,\textit{walking}} : \mathrm{Doing}(X, \textit{walking}) \leftarrow \mathrm{Detector}(X, \textit{walking})$

Easily Encode Intuitions

- Proximity: People that are close (in frame) are likely doing the same action

  $w_{prox,a}$: Doing(X, a) ← Close(X, Y) ∧ Doing(Y, a)

  - Closeness is measured via a radial basis function

- Continuity: People are likely to continue doing the same action

  $w_{persist,a}$: Doing(Y, a) ← Same (X, Y) ∧ Doing(X, a)

  - Requires tracking & ID maintenance rule:

    w_{id} : Same(X,Y) ← Sequential(X,Y) ∧ Close(X,Y)

Other Rules

- Action transitions
- Frame/scene consistency
- Priors
- (Partial-)Functional Constraints

Collective Activity Detection Model

w_{id} : Same(X, Y) $\leftarrow$ Sequential(X, Y) $\wedge$ Close(X, Y)

$w_{idprior}$: ~SamePerson(X, Y)

For all actions a:

$w_{local,a}$: Doing(X, a) $\leftarrow$ Detector(X, a)

$w_{frame,a}$: Doing(X, a) $\leftarrow$ Frame(X, F) $\wedge$ FrameAction(F, a)

$w_{prox,a}$: Doing(X, a) $\leftarrow$ Close(X, Y) $\wedge$ Doing(Y, a)

$w_{persist,a}$: Doing(Y, a) $\leftarrow$ SamePerson(X, Y) $\wedge$ Doing(X, a)

$w_{prior,a}$: ~Doing(X, a)

[London et al., 2013]

PSL Code

```
/*** MODEL DEFINITION ***/

PSLModel m = new PSLModel(this, data);

/* PREDICATES */

// target
m.add predicate: "doing", types: [ArgumentType.UniqueID,ArgumentType.Integer];
m.add predicate: "sameObj", types: [ArgumentType.UniqueID,ArgumentType.UniqueID];

// observed
m.add predicate: "inFrame", types: [ArgumentType.UniqueID,ArgumentType.Integer,ArgumentType.Integer];
m.add predicate: "inSameFrame", types: [ArgumentType.UniqueID,ArgumentType.UniqueID];
m.add predicate: "inSeqFrames", types: [ArgumentType.UniqueID,ArgumentType.UniqueID];
m.add predicate: "dims", types: [ArgumentType.UniqueID,ArgumentType.Integer,ArgumentType.Integer];
m.add predicate: "detector", types: [ArgumentType.UniqueID,ArgumentType.Integer];
m.add predicate: "frameAction", types: [ArgumentType.Integer,ArgumentType.Integer];

/* FUNCTIONAL PREDICATES */

m.add function: "close", implementation: new ClosenessFunction(0, 1e6, 0.1, true);
m.add function: "seqClose", implementation: new ClosenessFunction(100, 4.0, 0.7, true);
m.add function: "notMoved", implementation: new ClosenessFunction(10, 1.0, 0.0, false);
```

PSL Code

```
/* TRACKING RULES */

// ID maintenance
m.add rule: ( inSeqFrames(BB1,BB2) & dims(BB1,X1,Y1) & dims(BB2,X2,Y2)
                & seqClose(X1,X2,Y1,Y2) ) >> sameObj(BB1,BB2), weight: 1.0;

// Prior on sameObj
m.add rule: ~sameObj(BB1,BB2), weight: 0.01;

/* ACTION RULES */

def actions = ["crossing","standing","queueing","walking","talking"];
for (int a : actions) {

    // Local detectors
    m.add rule: detector(BB,a) >> doing(BB,a), weight: 1.0;

    // Frame consistency
    m.add rule: ( inFrame(BB,S,F) & frameLabel(F,a) ) >> doing(BB,a), weight: 0.1;

    // Persistence
    m.add rule: ( sameObj(BB1,BB2) & doing(BB1,a) ) >> doing(BB2,a), weight: 1.0;

    // Proximity
    m.add rule: ( inSameFrame(BB1,BB2) & doing(BB1,a) & dims(BB1,X1,Y1) & dims(BB2,X2,Y2)
                    & close(X1,X2,Y1,Y2) ) >> doing(BB2,a), weight: 0.1;

    // Prior on doing
    m.add rule: ~doing(BB,a), weight: 0.01;

}
```

PSL Code

```
/* FUNCTIONAL CONSTRAINTS */

// Functional constraint on doing means that it should sum to 1 for each BB
m.add PredicateConstraint.Functional, on: doing;

// (Inverse) Partial functional constraint on sameObj
m.add PredicateConstraint.PartialFunctional, on: sameObj;
m.add PredicateConstraint.PartialInverseFunctional, on: sameObj;
```

[London, et al., CVPR WS 2013]

Results on Activity Recognition

	crossing	waiting	queueing	talking	dancing	jogging
crossing	92.69%	4.30%	2.50%			0.50%
waiting	11.30%	62.80%	24.10%	1.70%		0.10%
queueing	4.20%	17.70%	76.70%	0.80%	0.50%	0.10%
talking	0.60%	6.09%	11.79%	77.22%	0.90%	3.40%
dancing	0.40%		0.30%	1.10%	98.10%	0.10%
jogging			0.10%			99.90%

Recall matrix between different activity types

Accuracy metrics compared against baseline features

Method	5 Activities		6 Activities	
	Acc.	F1	Acc.	F1
HOG	.474	.481	.596	.582
HL-MRF + HOG	**.598**	**.603**	**.793**	**.789**
ACD	.675	.678	.835	.835
HL-MRF + ACD	**.692**	**.693**	**.860**	**.860**

PSL Applications

Sample Applications

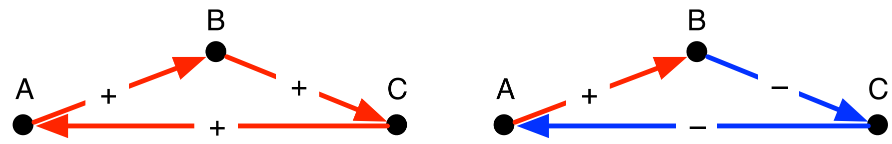
- Social Trust Prediction
- Latent Group Modeling
- Learner Engagement in MOOCs
- Knowledge Graph Identification

Social Trust Prediction

- Competing models from social psychology of strong ties
  - Structural balance [Granovetter ’73]
  - Social status [Cosmides et al., ’92]
- Effects of both models present in online social networks
  - [Leskovec, Huttenlocher, & Kleinberg, 2010]

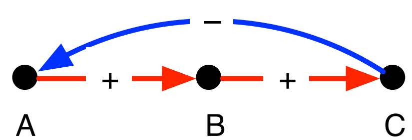
Structural Balance vs. Social Status

- Structural balance: strong ties are governed by tendency toward balanced triads



  - e.g., the enemy of my enemy...

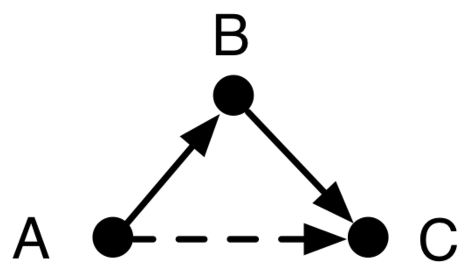
- Social status: strong ties indicate unidirectional respect, "looking up to", expertise status



  - e.g., patient-nurse-doctor, advisor-advisee

[Huang, Kimmig & Getoor, SBP '13]

Structural Balance in PSL

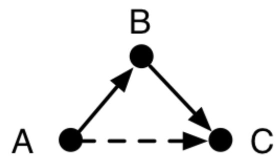


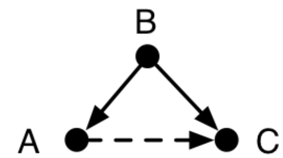
$$\mathrm{Knows}(A,B) \wedge \mathrm{Knows}(B,C) \wedge \mathrm{Knows}(A,C)$$
$$\wedge \mathrm{Trusts}(A,B) \wedge \mathrm{Trusts}(B,C) \Rightarrow \mathrm{Trusts}(A,C),$$

$$\begin{aligned}
\mathrm{Tr}(A,B) \wedge \mathrm{Tr}(B,C) &\Rightarrow \mathrm{Tr}(A,C),\\
\mathrm{Tr}(A,B) \wedge \neg\mathrm{Tr}(B,C) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(A,B) \wedge \mathrm{Tr}(B,C) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(A,B) \wedge \neg\mathrm{Tr}(B,C) &\Rightarrow \mathrm{Tr}(A,C)
\end{aligned}$$

[Huang, Kimmig & Getoor, SBP '13]

Structural Balance in PSL



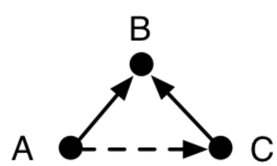


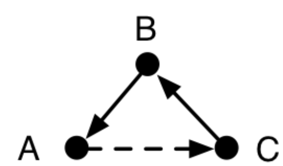
$$\begin{aligned}
\mathrm{Tr}(A,B) \wedge \mathrm{Tr}(B,C) &\Rightarrow \mathrm{Tr}(A,C),\\
\mathrm{Tr}(A,B) \wedge \neg\mathrm{Tr}(B,C) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(A,B) \wedge \mathrm{Tr}(B,C) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(A,B) \wedge \neg\mathrm{Tr}(B,C) &\Rightarrow \mathrm{Tr}(A,C),
\end{aligned}$$

$$\begin{aligned}
\mathrm{Tr}(B,A) \wedge \mathrm{Tr}(B,C) &\Rightarrow \mathrm{Tr}(A,C),\\
\mathrm{Tr}(B,A) \wedge \neg\mathrm{Tr}(B,C) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(B,A) \wedge \mathrm{Tr}(B,C) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(B,A) \wedge \neg\mathrm{Tr}(B,C) &\Rightarrow \mathrm{Tr}(A,C),
\end{aligned}$$

$$\begin{aligned}
\mathrm{Tr}(A,B) \wedge \mathrm{Tr}(C,B) &\Rightarrow \mathrm{Tr}(A,C),\\
\mathrm{Tr}(A,B) \wedge \neg\mathrm{Tr}(C,B) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(A,B) \wedge \mathrm{Tr}(C,B) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(A,B) \wedge \neg\mathrm{Tr}(C,B) &\Rightarrow \mathrm{Tr}(A,C),
\end{aligned}$$

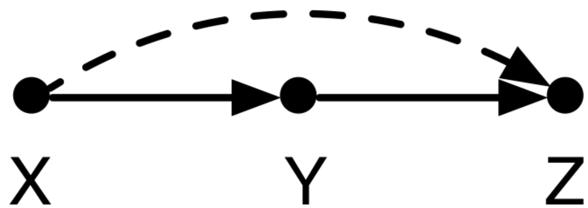
$$\begin{aligned}
\mathrm{Tr}(B,A) \wedge \mathrm{Tr}(C,B) &\Rightarrow \mathrm{Tr}(A,C),\\
\mathrm{Tr}(B,A) \wedge \neg\mathrm{Tr}(C,B) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(B,A) \wedge \mathrm{Tr}(C,B) &\Rightarrow \neg\mathrm{Tr}(A,C),\\
\neg\mathrm{Tr}(B,A) \wedge \neg\mathrm{Tr}(C,B) &\Rightarrow \mathrm{Tr}(A,C)
\end{aligned}$$





[Huang, Kimmig & Getoor, SBP '13]

Social Status in PSL

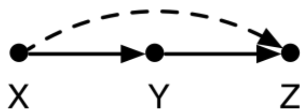


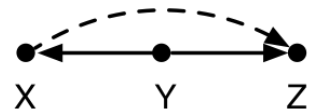
$$\mathrm{Tr}(X, Y) \wedge \mathrm{Tr}(Y, Z) \Rightarrow \mathrm{Tr}(X, Z)$$

$$\neg\mathrm{Tr}(X, Y) \wedge \neg\mathrm{Tr}(Y, Z) \Rightarrow \neg\mathrm{Tr}(X, Z)$$

[Huang, Kimmig & Getoor, SBP ‘13]

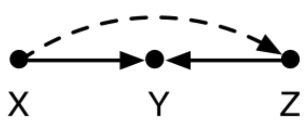
Social Status in PSL

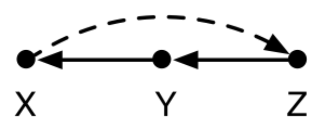




$$
\begin{aligned}
\mathrm{Tr}(X,Y) \wedge \mathrm{Tr}(Y,Z) &\Rightarrow \mathrm{Tr}(X,Z), & \mathrm{Tr}(Y,X) \wedge \neg\mathrm{Tr}(Y,Z) &\Rightarrow \neg\mathrm{Tr}(X,Z),\\
\neg\mathrm{Tr}(X,Y) \wedge \neg\mathrm{Tr}(Y,Z) &\Rightarrow \neg\mathrm{Tr}(X,Z), & \neg\mathrm{Tr}(Y,X) \wedge \mathrm{Tr}(Y,Z) &\Rightarrow \mathrm{Tr}(X,Z),
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{Tr}(X,Y) \wedge \neg\mathrm{Tr}(Z,Y) &\Rightarrow \mathrm{Tr}(X,Z), & \mathrm{Tr}(Y,X) \wedge \mathrm{Tr}(Z,Y) &\Rightarrow \neg\mathrm{Tr}(X,Z),\\
\neg\mathrm{Tr}(X,Y) \wedge \mathrm{Tr}(Z,Y) &\Rightarrow \neg\mathrm{Tr}(X,Z), & \neg\mathrm{Tr}(Y,X) \wedge \neg\mathrm{Tr}(Z,Y) &\Rightarrow \mathrm{Tr}(X,Z)
\end{aligned}
$$





[Huang, Kimmig & Getoor, SBP '13]

Evaluation

- User-user trust ratings from two different online social networks
- Observe some ratings, predict held-out
- Eight-fold cross validation on two data sets:
  - FilmTrust - movie review network, trust ratings from 1-10
  - Epinions - product review network, trust / distrust ratings {-1, 1}

[Huang, Kimmig & Getoor, SBP '13]

Compared Methods

- TidalTrust: graph-based propagation of trust
  - Predict trust via breadth-first search to combine closest known relationships
- EigenTrust: spectral method for trust
  - Predict trustworthiness of nodes based on eigenvalue centrality of weighted trust network
- Average baseline: predict average trust score for all relationships

FilmTrust Experiment

- Normalize [1,10] rating to [0,1]
- Prune network to largest connected-component
- 1,754 users, 2,055 relationships
- Compare mean average error, Spearman's rank coefficient, and Kendall-tau distance

Method	MAE	τ	ρ	MAE*	τ*	ρ*
Average	**0.210**	n/a	n/a	n/a	n/a	n/a
EigenTrust	0.339	−0.054	−0.074	0.339	−0.054	−0.074
TidalTrust	0.229	0.059	0.078	0.236	0.089	0.117
PSL-Balance	**0.207**	**0.136**	**0.176**	**0.193**	**0.235**	**0.314**
PSL-Balance-Recip	**0.207**	**0.139**	**0.188**	**0.193**	**0.241**	**0.318**
PSL-Status	0.224	**0.112**	**0.144**	**0.230**	**0.205**	**0.277**
PSL-Status-Inv	0.224	0.065	0.085	0.238	**0.143**	**0.189**

* measured on only non-default predictions

[Huang, Kimmig & Getoor, SBP '13]

Epinions Experiment

- Snowball sample of 2,000 users from Epinions data set
- 8,675 trust scores normalized to {0,1}
- Measure area under precision-recall curve for distrust edges (rarer class)

Method	AUC
Average	0.070
PSL-Balance	**0.317**
PSL-Balance-Recip	**0.343**
PSL-Status	**0.297**
PSL-Status-Inv	0.280
EigenTrust	0.131
TidalTrust	0.130

[Bach, Huang, Getoor, ICML WS 2013]

Learning Latent Groups

- Can we better understand political discourse in social media by learning groups of similar people?
- Case study: 2012 Venezuelan Presidential Election
  - Incumbent: Hugo Chávez
  - Challenger: Henrique Capriles

Learning Latent Groups

- South American tweets collected from 48-hour window around election.
- Selected 20 top users
  - Candidates, campaigns, media, and most retweeted
- 1,678 regular users interacted with (mentioned or retweeted) at least one top user *and* used at least one hashtag in another tweet
- Those regular users had 8,784 interactions with non-top users

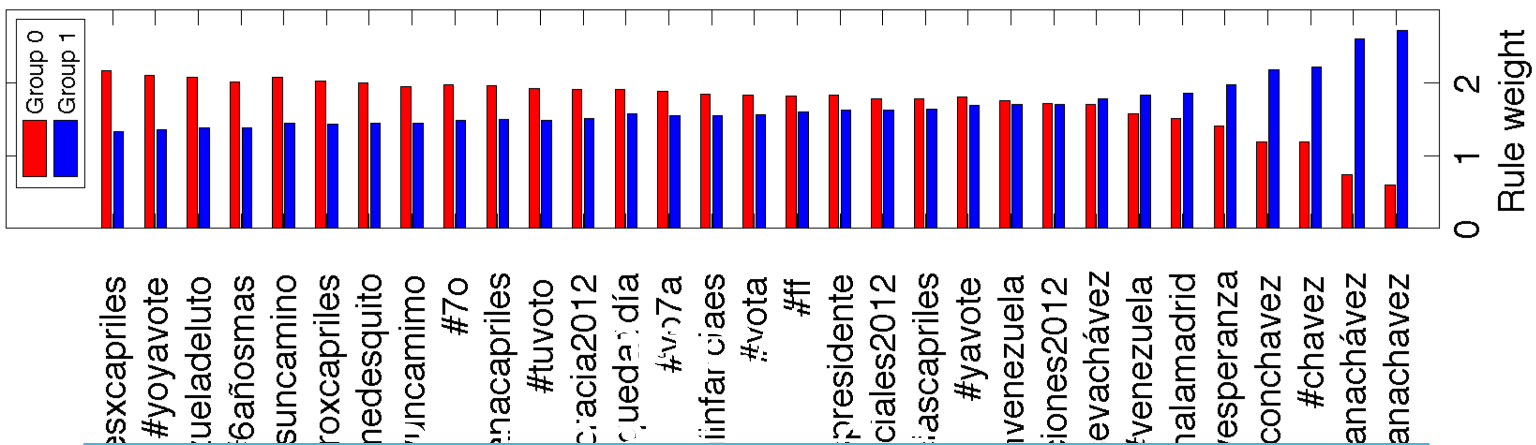
Learning Latent Groups

$$w_{h,g} : \textsc{UsedHashtag}(U, h) \rightarrow \textsc{InGroup}(U, g) \quad \forall h \in H, \forall g \in \mathcal{G}$$

$$\begin{aligned} w_{\text{social}} : &\textsc{RegularUserLink}(U_1, U_3) \\ &\wedge \textsc{RegularUserLink}(U_2, U_3) \wedge U_1 \neq U_2 \\ &\wedge \textsc{InGroup}(U_1, G) \rightarrow \textsc{InGroup}(U_2, G) \end{aligned}$$

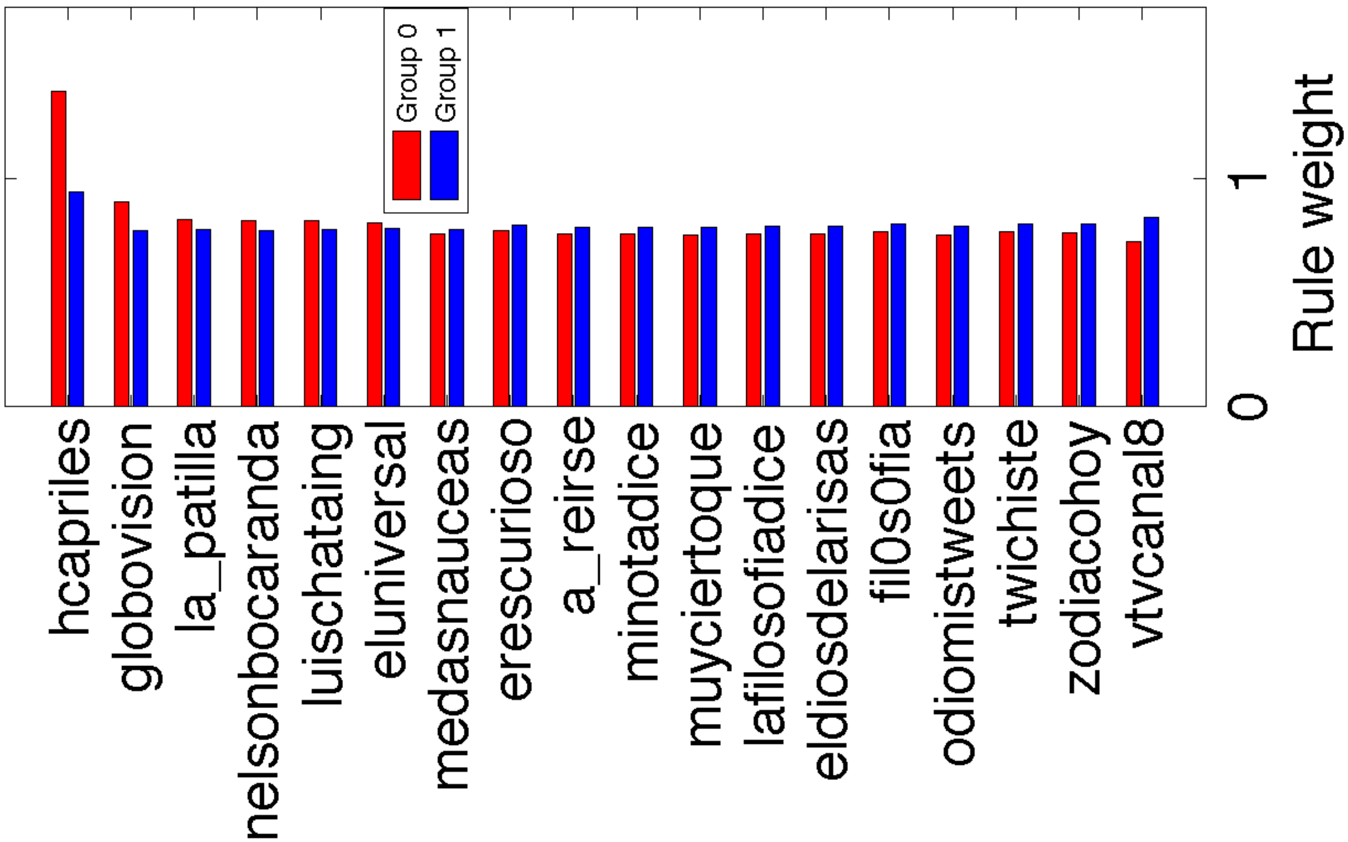
$$w_{g,t} : \textsc{InGroup}(U, g) \rightarrow \textsc{TopUserLink}(U, t) \quad \forall g \in \mathcal{G}, \forall t \in T$$

Learning Latent Groups



‘RT @loriselc: #6AñosMas De pobreza, marginalidad, anarquía, delincuencia, corrupción, división, odio, impunidad, muerte.’
-elchongo

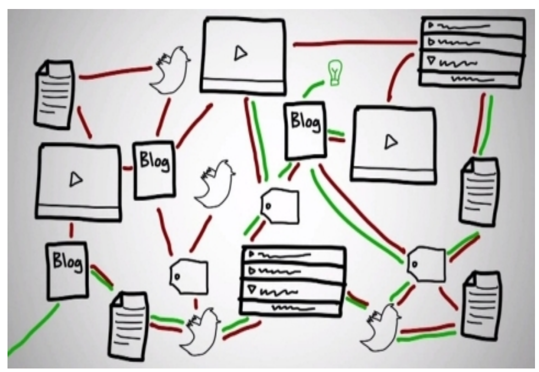
Learning Latent Groups



[Ramesh, Goldwasser, Huang, Daume III, Getoor, NIPS WS 2013]

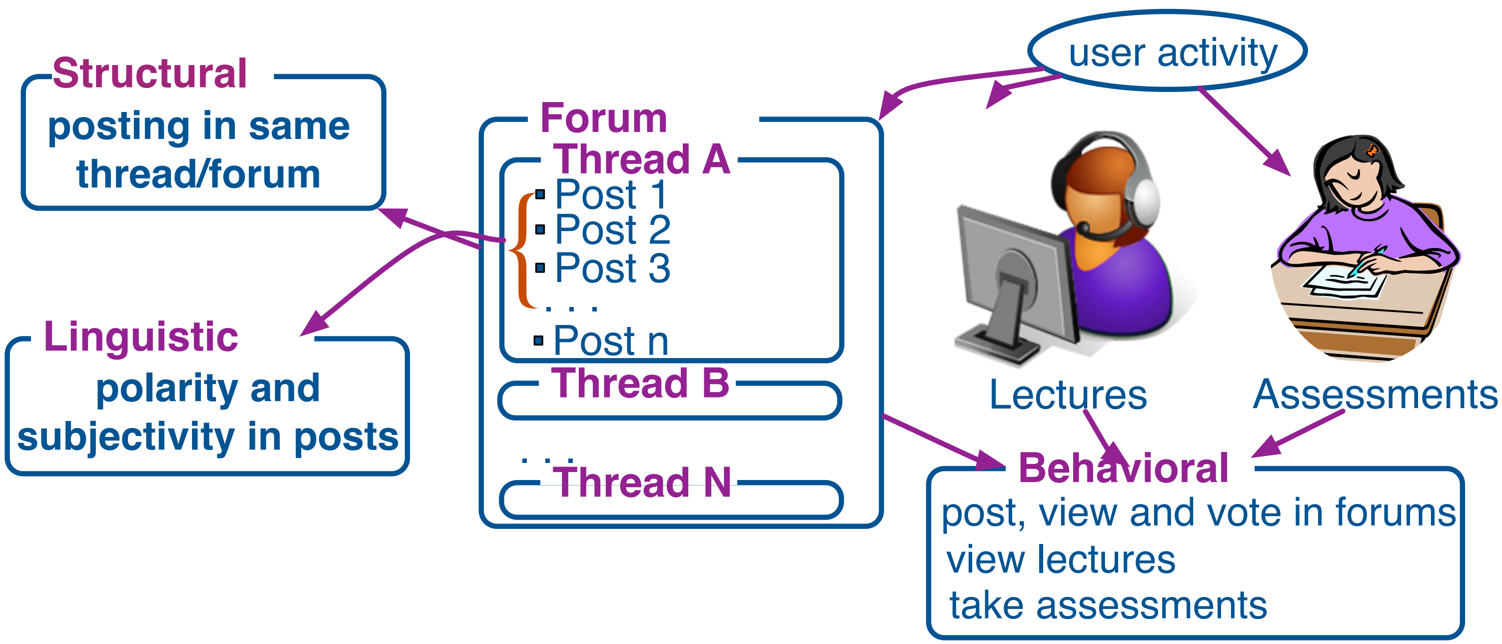
Learner Engagement in MOOCs

- MOOCs boast large number of registrants, but high dropout rate one of the key challenges
- Understanding student engagement essential
  - To understand student activity patterns
  - To suggest interventions to improve learning outcomes, retention and completion



Latent Engagement Model in PSL

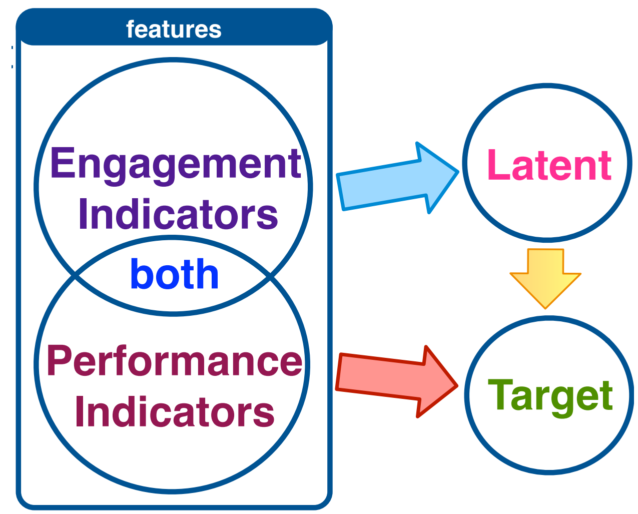
- Leverage behavioral, linguistic, structural and temporal features
- Engagement-types – active and passive as latent variables



PSL Learner Performance Models

- Simple PSL Model
  - Infers **learner performance** from features

- Latent Engagement PSL Model
  - Features grouped into engagement indicators and performance indicators
  - Infers **learner engagement** as hidden variable to predict **learner performance**



PSL Learner Performance Models: Example PSL rules

Simple PSL Model

$$behavioral : \textsc{PostActivity}(U) \wedge \textsc{Reputation}(U) \Rightarrow \textsc{Perf}(U)$$

$$linguistic : \textsc{Posts}(U, P) \wedge \textsc{Positive}(P) \Rightarrow \textsc{Perf}(U)$$

$$structural : \textsc{Posts}(U_1, P_1) \wedge \textsc{Posts}(U_2, P_2) \wedge \textsc{Perf}(U_1) \wedge \textsc{SameThread}(P_1, P_2) \Rightarrow \textsc{Perf}(U_2)$$

$$temporal : \textsc{LastQuiz}(U_1, T_1) \wedge \textsc{LastPost}(U_1, T_1) \wedge \textsc{LastLecture}(U_1, T_1) \Rightarrow \neg\textsc{Perf}(U_1)$$

Latent Engagement PSL Model

$$behavioral : \textsc{PostActivity}(U) \wedge \textsc{SubmitsQuiz}(U) \Rightarrow \textsc{eActive}(U)$$

$$linguistic : \textsc{Posts}(U, P) \wedge \textsc{Positive}(P) \Rightarrow \textsc{eActive}(U)$$

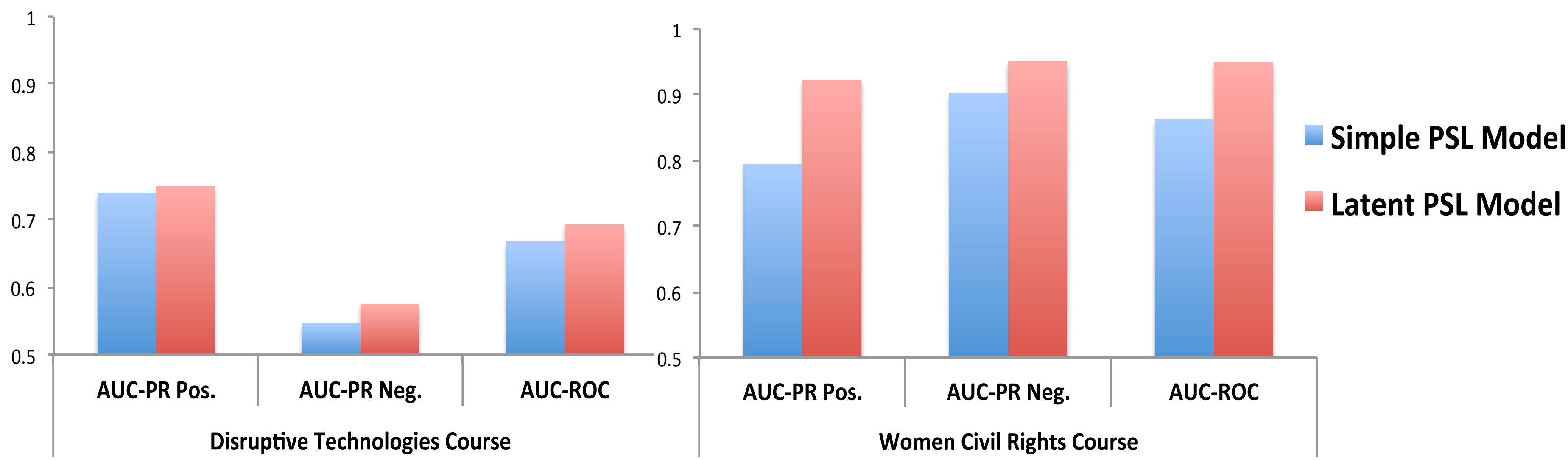
$$structural : \textsc{Posts}(U_1, P_1) \wedge \textsc{Posts}(U_2, P_2) \wedge \textsc{eActive}(U_1) \wedge \textsc{SameThread}(P_1, P_2) \Rightarrow \textsc{eActive}(U_2)$$

$$temporal : \textsc{LastQuiz}(U, T_1) \wedge \textsc{LastLecture}(U, T_1) \wedge \textsc{LastPost}(U, T_1) \Rightarrow \textsc{Disengaged}(U)$$

$$inference : \textsc{eActive}(U) \wedge \textsc{Reputation}(U) \Rightarrow \textsc{Perf}(U)$$

Preliminary Experimental Results

- Modeling latent student engagement helps in predicting student performance



Engagement and sentiment in forum-posts

Engaged learner (positive sentiment)

performance = 0.7508; disengagement = 0.0

"Prof. Lucas, Thank you for a great course! And thank you Coursera!"

Engaged learner (negative sentiment)

performance = 0.8032; disengagement = 0.0

"I have also received a 9, the most disappointing thing is that I have only received good or passing comments from my peers, 3 of 5 did not post any comment about my work."

Disengaged learner (negative sentiment)

performance = 0.5; disengagement = 0.675

"I agree completely. I used a lot of time on my assignment and got 7.5, think the evaluation criteria were wrong, it shouldn't be rated on whether you have 3 or 4 innovations in your description but on a subjective measure."

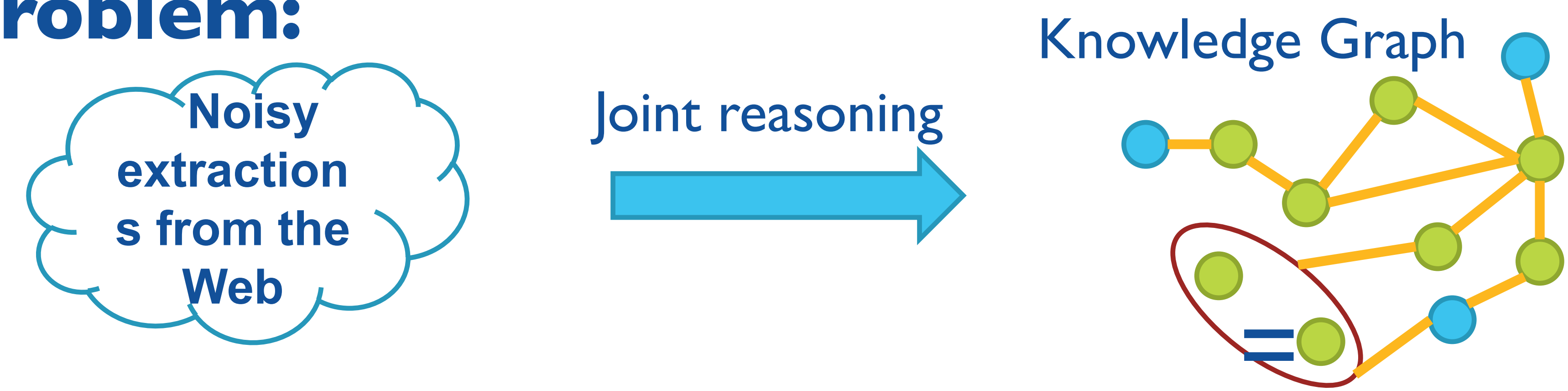
Pujara, Miao, Getoor, Cohen, ISWC 2013
Best Student Paper Award

Knowledge Graph Identification

- **Problem**: Collectively reason about noisy, inter-related fact extractions
- **Task**: NELL fact-promotion (web-scale IE)
  - Millions of extractions, with entity ambiguity and confidence scores
  - Rich ontology: Domain, Range, Inverse, Mutex, Subsumption
- **Goal**: Determine which facts to include in NELL's knowledge base

Knowledge Graph Identification

Problem:



Solution: *Knowledge Graph Identification* (KGI)

- Performs *graph identification*:
  - entity resolution
  - collective classification
  - link prediction
- Enforces *ontological constraints*
- Incorporates *multiple uncertain sources*

Graph Identification in KGI

Noisy Extractions:

$$\textsc{CandRel}_T(E_1, E_2, R) \xrightarrow{w_{CR_T}} \textsc{Rel}(E_1, E_2, R)$$

$$\textsc{CandLbl}_T(E, L) \xrightarrow{w_{CL_T}} \textsc{Lbl}(E, L)$$

Entity Resolution:

$$\textsc{SameEnt}(E_1, E_2) \tilde{\wedge} \textsc{Lbl}(E_1, L) \Rightarrow \textsc{Lbl}(E_2, L)$$

$$\textsc{SameEnt}(E_1, E_2) \tilde{\wedge} \textsc{Rel}(E_1, E, R) \Rightarrow \textsc{Rel}(E_2, E, R)$$

$$\textsc{SameEnt}(E_1, E_2) \tilde{\wedge} \textsc{Rel}(E, E_1, R) \Rightarrow \textsc{Rel}(E, E_2, R)$$

KGI Representation of Ontological Rules

$$\textsc{Dom}(R, L) \,\tilde{\wedge}\, \textsc{Rel}(E_1, E_2, R) \implies \textsc{Lbl}(E_1, L)$$

$$\textsc{Rng}(R, L) \,\tilde{\wedge}\, \textsc{Rel}(E_1, E_2, R) \implies \textsc{Lbl}(E_2, L)$$

$$\textsc{Inv}(R, S) \,\tilde{\wedge}\, \textsc{Rel}(E_1, E_2, R) \implies \textsc{Rel}(E_2, E_1, R)$$

$$\textsc{Sub}(L, P) \,\tilde{\wedge}\, \textsc{Lbl}(E, L) \implies \textsc{Lbl}(E, P)$$

$$R\textsc{Sub}(R, S) \,\tilde{\wedge}\, \textsc{Rel}(E_1, E_2, R) \implies \textsc{Rel}(E_1, E_2, S)$$

$$\textsc{Mut}(L_1, L_2) \,\tilde{\wedge}\, \textsc{Lbl}(E, L_1) \implies \neg\textsc{Lbl}(E, L_2)$$

$$R\textsc{Mut}(R_1, R_2) \,\tilde{\wedge}\, \textsc{Rel}(E_1, E_2, R) \implies \neg\textsc{Rel}(E_1, E_2, R_2)$$

Adapted from Jiang et al., ICDM 2012

Illustration of KGI

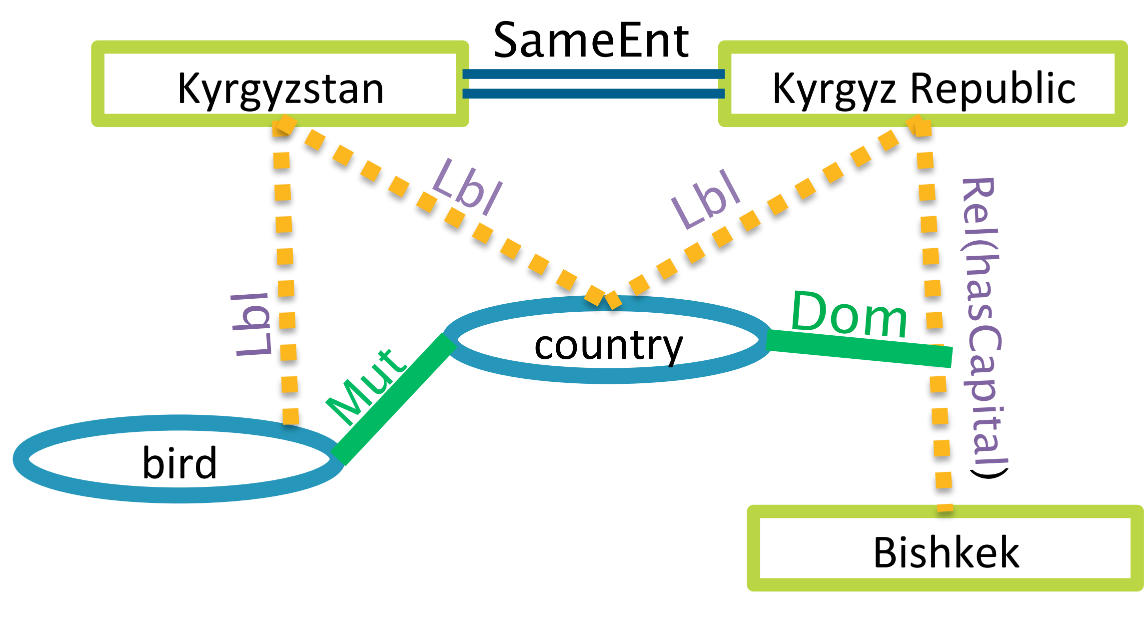
Extractions:
Lbl(Kyrgyzstan, bird)
Lbl(Kyrgyzstan, country)
Lbl(Kyrgyz Republic, country)

Rel(Kyrgyz Republic, Bishkek,
hasCapital)

Ontology:
Dom(hasCapital, country)
Mut(country, bird)

Entity Resolution:
SameEnt(Kyrgyz Republic,
Kyrgyzstan)

Representation as a noisy knowledge graph



After Knowledge Graph Identification

Datasets & Metrics

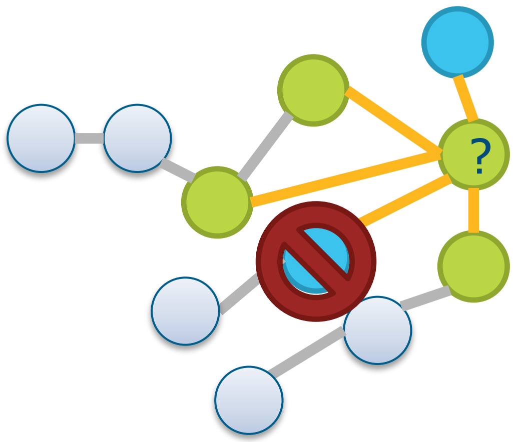
- Data from Never-ending Language Learner (NELL) from iteration 165
- Consists of over 1M extractions and a rich ontology
- Evaluation set from (Jiang, ICDM12) with 4.5K labeled extractions
- Report AUC-PR and running time

Inputs	
Candidate Labels	1.2M
Candidate Relations	100K
Types	
Unique Labels	235
Unique Relations	221
Ontology	
Dom	418
Rng	418
Inv	418
Sub	288
RSub	461
Mut	17.4K
RMut	48.5K

NELL Evaluation: two settings

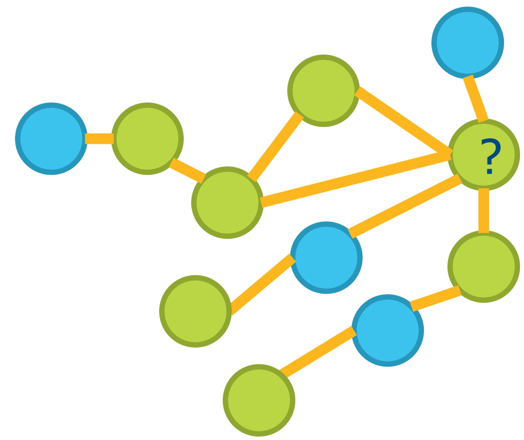
Query Set: restrict to a subset of KG

(Jiang, ICDM12)



- Closed-world model
- Uses a target set, subset of KG
- Derived from 2-hop neighborhood
- Excludes trivially satisfied variables

Complete: Infer full knowledge graph



- Open-world model
- All possible entities, relations, labels
- Inference assigns truth value to each variable

Pujara, Miao, Getoor, Cohen, ISWC 2013

NELL experiments

- **Task:** Use 1.3M NELL extractions and 68K ontology relations to predict a **query set** or build a **complete** knowledge graph
- **Comparisons:** baseline (confidence values), NELL (consistency heuristics), MLN (marginals with MC-SAT), KGI using PSL
- **Performance**: PSL improves F1/AUC; takes just **10 seconds** on query set; builds complete KG (4.3M facts) in **130 minutes**

Method	Query Set		Complete	
	AUC	F1	AUC	F1
Baseline	.873	.828		
NELL	.765	.673	.765	.673
MLN (Jiang, 12)	.899	.836		
PSL-KGI	.904	.853	.892	.848

Additional Application Domains

- Computer Vision
  - Low-level image reconstruction
  - Activity recognition in Video
- Computational Biology & Health Informatics
  - Drug-target prediction
  - Event discovery in EMR data
- Computational Social Science
  - Inferring bias in political discourse
  - Psychological modeling on online social networks
- Information Integration & Extraction
  - Entity resolution
  - Ontology alignment & schema mapping
- Upcoming: climate graphs, discourse analysis, more!

Theory

Theoretical Guarantees?

- Questions:
  - Are there theoretical guarantees for learning templated models (e.g. HL-MRFs)?
  - Which models come with good guarantees?
  - What makes a “good” model?

- These questions are often answered by studying *generalization*

Generalization Bounds

- Learning setting:
  - Learner gets random sample from distribution over structured examples
    - *Possibly gets only one large example!*
  - Learner minimizes *empirical error* on training set

- What is the expected error on future examples?

$$\text{(future error)} \leq \text{(empirical error)} + ?$$

- Analysis of *generalization* gives bounds on future error
  - Typical bound:

$$\text{(future error)} \leq \text{(empirical error)} + \frac{\sqrt{\text{(model complexity)}}}{\sqrt{\text{(size of data)}}}$$

[London, Huang, Taskar & Getoor, ICML'13]

Generalization Bounds

- What is "size of data" for structured data?
  - Traditional learning theory says: # of i.i.d. structured examples
  - But each example is typically very large, relative to # of model parameters
  - Why not (# examples) x (size of example) ?
    - *Careful! Variables are no longer i.i.d.*

- New theory:

$$(\text{future error}) \leq (\text{empirical error}) + \frac{\sqrt{(\text{model complexity})}}{\sqrt{(\,(\#\text{ examples}) \times (\text{size of example})\,)}}$$

[London, Huang, Taskar & Getoor, ICML'13]

Generalization Bounds

- New theory says that generalization can happen from *very few training examples* – *even just one*!
  - Common scenario in structured prediction
- Bounds depend on properties of the model/data:
  - # of parameters
  - *collective stability*: "smoothness" of inference function
  - network structure
  - amount of dependence in distribution
- Gives new insight into when models generalize
  - Example: templated models with strongly convex inference, when data has "weak" dependence

Ongoing Research

- Many open questions!
- Examine generalization of different classes of structured predictors
- Analyze transductive learning setting
  - Data is fixed (i.e., no distribution on future examples)
  - Training data sampled randomly from fixed pool
  - Learned model predicts on remaining data
  - *Very common setting for relational data!*
- Accommodate weaker dependence/structural assumptions

Conclusion

Closing Comments

- Great opportunities to do good work and do useful things in the current era of big data, data analytics, and network science
  – 'entity-oriented data science'
- Statistical relational learning provides some of the tools, much work still needed, developing theoretical bounds for relational learning, scalability, etc.
- Compelling applications abound!

Looking for students & postdocs @UCSC

Probabilistic
Soft
Logic

psl.umiacs.umd.edu