A Scalable Approach for Markov Random Fields over Continuous-Valued Variables

Lise Getoor

University of Maryland, College Park

http://psl.umiacs.umd.edu
Acknowledgements

- **Core Contributors**
  - Stephen Bach
  - Matthias Broecheler
  - Bert Huang

- **Additional Contributors**
  - Shobeir Fakhraei, Angelika Kimming, Stanley Kok, Ben London, Alex Memory, Hui Miao, Lily Mihalkova, Eric Norris, Jay Pujara, Theo Rekatsinas, Arti Ramesh

- **Funding Acknowledgments**
  - Google, IARPA, NSF, Maryland Industrial Partners, Yahoo!

*On job market next year!*
Applications

1. Ontology Alignment
2. Personalized Medicine
3. Social Media
4. Computational Sustainability
5. Massively Online Education
What’s the commonality?

Collective Probabilistic Reasoning in Relational Domains
What’s the commonality?

Collective Probabilistic Reasoning in Relational Domains

Statistical Relational Learning (SRL)

[Getoor & Taskar ’07]
Need for ML* tools which can:

1. Make use of logical structure
2. Handle uncertainty
3. Perform collective inference

* Machine Learning

for this talk

RuleML = Rule Machine Learning, 😊
Alphabet Soup of SRL

PSL?
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
Entity Resolution

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

```
A
  friend
  C, D

B
  friend
  F, G

E
  name
  =

C

D

F

G

H
```

- John Smith
- J. Smith
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

- 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 \emptyset \{B.friends\} \implies 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’’

\[ A \approx B \land B \approx C \implies A \approx C : \infty \]
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

- 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”
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”
Node Labeling
Voter Opinion Modeling

Status update

Tweet
Voter Opinion Modeling
Voter Opinion Modeling

\[\text{vote}(A, P) \land \text{friend}(B, A) \rightarrow \text{vote}(B, P) : 0.3\]

\[\text{vote}(A, P) \land \text{spouse}(B, A) \rightarrow \text{vote}(B, P) : 0.8\]
Multiple Ontologies

- **Organization**
  - **Service & Products**: Software, Hardware, IT Services
  - **Customers**: Developer, Sales Person, Staff
  - **Employees**: work for

- **Products & Services**: Software Dev, Hardware, Consulting
  - **Customer**: helps
  - **Employee**: works for
  - **Company**: develop

- **Customer**: helps
  - **Employee**: sells
Ontology Alignment

Match, Don’t Match?
Ontology Alignment

Service & Products
- Software
- Hardware
- IT Services

Customers
- Developer
- Sales Person
- Staff

Employees
- Organization

Product & Services
- Customer
- Employee
- Technician
- Sales
- Accountant

Software Dev

Similar to what extent?
Logic Foundation
Rules

- Atoms are real valued
  - Interpretation $I$, atom $A$: $I(A) \in [0,1]$
  - We will omit the interpretation and write $A \in [0,1]$

- $\lor$, $\land$ are combination functions
  - $T$-norms: $[0,1]^n \rightarrow [0,1]$
Rules

\[ H_1 \lor \ldots \lor H_m \leftarrow B_1 \land B_2 \land \ldots \land B_n \]

- Combination functions (Lukasiewicz T-norm)
  - \( A \lor B = \min(1, A + B) \)
  - \( A \land B = \max(0, A + B - 1) \)
Satisfaction

\[ H_1 \lor \ldots \lor H_m \leftarrow B_1 \land B_2 \land \ldots \land B_n \]

- Establish Satisfaction
  - \( \lor (H_1, \ldots, H_m) \leftarrow \land (B_1, \ldots, B_n) \)

\[ \geq 0.5 \quad H_1 \leftarrow B_1:0.7 \land B_2:0.8 \]
Distance to Satisfaction

\[ H_1 \lor \ldots \lor H_m \leftarrow B_1 \land B_2 \land \ldots \land B_n \]

- Distance to Satisfaction
  - \[ \max( \land (B_1, \ldots, B_n) - \lor (H_1, \ldots, H_m), 0) \]

<table>
<thead>
<tr>
<th></th>
<th>( H_1:0.7 \leftarrow B_1:0.7 \land B_2:0.8 )</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( H_1:0.2 \leftarrow B_1:0.7 \land B_2:0.8 )</td>
<td>0.3</td>
</tr>
</tbody>
</table>

[Broecheler, et al., UAI ‘10]
Rule Weights

\[ W: H_1 \lor \ldots \lor H_m \leftrightarrow B_1 \land B_2 \land \ldots \land B_n \]

- Weighted Distance to Satisfaction
  \[ d(R, I) = W \cdot \max( \land (B_1, \ldots, B_n) - \lor (H_1, \ldots, H_m), 0) \]
Let’s Review

- 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 Model

Probability density over interpretation $I$

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

Ground rule’s distance to satisfaction

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

Normalization constant

Rule weight

Distance exponent (in $\{1, 2\}$)

Ground rules
Hinge-loss MRFs
Hinge-loss Markov Random Fields

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

- 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:

\[
\arg \max_Y P(Y \mid X) = \arg \min_Y \sum_{j=1}^m w_j \max \{ \ell_j(Y, X), 0 \}^p_j
\]

- 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]
### Speed

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.

<table>
<thead>
<tr>
<th></th>
<th>Cora</th>
<th>Citeseer</th>
<th>Epinions</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete MRF</td>
<td>110.9 s</td>
<td>184.3 s</td>
<td>212.4 s</td>
<td>344.2 s</td>
</tr>
<tr>
<td>HL-MRF</td>
<td>0.4 s</td>
<td>0.7 s</td>
<td>1.2 s</td>
<td>0.6 s</td>
</tr>
</tbody>
</table>

[Bach et al., UAI 2013; London et al., 2013]
Compiling PSL $\rightarrow$ HL-MRF

- Ground out first-order rules
  - Variables: soft-truth values of atoms
  - Hinge-loss potentials: weighted *distances to satisfaction* of ground rules

- $w : A \rightarrow B$
- $w : \neg A \land B$
- $w \times (1 - \min\{1 - A + B, 1\})$
- $w \times \max\{A - B, 0\}$

- The effect is assignments that satisfy weighted rules more are more probable
Inference Meta-Algorithm

Function: MAP-Inference

1.1 \( I_0(y) \leftarrow \text{all zeros assignment} \)
1.2 \( R \leftarrow \text{all grounded rules activated by } I(x) \cup I_0(y) \)
1.3 while \( R \text{ has been updated} \) do
1.4 \( i \leftarrow \text{current iteration} \)
1.5 \( O \leftarrow \text{generateConvexProb}(R) \)
1.6 \( I_i(y) \leftarrow \text{optimize}(O) \)
1.7 foreach Proposition \( y \in y \) do
1.8 \( \text{if } I_i(y) > \theta (\theta = 0.01) \text{ then} \)
1.9 \( R_y \leftarrow \text{activated rules containing } y \) \( R \leftarrow R \cup R_y \)
1.10 end
1.11 end
1.12 end

Each ground rule constitutes a linear or conic constraint, introducing a rule-specific “dissatisfaction” variable that is added to the objective function.
Inference Meta-Algorithm

Function: MAP-Inference

1.1 \( I_0(y) \leftarrow \) all zeros assignment
1.2 \( R \leftarrow \) all grounded rules activated by \( I(x) \cup I_0(y) \)
1.3 while \( R \) has been updated do
1.4 \( i \leftarrow \) current iteration
1.5 \( O \leftarrow \) generateConvexProb\((R)\)
1.6 \( I_i(y) \leftarrow \) optimize\((O)\)
1.7 foreach Proposition \( y \in y \) do
1.8 \[ \text{if } I_i(y) > \theta (\theta = 0.01) \text{ then} \]
1.9 \[ R_y \leftarrow \text{activated rules containing } y \]
1.10 \[ R \leftarrow R \cup R_y \]
1.11 end
1.12 end

Find most probable assignment using consensus optimization (ADMM) subroutine
Inference Meta-Algorithm

**Function**: MAP-Inference

1.1 $I_0(y) \leftarrow$ all zeros assignment
1.2 $R \leftarrow$ all grounded rules activated by $I(x) \cup I_0(y)$
1.3 while $R$ has been updated do
1.4 \hspace{1em} $i \leftarrow$ current iteration
1.5 \hspace{1em} $O \leftarrow$ generateConvexProb($R$)
1.6 \hspace{1em} $I_i(y) \leftarrow$ optimize($O$)
1.7 \hspace{1em} foreach Proposition $y \in y$ do
1.8 \hspace{2em} if $I_i(y) > \theta (\theta = 0.01)$ then
1.9 \hspace{3em} $R_y \leftarrow$ activated rules containing $y$
1.10 \hspace{3em} $R \leftarrow R \cup R_y$
1.11 \hspace{1em} end
1.12 end

Conservative Grounding: Most rules trivially have satisfaction distance=0. Save time and space by not grounding them out in the first place.

Don’t reason about it if you don’t absolutely have to!
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 (1\textsuperscript{st} and 2\textsuperscript{nd} step):
  1. Update Lagrangian multiplier

\[
y^{k+1}_j \leftarrow y^k_j + \rho (x^k_j - X^k_j)
\]

2. Update each sub problem

\[
x^{k+1}_j \leftarrow \arg \min_{x_j} \lambda_j \phi_j (x_j) + \frac{\rho}{2} \left\| x_j - X^k_j + \frac{1}{\rho} y^{k+1}_j \right\|^2
\]

\[
x^{k+1}_j \leftarrow \arg \min_{x_j} I_j [C_j (x_j)] + \frac{\rho}{2} \left\| x_j - X^k_j + \frac{1}{\rho} y^{k+1}_j \right\|^2
\]

3. Update the global variables

\[
z^{k+1}_g \leftarrow \frac{1}{s_g} \sum_{G(i,j)=g} \left( x^{k+1}_j + \frac{y^{k+1}_i}{\rho} \right)_{ij}
\]
Distributed MAP: MapReduce

Pros:
- Straightforward Design

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

Job Bootstrap
- Load global variable $X$ as side data
- Read global variable $X$

Mapper
- Sub problem
- Local variable copy
- $X_1$, $X_m$, $X_{m+1}$, $X_{m+r}$
- $z_1$, $z_2$, $z_q$, $z_p$

Reducer
- Update global component
- $z_1$, $z_2$, $z_q$, $z_p$
- Write new global variable

HDFS or HBase
- Read/write subproblem

Miao, Liu, Getoor, under review
Distributed MAP: GraphLab

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

Miao, Liu, Getoor, under review
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

\begin{align*}
\text{vote}(A, P) \land \text{friend}(B, A) \rightarrow \text{vote}(B, P) : 0.3
\end{align*}

\begin{align*}
\text{vote}(A, P) \land \text{spouse}(B, A) \rightarrow \text{vote}(B, P) : 0.8
\end{align*}

| Name   | |Subproblem| |Consensus| |Edge| |Fit in One Machine? |
|--------|-----------------|-----------------|-----------------|-----------------|-----------------|------------------|
| SN1M   | 3,307,971       | 1,102,498       | 6,011,257       | Yes             |
| SN2M   | 6,656,775       | 2,101,072       | 12,107,131      | No              |
| SN3M   | 9,962,627       | 3,149,103       | 18,113,119      | No              |
| SN4M   | 13,349,751      | 4,203,703       | 24,288,223      | No              |

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

- Learn from training data
- No need to hand-code rule-weights
- Various methods:
  - approximate maximum likelihood
    Bach, et al., UAI '10
  - maximum pseudo-likelihood
  - large-margin estimation
    Broecheler et al., UAI '10
Weight Learning

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

- Implemented in Java / Groovy
- Declarative model definition and imperative model interaction
- Performance oriented
  - Database backend
  - Memory efficient data structures
  - Multi-threaded inference algorithm using ADMM
Probabilistic Soft Logic

System Overview

Input Data

Graph Preprocessing

RDBMS

Input Model

Rules
A ≈ B ← similarID(A.name, B.name)
{A.subClass} ≈ {B.subClass} ← A ≈ B

Constraints
Partial functional: ~

Similarity Functions
similarID(A,B) = new SimFun(){}

Constraints

Input Model

Groovy PSL

Programming Environment

Grounding Framework

Reasoner + Learning

Factor Graph

Analysis & Evaluation Tools

Inference Result

Optimization Toolbox

Similarity Functions
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:

\[
\begin{align*}
w_{\text{local}, a} : & \text{Doing}(X, a) \leftarrow \text{Detector}(X, a) \\
e.g., & \quad w_{\text{local, walking}} : \text{Doing}(X, \text{walking}) \leftarrow \text{Detector}(X, \text{walking})
\end{align*}
\]
Easily Encode Intuitions

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

  \[ w_{\text{prox},a} : \text{Doing}(X, a) \leftarrow \text{Close}(X, Y) \land \text{Doing}(Y, a) \]
  
  - Closeness is measured via a radial basis function

- **Proximity**: People are likely to continue doing the same action

  \[ w_{\text{persist},a} : \text{Doing}(Y, a) \leftarrow \text{Same}(X, Y) \land \text{Doing}(X, a) \]
  
  - Requires tracking & ID maintenance rule:

  \[ w_{\text{id}} : \text{Same}(X,Y) \leftarrow \text{Sequential}(X,Y) \land \text{Close}(X,Y) \]
Other Rules

▪ Action transitions
▪ Frame/scene consistency
▪ Priors
▪ (Partial-)Functional Constraints
Collective Activity Detection Model

\[ w_{id} : \text{Same}(X, Y) \leftarrow \text{Sequential}(X, Y) \land \text{Close}(X, Y) \]

\[ w_{idprior} : \neg\text{SamePerson}(X, Y) \]

For all actions a:

\[ w_{local,a} : \text{Doing}(X, a) \leftarrow \text{Detector}(X, a) \]

\[ w_{frame,a} : \text{Doing}(X, a) \leftarrow \text{Frame}(X, F) \land \text{FrameAction}(F, a) \]

\[ w_{prox,a} : \text{Doing}(X, a) \leftarrow \text{Close}(X, Y) \land \text{Doing}(Y, a) \]

\[ w_{persist,a} : \text{Doing}(Y, a) \leftarrow \text{SamePerson}(X, Y) \land \text{Doing}(X, a) \]

\[ w_{prior,a} : \neg\text{Doing}(X, a) \]

[London et al., 2013]
PSL Code

```java
/** 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);
```
/* TRACKING RULES */

// ID maintenance
m.add rule: ( inSegFrames(BB1, BB2) & dims(BB1, X1, Y1) & dims(BB2, X2, Y2) & segClose(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;
}
/* 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;
Foundations Summary
Foundations Summary

- **Design** probabilistic models using declarative language
  - Syntax based on **first-order logic**
- **Inference** of most-probable explanation is fast **convex optimization** (ADMM)
- **Learning** algorithms for training rule weights from labeled data
PSL Applications
### Document Classification

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Citesee</th>
<th>Cora</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL-MRF-Q (MLE)</td>
<td>0.729</td>
<td>0.816</td>
</tr>
<tr>
<td>HL-MRF-Q (MPLE)</td>
<td>0.729</td>
<td>0.818</td>
</tr>
<tr>
<td>HL-MRF-Q (LME)</td>
<td>0.683</td>
<td>0.789</td>
</tr>
<tr>
<td>HL-MRF-L (MLE)</td>
<td>0.724</td>
<td>0.802</td>
</tr>
<tr>
<td>HL-MRF-L (MPLE)</td>
<td>0.729</td>
<td>0.808</td>
</tr>
<tr>
<td>HL-MRF-L (LME)</td>
<td>0.695</td>
<td>0.789</td>
</tr>
<tr>
<td>MLN (MLE)</td>
<td>0.686</td>
<td>0.756</td>
</tr>
<tr>
<td>MLN (MPLE)</td>
<td>0.715</td>
<td>0.797</td>
</tr>
<tr>
<td>MLN (LME)</td>
<td>0.687</td>
<td>0.783</td>
</tr>
</tbody>
</table>

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.
Computer Vision Applications

- **Low-level vision:**
  - image reconstruction

- **High-level vision:**
  - activity recognition in videos
**Image Reconstruction**

Table 5: Mean squared errors per pixel for image reconstruction. HL-MRFs produce the most accurate reconstructions on the Caltech101 and the left-half Olivetti faces, and only sum-product networks produce better reconstructions on Olivetti bottom-half faces. Scores for other methods are taken from Poon and Domingos [18].

<table>
<thead>
<tr>
<th></th>
<th>HL-MRF-Q (MLE)</th>
<th>SPN</th>
<th>DBM</th>
<th>DBN</th>
<th>PCA</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caltech-Left</td>
<td>1751</td>
<td>1815</td>
<td>2998</td>
<td>4960</td>
<td>2851</td>
<td>2327</td>
</tr>
<tr>
<td>Caltech-Bottom</td>
<td>1863</td>
<td>1924</td>
<td>2656</td>
<td>3447</td>
<td>1944</td>
<td>2575</td>
</tr>
<tr>
<td>Olivetti-Left</td>
<td>932</td>
<td>942</td>
<td>1866</td>
<td>2386</td>
<td>1076</td>
<td>1527</td>
</tr>
<tr>
<td>Olivetti-Bottom</td>
<td>1202</td>
<td>918</td>
<td>2401</td>
<td>1931</td>
<td>1265</td>
<td>1793</td>
</tr>
</tbody>
</table>

Figure 1: Example results on image reconstruction of Caltech101 (left) and Olivetti (right) faces. From left to right in each column: (1) true face, left side predictions by (2) HL-MRFs and (3) SPNs, and bottom half predictions by (4) HL-MRFs and (5) SPNs. SPN reconstructions are downloaded from Poon and Domingos [18].

Table 5. HL-MRFs produce the best mean squared error on the left- and bottom-half settings for the Caltech101 set and the left-half setting in the Olivetti set. Only sum product networks produce lower error on the Olivetti bottom-half faces. Some reconstructed faces are displayed in Figure 1, where that the shallow, pixel-based HL-MRFs produce comparably convincing images to sum-product networks, especially in the left-half setting, where HL-MRF can learn which pixels are likely to mimic their horizontal mirror. While neither method is particularly good at reconstructing the bottom half of faces, the quantitative difference between the deep SPN and the shallow HL-MRF reconstructions is that SPNs seem to hallucinate different faces, often with some artifacts, while HL-MRFs predict blurry shapes roughly the same pixel intensity as the observed, top half of the face. The tendency to better match pixel intensity helps HL-MRFs score better quantitatively on the Caltech101 faces, where the lighting conditions are more varied than in Olivetti.

Training and predicting with these pixel-based HL-MRFs takes little time. In our experiments, training takes about 1.5 hours on a 24-core machine, while predicting takes about a second per image. While Poon and Domingos [18] report faster training with SPNs, both HL-MRFs and SPNs clearly belong to a class of faster models when compared to DBNs and DBMs, which can take days to train on modern hardware.

6 CONCLUSION

We have shown that HL-MRFs are a flexible and interpretable class of models, capable of modeling a wide variety of domains. HL-MRFs admit fast, convex inference, because their density functions are log-concave. The MPE inference algorithm we introduce is applicable to the full class of HL-MRFs. With this fast, general algorithm, we are the first to show results using quadratic HL-MRFs on real-world data. In our experiments, HL-MRFs match or exceed the predictive performance of state-of-the-art methods on four diverse tasks. The natural mapping between hinge-loss potentials and logic rules makes HL-MRFs easy to define and interpret.

---

[Bach, et al., UAI 2013]
Activity Recognition in Videos

[London, et al., CVPR WS 2013]
Results on Activity Recognition

Recall matrix between different activity types

Accuracy metrics compared against baseline features

<table>
<thead>
<tr>
<th>Method</th>
<th>5 Activities</th>
<th></th>
<th>6 Activities</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc.</td>
<td>F1</td>
<td>Acc.</td>
<td>F1</td>
</tr>
<tr>
<td>HOG</td>
<td>.474</td>
<td>.481</td>
<td>.596</td>
<td>.582</td>
</tr>
<tr>
<td>HL-MRF + HOG</td>
<td>.598</td>
<td>.603</td>
<td>.793</td>
<td>.789</td>
</tr>
<tr>
<td>ACD</td>
<td>.675</td>
<td>.678</td>
<td>.835</td>
<td>.835</td>
</tr>
<tr>
<td>HL-MRF + ACD</td>
<td>.692</td>
<td>.693</td>
<td>.860</td>
<td>.860</td>
</tr>
</tbody>
</table>
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
Structural Balance in PSL

Knows(A, B) \land Knows(B, C) \land Knows(A, C)
\land Trusts(A, B) \land Trusts(B, C) \Rightarrow Trusts(A, C),

Tr(A, B) \land Tr(B, C) \Rightarrow Tr(A, C),
Tr(A, B) \land \neg Tr(B, C) \Rightarrow \neg Tr(A, C),
\neg Tr(A, B) \land Tr(B, C) \Rightarrow \neg Tr(A, C),
\neg Tr(A, B) \land \neg Tr(B, C) \Rightarrow Tr(A, C)
Structural Balance in PSL

\[
\begin{align*}
\text{Tr}(A, B) \land \text{Tr}(B, C) & \implies \text{Tr}(A, C), \\
\text{Tr}(A, B) \land \neg \text{Tr}(B, C) & \implies \neg \text{Tr}(A, C), \\
\neg \text{Tr}(A, B) \land \text{Tr}(B, C) & \implies \neg \text{Tr}(A, C), \\
\neg \text{Tr}(A, B) \land \neg \text{Tr}(B, C) & \implies \text{Tr}(A, C), \\
\text{Tr}(A, B) \land \text{Tr}(C, B) & \implies \text{Tr}(A, C), \\
\text{Tr}(A, B) \land \neg \text{Tr}(C, B) & \implies \neg \text{Tr}(A, C), \\
\neg \text{Tr}(A, B) \land \text{Tr}(C, B) & \implies \neg \text{Tr}(A, C), \\
\neg \text{Tr}(A, B) \land \neg \text{Tr}(C, B) & \implies \text{Tr}(A, C),
\end{align*}
\]
Social Status in PSL

\[
\text{Tr}(X, Y) \land \text{Tr}(Y, Z) \Rightarrow \text{Tr}(X, Z) \\
\neg \text{Tr}(X, Y) \land \neg \text{Tr}(Y, Z) \Rightarrow \neg \text{Tr}(X, Z)
\]
Social Status in PSL

\[ \text{Tr}(X, Y) \land \text{Tr}(Y, Z) \Rightarrow \text{Tr}(X, Z), \]
\[ \neg \text{Tr}(X, Y) \land \neg \text{Tr}(Y, Z) \Rightarrow \neg \text{Tr}(X, Z), \]
\[ \text{Tr}(X, Y) \land \neg \text{Tr}(Z, Y) \Rightarrow \text{Tr}(X, Z), \]
\[ \neg \text{Tr}(X, Y) \land \text{Tr}(Z, Y) \Rightarrow \neg \text{Tr}(X, Z), \]

\[ \text{Tr}(Y, X) \land \neg \text{Tr}(Y, Z) \Rightarrow \neg \text{Tr}(X, Z), \]
\[ \neg \text{Tr}(Y, X) \land \text{Tr}(Y, Z) \Rightarrow \text{Tr}(X, Z), \]
\[ \text{Tr}(Y, X) \land \text{Tr}(Z, Y) \Rightarrow \neg \text{Tr}(X, Z), \]
\[ \neg \text{Tr}(Y, X) \land \neg \text{Tr}(Z, Y) \Rightarrow \text{Tr}(X, Z) \]
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}
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

[Huang, et al., SBP ‘13]
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

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>$\tau$</th>
<th>$\rho$</th>
<th>MAE*</th>
<th>$\tau^*$</th>
<th>$\rho^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td><strong>0.210</strong></td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>EigenTrust</td>
<td>0.339</td>
<td>-0.054</td>
<td>-0.074</td>
<td>0.339</td>
<td>-0.054</td>
<td>-0.074</td>
</tr>
<tr>
<td>TidalTrust</td>
<td>0.229</td>
<td>0.059</td>
<td>0.078</td>
<td>0.236</td>
<td>0.089</td>
<td>0.117</td>
</tr>
<tr>
<td>PSL-Balance</td>
<td><strong>0.207</strong></td>
<td>0.136</td>
<td>0.176</td>
<td>0.193</td>
<td>0.235</td>
<td>0.314</td>
</tr>
<tr>
<td>PSL-Balance-Recip</td>
<td><strong>0.207</strong></td>
<td>0.139</td>
<td>0.188</td>
<td>0.193</td>
<td>0.241</td>
<td>0.318</td>
</tr>
<tr>
<td>PSL-Status</td>
<td>0.224</td>
<td><strong>0.112</strong></td>
<td>0.144</td>
<td>0.230</td>
<td>0.205</td>
<td>0.277</td>
</tr>
<tr>
<td>PSL-Status-Inv</td>
<td>0.224</td>
<td>0.065</td>
<td>0.085</td>
<td>0.238</td>
<td><strong>0.143</strong></td>
<td>0.189</td>
</tr>
</tbody>
</table>

* measured on only non-default predictions

[Huang, et al., 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)

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>0.070</td>
</tr>
<tr>
<td>PSL-Balance</td>
<td>0.317</td>
</tr>
<tr>
<td>PSL-Balance-Recip</td>
<td>0.343</td>
</tr>
<tr>
<td>PSL-Status</td>
<td>0.297</td>
</tr>
<tr>
<td>PSL-Status-Inv</td>
<td>0.280</td>
</tr>
<tr>
<td>EigenTrust</td>
<td>0.131</td>
</tr>
<tr>
<td>TidalTrust</td>
<td>0.130</td>
</tr>
</tbody>
</table>
Drug-Target Interaction Prediction

- New drugs take a decade to reach market.
- Development cost reaches 2 billion US dollars.
- Most novel drug candidates never get approved.

Drug repurposing: Finding new uses for approved drugs

[Fakhraei, et al., BioKDD’13]
Drug-Target Interaction Prediction

Computational predictions focus biological investigations

**Data:** drug-target (gene product) interaction network + drug-drug and target-target similarities

**Task:** link prediction
Drug-Target Interaction Prediction

\[
\text{SimilarTarget}_\beta(T_1, T_2) \land \text{Interacts}(D, T_2) \rightarrow \text{Interacts}(D, T_1)
\]

\[
\text{SimilarDrug}_\alpha(D_1, D_2) \land \text{Interacts}(D_2, T) \rightarrow \text{Interacts}(D_1, T)
\]

\[
\text{SimilarDrug}_\alpha(D_1, D_2) \land \text{SimilarTarget}_\beta(T_1, T_2) \land \text{Interacts}(D_2, T_2) \rightarrow \text{Interacts}(D_1, T_1)
\]
Drug-Target Interaction Prediction

- 315 Drugs, 250 Targets
- 78,750 possible interactions, 1,306 observed interactions
- 5 drug-drug similarities, 3 target-target similarities

<table>
<thead>
<tr>
<th>Method</th>
<th>AUROC</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSL</td>
<td>0.931 ± 0.018</td>
<td>10-fold CV</td>
</tr>
<tr>
<td>Perlman, et al. 2011</td>
<td>0.935</td>
<td></td>
</tr>
<tr>
<td>Yamanishi, et al. 2008</td>
<td>0.884</td>
<td>with sampling</td>
</tr>
<tr>
<td>Bleakley, et al. 2009</td>
<td>0.814</td>
<td></td>
</tr>
</tbody>
</table>

![Graph showing precision vs. Top N Predictions with and without weight learning.](Fakhraei, et al., BioKDD’13)
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

Left: This photograph was produced by Agência Brasil, a public Brazilian news agency. This file is licensed under the Creative Commons Attribution 3.0 Brazil license. Right: This photograph was produced by Wilfredor. This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license.
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 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

\[ w_{h,g} : \text{UsedHashtag}(U, h) \rightarrow \text{InGroup}(U, g) \]
\[ \forall h \in H, \forall g \in G \]

\[ w_{\text{social}} : \text{RegularUserLink}(U_1, U_3) \]
\[ \land \text{RegularUserLink}(U_2, U_3) \land U_1 \neq U_2 \]
\[ \land \text{InGroup}(U_1, G) \rightarrow \text{InGroup}(U_2, G) \]

\[ w_{g,t} : \text{InGroup}(U, g) \rightarrow \text{TopUserLink}(U, t) \]
\[ \forall g \in G, \forall t \in T \]
Learning Latent Groups

Algorithm  Hard Expectation Maximization

Input: model \( P(Y, Z|X; \lambda) \), initial parameters \( \lambda^0 \)

\[ t \leftarrow 1 \]

while not converged do

\[ Z^t = \operatorname{arg\,max}_Z P(Z|Y, X; \lambda^{t-1}) \]

\[ \lambda^t = \operatorname{arg\,max}_\lambda P(Y, Z^t|X; \lambda) \]

\[ t \leftarrow t + 1 \]

end while
Learning Latent Groups
Learning Latent Groups
Schema Matching

- Correspondences between source and target schemas
- Matching rules
  - “If two concepts are the same, they should have similar subconcepts’’
  - “If the domains of two attributes are similar, they may be the same’’

\[
\begin{align*}
\text{develop}(A, B) & \leq \text{provides}(A, B) \\
\text{Company}(A) & \leq \text{Organization}(A) \\
\text{Products} & \leq \text{Service} \\
\text{Services}(B) & \leq \text{Service} & \text{Products}(B)
\end{align*}
\]
Schema Mapping

- Input: Schema matches
- Output: S-T query pairs (TGD) for exchange or mediation
- Mapping rules
  - “Every matched attribute should participate in some TGD.”
  - “The solutions to the queries in TGDs should be similar.”

∃Portfolio P, develop(A, P) ∧ includes(P, B) <= provides(A, B) . . .
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:**
- Noisy extractions from the Web

**Solution:** *Knowledge Graph Identification (KGI)*
- Performs *graph identification*:
  - entity resolution
  - collective classification
  - link prediction
- Enforces *ontological constraints*
- Incorporates *multiple uncertain sources*

Joint reasoning leads to Knowledge Graph
Graph Identification in KGI

Noisy Extractions:

\[ \text{CANDREL}_T(E_1, E_2, R) \xrightarrow{\mathcal{W}_{CRT}} \text{REL}(E_1, E_2, R) \]
\[ \text{CANDLBL}_T(E, L) \xrightarrow{\mathcal{W}_{CLT}} \text{LBL}(E, L) \]
\[ \text{SAMEENT}(E_1, E_2) \not\sim \text{LBL}(E_1, L) \Rightarrow \text{LBL}(E_2, L) \]
\[ \text{SAMEENT}(E_1, E_2) \not\sim \text{REL}(E_1, E, R) \Rightarrow \text{REL}(E_2, E, R) \]
\[ \text{SAMEENT}(E_1, E_2) \not\sim \text{REL}(E, E_1, R) \Rightarrow \text{REL}(E, E_2, R) \]
KGI Representation of Ontological Rules

\[ \text{DOM}(R, L) \sim \text{REL}(E_1, E_2, R) \implies \text{LBL}(E_1, L) \]

\[ \text{RNG}(R, L) \sim \text{REL}(E_1, E_2, R) \implies \text{LBL}(E_2, L) \]

\[ \text{INV}(R, S) \sim \text{REL}(E_1, E_2, R) \implies \text{REL}(E_2, E_1, R) \]

\[ \text{SUB}(L, P) \sim \text{LBL}(E, L) \implies \text{LBL}(E, P) \]

\[ \text{RSUB}(R, S) \sim \text{REL}(E_1, E_2, R) \implies \text{REL}(E_1, E_2, S) \]

\[ \text{MUT}(L_1, L_2) \sim \text{LBL}(E, L_1) \implies \neg \text{LBL}(E, L_2) \]

\[ \text{RMT}(R_1, R_2) \sim \text{REL}(E_1, E_2, R) \implies \neg \text{REL}(E_1, E_2, R_2) \]

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

Kyrgyzstan

Rel(hasCapital)

Bishkek

Kyrgyz Republic

country

Lbl

bird

Mut

Dom

SameEnt
Datasets & Results

- Evaluation on NELL dataset from iteration 165:
  - 1.7M candidate facts
  - 70K ontological constraints
- Predictions on 25K facts from a 2-hop neighborhood around test data
- Beats other methods, runs in just 10 seconds!
- Also supports lazy inference of complete knowledge graph (100 minutes)

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>.828</td>
<td>.873</td>
</tr>
<tr>
<td>NELL</td>
<td>.673</td>
<td>.765</td>
</tr>
<tr>
<td>MLN (Jiang, 12)</td>
<td>.836</td>
<td>.899</td>
</tr>
<tr>
<td>KGI-PSL</td>
<td>.853</td>
<td>.904</td>
</tr>
</tbody>
</table>
Conclusion
Conclusion

- **PSL:**
  - expressive, declarative framework for structured machine learning problems
  - scalable

- Much ongoing work, including incorporating latent variables, structure learning, distributing

- Interested in applying it to a variety of domains

- Encourage you to try it!

http://psl.umiacs.umd.edu

Looking for students & postdocs!