Probabilistic Soft Logic:  
A New Framework for Statistical Relational Learning

Stephen H. Bach  
Lise Getoor, Bert Huang, Matthias Broecheler

Additional Contributors: Shobeir Fakhraei, Angelika Kimming, Ben London, Alex Memory, Hui Miao, Lily Mihalkova, Eric Norris, Jay Pujara, Theo Rekatsinas, Arti Ramesh
Class Outline

- Relational Data and Statistical Relational Learning
- Probabilistic Soft Logic
  - Motivation
  - Foundations
  - Inference
  - Learning
  - Applications
- Hands-on Demonstration
Relational Data and Statistical Relational Learning
Machine Learning 101

- Classic machine learning tasks:
  - Classification
  - Regression
  - Clustering
Machine Learning 101

- Algorithms generally designed for data that are
  - Independent
  - Identically distributed

Knowing my label doesn’t tell you anything about any other labels!

I came from the same distribution as all the other points!
A lot of real world data are NOT I.I.D.!

Sometimes machine learning tries to predict small, independent structures:

Even this isn’t how the real world works!
Relational Data

- Entities
- Relations

Also called:
- Structured data
- Network data
- Multi-relational data
- Heterogeneous networks
Relational Data

- Social Networks
Relational Data

- The Web
Relational Data

- Biological Networks

http://www.eecs.tufts.edu/~kdorosch/
(Implicitly) Relational Data

- Natural Language

(Implicitly) Relational Data

- Video

- crossing
- waiting
- queueing
- walking
- talking
- dancing
- jogging
Learning probabilistic models of these complex, relational data is the field of **statistical relational learning**.
Relational Dependencies

- Some joint configurations are more probable than others
Relational Dependencies

- Some joint configurations are more probable than others
Statistical Relational Learning

- Model relational dependencies
- Learn to predict unknowns jointly

How do we build statistical relational models?
Generative Models

- Score in terms of conditional probability distributions

Examples:
- Probabilistic Relational Models (Friedman et al., 1999)
- BLOG (Milch et al., 2005)
- Church (Goodman et al., 2008)
- Relational Topic Models (Chang and Blei, 2009)
Generative Models

- Nice generative story
- Relatively easy learning
- Careful construction required
- Model both the observations and the unknowns
Discriminative Models

- Score over unknowns only
- Discriminant function
  \[ f(Y, X) = \sum_{j=1}^{m} w_j \phi_j(Y, X) \]
- Common probabilistic interpretation
  \[ p(Y|X) = \frac{1}{Z(w, X)} \exp \left[ -f(Y, X) \right] \]

Examples:
- Max-margin Markov Nets (Taskar et al., 2003)
- Structural SVMs (Tsochantaridis et al., 2005)
- Markov Logic Networks (Richardson and Domingos, 2006)
Discriminative Models

- Easy to define

- One-size-fits-all* algorithms

- Can suffer from limited scalability
Probabilistic Soft Logic
Probabilistic Soft Logic (PSL)

**Declarative language** based on logic to express statistical relational learning problems

- **Predicate**: relationship or property  
  e.g., Friends(A, B)
- **Atom**: (continuous) random variable  
  e.g., Friends(Steve, Jay) = ?
- **Rule**: capture dependency or constraint  
  e.g., $3.0 : \text{Friends}(A, B) \& \text{Friends}(B, C) \Rightarrow \text{Friends}(A, C)$
- **Set**: define aggregates  
  e.g., $\text{Average}[\text{Friends}(Steve, X)]$

**PSL Program** = Rules + Input DB
Probabilistic Soft Logic (PSL)

Declarative language based on logic to express statistical relational learning problems

- Predicate: relationship or property
e.g., Friends(A, B)

- Atom: (continuous) random variable
e.g., Friends(Steve, Jay) = ?

- Rule: capture dependency or constraint
e.g., 3.0 : Friends(A, B) & Friends(B, C) ⇒ Friends(A, C)

- Set: define aggregates
e.g., Average[Friends(Steve, X)]

PSL Program = Rules + Input DB
Entity Resolution

- **Entities**
  - People References

- **Attributes**
  - Name

- **Relationships**
  - Friendship

- **Goal**: Identify references that denote the same person

Diagram:

- Node A labeled John Smith
  - Child C
    - Child E
  - Child D
  - Friend B labeled J. Smith
    - Child F
      - Child H
    - Friend E

Relations:
- A and B are equal
- C and F are equal
- D and E are equal
- H
- E
Entity Resolution

- **Use rules to express relational dependencies**
  - “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

- Use rules to express relational dependencies
  - “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.”

2.0 : Name(A, NameA) & Name(B, NameB) & Similar(NameA, NameB) => SamePerson(A, B)
Entity Resolution

- Use rules to express relational dependencies
  - “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.”

1.5 : SimilarFriends(A, B) => SamePerson(A, B)
Entity Resolution

- Use rules to express relational dependencies
  - “If two people have similar names, they are probably the same.”
  - “If two people have similar friends, they are probably the same.”
  - “If A is B and B is C, then A and C must also denote the same person.”

20.0 : SamePerson(A, B) & SamePerson(B, C) => SamePerson(A, C)
Link Prediction

- **Entities**
  - People, Emails

- **Attributes**
  - Words in emails

- **Relationships**
  - communication, work relationship

- **Goal: Identify work relationships**
  - Supervisor, subordinate, colleague
Link Prediction

- 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.”
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.”

1.0 : Contains(E, “due”) => HasType(E, “deadline”)
Link Prediction

- 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.”

\[ 2.0 : \text{Sent}(A, B, E) \& \text{HasType}(E, \text{“deadline”}) \Rightarrow \text{Supervisor}(A, B) \]
Link Prediction

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.”

1.5 : \text{Supervisor}(A, B) \& \text{Supervisor}(A, C) 
=> \text{Colleagues}(B, C)
Collective Classification

1.0 : Mentioned(A, “Barack Obama”) => Votes(A, “Democrat”)

5.0 : Donated(A, “Republican”) => Votes(A, “Republican”)
Collective Classification

1.0 : Votes(A, P) & Friends(A, B) => Votes(B, P)

5.0 : Votes(A, P) & Spouse(A, B) => Votes(B, P)
Probabilistic Soft Logic: Logic Foundation
“Lifted” Rules

\[ H_1(X) \lor \ldots \lor H_m(X) \leftrightarrow B_1(X) \land \ldots \land B_n(X) \]

- 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 \( A \in [0,1] \)

[Broecheler, et al., UAI ‘10]
Combination Functions

- $\lor, \land : [0,1]^n \rightarrow [0,1]$

- Rules will behave like Boolean logic
  - If body is low, rule is always “happy”

Friends(Jay, Steve) $\land$ Friends(Steve, Ben) $\rightarrow$ Friends(Jay, Ben)

Friends(Jay, Steve) $\land$ Friends(Steve, Ben) $\rightarrow$ Friends(Jay, Ben)
Combination Functions

- $\lor, \land : [0,1]^n \rightarrow [0,1]$

- Rules will behave like Boolean logic
  - If body is high, rule only “happy” if head is high

- $\text{Friends}(Jay, Steve) \land \text{Friends}(Steve, Ben) \rightarrow \text{Friends}(Jay, Ben)$

- $\text{Friends}(Jay, Steve) \land \text{Friends}(Steve, Ben) \rightarrow \neg \text{Friends}(Jay, Ben)$
Combination Functions

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

\[ H_1(X) \leftarrow B_1(X) \land B_2(X) \]

- Establish Satisfaction

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

\( H_1(X) \lor \ldots \lor H_m(X) \leftarrow B_1(X) \land \ldots \land B_n(X) \)

\( \text{Distance to Satisfaction} \)

\( \text{Max}(\land (B_1(X), \ldots, B_n(X)) - \lor (H_1(X), \ldots, H_m(X)), 0) \)

<table>
<thead>
<tr>
<th>( H_1(x) )</th>
<th>0.7</th>
<th>( B_1(x) )</th>
<th>0.7</th>
<th>( B_2(x) )</th>
<th>0.8</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H_1(x) )</td>
<td>0.2</td>
<td>( B_1(x) )</td>
<td>0.7</td>
<td>( B_2(x) )</td>
<td>0.8</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Distance to Satisfaction

\[ H_1(X) \lor \ldots \lor H_m(X) \leftarrow B_1(X) \land \ldots \land B_n(X) \]

- Distance to Satisfaction
  \[ \text{Max}(\land (B_1(X), \ldots, B_n(X)) - \lor (H_1(X), \ldots, H_m(X)), 0) \]

- Weighted Rules
  \[ W_r: H_1(X) \lor \ldots \lor H_m(X) \leftarrow B_1(X) \land \ldots \land B_n(X) \]

- Weighted Distance to Satisfaction
  \[ W_r \cdot \max(\land (B_1(x), \ldots, B_n(x)) - \lor (H_1(x), \ldots, H_m(x)), 0) \]
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.
  - (One distance for each ground rule)
- How to decide which assignment is best?
Probabilistic Soft Logic: Probability Foundation
Probabilistic Model

\[ f(I) = \frac{1}{Z} \exp\left[-\sum_{r \in R} \lambda_r (d_r(I))^p\right] \]

- **Rule’s weight**
- **Rule’s distance to satisfaction**
  \[ d_r(I) = \max(0, I_{r,\text{body}} - I_{r,\text{head}}) \]
- **Distance exponent** in \{1, 2\}
- **Normalization constant**
- **Set of ground rules**
- **Probability density over interpretation I**
Hinge-loss Markov Random Fields

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

- PSL models ground out to HL-MRFs
- Continuous variables in [0,1]
- Potentials are hinge-loss functions
- Subject to arbitrary linear constraints
- Log-concave!

Bach et al., UAI 2013
Probabilistic Soft Logic: Example Program
Toy Example: A Social Network

Ali’s Interests:
Photography
Monster Truck Rallies

Brad’s Interests:
Photography
Nature Walks

Claudia’s Interests:
Photography
Geocaching

Dave’s Interests:
Photography
Geocaching
Toy Example: A Social Network

Ali’s Interests: Photography, Monster Truck Rallies

Brad’s Interests: Photography, Nature Walks

Claudia’s Interests: Photography, Geocaching

Dave’s Interests: Photography, Geocaching
Toy Example: A Social Network

Brad's Interests:
- Photography
- Nature Walks

Dave's Interests:
- Photography
- Geocaching

Claudia's Interests:
- Photography
- Geocaching

Ali’s Interests:
- Photography
- Monster Truck Rallies

RULES:

2.0 \text{Interest}(P_1, I) \& \text{Interest}(P_2, I) \implies \text{Friends}(P_1, P_2)

2.5 \text{Friends}(P_1, P_2) \& \text{Friends}(P_2, P_3) \implies \text{Friends}(P_1, P_3)

5.0 \text{Interest}(P_1, \text{Monster Truck Rallies}) \& \text{Interest}(P_2, \text{Nature Walks}) \implies !\text{Friends}(P_1, P_2)

1.0 !\text{Friends}(P_1, P_2)
Constructing a HL-MRF

1) Ground out all rules
Example Groundings

- Interest(P1, I) & Interest(P2, I) $\Rightarrow$ Friends(P1, P2)
  - Interest(Ali, Photography) & Interest(Brad, Photography) $\Rightarrow$ Friends(Ali, Brad)
  - Interest(Claudia, Photography) & Interest(Dave, Photography) $\Rightarrow$ Friends(Claudia, Dave)
  - Interest(Claudia, Geocaching) & Interest(Dave, Geocaching) $\Rightarrow$ Friends(Claudia, Dave)
  - etc.

- Interest(P1, M.T.R.) & Interest(P2, N.W.) $\Rightarrow$ !Friends(P1, P2)
  - Interest(Ali, M.T.R.) & Interest(Brad, N.W.) $\Rightarrow$ !Friends(Ali, Brad)

- Friends(P1, P2) & Friends(P2, P3) $\Rightarrow$ Friends(P1, P3)
  - Friends(Ali, Brad) & Friends(Brad, Dave) $\Rightarrow$ Friends(Ali, Dave)
  - etc.
Constructing a HL-MRF

1) Ground out all rules
2) Convert ground rules to hinge-loss functions
Example hinge-loss function

- Start with a ground rule
  - Friends(Ali, Brad) & Friends(Brad, Dave) \(\Rightarrow\) Friends(Ali, Dave)

- Map atoms to random variables
  - Friends(Ali, Brad) = \(Y_1\)
  - Friends(Brad, Dave) = \(Y_2\)
  - Friends(Ali, Dave) = \(Y_3\)

- Interpret with \(t\)-norm
  - \(\min\{2 - Y_1 - Y_2 + Y_3, 1\}\)
Example hinge-loss function

- Subtract from 1 to find distance to satisfaction
  
  \[ 1 - \min\{2 - Y_1 - Y_2 + Y_3, 1\} \]
  
  \[ = \max\{Y_1 + Y_2 - Y_3 - 1, 0\} \]
Constructing a HL-MRF

- 1) Ground out all rules
- 2) Convert ground rules to hinge-loss functions
- 3) Weight hinge-loss functions and embed in HL-MRF

\[
p(Y|X) = \frac{1}{Z(w, X)} \exp \left[ - \sum_{j=1}^{m} w_j \left\{ \max \{ \ell_j(Y, X), 0 \} \right\}^{1,2} \right]
\]
Inference
Making Predictions

- Want to find a most probable explanation (MPE)

\[
\arg\max_Y P(Y|X) \equiv \arg\min_Y f(Y, X) \quad Y \in [0,1]^n
\]

\[
\equiv \arg\min_Y \sum_{j=1}^m w_j \phi_j(Y, X) \quad Y \in [0,1]^n
\]

s.t. \( C_k(Y, X) = 0, \quad \forall k \in \mathcal{E} \)
and \( C_k(Y, X) \geq 0, \quad \forall k \in \mathcal{I} \)
Why MPE?

- Best scoring assignment to unknown variables
- But it's continuous valued!

- How do we interpret continuous values?
  - Round
  - Treat as ranking
Convex Optimization

- MPE inference is a convex optimization problem

- Could use off-the-shelf toolkits

- Best general methods have $O(n^{3.5})$ time complexity
Alternating Direction Method of Multipliers

- Perform inference with ADMM framework

- Optimize subproblems (hinge losses and constraints) independently, in parallel

- Auxiliary variables (Lagrange multipliers) ensure consensus reached across subproblems

Bach et al., NIPS 2012
ADMM Scalability

Bach et al., NIPS 2012
Distributed MPE: GraphLab

subproblem node

$X_1$

$X_m$

$X_{m+1}$

$X_{m+r}$

global variable component

$Z_1$

$Z_2$

$Z_q$

$Z_p$

gather
get $z$

apply
update $y$
update $x$

scatter
notify $z$

get local $z,y$

apply
update $z$

scatter
unless converge
notify $X$

update $i$

update $i+1$
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, Getoor, UAI 2013
Experimental Results
Experiments: Collective Classification

- Given a networked collection of documents
- Observe some labels
- Predict remaining labels using link structure

<table>
<thead>
<tr>
<th>Method</th>
<th>Citeseer</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>MRF (MLE)</td>
<td>0.686</td>
<td>0.756</td>
</tr>
<tr>
<td>MRF (MPLE)</td>
<td>0.715</td>
<td>0.797</td>
</tr>
<tr>
<td>MRF (LME)</td>
<td>0.687</td>
<td>0.783</td>
</tr>
</tbody>
</table>

Table 1: Average accuracy of classification by HL-MRFs and discrete MRFs. Scores statistically equivalent to the best scoring method are typed in bold.

Bach et al., UAI 2013
Experiments: Social Trust Prediction

- Given a social network
- Observe some trust links
- Predict remaining trust links using structural balance

<table>
<thead>
<tr>
<th></th>
<th>ROC</th>
<th>P-R (+)</th>
<th>P-R (-)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL-MRF-Q (MLE)</td>
<td>0.822</td>
<td>0.978</td>
<td>0.452</td>
</tr>
<tr>
<td>HL-MRF-Q (MPLE)</td>
<td>0.832</td>
<td>0.979</td>
<td>0.482</td>
</tr>
<tr>
<td>HL-MRF-Q (LME)</td>
<td>0.814</td>
<td>0.976</td>
<td>0.462</td>
</tr>
<tr>
<td>HL-MRF-L (MLE)</td>
<td>0.765</td>
<td>0.965</td>
<td>0.357</td>
</tr>
<tr>
<td>HL-MRF-L (MPLE)</td>
<td>0.757</td>
<td>0.963</td>
<td>0.333</td>
</tr>
<tr>
<td>HL-MRF-L (LME)</td>
<td>0.783</td>
<td>0.967</td>
<td>0.453</td>
</tr>
<tr>
<td>MRF (MLE)</td>
<td>0.655</td>
<td>0.942</td>
<td>0.270</td>
</tr>
<tr>
<td>MRF (MPLE)</td>
<td>0.725</td>
<td>0.963</td>
<td>0.298</td>
</tr>
<tr>
<td>MRF (LME)</td>
<td>0.795</td>
<td><strong>0.973</strong></td>
<td>0.441</td>
</tr>
</tbody>
</table>

Table 1: Average area under ROC and precision-recall curves of social-trust prediction by HL-MRFs and discrete MRFs. Scores statistically equivalent to the best scoring method by metric are typed in bold.

Bach et al., UAI 2013
## HL-MRF Scalability

Table 1: Average inference times (reported in seconds) of single-threaded HL-MRFs and discrete MRFs.

<table>
<thead>
<tr>
<th></th>
<th>Citeseer</th>
<th>Cora</th>
<th>Epinions</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL-MRF-Q</td>
<td>0.42</td>
<td>0.70</td>
<td>0.32</td>
</tr>
<tr>
<td>HL-MRF-L</td>
<td>0.46</td>
<td>0.50</td>
<td>0.28</td>
</tr>
<tr>
<td>MRF</td>
<td>110.96</td>
<td>184.32</td>
<td>212.36</td>
</tr>
<tr>
<td>Variables</td>
<td>10K</td>
<td>10K</td>
<td>1K</td>
</tr>
<tr>
<td>Potentials and Constraints</td>
<td>14K</td>
<td>19K</td>
<td>18K</td>
</tr>
</tbody>
</table>
Probabilistic Soft Logic: Demonstration
Installing PSL and Examples

- **Prerequisites:**
  - Java 6
  - Maven 3 (http://maven.apache.org)
Installing PSL and Examples

- [http://psl.cs.umd.edu](http://psl.cs.umd.edu)

---

**Introduction**

Probabilistic soft logic (PSL) is a modeling language (with accompanying implementation) for learning and predicting in relational domains. Such tasks occur in many areas such as natural language processing, social-network analysis, computer vision, and machine learning in general.

PSL allows users to describe their problems in an intuitive, logic-like language and then apply their models to data.

Details:

- PSL models are templates for hinge-loss Markov random fields (HL-MRFs), a powerful class of probabilistic graphical models.
- HL-MRFs are extremely scalable models because they are log-concave densities over continuous variables that can be optimized using the alternating direction method of multipliers.
- See the publications page for more technical information and applications.

The following presentation provides more information on PSL:

- [http://psl.umiacs.umd.edu](http://psl.umiacs.umd.edu)

**Probabilistic Soft Logic:**
Installing PSL and Examples

- https://github.com/linqs/psl
Installing PSL and Examples

- https://github.com/linqs/psl/wiki
Installing PSL and Examples

- https://github.com/linqs/psl/wiki/Installing-examples