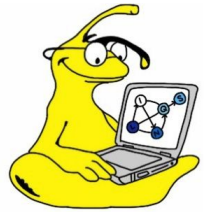
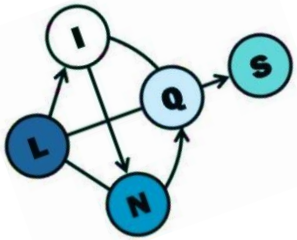
Additional Topics

Eriq Augustine and Golnoosh Farnadi
UC Santa Cruz
MLTrain 2018



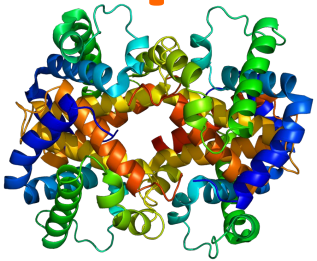
psl.linqs.org
github.com/linqs/psl

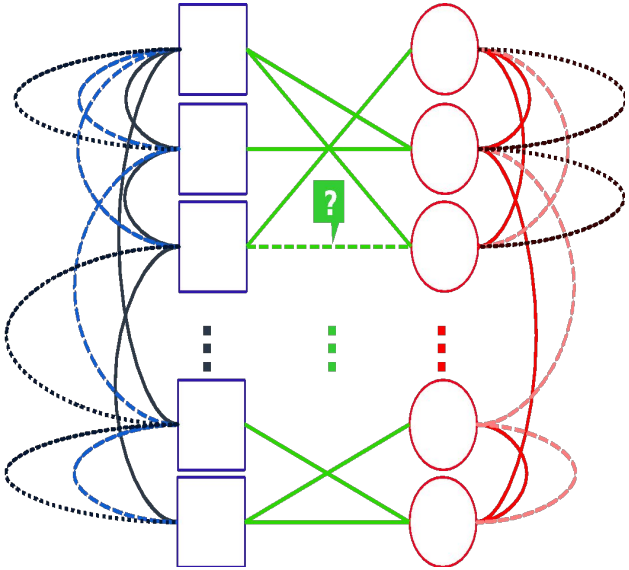


Additional PSL Models

Model - Drug Interaction Discovery

Predicting new drug-protein interactions for drug discovery, repurposing, side-effect prediction, and personalized medicine.





Model - Drug Interaction Discovery

```
// Drug similarity triadic structure.
20: Interacts(D1,T) & ChemicalSimilar(D1,D2)   -> Interacts(D2,T)
20: Interacts(D1,T) & SideEffectSimilar(D1,D2) -> Interacts(D2,T)
30: Interacts(D1,T) & AnnotationSimilar(D1,D2) -> Interacts(D2,T)

// Target similarity triadic structure.
30: Interacts(D,T1) & SequenceSimilar(T1,T2) -> Interacts(D,T2)
20: Interacts(D,T1) & OntologySimilar(T1,T2) -> Interacts(D,T2)

// Both similarities tetrad structure.
30: Interacts(D1,T1) & SequenceSimilar(T1,T2) & ChemicalSimilar(D1,D2)
      -> Interacts(D2,T2)
40: Interacts(D1,T1) & OntologySimilar(T1,T2) & SideEffectSimilar(D1,D2)
      -> Interacts(D2,T2)

//Prior
10: !Interacts(D,T)
```

https://github.com/shobeir/fakhraei_tcbb2014

Model - Drug Interaction Discovery

Task: Find new interactions between drugs and proteins targets in the drugbank dataset.

Newly Discovered Interactions

DRUGBANK
Open Data Drug & Drug Target Database

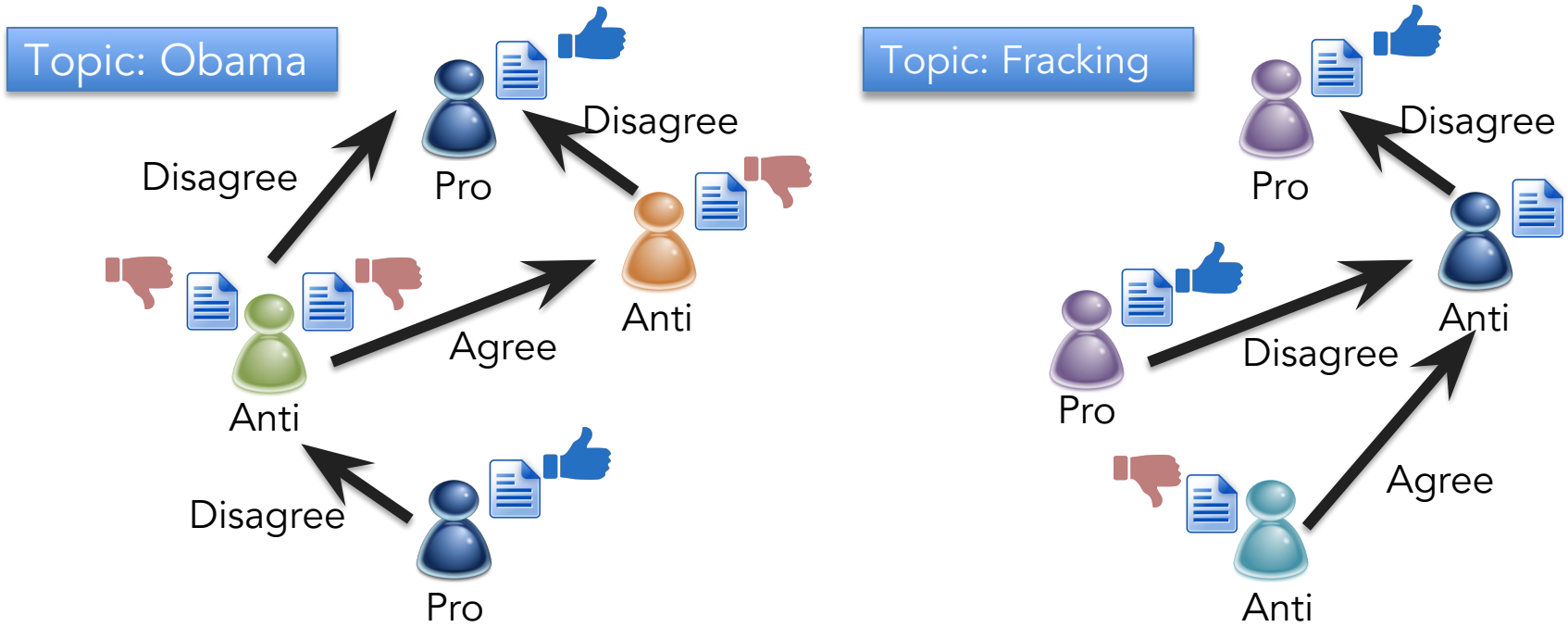
	AUC	AUPR	P@130
Perlman et al.	0.921	0.309	0.393
PSL-Model	0.926	0.344	**0.460**

Found 197 out of 78,750 possible interactions!

Network-based Drug-Target Interaction Prediction with Probabilistic Soft Logic, S. Fakhraei, B. Huang, L. Raschid, and L. Getoor, IEEE Transactions on Computational Biology and Bioinformatics (IEEE-TCBB), 2014. (Cover)
https://lings.soe.ucsc.edu/node/9

Model - Debate Stance Classification

Jointly infer users' attitude on topics and polarity of interaction from online debate forum threads.



Joint Models of Disagreement and Stance, Sridhar, Foulds, Huang, Getoor & Walker, Annual Meeting of the Association for Computational Linguistics (ACL), 2015
https://lings.soe.ucsc.edu/node/258

Model - Debate Stance Classification

```
// Priors from local text classifiers
1:  PriorPro(U,T)                  ->  Pro(U,T)
1:  PriorDisagree(U1,U2)           ->  Disagrees(U1,U2)

// Rules for stance
5:  Disagrees(U1,U2) & Pro(U1,T)  ->  !Pro(U2,T)
5: !Disagrees(U1,U2) & Pro(U1,T)  ->   Pro(U2,T)

// Rules for disagreement
5:  Pro(U1,T)        & Pro(U1,T)  ->  !Disagrees(U1,U2)
5: !Pro(U1,U2)       & Pro(U1,T)  ->   Disagrees(U1,U2)
```

Model - Debate Stance Classification

Task: Predict post and user stance on topics from two online debate forums:

- 4Forums.com: ~300 users,~6000 posts
- CreateDebate.org: ~300 users, ~1200 posts

4Forums.com

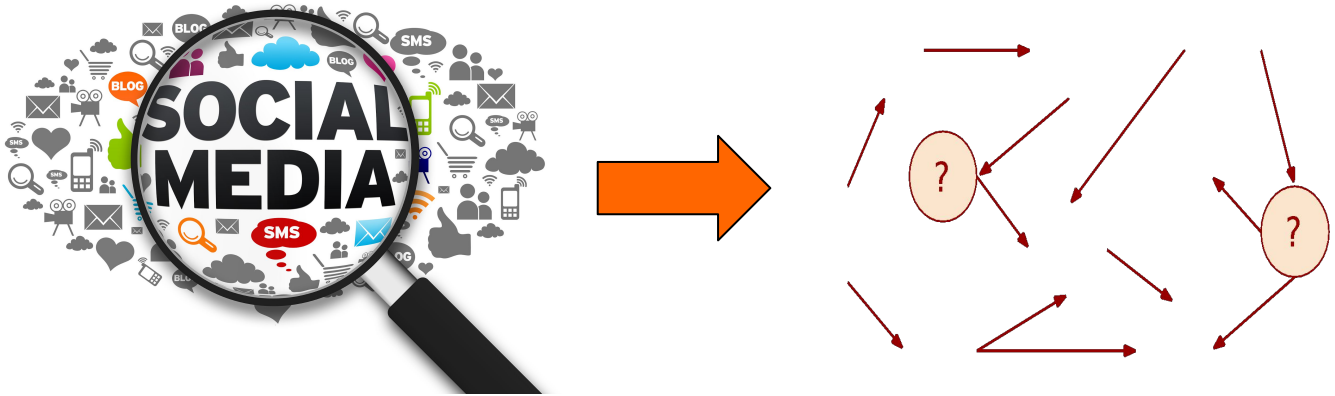
	User Stance Accuracy	Post Stance Accuracy
Logistic Regression Baseline	72.0	69.0
PSL-Post	73.7	72.5
PSL-Author	77.1	80.3

CreateDebate.org

	User Stance Accuracy	Post Stance Accuracy
Logistic Regression Baseline	70.2	62.7
PSL-Post	73.2	66.2
PSL-Author	74.0	72.7

Model - Finding Social Spammers

Find spammers in social media.



Collective Spammer Detection in Evolving Multi-Relational Social Networks, S. Fakhraei, J. Foulds, M. Shashanka, L. Getoor. ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2015
https://lings.soe.ucsc.edu/node/251

Model - Finding Social Spammers

```
// User generated reports
30: Credible(U1) & ReportedSpammer(U1,U2) -> Spammer(U2)

// Collective credibility
25: Spammer(U2)  & ReportedSpammer(U1,U2) -> Credible(U1)
25: !Spammer(U2) & ReportedSpammer(U1,U2) -> !Credible(U1)

// Prior credibility
20: PriorCredible(U)  -> Credible(U)
20: !PriorCredible(U) -> !Credible(U)

// Prior
10: !Spammer(U)
```

https://github.com/shobeir/fakhraei_kdd2015

Model – Spammer Detection

Task: Detecting social spammers in tagged.com social network using user-generated spammer reports.

- Attributes: Gender, Age, Account Age, Label
- Links: 8 Actions such as Like, Poke, Report Abuse, etc.

Spammers Detected

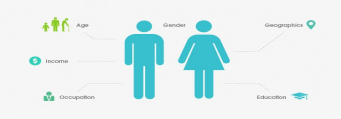
TAGGED

	AUC	AUPR
Using only reports	0.611	0.674
Using report and credibility	0.862	0.869
PSL (fully collective model)	0.873	0.884

Model - Hybrid Recommender Systems

Improve recommendations by combining data sources & recommenders.

ratings ***content*** ***social*** ***demographic***



Predicted Ratings

Hybrid Recommender (HyPER)

Matrix Factorization

Item-based Collaborative Filtering

...

Bayesian Probabilistic Matrix Factorization

...

HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems Kouki, Fakhraei, Foulds, Eirinaki, Getoor, RecSys15

https://lings.soe.ucsc.edu/node/257

Model - Hybrid Recommender Systems

```
// Similar Items
10: Rating(U,I1) & PearsonSimilarityItems(I1,I2) -> Rating(U,I2)
10: Rating(U,I1) & ContentSimilarityItems(I1,I2) -> Rating(U,I2)

// Similar Users
10: Rating(U1,I) & PearsonSimilarityUsers(U1,U2) -> Rating(U2,I)
10: Rating(U1,I) & CosineSimilarityUsers (U1,U2) -> Rating(U2,I)

// Social Information
10: Friends(U1,U2) & Rating(U1,I) -> Rating(U2,I)

// Other Recommenders
10: MFRating(U,I)   -> Rating(U,I)
10: BPMFRating(U,I) -> Rating(U,I)

// Average Priors
1:  AvgUserRating(U) -> Rating(U,I)
1:  AvgItemRating(I) -> Rating(U,I)
```

https://github.com/pkouki/recsys2015

Model - Hybrid Recommender Systems

Task: Predict missing ratings

- Yelp: 34K users, 3.6K items, 99K ratings, 81K friendships, 500 business categories
- Last.fm: 1.8K users, 17K items, 92K ratings, 12K friendships, 9.7K artist tags

yelp

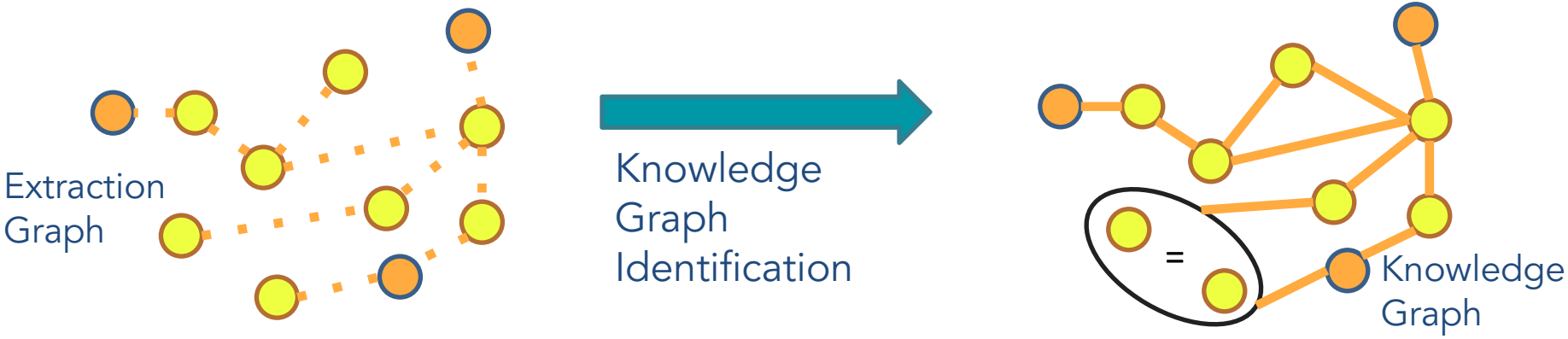
Model	RMSE
Item-based	1.216
MF	1.251
BPMF	1.191
Naïve Hybrid	1.179
BPMF-SRIC	1.191
HyPER	**1.173**

last·fm™

Model	RMSE
Item-based	1.408
MF	1.178
BPMF	1.008
Naïve Hybrid	1.067
BPMF-SRIC	1.015
HyPER	**1.001**

Model - Knowledge Graph Identification

Refine noisy knowledge extractions into an accurate knowledge graph.



Knowledge Graph Identification, Pujara, Miao, Getoor, & Cohen, ISWC, 2013
https://lings.soe.ucsc.edu/node/28

Model - Knowledge Graph Identification

```
// Ontological relationships
100: Subsumes(L1,L2)   & Label(E,L1)      ->  Label(E,L2)
100: Exclusive(L1,L2)  & Label(E,L1)      -> !Label(E,L2)
100: Inverse(R1,R2)    & Relation(R1,E,O) ->  Relation(R2,O,E)
100: Domain(R,L)       & Relation(R,E,O)  ->  Label(E,L)
100: Range(R,L)        & Relation(R,E,O)  ->  Label(O,L)

// Entity resolution
10: SameEntity(E1,E2) & Label(E1,L)      ->  Label(E2,L)
10: SameEntity(E1,E2) & Relation(R,E1,O) ->  Relation(R,E2,O)

// Integrating knowledge sources
1: LabelNYT(E,L)            ->  Label(E,L)
1: LabelYouTube(E,L)        ->  Label(E,L)
1: RelationWikipedia(R,E,O) ->  Relation(R,E,O)

// Priors
1: !Relation(R,E,O)
1: !Label(E,L)
```

https://github.com/lings/KnowledgeGraphIdentification

Model - Knowledge Graph Identification

Task: Construct a knowledge graph from millions of web text extractions from CMU's NELL project.

Knowledge graph for an explicit test set

	AUC	F1
Baseline	0.873	0.828
NELL	0.765	0.673
MLN (Jiang, 12)	0.899	0.836
PSL-KGI	0.904	0.853

Running Time: Inference completes in 10 seconds, produces **25K facts**

Complete knowledge graph including all NELL candidates

	AUC	F1
NELL	0.765	0.634
PSL-KGI	0.892	0.848

Running Time: Inference completes in 130 minutes, produces **4.3M facts**

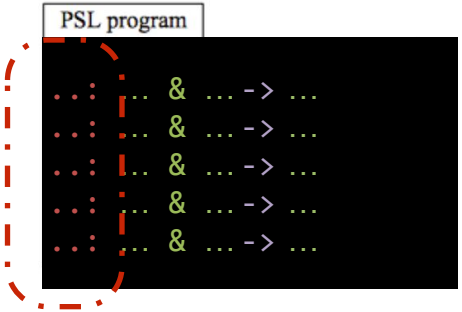
Using Statistics & Semantics to Turn Data Into Knowledge, Pujara, Miao, Getoor, & Cohen, AI Magazine, 2015
https://lings.soe.ucsc.edu/node/272

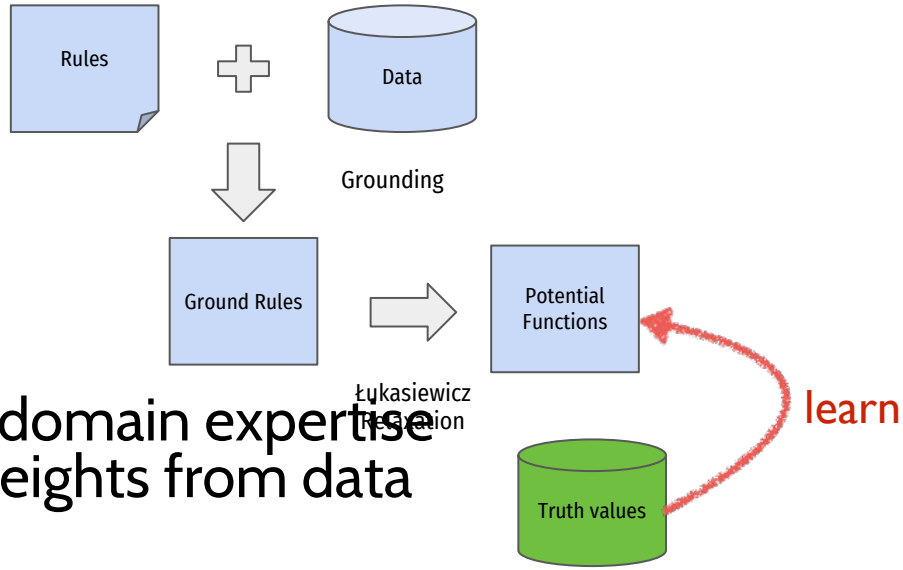
Advanced Topics

Advanced Topics (not covered)

- Temporal & spatial modeling
- **Weight learning**
- **Structure learning**
- Causal modeling
- **Lifted Inference**
- **Fairness**
- Decision making

Weight Learning





weight

- Manual weights given users' domain expertise
- PSL supports learning rule weights from data

Weight Learning

- **Maximum-likelihood Estimation:** performs approximate maximum-likelihood estimation using MPE inference to approximate the gradient of the log-likelihood

$$\frac{\partial \log p(\mathbf{Y}|\mathbf{X})}{\partial \Lambda_q} = \mathbb{E}_{\Lambda}\left[\Phi_q(\mathbf{Y},\mathbf{X})\right] - \Phi_q(\mathbf{Y},\mathbf{X})$$

weight

expectation

Weight Learning

- **Maximum-pseudolikelihood Estimation:** which maximizes the likelihood of each variable conditioned on all other variables

$$\frac{\partial \log P^*(Y|X)}{\partial \Lambda_q} = \sum_{i=1}^{n} \mathbb{E}_{Y_i|\mathrm{MB}} \left[\sum_{j \in t_q : i \in \phi_j} \phi_j(\mathbf{Y}, \mathbf{X}) \right] - \Phi_j(\mathbf{Y}, \mathbf{X}).$$

Markov blanket

- **Large Ma**

$$\arg\min_{\tilde{\mathbf{Y}}} \Lambda^\top \Phi(\tilde{\mathbf{Y}}, \mathbf{X}) - L(\mathbf{Y}, \tilde{\mathbf{Y}}).$$

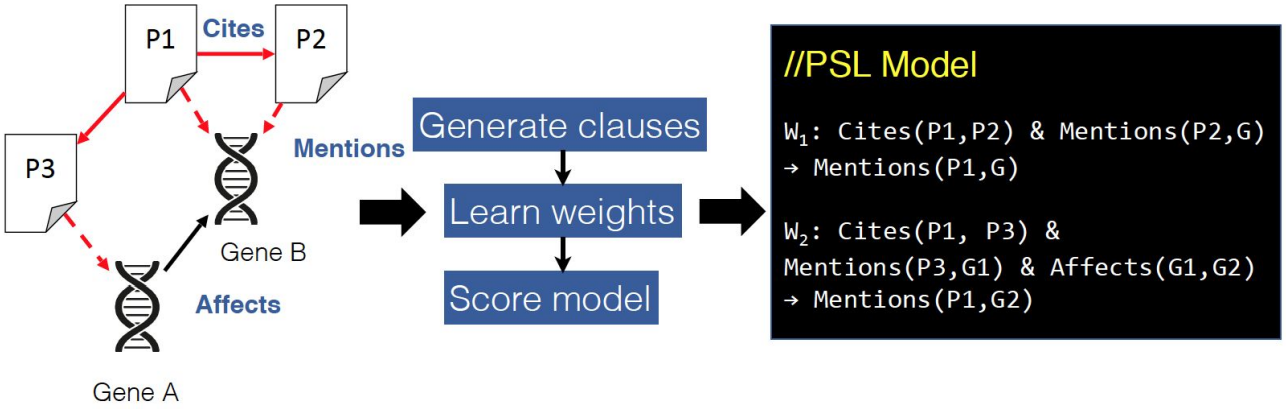
Result Highlights

	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
MRF (MLE)	0.686	0.756
MRF (MPLE)	0.715	0.797
MRF (LME)	0.687	0.783

S.H Bach, et. al, Hinge-loss Markov Random Fields: Convex Inference for Structured Prediction, UAI 2013

Structure Learning

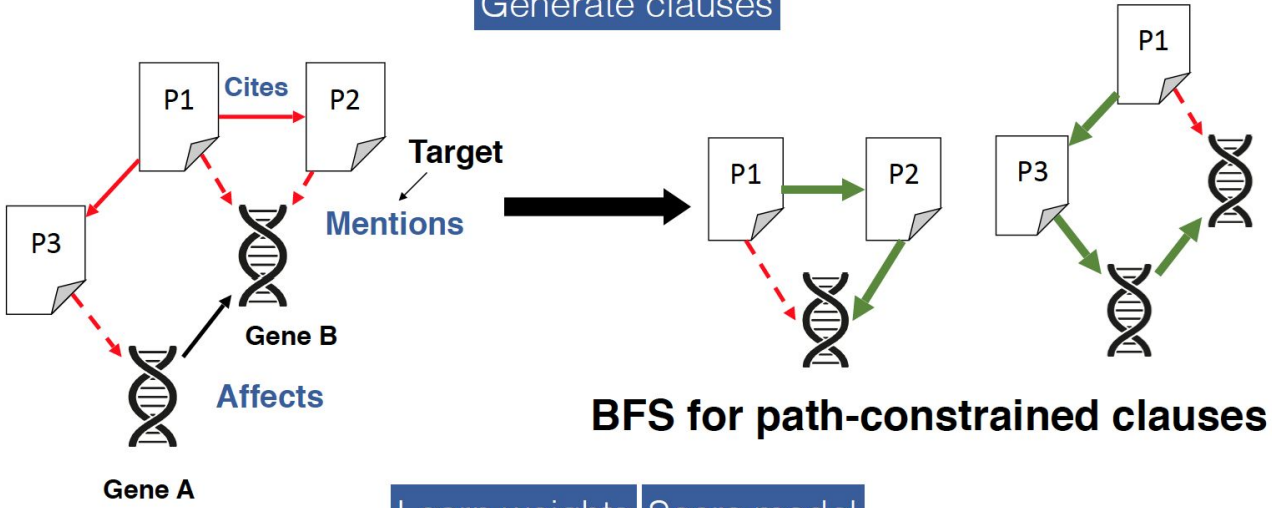
- Learn weighted logical clauses from relational data



- **Challenges**: combinatorial clause search; repeated weight learning; intractable likelihood

Structure Learning in PSL

Generate clauses



BFS for path-constrained clauses

Learn weights | Score model

```
W1: Cites(P1,P2) & Mentions(P2,G) → Mentions(P1,G)
W2: Cites(P1, P3) & Mentions(P3,G1) & Affects(G1,G2) → Mentions(P1,G2)
```

Piecewise pseudolikelihood scoring: only weight learning!

Result Highlights

5 fold CV:

Dataset	Greedy	PPLL
Fly	0.95	0.97
Yeast	0.86	0.90
DrugBank	0.66	0.76
Freebase	0.65	0.65

Significant AUC gains

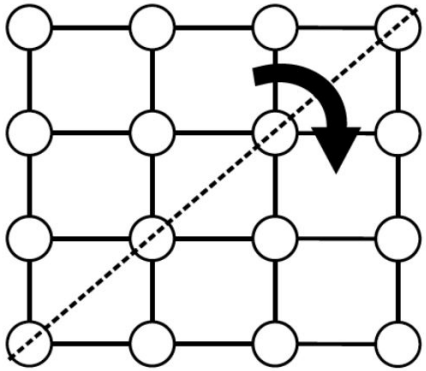
Runtimes (in log sec):

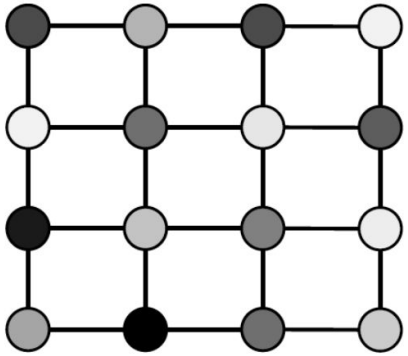
Scalability

Embar, Sridhar, Farnadi & Getoor, *StarAI* 2018

Lifted Inference in PSL

Lifted inference gives exponential speedups in symmetric graphical models.





Existing lifted inference approaches focus on discrete graphical models

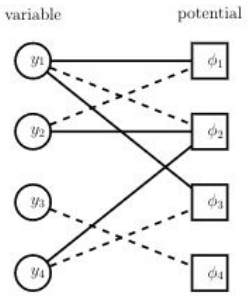
How to find symmetry in PSL with continuous atoms?

Lifted Inference in PSL

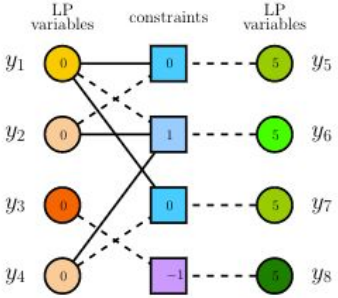
PSL program

```
..: ... & ... -> ...
..: ... & ... -> ...
..: ... & ... -> ...
..: ... & ... -> ...
..: ... & ... -> ...
```

convert to a bipartite graph



color the graph with a
color refinement algorithm

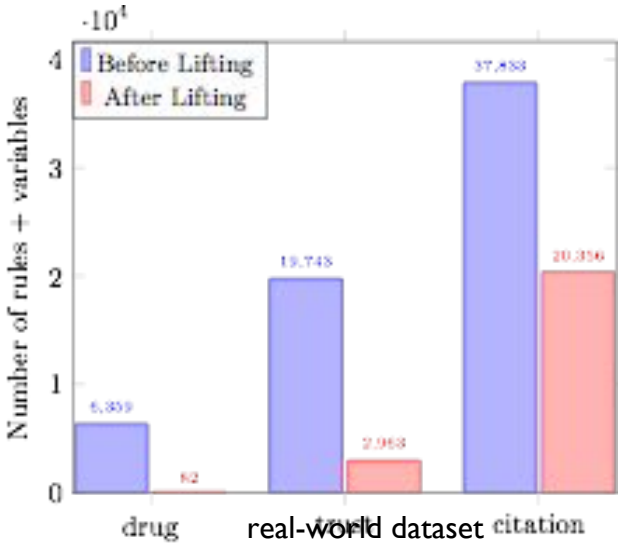


PSL program

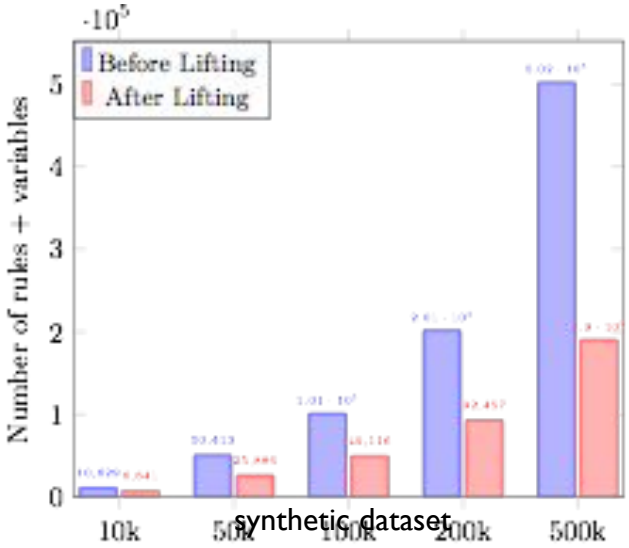
```
..: ... & ... -> ...
..: ... & ... -> ...
```

[in progress]

Result Highlights



real-world dataset



synthetic dataset

We observe a 3 to 68 times speed up in inference

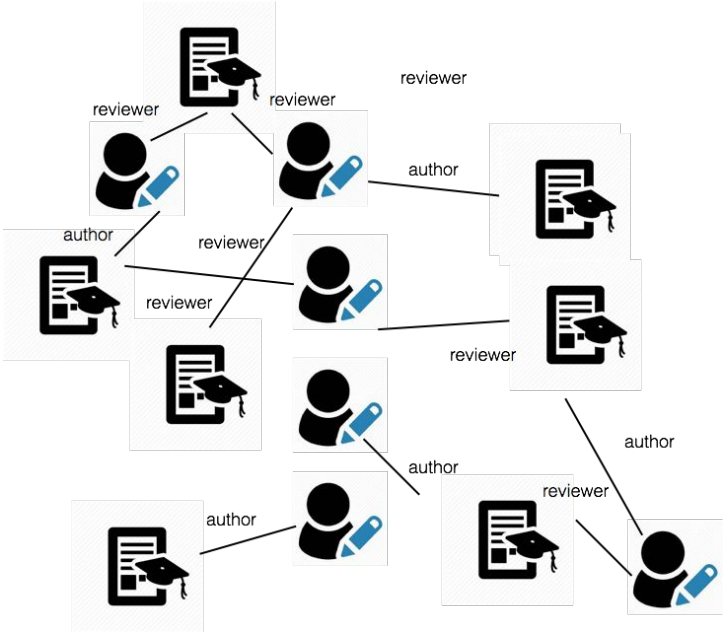
[in progress]

Fairness in Relational Domain

The goal of fairness-aware machine learning: is to ensure that the decisions made by an algorithm **do not discriminate against a population of individuals**

Challenge: Existing fairness approaches are based solely on attributes of individuals e.g., age, gender, race, etc.

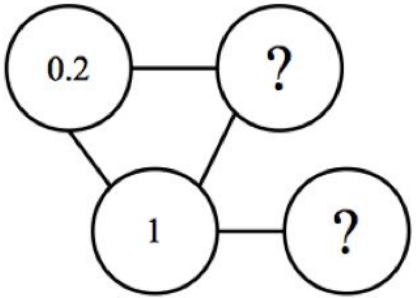
Our contribution: We introduce new notions of fairness that are able to capture the relational structure in a domain, e.g., citation network, corporate hierarchy, social network. etc.

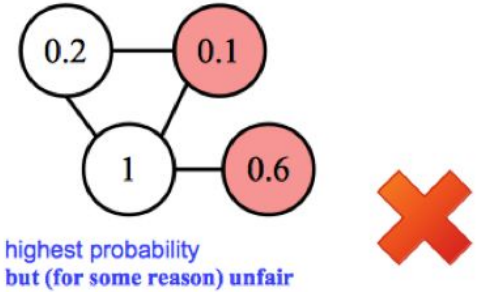


MAP Inference in PSL

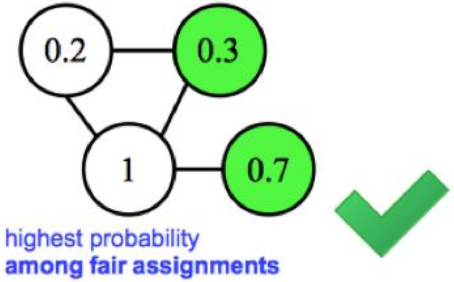
$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(w, \mathbf{X})} \exp \sum_{j=1}^{m} w_j \phi_j$$

$$I_{MAP}(Y) = arg \max_{I(Y)} P(I(Y)|I(X))$$



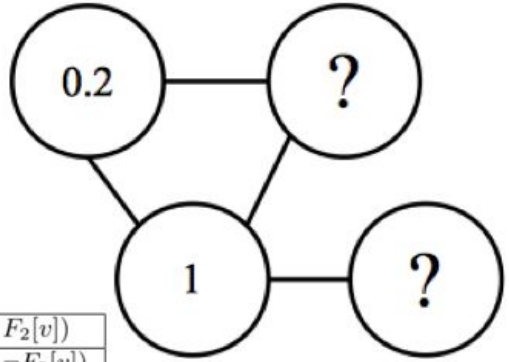


highest probability
but (for some reason) unfair



highest probability
among fair assignments

Fair MAP Inference in PSL



PSL program

```
..: ... & ... -> ...
..: ... & ... -> ...
..: ... & ... -> ...
..: ... & ... -> ...
..: ... & ... -> ...
```

$\mathbf{a}$	$\sum_{v \in D_v} I(\neg d(v) \wedge F_1[v] \wedge F_2[v])$
$\mathbf{c}$	$\sum_{v \in D_v} I(\neg d(v) \wedge F_1[v] \wedge \neg F_2[v])$
n_1	$\sum_{v \in D_v} I(F_1[v] \wedge F_2[v])$
n_2	$\sum_{v \in D_v} I(F_1[v] \wedge \neg F_2[v])$

δ-fairness measure	Constraints
$-\delta \leq RD \leq \delta$	$n_2\mathbf{a} - n_1\mathbf{c} - n_1 n_2 \delta \leq 0$ $n_2\mathbf{a} - n_1\mathbf{c} + n_1 n_2 \delta \geq 0$
$1-\delta \leq RR \leq 1+\delta$	$n_2\mathbf{a} - (1+\delta) n_1\mathbf{c} \leq 0$ $n_2\mathbf{a} - (1-\delta) n_1\mathbf{c} \geq 0$
$1-\delta \leq RC \leq 1+\delta$	$-n_2\mathbf{a} + (1+\delta) n_1\mathbf{c} - \delta n_1 n_2 \leq 0$ $-n_2\mathbf{a} + (1-\delta) n_1\mathbf{c} + \delta n_1 n_2 \geq 0$

argmax ...
s.t.
(constraint)
(constraint)
(constraint)
(fairness constraint)

Enforce fairness by adding extra linear constraints to the optimization problem

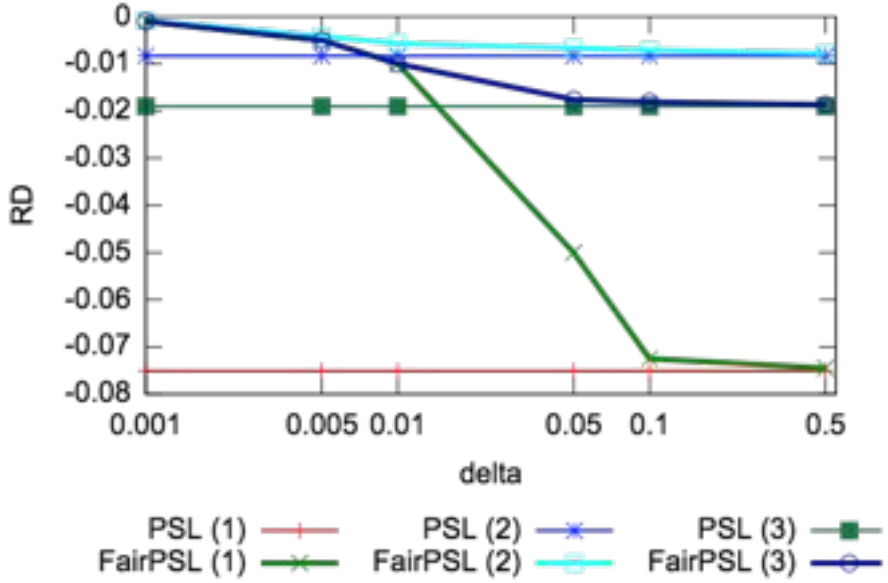
$$I_{MAP}(Y) = \arg\max_{I(Y)} P(I(Y)|I(X))$$

Result Highlights

The paper reviewing problem: Ensure fair acceptance rate for students from high rank universities and low rank universities

We show our approach enforces fariness guarantees while preserving the accuracy of the predictions.

#1 has 102 papers, dataset #2 has 109 papers and dataset #3 has 101 papers
delta-fairness with five thresholds {0.001, 0.005, 0.01, 0.05, 0.1, 0.5}



The Code and data are available:
https://github.com/gfarnadi/FairPSL

Farnadi, Babaki & Getoor, *AAAI/ACM Conference on AI, Ethics, and Society* 2018

PSL Takeaways & Resources

PSL Takeaways

- Declarative language able to represent richly structured domains
- Supports collective reasoning – dependencies in inputs and outputs
- Mixes logical and probabilistic reasoning in flexible and scalable manner
- Applicable to wide variety of problems ranging from data integration & fusion to modeling socio-behavioral and scientific domains
- Eager to apply to additional domains, come talk with us if you are interested!

References

- Websites:
  - PSL: https://psl.linqs.org
  - LINQS: linqs.org
  - D3: https://d3.ucsc.edu
- Papers:
  - Main PSL Paper:
    Hinge-Loss Markov Random Fields and Probabilistic Soft Logic, Stephen Bach, Matthias Broecheler, Bert Huang, Lise Getoor, JMLR 2017
  - LINQS Publications: https://linqs.soe.ucsc.edu/biblio

Code

- Main Repository: https://github.com/linqs/psl
- Dev Repository: https://github.com/eriq-augustine/psl
- Examples: https://github.com/linqs/psl-examples
- Documentation:
  - API Reference: https://linqs-data.soe.ucsc.edu/psl-docs
  - Stable Wiki: https://github.com/linqs/psl/wiki
  - Development Wiki: https://github.com/eriq-augustine/psl/wiki

Thanks

- Dhanya Sridhar & Jay Pujara for slide material
- LINQS research group
- PSL Users & Contributors
- UCSC D3 Data Science Center Members
- Nick Vasiloglou II & Relational.AI
- UAI Organizers

Questions?