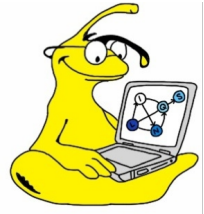
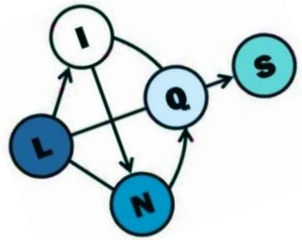
Link Prediction

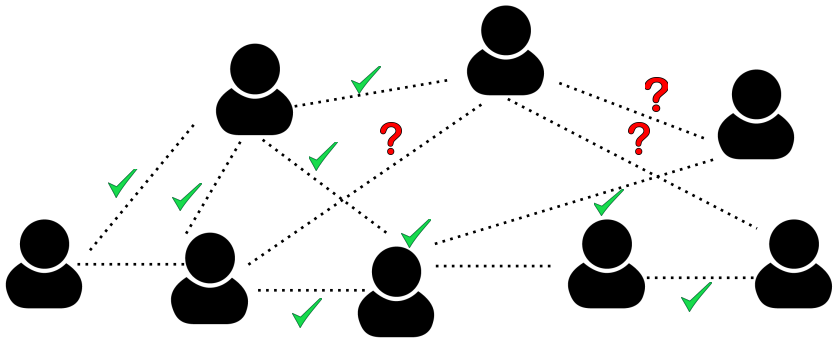
Eriq Augustine and Golnoosh Farnadi
UC Santa Cruz
MLTrain 2018



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



What is Link Prediction?

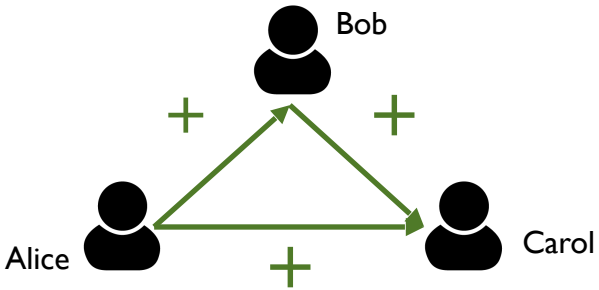


```
Trusts(U,V)
```

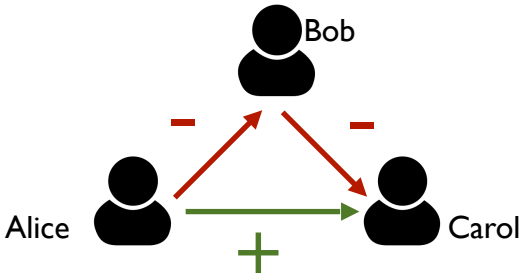
User “**U**” Trust User “**V**”

Social Trust Models: Inferring Trust Networks

- **Model #1: Structural balance** (Granovetter, '73). Strong ties governed by tendency toward balanced triads. E.g.,
  - "Any friend of yours is a friend of mine."

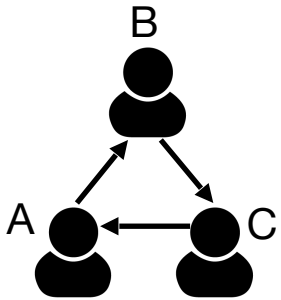


  - "The enemy of my enemy is my friend."



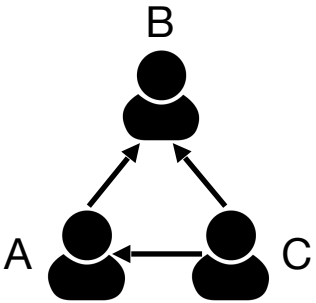
A Flexible Framework for Probabilistic Models of Social Trust, Huang, Kimmig, Getoor, Golbeck, International Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction (SBP), 2013

PSL Structural Balance



Cycle Structure

```
//Rules for cycle structure

  1:  Trusts(A,B) & Trusts(B,C)-> Trusts(C,A)
  1:  !Trusts(A,B) & !Trusts(B,C)-> Trusts(C,A)
```

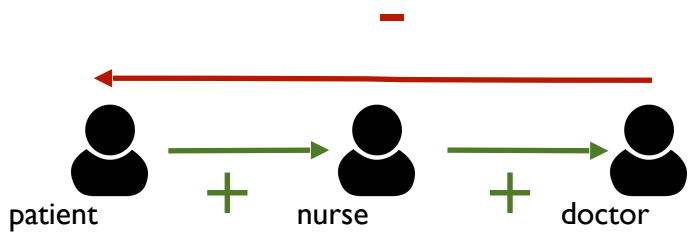


Non-Cycle Structure

```
//Rules for Non-cycle structure

  1:  Trusts(A,B) & Trusts(C,B)-> Trusts(C,A)
  1:  !Trusts(A,B) & !Trusts(C,B)-> Trusts(C,A)
  1:  !Trusts(A,B) & Trusts(C,B)-> !Trusts(C,A)
  1:  Trusts(A,B) & !Trusts(C,B)-> !Trusts(C,A)
```

Social Trust Models: Inferring Trust Networks

- **Model #2: Social status** (Cosmides & Tooby, '92). Strong ties indicate unidirectional respect, "looking up to," expertise status



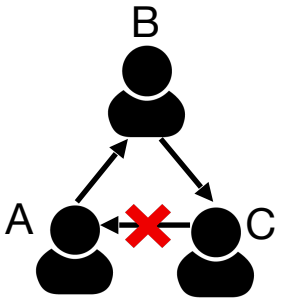
e.g., advisor-advisee, patient-nurse-doctor

- Leskovec et al. (2010) explored occurrence of both models in data and single-edge prediction

PSL Social Status

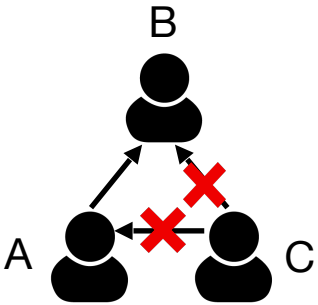
Cycle Structure

```
//Rules for cycle structure

  1:  Trusts(A,B) & Trusts(B,C)-> !Trusts(C,A)
  1:  !Trusts(A,B) & !Trusts(B,C)-> Trusts(C,A)
```



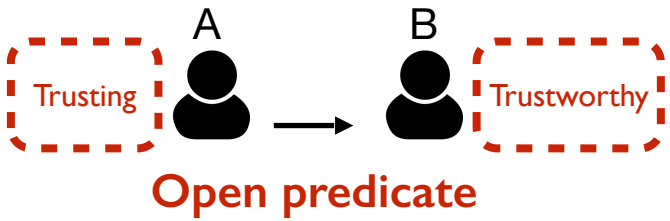
Non-Cycle Structure

```
//Rules for Non-cycle structure

  1:  Trusts(A,B) & !Trusts(C,B)-> !Trusts(C,A)
  1:  !Trusts(A,B) & Trusts(C,B)-> Trusts(C,A)
```



Latent Variable Model

- **Model #3: Latent model**




```
//Rules for latent model

  1:  Trusting(A)-> Trusts(A,B)
  1:  Trustworthy(B)-> Trusts(A,B)
  1:  Trusting(A) & Trustworthy(B) -> Trusts(A,B)
  1:  Trusts(A,B) -> Trusting(A)
  1:  Trusts(A,B)-> Trustworthy(B)
```

Hands on

- **Data**: 2K user sample of Epinions network and 8.7K signed trust relationships

- git clone https://github.com/linqs/psl-examples.git
- cd psl-examples/trust-prediction/cli
- git checkout uai18

- Models:
  - Balance
  - Status
  - Latent

PSL Model for Trust Prediction (Balance Theory)

```
//Rules for cycle and non-cyclic structure

1.0: Knows(A, B) & Knows(B, C) & Knows(A, C) & Trusts(A, B) & Trusts(B, C) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2
1.0: Knows(A, B) & Knows(B, C) & Knows(A, C) & Trusts(A, B) & !Trusts(B, C) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(A, B) & Knows(B, C) & Knows(A, C) & !Trusts(A, B) & Trusts(B, C) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(A, B) & Knows(B, C) & Knows(A, C) & !Trusts(A, B) & !Trusts(B, C) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2
1.0: Knows(A, B) & Knows(C, B) & Knows(A, C) & Trusts(A, B) & Trusts(C, B) & (A != B) & (B != C) & (A != C) -> Trusts(A, C)
1.0: Knows(A, B) & Knows(C, B) & Knows(A, C) & Trusts(A, B) & !Trusts(C, B) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(A, B) & Knows(C, B) & Knows(A, C) & !Trusts(A, B) & Trusts(C, B) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(A, B) & Knows(C, B) & Knows(A, C) & !Trusts(A, B) & !Trusts(C, B) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(B, C) & Knows(A, C) & Trusts(B, A) & Trusts(B, C) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(B, C) & Knows(A, C) & Trusts(B, A) & !Trusts(B, C) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(B, C) & Knows(A, C) & !Trusts(B, A) & Trusts(B, C) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(B, C) & Knows(A, C) & !Trusts(B, A) & !Trusts(B, C) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(C, B) & Knows(A, C) & Trusts(B, A) & Trusts(C, B) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(C, B) & Knows(A, C) & Trusts(B, A) & !Trusts(C, B) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(C, B) & Knows(A, C) & !Trusts(B, A) & Trusts(C, B) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(C, B) & Knows(A, C) & !Trusts(B, A) & !Trusts(C, B) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2

1.0: Knows(A, B) & Knows(B, A) & Trusts(A, B) -> Trusts(B, A) ^2
1.0: Knows(A, B) & Knows(B, A) & !Trusts(A, B) -> !Trusts(B, A) ^2
```

Data file for Trust prediction

predicates:

Trusts/2: open

Knows/2: closed

Prior/1: closed

observations:

Trusts: ../data/trust-prediction/eval/trusts_obs.txt

Knows: ../data/trust-prediction/eval/knows_obs.txt

Prior: ../data/trust-prediction/eval/prior_obs.txt

targets:

Trusts: ../data/trust-prediction/eval/trusts_target.txt

truth:

Trusts: ../data/trust-prediction/eval/trusts_truth.txt

PSL Model for Trust Prediction (Social Status)

```
//Rules for cycle and non-cyclic structure

1.0: Knows(A, B) & Knows(B, C) & Knows(A, C) & Trusts(A, B) & Trusts(B, C) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2
1.0: Knows(A, B) & Knows(B, C) & Knows(A, C) & !Trusts(A, B) & !Trusts(B, C) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(A, B) & Knows(C, B) & Knows(A, C) & Trusts(A, B) & !Trusts(C, B) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2
1.0: Knows(A, B) & Knows(C, B) & Knows(A, C) & !Trusts(A, B) & Trusts(C, B) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(B, C) & Knows(A, C) & Trusts(B, A) & !Trusts(B, C) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(B, C) & Knows(A, C) & !Trusts(B, A) & Trusts(B, C) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(C, B) & Knows(A, C) & Trusts(B, A) & Trusts(C, B) & (A != B) & (B != C) & (A != C) -> !Trusts(A, C) ^2
1.0: Knows(B, A) & Knows(C, B) & Knows(A, C) & !Trusts(B, A) & !Trusts(C, B) & (A != B) & (B != C) & (A != C) -> Trusts(A, C) ^2


1.0: Knows(A, B) & Knows(B, A) & Trusts(A, B) -> !Trusts(B, A) ^2
1.0: Knows(A, B) & Knows(B, A) & !Trusts(A, B) -> Trusts(B, A) ^2

// two-sided prior

1.0: Knows(A, B) & Prior('0') -> Trusts(A, B) ^2
1.0: Knows(A, B) & Trusts(A, B) -> Prior('0') ^2
```

Get the status model from:

- psl-examples/trust-prediction/cli/alternate-models

PSL Model for Trust Prediction (Latent)

```
//Latent trusting/trustworthy rules

1.0: Knows(A, B) & Trusting(A) -> Trusts(A, B) ^2
1.0: Knows(A, B) & Trustworthy(B) -> Trusts(A, B) ^2
1.0: Knows(A, B) & Trusting(A) & Trustworthy(B) -> Trusts(A, B) ^2
1.0: Knows(A, B) & Trusts(A, B) -> Trusting(A) ^2
1.0: Knows(A, B) & Trusts(A, B) -> Trustworthy(B) ^2

// two-sided prior

1.0: Knows(A, B) & Prior('0') -> Trusts(A, B) ^2
1.0: Knows(A, B) & Trusts(A, B) -> Prior('0') ^2

// negative prior
1.0:~Trusts(A, B) ^2
```

Get the latent model and data from:

- psl-examples/trust-prediction/cli/alternate-models

Data file for Trust prediction (Latent model)

predicates:

Trusts/2: open
Trusting/1: open
Trustworthy/1: open
Knows/2: closed
Prior/1: closed

observations:

Trusts: ../data/trust-prediction/eval/trusts_obs.txt
Knows: ../data/trust-prediction/eval/knows_obs.txt
Prior: ../data/trust-prediction/eval/prior_obs.txt

targets:

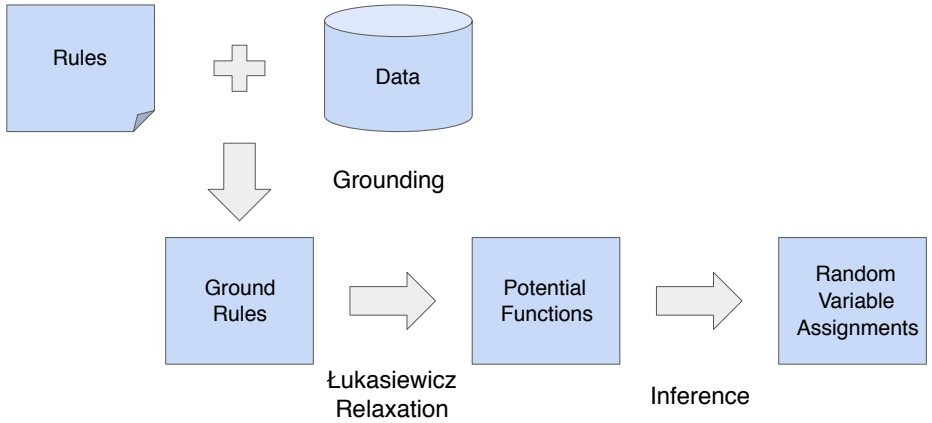
Trusts: ../data/trust-prediction/eval/trusts_target.txt

truth:

Trusts: ../data/trust-prediction/eval/trusts_truth.txt

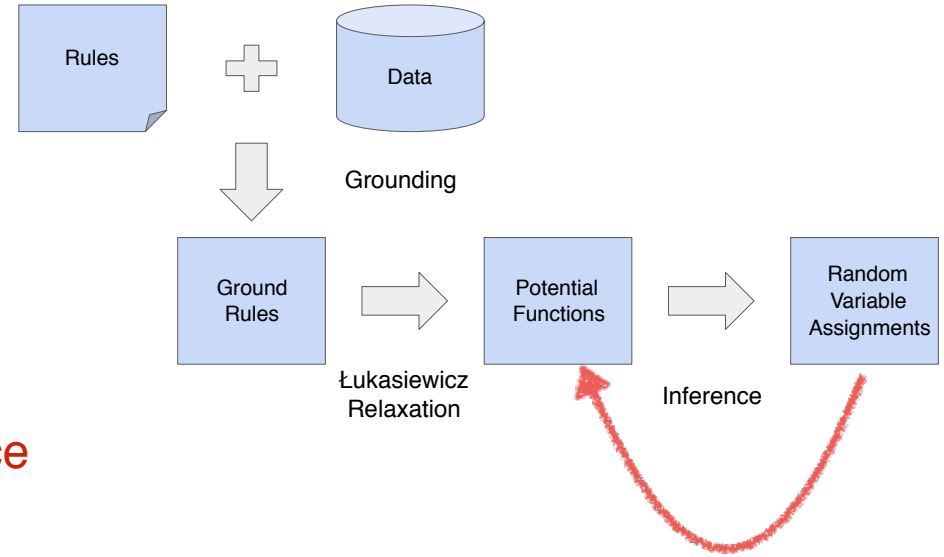
Inference for latent models

MAP Inference



Inference for latent models

MAP Inference



Lazy MAP inference

PSL Model for Trust Prediction (Latent)

```
//Latent trusting/trustworthy rules

1.0: Knows(A, B) & Trusting(A) -> Trusts(A, B) ^2
1.0: Knows(A, B) & Trustworthy(B) -> Trusts(A, B) ^2
1.0: Knows(A, B) & Trusting(A) & Trustworthy(B) -> Trusts(A, B) ^2
1.0: Knows(A, B) & Trusts(A, B) -> Trusting(A) ^2
1.0: Knows(A, B) & Trusts(A, B) -> Trustworthy(B) ^2

// two-sided prior

1.0: Knows(A, B) & Prior('0') -> Trusts(A, B) ^2
1.0: Knows(A, B) & Trusts(A, B) -> Prior('0') ^2

// negative prior
1.0:~Trusts(A, B) ^2
```

Change the parameter in run.sh:
—infer **org.linqs.psl.application.inference.LazyMPEInference**

Evaluation Result (Trust prediction- Epinions data)

Type of the model	AUC	AUC (positive)	AUC (negative)
Balance Theory	0.808961	0.973752	0.450463
Social Status	0.633428	0.946462	0.231061
Latent Model	0.917668	0.991246	0.557115

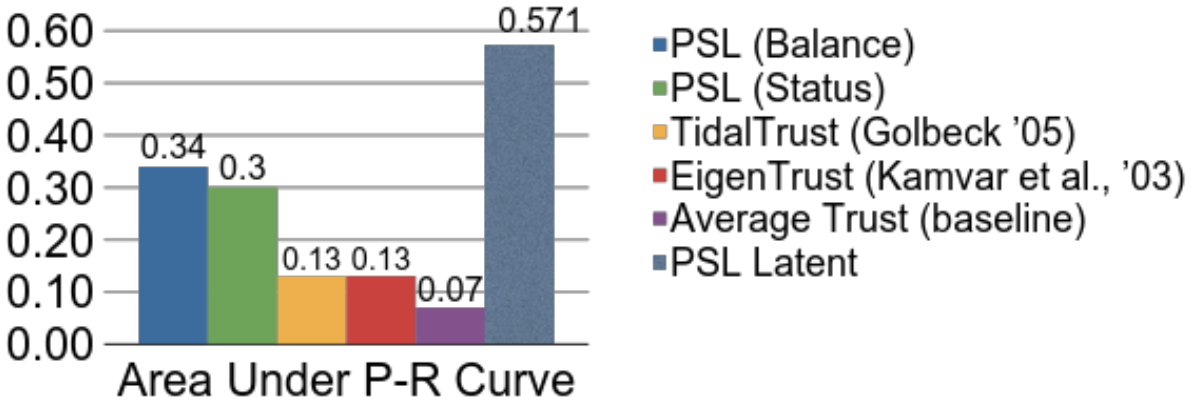
Evaluation Result (Trust prediction- Epinions data)

Type of the model	AUC	AUC (positive)	AUC (negative)
Balance Theory	0.808961	0.973752	0.450463
Social Status	0.633428	0.946462	0.231061
Latent Model	0.917668	0.991246	0.557115

? ? ? ?

Weight learning? **Other combinations?** **Different rules?**

Predicting Distrust



- **Data**: 2K user sample of Epinions network and 8.7K signed trust relationships
- 8-fold cross-validation
- Area under precision-recall curve for rarer **distrust links**

A Flexible Framework for Probabilistic Models of Social Trust, Huang, Kimmig, Getoor, Golbeck,
International Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction (SBP), 2013