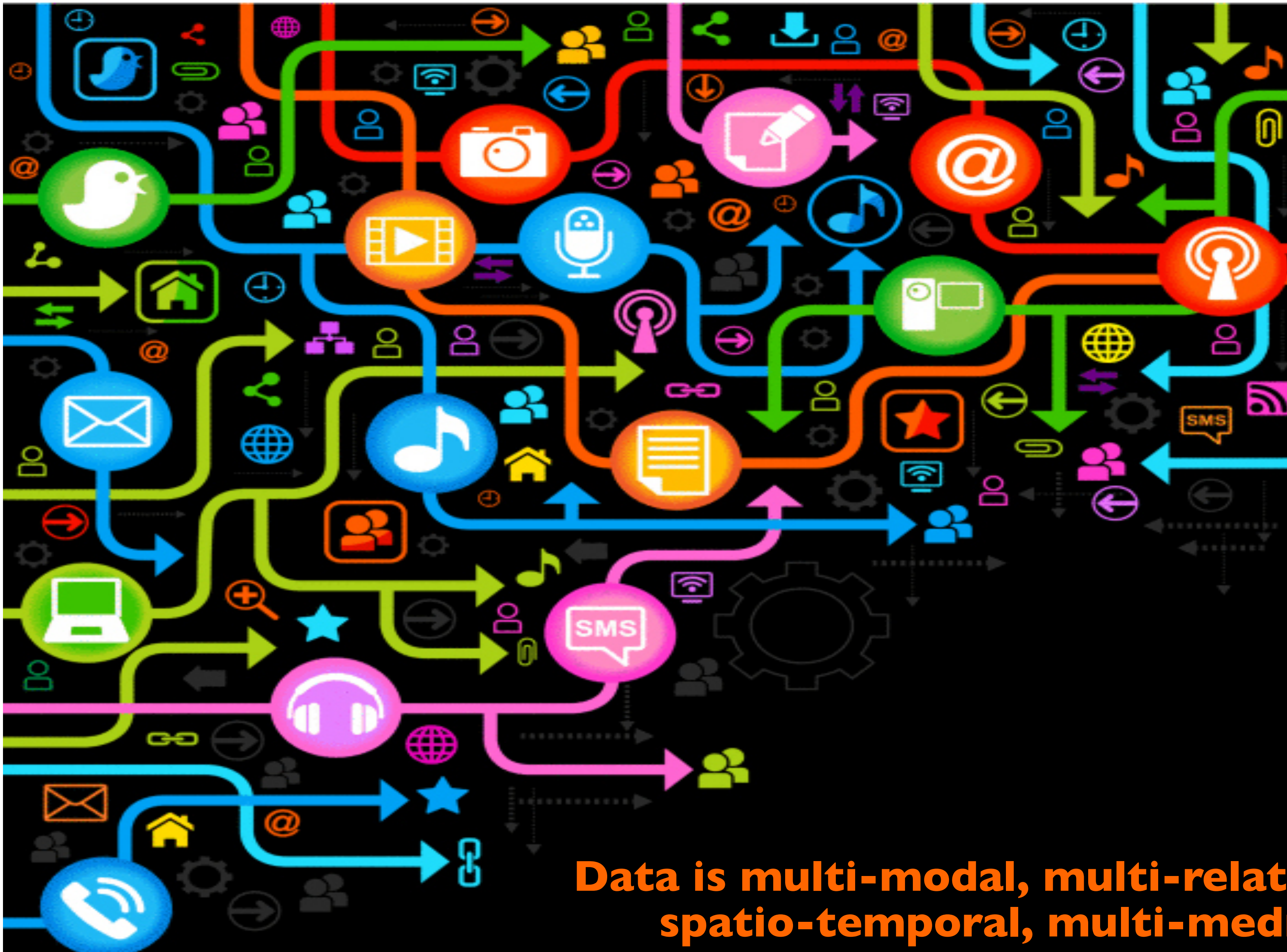
Scalable Machine Learning for Graphs

Lise Getoor

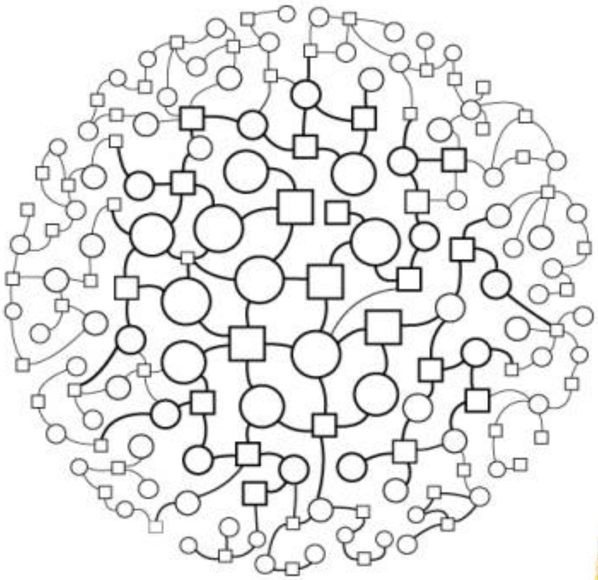
University of California, Santa Cruz

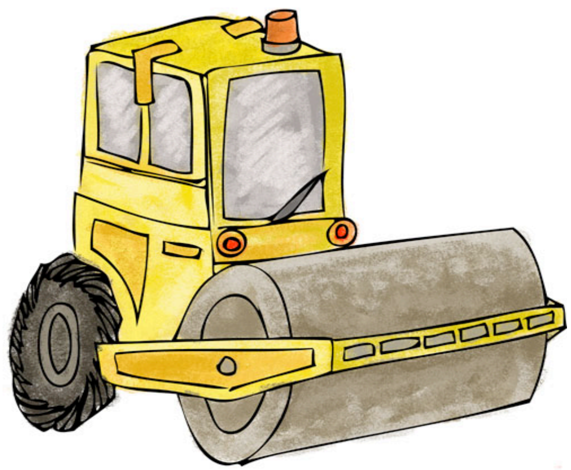
@lgetoor

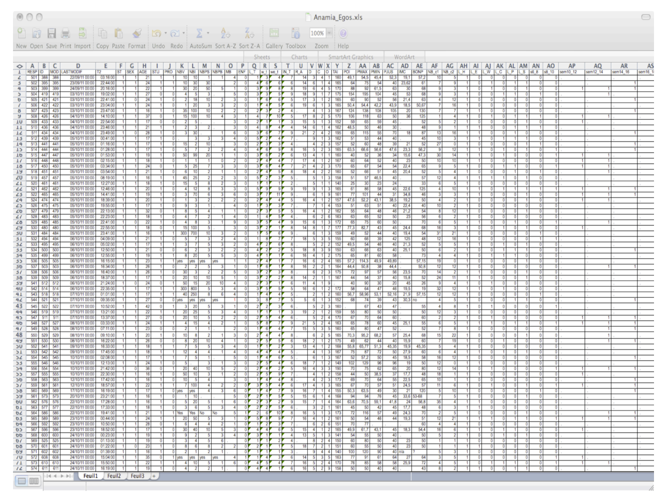
Stanford Scaled Machine Learning Conference
August 2, 2016



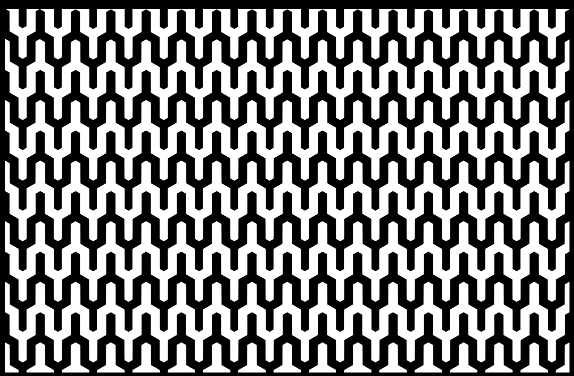
Data is multi-modal, multi-relational, spatio-temporal, multi-media






Anamia_Egos.xls

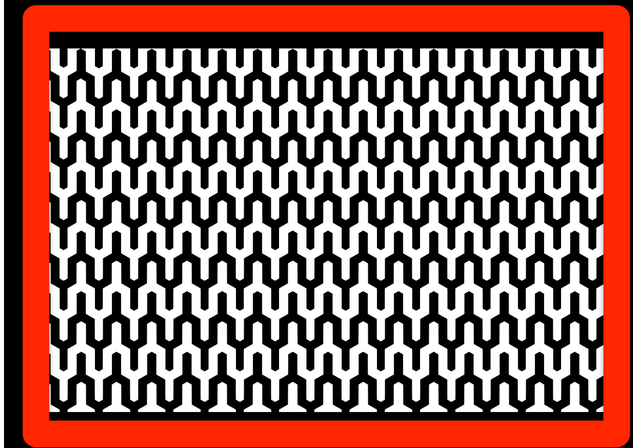
NEED:
Scalable Machine Learning for Graphs



Patterns

Key Ideas

Tools



Patterns

Key Ideas

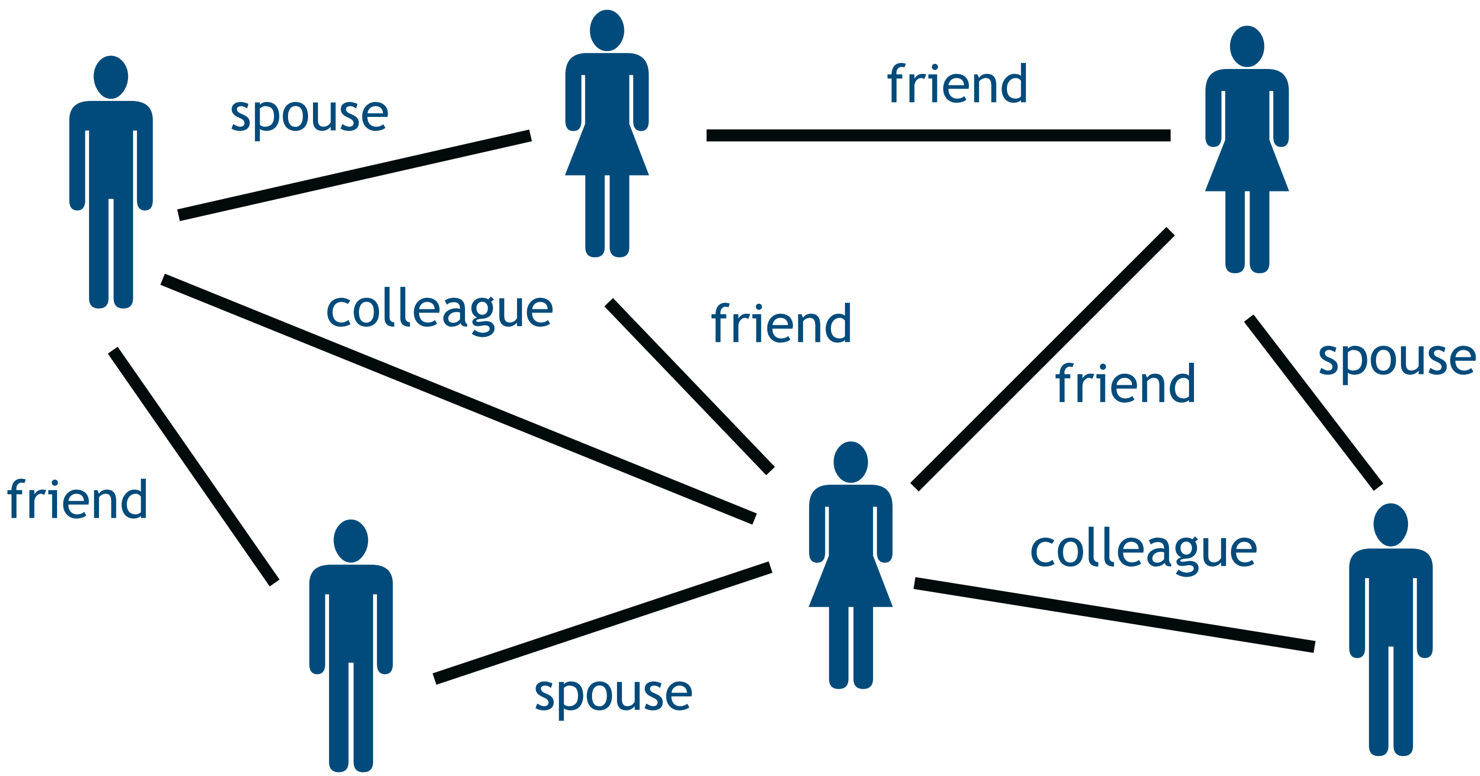
Tools

Common Graph Inference Patterns

- ☐ Collective Classification
- ☐ Link Prediction
- ☐ Entity Resolution

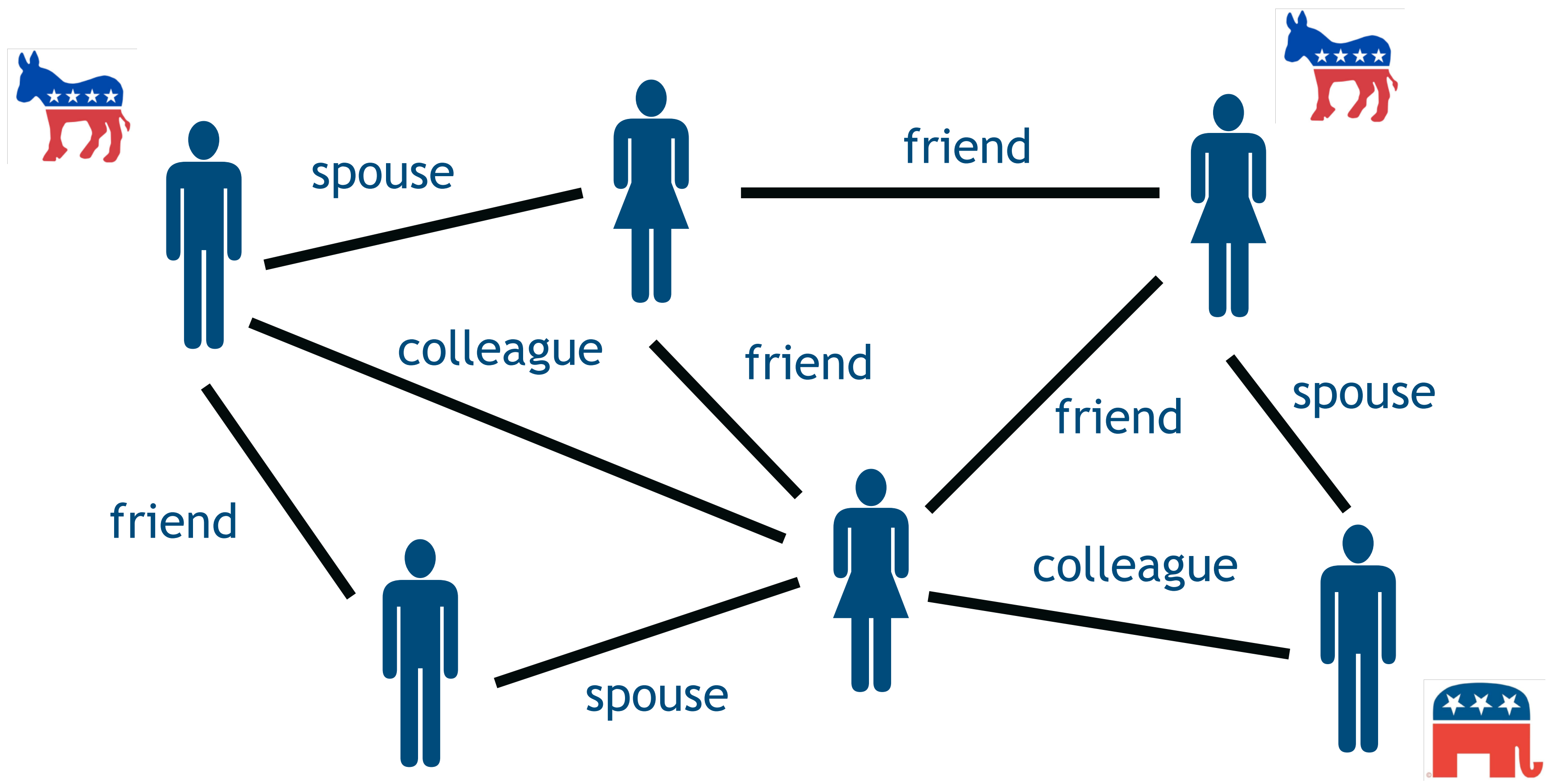
Collective Classification: inferring the labels of nodes in a graph

Collective Classification

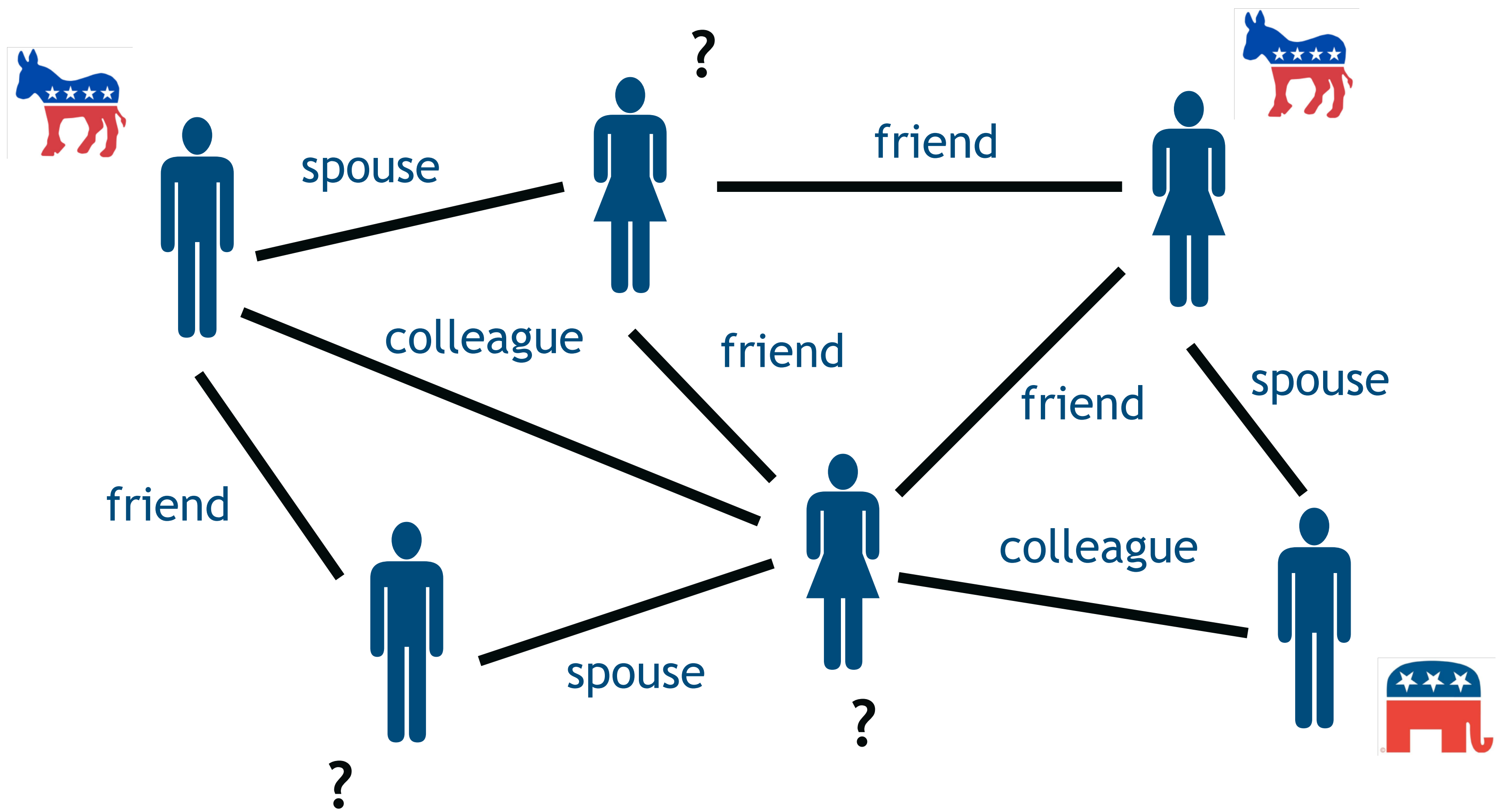


Question: or ?

Collective Classification



Collective Classification



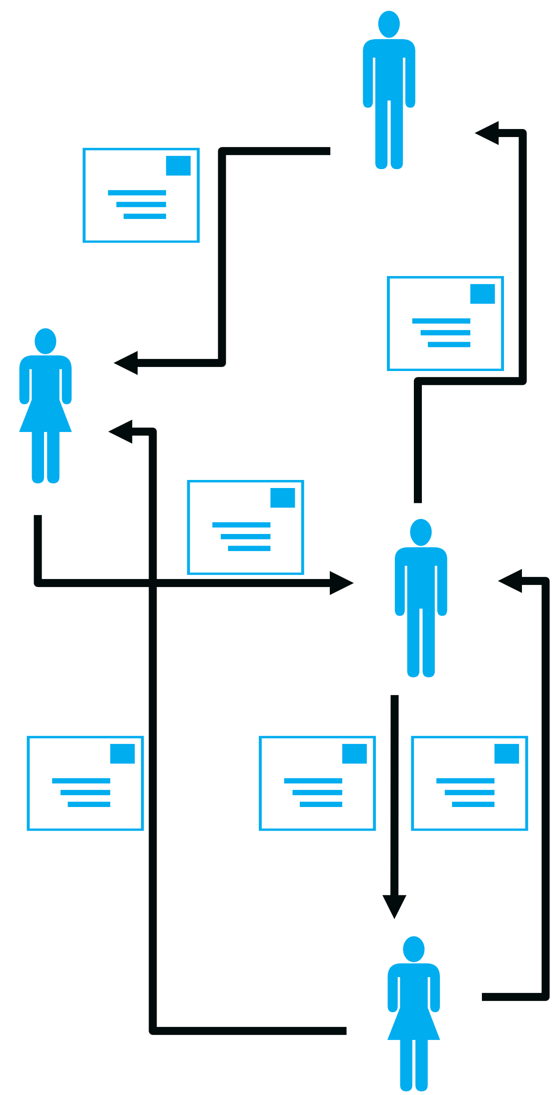
Common Graph Inference Patterns

- [x] Collective Classification
- [ ] Link Prediction
- [ ] Entity Resolution

Link Prediction: inferring the existence of edges in a graph

Link Prediction

- Entities
  - People, Emails
- Observed relationships
  - communications, co-location
- Predict relationships
  - Supervisor, subordinate, colleague

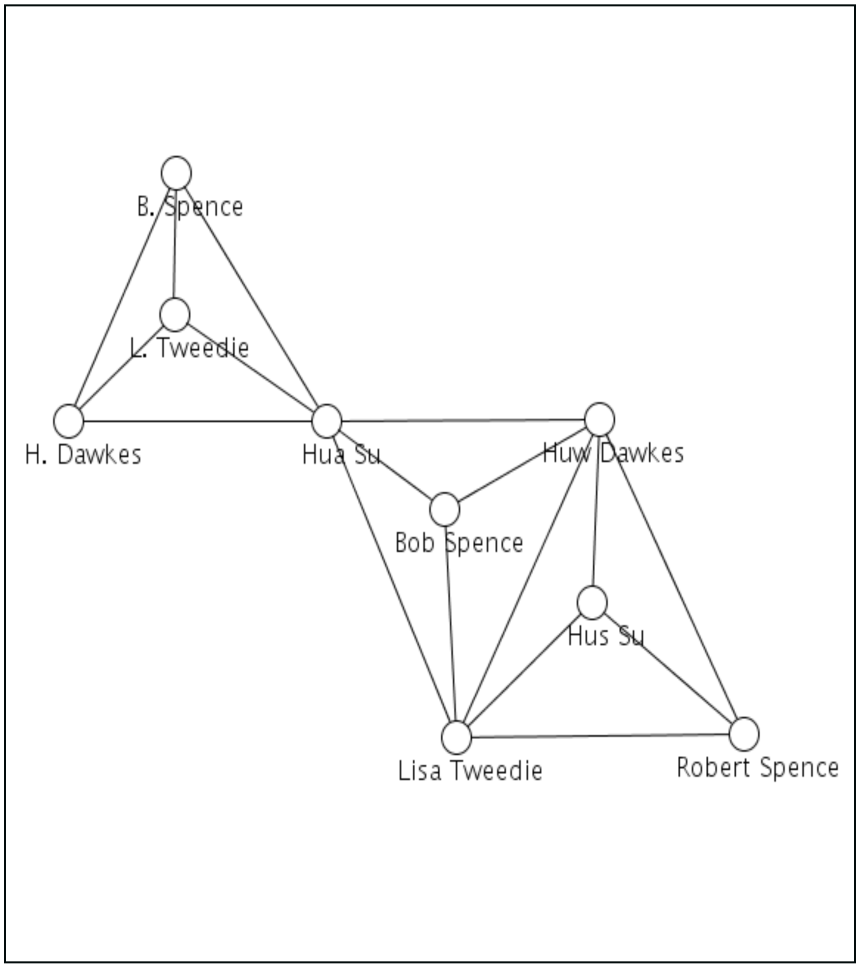


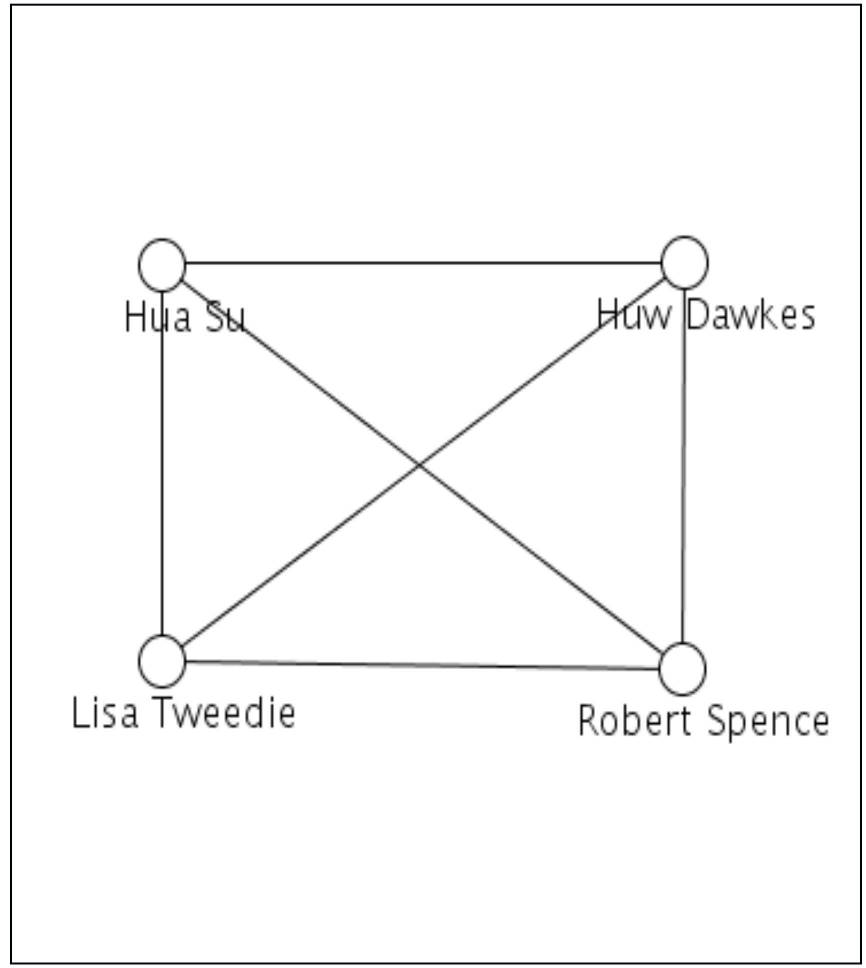
Common Graph Inference Patterns

- ☑ Collective Classification
- ☑ Link Prediction
- ☐ Entity Resolution

Entity Resolution: determining which nodes refer to same underlying entity

Entity Resolution & Network Analysis





before

after

Common Graph Inference Patterns

☑ Collective Classification

☑ Link Prediction

☑ Entity Resolution

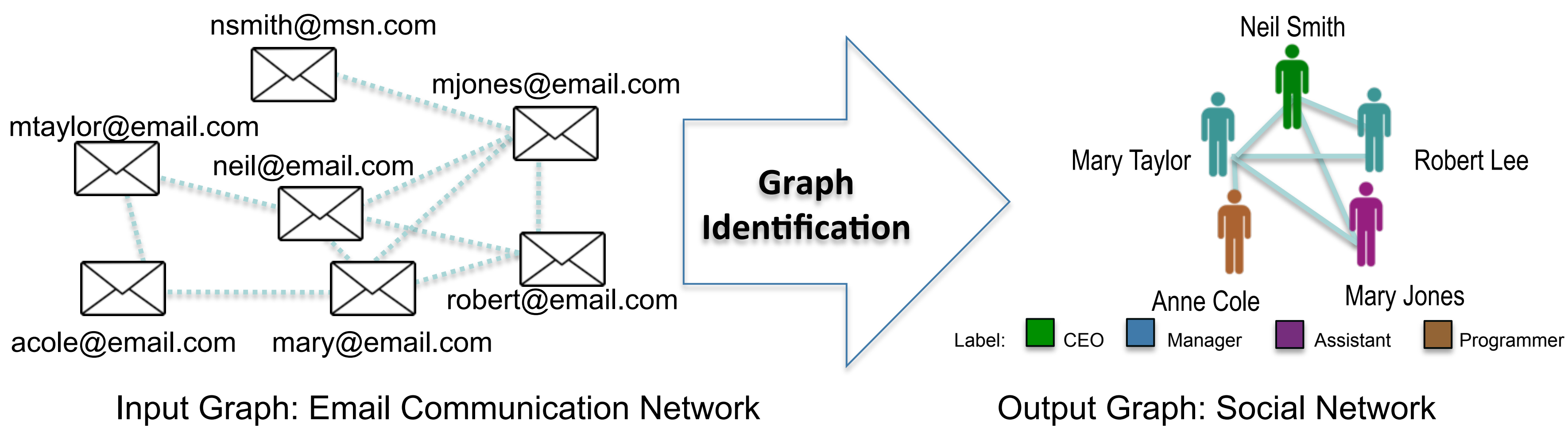
Namata, Kok, Getoor, TKDD 2015

My *favorite* graph inference pattern

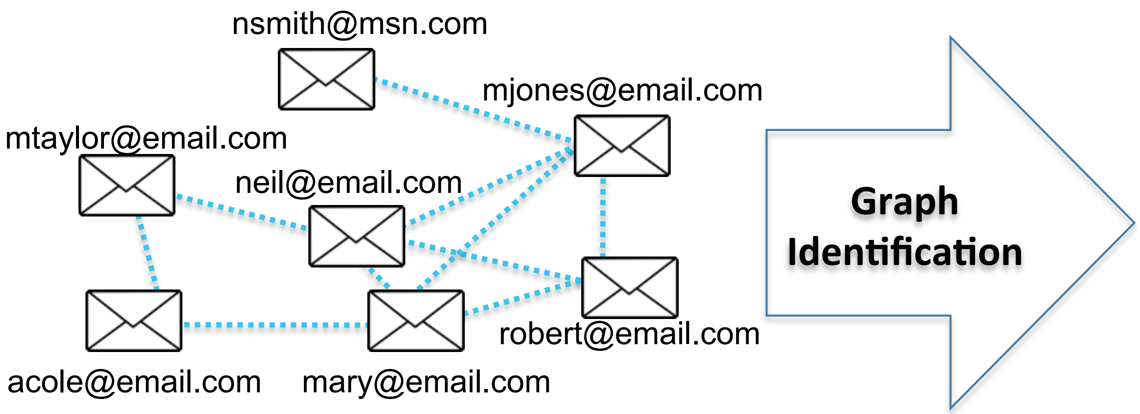
Graph Identification

combines ER, LP & CC

Graph Identification



Graph Identification

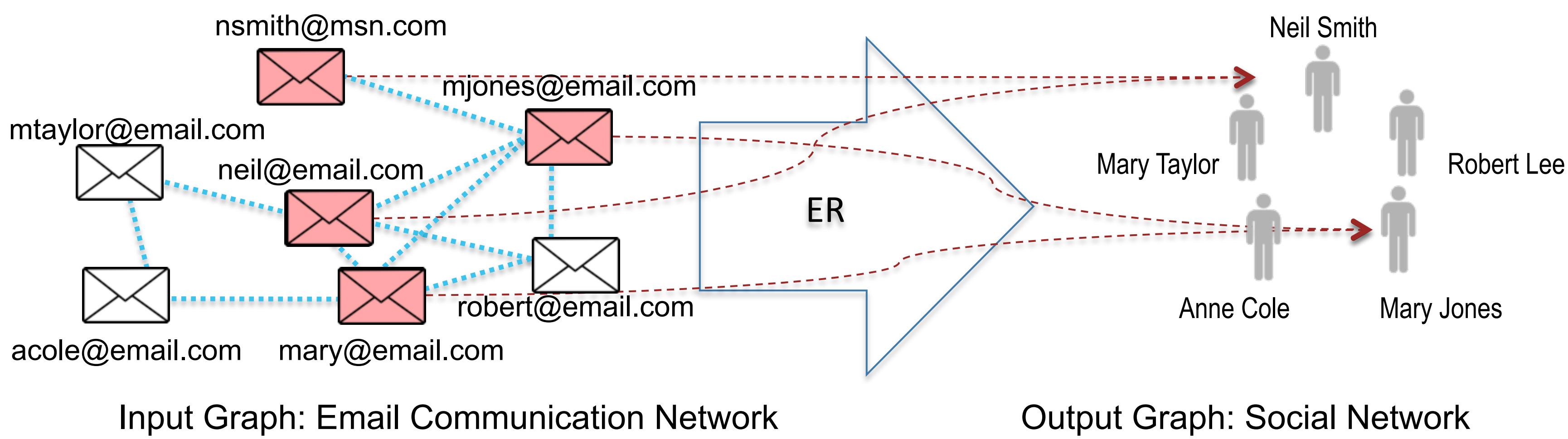


Input Graph: Email Communication Network

Output Graph: Social Network

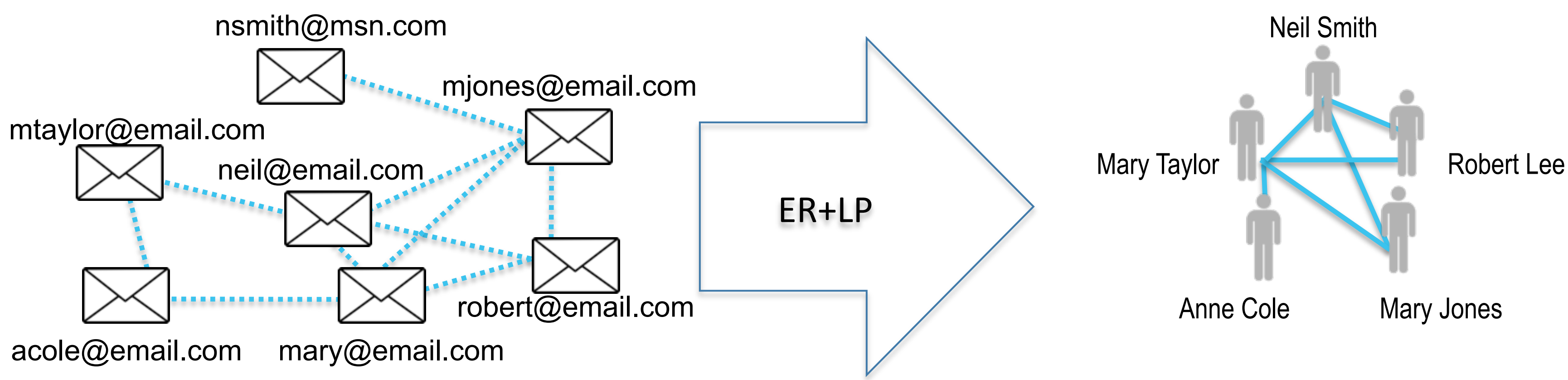
• What's involved?

Graph Identification



- What's involved?
  - Entity Resolution (ER): Map input graph nodes to output graph nodes

Graph Identification

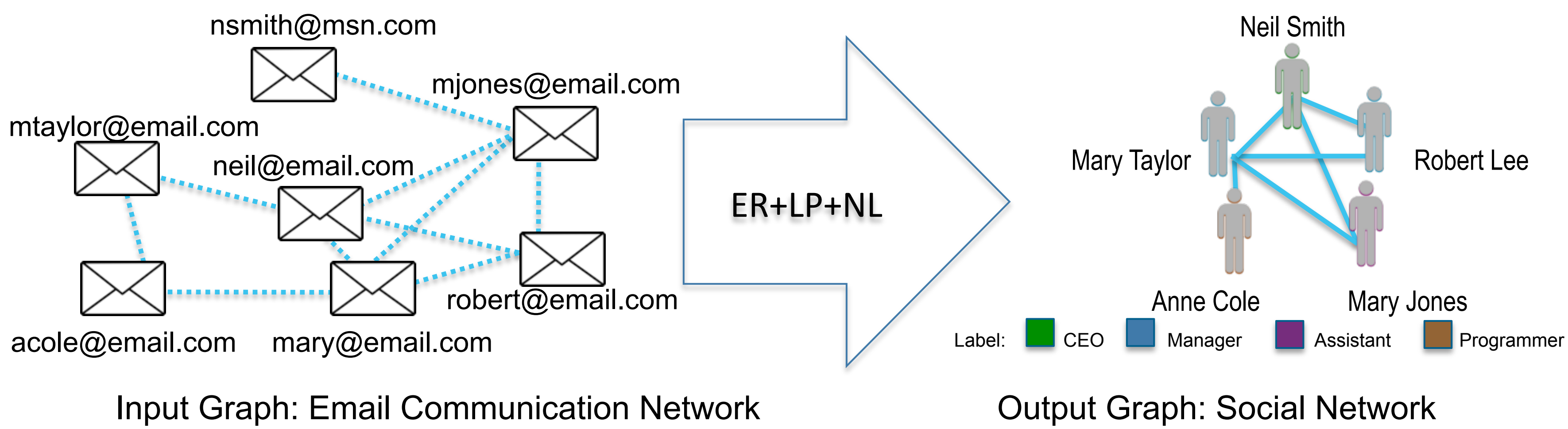


Input Graph: Email Communication Network

Output Graph: Social Network

- What's involved?
  - Entity Resolution (ER): Map input graph nodes to output graph nodes
  - Link Prediction (LP): Predict existence of edges in output graph

Graph Identification

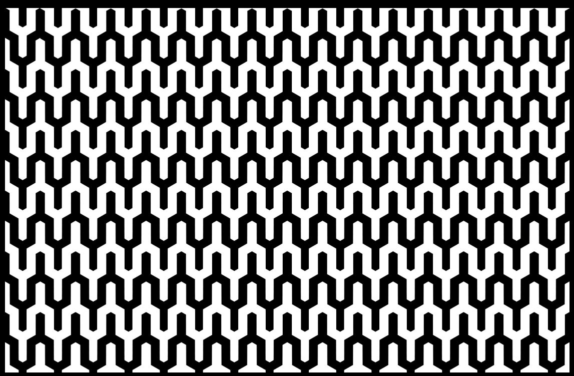


- What's involved?
  - Entity Resolution (ER): Map input graph nodes to output graph nodes
  - Link Prediction (LP): Predict existence of edges in output graph
  - Node Labeling (CC): Infer the labels of nodes in the output graph

Namata, Kok, Getoor, TKDD 2015

Graph Identification

- Goal:
  - Given an **input graph** infer an **output graph**
- Three major components:
  - **Entity Resolution (ER):** Infer the set of nodes
  - **Link Prediction (LP):** Infer the set of edges
  - **Collective Classification (CC):** Infer the node labels
- *Challenge: The components are intra and inter-dependent*



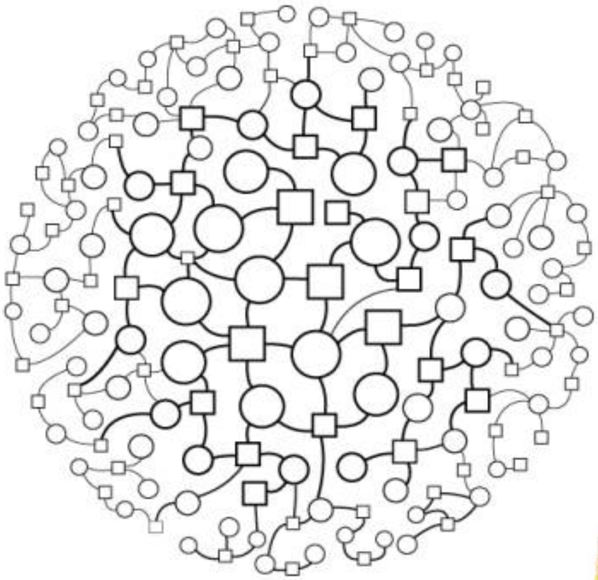
Patterns

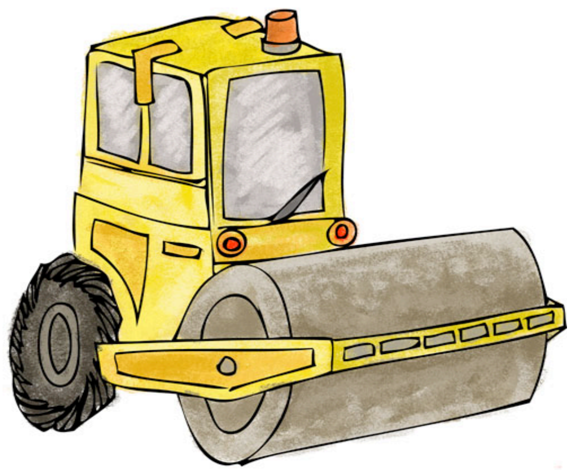
Key Ideas

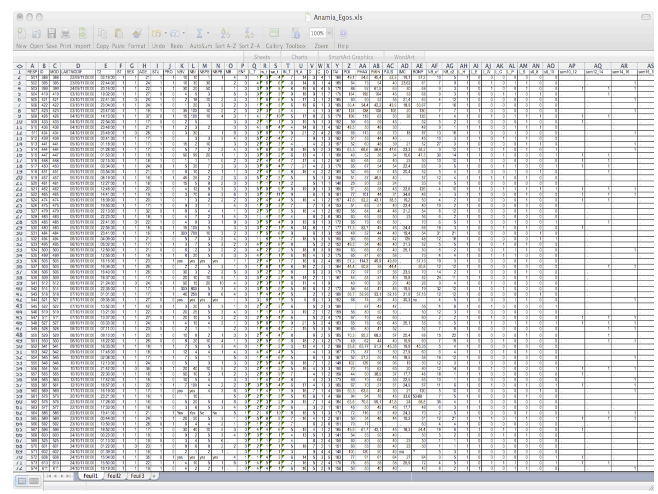
Tools

Key Ideas

- ☐ Feature Construction
- ☐ Collective Reasoning
- ☐ Lifted Models







Key Idea #1: Declarative Feature Construction

Feature Engineering

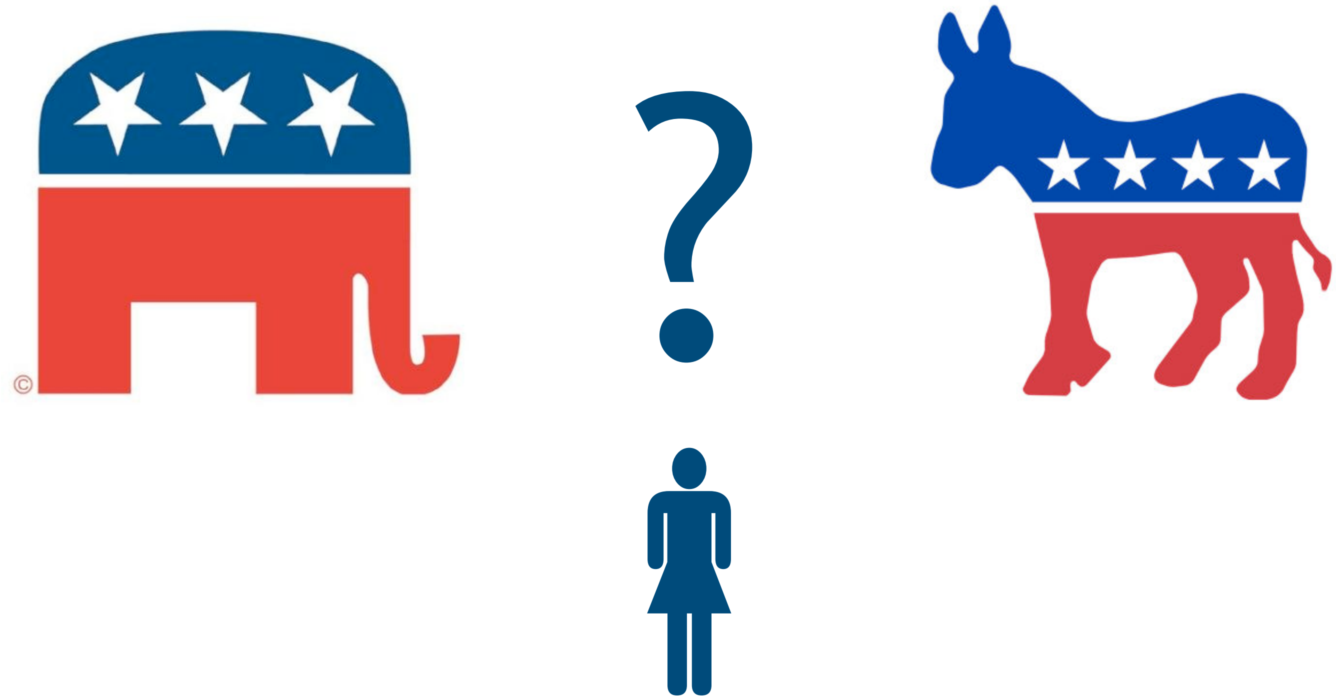
- Feature informativeness is key to the success
- In most settings, useful domain knowledge available
  - About node types, link types, attributes
  - Provide a declarative way of using of it!!
- Bring together techniques from DB, AI and ML

Opportunity!

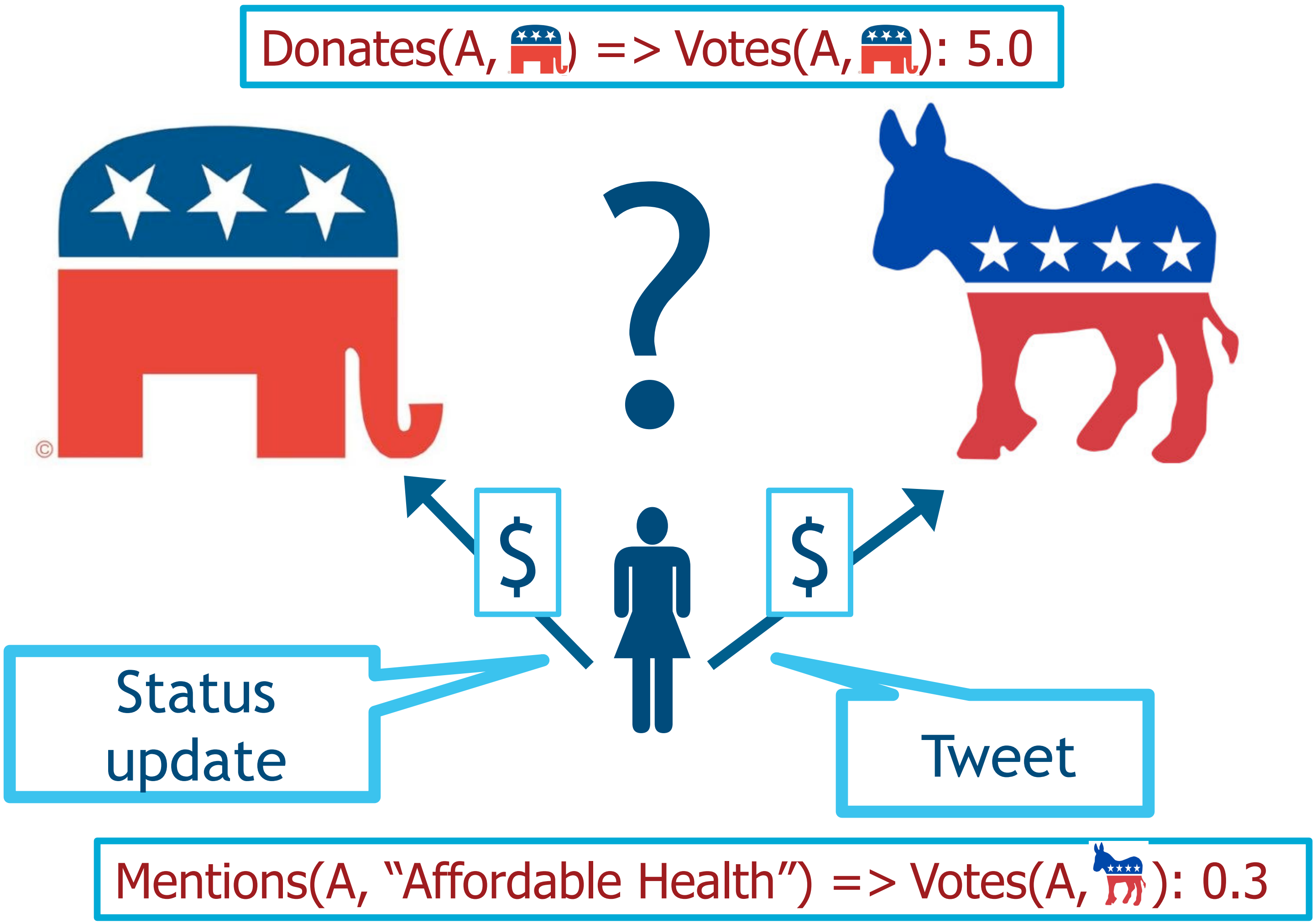
Key Ideas

- [x] Feature Construction
- [ ] Collective Reasoning
- [ ] Lifted Models

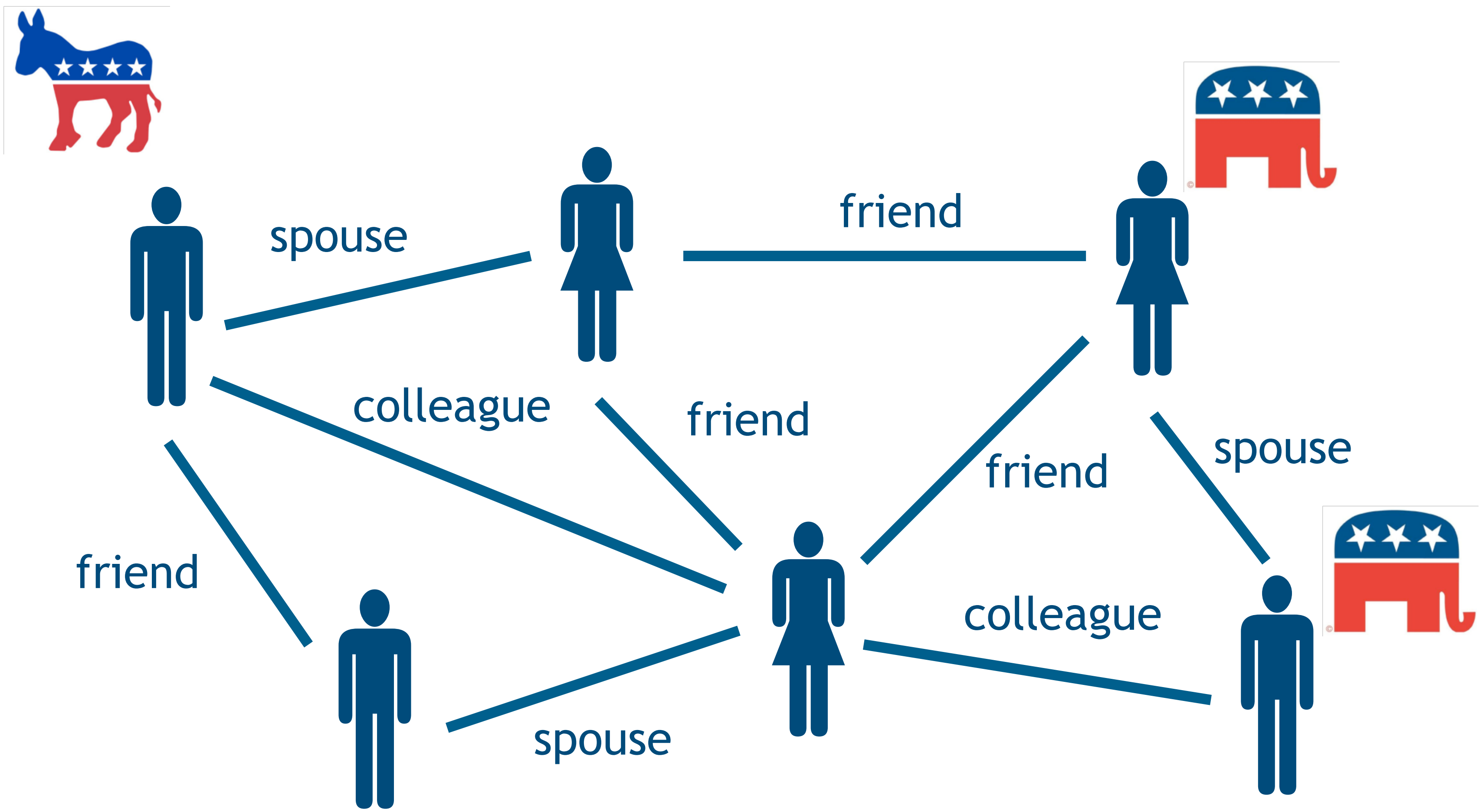
Collective Classification



Collective Classification

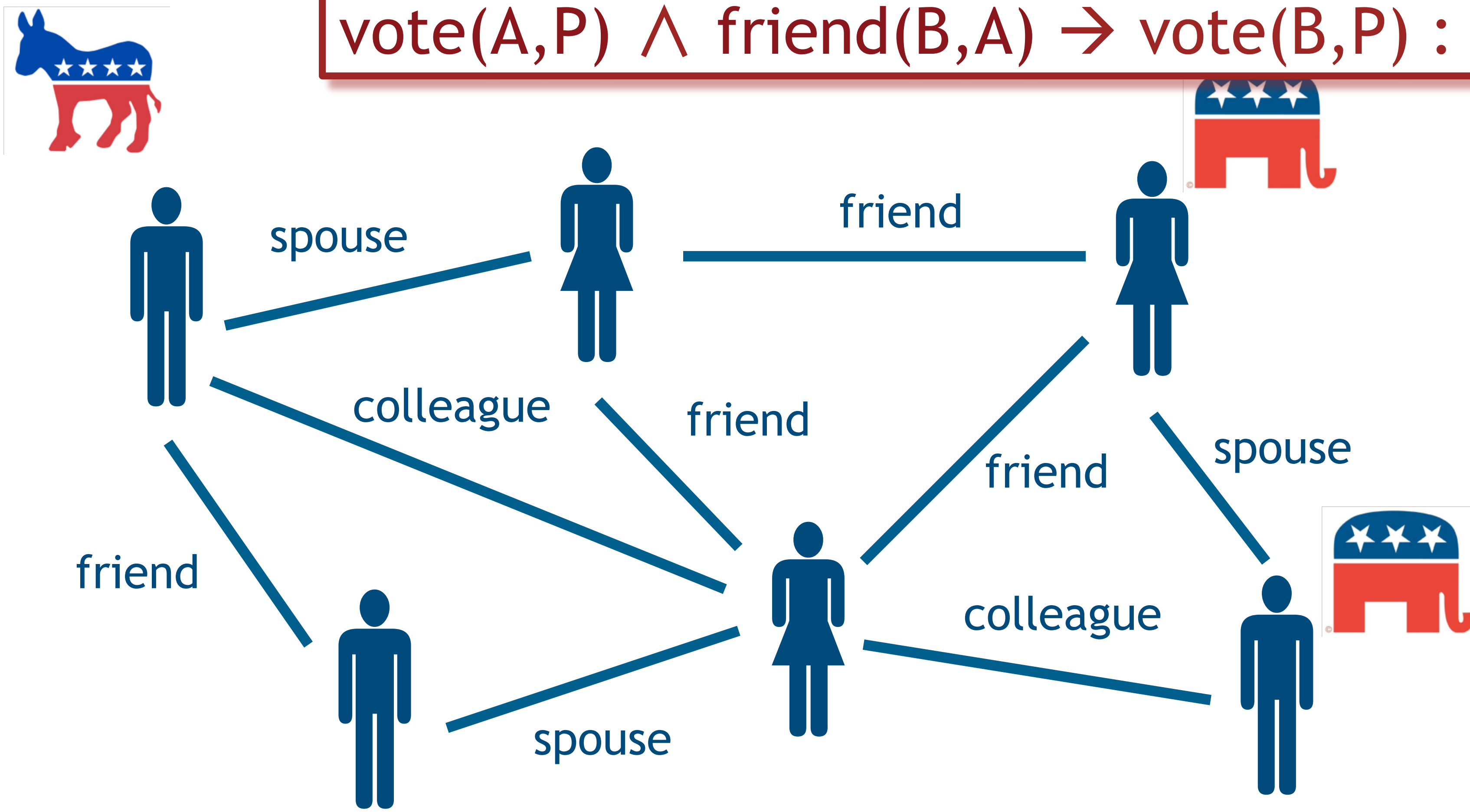


Collective Classification



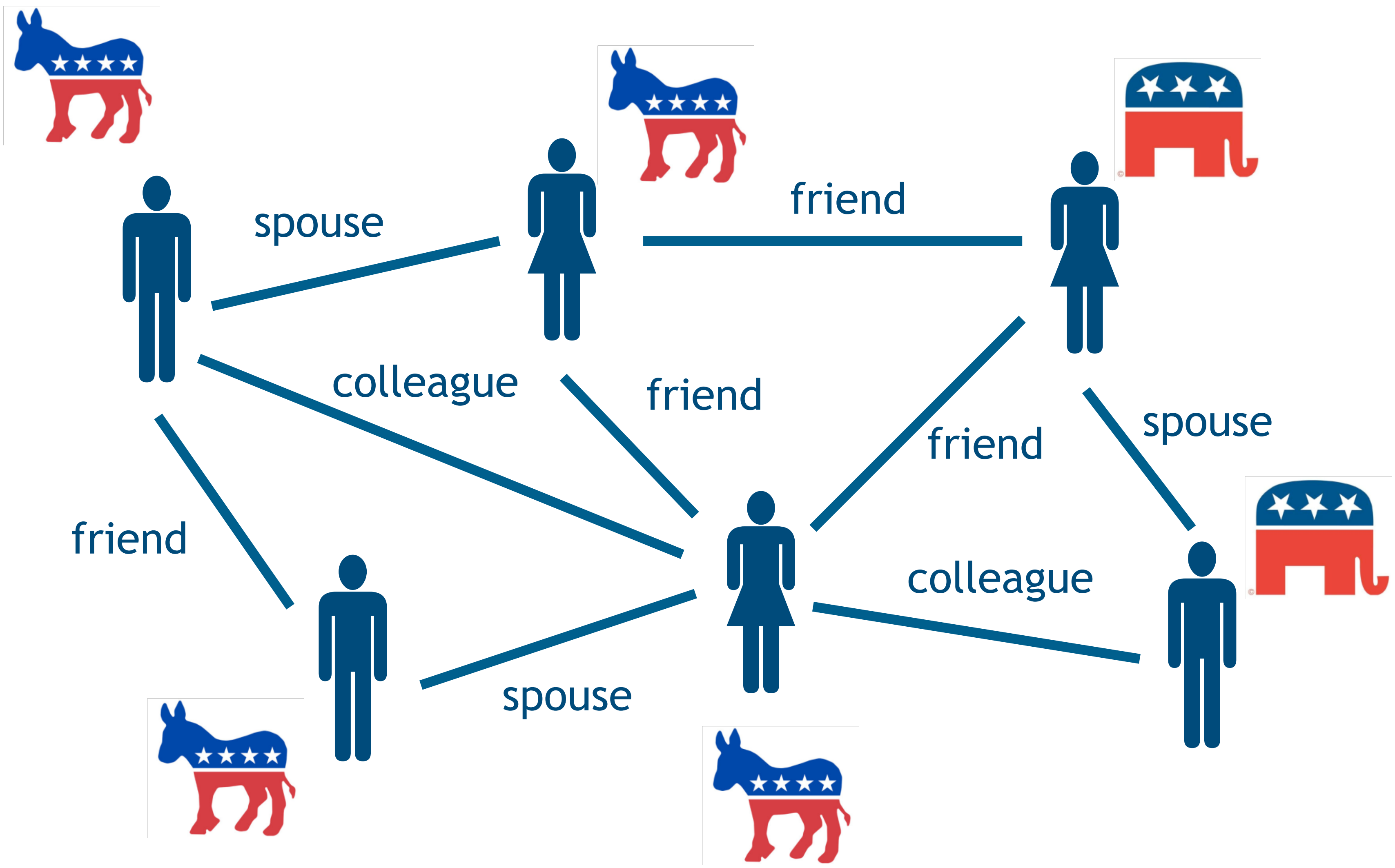
Collective Classification

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



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

Collective Classification



Key Ideas

- ☑ Feature Construction
- ☑ Collective Reasoning
- ☐ Lifted Models

Lifted Models

- Capture common, repeated patterns of dependencies

- Often, underlying semantics are defined using *Lifted Graphical Models*

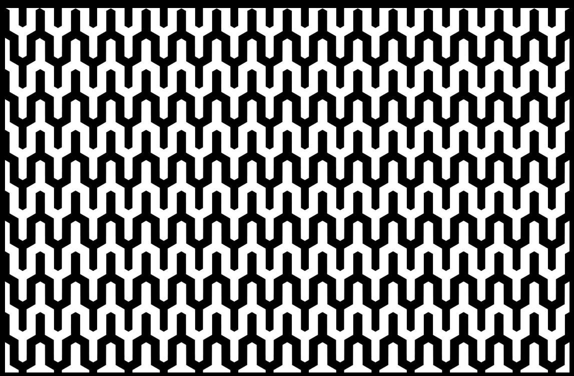
  Lifted Graphical Models: A Survey, Angelika Kimmig, Lily Mihalkova, Lise Getoor, Machine Learning Journal, 2014.

- Benefits:
  - fewer parameters, better learning and generalization
  - Inference can be made faster when one can exploit the commonalities

Opportunity!

Key Ideas

- ☑ Feature Construction
- ☑ Collective Reasoning
- ☑ Lifted Models


Patterns


Key Ideas


Tools

http://psl.umiacs.umd.edu

Probabilistic Soft Logic


Stephen Bach


Matthias Broecheler


Bert Huang


Alex Memory


Lily Mihalkova


Jimmy Foulds


Angelika Kimmig


Jay Pujara


Ben London


Arti Ramesh


Shobeir Fakhraei


Hui Miao


Dhanya Sridhar


Shachi Kumar

Probabilistic Soft Logic (PSL)

Declarative language based on logics to express collective probabilistic inference problems

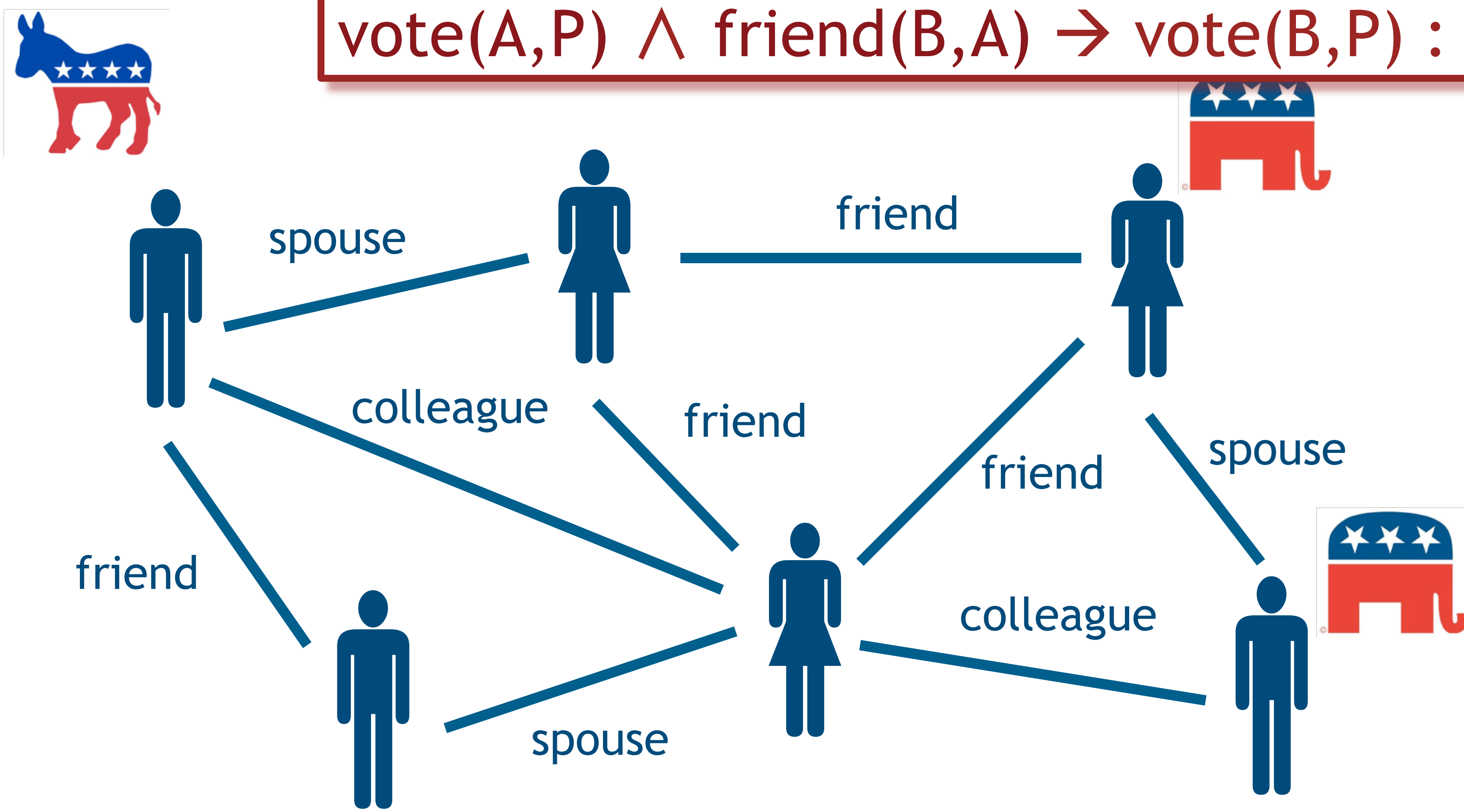
- Predicate = relationship or property
- Atom = **(continuous)** random variable
- Rule = capture dependency or **constraint**
- Set = define **aggregates**

PSL Program = Rules + Input DB

Reference*: Hinge-Loss Markov Random Fields and Probabilistic Soft Logic, Stephen H. Bach, Matthias Broecheler, Bert Huang, Lise Getoor, arXiv 2015*

Collective Classification

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



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

PSL Foundations

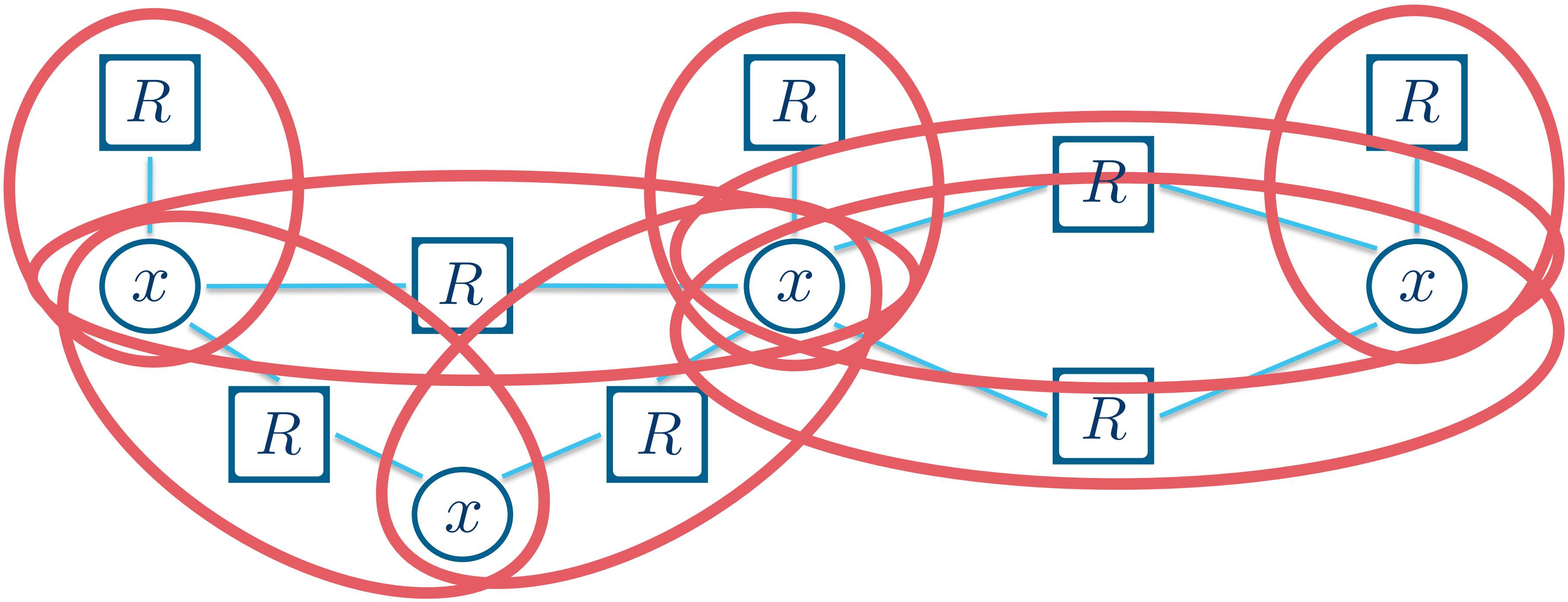
- PSL makes large-scale reasoning scalable by mapping logical rules to convex functions and defines a *hinge-loss Markov Random field*:

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

- Three principles justify this mapping [Bach et al., AIStats 14]:
  - LP programs for MAX SAT with approximation guarantees [Goemans & Williamson 94]
  - Pseudomarginal LP relaxations of Boolean Markov random fields [Wainwright et al. 02]
  - Łukasiewicz logic, a logic for reasoning about continuous values [Klir & Yuan 95]

Scalable Approximate Inference

- But linear programming algorithms do not scale well to big probabilistic models (Yanover et al. JMLR 06)



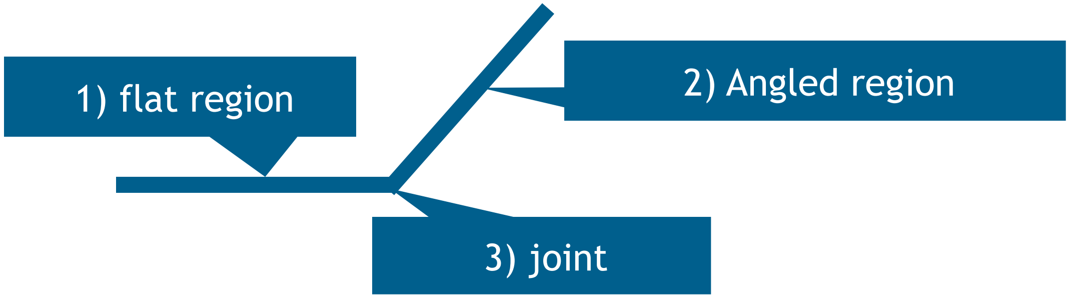
- Instead of solving the problem as one big optimization, **decompose** the problem based on its **graphical structure**

Scalable Inference with ADMM

- Alternating direction method of multipliers (ADMM)
  - Decompose a problem into subproblems
  - Iteratively solve and recombine the solutions via message passing
  - Guaranteed to converge for convex problems (Boyd et al., 2012)

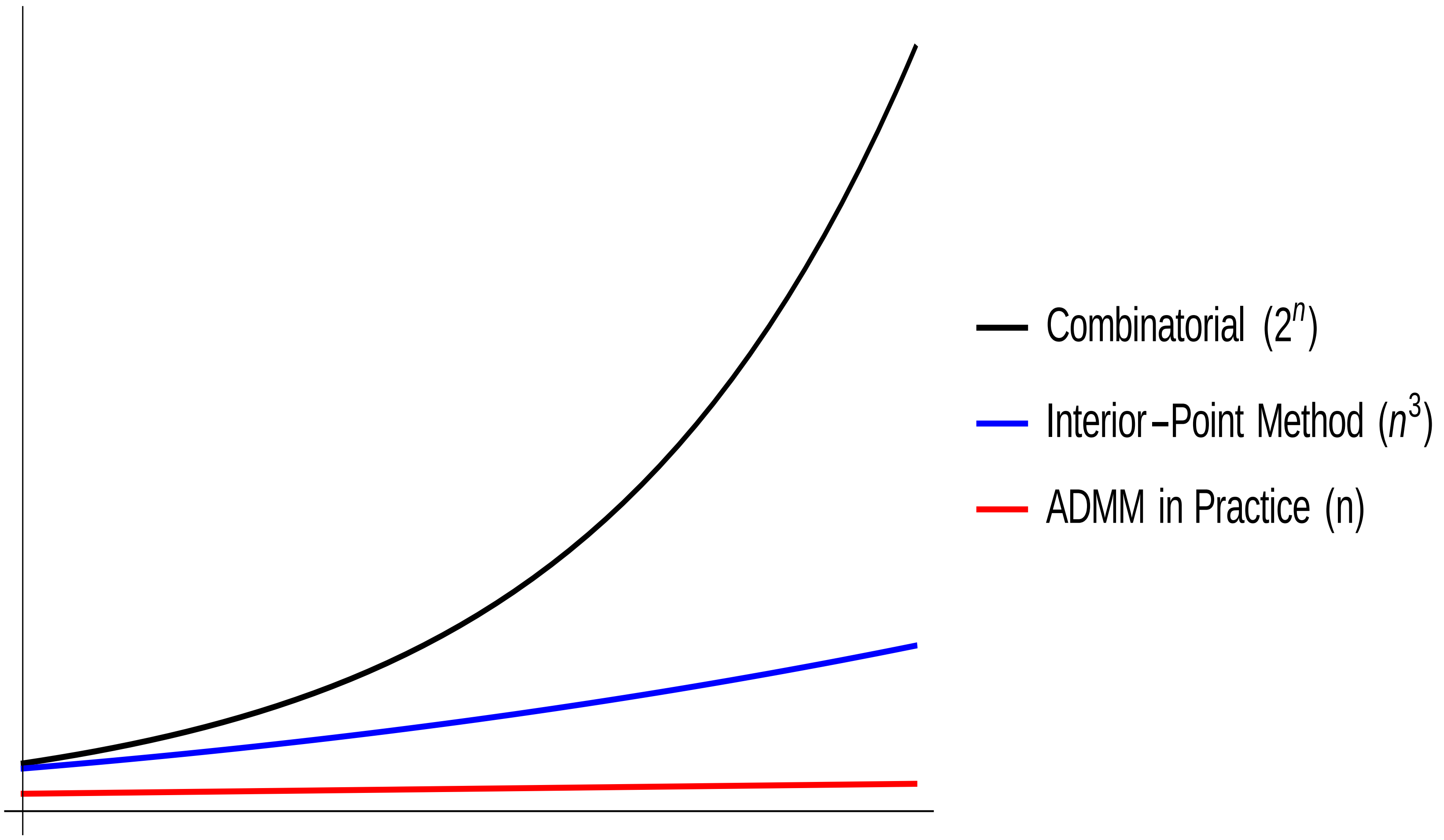
Scalable Inference with ADMM

- Alternating direction method of multipliers (ADMM)
  - Decompose a problem into subproblems
  - Iteratively solve and recombine the solutions via message passing
  - Guaranteed to converge for convex problems (Boyd et al., 2012)
- Important question: how to solve subproblems **quickly?**
  - Each subproblem is a hinge loss plus a quadratic penalty, and the solution must be one of the parts of the hinge:



  - Reduces to simple, often trivial, system of linear equations

Worst-Case Comparison



PSL Summary

Probabilistic
Soft
Logic

psl.umiacs.umd.edu

- PSL is a declarative probabilistic programming language for **collective inference problems on richly structured graph data**
- MAP Inference in PSL translates into convex optimization problem -> **inference is really fast**
- Inference further enhanced with **state-of-the-art optimization and distributed processing paradigms** such as ADMM & GraphLab -> **inference even faster**
- **Outperforms discrete MRFs** in speed and often accuracy
- Support for weight learning, latent variables, and more
- **PSL is *flexible***: many applications, looking for more
- PSL is **open-source**, code, data, tutorials available online

Sample Applications

Collective Classification with PSL

```
/* Local rules */
5.0: Donates(A, P) -> Votes(A, P)
0.3: Mentions(A, “Affordable Health”) -> Votes(A, “Democrat”)
0.3: Mentions(A, “Tax Cuts”) -> Votes(A, “Republican”)

/* Relational rules */
1.0: Votes(A,P) & Spouse(B,A) -> Votes(B,P)
0.3: Votes(A,P) & Friend(B,A) -> Votes(B,P)
0.1: Votes(A,P) & Colleague(B,A) -> Votes(B,P)

/* Range constraint */
Votes(A, “Republican”) + Votes(A, “Democrat”) = 1.0 .
```

Link Prediction with PSL

```
/* Message classification rules */
0.6 : HasWord(E, “due”) -> Type(E, “deadline”)
. . .

/* Link prediction rules */
0.8 : Sends(A,B,E) &Type(E, “deadline”) -> Supervisor(A,B)
. . .
0.8: Supervisor(A,B) & Supervisor(A,C) -> Colleague(B,C)

/* “Priors” */
0.01: !Supervisor(A, B)
0.01: !Colleague(A, B)

/* Domain and range constraints */
Type(M, +T) = 1.0
Supervisor(+A, B) <= 1.0
```

Entity Resolution with PSL

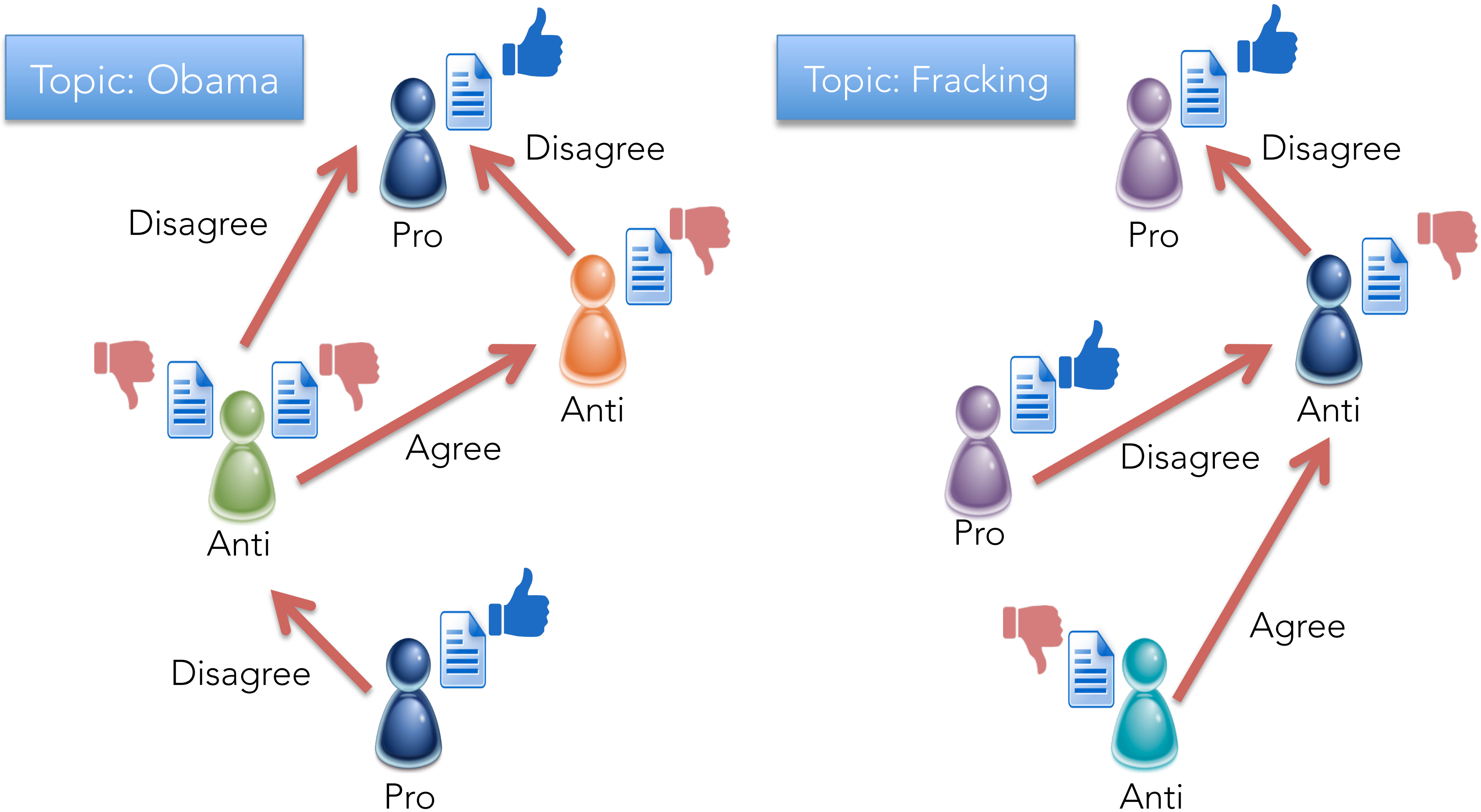
```
/* Entity resolution rules */
0.8: SameBlock(A,B) & Name(A, N1) & Name(B, N2) & Similar(N1, N2) ->
Same(A, B)
0.6: SameBlock(A,B) & SimilarFriendSet(A, B) -> Same(A, B)
SameBlock(A, C) & Same(A, B) && Same(B, C) -> Same(A, C) .

/* “Prior” */
0.01 : !Same(A, B)

/* Domain and range constraints */
Same(A, +B) <= 1.0
{SameBlock(A, B)}
Same(+A, B) <= 1.0
{SameBlock(A, B)}
```

PSL Empirical Highlights

- Compared with discrete MRFs:

	Collective Classification		Trust Prediction	
PSL	81.8%	0.7 sec	.482 AuPR	0.32 sec
Discrete	79.7%	184.3 sec	.441 AuPR	212.36 sec

Debate Stance Classification



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

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

https://github.com/linqs/stance_class/

Predicting Stance on Online Forums

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

4Forums.com

	User Stance Accuracy	Post Stance Accuracy
Logistic Regression Baseline	72.0	69.0
PSL-Stance	**77.1**	**80.3**

CreateDebate.org

	User Stance Accuracy	Post Stance Accuracy
Logistic Regression Baseline	70.2	62.7
PSL-Stance	**74.0**	**72.7**

Sridhar et al., Joint Models of Disagreement and Stance, ACL 2015

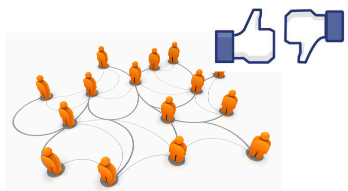
Hybrid Recommender Systems

Improve recommendations by combining
data sources & recommenders

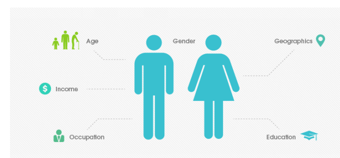
ratings

content

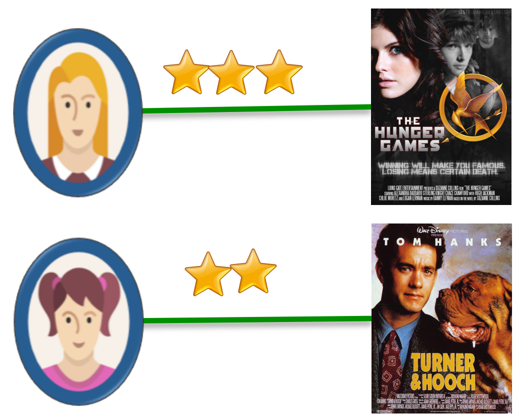
social



demographic



Predicted Ratings



Hybrid Recommender (HyPER)

Matrix Factorization

Item-based Collaborative Filtering

. . .

Bayesian Probabilistic Matrix Factorization

. . .

PSL for 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

Predicting Ratings with HyPER

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**

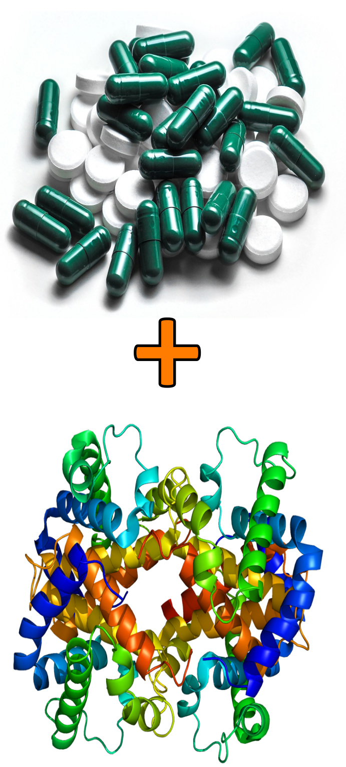
last.fm

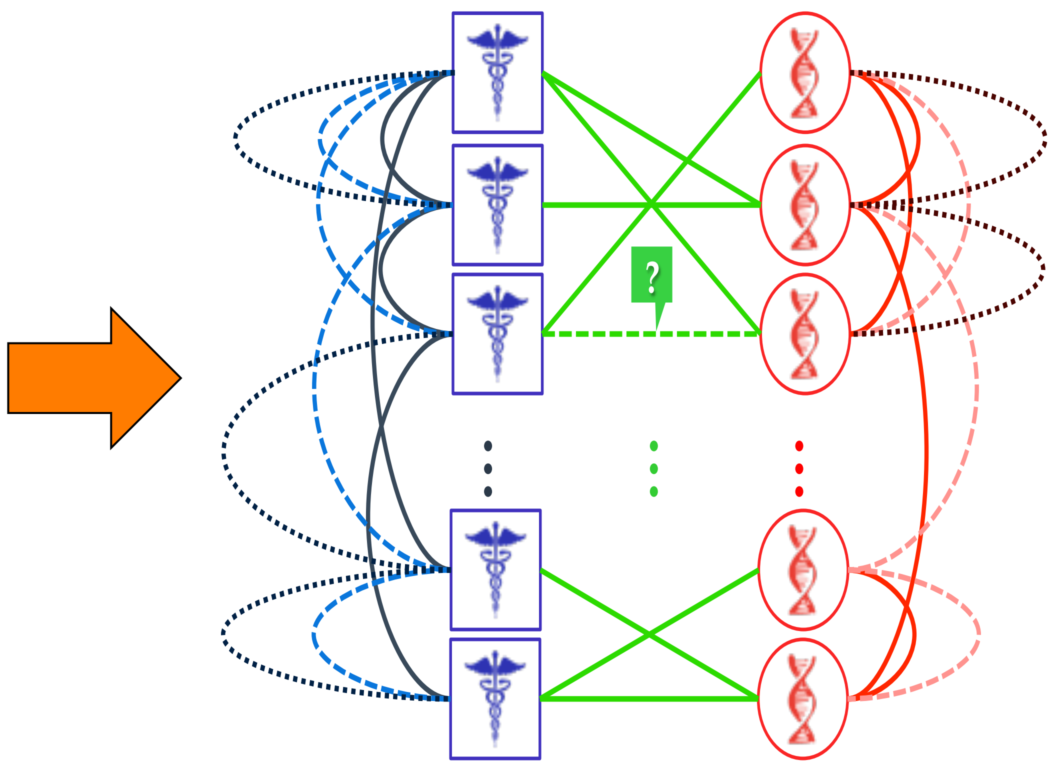
Model	RMSE
Item-based	1.408
MF	1.178
BPMF	1.008
Naïve Hybrid	1.067
BPMF-SRIC	1.015
HyPER	**1.001**

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

Drug Discovery

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





PSL for Drug-Protein Interaction Prediction

```
//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

Predicting New Interactions in Drugbank

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

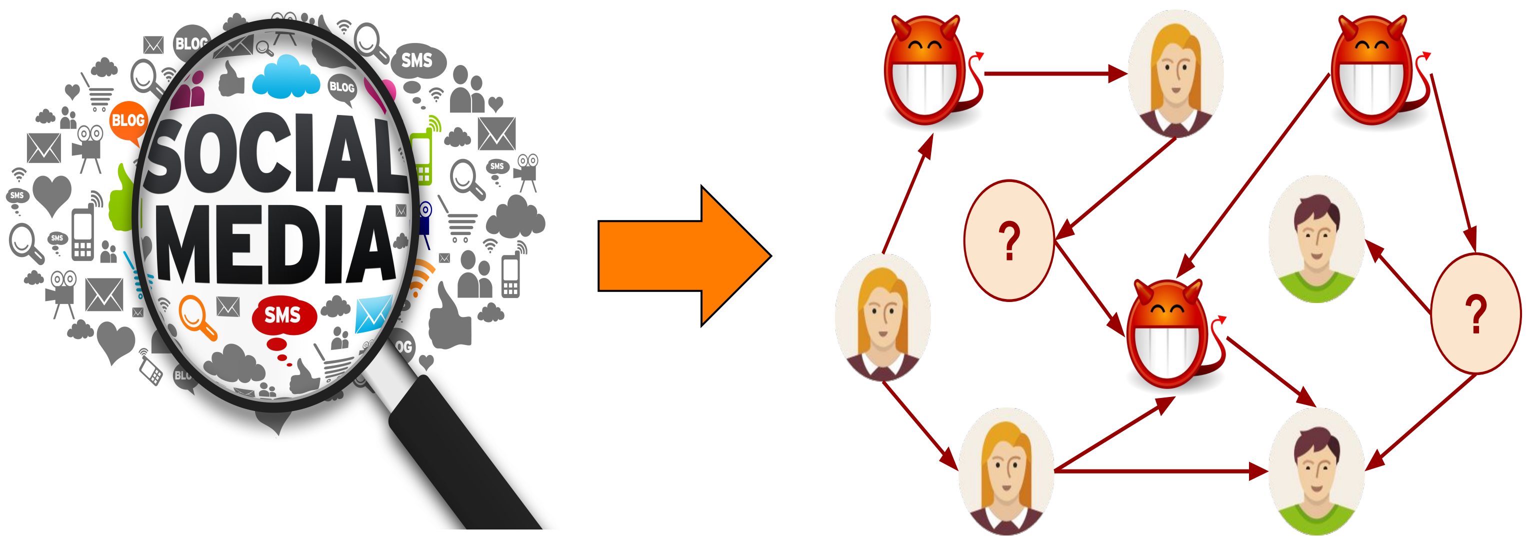
Newly Discovered Interactions DRUGBANK Open Data Drug & Drug Target Database

	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. (Featured on the cover)

Social Spammer Detection



Importance:

- 1 in 200 social messages contain spam
- Social spam grew by more than 350% between Jan-Jul 2013

PSL for Social Spammer Detection

```
//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

Finding Social Spammers in Tagged.com

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

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**

Finding the 4% spammers out of 116,284 users

"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

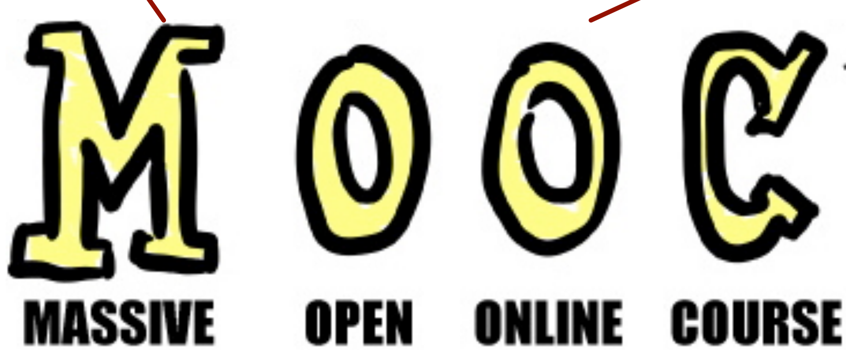
Modeling Student Engagement in MOOCs

Motivation: Help MOOC instructors understand student activity

Problem: How to model student engagement in MOOCs?

- Large number of registrants
- Low completion rate

Model engagement using
- Online behavior
- Linguistic analysis of posts
- structural attributes from forum interaction



PSL for Latent Student Engagement

```
//Behavioral
100: VoteActivity(U)   & ViewActivity(U)      ->  EngagementPassive(U)
100: PostActivity(U)                          ->  EngagementActive(U)

//Linguistic
100: Posts(U,P)   & Positive(P)    ->  EngagementActive(U)
100: Posts(L1,L2)  & Upvote(P)     ->  EngagementActive(U)
100: Posts(U,P)   & !Positive(P)   ->  !EngagementActive(U)

//Structural
100: Posts(U1,P1) & Posts(U2,P2) & SameThread(P1,P2) & EngagementActive(U1)
                                                                       ->
EngagementActive(U2)
//Temporal
100: LastPost(U, “start”)   & LastQuiz(U,“start”) ->  Disengaged(U)

//Engagement and Course Success
100: EngagementActive(U) & EngagementPassive(U)    ->  CourseSuccess(U)
100: Disengaged(U)                                 ->  !CourseSuccess(U)
```

https://github.com/artir/ramesh-aaai14/

Course Success Prediction

Task: Predict course performance and completion of MOOC students using latent engagement model

Predicting Course Performance

Model	AUC (+ve)	AUC (-ve)
Baseline	0.263	0.761
DIRECT	0.794	0.881
LATENT	**0.922**	**0.950**

Predicting Course Completion

Model	AUC (+ve)	AUC (-ve)
Baseline	0.508	0.995
DIRECT	0.565	0.995
LATENT	**0.816**	**0.998**

# of students: 200,000

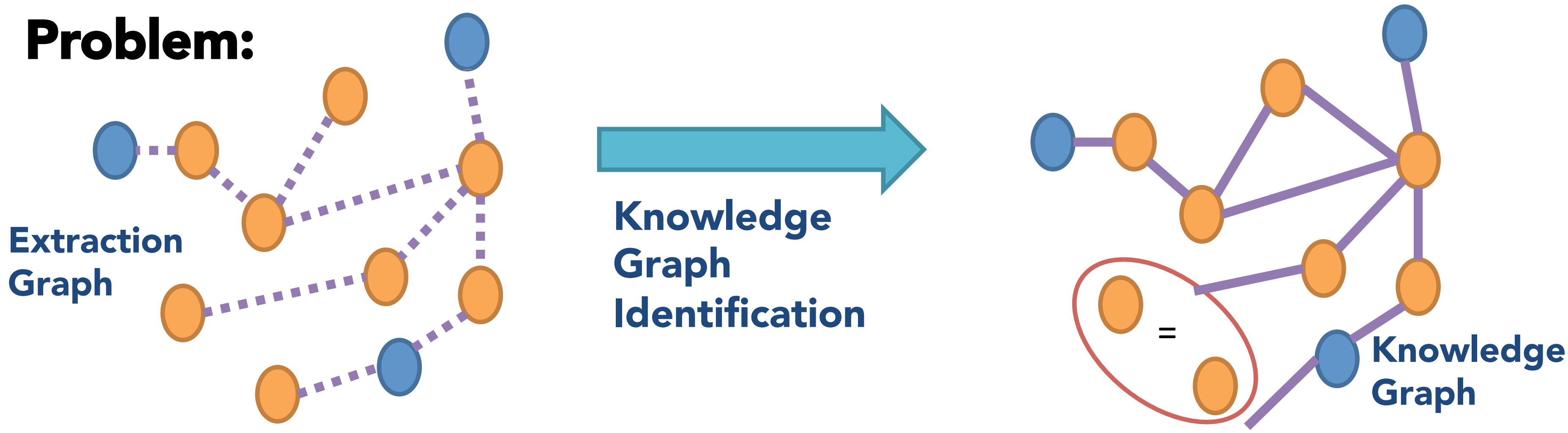
# of posts: 100,000

# of latent and target variables: 800,000

Ramesh et al., AAAI13; Ramesh et al., L@S14

Knowledge Graph Identification

Problem:



Solution: *Knowledge Graph Identification* (KGI)

- Performs *graph identification*:
  – entity resolution
  – node labeling
  – link prediction
- Enforces *ontological constraints*
- Incorporates *multiple uncertain sources*

"Knowledge Graph Identification"
Pujara et al., *ISWC*, 2013

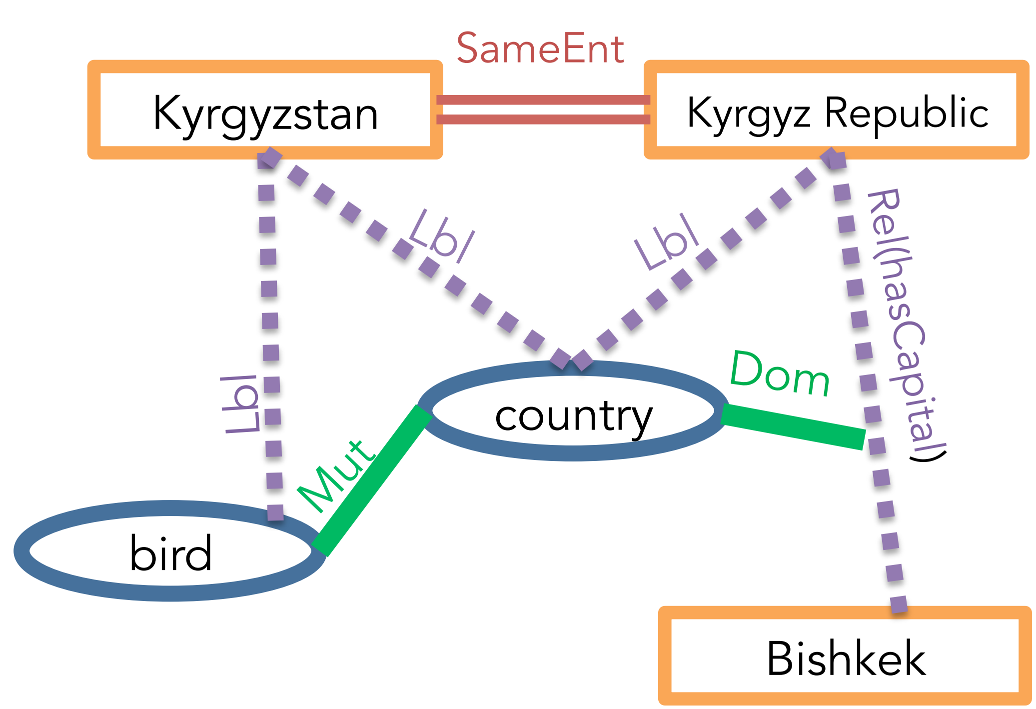
Illustration of KGI

Uncertain Extractions:
.5: Lbl(Kyrgyzstan, bird)
.7: Lbl(Kyrgyzstan, country)
.9: Lbl(Kyrgyz Republic, country)
.8: Rel(Kyrgyz Republic, Bishkek, hasCapital)

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

Entity Resolution:
SameEnt(Kyrgyz Republic, Kyrgyzstan)

(Annotated) Extraction Graph



After Knowledge Graph Identification

PSL for 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/linqs/KnowledgeGraphIdentification/

Knowledge Graph Construction on NELL

Task: Construct 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**

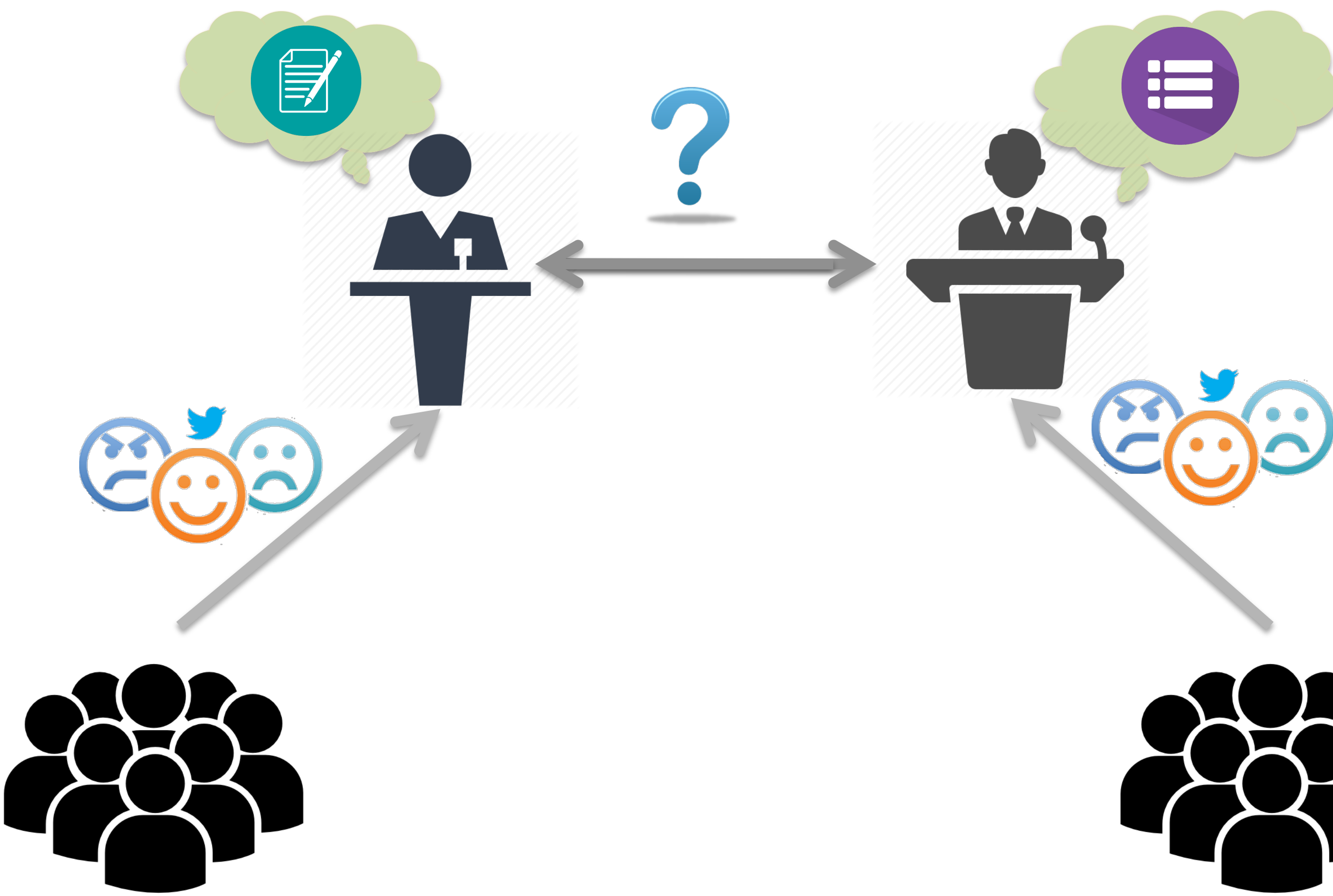
Complete knowledge graph including all NELL candidates

	AUC	F1
NELL	0.765	0.634
PSL-KGI	**0.892**	**0.848**

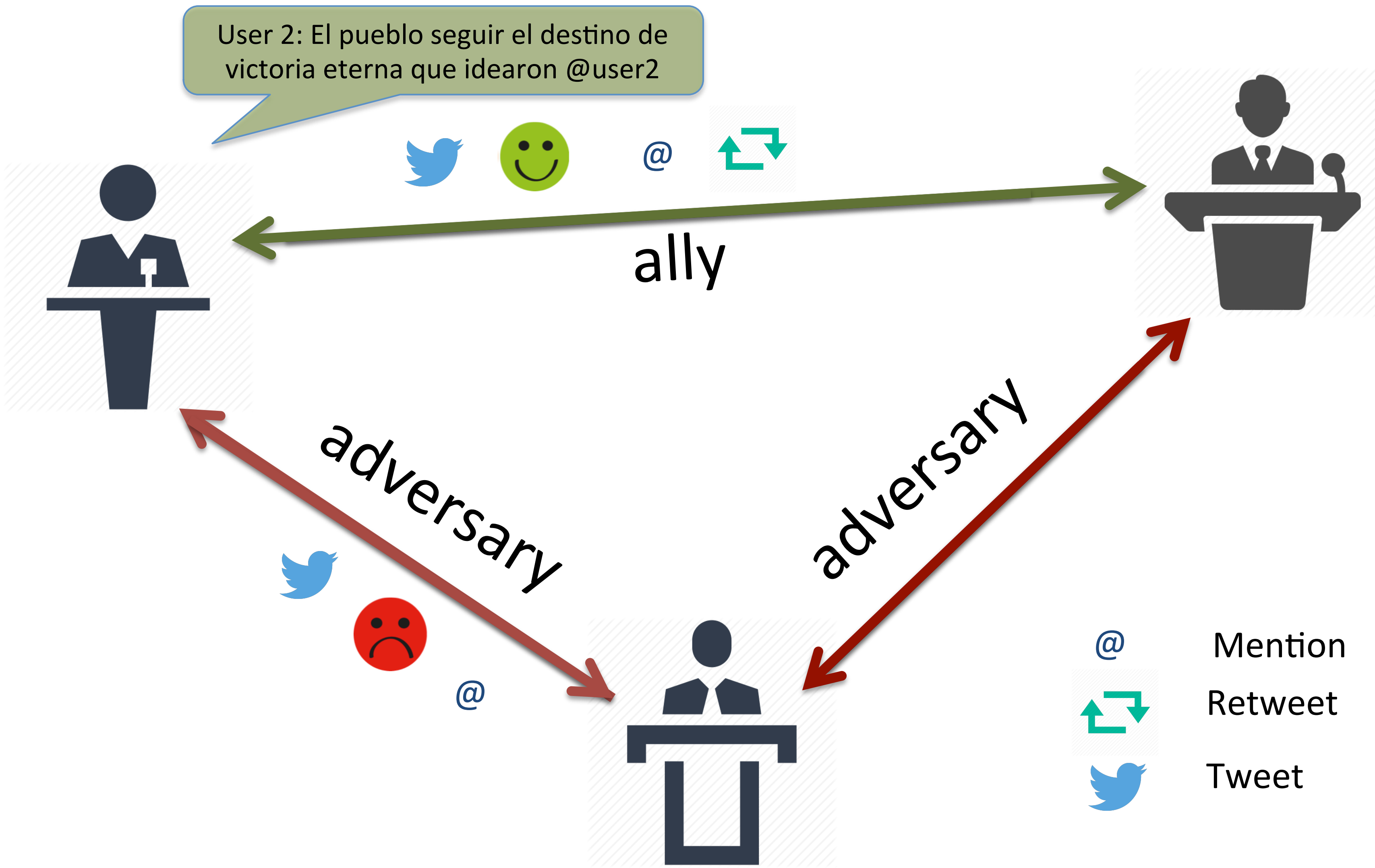
Running Time: Inference completes in 130 minutes, produces **4.3M facts**

"Using Statistics & Semantics to Turn Data Into Knowledge" Pujara et al., *AI Magazine*, 2015

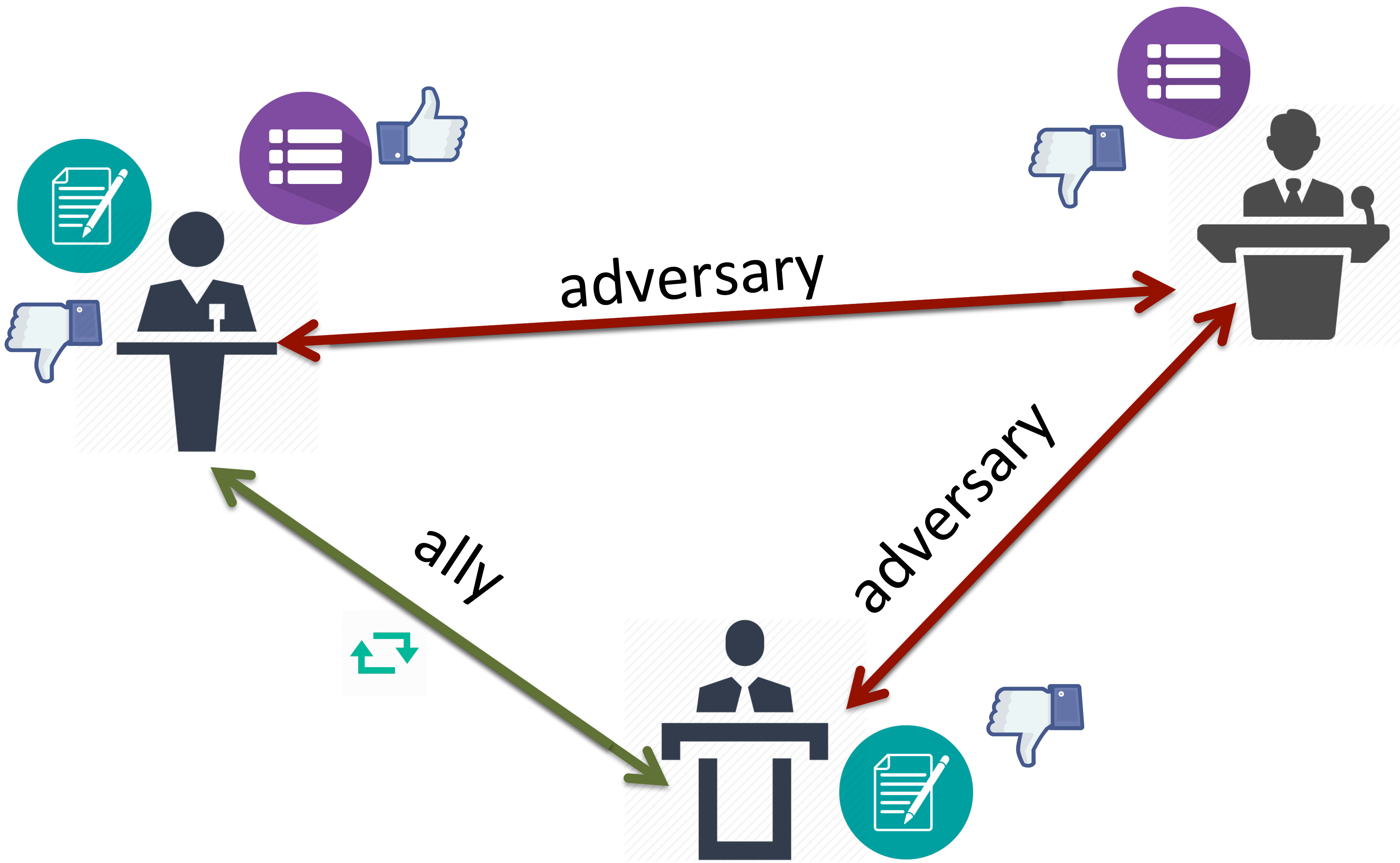
Inferring Strategic Relationships between Organizations



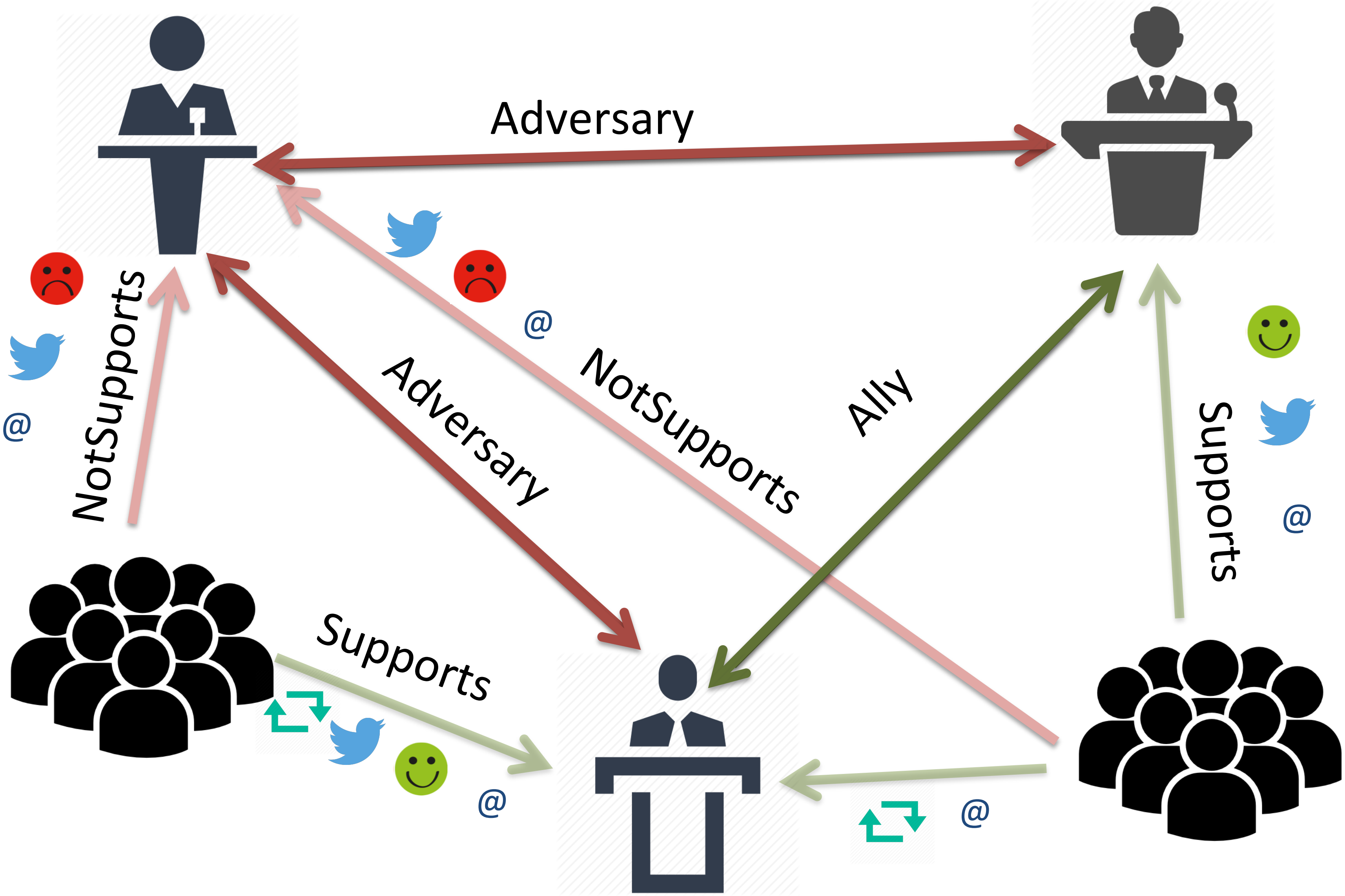
Inferring relationships using sentiment of mention tweets between organizations



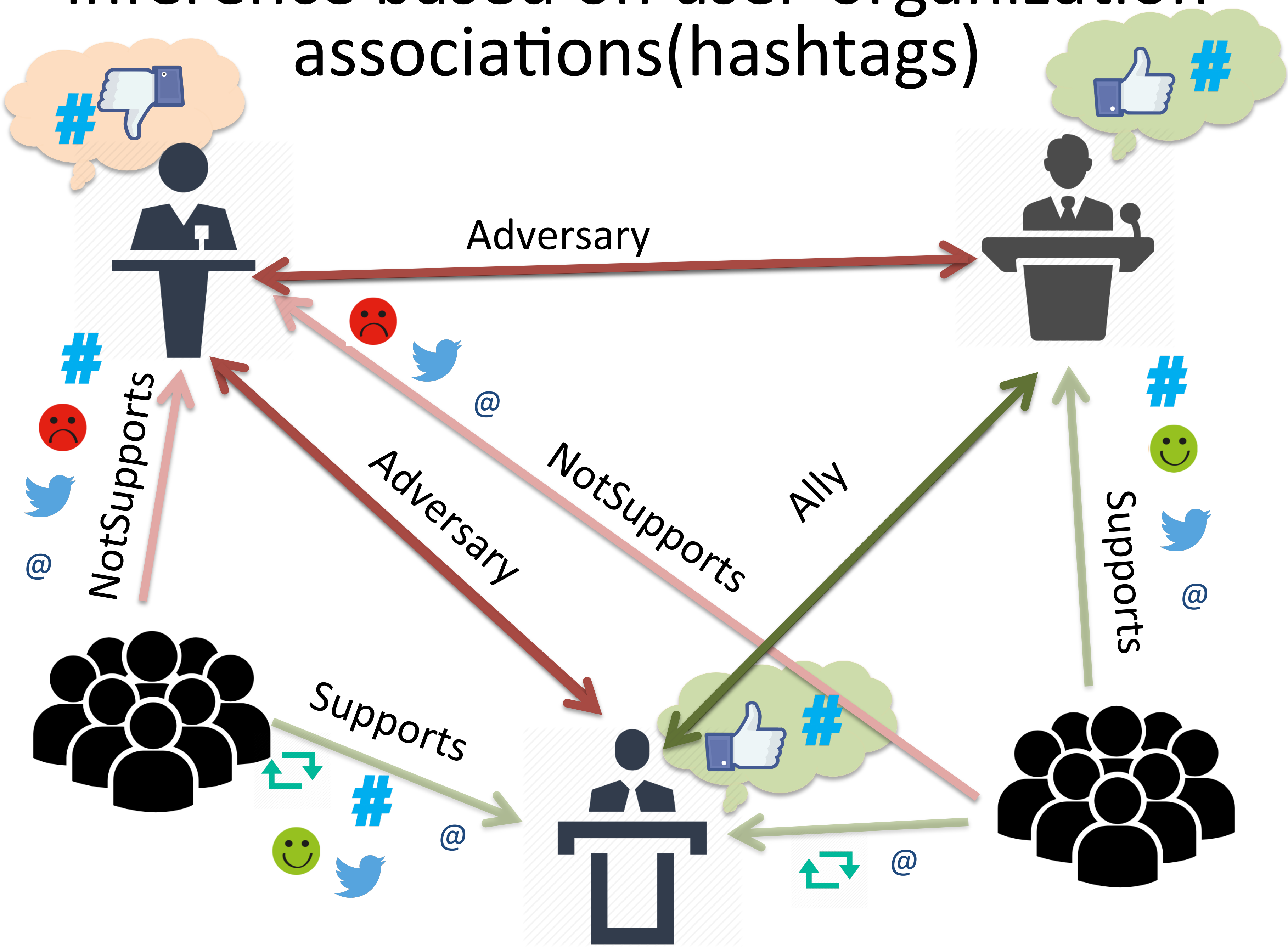
Inference based on sentiment towards topics



Inference based on user-organization associations(mentions)



Inference based on user-organization associations(hashtags)



PSL to infer Strategic Relationships

```
// Encoding interactions between organizations
  5: OrgRetweetsOrg(O1,O2)  ->  Ally(O1,O2)
  1: OrgPosts(T, O1) & Mentions(T,O2) & Positive(T) -> Ally(O1,O2)
  1: OrgPosts(T, O1) & Mentions(T,O2) & Negative(T) -> Adversary(O1,O2)

// Encoding organization sentiment towards topics
  1: OrgPosts(T, O1) & Topic(T,topic) & Positive(T) -> OrgLikesTopic(O,topic)
  1: OrgLikesTopic(O1,topic) & OrgLikesTopic(O2,topic) -> Adversary(O1,O2)

// Encoding user associations with organizations
  5: UserRetweetsOrg(U,O) -> Supports(U,O)
  1: Userposts (T,U) & Mentions(T,O) & Positive(T) -> Supports(U,O)
  1: Userposts (T,U) & Mentions(T,O) & Negative(T) -> NotSupports(U,O)
  1: Supports(U,O1) & Supports(U,O2) -> Ally(O1,O2)
  1: Supports(U,O1) & NotSupports(U,O2) -> Adversary(O1,O2)

// Encoding hashtags
  1: Userposts(T,U) & Hashtag(T,H) & Positive(T) & Supports(U,O)->
                                                     OrgLikesHashtag(O,H)
  1: Userposts(T,U) & Hashtag(T,H) & Negative(T) & Supports(U,O)->
                                                  OrgDisLikesHashtag(O,H)
  1: OrgLikesHashtag(O1,H) & OrgDislikesHashtag(O2,H) -> Adversary(O1,O2)
```

COMPARISON OF MODELS FOR INFERRING STRATEGIC RELATIONSHIPS

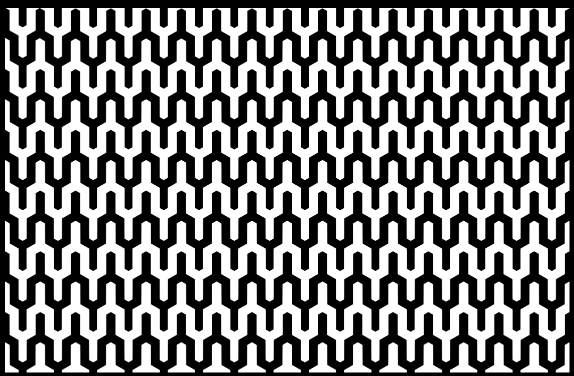
MODEL TYPE	MODEL	# PAIRS INFERRED	# CORRECTLY INFERRED
BASELINE MODEL	Aggregate	7	**7**
PSL BASE MODEL	PSL_OrgMention	7	**7**
	PSL_OrgRetweet	7	6
SOCIAL CONTEXT MODELS	PSL_UserMention	16	13
	PSL_UserHashtag	**22**	**18**
ISSUE BASED MODELS	PSL_TopicSentiment	**22**	17
	PSL_TopicSentimentAndMentions	**22**	**19**
COMBINED MODEL	PSL_Combined	**22**	**19**

COMPARISION OF MODELS

		Direct Interaction Models		Social Context Models		Issue Based Models			Combined
Org1, Org2	**Aggr. Model**	**PSL Org Mention**	**PSL Org Retweet**	**PSL User Mention**	**PSL User Hashtag**	**PSL Topic Sentiment**	**PSL Topic Sentiment Reciprocity**	**PSL Topic Sentiment Mention**	**PSL All Combined**
E. P. Nieto ,SNTE	-	-	-	-	0.96	0.92	0.80	0.89	0.93
D. Cabello, H. Capriles	-	-	-	0.51	0.98	0.92	0.81	0.94	0.93
D. Cabello, N. Maduro	-	0.54	0.55	0.53	0.97	0.86	0.86	0.55	0.56
H. Capriles, N. Maduro		1	1	0.53	0.99	0.92	0.81	1	1
C. Kirchner, M. Macri		-	-	0.54	0.54	0.95	0.92	1	1
H. Chavez, N. Maduro		-	-	0.53	0.98	0.81	0.78	0.57	0.57
S. Pinera, A. Chadwick	-	-	-	0.57	0.98	0.92	0.87	0.87	0.66
EPN, A. M. Lopez	-	-	-	0.51	0.76	0.99	0.96	0.97	0.93
C. Kirchner, N. Maduro		-	-	0.51	0.89	0.92	0.75	0.61	0.62
A. U. Velez, J. M. Santos		0.53	0.52	0.51	0.68	0.88	0.84	0.51	0.54
SNTE, E. Gordillo	-	-	-	-	0.97	0.92	0.80	0.89	0.93
J. M. Santos, N. Maduro	-	-	-	0.52	0.82	0.93	0.83	0.97	0.93
E. P. Nieto, sicilia oficial	-	-	-	-	0.96	0.92	0.80	0.89	0.93
J. A Cordova, E. Gordillo	-	-	-	-	0.97	0.92	0.80	0.89	0.75
C. Kirchner, J. M. Santos	-	-	-	0.51	0.64	0.94	0.85	0.95	0.93
P. Espinosa, J. A Cordova	-	-	-	-	0.98	0.92	0.80	0.89	0.50
J. Meade, E. P. Nieto		0.76	0.79	0.53	0.52	0.86	0.855	0.78	0.80
N. Maduro, A. U. Velez		-	-	0.54	0.83	0.91	0.84	0.94	0.93
D. Cabello, J. M. Santos		0.64	0.63	0.52	0.77	0.97	0.84	0.97	0.93
D. Cabello,H. Chavez		0.67	0.75	0.70	0.98	0.97	0.93	0.73	0.63
H Chavez, H. Capriles		-	-	-	0.99	0.96	0.86	0.89	0.93
CNC_CEN, E. P. Nieto		0.54	0.54	0.58	0.54	0.98	0.91	0.54	0.54

Discussion

Scalable ML for Graphs



Patterns

Key Ideas

Tools

Many exciting opportunities to develop
new theory, scalable algorithms & apply them
to compelling business, scientific and social problems!

Thank You!

Probabilistic
Soft
Logic

psl.umiacs.umd.edu

DSSC
DATA SCIENCE
SANTA CRUZ

Contact information:
getoor@ucsc.edu

@lgetoor