Several Recent Work on Social Mining: A Brief Review

Detecting Community Kernels in Large Social Networks	ICDM'11	[WLTH]
Who Will Follow You Back? Reciprocal Relationship Prediction	CIKM'11	[HLT]*
Inferring Social Ties Across Heterogeneous Networks	WSDM'12	[TLK]

*: alphabetical order

Liaoruo Wang	Tiancheng Lou	Jie Tang	John E. Hopcroft	Jon Kleinberg
Cornell	IIIS, Tsinghua	Tsinghua	Cornell	Cornell

Outline

- Detecting Community Kernels in Large Social Networks
- Who Will Follow You Back? Reciprocal Relationship Prediction
- Inferring Social Ties Across Heterogeneous Networks (very briefly)
- Related topics
- Summary and conclusions

Outline

- **Detecting Community Kernels in Large Social Networks**
- Who Will Follow You Back? Reciprocal Relationship Prediction
- Inferring Social Ties Across Heterogeneous Networks (very briefly)
- Related topics
- Summary and conclusions

Detecting Community Kernels
Motivation

- **“Pareto Principle”**
    - Less than 1% of the Twitter users (e.g. Lady Gaga, Kaifu Lee) produce 50% of its content, while the others (e.g. fans, followers, readers) have much less influence and completely different social behavior.
- **2 types of users:** very different influence and behavior

Detecting Community Kernels
Motivation

- **Challenges**
  - Distinguish stars (“kernels”) from others (“auxiliary community”)
  - Distinguish among stars

Detecting Community Kernels
Problem "Definition"

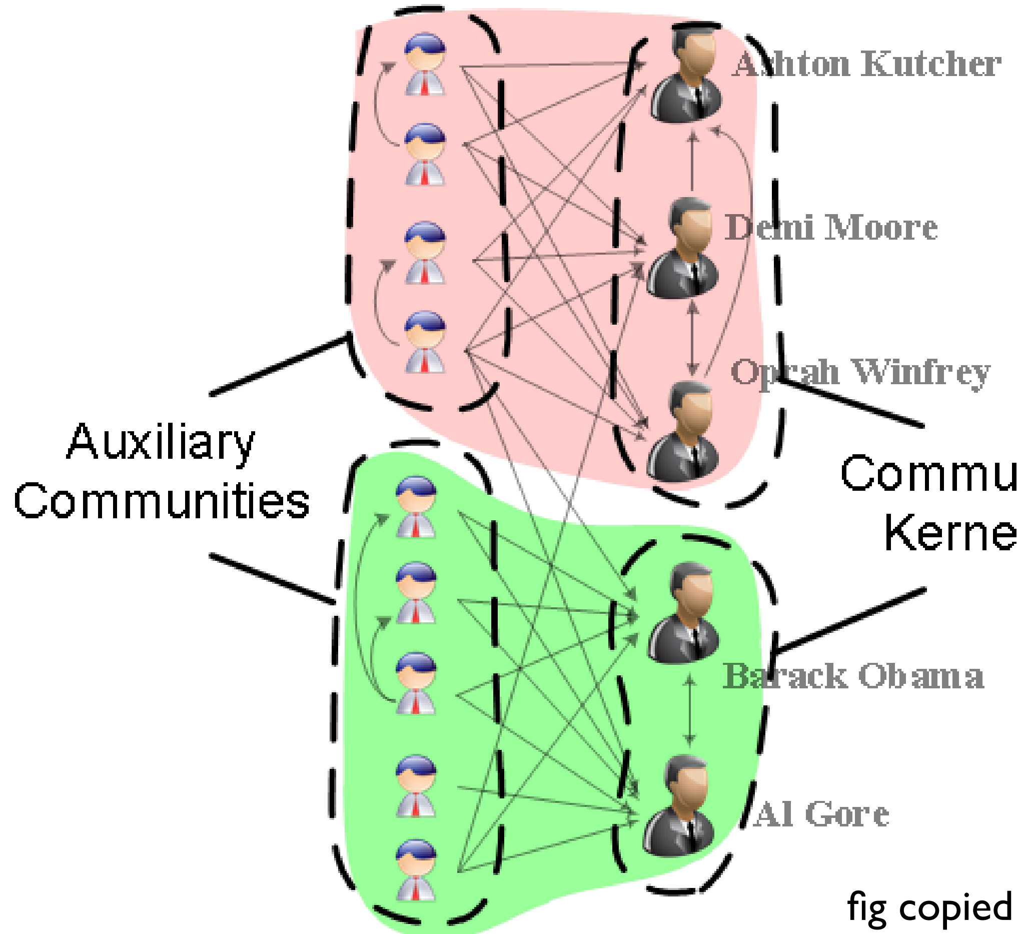


fig copied from this paper

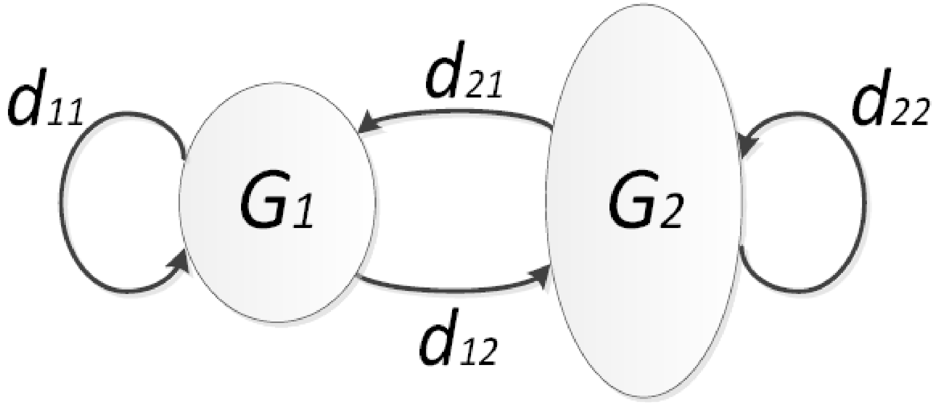
- Identify kernel members from auxiliary members
- Determine the "structure" of community kernels

Detecting Community Kernels
Unbalanced Weakly-Bipartite (UWB) Structure

- Empirical property of many real-world networks

$$d_{21} > d_{11} > d_{22} \gg d_{12}$$

$$d_{ij} = \frac{|E(V_i, V_j)|}{|V_j|}, \quad i, j \in \{1, 2\}$$



Network	d_{21}	d_{11}	d_{22}	d_{12}
Coauthor	14.19	5.34	4.42	0.37
Wikipedia	1689.31	104.22	4.69	0.60
Twitter	110.78	26.78	2.94	0.29
Slashdot	180.90	84.56	10.75	0.64
Citation	76.69	35.81	23.80	0.26

fig copied from this paper

Detecting Community Kernels
Proposed Algorithms

- Greedy
- Weight-Balanced Algorithm

Detecting Community Kernels
Greedy

- Input: graph G; kernel size (max # of vertices in a kernel): k.
- Output: community kernels $K = \{K_1, ..., K_L\}$
- Algorithm
  - init S to contain a random vertex
  - iteratively (k times) add to S
    - the vertex with most connections to S
  - add S to community kernels: K={K,S}
- Fast: O(V+E). But no approximation bound.
- Prone to initialization. Need multiple random initializations.

Detecting Community Kernels
WEBA

- Each vertex v has a weight vector: $\vec{w}(v) = \{w_1(v), \cdots, w_l(v)\}$ to represent its relative importance for each community kernels
- Optimization framework:

$$\max \quad \mathcal{L}(\vec{w}) = \sum_{(u,v)\in E} \vec{w}(u) \cdot \vec{w}(v)$$

subject to $\sum_{v\in V} w_i(v) = k,\ \forall i \in \{1, \cdots, l\}$

$$\sum_{1\le i\le l} w_i(v) \le 1,\ \forall v \in V$$

$$w_i(v) \ge 0, \forall v \in V,\ \forall i \in \{1, \cdots, l\}$$

- Intractable and thus need approximation
  - by solving its 1-dim version L(w)

Detecting Community Kernels
WEBA Properties

- Theorem 1. A global maximum of the objective function *L(w)* corresponds to a community kernel.
- However, maximizing *L(w)* is still NP-Hard (or is it?)
- Approximating *L(w)*:
    - init *S* using Greedy algorithm
    - using local heuristic to update S until convergence

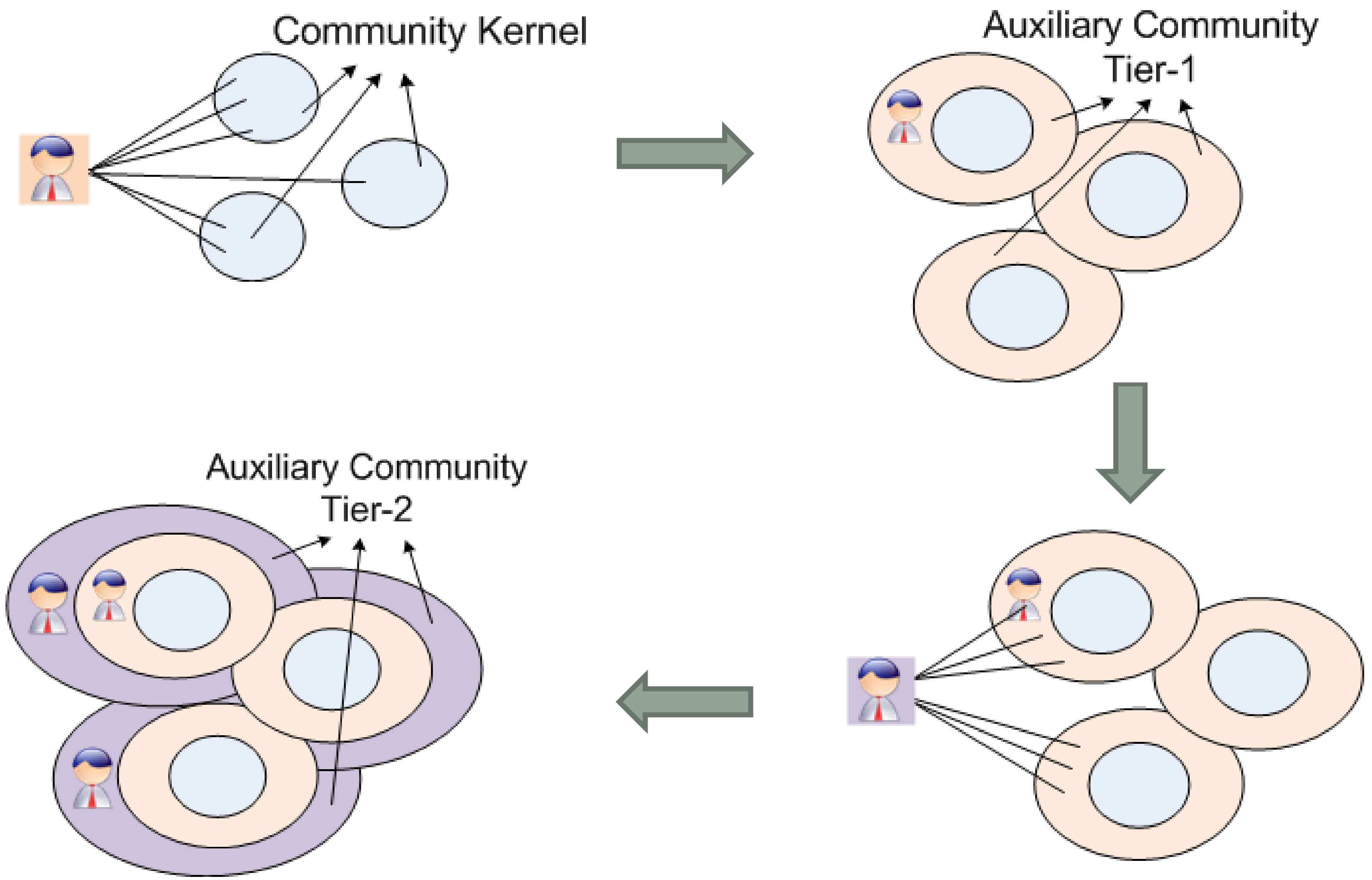
Detecting Community Kernels
WEBA Pseudocode

Input: $G = (V, E)$ and kernel size k
Output: community kernels $\mathbf{K} = \{\mathcal{K}_1, \mathcal{K}_2, \cdots, \mathcal{K}_\ell\}$
$\mathbf{K} \leftarrow \emptyset$
repeat
  $S \leftarrow$ a subset returned by $\textsc{Greedy}(G, k)$
  $\forall v \in S,\ w(v) \leftarrow 1;\ \forall v \notin S,\ w(v) \leftarrow 0$
  while $\exists\ u, v \in V$ *satisfying the relaxation conditions* **do**
    if $(u, v) \notin E$ **then** $\delta \leftarrow \min\{1 - w(u), w(v)\}$
    else $\delta \leftarrow \min\left\{1 - w(u), w(v), \frac{nw(u) - nw(v)}{2}\right\}$
    pick one pair $\{u, v\}$ with the maximum δ value
    $w(u) \leftarrow w(u) + \delta,\ \ w(v) \leftarrow w(v) - \delta$
  $C \leftarrow \{v \in V \mid w(v) = 1\}$
  if $C \notin \mathbf{K}$ **then** $\mathbf{K} \leftarrow \{\mathbf{K}, C\}$
until $O(|V|/k)$ *times*;
return $\mathbf{K}$

Detecting Community Kernels
WEBA Guarantees

- Theorem 2.
  - WEBA is guaranteed to converge.
- Theorem 3.
  - For any assigned weights $\{w(v), \forall v \in V\}$ and any $\varepsilon > 0$, after
$$\max\left\{\frac{4k^3D^5}{\varepsilon^2}, \frac{2mkD^3}{\varepsilon}\right\}$$
iterations, we have $\mathcal{L}(w^*(v)) - \mathcal{L}\big(w(v)\big) \leq \varepsilon$.

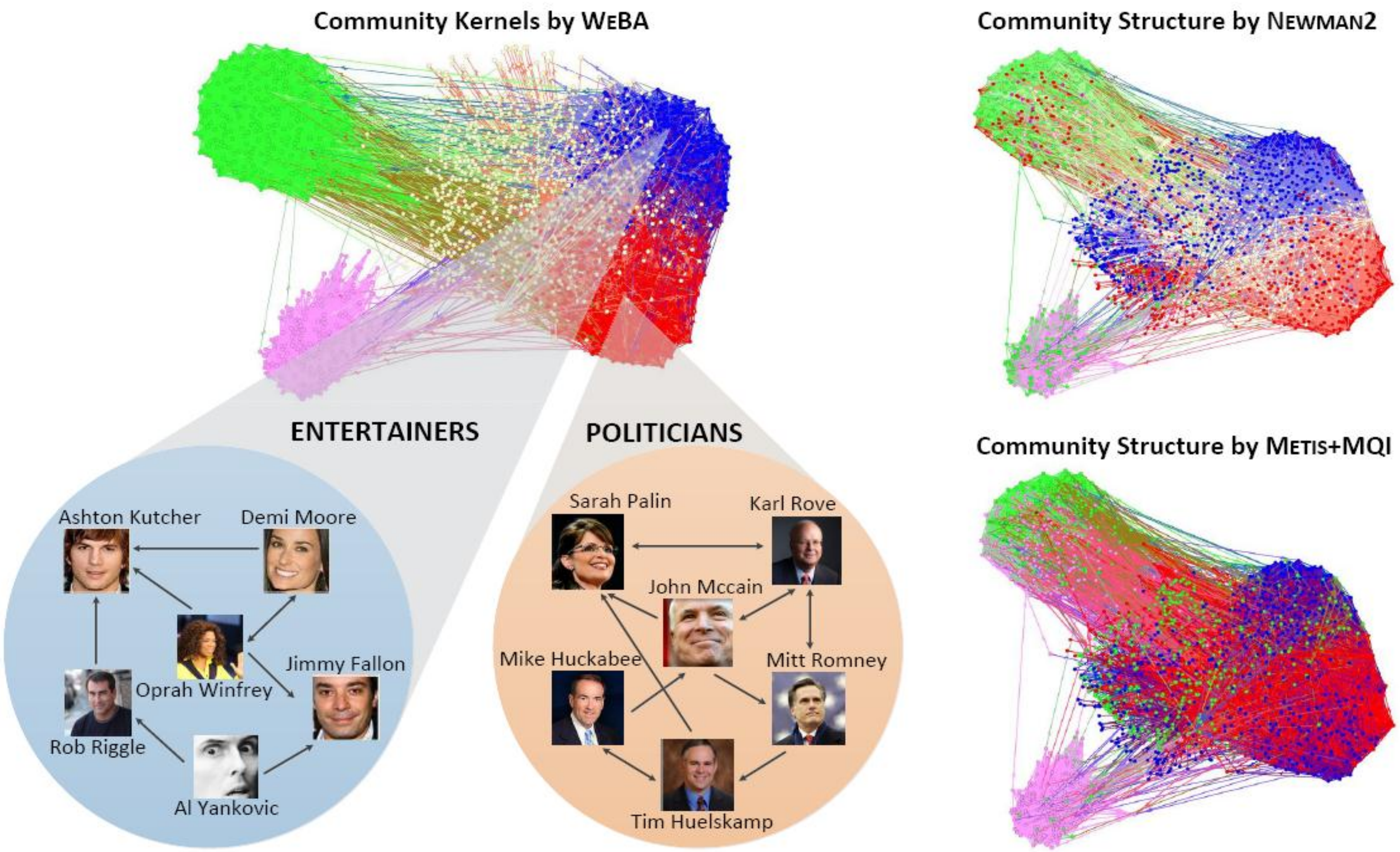
Detecting Community Kernels
Find auxiliary community



Detecting Community Kernels
Experiment: Setup

- Data sets
  - Coauthor (kernel = PC member)
  - Wikipedia (kernel = admins)
  - Twitter
- 8 different compared algorithm

Detecting Community Kernels
Experiment: Visualization



Detecting Community Kernels
Experiment: Results

- On average, WEBA improves Precision by 340% (wiki) and 70% (coauthor), and improves Recall by 130% (wiki) and 41% (coauthor).

	Precision						Recall					
	wiki		coauthor				wiki		coauthor			
	Talk	User	AI	…	NC	Average	Talk	User	AI	…	NC	Average
LSP	0.061	0.085	0.502	…	0.342	0.573	0.171	0.315	0.458	…	0.398	0.561
d-LSP	0.051	0.091	0.528	…	0.504	0.617	0.427	0.273	0.519	…	0.463	0.609
p-LSP	0.046	0.082	0.678	…	0.403	0.641	0.442	0.237	0.337	…	0.491	0.574
METIS+MQI	0.049	0.012	0.847	…	0.055	0.488	0.062	0.361	0.089	…	0.077	0.379
LOUVAIN	0.063	0.122	0.216	…	0.272	0.437	0.388	0.348	0.184	…	0.19	0.343
NEWMAN1	0.033	0.203	0.4	…	0.259	0.431	0.009	0.077	0.306	…	0.174	0.311
NEWMAN2	0.039	0.085	0.298	…	0.613	0.463	0.029	0.075	0.364	…	0.467	0.335
α-β	0.324	0.336	0.443	…	0.747	0.626	0.422	0.427	**0.602**	…	0.568	0.654
WEBA	**0.456**	**0.46**	**0.852**	…	**0.837**	**0.911**	**0.589**	**0.57**	0.577	…	**0.582**	**0.664**
GREEDY	0.334	0.403	0.83	…	0.746	0.752	0.432	0.499	0.545	…	0.56	0.659

87%

Detecting Community Kernels

Experiment: Other results

- F1-score and recall improved up to 300%
- not sensitive to parameters
- fast, parallelization etc.

Outline

- Detecting Community Kernels in Large Social Networks
- **Who Will Follow You Back? Reciprocal Relationship Prediction**
- Inferring Social Ties Across Heterogeneous Networks (very briefly)
- Related topics
- Summary and conclusions

Reciprocal Relationship Prediction
Motivation

- Background: 2 kinds of relationship
  - one-way (aka parasocial) relationship (Twitter)
  - two-way (aka reciprocal) relationship (Facebook)
    - usually developed from one-way relationships
- Problem: predict the formation of two-way relationships
  - micro-level dynamics
  - underlying community structure?
  - how users influence each other?

Reciprocal Relationship Prediction
Motivation

- **Challenges**
  - How to model the formation of two-way relationships?
    - Will Alice follow-back Bob?
  - How to combine many social theories into the prediction model?

Reciprocal Relationship Prediction
Problem Definition

- Given a network, $G=\{V, E, \boldsymbol{X}, Y\}$
  - $\boldsymbol{X}$: edge-specific features (fully observed)
  - Y: follow-back behavior
    - partially observed
- Goal: predict unknown Y.

Reciprocal Relationship Prediction
Proposed Model

- **Triad Factor Graph (TriFG) Model**
  - incorporate social theories over triads into factor graph model
- **Goal**: compute the posterior *P(Y|***X***,G)*. By Bayes theorem,

$$
\begin{aligned}
P(Y|\mathbf{X},G) &\propto P(\mathbf{X}|Y)P(Y|G) \\
&\propto P(Y|G)\prod_e P(\mathbf{x}_e|y_e)
\end{aligned}
$$

- Problem: model *P(Y|G)* and $P(\mathbf{x}_e|y_e)$
  - Using Markov Random Field (MRF).
  - Hammersly-Clifford theorem

$$
\begin{aligned}
P(\mathbf{x}_e|y_e) &= \frac{1}{Z_1}\exp\left\{\sum_d \alpha_d f_d(x_{ed}, y_e)\right\} \\
P(Y|G) &= \frac{1}{Z_2}\exp\left\{\sum_c \sum_k \mu_k h_k(Y_c)\right\}
\end{aligned}
$$

here combines social theories

Reciprocal Relationship Prediction
Proposed Model

- **Triad Factor Graph (TriFG) Model**
  - incorporate social theories over triads into factor graph model
- **Goal**: compute the posterior *P(Y|**X**,G)*. By Bayes theorem,

$$
\begin{aligned}
P(Y|\mathbf{X},G) &\propto P(\mathbf{X}|Y)P(Y|G) \\
&\propto P(Y|G)\prod_{e} P(\mathbf{x}_e|y_e)
\end{aligned}
$$

- Problem: model *P(Y|G)* and $P(\mathbf{x}_e|y_e)$
  - Using Markov Random Field (MRF).
  - Hammersly-Clifford theorem

$$
\begin{aligned}
P(\mathbf{x}_e|y_e) &= \frac{1}{Z_1}\exp\left\{\sum_{d}\alpha_d f_d(x_{ed}, y_e)\right\} \\
P(Y|G) &= \frac{1}{Z_2}\exp\left\{\sum_{c}\sum_{k}\mu_k h_k(Y_c)\right\}
\end{aligned}
$$

also know as
Conditional Random Field

here combines social theories

Reciprocal Relationship Prediction
Learning and Prediction

- Framework
  - maximize log-likelihood to find best parameters (using gradient descent)

$$O(\theta) = \sum_{e} \sum_{d} \alpha_d f_d(x_{ed}, y_e) + \sum_{c} \sum_{k} \mu_k h_k(Y_c) - \log Z$$

  - using estimated parameters to predict unknown variables
- Challenges
  - logZ is intractable: even compute the gradient is NP-hard
    - using Loopy Belief Propagation as an approximation

Reciprocal Relationship Prediction
Learning and Prediction

- Framework
  - maximize log-likelihood to find best parameters (using gradient descent)

standard MRF learning problem

sum over all triads!

$$O(\theta) = \sum_e \sum_d \alpha_d f_d(x_{ed}, y_e) + \sum_c \sum_k \mu_k h_k(Y_c) - \log Z$$

  - using estimated parameters to predict unknown variables

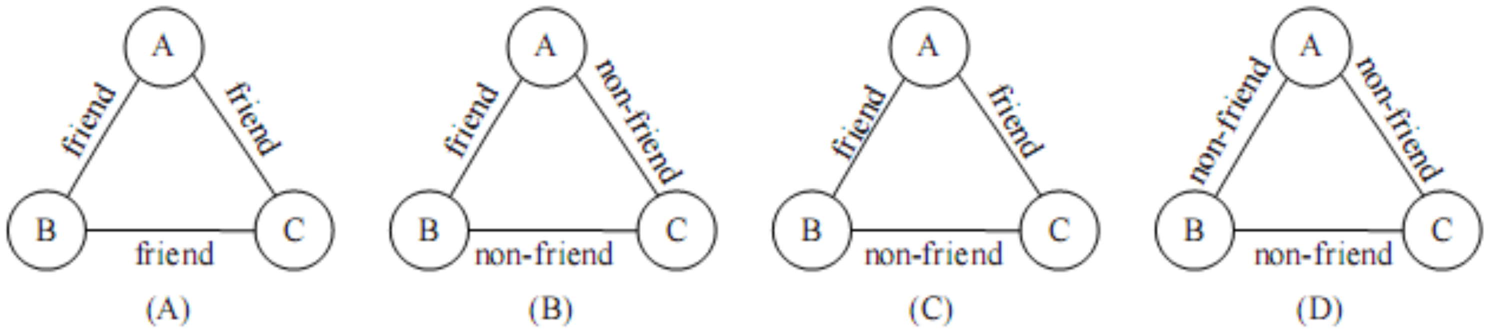
standard MRF MAP problem

- Challenges
  - logZ is intractable: even compute the gradient is NP-hard
    - using Loopy Belief Propagation as an approximation

standard MRF learning approach

Reciprocal Relationship Prediction
Features

- Edge-specific features
  - Geographic distance between users
  - Link homophily: users with common friends tend to follow each other
  - Status homophily: elite users tend to follow each other.
  - Retweet-reply-network is correlated with two-way relationships
- Triad features
  - structural balance social theory



balanced

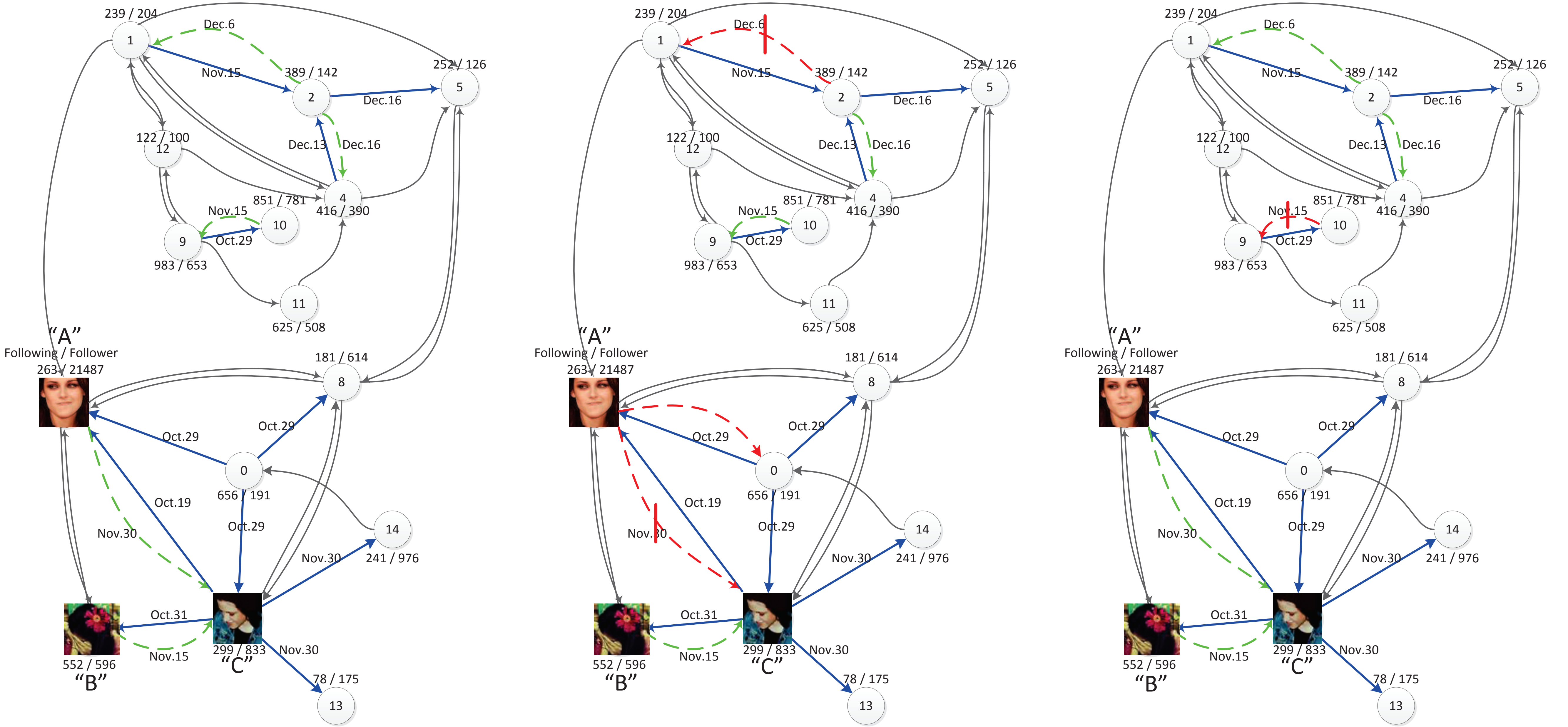
not-balanced

Reciprocal Relationship Prediction
Experiment: Setup

- Data sets
  - Twitter (with time-stamp)
- Baseline
  - SVM, Logistic regression, CRF (without unlabeled data)

Reciprocal Relationship Prediction

Experiment: Case Study



(a) Ground Truth

(b) SVM

(c) Our approach (TriFG)

Reciprocal Relationship Prediction
Experiment: Result

- Inferred 90% follow-back behavior

Data	Algorithm	Prec.	Rec.	F1	Accu.
Test Case 1	SVM	0.6908	0.6129	0.6495	0.9590
	LRC	0.6957	0.2581	0.3765	0.9510
	CRF-balance	0.9968	0.5161	0.6801	0.9670
	CRF	**1.0000**	0.6290	0.7723	0.9770
	wTriFG	0.9691	0.5483	0.7004	0.9430
	TriFG	**1.0000**	**0.8548**	**0.9217**	**0.9910**
Test Case 2	SVM	0.7323	0.6212	0.6722	0.9534
	LRC	0.8333	0.3030	0.4444	0.9417
	CRF-balance	0.9444	0.5151	0.6667	0.9114
	CRF	**1.0000**	0.6333	0.7755	0.9717
	wTriFG	0.9697	0.5697	0.7177	0.9389
	TriFG	**1.0000**	**0.8788**	**0.9355**	**0.9907**

27%

Reciprocal Relationship Prediction
Experiment: Other Result

- Better than other graph-based algorithm
- Fast, convergence, etc.

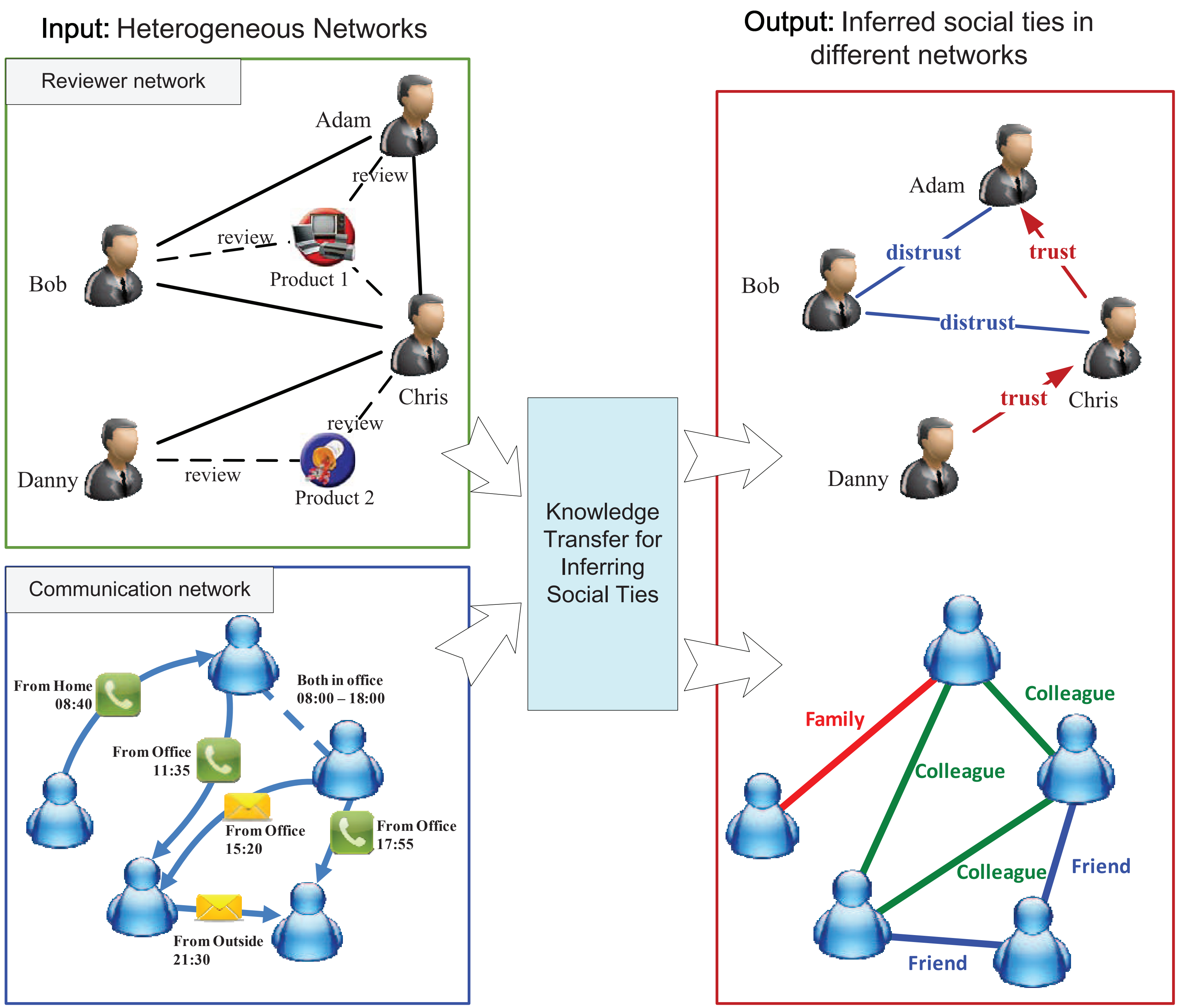
Outline

- Detecting Community Kernels in Large Social Networks
- Who Will Follow You Back? Reciprocal Relationship Prediction
- **Inferring Social Ties Across Heterogeneous Networks**
  (very briefly)
- Related topics
- Summary and conclusions

Inferring Social Ties (in 5 slides)
Motivation

- Background
  - Many different types of social "ties" (aka. relationship).
  - Many different types of online social networks.
  - Labeled relationships are scare.
- Problem
  - Leverage labeled relationships from one network to infer type of relationships in another different network

Inferring Social Ties
Motivating Example



Reciprocal Relationship Prediction
Proposed Model

- **Triad Factor Graph (TriFG) Model**
  - incorporate social theories over triads into factor graph model
- **Goal**: compute the posterior *P(Y|**X**,G)*. By Bayes theorem,

$$
\begin{aligned}
P(Y|\mathbf{X},G) &\propto P(\mathbf{X}|Y)P(Y|G) \\
&\propto P(Y|G)\prod_e P(\mathbf{x}_e|y_e)
\end{aligned}
$$

Y: follow-back?

- Problem: model $P(Y|G)$ and $P(\mathbf{x}_e|y_e)$
  - Using Markov Random Field (MRF).
  - Hammersly-Clifford theorem

$$
\begin{aligned}
P(\mathbf{x}_e|y_e) &= \frac{1}{Z_1}\exp\left\{\sum_d \alpha_d f_d(x_{ed}, y_e)\right\} \\
P(Y|G) &= \frac{1}{Z_2}\exp\left\{\sum_c \sum_k \mu_k h_k(Y_c)\right\}
\end{aligned}
$$

also know as Conditional Random Field

here combines social theories

~~Reciprocal Relationship Prediction~~
Proposed Model

- **Triad Factor Graph (TriFG) Model**
  - incorporate social theories over triads into factor graph model
- **Goal**: compute the posterior $P(Y|\mathbf{X},G)$. By Bayes theorem,

$$
\begin{aligned}
P(Y|\mathbf{X},G) &\propto P(\mathbf{X}|Y)P(Y|G) \\
&\propto P(Y|G)\prod_e P(\mathbf{x}_e|y_e)
\end{aligned}
$$

Y: follow-back?

- Problem: model $P(Y|G)$ and $P(\mathbf{x}_e|y_e)$
  - Using Markov Random Field (MRF).
  - Hammersly-Clifford theorem

$$
\begin{aligned}
P(\mathbf{x}_e|y_e) &= \frac{1}{Z_1}\exp\left\{\sum_d \alpha_d f_d(x_{ed}, y_e)\right\} \\
P(Y|G) &= \frac{1}{Z_2}\exp\left\{\sum_c \sum_k \mu_k h_k(Y_c)\right\}
\end{aligned}
$$

also know as Conditional Random Field

here combines social theories

Inferring Social Ties
Proposed Model

- ~~**Triad Factor Graph (TriFG) Model**~~
- **Transfer-based Factor Graph (TranFG) Model**
- **Goal**: compute the posterior *P(Y|**X**,G)*. By Bayes theorem,

$$
\begin{aligned}
P(Y|\mathbf{X},G) &\propto P(\mathbf{X}|Y)P(Y|G) \\
&\propto P(Y|G)\prod_e P(\mathbf{x}_e|y_e)
\end{aligned}
$$

Y: type of social tie

- Problem: model $P(Y|G)$ and $P(\mathbf{x}_e|y_e)$
  - Using Markov Random Field (MRF).
  - Hammersly-Clifford theorem

$$
\begin{aligned}
P(\mathbf{x}_e|y_e) &= \frac{1}{Z_1}\exp\left\{\sum_d \alpha_d f_d(x_{ed}, y_e)\right\} \\
P(Y|G) &= \frac{1}{Z_2}\exp\left\{\sum_c \sum_k \mu_k h_k(Y_c)\right\}
\end{aligned}
$$

also know as
Conditional Random Field

here combines social theories

Inferring Social Ties
Learning and Prediction

- Framework
  - maximize log-likelihood (using Loopy Belief Propagation)
    - "learn across heterogeneous networks"

source network target network

$$\begin{aligned} O(\alpha,\beta,\mu) &= O_S(\alpha,\mu) + O_T(\beta,\mu) \\ &= \sum_{e\in E^S}\sum_d \alpha_d g_d(x_{ed}^S, y_e^S) + \sum_{e\in E^T}\sum_{d'} \beta_{d'} g'_{d'}(x_{ed'}^T, y_e^T) \\ &\quad + \sum_k \mu_k \left(\sum_c h_k(Y_c^S) + \sum_{c'} h_k(Y_{c'}^T) \right) \\ &\quad - \log Z \end{aligned}$$

Objective in "Reciprocal Relationship Prediction"

$$O(\theta) = \sum_e \sum_d \alpha_d f_d(x_{ed}, y_e) + \sum_c \sum_k \mu_k h_k(Y_c) - \log Z$$

Inferring Social Ties
Experiment

- Data sets
  - Epinions, Slashdot, Mobile, Coauthor, Enron
- Baseline methods
  - SVM, CRF, PFG (CRF which uses unlabeled data proposed by Jie Tang)
- Results
  - 8-28% improvements over alternative method on F1-score
  - fast, convergence, etc.

Outline

- Detecting Community Kernels in Large Social Networks
- Who Will Follow You Back? Reciprocal Relationship Prediction
- Inferring Social Ties Across Heterogeneous Networks (very briefly)
- **Related topics**
- **Summary and conclusions**

Related topics

- Community detection (DCK)
- Leader detection (DCK)
- Link prediction (RRP) (IST)
- Link classification (RRP) (IST)
- Each of these topic is very popular in recent years and have hundreds of related papers.

DCK: Detecting Community Kernels
RRP: Reciprocal Relationship Prediction
IST: Inferring Social Ties

Summary and conclusion

- Three papers in decent conferences produced in 6~8 months
- Common feature
    - Very consistent, careful and professional writing style
    - Almost same section titles:

Introduction	Problem Definition	Data and Observation	Model Framework	Experiments + Result and Analysis	Related Work	Conclusions

-
    - Carefully distinguish with existing problems and solutions
        - A good name to the problem and solution.
    - Extensive experiments and in depth data analysis

Thanks!

Question?