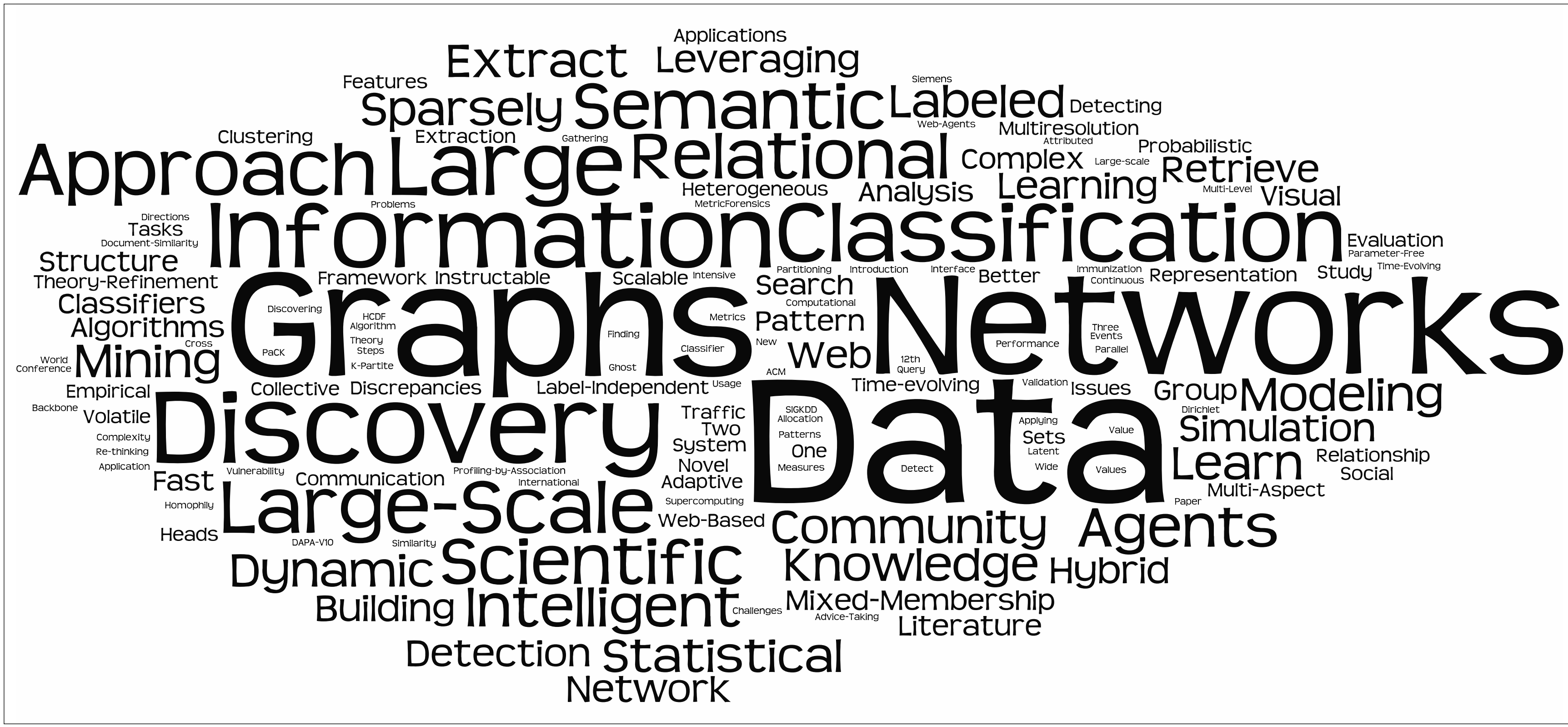
RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

Classification and Clustering in Large Complex Networks

Tina Eliassi-Rad
tina@eliassi.org

Spring 2011

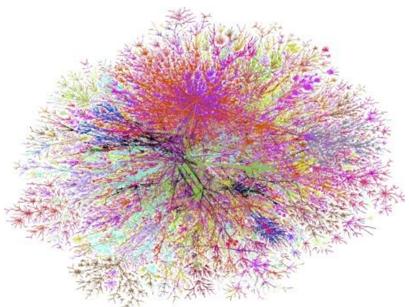
Wordle™ says…


Applications
Extract
Leveraging
Features
Siemens
Sparsely
Semantic
Labeled
Detecting
Web-Agents
Multiresolution
Clustering
Extraction
Gathering
Attributed
Probabilistic
Approach
Large
Relational
Complex
Large-scale
Retrieve
Heterogeneous
Analysis
Learning
Multi-Level
Visual
Directions
Problems
MetricForensics
Tasks
Document-Similarity
Information
Classification
Evaluation
Parameter-Free
Structure
Partitioning
Introduction
Interface
Immunization
Time-Evolving
Theory-Refinement
Framework
Instructable
Scalable
Intensive
Better
Continuous
Representation
Study
Search
Computational
Classifiers
Discovering
Graphs
Networks
Algorithms
HCDF
Algorithm
Metrics
Pattern
Theory
Steps
Finding
Three
Events
Cross
New
Performance
World
Conference
Mining
PaCK
K-Partite
Classifier
Web
12th
Query
Parallel
Ghost
ACM
Empirical
Collective
Discrepancies
Label-Independent
Usage
Time-evolving
Validation
Issues
Group
Modeling
Backbone
Traffic
SIGKDD
Allocation
Dirichlet
Volatile
Discovery
Two
Applying
Value
Simulation
Complexity
Patterns
Sets
System
Data
Re-thinking
One
Latent
Relationship
Application
Novel
Measures
Detect
Wide
Values
Learn
Social
Vulnerability
Profiling-by-Association
Fast
Communication
Adaptive
International
Multi-Aspect
Supercomputing
Paper
Homophily
Large-Scale
Web-Based
Community
Agents
Heads
DAPA-V10
Similarity
Scientific
Knowledge
Dynamic
Hybrid
Building
Intelligent
Challenges
Mixed-Membership
Advice-Taking
Literature
Detection
Statistical
Network

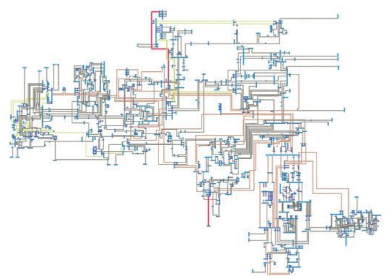
Complex Networks are Ubiquitous

Technological Networks

Internet

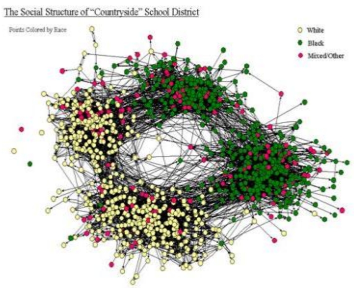


NY State Power Grid

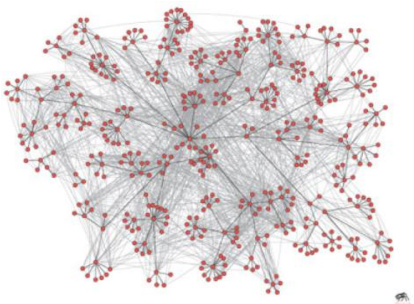


Social Networks

Friendship

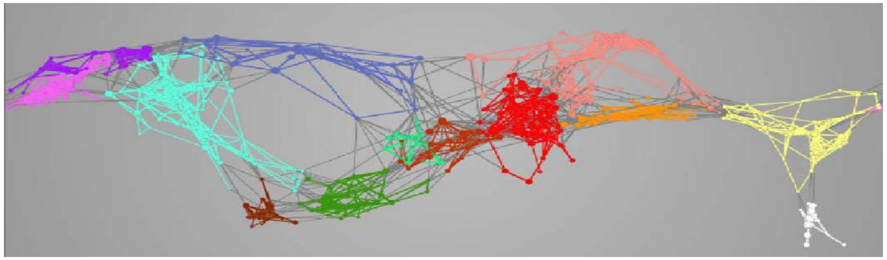


HP Emails



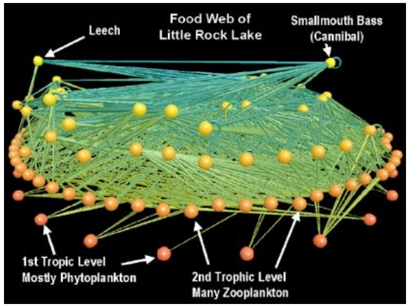
Information Networks

Map of Science

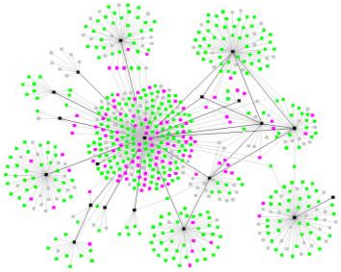


Biological networks

Food Web



Contagion of TB



Problems

Network Classifiers

Transfer Learning

Statistical Tests for Relational Classifiers

Community Discovery

Anomaly Detection

Re-identification

Pattern Matching

Link Analysis

Knowledge Representation

Applications

Humanities

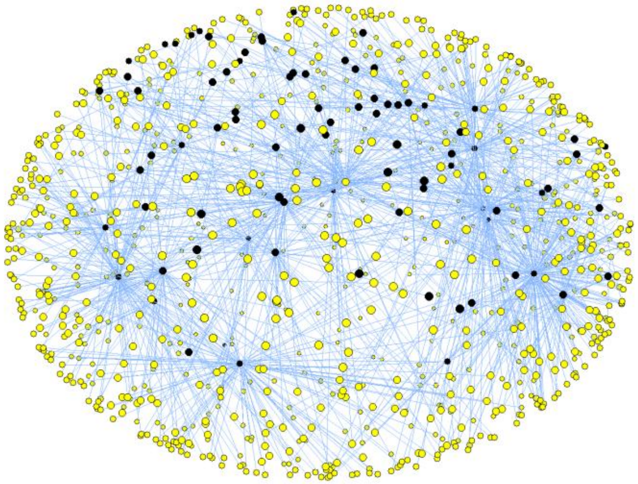
Cyber Situational Awareness

Social Science

Marketing

Search

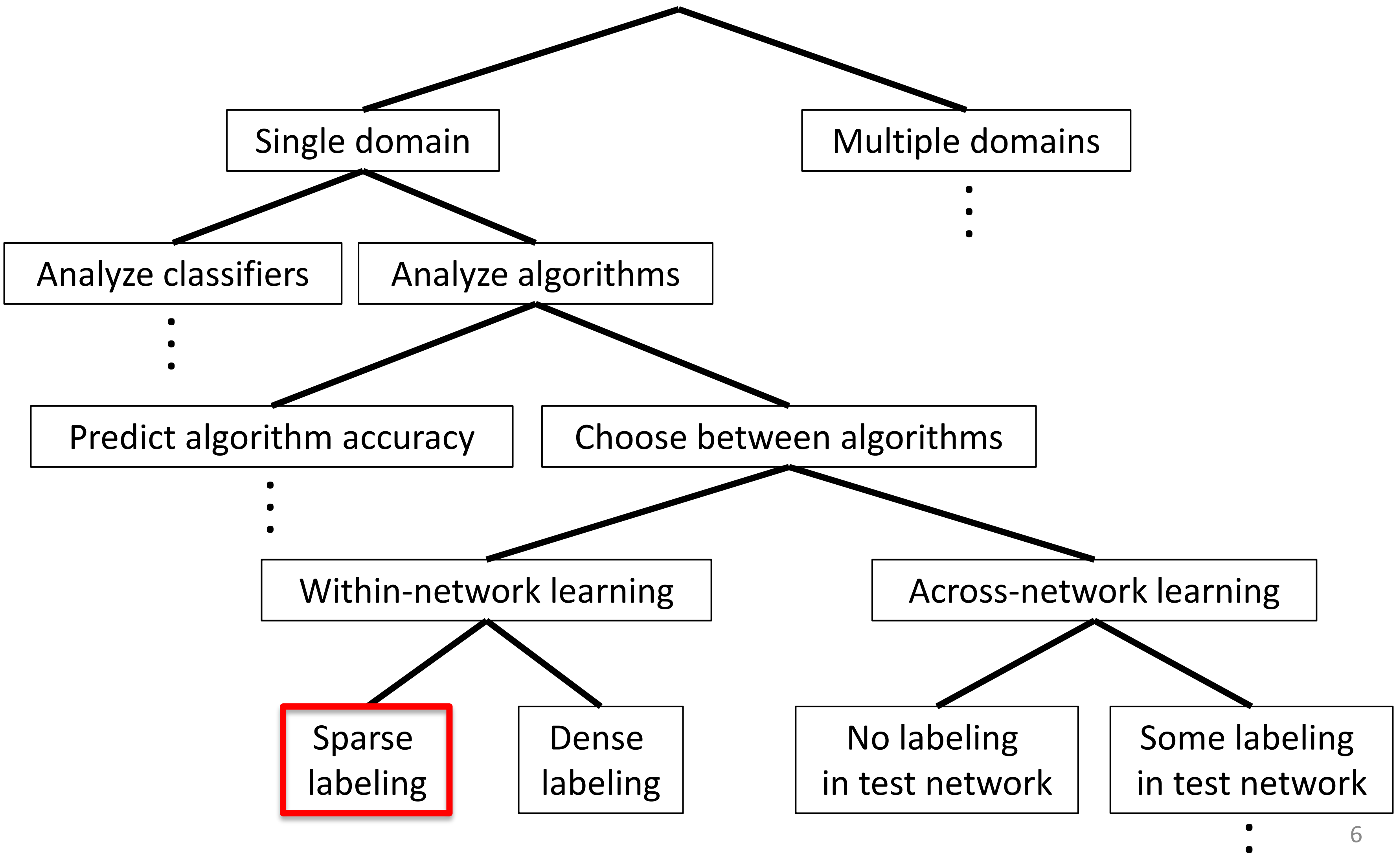
Smart Meters



Outline

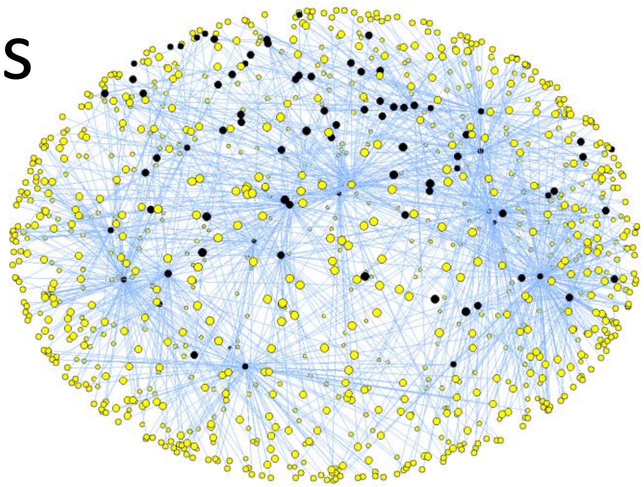
- Problem #1: Network (a.k.a. relational) classifiers
- Problem #2: Clustering on networks (a.k.a. community discovery)
- Conclusions

Relational Classifiers



Within-Network Classification

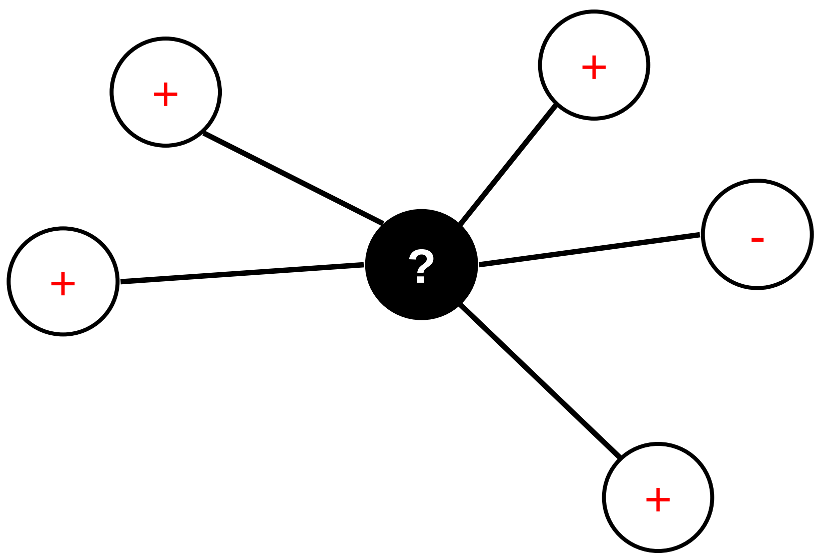
- **Given:** single network with labeled and unlabeled nodes



- **Goal:** assign labels to unlabeled nodes
- **Applications:** identify suspicious blog postings, fraudulent web pages, roles within an organization, research paper topics, etc.

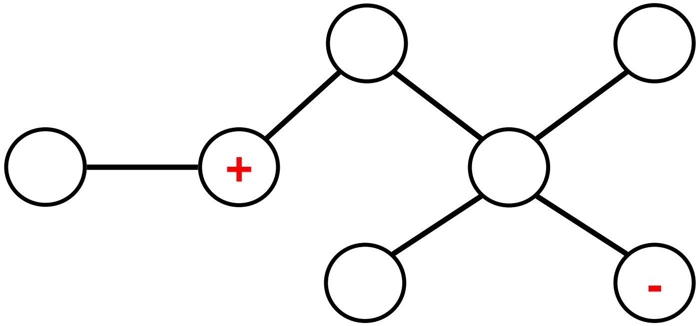
Background: Previous Work on Within-Network Classification

- Use dependencies between neighbors
  - Assume homophily
  - Learn dependencies

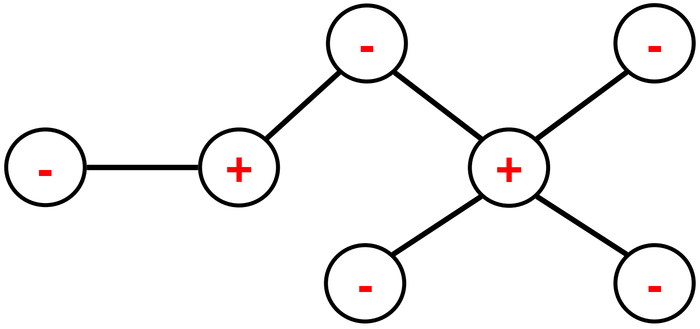


Our Work Addresses Two Challenges for Within-Network Classification

- C1: Sparse labeling



- C2: Non-homophily



Limitations of Existing Approaches for Within-Network Classification

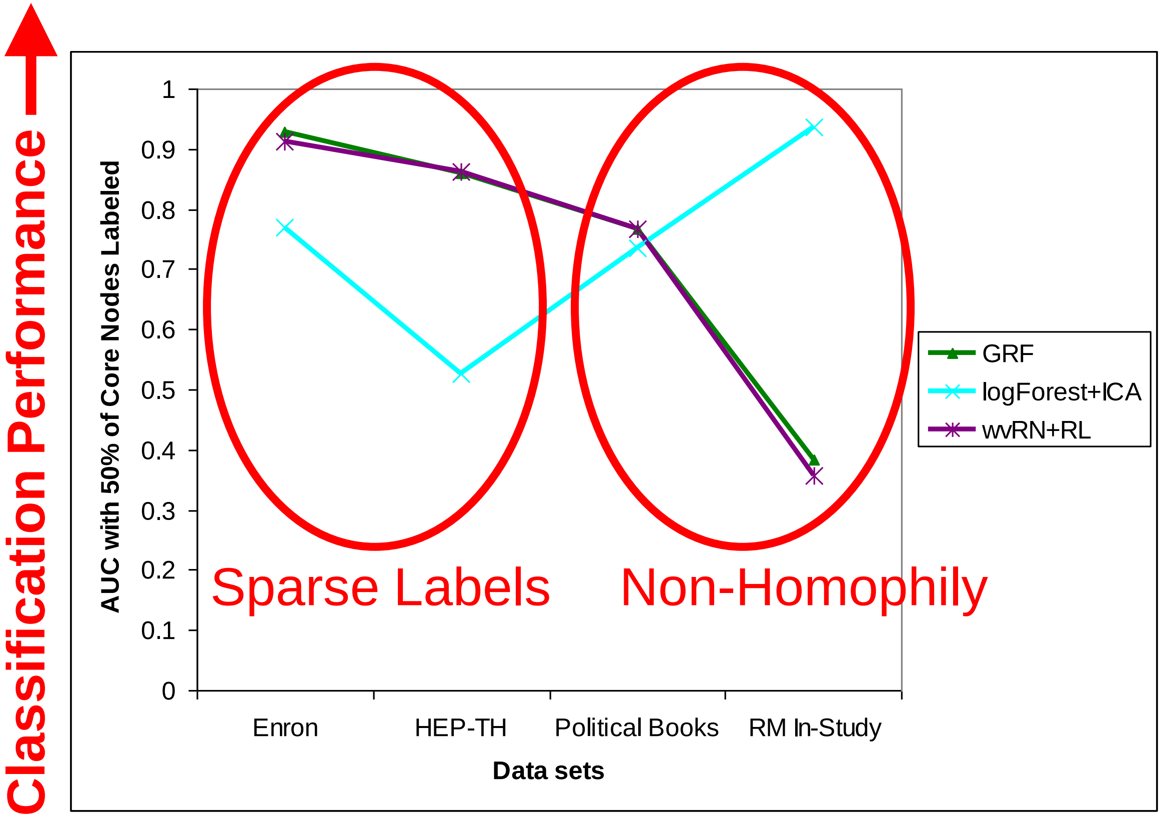
<u>Semi-supervised Learning</u>

Gaussian Random Fields
(Zhu et al., ICML 2003)

<u>Collective Classification</u>

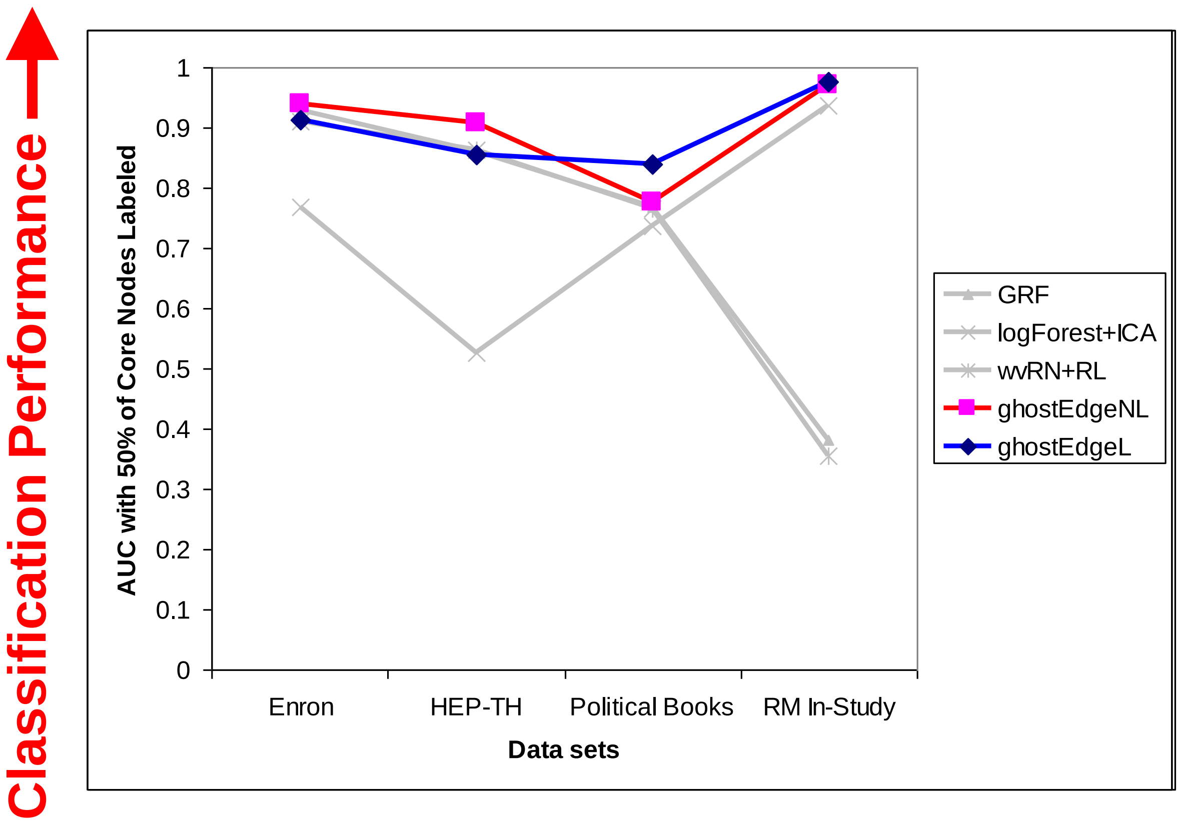
Link-based Classifier
(Lu & Getoor, ICML 2003)

Relational Neighbor Classifier
(Macskassy & Provost, KDD-MRDM 2003)



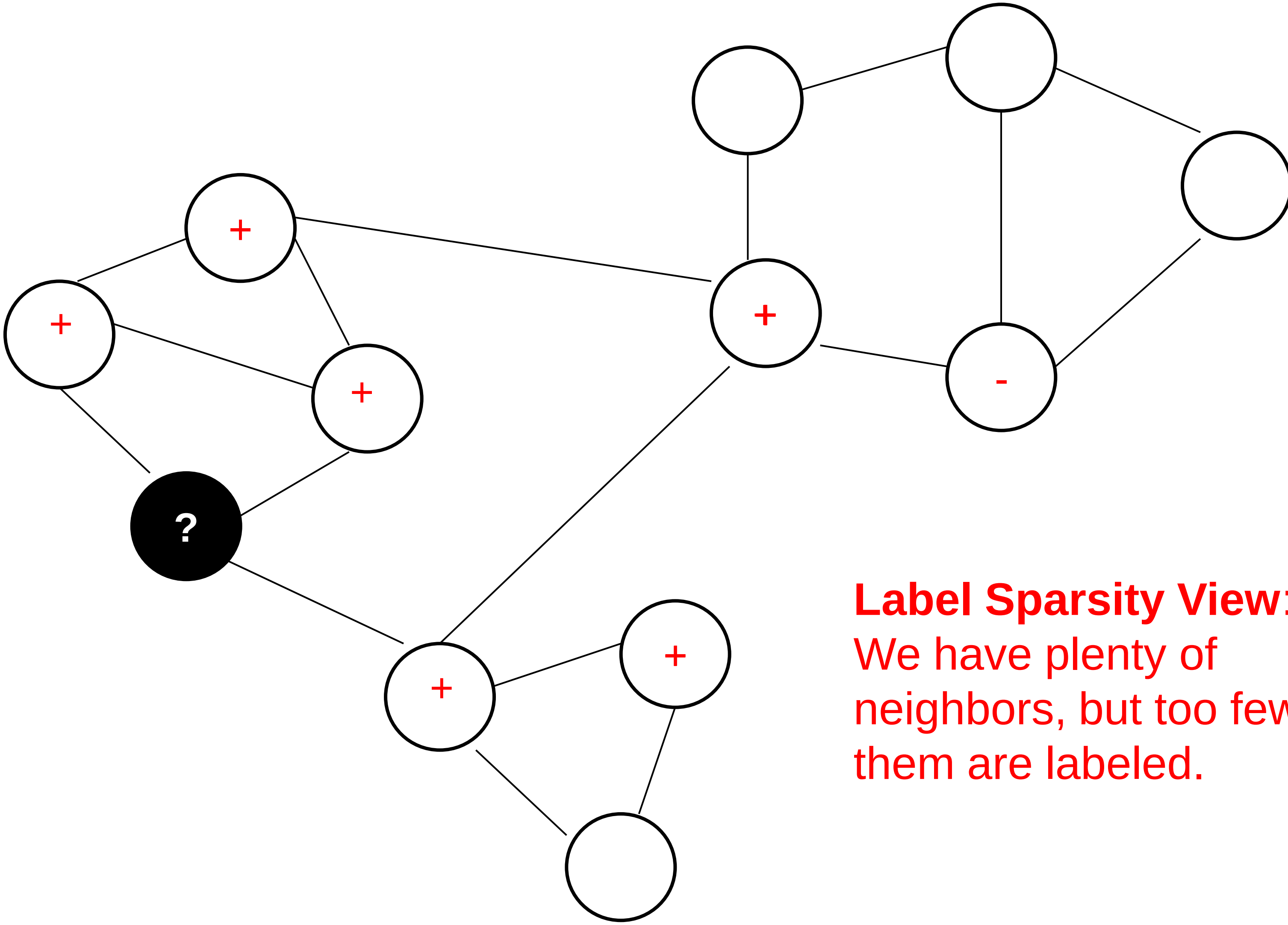
Our Solution: Ghost Edge Classifiers Have Consistently High Performance

- **Handle C1: Sparse labeling**
- **Handle C2: Non-homophily**



Challenge 1: Label Sparsity

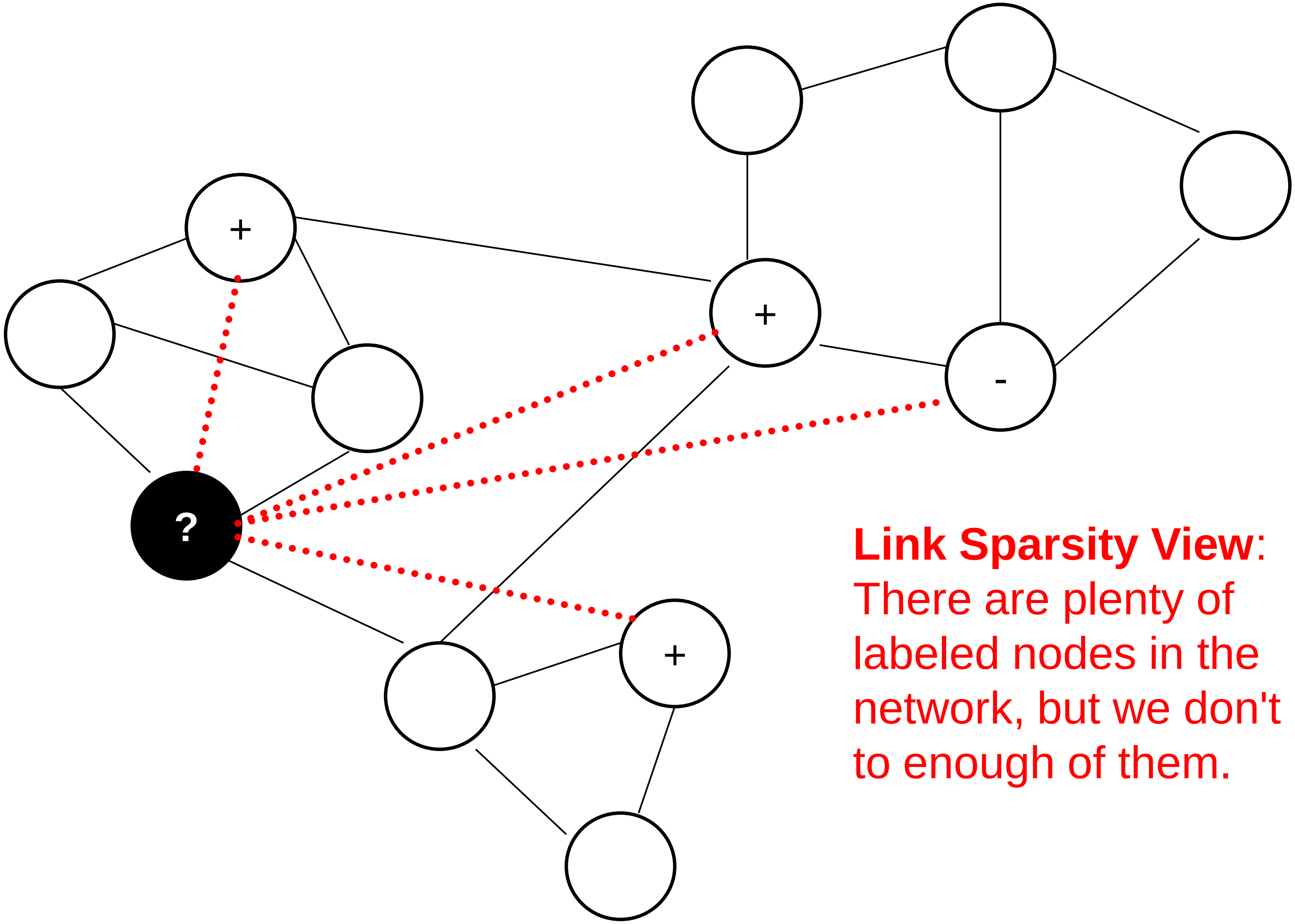
The Standard Approach



Label Sparsity View:
We have plenty of neighbors, but too few of them are labeled.

Challenge 1: Label Sparsity

The Ghost Edge Approach



Link Sparsity View: There are plenty of labeled nodes in the network, but we don't link to enough of them.

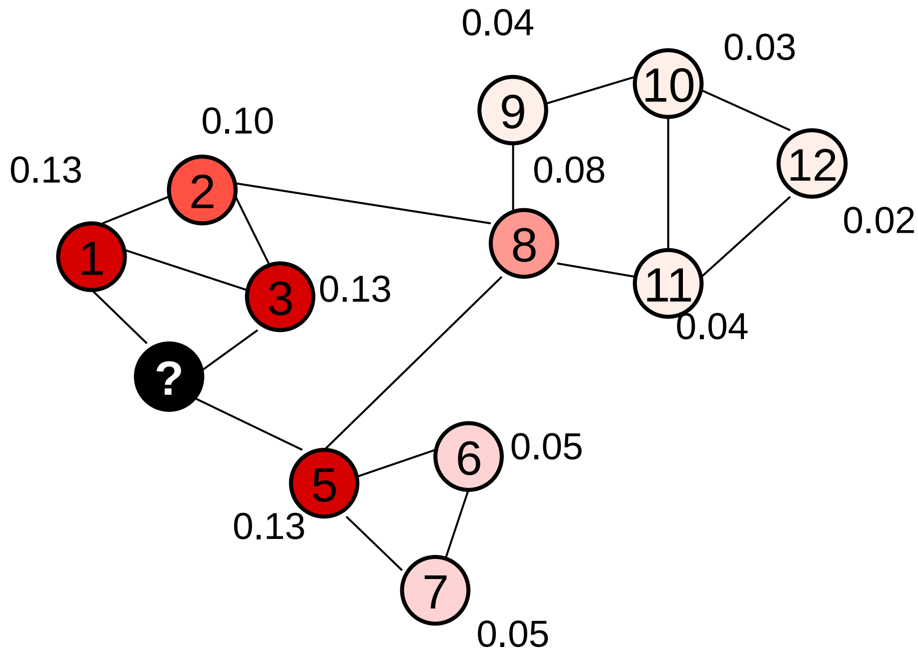
<u>Challenge 1: Label Sparsity</u>
Weighting Ghost Edges

- **Ghost edges increase the number of labeled neighbors per node.**

- **Key**: Ghost edge weights should correspond with correlation between node labels.
- **Conjecture**: Correlation is higher between labels of nodes that are "closer" to one another.

Challenge 1: Label Sparsity
Weighting Ghost Edges

- We measure node proximity using **random walk with restart** (RWR)



[Tong, Faloutsos, & Pan, ICDM 2006]

Challenge 2: Non-homphily

How to handle degrees of homophily?

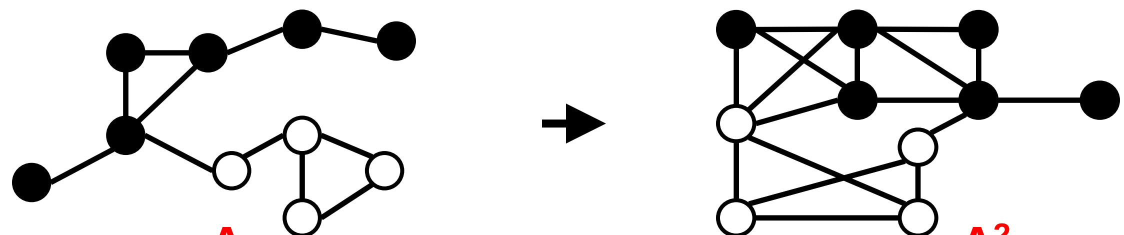
- The standard approach: use labeled data to learn dependencies
- Sparse labels make learning difficult
  - **Ghost edges increase the number of labeled neighbors per node**
- What if labels are extremely sparse?

Challenge 2: Non-homphily

How to handle degrees of homophily?

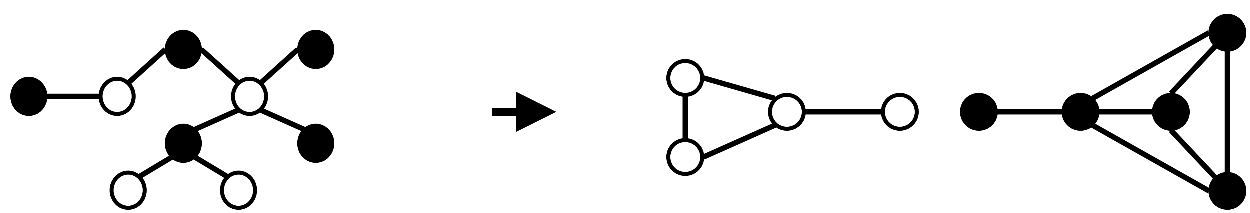
- **Homophily**

Even-step RWR



A A^2

- **Non-homophily**



B B^2

$$\vec{r}_i[t+1] = c\Delta^{2}\vec{r}_i[t] + (1-c)\vec{e}_i$$

$$\Delta = I - D^{-1/2} W D^{-1/2}$$

The GhostEdge Classifiers

- ***GhostEdgeNL***: non-learning method
  - Ignore observed edges.
  - Create ghost edges from unlabeled nodes to labeled nodes.
  - Take weighted vote of ghost edge neighbors.
- ***GhostEdgeL***: learning method
  - Uses labeled nodes to learn label-dependencies separately across for observed edge and ghost edges
  - Bins ghost edges by proximity scores and learn dependencies separately for each bin
  - Ensemble ***logForest*** classifier: Bag of logistic regression classifiers, each given a subset of features

Summary of Data Sets & Prediction Tasks

- **Enron**
  - Task: Executives?
  - |V| = 3081
  - |L| = 1055 ← **"Core" nodes**
  - |E| = 34,902
  - P(+) = 0.02
- **Political Books**
  - Task: Neutral?
  - |V| = 105
  - |L| = 105
  - |E| = 441
  - P(+) = 0.12
- **HEP-TH**
  - Task: Diff. Geometry?
  - |V| = 2999
  - |L| = 342 ← **"Core" nodes**
  - |E| = 36,014
  - P(+) = 0.06
- **Reality Mining**
  - Task: In-Study?
  - |V| = 1000
  - |L| = 1000
  - |E| = 31,509
  - P(+) = 0.08

Various Ways of Measuring Homophily on a Network

- ***Label* (or *Local*) *Consistency***
  - Ratio of links connecting nodes with the same class-label
- Neville & Jensen's ***Relational Autocorrelation***
  - Correlation measure on links connecting labeled nodes
- Newman's ***Assortative Mixing***
  - Bias in favor of connections between nodes with similar class-label
- Park & Barabasi's ***Dyadicity*** and ***Heterophilicity***
  - **Dyadicity:** connectedness between nodes with the same class-label compared to what is expected for a random configuration
  - **Heterophilicity:** connectedness between nodes with different class-labels compared to what is expected for a random configuration

Dyadicity and Heterophilicity

[Park & Barabasi, PNAS'07]

$$N = n_0 + n_1$$

$$M = m_{11} + m_{10} + m_{00}$$

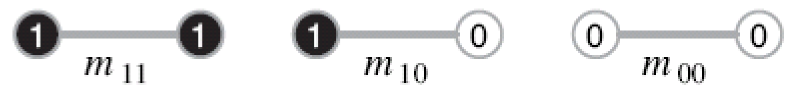
$$p = \frac{2M}{N(N-1)} \quad \textit{connectance}$$

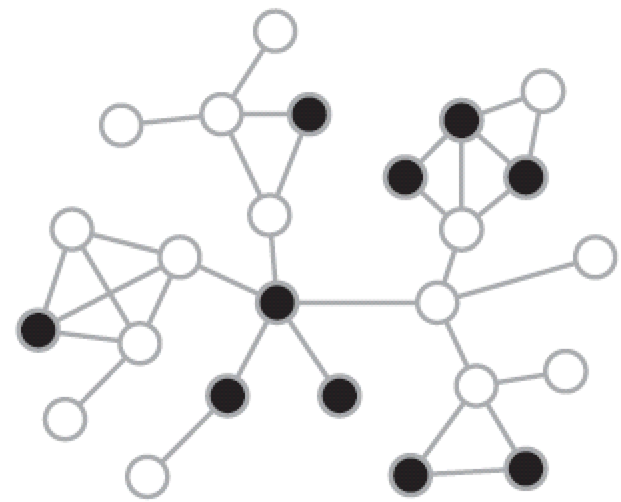
$$\overline{m_{11}} = \binom{n_1}{2} \times p = \frac{n_1(n_1 - 1)}{2} \times p$$

$$\overline{m_{10}} = \binom{n_1}{1}\binom{n_0}{1} \times p = n_1(N - n_1) \times p$$

$$D = \frac{m_{11}}{\overline{m_{11}}}$$

$$H = \frac{m_{10}}{\overline{m_{10}}}$$





N = 25

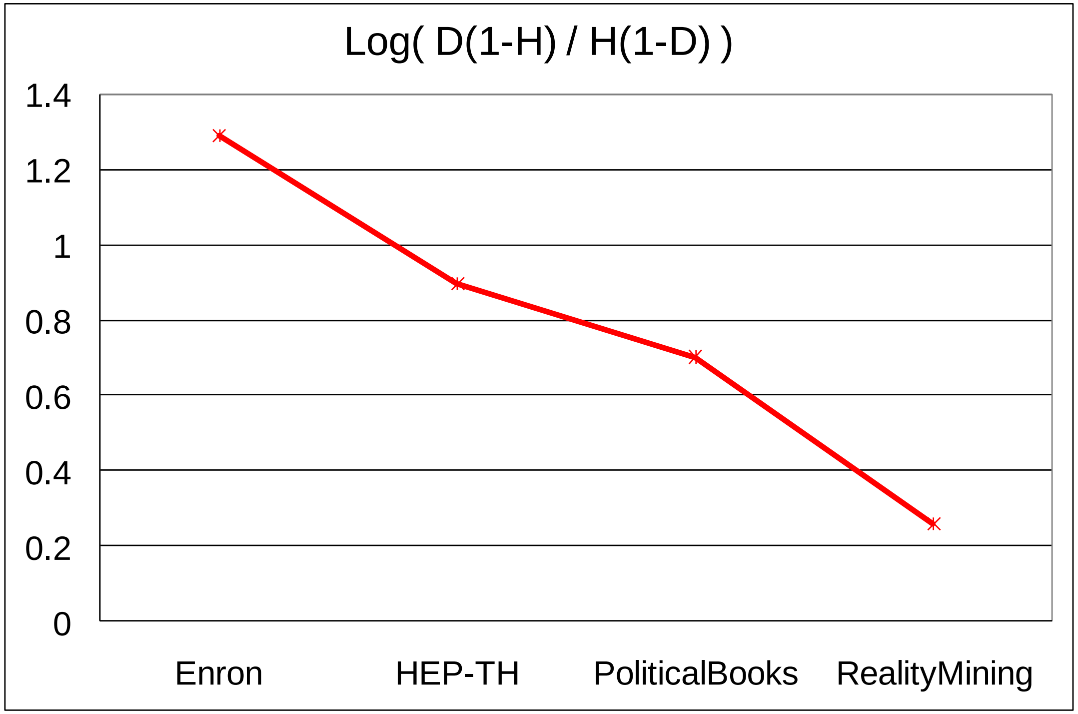
M = 32

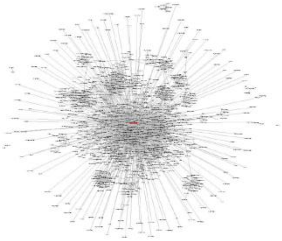
n_1 =10

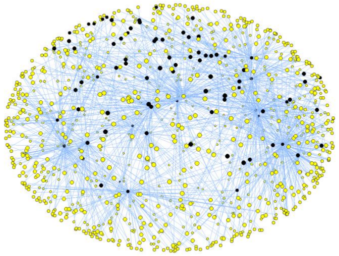
p = 0.11

Log Odds of Dyadicity & Heterophilicity

- Enron
  - Task: Executives?
  - P(+) = 0.02
- HEP-TH
  - Task: Diff. Geometry?
  - P(+) = 0.06
- Political Books
  - Task: Neutral?
  - P(+) = 0.12
- MIT Reality Mining
  - Task: In-Study?
  - P(+) = 0.08



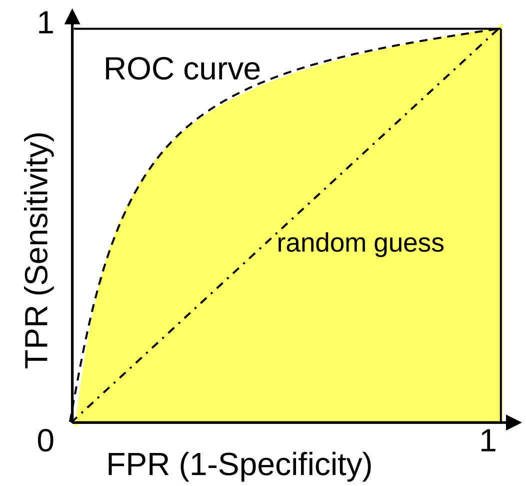




Experimental Methodology for Evaluating Within-Network Classifiers

- We **vary the proportion of core nodes labeled** from 10% to 90%
  - Remove labels from a class-stratified random sample of nodes
  - Labeled nodes are training instances
  - Unlabeled nodes are test instances
  - We use identical train/test splits for each classifier
  - We ensure that each node $i \in V$ occurs in the same number of test sets
- For each proportion labeled, we run **20 trials**
- We use ***Area Under the ROC curve*** **(*AUC*)** to measure classification performance

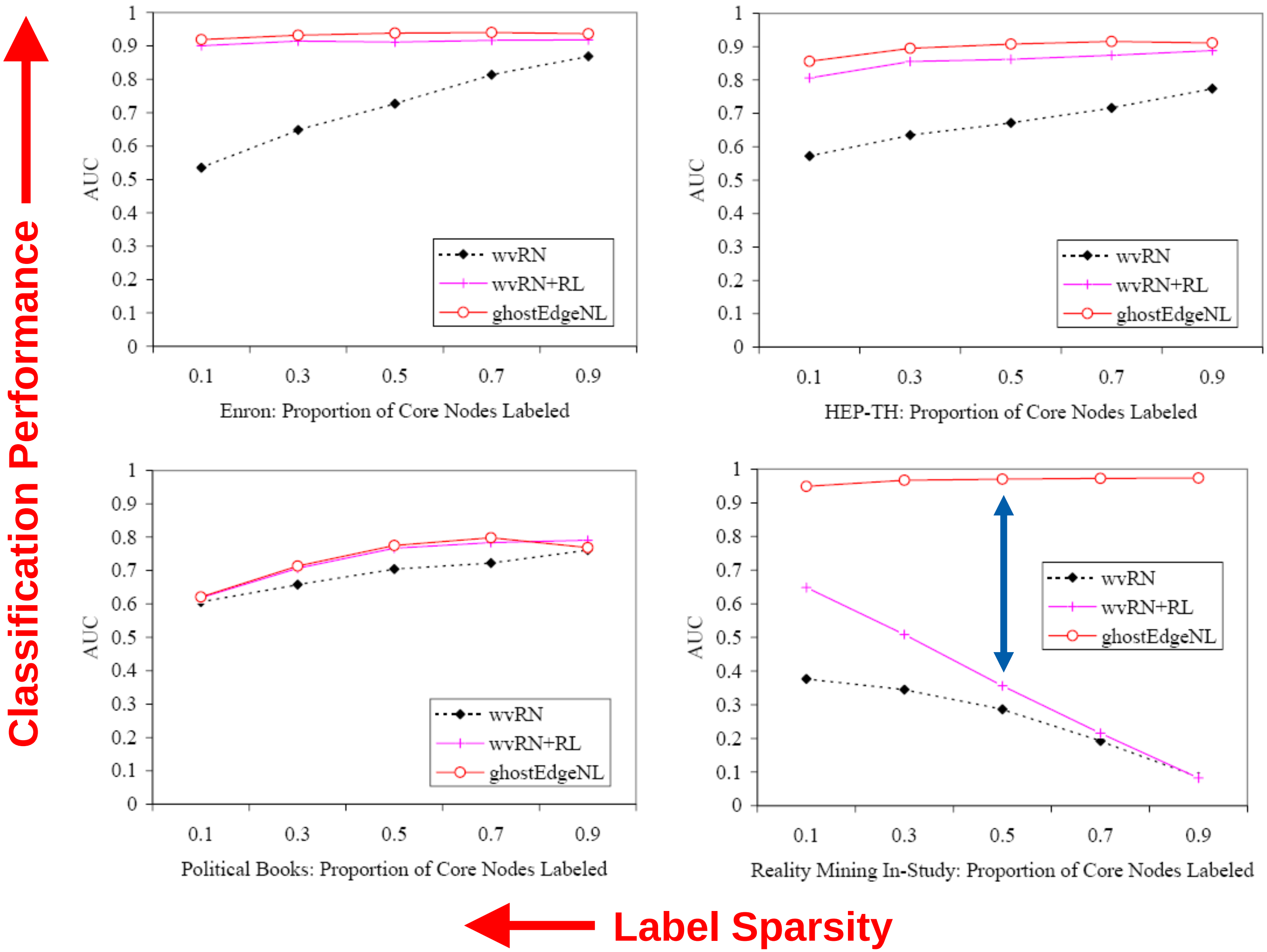
	Predicted Positive	Predicted Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative



We Ran Seven Individual Classifiers

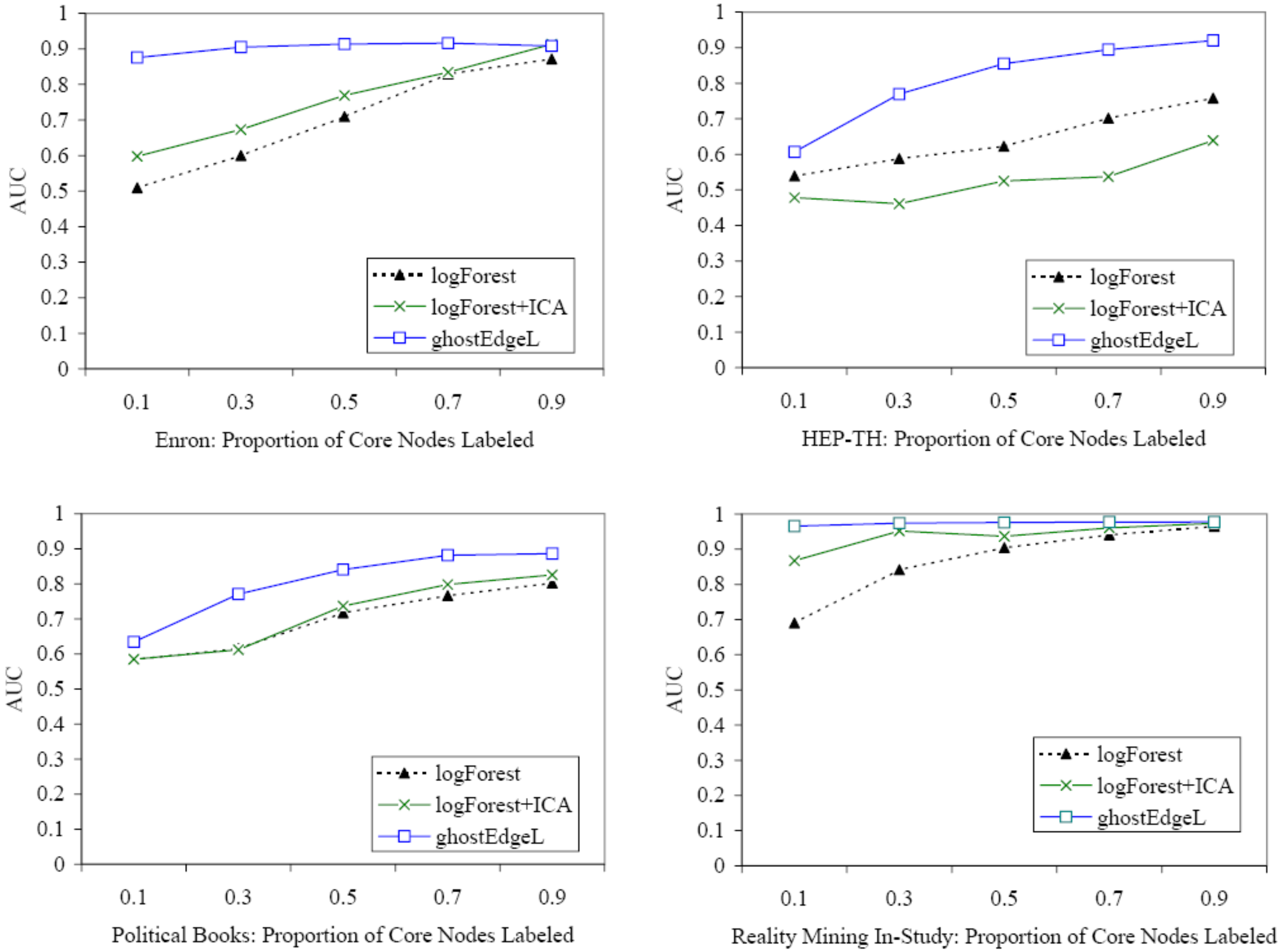
- <u>Relational Neighbor (Non-learning)</u>
  1. ***wvRN***: A relational neighbor model without collective classification
  2. ***wvRN+RL***: A relational neighbor model, which uses relaxation labeling for collective classification
  3. ***GRF***: Semi-supervised Gaussian random field model
  4. ***ghostEdgeNL***: Our ghostEdge-based classifier without learning
- <u>Link-Based (Learning)</u>
  5. ***logForest***: An ensemble logistic link-based model without collective classification
  6. ***logForest+ICA***: An ensemble logistic link-based model, which uses the iterative classification algorithm to perform collective classification
  7. ***ghostEdgeL***: Our ghostEdge-based classifier with learning

GhostEdgeNL is Top Performer Among Relational Neighbor Classifiers



GhostEdgeL is Top Performer Among Link-Based Classifiers

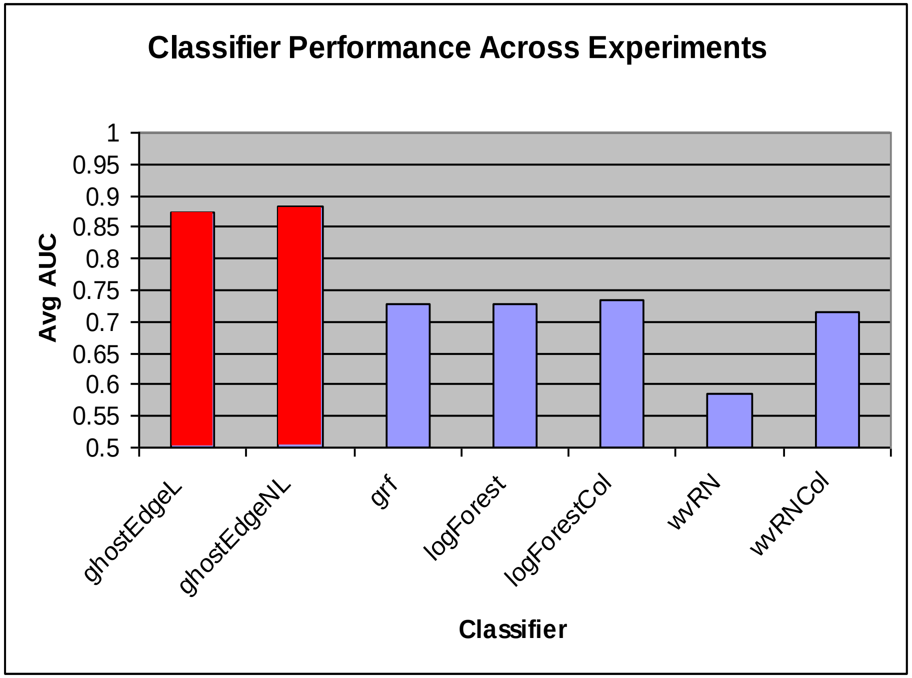
Classification Performance



Label Sparsity

Summary: Ghost Edges

- **C1: Label sparsity**
  - A1: Ghost Edges increase the number of labeled neighbors per node.
- **C2: Non-homophily**
  - A2a: GEs improve learning.
  - A2b: Even-step RWR maintains or increases relational autocorrelation.
- **Practitioner's guide**
  - Use GhostEdgeNL in extreme sparsity, high homophily
  - Use GhostEdgeL in moderate sparsity, non-homophily



- **Details in KDD'08,** also related AI Magazine '08, SNAKDD'08, ICDM'09, KAIS'11

Outline

- Problem #1: Network (a.k.a. relational) classifiers
- Problem #2: Clustering on networks (a.k.a. community discovery)
- Conclusions

Community Discovery

- Given a **graph** $G=(V, E)$
- Want a **community discovery procedure** with the following properties
  1. **Scalable**, where time and space complexity are strictly sub-quadratic w.r.t. the number of nodes
  2. **Nonparametric**, where number of communities need not be specified *a priori*
  3. **Consistent**, where effectiveness is consistently high across a wide range of domains
  4. **Effective**, where global connectivity patterns are successfully factored into communities that are highly predictive of individual links and robust to small perturbations in network structure

Measures of Effectiveness

- **Link prediction**
  - A good factorization of a graph's connectivity structure should accurately predict links based on the endpoints' respective communities: $P(s \rightarrow t \mid z_s, z_t)$
  - Measured by **Area Under the ROC** (AUC) on predictions of randomly held-out links
- **Robustness**
  - Community structures that are "significant / believable" should be able to withstand small perturbations in the network structure
    - [Karrer, Levina, and Newman, Phys. Rev. E. 2008]
  - Measured by **Variation of Information**: an entropy-based distance metric

Background

Hard Clustering

- **Fast modularity (FM)**
  - Maximizes modularity
  - A good clustering is one that maximizes intra-group links and minimizes inter-group links
- **Cross Associations (XA)**
  - Based on compression
  - Minimizes total encoding cost
  - A good clustering is one that produces dense “co-clusters”
    - Looks for same patterns of connectivity between nodes

Soft Clustering

- **Latent Dirichlet Allocation for Graphs (LDA-G)**
  - Nonparametric Bayesian model

    $$P(Z \mid R) \propto P(R \mid Z)\, P(Z)$$

  - Finds soft groups with mixed-memberships
  - Maximizes likelihood
  - A good clustering is one that finds multinomial distributions over all nodes that accurately describe which nodes are most associated with which communities and which ones are not

Modularity (FM)
[Clauset+, Phys. Rev. E. 2004]

- m = number of edges in the graph
- A_{vw} = 1 if $v \rightarrow w$; 0 otherwise
- k_v = degree of vertex v
- $\delta(i, j)$ = 1 if $i == j$; 0 otherwise
- Maximizes modularity, Q: measures the **fraction of all edges within groups** minus **the expected number in a random graph with the same degrees**

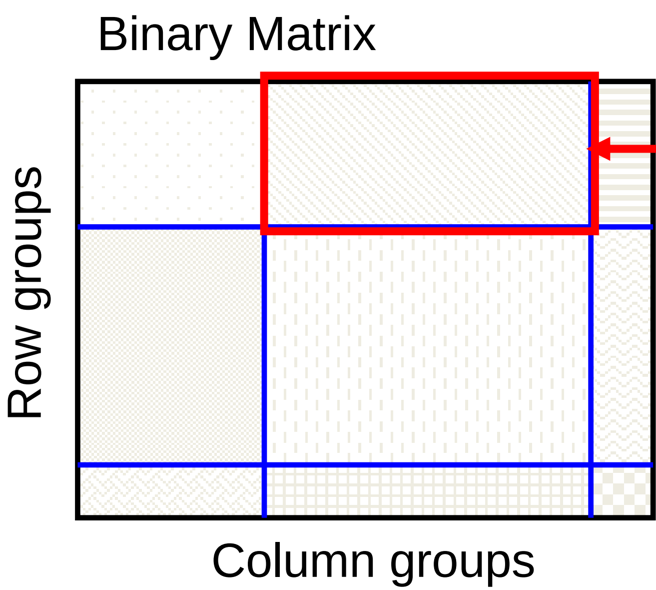
$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w)$$

- Produces "hard groups," where each vertex has a single group assignment
- Runtime complexity for $G=(V, E)$ is $O(\,|V| \log^2(|V|)\,)$

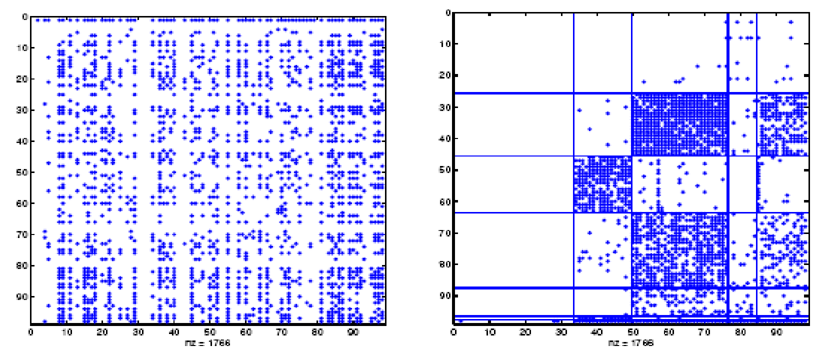
Cross-Associations (XA)
[Chakrabarti+, KDD 2004]

- Minimizes **total encoding cost** of the adjacency matrix = code cost + description cost

$$= \sum_i \left((n_i^1 + n_i^0) \times H(p_i^1) \right) + \sum_i \left(\text{cost of describing } n_i^1,\ n_i^0 \text{ and groups} \right)$$



$p_i^1 = n_i^1 / (n_i^1 + n_i^0)$



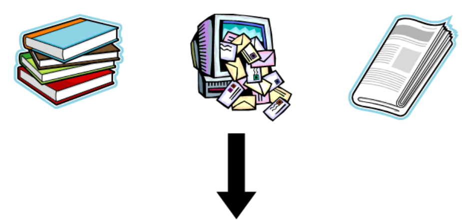
(a) before (b) after

- Produces row-groups and column-groups
- Runtime complexity for $G=(V, E)$ is $O(|E|)$

Latent Dirichlet Allocation for Graphs (LDA-G) [Henderson & Eliassi-Rad, ACM SAC 2009]

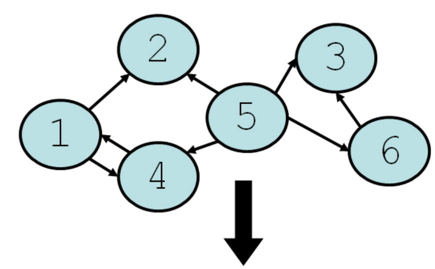
- Based on LDA [D. Blei+, JMLR 2003]
  - Nonparametric generative model for topic discovery in text documents
    - Number of topics is learned, not fixed
  - LDA takes a set of documents (represented as a bag of words)
  - Each document is a mixture of topics from a multinomial distribution
  - Each word is drawn from one of the topics
  - Uses infinite (Dirichlet) priors on documents and topics

LDA



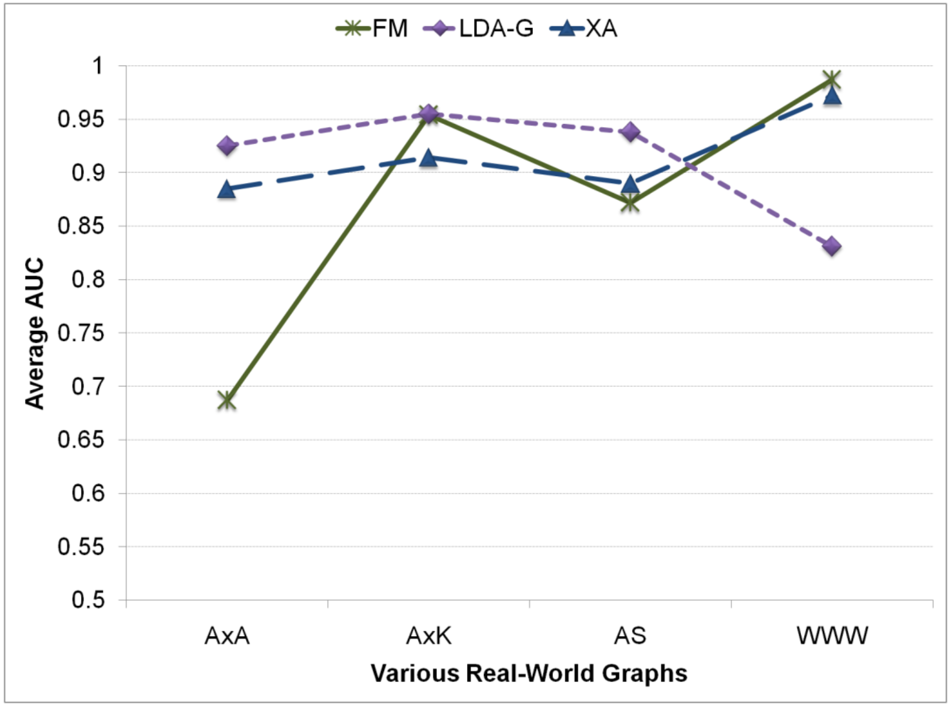
Doc 1	it, was, the, best, of, times, …
Doc 2	hi, mom, just, writing, to, let, you, know, …
Doc 3	we, hold, these, truths, to, be, self, evident, …
…	…

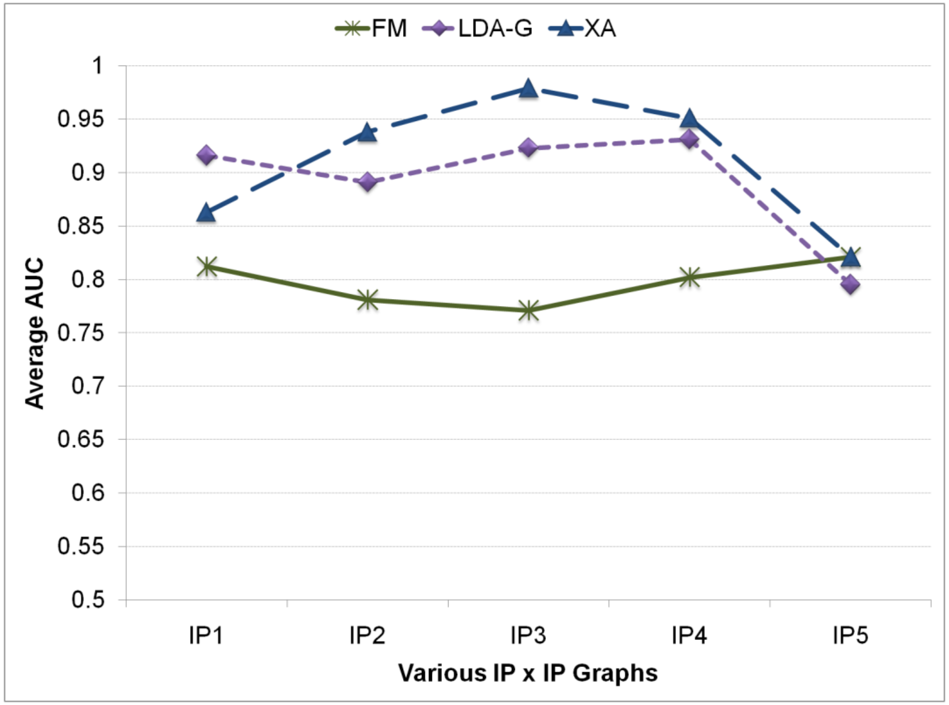
LDA-G



Doc 1	2, 4
Doc 2	<empty>
Doc 3	<empty>
Doc 4	1
Doc 5	2, 3, 4, 6
Doc 6	3

LDA-G, FM, and XA are not Consistent w.r.t. Link Prediction





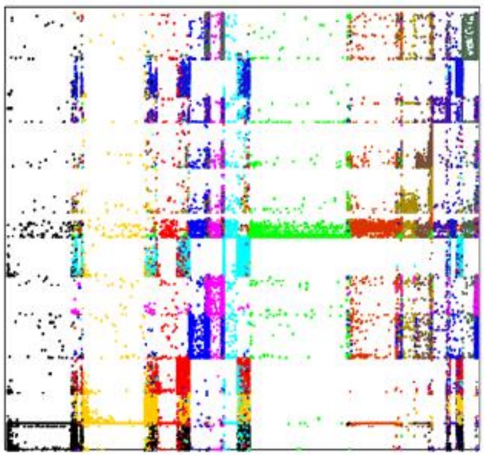
- Why are they not consistent?
- Can we fix it?

Hybrid Community Detection Framework (HCDF)

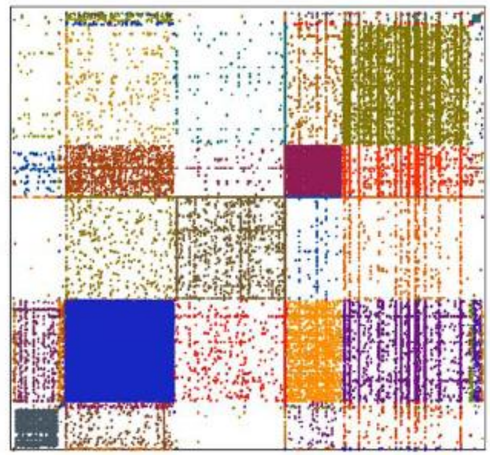
- HCDF is a Bayesian framework
- Incorporates communities discovered via other, non-Bayesian approaches, as **hints**
- Consists of two parts: (1) hint-giver and (2) hint-taker
- Leads to **improved effectiveness* of results compared to its constituents and consistency across various domains**
  - Hard clustering may not be able to explain all of a node's links
  - Soft clustering with mixed membership may get confused by giving a node "uniform" memberships across communities

* W.r.t. link prediction and robustness to small network perturbations

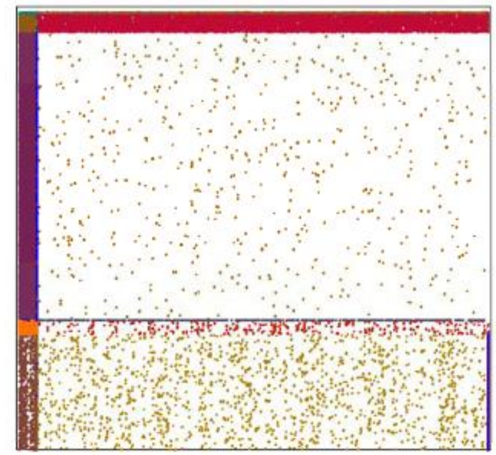
More Whitespace, Higher Average AUC on Link Prediction



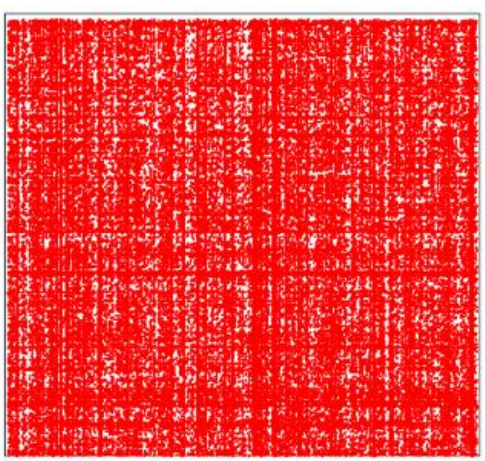
LDA-G
(Average AUC: 0.795)



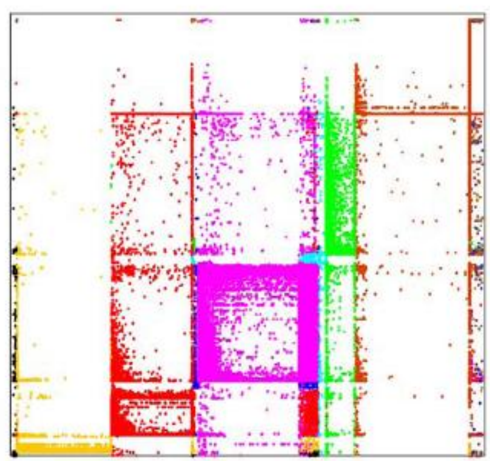
FM
(Average AUC: 0.821)



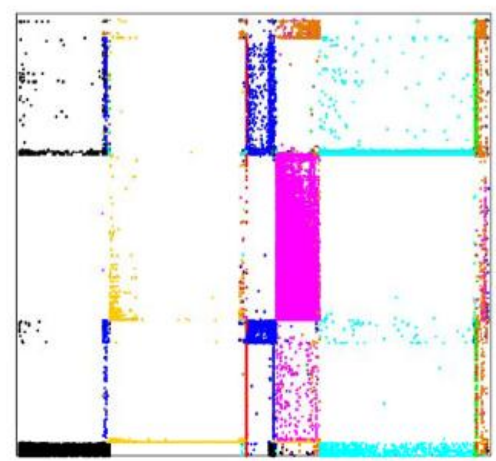
XA
(Average AUC: 0.821)



Adjacency Matrix for IP5



HCD-M
(Average AUC: 0.955)



HCD
(Average AUC: 0.968)

Hybrid Community Detection Framework (HCDF)

- **Hint-givers:** Non-Bayesian algorithms used in HCDF
  - Cross-Associations (XA) [Chakrabarti+, KDD'04], <u>Runtime: $O(|E|)$</u>
  - Fast Modularity (FM) [Clauset+, Phy. Rev. E'04], <u>Runtime: $O(|V| \bullet \log^2(|V|))$</u>
- **Hint-taker:** Bayesian algorithm used in HCDF
  - LDA-G [Henderson & Eliassi-Rad, ACM SAC'09]
    - Uses Gibbs sampling to infer posterior estimates, <u>Runtime: $O(|E|)$</u>
- Three different strategies for incorporating hints
  1. **Seed**, which used hints only as an initial configuration for LDA-G's inference procedure
  2. **Prior**, which propagates hints from one configuration to the next
  3. **Attribute**, which incorporates hints as additional link-attributes

HCDF with Attribute Coalescing Strategy

- Run XA (or FM) on input G=(*V*, *E*)
  - Produces groups, *A*, over nodes
- Run LDA-G on graph *G'* = (*V*, *E*, *A*)

LDA-G Model

$v_i | z_i, \phi^{(z_i)} \sim Discrete(\phi^{(z_i)})$ ------------ Multinomial from groups to target-nodes

$\phi \sim Dirichlet(\beta)$ ------------ Prior on target-nodes (**observables**)

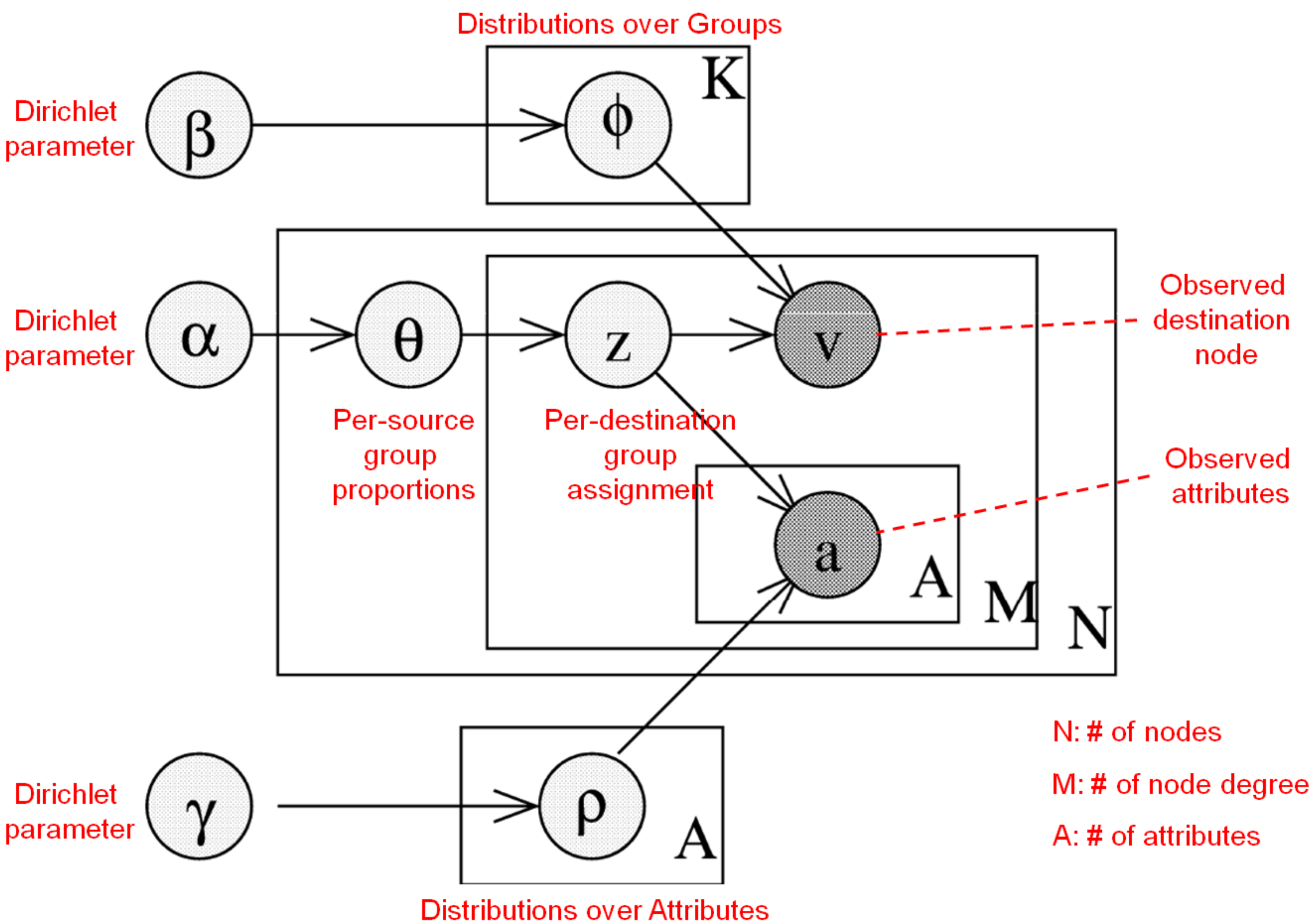
$z_i | \theta^{u_i} \sim Discrete(\theta^{u_i})$ ------------ Multinomial from source-nodes to groups

$\theta \sim Dirichlet(\alpha)$ ------------ Prior on groups (**latent variable**)

$a_{ij} | z_i, \rho_j^{(z_i)} \sim Discrete(\rho_j^{(z_i)})$ ---------- Multinomial from groups to link-attributes

$\rho \sim Dirichlet(\gamma)$ ------------ Prior on link-attributes (**observables**)

Plate Model for HCDF with Attribute Coalescing Strategy



HCD Algorithm

Algorithm 1 *HCD*(G)

Require: Graph: $G = [V, E]$

/* Run Cross-Association on G */

$[rowGroups, colGroups] = XA(G)$

$R = [rowGroups, colGroups]$

Ensure: $\forall i \in [1, A]$ and $\forall <u, v> \in E$: $R \equiv \{a_i^{<u,v>}\}$

/* Run *LDA-G* on graph G with attributes R and hyperparameters $\gamma \approx 10$ and $\alpha = \beta \approx 1$ */

/* Set the prior */

$a_{i1} \leftarrow a_i^{<u,\cdot>}$; $a_{i2} \leftarrow a_i^{<\cdot,v>}$

Initialize all count variables $n_u, n_u^k, n_k, n_k^v, n_k^{a_{i1}}, n_k^{a_{i2}}$ to 0

for each link $<u, v> \in E$ **do**

  Sample community $z_i = k$ using Equation 8

  Increment count variables $n_u, n_u^k, n_k, n_k^v, n_k^{a_{i1}}, n_k^{a_{i2}}$ by 1

end for

/* Run Gibbs sampling */

for each edge $<u, v> \in E$ with in community k **do**

  Decrement count variables $n_u, n_u^k, n_k, n_k^v, n_k^{a_{i1}}, n_k^{a_{i2}}$ by 1

  Sample community $z_i = k$ using Equation 8

  Increment count variables $n_u, n_u^k, n_k, n_k^v, n_k^{a_{i1}}, n_k^{a_{i2}}$ by 1

end for

$$p(z_i|\mathbf{z_{-i}}, \mathbf{u}, \mathbf{v}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_A}) \propto \frac{n_u^k + \alpha}{n_u + \alpha K} \cdot \frac{n_k^v + \beta}{n_k + \beta N} \prod_{i=1}^{A} \frac{n_k^{a_i} + \gamma}{n_k + A_i \gamma}$$

Summary of Real-World Graphs Used in Experiments

Real-World Graphs	Acronym	$\mid V \mid$	$\mid E \mid$
Autonomous Systems Graph	AS	11,461	32,730
Day 1: IP $\times$ IP	IP1	34,449	303,175
Day 2: IP $\times$ IP	IP2	33,732	320,754
Day 3: IP $\times$ IP	IP3	34,661	428,596
Day 4: IP $\times$ IP	IP4	34,730	425,368
Day 5: IP $\times$ IP	IP5	33,981	112,271
PubMed Author $\times$ Knowledge	AxK	37,436 (A) 117 (K)	119,443
PubMed Coauthorship	AxA	37,227	143,364
WWW Graph	WWW	325,729	1,497,135

Summary of Real-World Graphs Used in Experiments (cont.)

Acronym	$\mid V \mid$	$\mid E \mid$	# Components	% of V in LCC
AS	11,461	32,730	1	1
IP1	34,449	303,175	4	99.98%
IP2	33,732	320,754	8	99.96%
IP3	34,661	428,596	2	99.99%
IP4	34,730	425,368	2	99.99%
IP5	33,981	112,271	13	99.92%
AxK	37,346 (A) & 117 (K)	119,443	1	1
AxA	37,225	143,364	4,556	23.54%
WWW	325,729	1,497,135	1	1

Summary of Real-World Graphs Used in Experiments (cont.)

Data Graph	Average Degree	Clustering Coefficient	Average Path	Diameter	# of Articulation Points	% of V that are Articulation Points
AS	2.86	0.258	3.67	11	828	7.2%
IP1	8.80	0.198	3.23	7	1,258	3.7%
IP2	9.51	0.18	3.22	8	1,208	3.6%
IP3	12.37	0.198	3.04	6	920	2.7%
IP4	12.25	0.216	3.07	7	841	2.4%
IP5	3.30	0.058	3.54	7	1,524	4.5%
AxK	3.19	0	1.00	1	54	0.1%
AxA	3.85	0.49	8.85	23	1,467	3.9%
WWW	4.60	0.28	11.38	58	21,780	6.7%

Measuring Effectiveness Quantitatively: Link Prediction

- If you are building groups from the graph structure **only**, then those groups should be able to predict the structure back

- Link prediction in LDA-G: *u* → *v*

$$P(edge(v) \mid u) = \sum_{g \in Groups} \big(P(edge(v) \mid g) \times P(g \mid u) \big)$$

- Link prediction in FM and XA is based on density: *u* → *v*

$$\frac{\#\text{of edges from } group(u) \text{ to } group(v)}{\#\text{of possible edges from } group(u) \text{ to } group(v)}$$

Measuring Effectiveness Quantitatively: Link Prediction

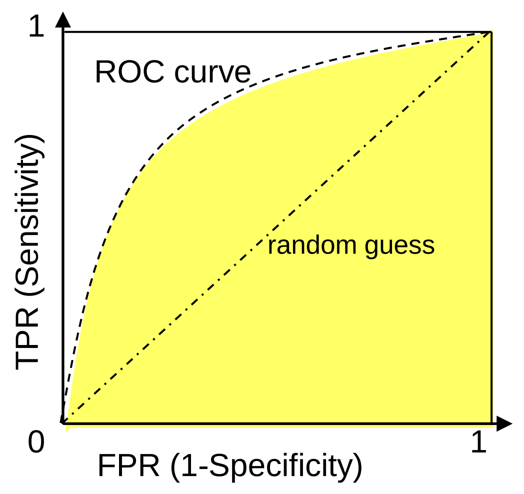
- A good factorization of a graph's connectivity structure can accurately predict links between nodes based on their respective communities
  - $P(s \rightarrow t \mid s, t, z_s, z_t)$
- Evaluate effectiveness by
  1. randomly holding out a number of links
  2. building a model
  3. using learnt model to predict held-out links
  4. measuring performance with area under ROC curve (AUC)

0	1	0	1	1
1	0	1	0	1
0	1	0	0	0
1	0	0	0	1
1	1	0	1	0

0	1		1	
1	0		0	1
		0	0	
1	0	0	0	1
	1		1	0

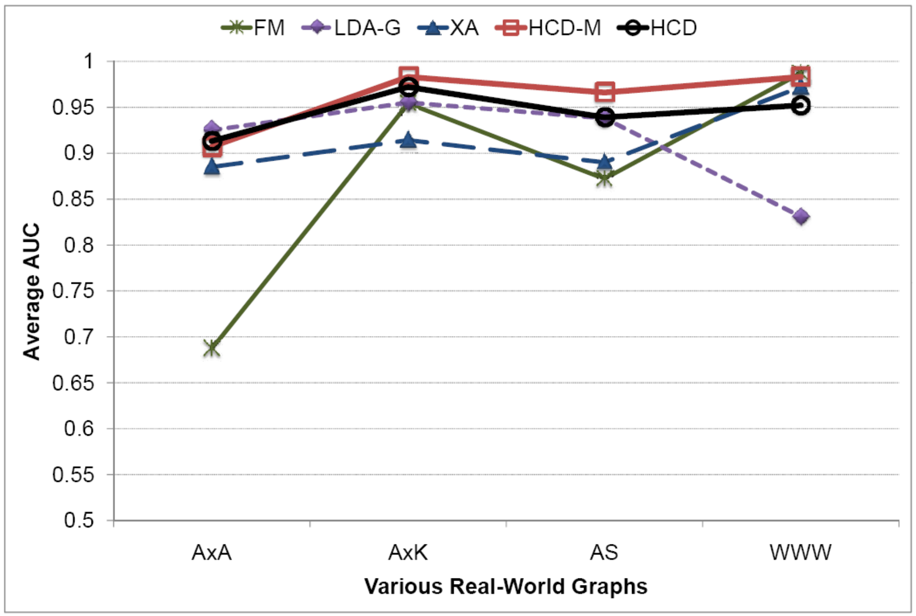
Experimental Methodology

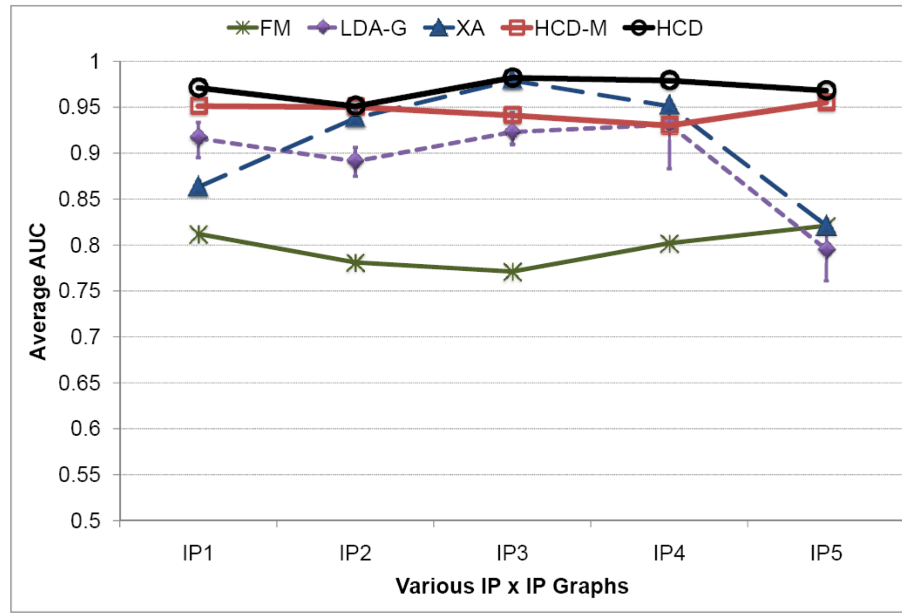
- Randomly select 500 present- and 500 absent-links
  - This is the held-out test
- Give the remaining links to each algorithm to find groups
- Each algorithm uses its discovered groups to estimate the probability that links in the held-out test set were present
- Use estimates to calculate the area under the ROC curve (AUC)
- Repeat the above process 5 times and report the average AUC



Link Prediction Performance Across Various Graphs

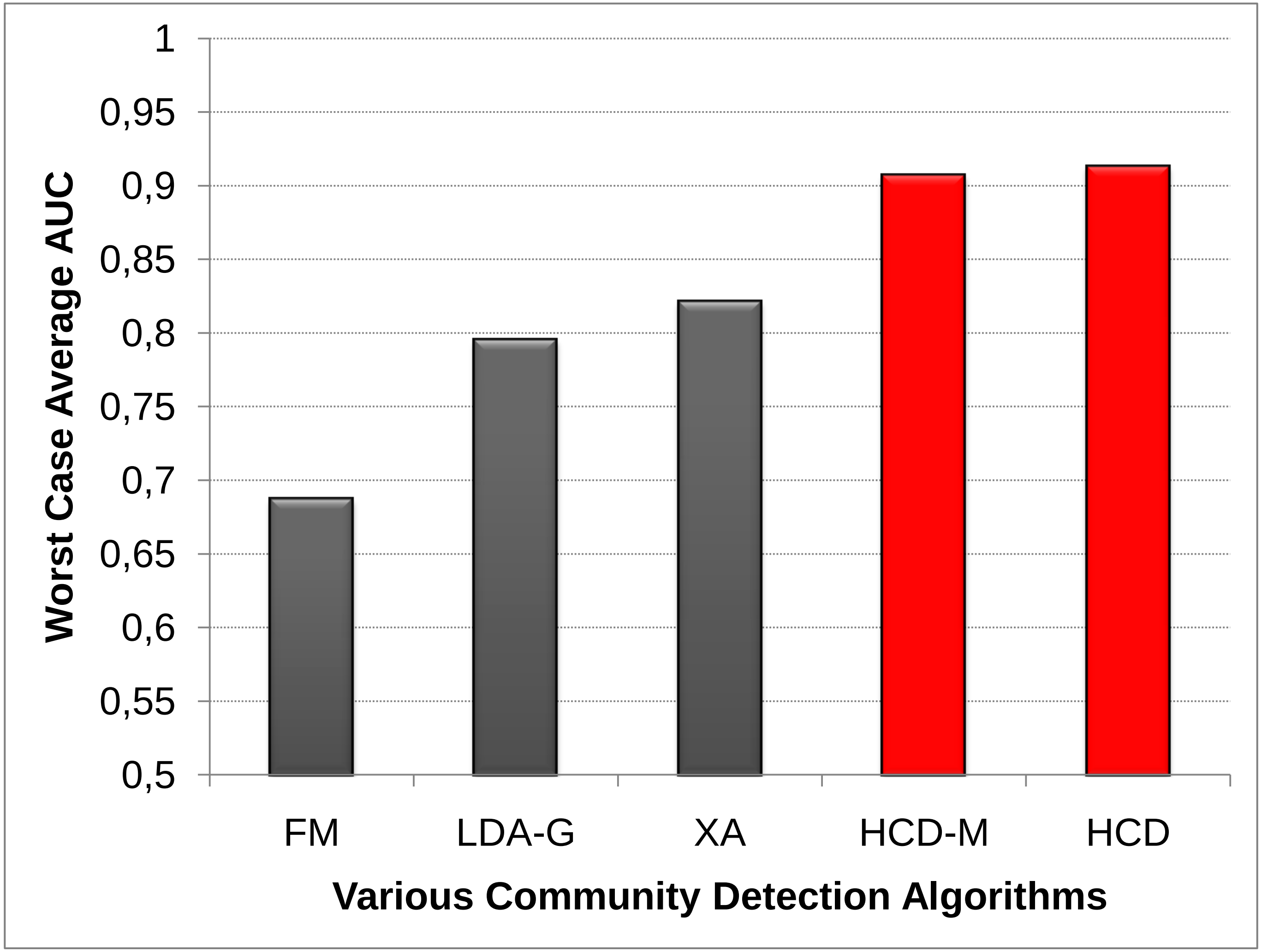
Higher is better





Hybrid methods' effectiveness w.r.t. link prediction is consistently high (≥ 0.9 AUC) across various domains

Worst-Case Link Prediction Performance Across All Graphs

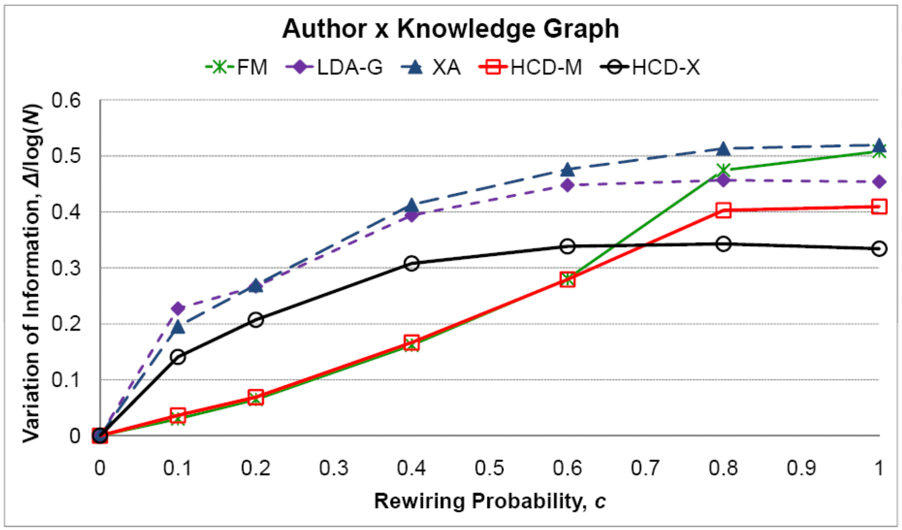


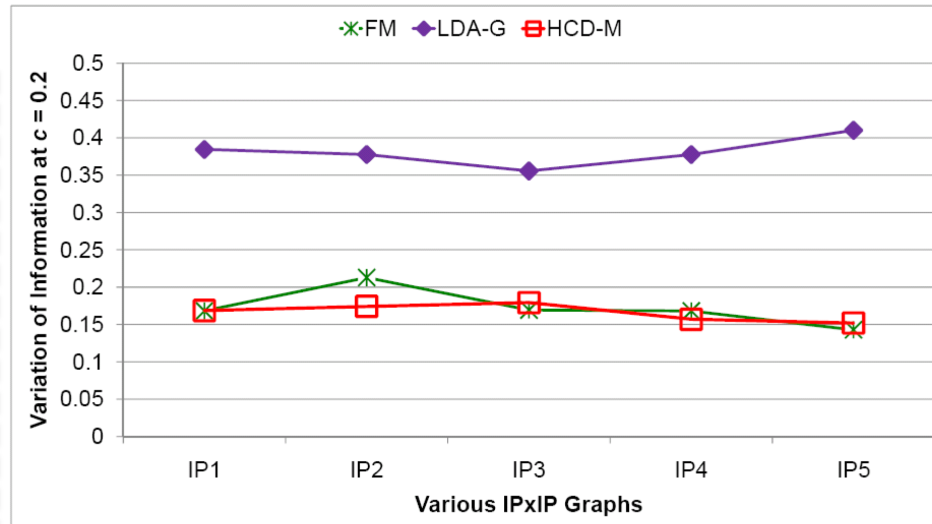
Measuring Effectiveness Quantitatively: Value of Information

- Perturb graph by randomly reassigning a number of its links
  - Rewiring parameter $c \in [0,1]$ determines fraction of links rewired
  - Links are rewired in a way that preserves the expected degree of each node in graph
- C = communities discovered on original graph, where $c = 0$
- C' = communities discovered on perturbed graphs, where $c \neq 0$
- Variation of Information, $\Delta(C, C') = \mathrm{H}(C|C') + \mathrm{H}(C'|C)$
  - $\mathrm{H}(C'/C)$ measures the information needed to describe C' given C
  - $\Delta(C, C') \in [0, \log(N)]$ treats each assignment as a message
    - Is a symmetric entropy-based measure of the distance between these messages

[Karrer, Levina, and Newman, Phys. Rev. E. 2008]

Robustness Measured across Various Real-World Graphs





Lower is better

Hybrid methods' effectiveness w.r.t. robustness is always better than or comparable to their constituents

Recall: FM shows poor link prediction performance on IP graphs

What’s Going on Here?

- Good link prediction can be thought of as a tradeoff between
  - Low entropy: If the adjacency matrix can be **compressed nicely** or mixed-membership distributions are **far from uniform**, we can better predict behavior of nodes
  - Flexibility: If a node exhibits multiple types of behavior, hard clustering may only model **a plurality** of the node’s edges
- LDA-G can generate high-entropy distributions
  - HCD-X and HCD-M use low-entropy hints when LDA-G is ambivalent
- XA and FM cannot model mixed behavior
  - HCD-X and HCD-M relax the hard clusters into soft clusters which can explain all links

What About a Super-Hybrid?

- Since HCD and HCD-M perform well, why not use hints from both XA and FM algorithms?
- We tried this, and it doesn’t work reliably
- When XA and FM disagree on groups, the “super-hybrid” performs well
- When they agree, the super-hybrid performs no better than HCD and HCD-M

Summary: Community Discovery

- Use a **hybrid approach** to community discovery on graphs for consistently effective community factorization across graphs from various domains
- Incorporate **hints as attributes** for coalescing strategy
- Use **link prediction and variation of information** as a quantitative measure on the communities discovered
- **Details in SDM'10,** also related NIPS Wkshp '09, ACM SAC'09, AAAI-SS'08, CIKM'08, DMKD'08

Outline

- Problem #1: Network (a.k.a. relational) classifiers
- Problem #2: Clustering on networks (a.k.a. community discovery)
- Conclusions

Problems

Applications

Network Classifiers

Transfer Learning

Statistical Tests for Relational Classifiers

Community Discovery

Anomaly Detection

Re-identification

Pattern Matching

Link Analysis

Knowledge Representation

Humanities

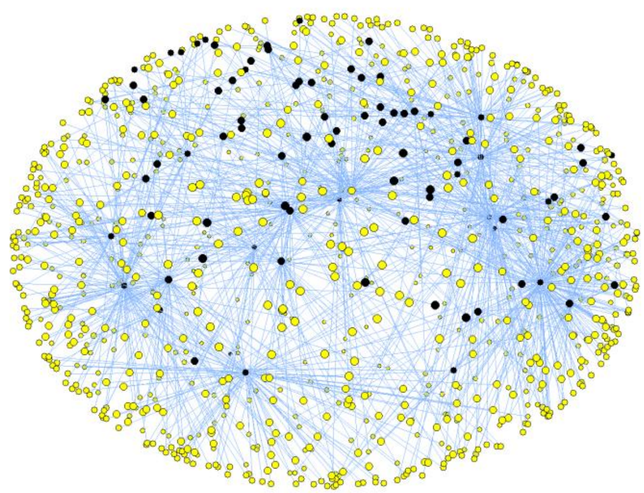
Cyber Situational Awareness

Social Science

Marketing

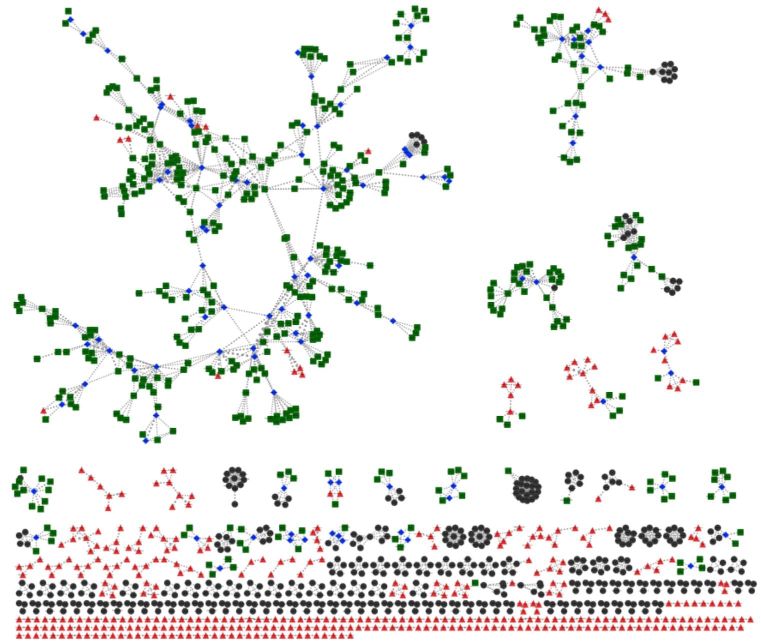
Search

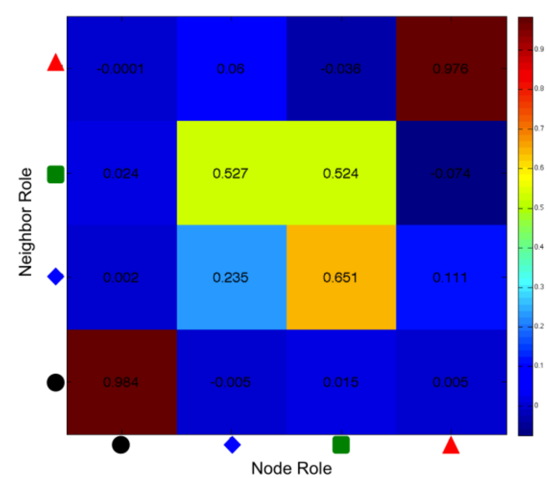
Smart Meters



Conclusions

- Complex networks are ubiquitous
- Lots of cool problems w.r.t. classification, clustering, and anomaly detection with real-world applications
  - Some solutions: Ghost Edges, HCDF
- Current & future work
  - A framework for capturing behavior in networks [KDD’11]





Thank You

- Papers available at http://eliassi.org/pubs.html
- Thanks to my collaborators on these works
  - LLNL: Brian Gallagher, Keith Henderson
  - CMU: Christos Faloutos+
  - Maryland: Lise Getoor+
  - Purdue: Jen Neville, Tao Wang
  - UC Berkeley: Kurt Miller
  - Google Labs: Spiros Papadimitriou
  - IBM Watson: Hanghang Tong
- Supported by AFRL, DTRA, LLNL, NSF

The End