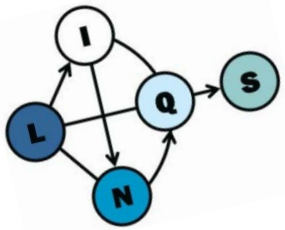
Collective Graph Identification

Lise Getoor

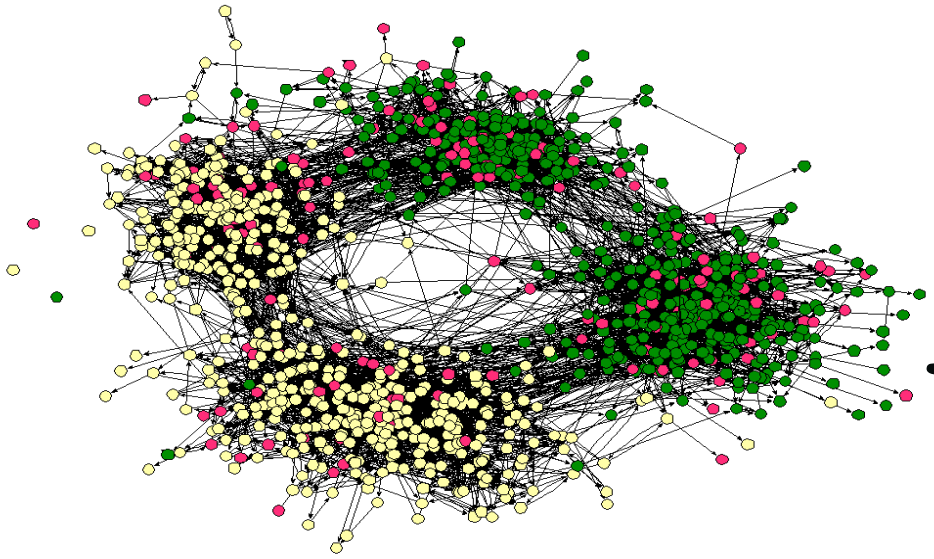
University of Maryland, College Park

Joint work with Galileo Namata

DIMACS/CCICADA Workshop on Data Quality Metrics
Feb 3, 2011



Motivation: Network Analysis



Network

+

Network Analysis

=

Who are the "central" individuals?

What are the communities?

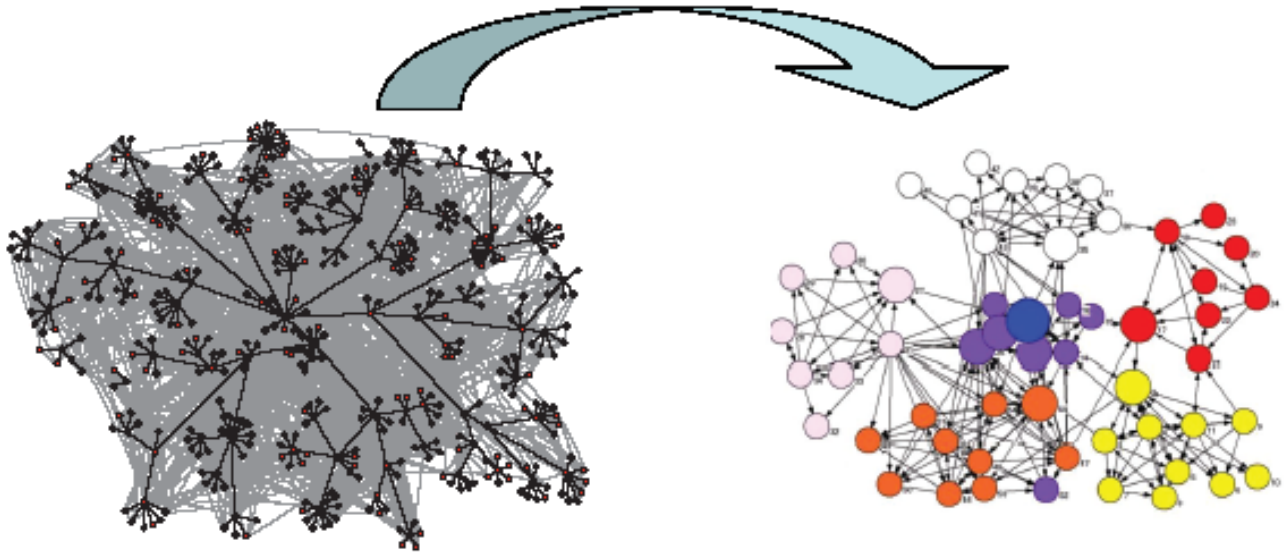
What are the common interaction patterns/motifs?

Wealth of Data

- Inundated with data describing networks
- But much of the data is
  - noisy and incomplete
  - at WRONG level of abstraction for analysis

Graph Identification

Graph Transformations



Data Graph ⇒ Information Graph

1. Entity Resolution: mapping email addresses to people
2. Link Prediction: predicting social relationship based on communication
3. Collective Classification: labeling nodes in the constructed social network

HP Labs, Huberman & Adamic

Overview: Graph Identification

- Many real world datasets are relational in nature
  - Social Networks – people related by relationships like friendship, family, enemy, boss_of, etc.
  - Biological Networks – proteins are related to each other based on if they physically interact
  - Communication Networks – email addresses related by who emailed whom
  - Citation Networks – papers linked by which other papers they cite, as well as who the authors are
- However, the observations describing the data are noisy and incomplete
- **graph identification problem** is to **infer** the appropriate **information graph** from the **data graph**

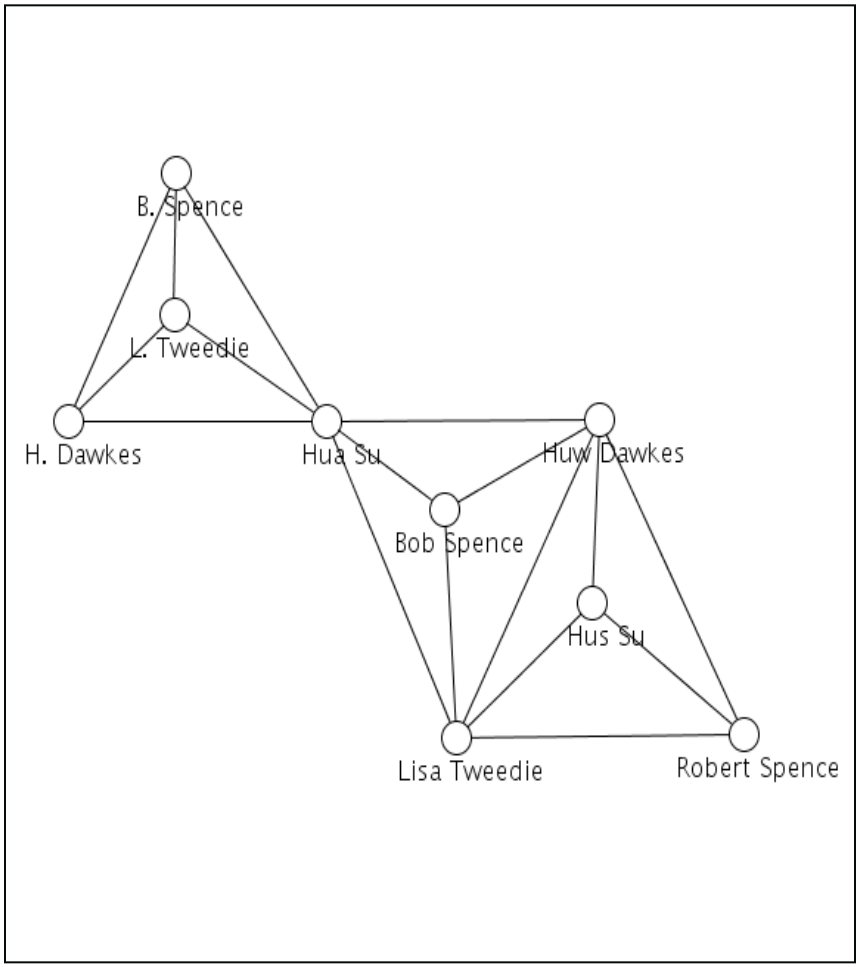
Roadmap

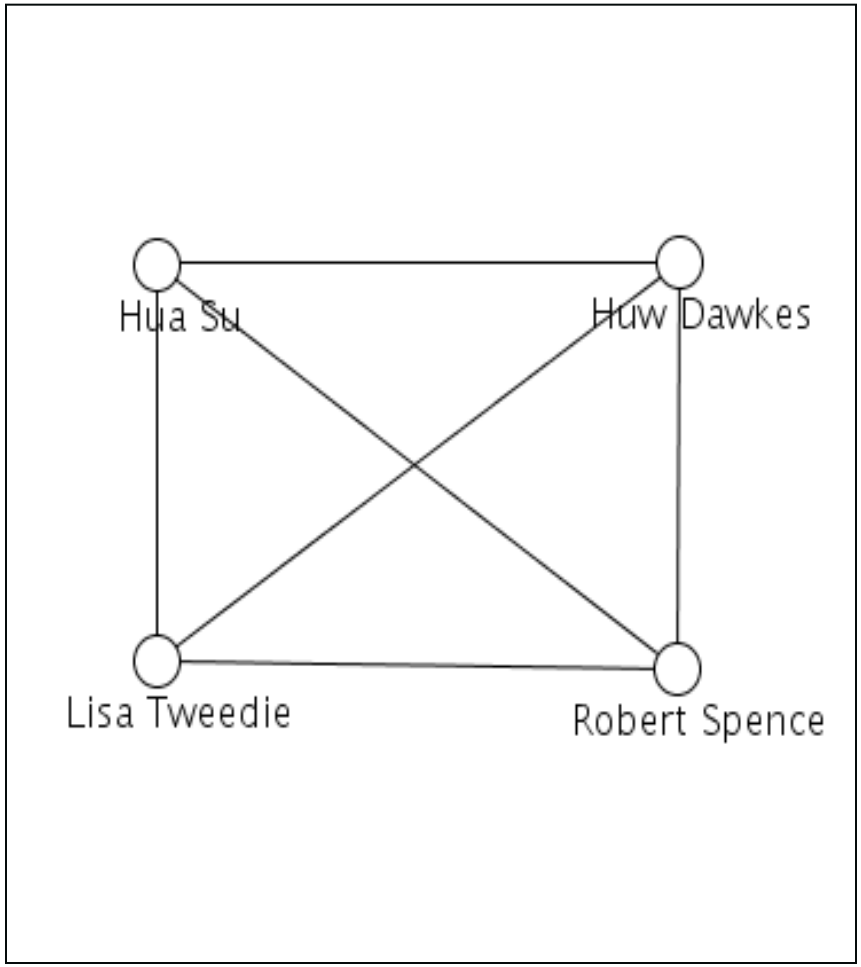
- The Problem
- **The Components**
  - Entity Resolution
  - Collective Classification
  - Link Prediction
- Putting It All Together
- Open Questions

Entity Resolution

- **The Problem**
- Relational Entity Resolution
- Algorithms

InfoVis Co-Author Network Fragment





before

after

The Entity Resolution Problem

John Smith

"John Smith"

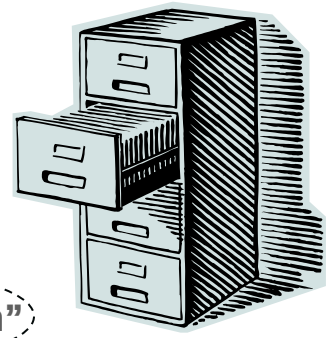
"J Smith"

James Smith

"Jim Smith"

"James Smith"

"J Smith"



Jonathan Smith

"Jon Smith"

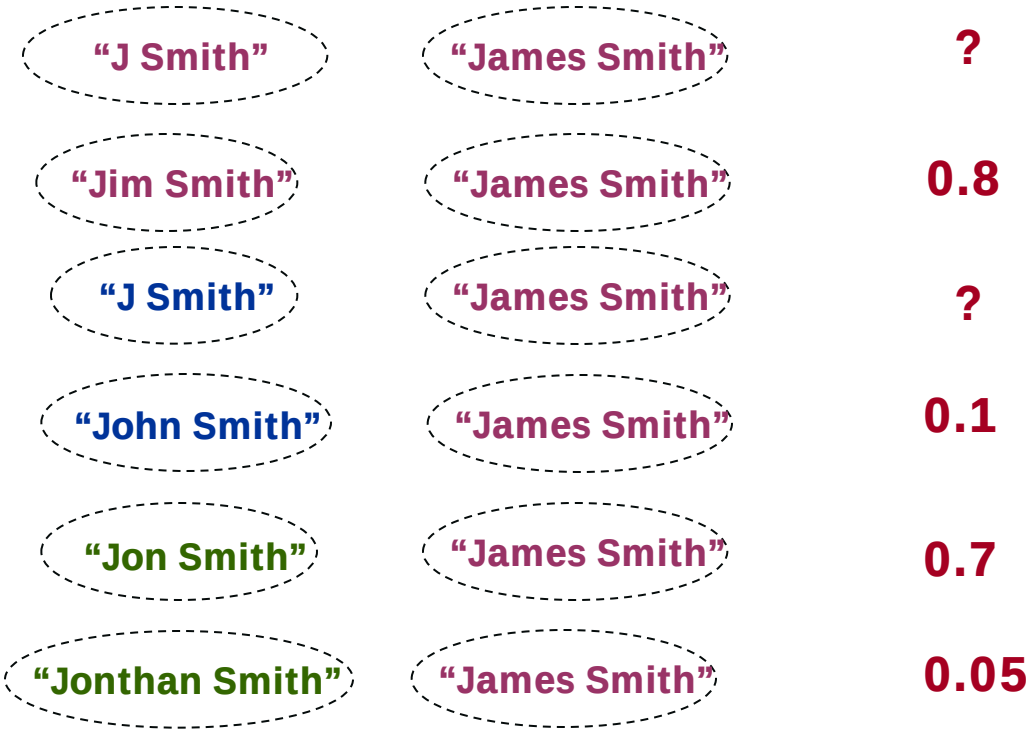
"Jonthan Smith"

Issues:

1. Identification
2. Disambiguation

Attribute-based Entity Resolution

Pair-wise classification



1. Choosing threshold: precision/recall tradeoff
2. Inability to disambiguate
3. Perform transitive closure?

Entity Resolution

- The Problem
- **Relational Entity Resolution**
- Algorithms

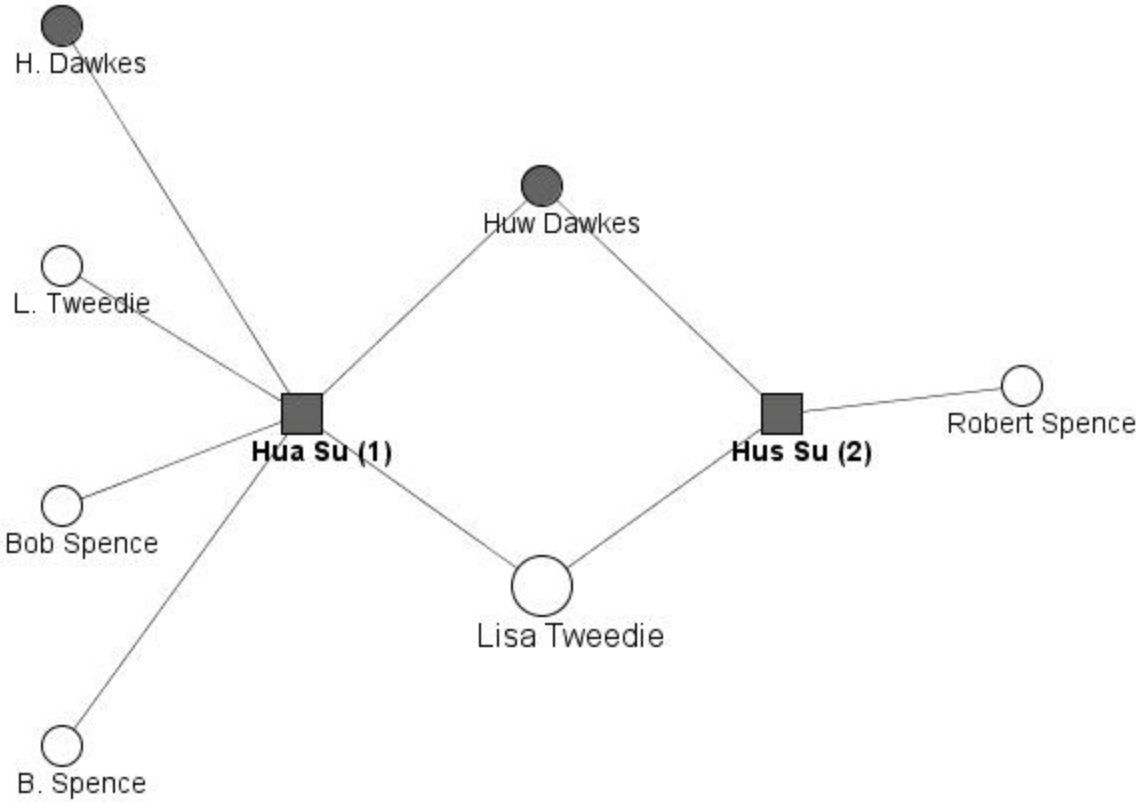
Relational Entity Resolution

- References not observed independently
  - Links between references indicate relations between the entities
  - Co-author relations for bibliographic data
  - To, cc: lists for email

- Use relations to improve identification and disambiguation

Pasula et al. 03, Ananthakrishna et al. 02, Bhattacharya & Getoor 04,06,07, McCallum & Wellner 04, Li, Morie & Roth 05, Culotta & McCallum 05, Kalashnikov et al. 05, Chen, Li, & Doan 05, Singla & Domingos 05, Dong et al. 05

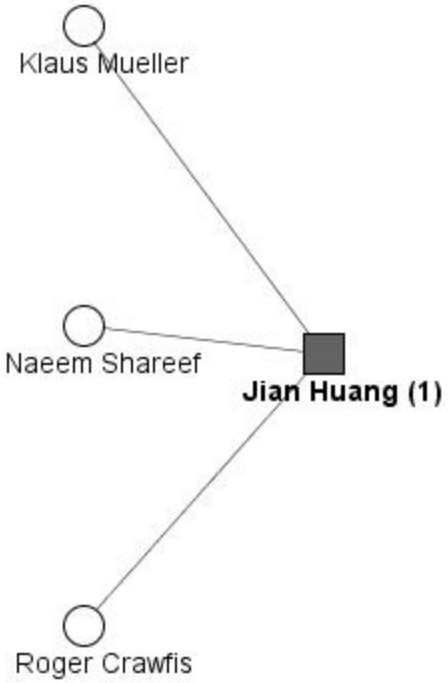
Relational Identification

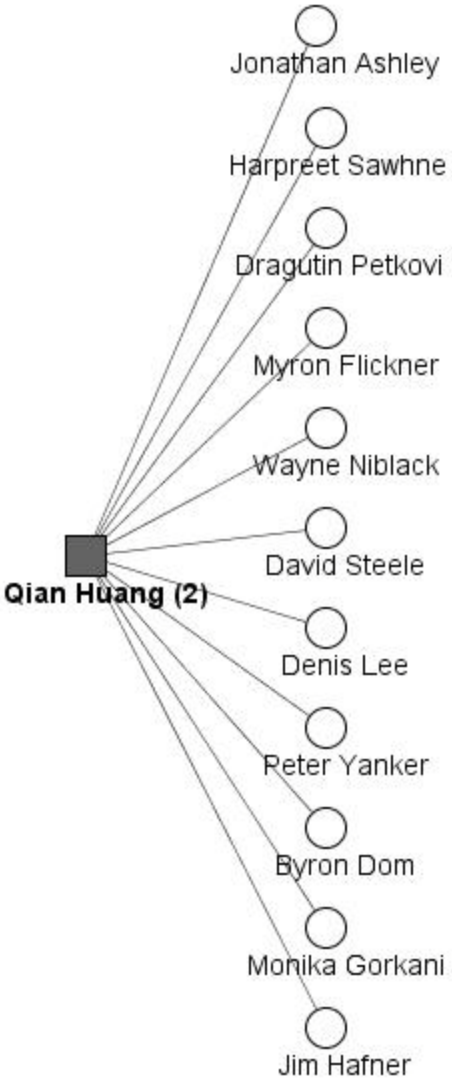


Very similar names.

Added evidence from shared co-authors

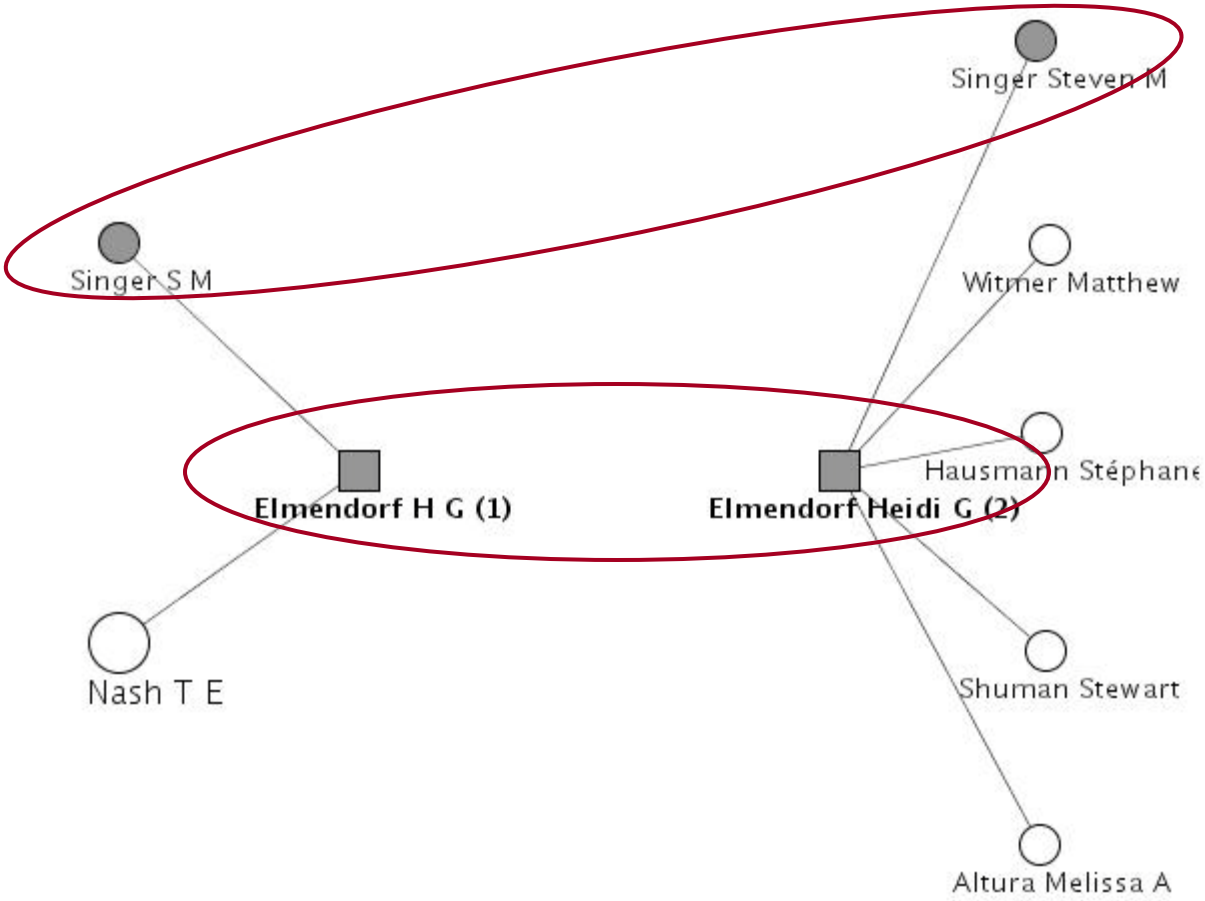
Relational Disambiguation





Very similar names but no shared collaborators

Collective Entity Resolution



One resolution provides evidence for another => joint resolution

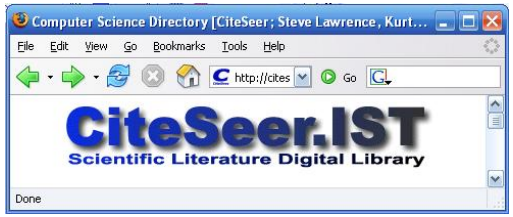
Entity Resolution with Relations

- Naïve Relational Entity Resolution
  - Also compare attributes of related references
  - Two references have co-authors w/ similar names

- **Collective Entity Resolution**
  - Use **discovered entities** of related references
  - Entities cannot be identified independently
  - Harder problem to solve

Entity Resolution

- The Problem
- Relational Entity Resolution
- **Algorithms**
  - **Relational Clustering (RC-ER)**
    - *Bhattacharya & Getoor, DMKD'04, Wiley'06, DE Bulletin'06,TKDD'07*



P1: **"***JOSTLE: Partitioning of Unstructured Meshes for Massively Parallel Machines"*, C. Walshaw, M. Cross, M. G. Everett, S. Johnson

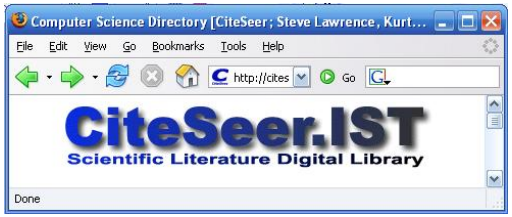
P2: **"***Partitioning Mapping of Unstructured Meshes to Parallel Machine Topologies"*, C. Walshaw, M. Cross, M. G. Everett, S. Johnson, K. McManus

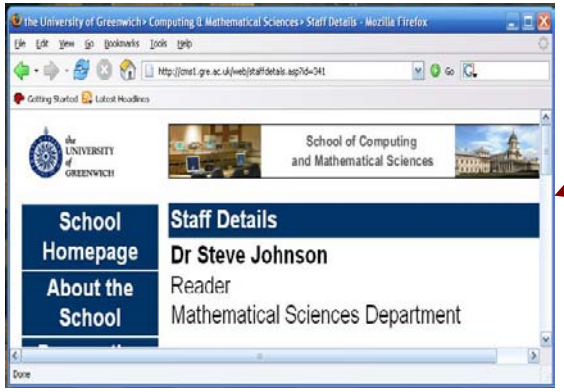
P3: **"***Dynamic Mesh Partitioning: A Unied Optimisation and Load-Balancing Algorithm"*, C. Walshaw, M. Cross, M. G. Everett

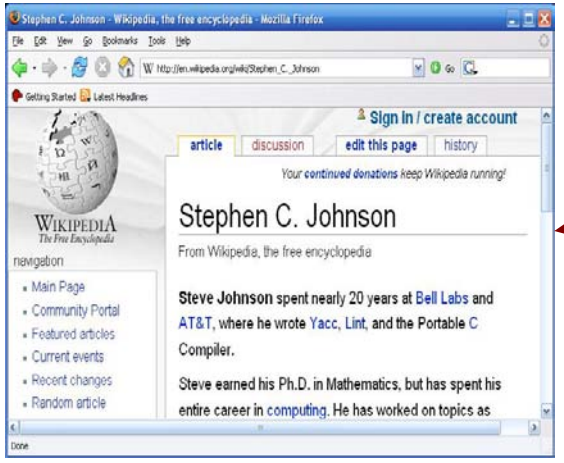
P4: **"***Code Generation for Machines with Multiregister Operations"*, Alfred V. Aho, Stephen C. Johnson, Jefferey D. Ullman

P5: **"***Deterministic Parsing of Ambiguous Grammars"*, A. Aho, S. Johnson, J. Ullman

P6: **"***Compilers: Principles, Techniques, and Tools"*, A. Aho, R. Sethi, J. Ullman







P1: "*JOSTLE: Partitioning of Unstructured Meshes for Massively Parallel Machines*", C. Walshaw, M. Cross, M. G. Everett, **S. Johnson**

P2: "*Partitioning Mapping of Unstructured Meshes to Parallel Machine Topologies*", C. Walshaw, M. Cross, M. G. Everett, **S. Johnson**, K. McManus

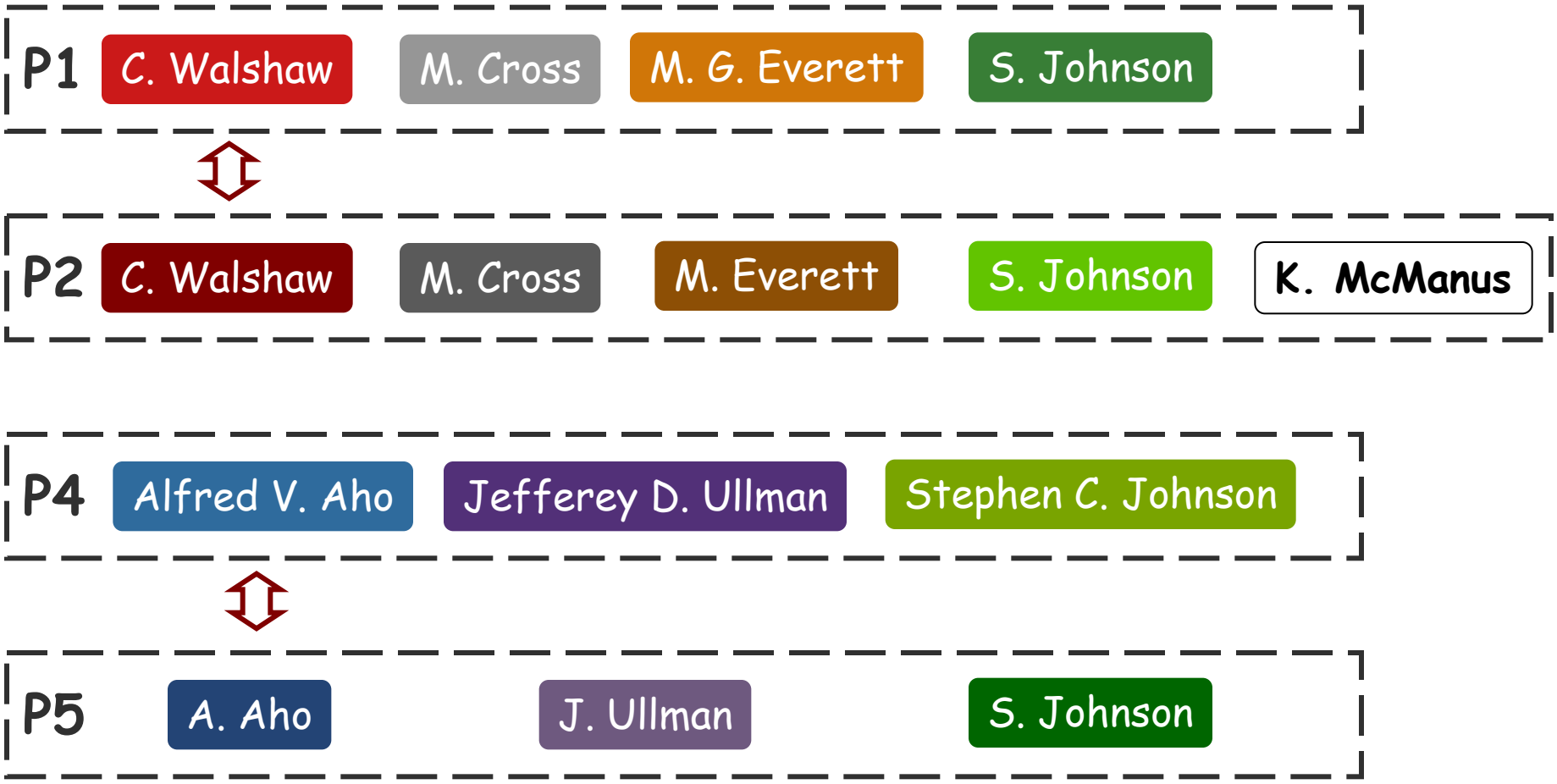
P3: "*Dynamic Mesh Partitioning: A Unied Optimisation and Load-Balancing Algorithm*", C. Walshaw, M. Cross, M. G. Everett

P4: "*Code Generation for Machines with Multiregister Operations*", Alfred V. Aho, **Stephen C. Johnson**, Jefferey D. Ullman

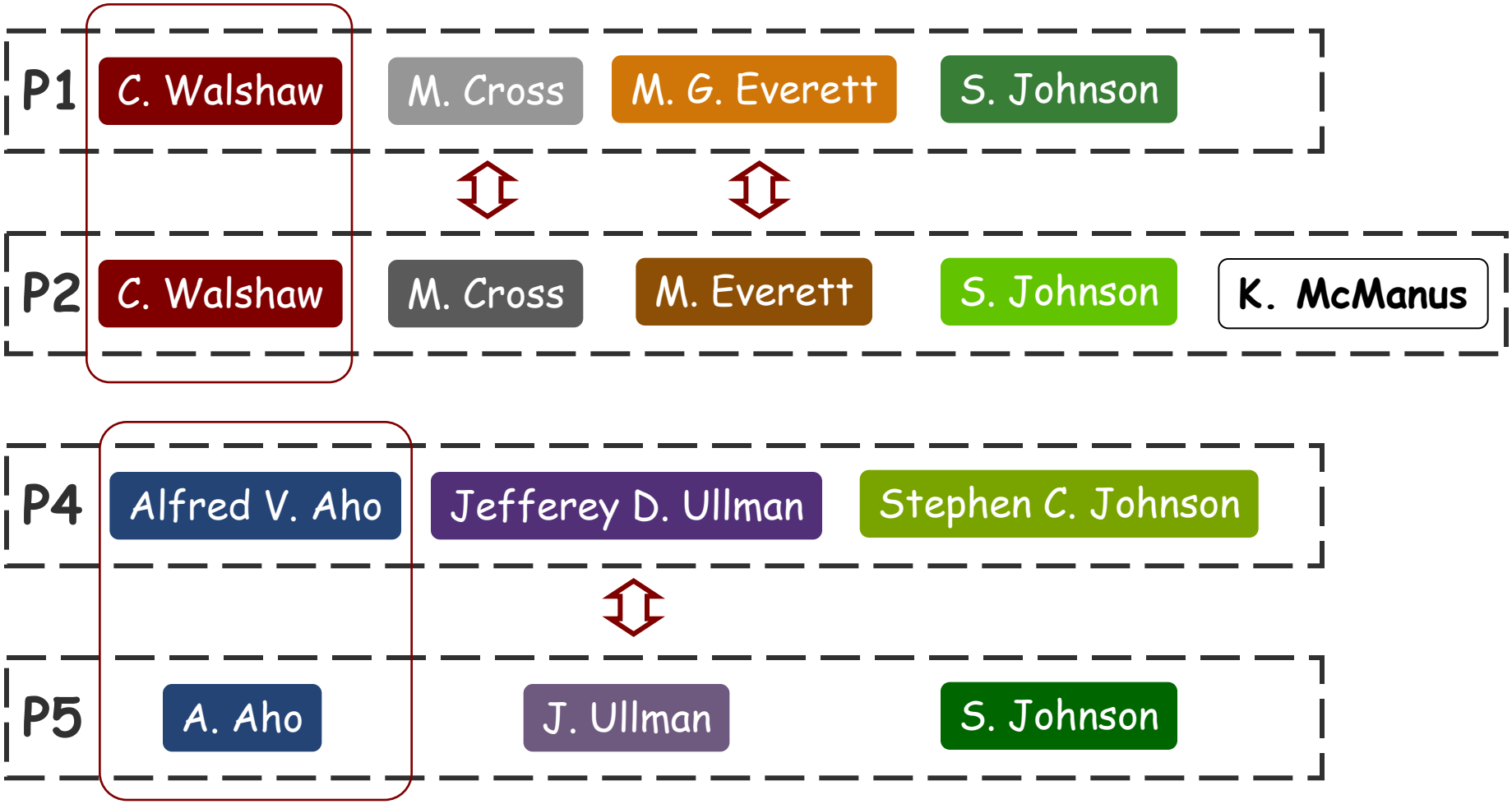
P5: "*Deterministic Parsing of Ambiguous Grammars*", A. Aho, **S. Johnson**, J. Ullman

P6: "*Compilers: Principles, Techniques, and Tools*", A. Aho, R. Sethi, J. Ullman

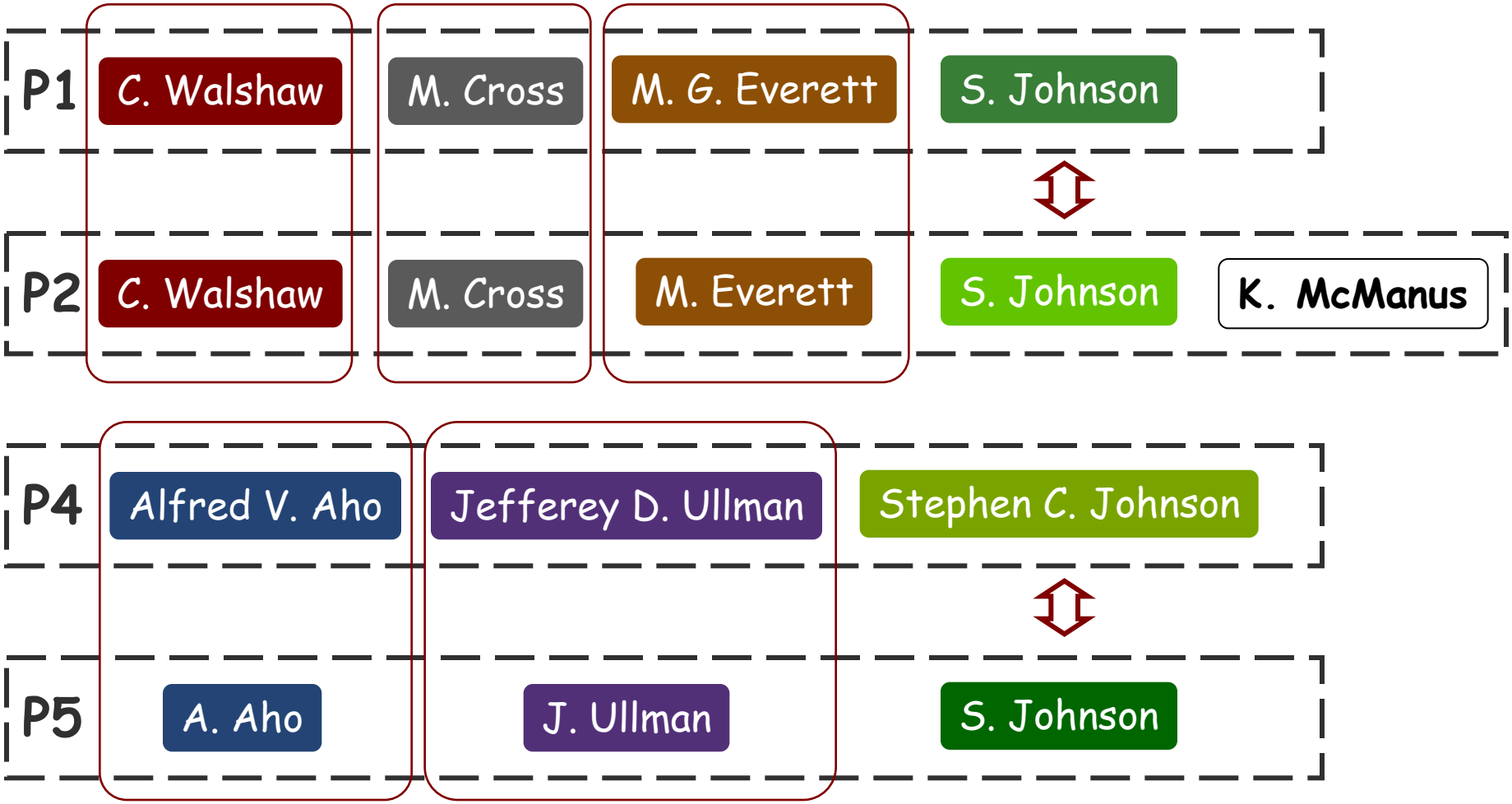
Relational Clustering (RC-ER)



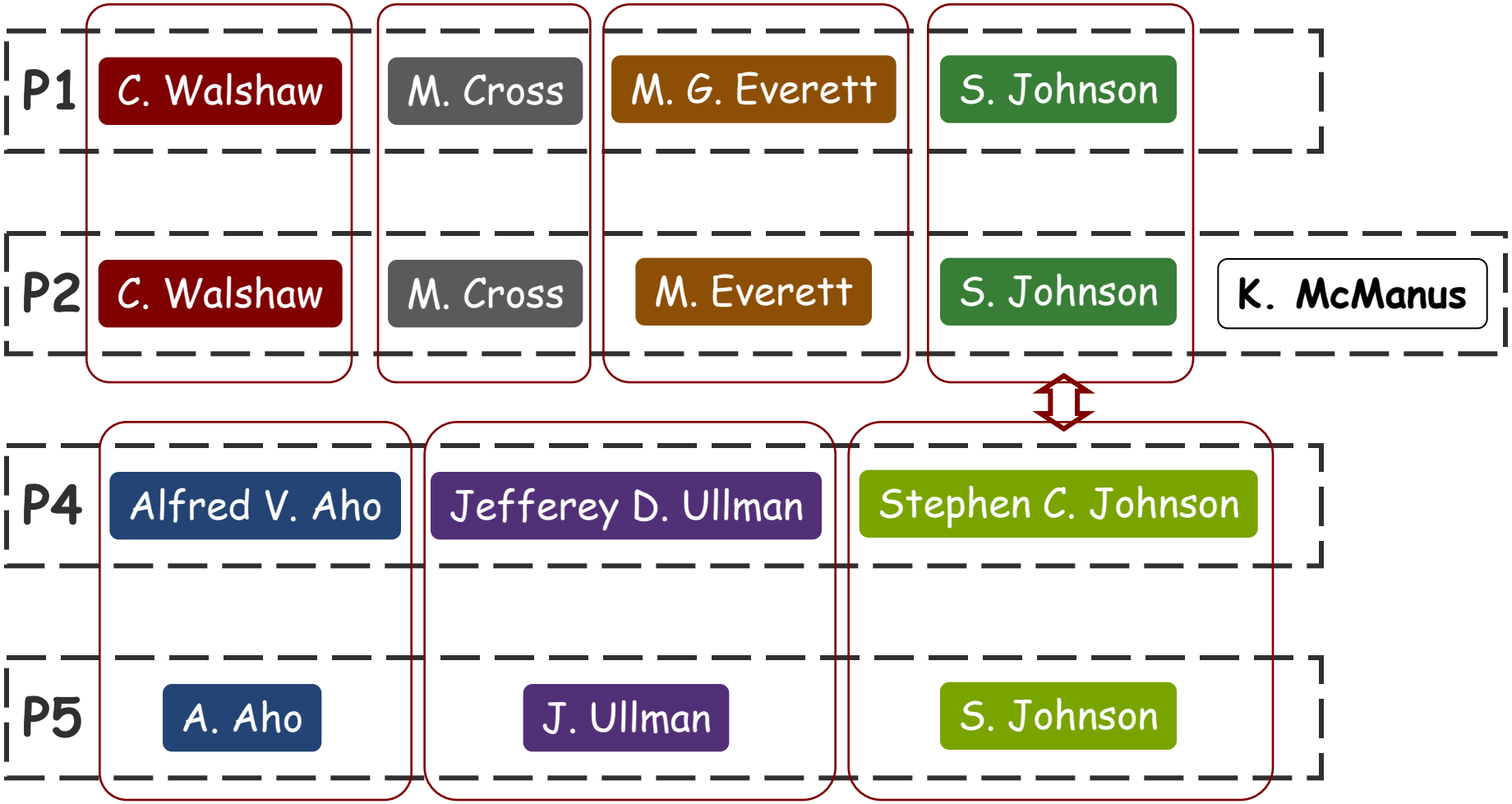
Relational Clustering (RC-ER)



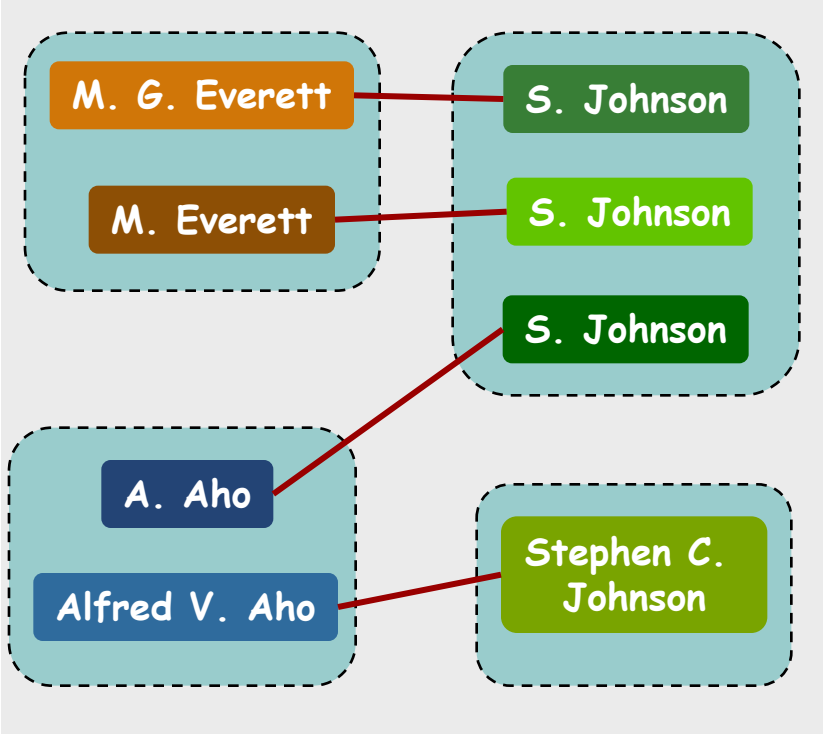
Relational Clustering (RC-ER)



Relational Clustering (RC-ER)

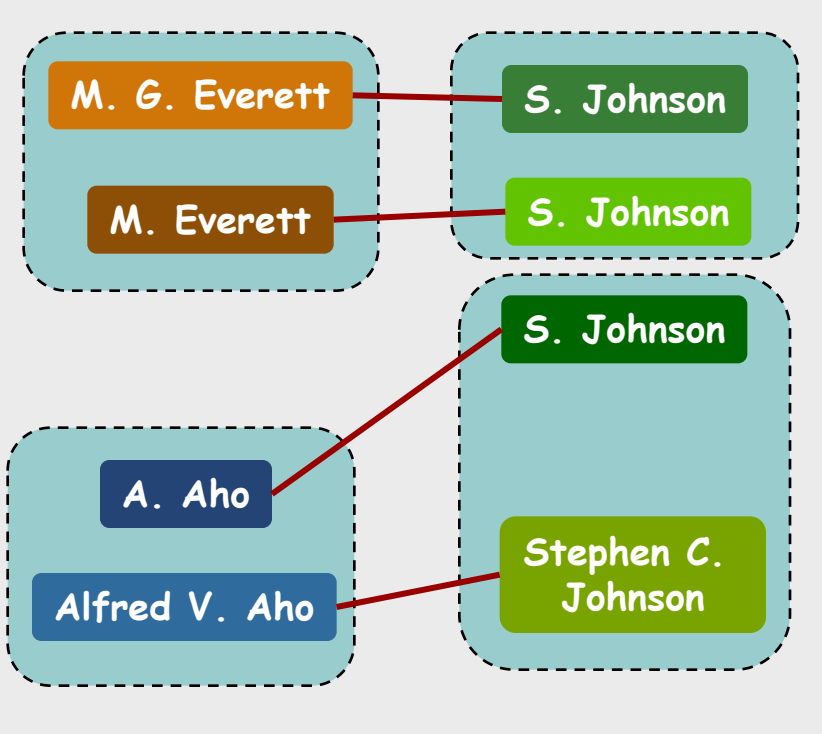


Cut-based Formulation of RC-ER



Good separation of attributes
Many cluster-cluster relationships

- Aho-Johnson1, Aho-Johnson2, Everett-Johnson1



Worse in terms of attributes
Fewer cluster-cluster relationships

- Aho-Johnson1, Everett-Johnson2

Objective Function

- **Minimize:**

$$\sum_i \sum_j w_A sim_A(c_i, c_j) + w_R sim_R(c_i, c_j)$$

weight for attributes — similarity of attributes — weight for relations — Similarity based on relational edges between c_i and c_j

- **Greedy clustering algorithm:** merge cluster pair with max reduction in objective function

$$\Delta(c_i, c_j) = w_A sim_A(c_i, c_j) + w_R(|N(c_i) \bigcap |N(c_j)|)$$

Similarity of attributes — Common cluster neighborhood

Measures for Attribute Similarity

- Use best available measure for each attribute
  - Name Strings: *Soft TF-IDF, Levenstein, Jaro*
  - Textual Attributes: *TF-IDF*

- Aggregate to find similarity between clusters
  - Single link, Average link, Complete link
  - Cluster representative

Comparing Cluster Neighborhoods

- Consider neighborhood as multi-set

- Different measures of set similarity
  - Common Neighbors: Intersection size
  - Jaccard's Coefficient: Normalize by union size
  - Adar Coefficient: Weighted set similarity
  - Higher order similarity: Consider neighbors of neighbors

Relational Clustering Algorithm

1. Find similar references using ‘blocking’
2. Bootstrap clusters using attributes and relations
3. Compute similarities for cluster pairs and insert into priority queue

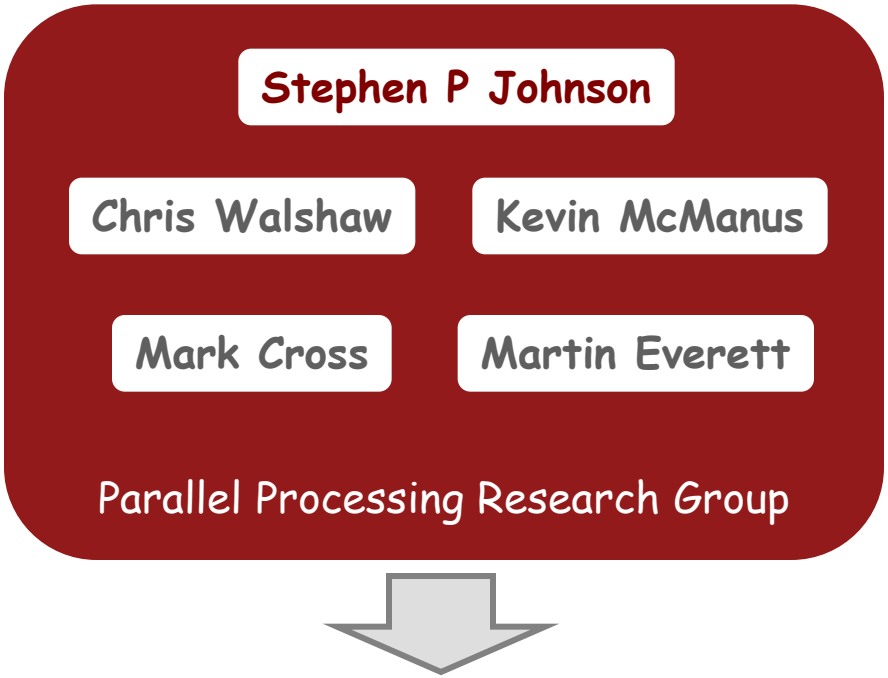
4. Repeat until priority queue is empty
5. Find ‘closest’ cluster pair
6. Stop if similarity below threshold
7. Merge to create new cluster
8. Update similarity for ‘related’ clusters

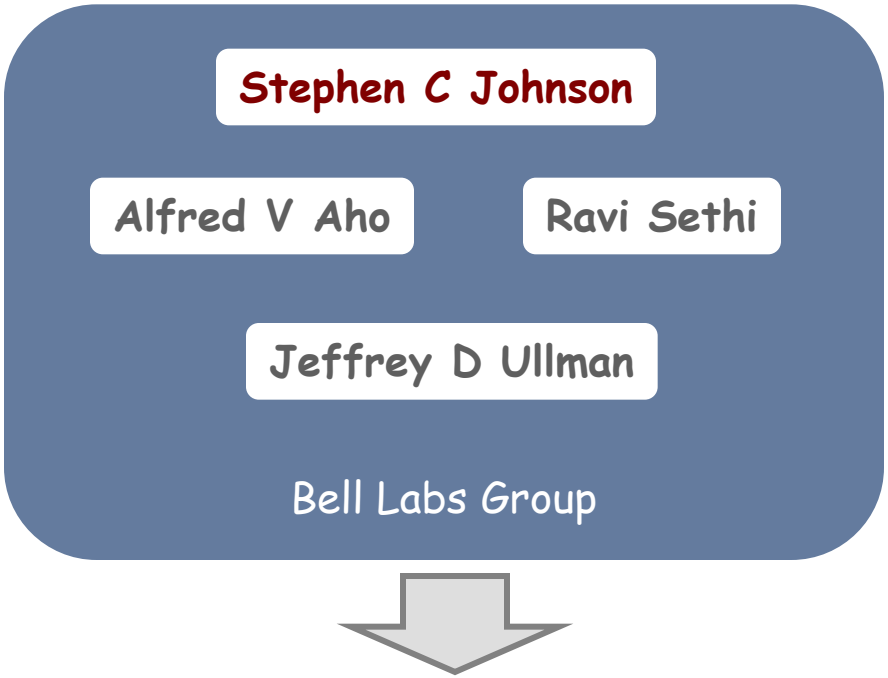
- O(n k log n) algorithm w/ efficient implementation

Entity Resolution

- The Problem
- Relational Entity Resolution
- **Algorithms**
  - Relational Clustering (RC-ER)
  - **Probabilistic Model (LDA-ER)**
    - *SIAM SDM'06, Best Paper Award*
  - Experimental Evaluation

Discovering Groups from Relations





P1: C. Walshaw, M. Cross, M. G. Everett, **S. Johnson**

P2: C. Walshaw, M. Cross, M. G. Everett, **S. Johnson**, K. McManus

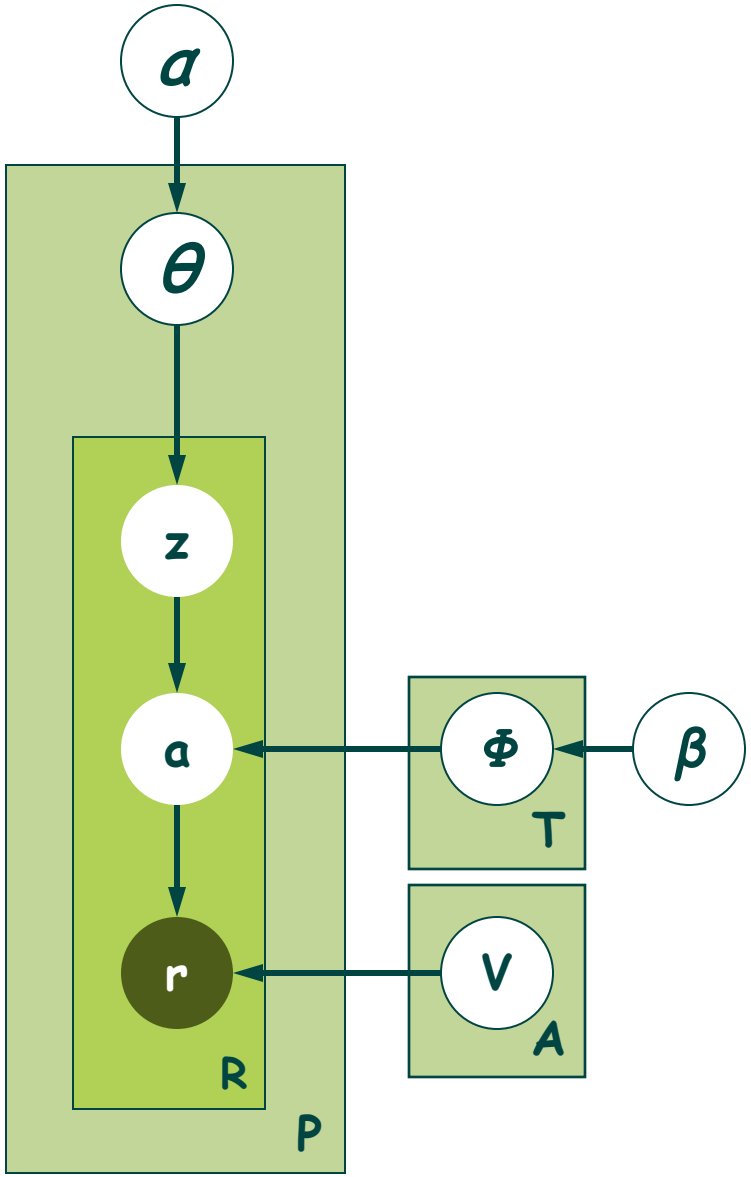
P3: C. Walshaw, M. Cross, M. G. Everett

P4: Alfred V. Aho, **Stephen C. Johnson**, Jefferey D. Ullman

P5: A. Aho, **S. Johnson**, J. Ullman

P6: A. Aho, R. Sethi, J. Ullman

Latent Dirichlet Allocation ER



- Entity label ***a*** and group label ***z*** for each reference ***r***
- ***Θ*****:** ‘mixture’ of groups for each co-occurrence
- Φ_z**:** multinomial for choosing entity ***a*** for each group ***z***
- V_a**:** multinomial for choosing reference ***r*** from entity ***a***
- Dirichlet priors with α and β

Entity Resolution

- The Problem
- Relational Entity Resolution
- **Algorithms**
  - Relational Clustering (RC-ER)
  - Probabilistic Model (LDA-ER)
  - **Experimental Evaluation**

Evaluation Datasets

- CiteSeer
  - 1,504 citations to machine learning papers (Lawrence et al.)
  - 2,892 references to 1,165 author entities
- arXiv
  - 29,555 publications from High Energy Physics (KDD Cup'03)
  - 58,515 refs to 9,200 authors
- Elsevier BioBase
  - 156,156 Biology papers (IBM KDD Challenge '05)
  - 831,991 author refs
  - Keywords, topic classifications, language, country and affiliation of corresponding author, etc

Baselines

- **A:** Pair-wise duplicate decisions w/ attributes only
  - **Names:** *Soft-TFIDF* with *Levenstein, Jaro, Jaro-Winkler*
  - **Other textual attributes:** *TF-IDF*
- **A*:** Transitive closure over **A**

- **A+N**: Add attribute similarity of co-occurring refs
- **A+N***: Transitive closure over **A+N**

- Evaluate pair-wise decisions over references
- F1-measure (harmonic mean of precision and recall)

ER over Entire Dataset

Algorithm	CiteSeer	arXiv	BioBase
A	0.980	0.976	0.568
A*	0.990	0.971	0.559
A+N	0.973	0.938	0.710
A+N*	0.984	0.934	0.753
RC-ER	**0.995**	**0.985**	**0.818**
LDA-ER	0.993	0.981	0.645

- RC-ER & LDA-ER outperform baselines in all datasets
- Collective resolution better than naïve relational resolution
- RC-ER and baselines require threshold as parameter
  - Best achievable performance over all thresholds
- Best RC-ER performance better than LDA-ER
- LDA-ER does not require similarity threshold

***Collective Entity Resolution In Relational Data*, Indrajit Bhattacharya and Lise Getoor, ACM Transactions on Knowledge Discovery and Datamining, 2007**

ER over Entire Dataset

Algorithm	CiteSeer	arXiv	BioBase
A	0.980	0.976	0.568
A*	0.990	0.971	0.559
A+N	0.973	0.938	0.710
A+N*	0.984	0.934	0.753
RC-ER	**0.995**	**0.985**	**0.818**
LDA-ER	0.993	0.981	0.645

- CiteSeer: Near perfect resolution; 22% error reduction
- arXiv: 6,500 additional correct resolutions; 20% error reduction
- BioBase: Biggest improvement over baselines

Roadmap

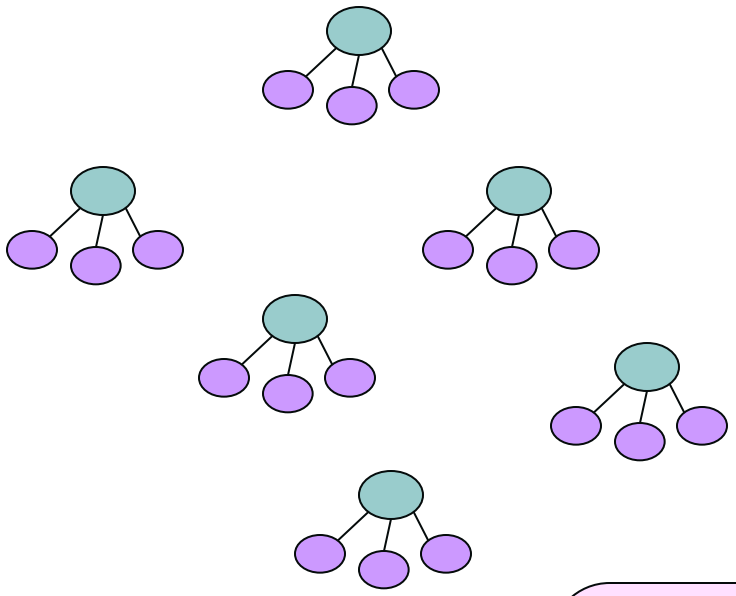
- The Problem
- **The Components**
  - Entity Resolution
  - **Collective Classification**
  - Link Prediction
- Putting It All Together
- Open Questions

Collective Classification

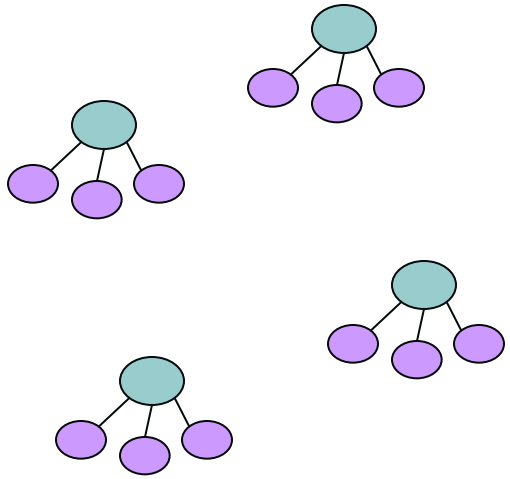
- **The Problem**
- Collective Relational Classification
- Algorithms

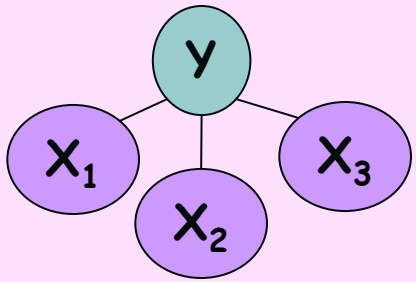
Traditional Classification

Training Data



Test Data

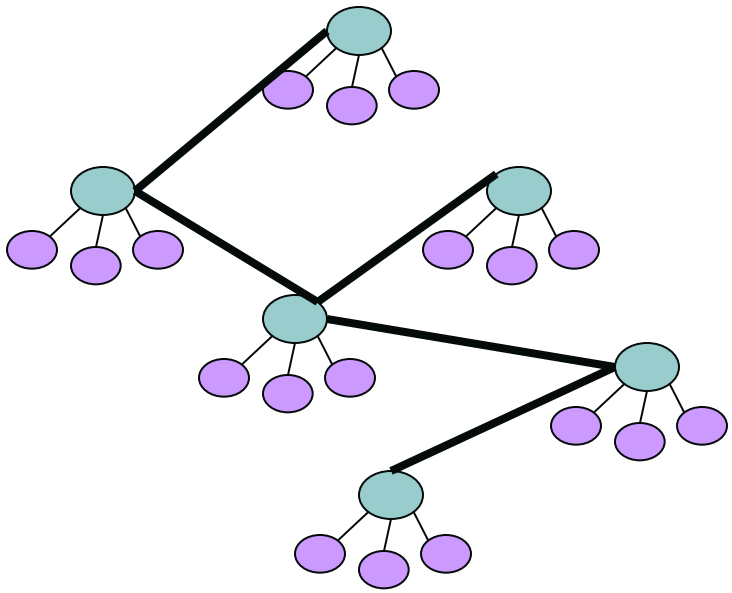




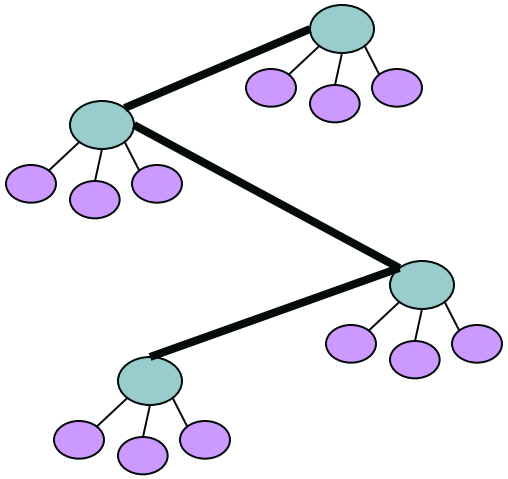
Predict Y based on attributes X_i

Relational Classification (1)

Training Data



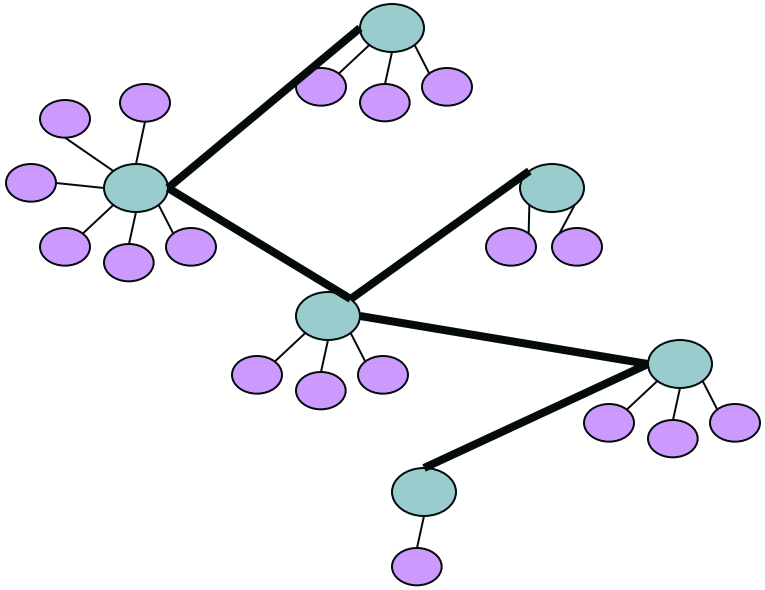
Test Data



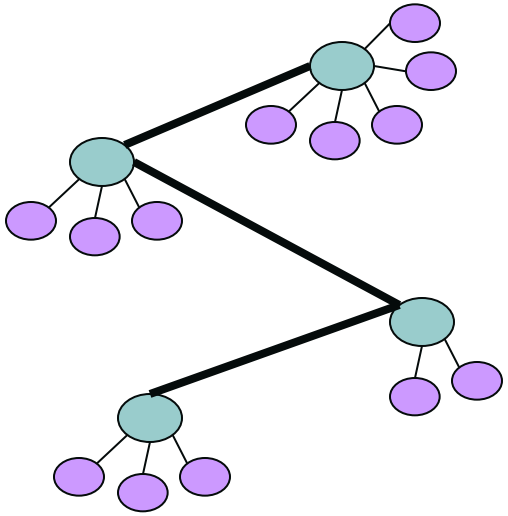
Correlations among linked instances
autocorrelation: labels are likely to be the same
homophily: similar nodes are more likely to be linked

Relational Classification (2)

Training Data

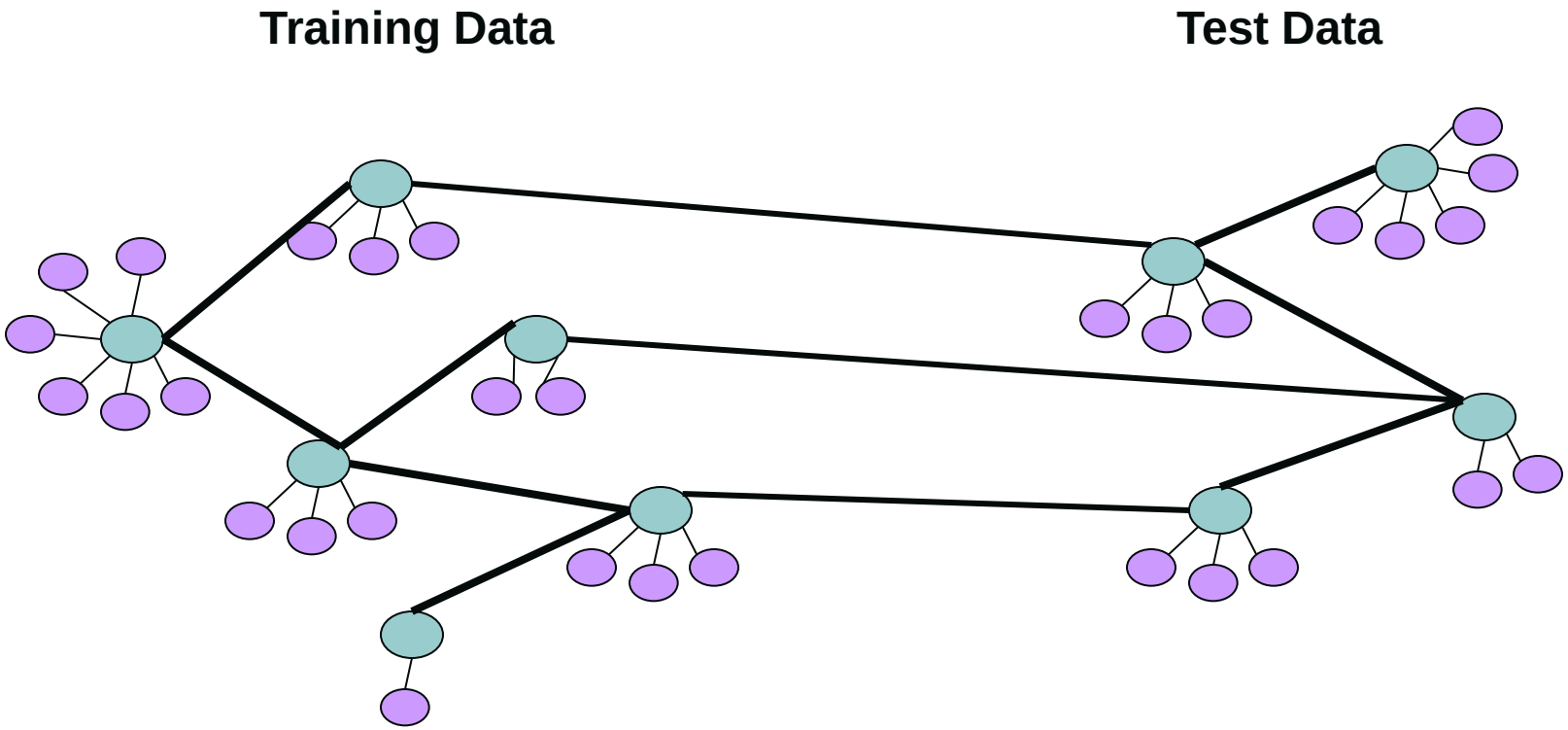


Test Data



Irregular graph structure

Relational Classification (3)



Links between training set & test set
learning with partial labels or within network classification

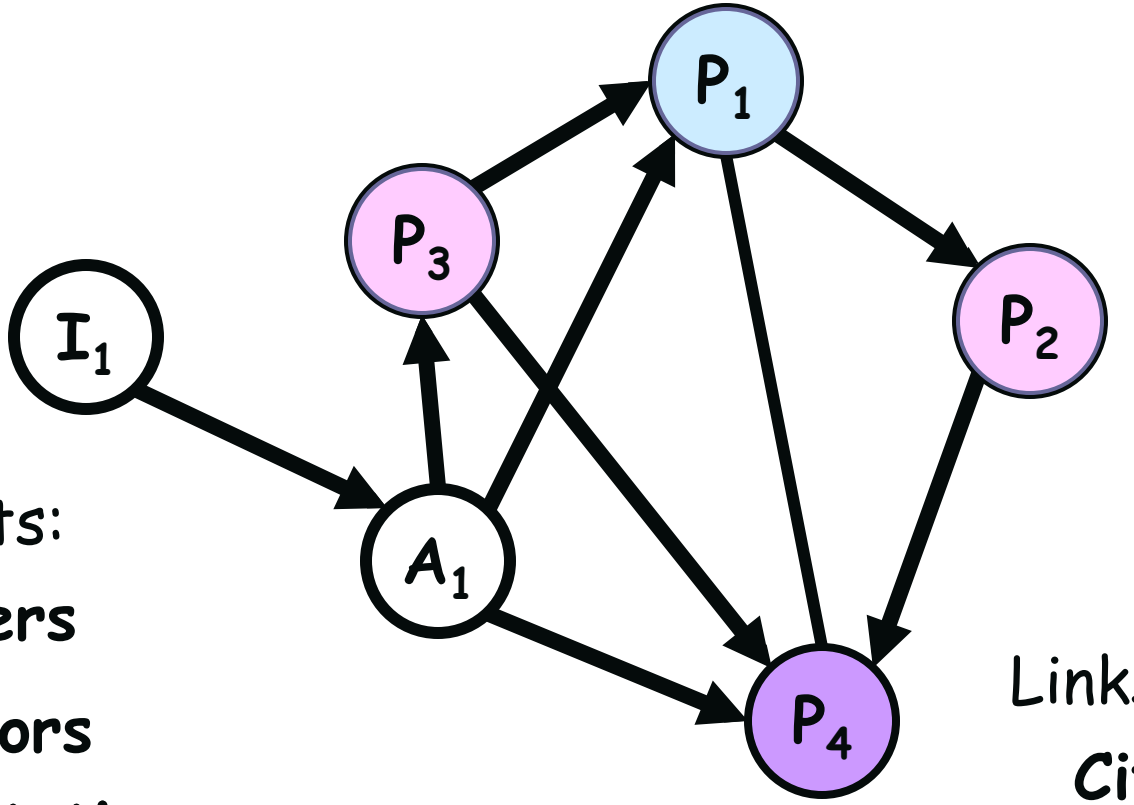
The Problem

- Relational Classification: predicting the category of an object based on its attributes *and* its links *and* attributes of linked objects

- Collective Classification: jointly predicting the categories for a collection of connected, unlabelled objects

Neville & Jensen 00, Taskar , Abbeel & Koller 02, Lu & Getoor 03, Neville, Jensen & Galliger 04, Sen & Getoor TR07, Macskassy & Provost 07, Gupta, Diwam & Sarawagi 07, Macskassy 07, McDowell, Gupta & Aha 07

Example: Linked Bibliographic Data



Objects:

- **Papers**
- **Authors**
- **Institutions**

Links:

- **Citation**
- **Co-Citation**
- **Author-of**
- **Author-affiliation**

Labels:

Feature Construction

- Objects are linked to a **set** of objects. To construct features from this set of objects, we need feature aggregation methods

Kramer, Lavrac & Flach 01, Perlich & Provost 03, 04, 05, Popescul & Ungar 03, 05, 06, Lu & Getoor 03, Gupta, Diwam & Sarawagi 07

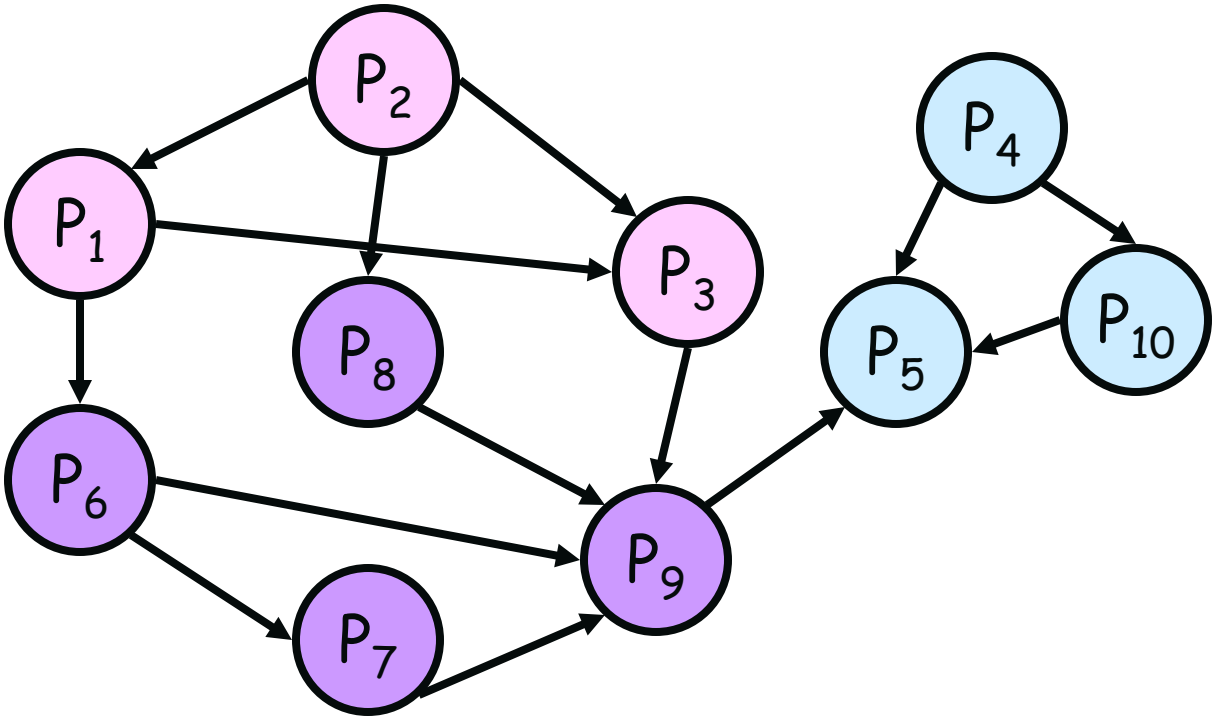
Formulation

- Local Models
  - Collection of Local Conditional Models
  - Inference Algorithms:
    - Iterative Classification Algorithm (ICA)
    - Gibbs Sampling (Gibbs)

- Global Models
  - (Pairwise) Markov Random Fields
  - Inference Algorithms:
    - Loopy Belief Propagation (LBP)
    - Mean Field Relaxation Labeling (MF)

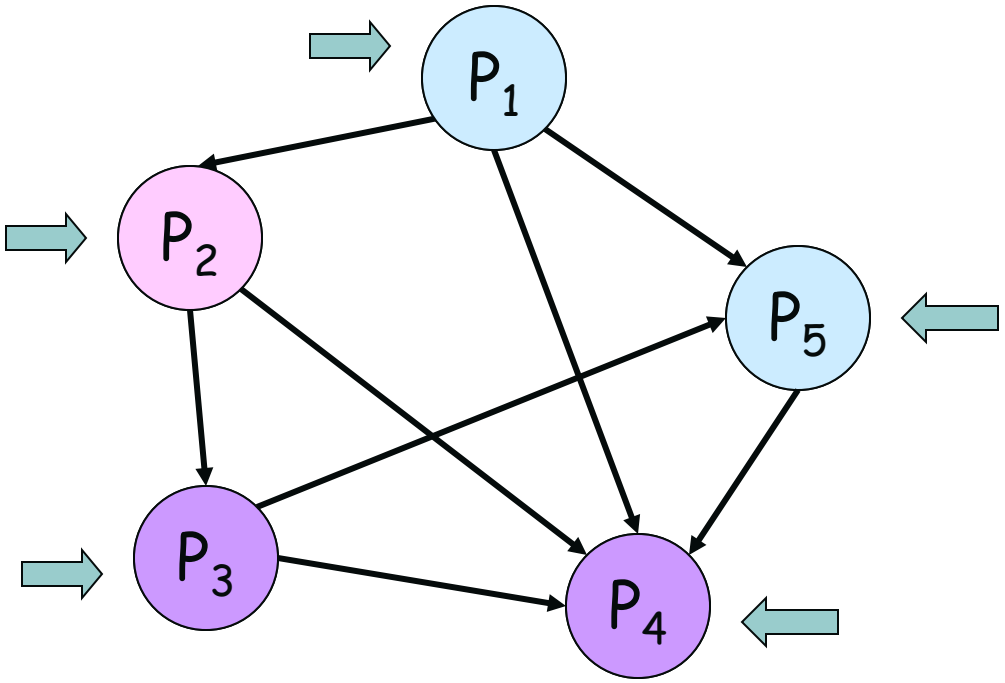
ICA: Learning

- label set:



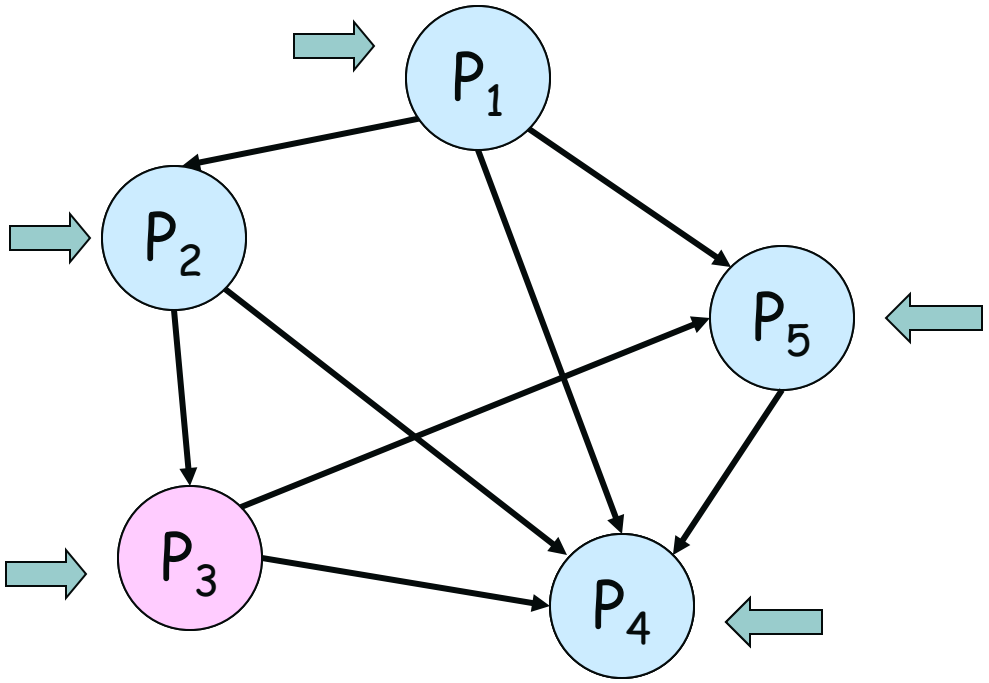
Learn model from fully labeled training set

ICA: Inference (1)



Step 1: Bootstrap using object attributes only

ICA: Inference (2)



Step 2: Iteratively update the category of each object, based on linked object's categories

Experimental Evaluation

- Comparison of Collective Classification Algorithms
  - Mean Field Relaxation Labeling (MF)
  - Iterative Classification Algorithm (ICA)
  - Gibbs Sampling (Gibbs)
  - Loopy Belief Propagation (LBP)
  - Baseline: Content Only

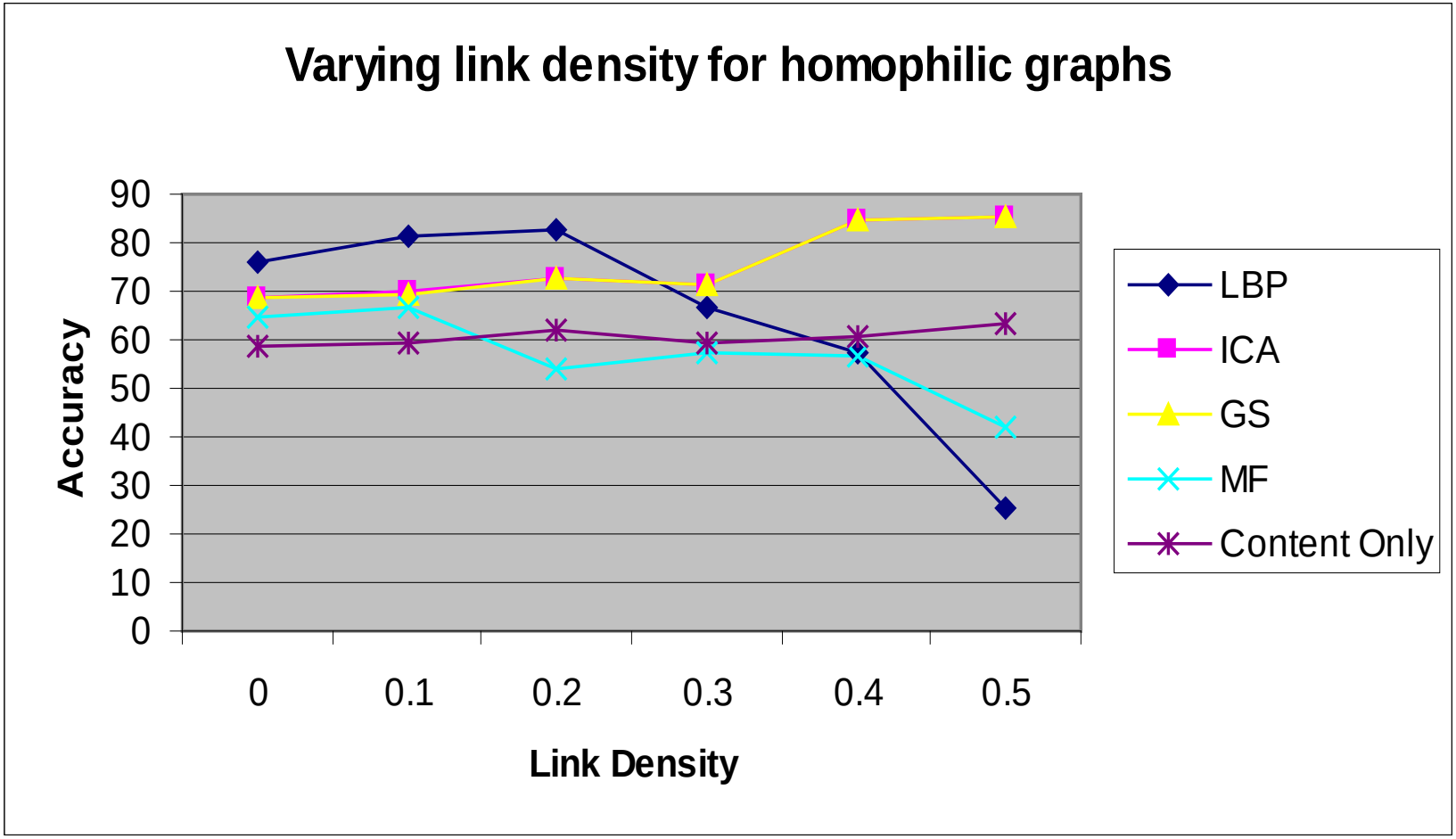
- Datasets
  - Real Data
    - Bibliographic Data (Cora & Citeseer), WebKB, etc.
  - Synthetic Data
    - Data generator which can vary the class label correlations (homophily), attribute noise, and link density

Results on Real Data

Algorithm	Cora	CiteSeer	WebKB
Content Only	66.51	59.77	62.49
ICA	74.99	62.46	**65.99**
Gibbs	74.64	62.52	65.64
MF	79.70	**62.91**	65.65
LBP	**82.48**	62.64	65.13

Sen, Namata, Bilgic, Getoor, Gallagher, Eliassi-Rad, AI Magazine 07

Effect of Structure



Results clearly indicate that algorithms' performance depends (in non-trivial ways) on structure

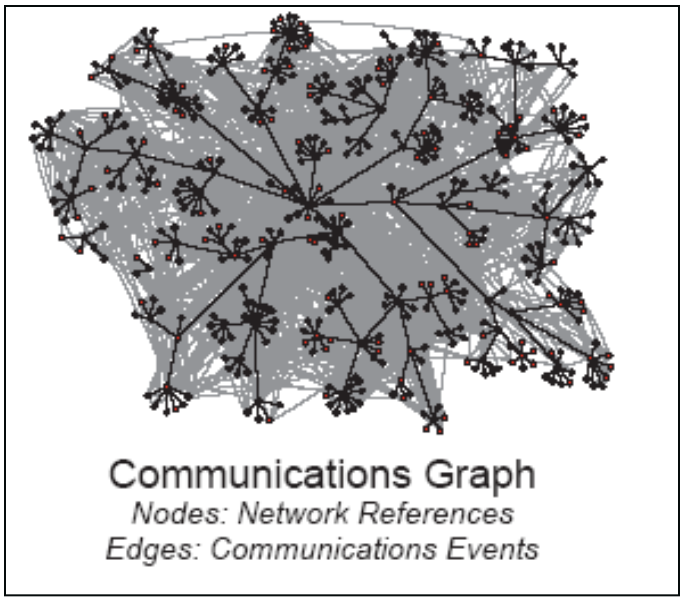
Roadmap

- The Problem
- **The Components**
  - Entity Resolution
  - Collective Classification
  - **Link Prediction**
- Putting It All Together
- Open Questions

Link Prediction

- **The Problem**
- Predicting Relations
- Algorithms
  - Link Labeling
  - Link Ranking
  - Link Existence

Links in Data Graph

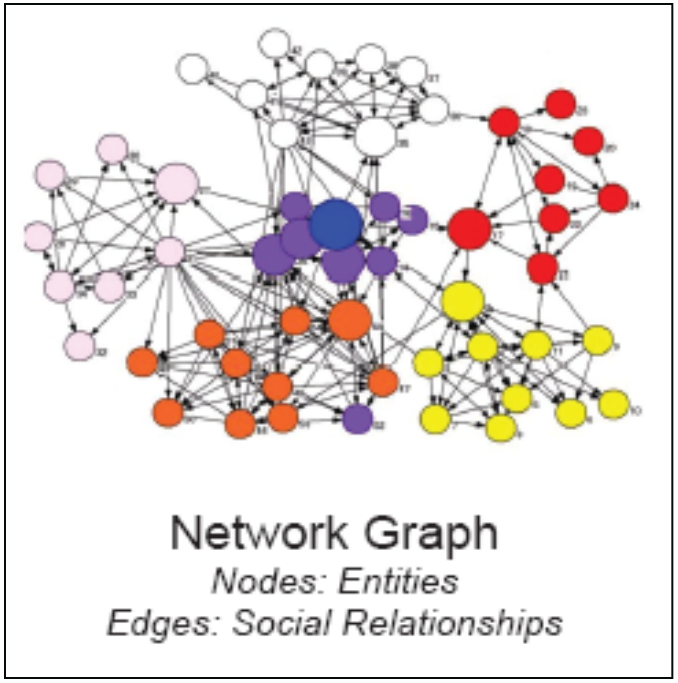


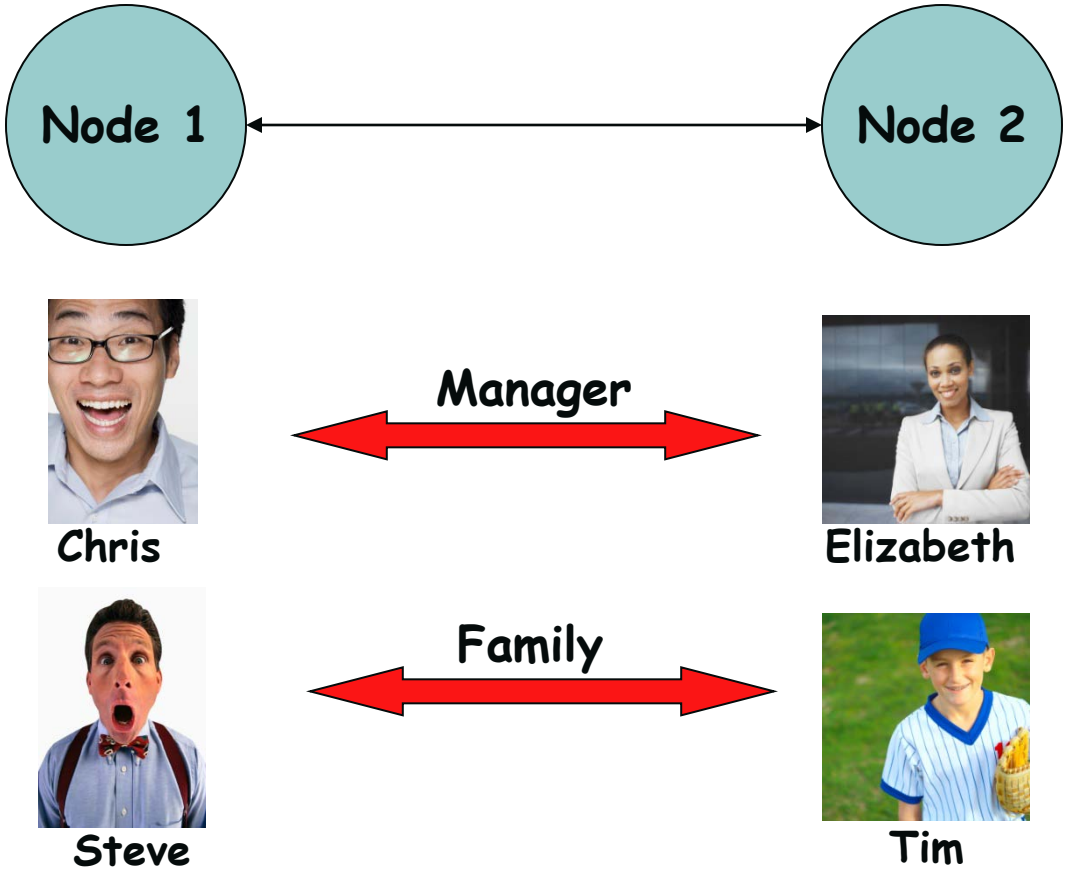
chris@enron.com ⟷ Email ⟷ liz@enron.com

chris37 ⟷ IM ⟷ lizs22

555-450-0981 ⟷ TXT ⟷ 555-901-8812

⇒ Links in Information Graph





Roadmap

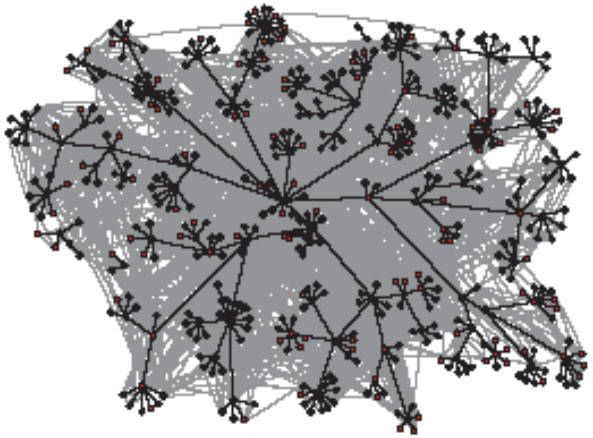
- The Problem
- The Components
- **Putting It All Together**
- Open Questions

Putting Everything together….

Collaborative Social Network Discovery

Entity Resolution

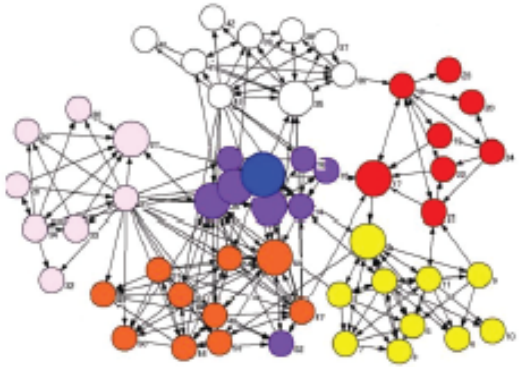
Relationship Identification



Communications Graph

Nodes: Network References

Edges: Communications Events



Network Graph

Nodes: Entities

Edges: Social Relationships

Graph Identification

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

Dependencies

- Intra-dependent
  - Two nodes more likely to be co-referent if their neighbors are co-referent
  - Two nodes are more likely to be linked if they link to common nodes
  - Label of a node depends on the labels of related nodes
- Inter-dependent
  - Two nodes are more likely to be co-referent if they have the same *inferred* label
  - Two nodes are more likely to be linked depending on their *inferred* labels
  - Label of a node depends on *inferred* linked nodes

Classifiers

- Base Classifiers
  - Can use any conditional model as base classifier (i.e., logistic regression, decision trees, SVMs, naïve Bayes, etc.)
  - Local Classifiers – use only local attribute info for a node or edge
  - Relational Classifiers – can use info from relational neighborhood

Classifiers

- Base Classifiers
  - Can use any conditional model as base classifier (i.e., logistic regression, decision trees, SVMs, naïve Bayes, etc.)
  - Local Classifiers – use only local attribute info for a node or edge
  - Relational Classifiers – can use info from relational neighborhood
- **Collective classifiers**
  - **Use local classifiers to bootstrap classification process**
  - **Iteratively apply relational classifiers**

Classifiers

- Base Classifiers
  - Can use any conditional model as base classifier (i.e., logistic regression, decision trees, SVMs, naïve Bayes, etc.)
  - Local Classifiers – use only local attribute info for a node or edge
  - Relational Classifiers – can use info from relational neighborhood
- Collective classifiers
  - Use local classifiers to bootstrap classification process
  - Iteratively apply relational classifiers
- **Coupled Classifiers**
  - **Apply the collective classifiers in order such that collective classifiers can use the predictions of earlier classifiers when computing relational features**
    - **Pipeline – Apply the components one at a time, in a particular sequence**
    - **Coupled Collective Classifiers – Apply components iteratively**

Coupled Collective Classification (C^3) Algorithm

- Focus is on coupling the inference of the three components using conditional models
- Conditional models applied in two phases
  - Phase 1: Local models using only local features
    - Bootstraps the process
  - Phase 2: Relational models using intra- and inter-relational features
    - Infer assignments using local and intra- and inter-relational information
- Cyclic dependencies handled by iteratively apply relational models

C^3 Variants

- Capture more dependencies can also mean introducing more channels for error propagation
- Variant 1: Confidence-Based Inference
  - Some predictions are more confident than others
  - Commit more confident predictions earlier
- Variant 2: Stacked Learning (Kou & Cohen 07)
  - Instead of using the true assignments for relational features during training, use *inferred* assignments

Experimental Evaluation

- Datasets:
  - Citation Networks
    - Citeseer – 3312 paper nodes, 4732 citation edges, 6 possible labels
    - Cora – 2708 paper nodes, 5428 citation edges, 7 possible labels
  - Partitioned to three disjoint networks and created noisy versions of each; varied amount of noise (Low, Medium, High)
  - Given noisy network, infer the original network
- Conditional models: linear SVM
- Evaluate average F1 performance over ER, LP, CC

Algorithms

- Baselines:
  - LOCAL: apply only the local models
  - INTRA: apply relational classifiers using only intra-relational features
- PIPELINE: apply collective classifiers for each component in the pipeline
- C^3 Variants:
  - C^3: the basic algorithm
  - C^3+C: C^3 using confidence based inference
  - C^3+S: C^3 using stacking
  - C^3+SC: C^3 using stacking and confidence based inference
- Gibbs: apply pseudo-Gibbs sampling over the conditional models

General Trends: Citeseer

	Low Noise				Medium Noise				High Noise			
	ER (f1)	LP (f1)	NL (f1)	Avg.	ER (f1)	LP (f1)	NL (f1)	Avg.	ER (f1)	LP (f1)	NL (f1)	Avg.
LOCAL	**0.999**	0.853	0.656	0.836	0.993	0.707	0.633	0.778	0.954	0.650	0.602	0.735
INTRA	**0.999**	0.852	0.660	0.837	0.995	0.706	0.639	0.780	0.956	0.647	0.621	0.741
ELN	**0.999**	0.906	0.684	0.863	0.995	0.851	0.675	0.840	0.956	0.780	0.634	0.790
ENL	**0.999**	0.916	0.679	0.865	0.995	0.872	0.665	0.844	0.956	0.808	0.633	0.799
LEN	**0.999**	0.852	0.678	0.843	0.994	0.706	0.666	0.789	0.953	0.647	0.625	0.742
LNE	**0.999**	0.852	0.663	0.838	0.994	0.706	0.643	0.781	0.953	0.647	0.608	0.736
NEL	**0.999**	0.916	0.660	0.858	0.993	0.872	0.639	0.835	0.959	**0.812**	0.621	0.797
NLE	**0.999**	0.863	0.660	0.840	0.993	0.754	0.639	0.795	0.955	0.694	0.621	0.757
Gibbs	**0.999**	**0.924**	0.676	0.866	0.942	**0.891**	0.666	0.833	0.613	0.840	0.621	0.691
C³	**0.999**	0.917	0.683	0.866	0.995	0.870	0.670	0.845	0.959	0.809	0.638	**0.802**
C³+C	**0.999**	0.917	0.684	0.867	0.995	0.872	0.667	0.845	0.957	0.810	0.634	0.800
C³+S	**0.999**	0.917	0.700	0.872	**0.996**	0.868	**0.684**	**0.849**	**0.965**	0.775	0.651	0.797
C³+SC	**0.999**	0.918	**0.701**	**0.873**	0.995	0.869	0.681	0.848	0.962	0.773	**0.654**	0.797

- Capturing more dependencies result in improved performance
- C³ algorithm generally the best **performing for each task and overall**

General Trends: Cora

	Low Noise				Medium Noise				High Noise			
	ER (f1)	LP (f1)	NL (f1)	Avg.	ER (f1)	LP (f1)	NL (f1)	Avg.	ER (f1)	LP (f1)	NL (f1)	Avg.
LOCAL	0.983	0.816	0.719	0.839	0.950	0.702	0.682	0.778	0.910	0.483	0.613	0.669
INTRA	0.975	0.812	0.735	0.841	0.938	0.694	0.694	0.775	0.886	0.470	0.657	0.671
ELN	0.975	0.906	0.774	0.885	0.938	0.867	0.722	0.842	0.886	0.762	0.657	0.768
ENL	0.975	0.918	0.765	0.886	0.938	0.882	0.728	0.849	0.886	0.774	0.663	0.774
LEN	0.972	0.812	0.764	0.849	0.932	0.694	0.711	0.779	0.892	0.470	0.632	0.665
LNE	0.974	0.812	0.739	0.842	0.937	0.694	0.674	0.768	0.895	0.470	0.610	0.659
NEL	0.977	0.916	0.735	0.876	0.943	0.881	0.694	0.839	0.897	0.806	0.657	0.787
NLE	0.975	0.837	0.735	0.849	0.942	0.769	0.694	0.802	0.894	0.628	0.657	0.726
Gibbs	0.943	0.932	0.772	0.882	0.742	**0.895**	0.690	0.776	0.365	**0.835**	0.620	0.607
C³	0.977	**0.919**	0.767	0.888	0.943	0.880	0.724	0.849	0.892	0.792	0.663	0.782
C³+C	0.976	0.918	0.772	0.889	0.943	0.882	0.716	0.847	0.894	0.797	0.660	0.784
C³+S	**0.984**	0.915	**0.790**	**0.896**	**0.961**	0.882	**0.767**	**0.870**	**0.921**	0.809	**0.684**	**0.804**
C³+SC	0.983	0.916	0.786	0.895	0.962	0.880	0.759	0.867	0.919	0.802	0.682	0.801

- Capturing more dependencies result in improved performance
- C³ algorithm generally the best **performing for each task and overall**

Improvements are Significant

Citeseer

	LOCAL	INTRA	ELN	ENL	LEN	LNE	NEL	NLE	Gibbs	C^3	C^3+C	C^3+S	C^3+SC
LOCAL	--	0	0	0	0	0	0	0	1	0	0	0	0
INTRA	4	--	0	0	0	0	0	0	1	0	0	0	0
ELN	8	5	--	1	2	4	3	3	2	0	0	0	0
ENL	7	5	1	--	4	4	2	2	2	0	0	0	0
LEN	3	2	0	0	--	1	1	1	1	0	0	0	0
LNE	0	0	0	0	0	--	0	0	1	0	0	0	0
NEL	5	4	2	0	3	4	--	0	2	0	0	0	0
NLE	2	2	0	0	1	1	0	--	1	0	0	0	0
Gibbs	4	4	1	0	3	3	1	3	--	0	0	1	0
C^3	7	5	2	0	4	4	4	3	3	--	1	0	0
C^3+C	5	6	2	0	5	4	2	3	3	0	--	0	0
C^3+S	8	7	4	3	6	6	5	7	6	3	2	--	0
C^3+SC	7	7	2	3	6	7	6	7	4	2	1	0	--

- Performed paired t-test (> 95%) between all algorithms pairs
- C^3 significantly outperforms other models in most cases

Improvements are Significant

Cora

	LOCAL	INTRA	ELN	ENL	LEN	LNE	NEL	NLE	Gibbs	C^3	C^3+C	C^3+S	C^3+SC
LOCAL	--	1	0	0	1	1	0	0	1	0	0	0	0
INTRA	1	--	0	0	0	2	0	0	1	0	0	0	0
ELN	5	4	--	0	4	7	0	1	3	0	0	0	0
ENL	8	8	0	--	6	7	4	3	3	0	0	0	1
LEN	3	0	0	0	--	1	0	0	1	0	0	0	0
LNE	0	0	0	0	1	--	0	0	1	0	0	0	0
NEL	7	6	0	0	5	7	--	0	3	0	0	0	0
NLE	3	3	1	1	2	3	0	--	2	0	0	0	0
Gibbs	5	5	1	1	3	5	2	2	--	2	1	1	1
C^3	7	8	2	1	7	9	3	2	3	--	0	0	0
C^3+C	7	8	2	1	7	8	2	3	4	0	--	0	0
C^3+S	9	8	3	3	8	8	4	5	6	3	1	--	1
C^3+SC	9	8	3	5	8	9	4	5	6	4	1	0	--

- Performed paired t-test (> 95%) between all algorithms pairs
- C^3 significantly outperforms other models in most cases

Summary so far…

- Graph identification is general framework for dealing with noisy structured data
- Here, we saw a preliminary approach based on collections of local classifiers

- Many open issues….

1. Query-time GI

- Instead of viewing as an off-line knowledge reformulation process

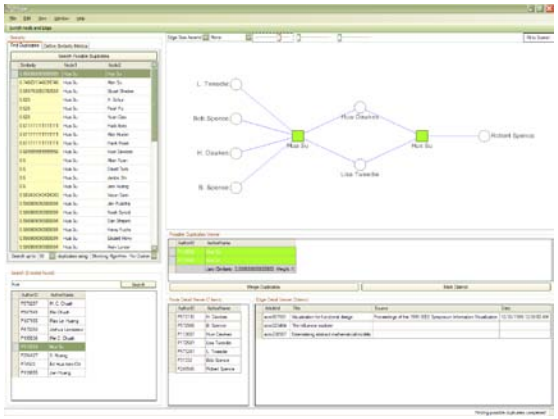
- consider as real-time data gathering with
  - varying resource constraints
  - ability to reason about value of information
  - e.g., what attributes are most useful to acquire? Which relationships? Which will lead to the greatest reduction in ambiguity?

- *Query-time Entity Resolution,* Bhattacharya & Getoor, Journal of Artificial Intelligence Research, 2007
- *Active Learning for Networked Data,* Bilgic, Mihalkova & Getoor, International Conference on Machine Learning, 2010

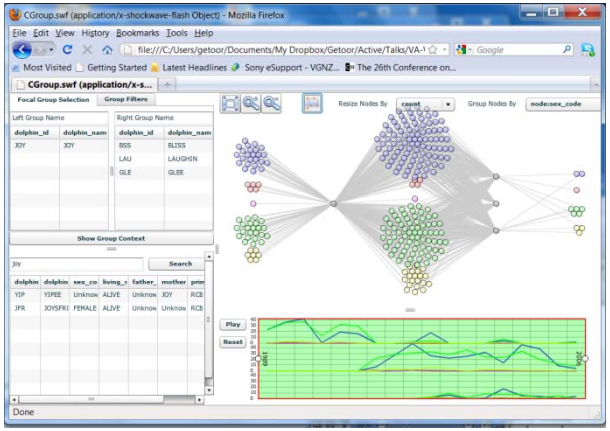
2. Visual Analytics for GI

- Combining rich statistical inference models with visual interfaces that support knowledge discovery and understanding
- Because the statistical confidence we may have in any of our inferences may be low, it is important to be able to have a human in the loop, to understand and validate results, and to provide feedback.
- Especially for graph and network data, a well-chosen visual representation, suited to the inference task at hand, can improve the accuracy and confidence of user input

Three Tools



D-Dupe



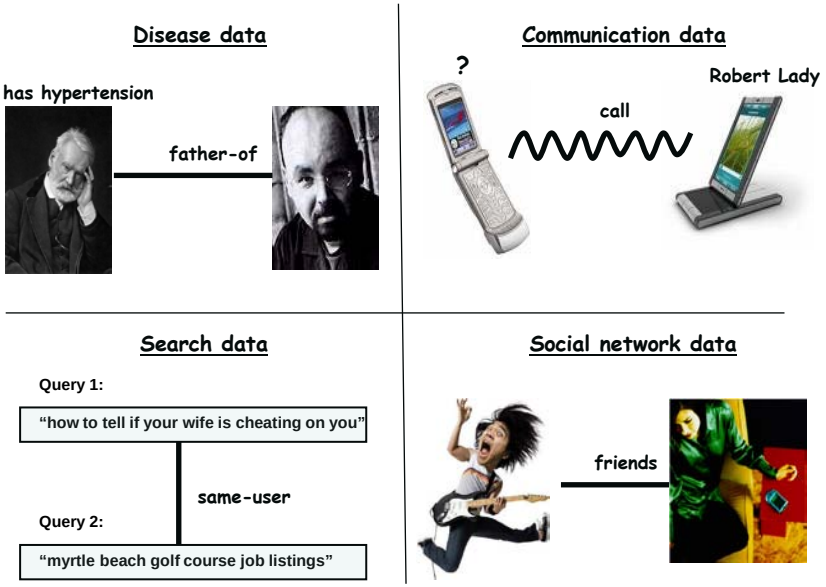
C-Group

G-View

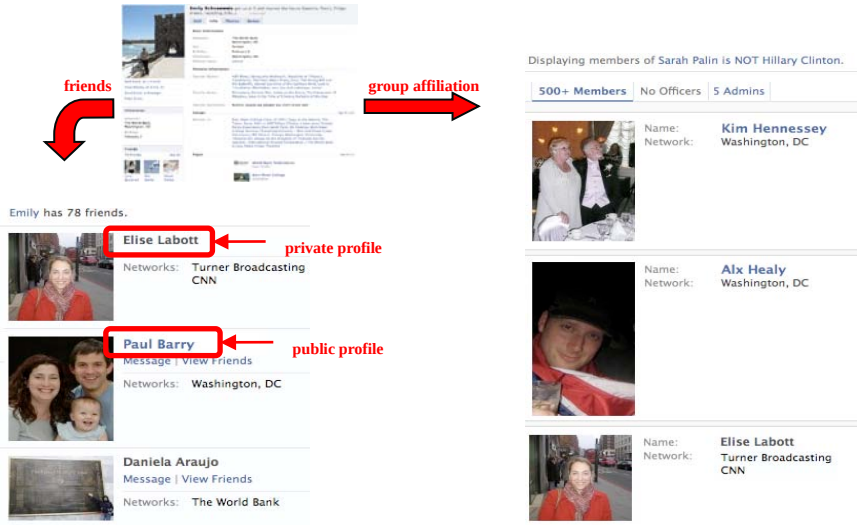
3. GI & Privacy

- Obvious privacy concerns that need to be taken into account!!!
- A better theoretical understanding of when graph identification is feasible will also help us understand what must be done to maintain privacy of graph data
- … Graph Re-Identification: study of anonymization strategies such that the information graph **cannot** be inferred from released data graph

Some relevant work



***Preserving the Privacy of Sensitive Relationships in Graph Data*, Zheleva and Getoor, PINKDD 07**



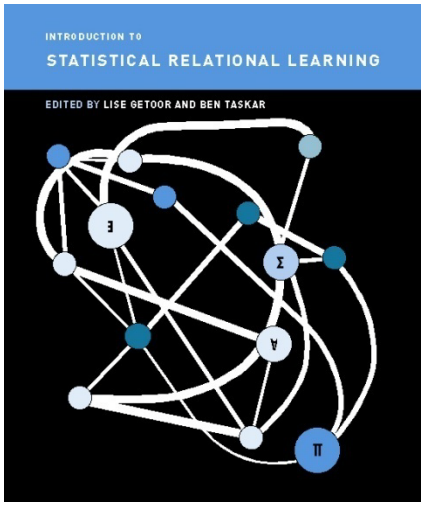
Privacy in Social Networks: A Survey, Zheleva and Getoor, book chapter in Social Network Data Analytics 2010.

To Join or Not to Join: the Illusion of Privacy in Online Social Networks, Zheleva and Getoor,, WWW 2009

Statistical Relational Learning (SRL)

- Methods that combine expressive knowledge representation formalisms such as relational and first-order logic with principled probabilistic and statistical approaches to inference and learning

Dagstuhl April 2007



- Hendrik Blockeel, Mark Craven, James Cussens, Bruce D'Ambrosio, Luc De Raedt, Tom Dietterich, Pedro Domingos, Saso Dzeroski, Peter Flach, Rob Holte, Manfred Jaeger, David Jensen, Kristian Kersting, Heikki Mannila, Andrew McCallum, Tom Mitchell, Ray Mooney, Stephen Muggleton, Kevin Murphy, Jen Neville, David Page, Avi Pfeffer, Claudia Perlich, David Poole, Foster Provost, Dan Roth, Stuart Russell, Taisuke Sato, Jude Shavlik, Ben Taskar, Lyle Ungar and many others

Conclusion

- Graph Identification
  - can be seen as a process of **data cleaning** and **knowledge reformulation**
  - In the context where we have some relational information that tells us about the structure of the graph that helps us to define features and statistical information to help us **learn** which reformulations are more promising than others
- While there are important pitfalls to take into account (confidence and privacy), there are many potential benefits and payoffs

Thanks!

http://www.cs.umd.edu/linqs

Work sponsored by the National Science Foundation,
KDD program, National Geospatial Agency, Google, Microsoft and Yahoo!

KDD Program