Graph Identification

Lise Getoor

University of Maryland, College Park

http://www.cs.umd.edu/~getoor

SFI Workshop on Power Grids as Complex Networks
May 18, 2012

DISCLAIMER

~~Physicist~~

~~Electrical Engineer~~

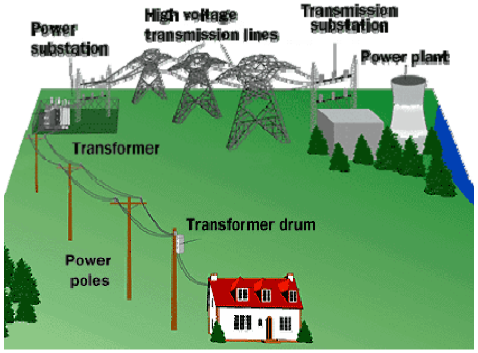
~~Statistician~~

Computer Scientist

machine learning

data miner

+



= ?

Three Take Away Messages

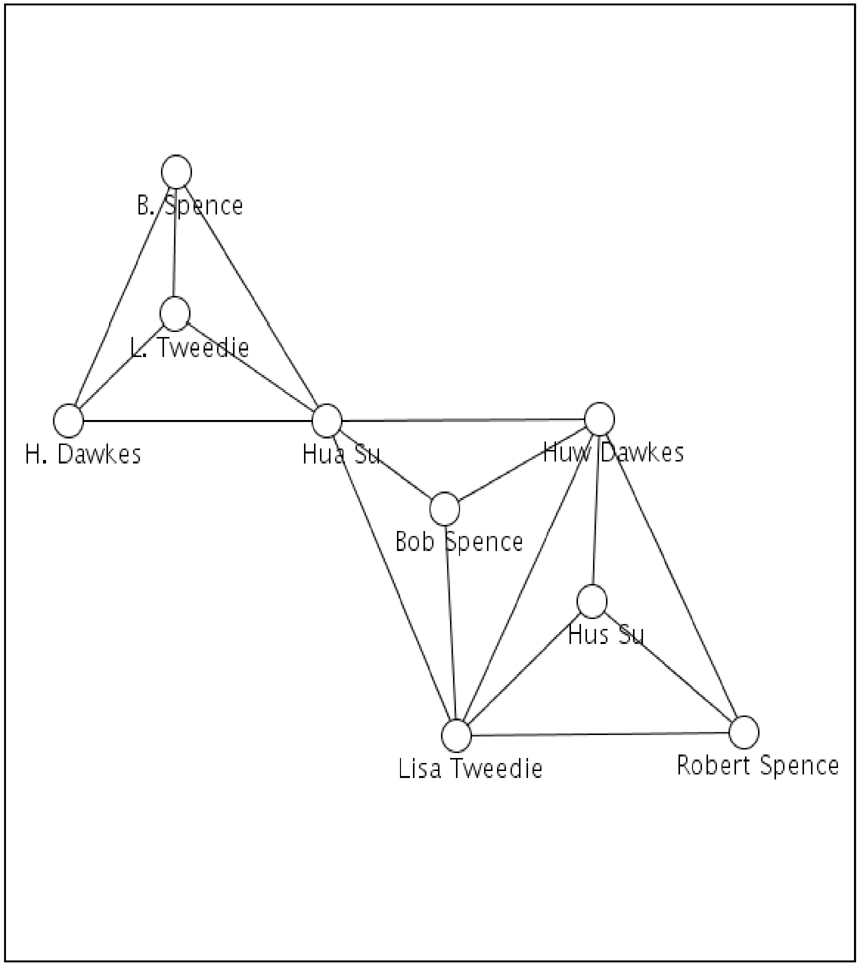
#1 Pitfall

#2 Paradigm

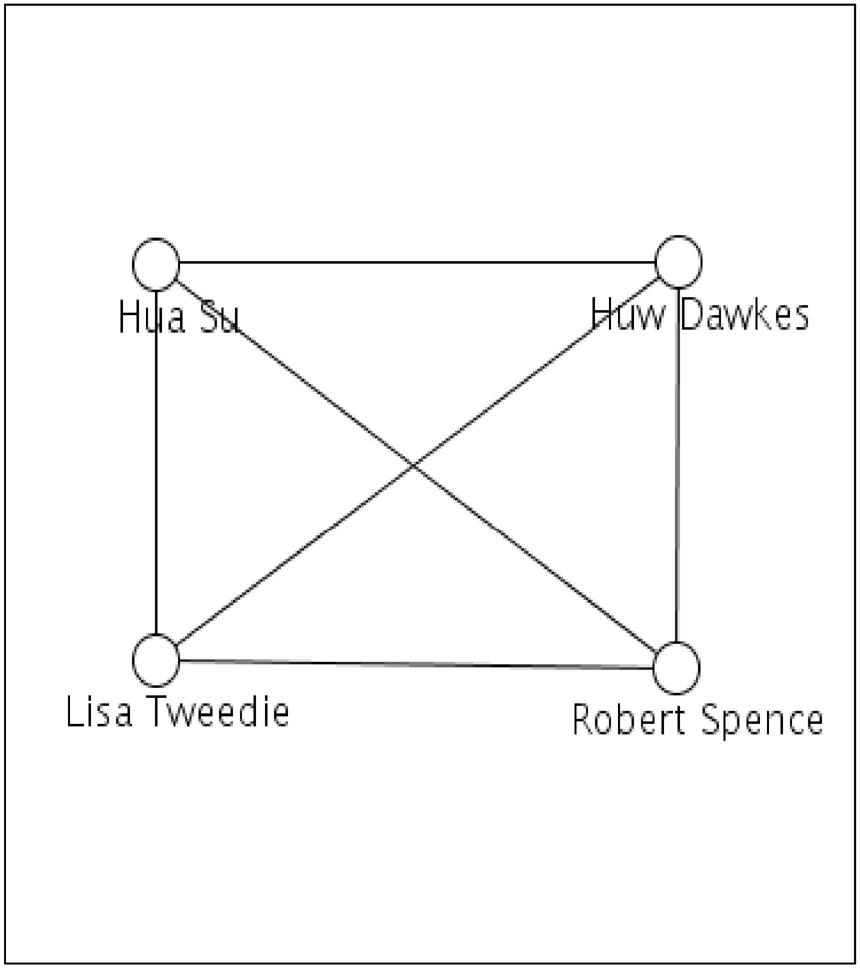
#3: View

#1: PITFALL

InfoVis Co-Author Network Fragment



before



after

IP Aliasing Problem [Willinger et al. 2009]

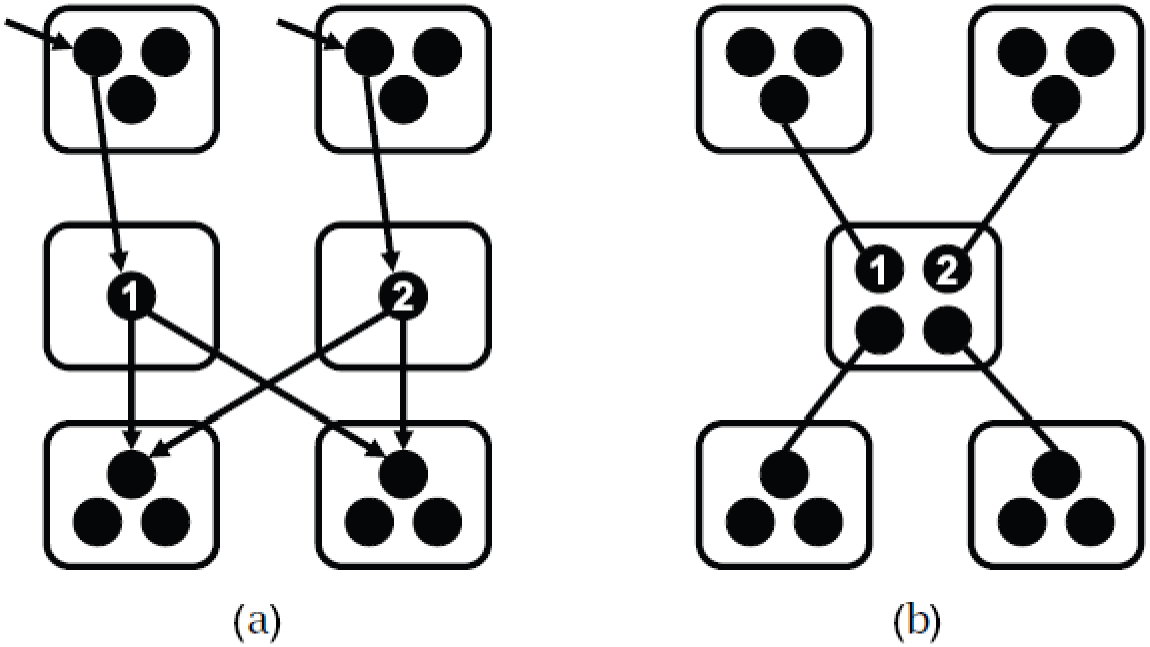


Figure 2. The IP alias resolution problem. Paraphrasing Fig. 4 of [50], `traceroute` **does not list routers (boxes) along paths but IP addresses of input interfaces (circles), and alias resolution refers to the correct mapping of interfaces to routers to reveal the actual topology. In the case where interfaces 1 and 2 are aliases, (b) depicts the actual topology while (a) yields an "inflated" topology with more routers and links than the real one.**

IP Aliasing Problem [Willinger et al. 2009]

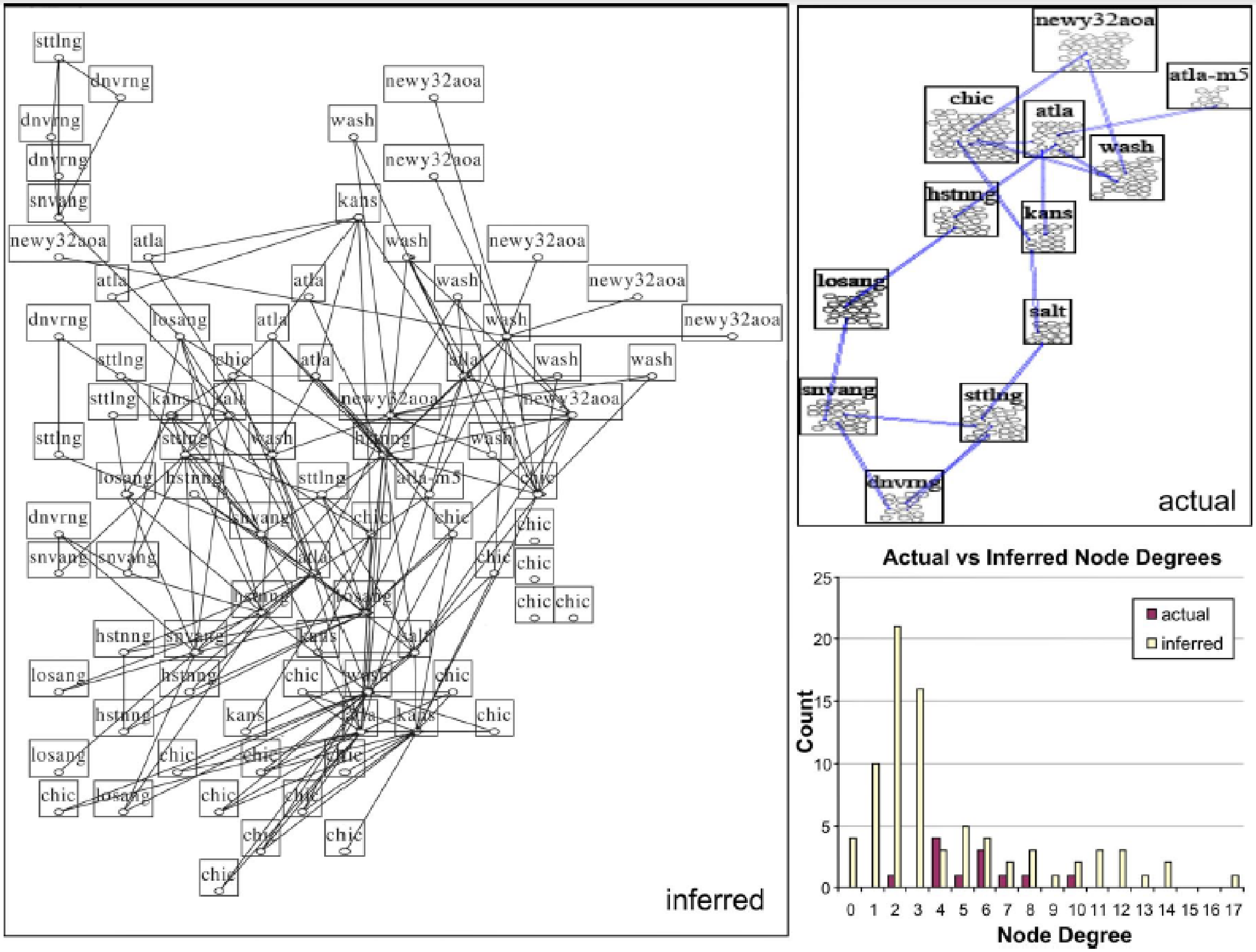


Figure 3. The IP alias resolution problem in practice. This is re-produced from [48] and shows a comparison between the Abilene/Internet2 topology inferred by Rocketfuel (left) and the actual topology (top right). Rectangles represent routers with interior ovals denoting interfaces. The histograms of the corresponding node degrees are shown in the bottom right plot. © 2008 ACM,

ENTITY RESOLUTION

Indrajit Bhattacharya

Collective Entity Resolution in Relational Data, Bhattacharya & Getoor, Transactions on Knowledge Discovery & Data Mining (TKDD), 2007

A Latent Dirichlet Model for Unsupervised Entity Resolution,
Bhattacharya & Getoor, SIAM Conference on Data Mining (SDM) , 2006

#2: PARADIGM

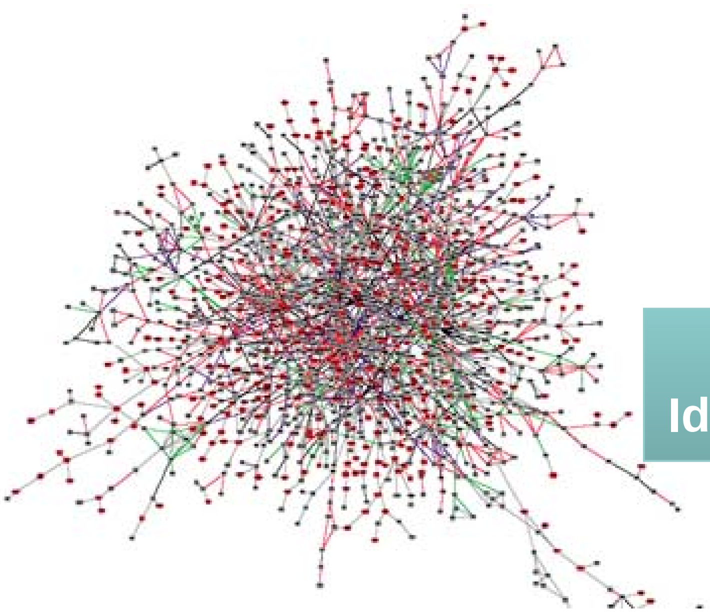
GRAPH IDENTIFICATION

Joint work with Galileo Namata and Stanley Kok

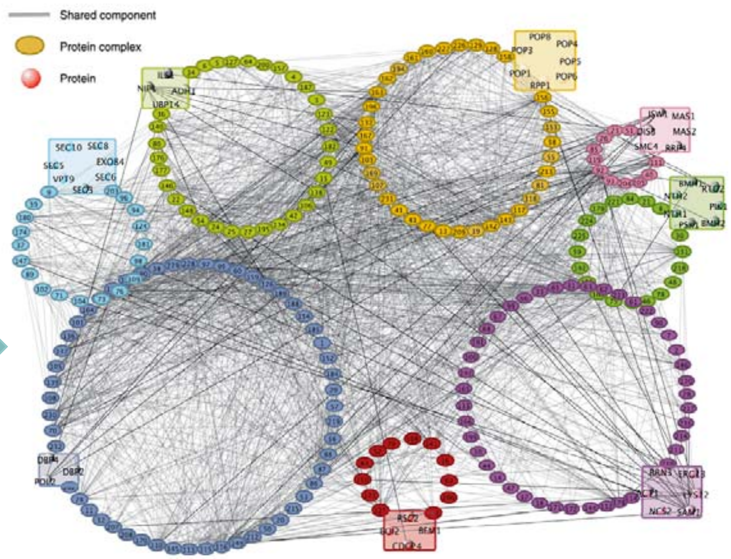
Wealth of Data

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

Transformation



Graph Identification



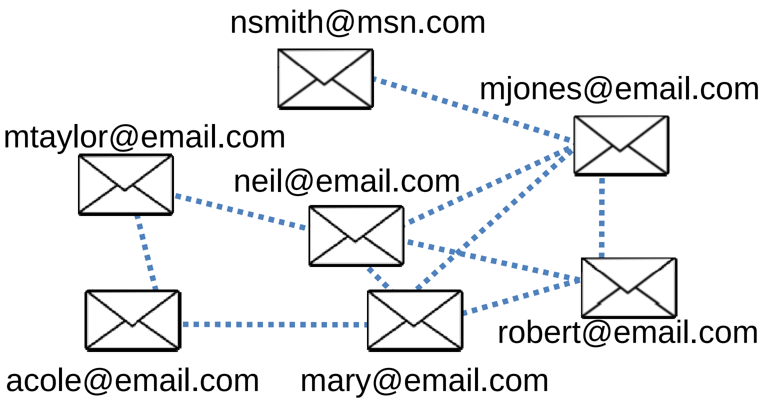
Input Graph

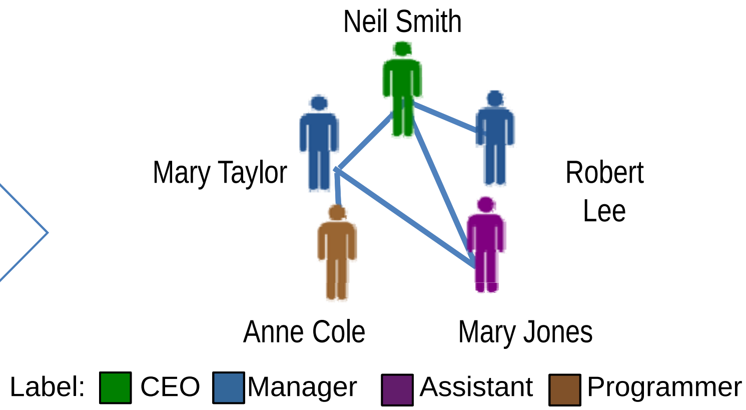
Output Graph

Available but inappropriate for analysis

Appropriate for further analysis

Motivation: Different Semantics





- Resolve email addresses
- Infer edges
- Infer labels

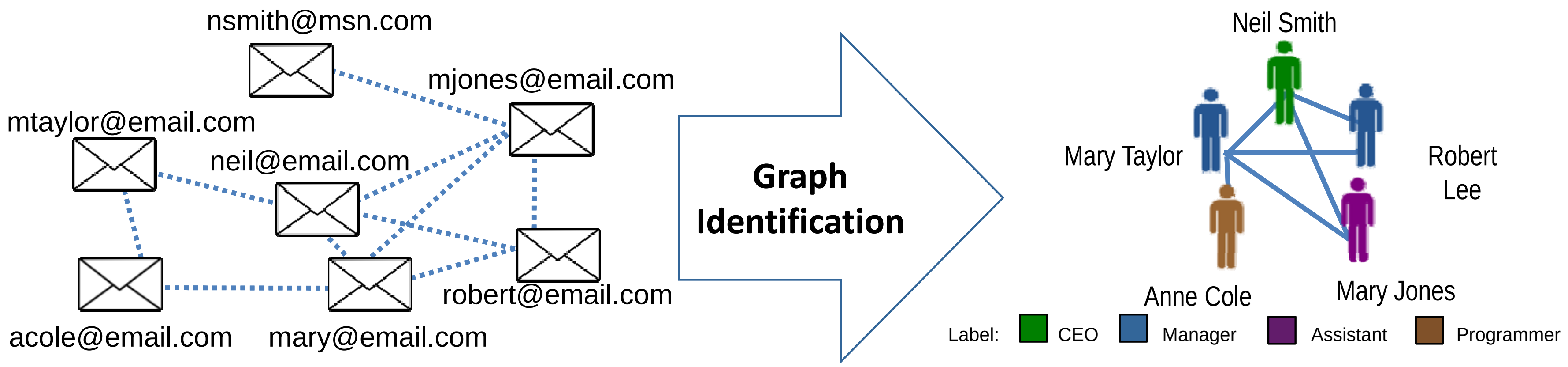
<u>Observational Network</u>
Nodes: Email Address
Edges: Communication
Node Attributes: Words

<u>Organizational Network</u>
Nodes: Person
Edges: Manages
Node Labels: Title

Graph Identification

- Goal:
  - Given an **input graph** infer an **output graph**
- Consists of 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
- Problem: The components are intra and inter-dependent

Graph Identification



nsmith@msn.com

mjones@email.com

mtaylor@email.com

neil@email.com

robert@email.com

acole@email.com

mary@email.com

Graph Identification

Neil Smith

Mary Taylor

Robert Lee

Anne Cole

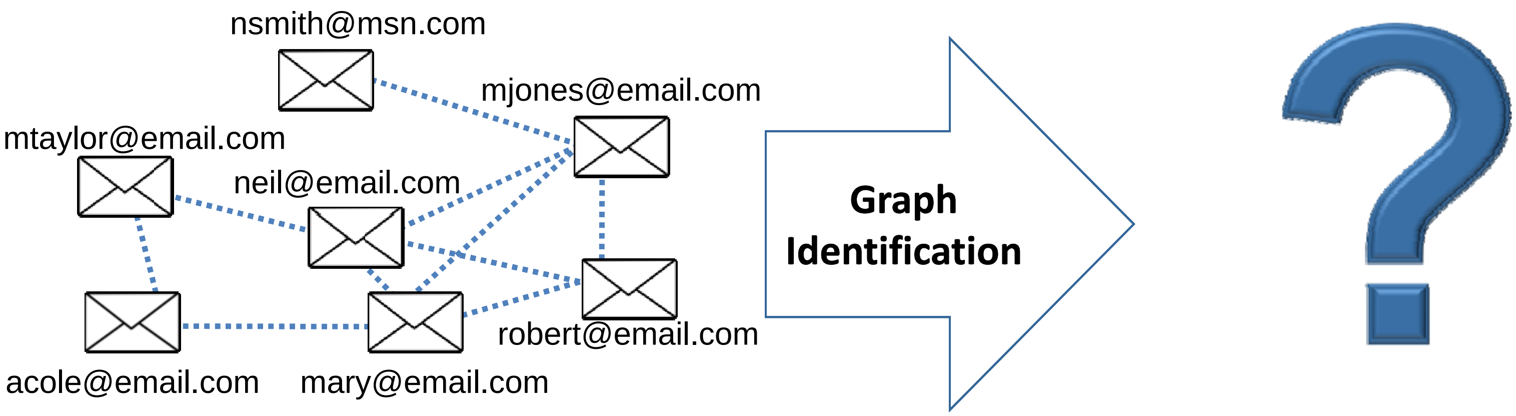
Mary Jones

Label: 🟩 CEO 🟦 Manager 🟪 Assistant 🟫 Programmer

Input Graph: Email Communication Network

Output Graph: Social Network

Graph Identification

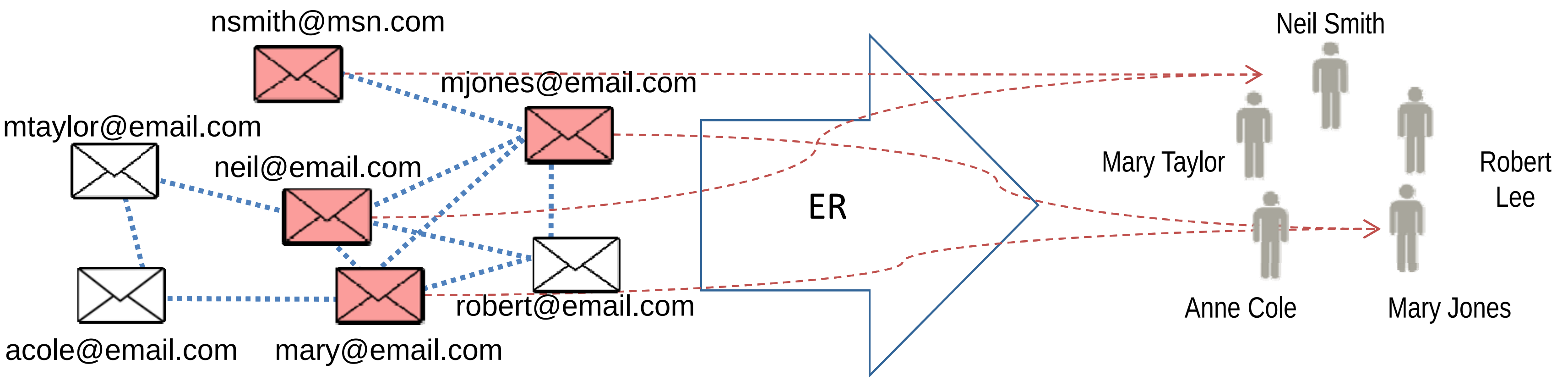


Input Graph: Email Communication Network

Output Graph: Social Network

- What’s involved?

Graph Identification

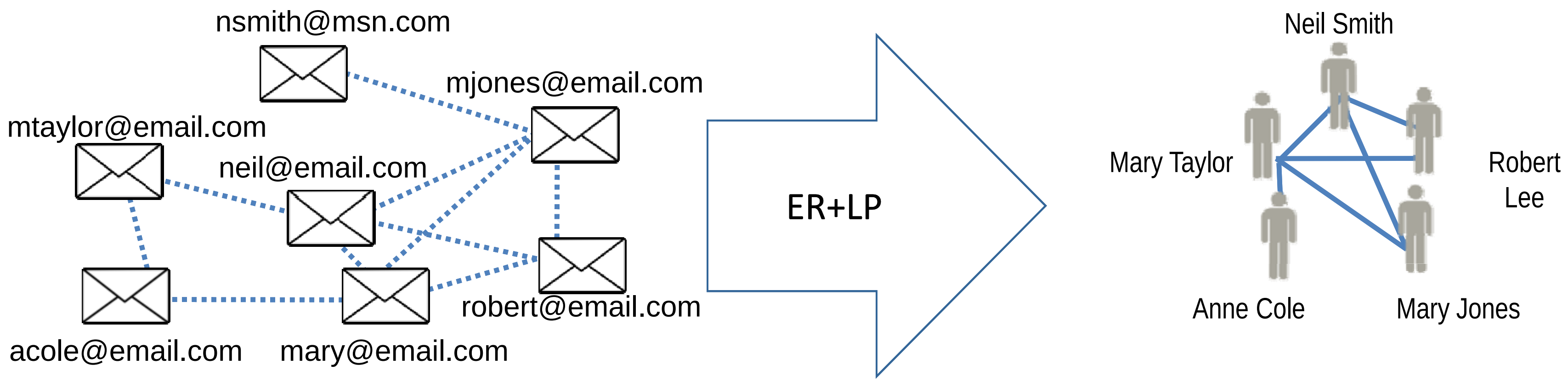


Input Graph: Email Communication Network

Output Graph: Social Network

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

Graph Identification

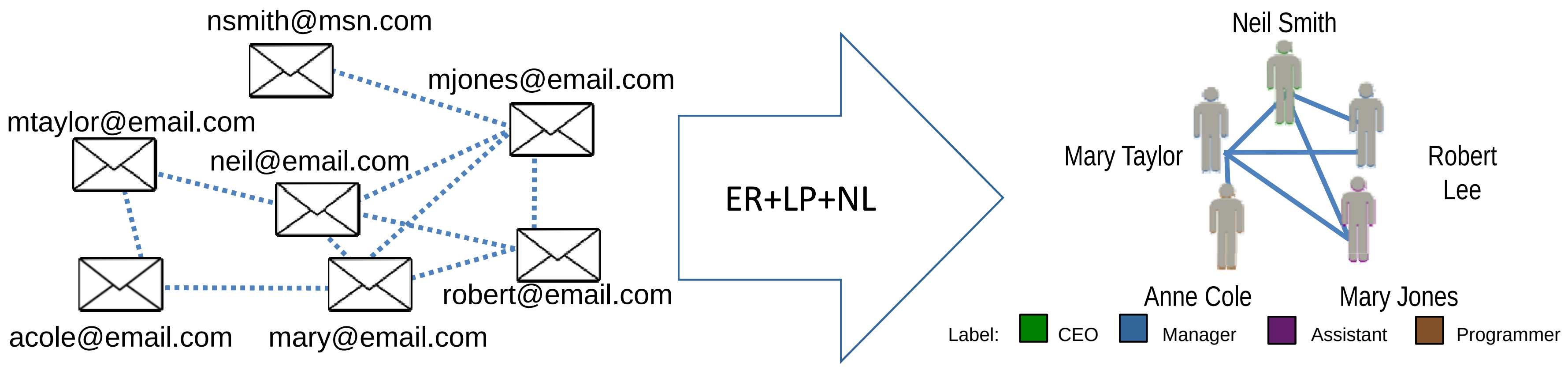


Input Graph: Email Communication Network

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

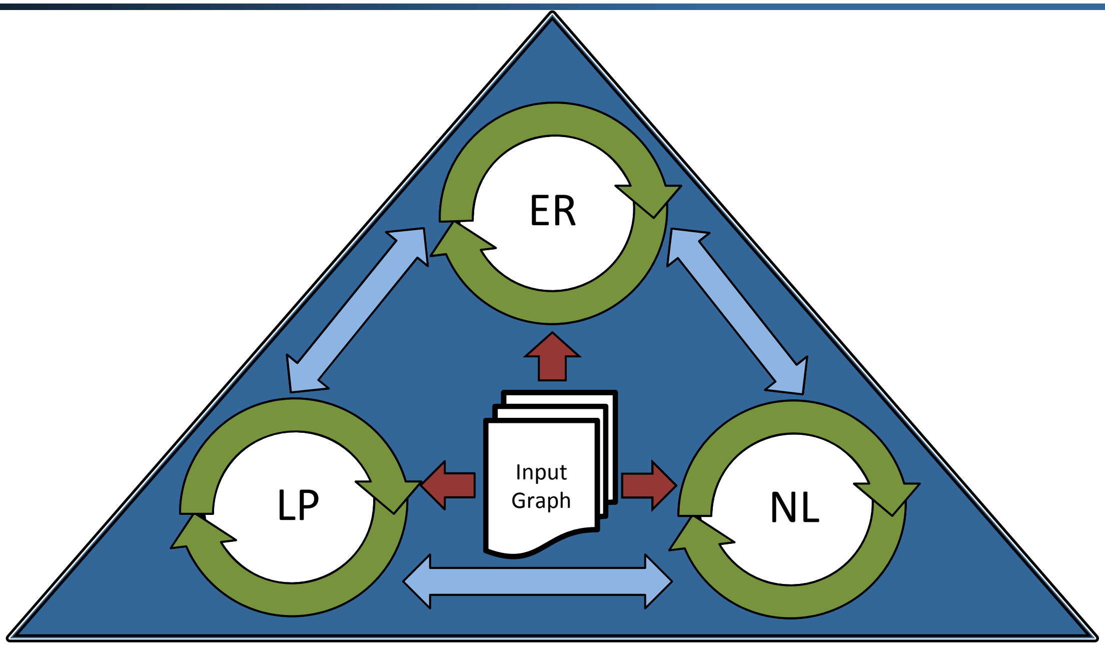


Input Graph: Email Communication Network

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
  - Node Labeling (NL): Infer the labels of nodes in the output graph

Problem Dependencies



- Most work looks at these tasks in ***<u>isolation</u>***
- In graph identification they are:
  - Evidence-Dependent – Inference depend on observed input graph
    e.g., ER depends on input graph
  - Intra-Dependent – Inference <u>within</u> tasks are dependent
    e.g., NL prediction depend on other NL predictions
  - Inter-Dependent – Inference <u>across</u> tasks are dependent
    e.g., LP depend on ER and NL predictions

Challenge

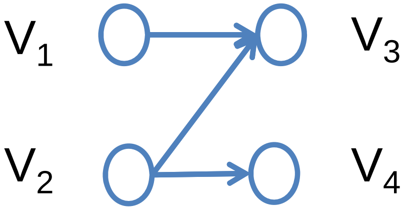
- How to perform graph identification given:
  - Multiple diverse tasks involved
  - Large number of dependencies
- Solution:
  - Iterative approach using **C**oupled **C**ollective **C**lassifiers

C^3

Problem Setup

- Random Variables: **y** = **r** U **l** U **n**
- Entity resolution: **r** = $\{r_{ij}\}$
  - Binary variable r_{ij} = 1 iff V_i and V_j are co-referent
- Link prediction: **l** = $\{l_{ij}\}$
  - Binary variable l_{ij} = 1 iff edge from V_i to V_j in the output graph
- Node
  - Di noting the label of a node V_i in the output graph

Quadratic in number of nodes?



Input Graph

r	**l**	**n**
r_{12}	l_{12}	n_1
r_{13}	l_{13}	n_2
r_{14}	l_{14}	n_3
r_{23}	…	n_4
r_{24}	l_{41}	
r_{34}	l_{42}	
	l_{43}	

Problem Definition

- Define a joint distribution over these random variables, $\mathbf{y} = \mathbf{r} \cup \mathbf{l} \cup \mathbf{n}$

$$P(\mathbf{y}) = \frac{1}{Z} \exp\left(\sum_{c \in \mathcal{C}} \phi_c(\mathbf{y}_c) \right)$$

- Represent as a log linear combination

$$P(\mathbf{y}) = \frac{1}{Z} \exp(\sum_{f \in \mathcal{F} : y \in \mathbf{y}} \mathbf{w}_f \cdot \mathbf{f}(\mathbf{y}_f))$$

- Given evidence **x**, graph identification problem can be defined via conditional Markov network

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp(\sum_{f \in \mathcal{F} : y \in \mathbf{y}} \mathbf{w}_f \cdot \mathbf{f}(\mathbf{x}_f, \mathbf{y}_f))$$

Problem Definition

- Define a joint distribution over these random variables, $\mathbf{y} = \mathbf{r} \cup \mathbf{l} \cup \mathbf{n}$

$$P(\mathbf{y}) = \Bigg(\; _{c\in\mathcal{C}} \; \Bigg)$$

Intractable!!!

- Represent as ear combination

$$P(\mathbf{y}) \quad \text{xp}(\sum_{f\in\mathcal{F}:y\in\mathbf{y}} \mathbf{w}_f \cdot \mathbf{f}(\mathbf{y}_f))$$

- Given evidence **x**, graph identification problem can be defined via conditional Markov network

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \exp(\sum_{f\in\mathcal{F}:y\in\mathbf{y}} \mathbf{w}_f \cdot \mathbf{f}(\mathbf{x}_f, \mathbf{y}_f))$$

C³ Approach

- We perform inference by:
  - Using a two-tiered iterative approach based on approximation:

$$P(\mathbf{r},\mathbf{l},\mathbf{n} \mid \mathbf{x}) \approx \left(\prod_{r\in\mathbf{r}} P(r|\mathbf{y}\setminus r,\mathbf{x})\right)\left(\prod_{l\in\mathbf{l}} P(l|\mathbf{y}\setminus l,\mathbf{x})\right)\left(\prod_{n\in\mathbf{n}} P(n|\mathbf{y}\setminus n,\mathbf{x})\right)$$

$$\approx \prod_{y\in\mathbf{r}\cup\mathbf{l}\cup\mathbf{n}} \frac{\exp\left(\sum_{f\in\mathcal{F}:y\in\mathbf{y}_f} w_f \cdot f(\mathbf{x}_f,\mathbf{y}_f)\right)}{Z(\mathbf{y}_f\setminus y,\mathbf{x})}$$

  - Assume weights of the features across the tasks are distinct to use standard classifiers (e.g., SVM, logistic regression) for prediction within each iteration

Feature Functions

- Rich set of features supported
  - Attribute and relational similarity measures
  - Structural properties and path existence
  - Aggregates over set of values

- Local Features $\mathbf{f}^{local}$: computed based solely on evidence

  e.g., $f_{ER}(V_i,V_j)$ = Cosine similarity of *observed* attributes of V_i and V_j

- Relational Features $\mathbf{f}^{rel}$: computed based on inferred values within tasks

  e.g., $f_{NL}(V_i)$ = Proportion of V_i *observed* neighbors with *predicted* label L

  and among tasks (coupling the classifiers)

  e.g., $f_{NL}(V_i)$ = Proportion of V_i *predicted* neighbors with *predicted* label L

C^3 Inference Variants

- Basic Model (C^3)
  - At each iteration, assign most probable value, recompute features. Repeat until convergence
- Simulated Annealing (C^3-SA)
  - For iteration *i*, with probability = (*i*/*maxIteration*), assign to variables to most likely value. Otherwise, sample value from probability distribution.
- Cautious Inference (C^3-CI)
  - At every iteration, commit only the top K = ((*i*/*maxIteration*)**numPerTask*) most confident values
- Gibbs Sampling (C^3-GS)
  - Sample value from probability distribution. After "burn-in" period, count number of times each value is sampled for each random variable. Assign random variable to most frequently assigned value.
- Expectation Maximation (C^3-EM)
  - Retrain relational classifiers at each iteration

Evaluation

- Evaluation using four real world datasets
  - Enron, Discourse, Cora, Citeseer
  - http://www.cs.umd.edu/projects/linqs/c3/
- Email Communication and Social Network
  - Networks manually annotated from Enron dataset
  - Email Network: 211 email nodes, 2837 communication edges
  - Social Network: 146 person nodes, 139 managerial edges, 7 labels
  - Problem: Given email communication network, infer the social network
- Discourse Opinion Network
  - Networks from Somasundaran et al. (2009)
  - Co-Occurrence Network: 4606 opinion nodes, 22925 co-occurrence edges
  - Opinion Reinforcement Network: 4606 opinion nodes, 3920 objects, 1045 reinforcement edges, 3 labels
- Vary amount of annotations (Low, Medium, High)

Evaluation

- Citation Networks
  - Cora – 2708 paper nodes, 5428 citation edges, 7 labels
  - Citeseer – 3312 paper nodes, 4732 citation edges, 6 labels
  - Generate reference, link, and attribute noise
    - Vary amount of noise (Low, Medium, High)
  - Problem: Given noisy "extracted" network, infer the citation network
- Vary amount of annotations (Low, Medium, High)

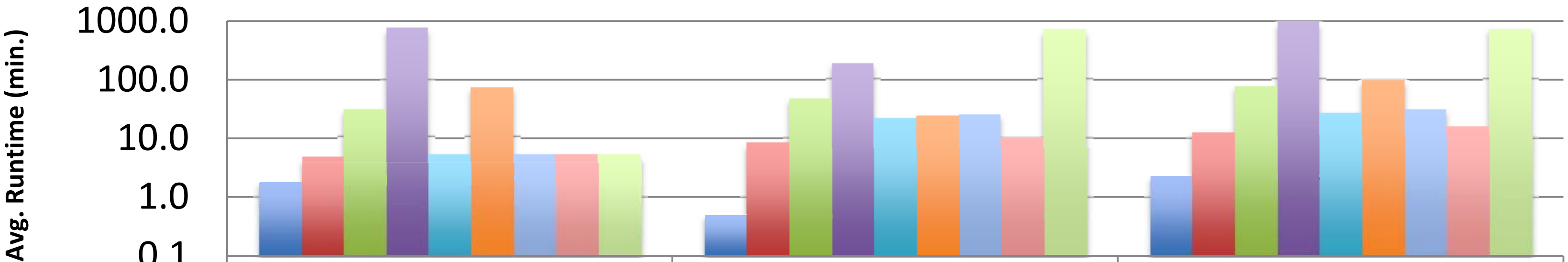
Algorithms

- **C3:** using SVM with linear kernel for classifiers
  - Component of Graph Alignment, Identification, and Analysis (GAIA) software library http://linqs.cs.umd.edu/gaia
- **LOCAL**: only the local features
- **INTRA**: relational classifiers using only features capturing intra-dependencies
- **PIPELINE**: relational classifiers in a pipeline
  - Perform tasks sequentially (evaluate all possible orderings)
  - **PIPELINE*** results for the best performing order
- **MLN**: Markov Logic Networks (Richardson and Domingos, 2006)

Results (Avg F1 performance)

		Citeseer (Vary Noise Level)			Cora (Vary Noise Level)			Enron	Discourse
		Low	**Medium**	**High**	**Low**	**Medium**	**High**		
Low	**LOCAL**	0.800	0.736	0.657	0.827	0.756	0.645	0.425	0.361
	INTRA	0.843	0.792	0.745	0.900	0.854	0.798	0.516	0.648
	PIPELINE*	0.871	0.834	0.793	0.939	0.911	0.878	0.559	0.706
	MLN	0.677	0.673	0.663	0.570	0.560	0.591	0.137	0.320
	C^3	**0.882**	**0.853**	**0.819**	**0.950**	**0.928**	**0.899**	**0.550**	**0.729**
Medium	**LOCAL**	0.786	0.725	0.648	0.821	0.747	0.639	0.363	0.309
	INTRA	0.833	0.782	0.730	0.889	0.840	0.778	0.465	0.545
	PIPELINE*	0.853	0.816	0.768	0.921	0.888	0.849	0.509	0.604
	MLN	0.425	0.534	0.563	0.456	0.519	0.470	0.143	0.217
	C^3	**0.861**	**0.828**	**0.782**	**0.934**	**0.900**	**0.862**	**0.515**	**0.658**
High	**LOCAL**	0.775	0.716	0.633	0.800	0.734	0.626	0.398	0.232
	INTRA	0.816	0.770	0.708	0.868	0.816	0.741	0.448	0.351
	PIPELINE*	0.831	0.795	0.743	0.895	0.861	0.811	0.479	0.419
	MLN	0.216	0.222	0.228	0.190	0.211	0.216	0.096	0.143
	C^3	**0.835**	**0.801**	**0.750**	**0.902**	**0.869**	**0.819**	**0.479**	**0.483**

Average Runtime Performance



	Learning Time	Inference Time	Total Time
🟦 LOCAL	1.8	0.5	2.3
🟥 INTRA	4.6	8.4	13.0
🟩 PIPELINE*	29.7	46.1	75.8
🟪 MLN	761.7	183.6	945.5
🟦 C3	5.2	21.5	26.7
🟧 C3-EM	71.6	23.7	95.3
🟦 C3-SA	5.2	25.0	30.1
🟥 C3-CI	5.2	10.6	15.8
🟩 C3-GS	5.2	728.2	733.4

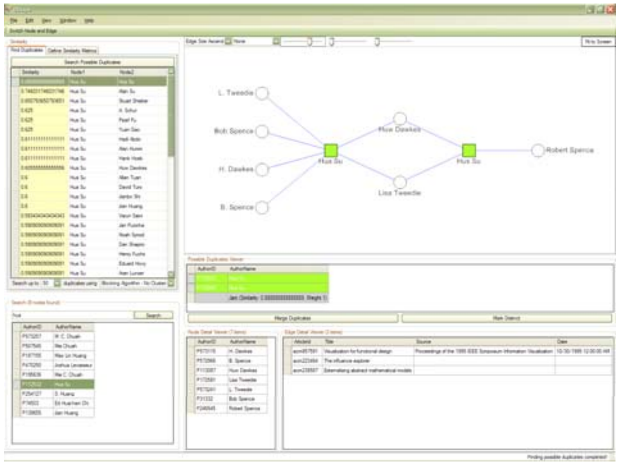
- Average runtime (in minutes) over a set of Cora experiments
  - Number of random variables: |**R**| ≈ 70000, |**L**| ≈ 35000, |**N**| ≈ 5900

#3: VIEW

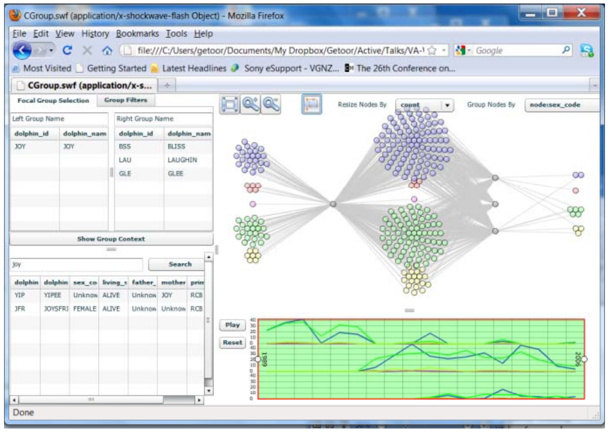
Visual Analytics

- Combining rich statistical inference models with visual interfaces that support knowledge discovery and understanding
- Because the statistical confidence in any of our inferences may be low, 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

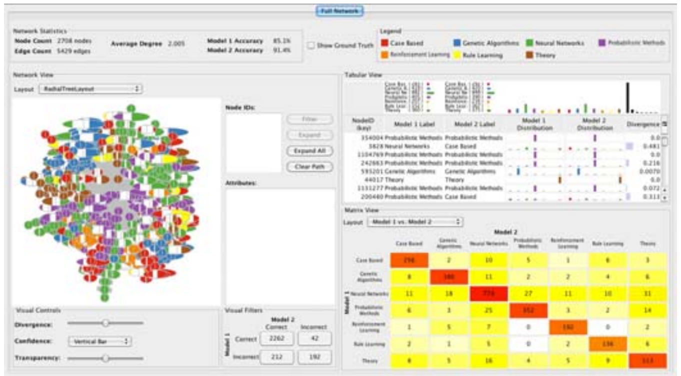
Three Viz Tools



D-Dupe



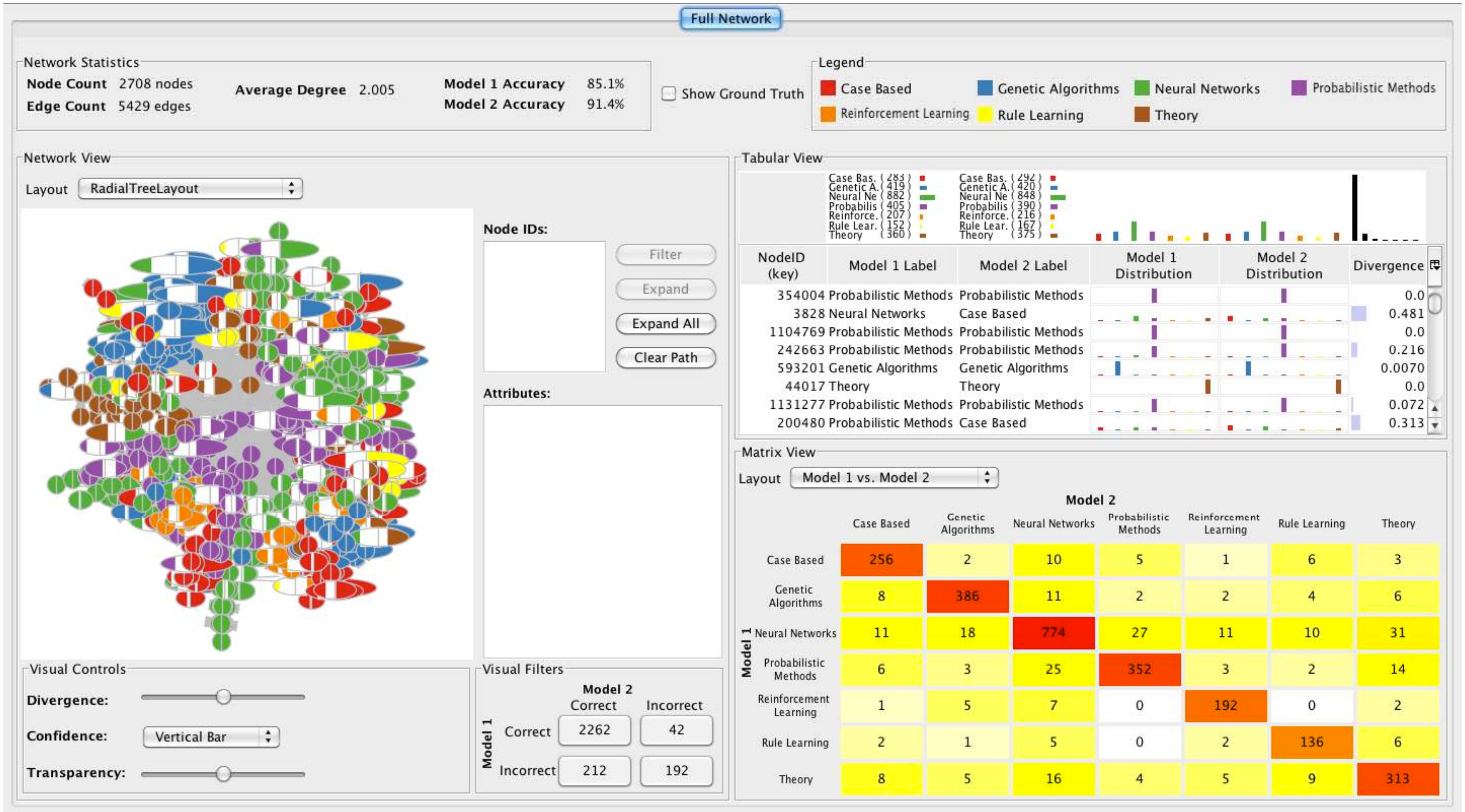
C-Group



Network Statistics
Node Count 2708 nodes
Edge Count 5429 edges
Average Degree 2.005
Model 1 Accuracy 85.1%
Model 2 Accuracy 91.4%

G-Pare

G-Pare: Comparative Analysis of Uncertain Graphs



Network Statistics

Node Count 2708 nodes
Edge Count 5429 edges
Average Degree 2.005
Model 1 Accuracy 85.1%
Model 2 Accuracy 91.4%

☐ Show Ground Truth

Legend

🟥 Case Based 🟦 Genetic Algorithms 🟩 Neural Networks 🟪 Probabilistic Methods
🟧 Reinforcement Learning 🟨 Rule Learning 🟫 Theory

Network View

Layout RadialTreeLayout

Node IDs:

Attributes:

Tabular View

	Model 1	Model 2
Case Bas.	(283)	(292)
Genetic A.	(419)	(420)
Neural Ne	(882)	(848)
Probabilis	(405)	(390)
Reinforce.	(207)	(216)
Rule Lear.	(152)	(167)
Theory	(360)	(375)

NodeID (key)	Model 1 Label	Model 2 Label	Model 1 Distribution	Model 2 Distribution	Divergence
354004	Probabilistic Methods	Probabilistic Methods			0.0
3828	Neural Networks	Case Based			0.481
1104769	Probabilistic Methods	Probabilistic Methods			0.0
242663	Probabilistic Methods	Probabilistic Methods			0.216
593201	Genetic Algorithms	Genetic Algorithms			0.0070
44017	Theory	Theory			0.0
1131277	Probabilistic Methods	Probabilistic Methods			0.072
200480	Probabilistic Methods	Case Based			0.313

Matrix View

Layout Model 1 vs. Model 2

Model 1 \ Model 2	Case Based	Genetic Algorithms	Neural Networks	Probabilistic Methods	Reinforcement Learning	Rule Learning	Theory
Case Based	256	2	10	5	1	6	3
Genetic Algorithms	8	386	11	2	2	4	6
Neural Networks	11	18	774	27	11	10	31
Probabilistic Methods	6	3	25	352	3	2	14
Reinforcement Learning	1	5	7	0	192	0	2
Rule Learning	2	1	5	0	2	136	6
Theory	8	5	16	4	5	9	313

Visual Controls

Divergence:

Confidence: Vertical Bar

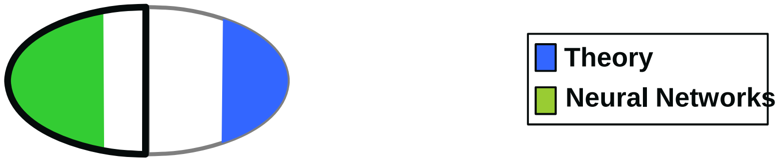
Transparency:

Visual Filters

Model 1 \ Model 2	Correct	Incorrect
Correct	2262	42
Incorrect	212	192

H. Sharara, A. Sopan , G. Namata, L.Getoor and L. Singh. "G-PARE: A Visual Analytic Tool for Comparative Analysis of Uncertain Graphs." submitted to *IEEE Conference on Visual Analytics Science and Technology (VAST'11)*

Node Visualization



- **Model 1 prediction: “Neural Networks” Model 2 prediction: “Theory”**

- **Model 1 is more confident in its prediction than Model 2**

- **Distributions of the two models vary significantly**

- **Model 1’s prediction matches the ground truth**

Case Study: Citation Network

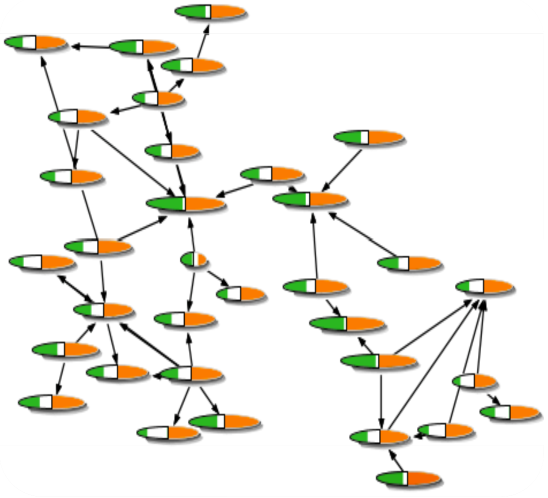
- Data set from Citeseer digital Library
  - 2120 publications with 3757 citation links
  - 3703 word vocabulary
  - Label indicating the topic of a paper

- Comparing two models for predicting the publication's topic
  - *Model 1* →(SVM) using only document content
  - *Model 2* →(Majority) using neighboring nodes' topics

Case Study: Citation Network

- Observations
  - Tabular view shows Model 2's predictions are skewed towards two topics
  - Network view shows large areas where the nodes are two-tone, where Model 2 is making the same incorrect prediction

- By filtering cases where Model 1 is correct and Model 2 is incorrect, we discover areas of flooding (propagation of error)



Conclusion

- **Pitfall:** Be sure that you are analyzing the right network!

- **Paradigm:** : Benefit in viewing analysis of noisy & incomplete data as statistical inference. ***Graph Identification*** is the process of inferring a 'correct' output graph from noisy input.

- **View: Visual tools** for comparative analytics are important for understanding and having confidence in models

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