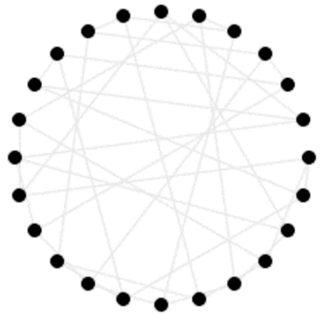
NYU STERN
NEW YORK UNIVERSITY · LEONARD N. STERN SCHOOL OF BUSINESS

Predictive modeling with social networks

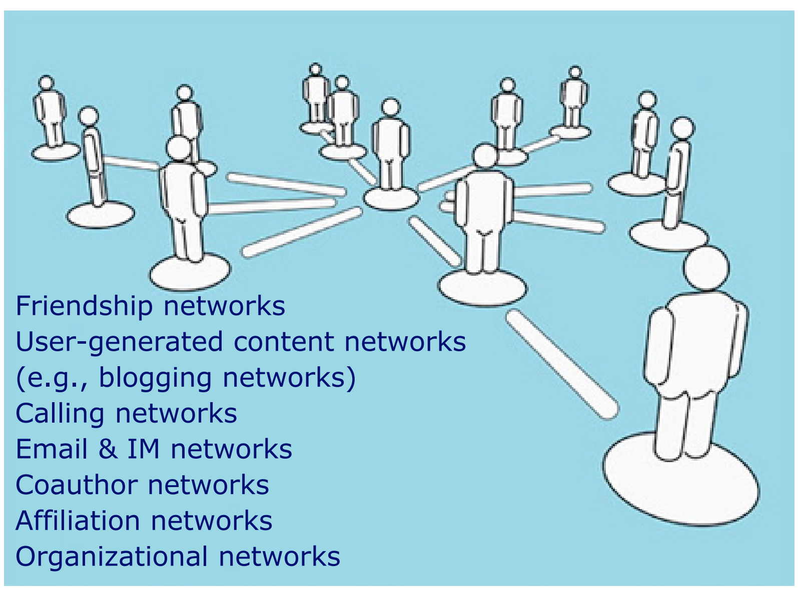
Jennifer Neville & Foster Provost



Tutorial at the Intl. Conf. on Weblogs and Social Media
May 2009

PURDUE UNIVERSITY

Social network data everywhere…

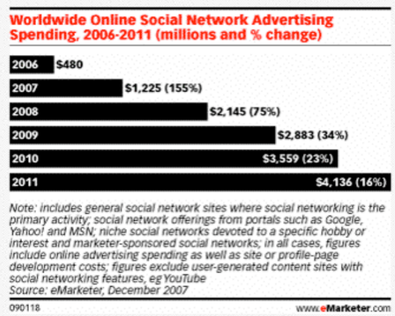


Friendship networks
User-generated content networks
(e.g., blogging networks)
Calling networks
Email & IM networks
Coauthor networks
Affiliation networks
Organizational networks

http://images.businessweek.com/ss/06/09/ceo_socnet/source/1.htm

eMarketer: Social Networking Ad Spend to Hit $4 Billion by 2011

December 14, 2007 — 07:57 AM PST — by Adam Ostrow —



eMarketer has a report out today that is a must-read for anyone in the social networking space. Among the highlights, eMarketer's research shows that 37% of the US adult population currently uses social networks, while 70% of teens do the same, with both numbers projected to rise significantly in coming years.

Meanwhile, the company projects that $1.2 billion will be spent advertising on social networks this year, with 70% of it going to the top two: MySpace and Facebook. By 2011, eMarketer projects total ad spend in the space growing to more than $4 billion.

Overall, the report paints a pretty rosy picture for all of us. What could send things

May 2009 update: Overall ad spending is down, but on-line advertising is faring better than off-line. Social network on-line advertisers report surprisingly little effect from recession.

BusinessWeek

MEDIA February 7, 2008, 5:00PM EST

Generation MySpace Is Getting Fed Up

Annoyed with the ad deluge on social networks, many users are spending less time on the sites

by Spencer E. Ante and Catherine Holahan

If you want to socialize with Chris Heritage, you won't find him on Facebook. The 27-year-old Port St. Lucie (Fla.) business analyst joined the social net year after his buddies bugged him to get an account. But he soon became fed up with the avalanche of ads, especially those detailing what his friends buying, and he quit the site in November. Now, Heritage expresses himself through a blog, happy to pay $6 a month to publish on a promo-free Web si worth it to not have to look at the ads," he says.

Uh-oh. Social networking was supposed to be the Next Big Thing on the Internet. MySpace, Facebook, and other sites have been attracting millions of building sprawling sites that companies are banking on to trigger an online advertising boom. Trouble is, the boom isn't booming anymore. Like Heritag people are spending less time on social networking sites or signing off altogether.

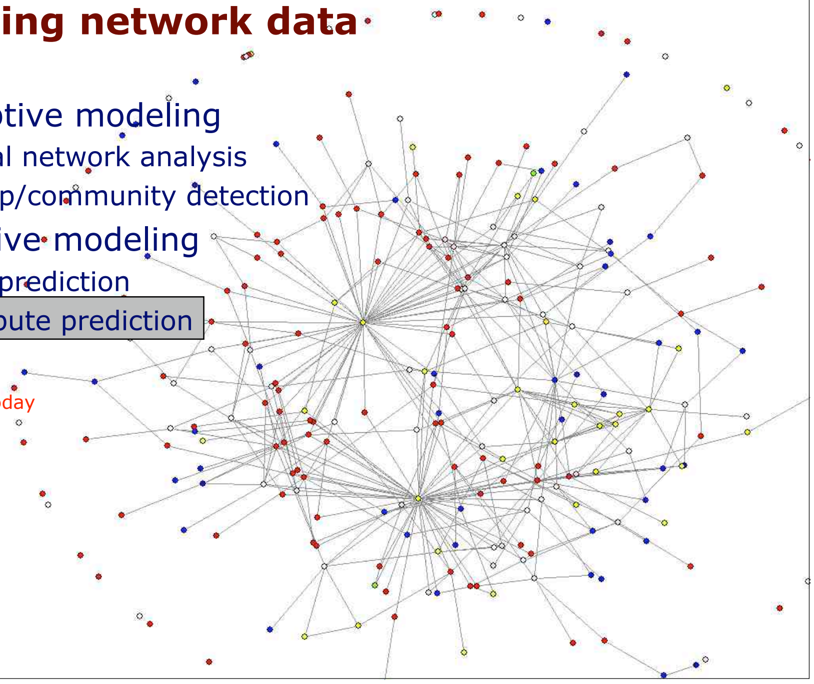
The MySpace generation may be getting annoyed with ads and a bit bored with profile pages. The average amount of time each user spends on social networking sites has fallen by 14% over the last four months, according to market researcher ComScore. MySpace, the largest social network, has slip peak of 72 million users in October to 68.9 million in December, ComScore says. The total number of people on such sites is still increasing at an 11.5 that's down sharply from past growth rates. "What you have with social networks is the most overhyped scenario in online advertising," says Tim Vande CEO of Specific Media, which places ads for customers on a variety of Web sites.

WISHFUL THINKING?

Advertising on social networking sites is growing fast. Last year global ad spending on these sites shot up 155%, to $1.2 billion, says researcher eMark year, eMarketer expects it to jump 75%, to $2.1 billion. During its Nov. 4 earnings call, News Corp. (NWS) gave an upbeat forecast for Fox Interactive M which includes MySpace.

But the forecasts for torrid growth may prove unrealistic. Besides the slowing user growth and declining time spent on these sites, users appear to be

Modeling network data



Descriptive modeling
- Social network analysis
- Group/community detection

Predictive modeling
- Link prediction
- Attribute prediction

our focus today

Goal of this tutorial

Our goal is *not* to give a comprehensive overview of relational learning algorithms (but we provide a long list of references and resources)

Our goal *is* to present

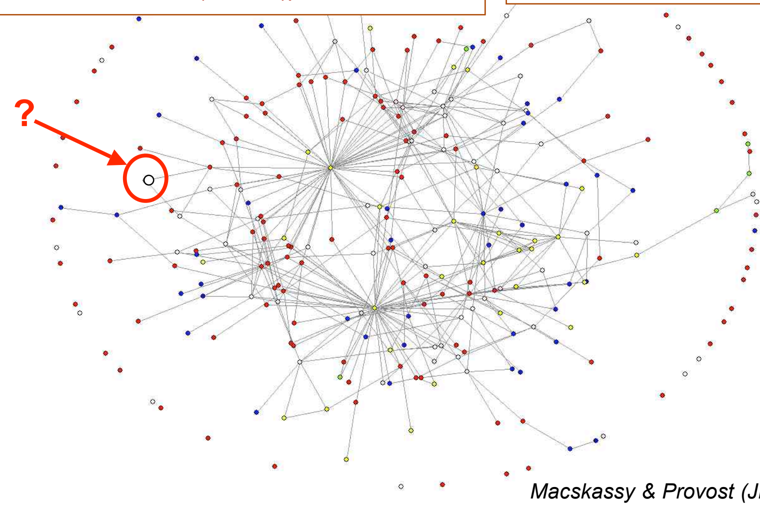
- the main ideas that differentiate predictive inference and learning with social network data,
- example techniques that embody these ideas,
- results, from real applications if possible
  - including a real application to social media (see supplemental slides)
- references and resources where you can learn more

In two hours we cannot hope to be comprehensive in our coverage of theory, techniques, or applications. We will present the most important concepts, illustrate with example techniques and applications, and provide a long list of additional resources.

The problem: Attribute Prediction in Networked Data

To start, we'll focus on the following inference problem:
For any node i, categorical variable y_i, and value c, estimate $p(y_i = c|\Delta_K)$

Δ_K is everything known about the network



Macskassy & Provost (JMLR 2007) provide a broad treatment for univariate networks

Outline of the tutorial: part I

The basics

- contemporary examples of social network inference in action
- what's different about network data?
- basic analysis framework
- (simple) predictive inference with univariate networks
  - disjoint inference
  - *network linkage can provide substantial power for inference, if techniques can take advantage of **relational autocorrelation***
- inductive inference (*learning*) in network data
  - disjoint learning – models learn correlation among attributes of labeled neighbors in the network

Note on terminology: In this tutorial, we use the term "inference" to refer to the making of predictions for variables' unknown values, typically using a model of some sort. We use "learning" to denote the building of the model from data (*inductive* inference). Generally we use the terminology common in statistical machine learning.
Note on acronyms: see reference guide at end of tutorial

Outline of the tutorial: part II

Moving beyond the basics

- collective inference
  - *network structure alone can provide substantial power for inference, if techniques can **propagate** relational autocorrelation*
  - *inferred covariates can influence each other*
- collective learning
  - learning using both the labeled and unlabeled parts of the network, requires collective inference
- social/data network vs. network of statistical dependencies
- **throughout:**
  - *example learning techniques*
  - *example inference techniques*
  - *example applications*

Supplemental topics

- methodology, evaluation, potential pathologies, understanding sources of error, other issues
- extended example with on-line social media data

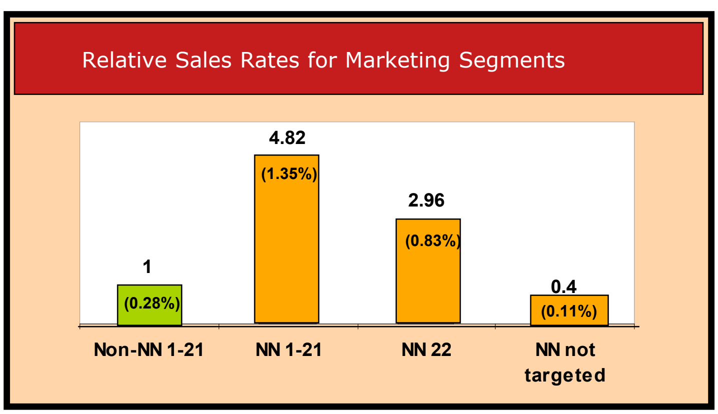
Let’s start with a real-world example

Example:
Network targeting *(Hill et al. '06)*

Define "Network Targeting" (NT)
- cross between viral marketing and traditional targeted marketing
- from simple to sophisticated…
  - construct variable(s) to represent whether the immediate network neighborhood contains existing customers
  - add social-network variables to targeting models, etc. (we'll revisit)
- then:
  - target individuals who are predicted (using the social network) to be the best prospects
  - simplest: target "network neighbors" of existing customers
  - this could expand "virally" through the network without any word-of-mouth advocacy, or could take advantage of it.

Example application:
- Product: new communications service
- Firm with long experience with targeted marketing
- Sophisticated segmentation models based on data, experience, and intuition
  - e.g., demographic, geographic, loyalty data
  - e.g., intuition regarding the types of customers known or thought to have affinity for this type of service

Sales rates are substantially higher for network neighbors *(Hill et al. '06)*



Firms increasingly are collecting data on explicit social networks of consumers

Other applications

Fraud detection
Targeted marketing
On-line advertising <-- extended example in supplemental slides
Bibliometrics
Firm/industry classification
Web-page classification
Epidemiology
Movie industry predictions
Personalization
Patent analysis
Law enforcement
Counterterrorism
…

TIME.com: Inside Bush's Secret Spy Net -- May 22, 2006 -- Page 1 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

TIME IN PARTNERSHIP WITH CNN

Quotes of the Da
"The diplomatic path
open." - Condoleezz
on North Korea

CURRENT ISSUE • SUBSCRIBE

TIME Archive TIME Mobile SUBSCRIBE TO TIME MAGAZINE FOR JUST $1.99

Home
Nation
World
Biz & Tech
Arts
Science & Health
Specials
Photos
Current Issue
Past Covers
TIME Blogs

Cover Story

BROOKS KRAFT / CORBIS FOR

Hayden fields media questions on Capitol Hill last week.

From the Magazine | Cover

Inside Bush's Secret Spy Net

Your phone records have been enlisted in th
war on terrorism. Should that make you wor
more or less?

By KAREN TUMULTY

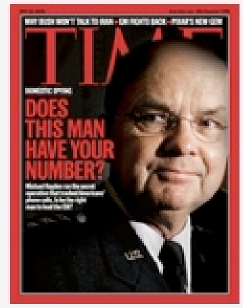
» SUBSCRIBE TO TIME PRINT E-MAIL MORE B
AUTHOR

Posted Sunday, May 14, 2006

Around the White House, an abrupt change in the

Does This Man Have Your Number?

• **Cover Story: Inside Bush's Secret Spy Net** Your phone records have been enlisted in the war on terrorism. Should that make you worry more or less?



• **Table of Contents**
May 22, 2006

THE ECONOMIC TIMES

Centre to map your phone network
14 Aug, 2007, 0038 hrs IST,Joji Thomas Philip, TNN

NEW DELHI: The government has decided to create a database of all mobile and fixed line calls within the country in an ambitious and unique attempt to track unlawf
activities by identifying calling patterns and mapping social networks.

The system will help the government track complete networks of "people who could possibly be involved in unlawful activities by creating a national database of all ind
Analysis of their call data records using advanced artificial intelligence techniques can help control unlawful activities," the department of telecom (DoT) has said.

The DoT's expenditure statement, which will be tabled in the Lok Sabha shortly, contains the broad outline of the plan and its rationale.

The Centre has already allocated Rs 15.4 crore to the Centre for Development of Telematics (C-DOT) to meet the initial costs associated with building this software pl
called 'Security Management for Law Enforcement Agencies'. C-DOT is an autonomous scientific and technical arm of the DoT.

The system will work like this: If you have a mobile or landline connection, the government will be able to keep track of the people with whom you interact with or talk
often — by scanning your telephone data records continuously. The calling pattern of every individual which consists of the frequently-called numbers will be tracked
analysed by a fully automated software platform that will be built by C-DOT.

This comes as the government feels that a database on both the identity and social networking matrix of all individuals based on their telephone usage pattern can he
provide useful inputs to the country's national security agencies. Mobile phone communication is playing an important role in tracking unlawful or terror-related activiti
Phone records and calling patterns of suspects have often helped security agencies achieve breakthroughs in important cases.

"With the massive and foreseeable subscriber base of 400 million over the next five years, there is a need for the development of computational approaches using artif
intelligence techniques, biometric devices, crypto analysis, voice recognition technologies, grid surveillance, encryption/decryption and mining databases for security
telecom and data networks and to provide useful inputs to the national security agencies," the DoT has said in the expenditure statement.

Many countries have surveillance laws

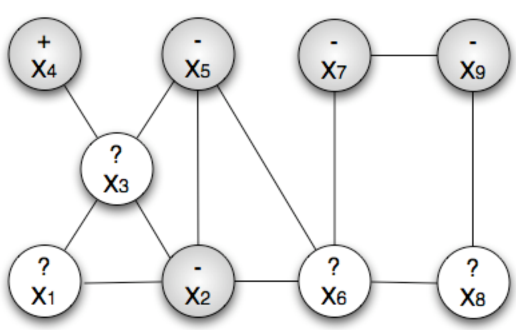
Globally, many countries are enacting surveillance laws which give governments more power to tap the communication systems. For instance, the US recently passe
Protect America Act of 2007', which gives its government sweeping powers to tap any and all electronic and telephonic communication by anyone and anywhere with
even obtaining a court order.

The move raises the issue of invasion of privacy. But the government has categorically made it clear that this software platform was not aimed at snooping into conver
or to carry out any warrant-less tapping programme, but would only be used to create a database that maps every individual's social circle — based on his or her tele
usage — for security reasons.

This security management system will act as a digital law enforcement agency that will be linked to the telecom networks of all service providers. "Information will be
encrypted tunnels and digitally signed to ensure that the integrity of information is preserved," the DoT report added.

So, what's different about networked data?

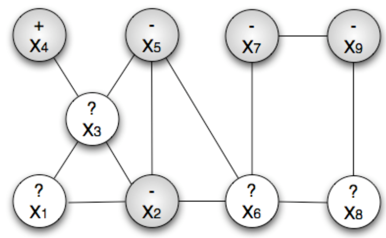
Data graph



Unique characteristics of networked data

Single data graph

- Partially labeled
- Widely varying link structure
- Often heterogeneous object and link types
- From predictive modeling perspective: graph contains both training data and application/testing data



Attribute dependencies

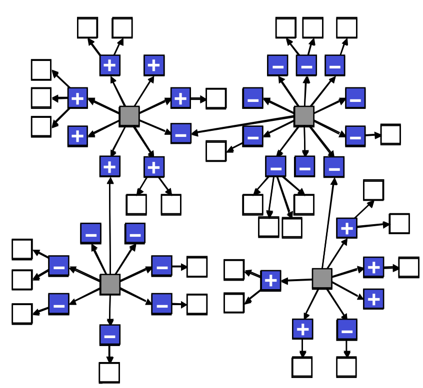
- (Auto)correlation among variables/attributes of linked entities
- Correlations between attribute values and link structure

Suggest key techniques:
guilt-by-association
network features
relational learning
collective inference

Relational autocorrelation

Correlation between the values of the same variable on related objects

- Related instance pairs: $P_R = \{(v_i,v_j) : e_{ik_1}, e_{k_1k_2}, \ldots, e_{k_l j} \in E_R\}$
- Dependence between pairs of values of X: $(x_i, x_j)\ s.t.\ (v_i, v_j) \in P_R$



High autocorrelation

Relational autocorrelation is ubiquitous

Marketing
- Product/service adoption among communicating customers (Domingos & Richardson '01, Hill et al '06)

Advertising
- On-line brand adv. (Provost et al. '09)

Fraud detection
- Fraud status of cellular customers who call common numbers (Fawcett & Provost '97, Cortes et al '01)
- Fraud status of brokers who work at the same branch (Neville & Jensen '05)

Movies
- Box-office receipts of movies made by the same studio (Jensen & Neville '02)

Web
- Topics of hyperlinked web pages (Chakrabarti et al '98, Taskar et al '02)

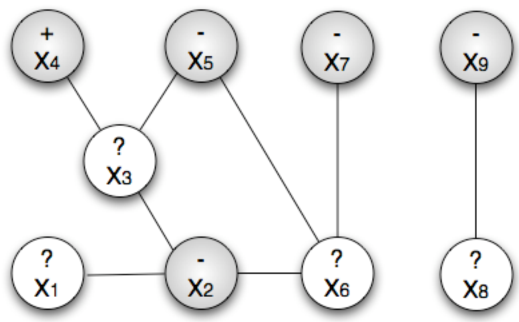
Biology
- Functions of proteins located in together in cells (Neville & Jensen '02)
- Tuberculosis infection among people in close contact (Getoor et al '01)

Business
- Industry categorization of corporations that share common boards members (Neville & Jensen '00)
- Industry categorization of corporations that co-occur in news stories (Bernstein et al '03)

Citation analysis
- Topics of coreferent scientific papers (Taskar et al '01, Neville & Jensen '03)

How can we incorporate autocorrelation into predictive inference?

Disjoint inference (no learning)



Use links to <u>labeled</u> nodes (i.e., guilt by association)

Is guilt-by-association justified theoretically?

Thanks to (McPherson, et al., 2001)

- *Birds of a feather, flock together*
  – attributed to Robert Burton (1577-1640)
- (*People) love those who are like themselves*
  -- Aristotle, *Rhetoric* and *Nichomachean Ethics*
- *Similarity begets friendship*
  -- Plato, *Phaedrus*
- *Hanging out with a bad crowd will get you into trouble*
  -- Foster's Mom

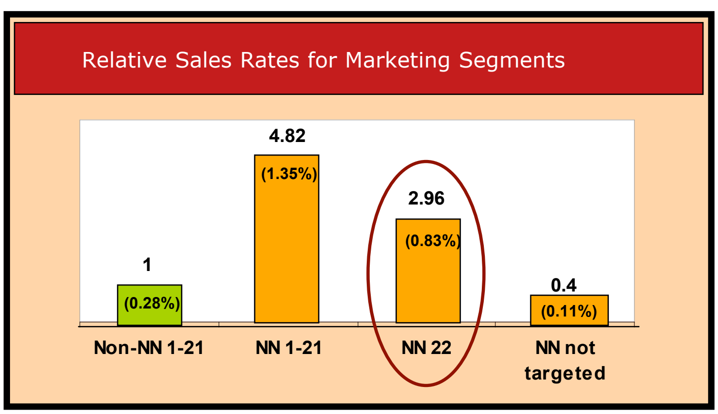
Is guilt-by-association justified theoretically?

Homophily

- fundamental concept underlying social theories
  - (e.g., Blau 1977)
- one of the first features noticed by analysts of social network structure
  - antecedents to SNA research from 1920's (Freeman 1996)
- fundamental basis for links of many types in social networks (McPherson, et al., Annu. Rev. Soc. 2001)
  - Patterns of homophily:
  - remarkably robust across widely varying types of relations
  - tend to get stronger as more relationships exist
- Now being considered in mathematical analysis of networks ("assortativity", e.g., Newman (2003))

Does it apply to non-social networks?

Disjoint inference



(Hill et al. '06)

Example models of network data

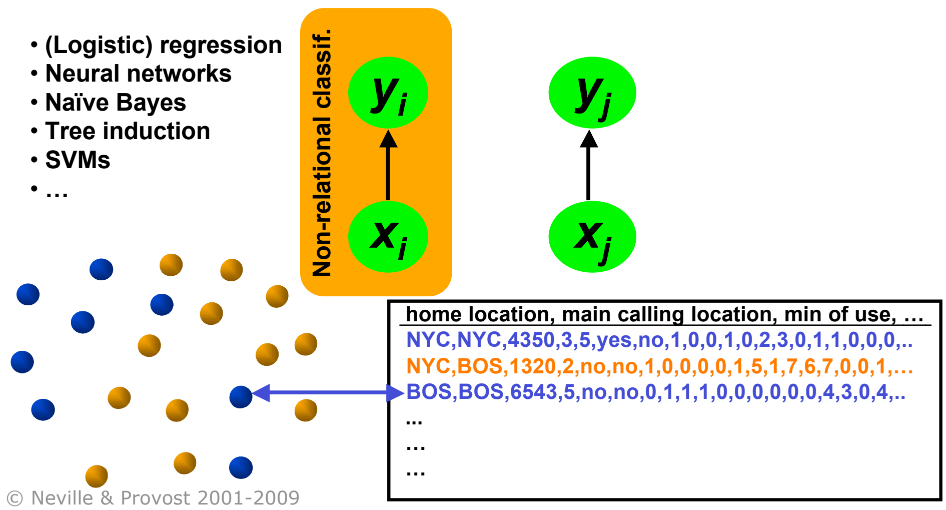
	Disjoint inference	
No learning	Basic NT, wvRN	

What if we add in learning?

Traditional learning and prediction

Methods:

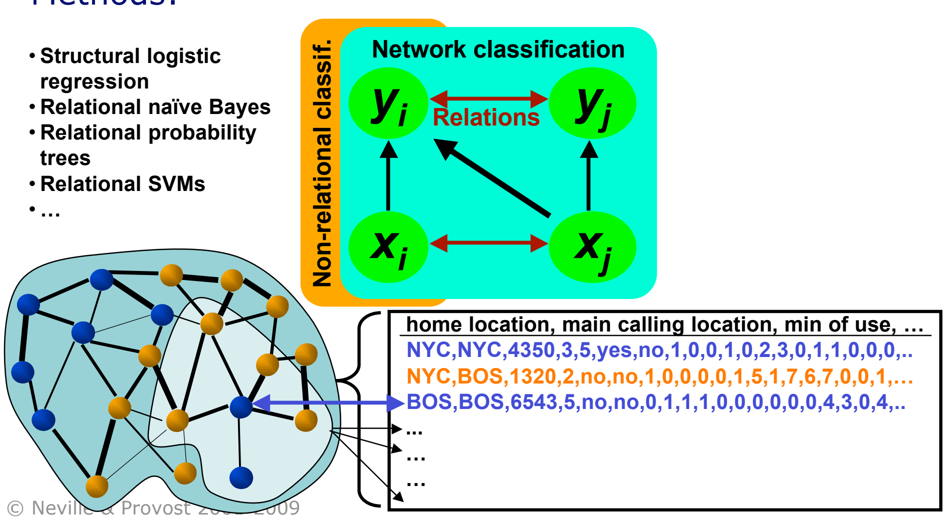
- (Logistic) regression
- Neural networks
- Naïve Bayes
- Tree induction
- SVMs
- …



Network learning and prediction

Methods:

- Structural logistic regression
- Relational naïve Bayes
- Relational probability trees
- Relational SVMs
- …



Relational learning

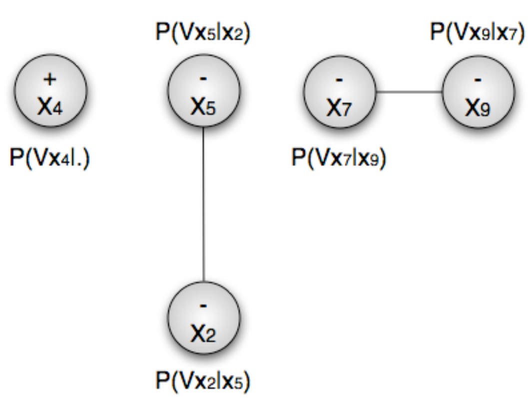
Learning where data cannot be represented as a single relation/table of independently distributed entities, without losing important information

Data may be represented as:
- a multi-table relational database, or
- a heterogeneous, attributed graph, or
- a first-order logic knowledge base

There is a huge literature on relational learning and it would be impossible to do justice to it in the short amount of time we have

For additional information, see:
- Pointers/bibliography on tutorial page
- International Conference on Inductive Logic Programming
- Cussens & Kersting's ICML'04 tutorial: Probabilistic Logic Learning
- Getoor's ICML'06/ECML'07 tutorials: Statistical Relational Learning
- Domingos's KDD'07/ICML'07 tutorials: Statistical Modeling of Relational Data
- Literature review in Macskassy & Provost JMLR'07

Disjoint learning: part I



Create (aggregate) features of (labeled) neighbors

(Perlich & Provost KDD'03) treat aggregation and relational learning feature construction

Social network features can be created for "flat" models

$$\hat{y} = f(\ldots x_G \ldots)$$

where x_G is a (vector of) network-based feature(s)

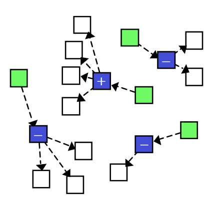
Example applications:

- Fraud detection
  - construct variables representing connection to known fraudulent accounts (Fawcett & Provost '97)
  - or the similarity of immediate network to known fraudulent accounts (Cortes et al. '01; Hill et al. '06b)
- Marketing (Hill et al. '06a)
- On-line Advertising (Provost et al. KDD'09)

Creation of SN features can be (more or less) systematic:
(Popescul & Ungar '03; Perlich & Provost '03,'06; Karamon et al. '07,'08; Gallagher & Eliassi-Rad '08; cf., Gartner '03)

Also: Ideas from hypertext classification extend to SN modeling:
- *hypertext classification has text + graph structure*
- construct variables representing (aggregations of) the classes of linked pages/documents (Chakrabarti et al. '98; Lu & Getoor '03)
- formulate as regularization/kernel combination (e.g., Zhang et al. KDD'06)
- see also (Qi & Davison, 2008)

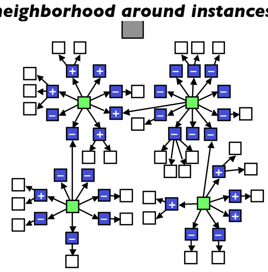
Disjoint learning of relational models



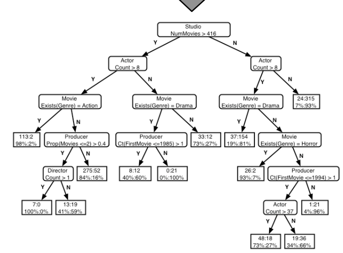
Create aggregate features of relational information

Receipt >$30M	Actor Count	Actor Award	Studio Location	Max Produc Exp
+	11	Y	CA	5yr
+	42	Y	NY	25yr
-	19	N	CA	13yr
...	...	...	...	...

Consider local relational neighborhood around instances



Learn (adapted) flat model



Example

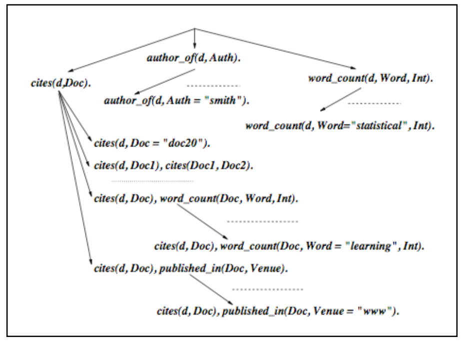
Structural logistic regression *(Popescul et al. '03)*

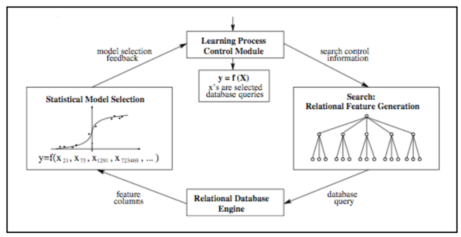
Features

- Based on boolean first-order logic features used in inductive logic programming
- Top-down search of refinement graph
- Includes additional database aggregators that result in scalar values (e.g. count, max)

Model

- Logistic regression
- Two-phase feature selection process with AIC/BIC





Example

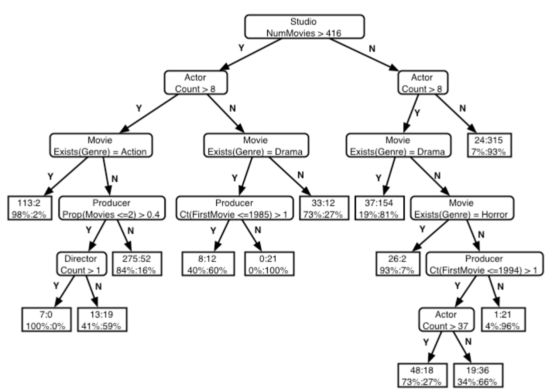
Relational probability trees *(Neville et al. '03)*

Features

- Uses set of aggregators to construct features (e.g., Size, Average, Count, Proportion)
- Exhaustive search within a user-defined space (e.g., <3 links away)

Model

- Decision trees with probability estimates at leaves
- Pre-pruning based on chi-square feature scores
- Randomization tests for accurate feature selection (more on this later)



Recall the network marketing example...

Learning patterns among labeled nodes

Features can be constructed that represent "guilt" of a node's neighbors: $\hat{y} = f(\ldots x_G \ldots)$

where x_G is a (vector of) network-based feature(s)

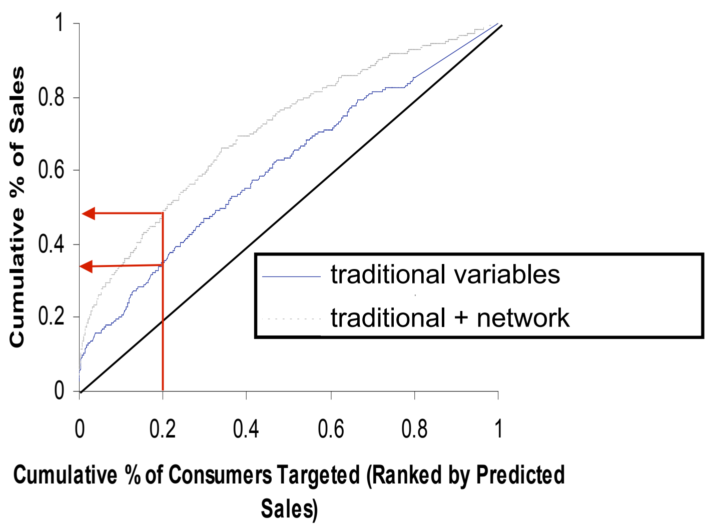
Example application:

Marketing (Hill et al. '06a)

Network features that model known customers

Attribute	Description
Degree	Number of unique customers communicated with before the mailer
# Transactions	Number of transactions to/from customers before the mailer
Seconds of communication	Number of seconds communicated with customers before mailer
Connected to influencer ?	Is an influencer in your local neighborhood?
Connected component s ize	Size of the connected component target belongs to.
Similarity (structural equivalence)	Max overlap in local neighborhood with existing customer

Lift in sales with network-based features



Similar results for predicting customer attrition/churn

Thanks to KXEN

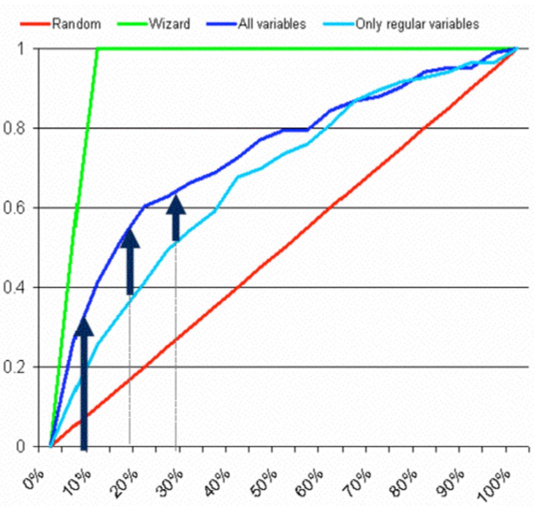
Two models

1. Regular var only
2. All variables

Social Network variables increase lift

… a lot !

- Overall : 40%
- First decile : 67%
- Second decile : 47%
- Third decile : 22%



see also (Dasgupta et al. EDBT'08) & (Birke '08) on social networks & telecom churn

Similar results for predicting customer attrition

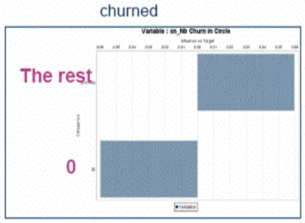
Thanks to KXEN

In model with all variables

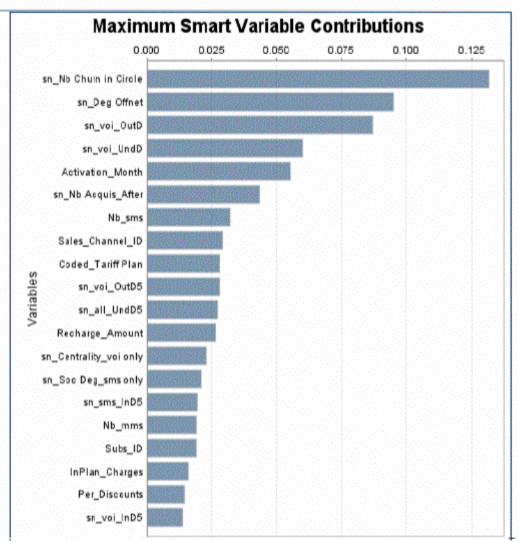
- Most significant variable

sn_Nb Churn in Circle

  - Number of « friends » who churned



- SN variables
  - Are among most significant
  - Account for 69% of information
- Regular variables
  - Are much less significant



Avenues for marketing

Simple targetting

☺ Uses « regular » variables only
- No need to compute more variables

☺ Builds predictive model
- Provide prediction of expected results

☹ Does not use social network variables
- Fail to exploit implicit information

Networked targetting

☺ Uses both regular and social variables
- Fully exploits all available information

☺ Builds predictive model
- Provides prediction of expected results
- Targets according to model
- Exploits mechanisms of social behavior

☹ Uses social network variables
- Needs to compute network variables

Viral marketing

☺ Uses « social network » variables only
- Fully exploits network information

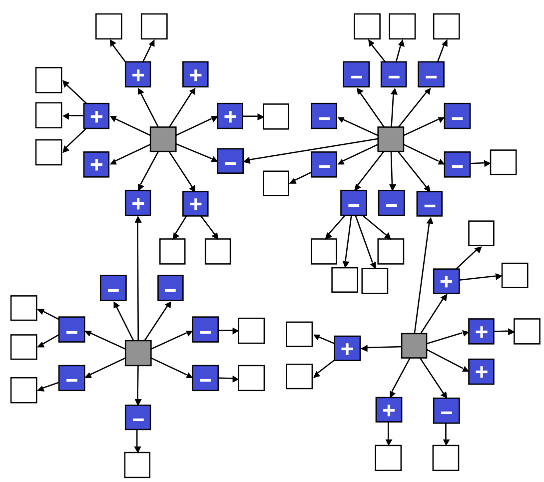
☺ Targets « influencers »
- Exploits mechanisms of social behavior
  - Word-of-mouth
  - Guilt-by-association
- Target is small

☹ Uses social network variables only
- Needs to compute network variables
- Fails to use regular variables

☹ Is not predictive
- Does not provide prediction of expected results

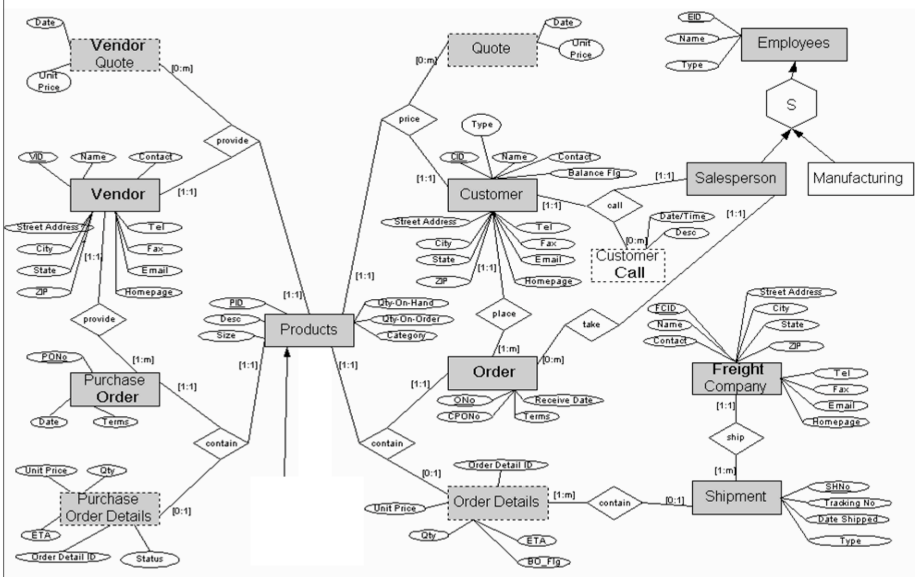
8

Disjoint learning: part II



Use node <u>identifiers</u> to create features
→ connections to specific individuals can be telling

Side note: not just for networked data – IDs can be useful for modeling any data in a multi-table RDB



Towards a theory of aggregation *(Perlich & Provost MLJ'06)*: A (recursive) Bayesian perspective

Traditional (naïve) Bayesian Classification:

$P(c|X)=P(X|c)*P(c)/P(X)$ Bayes' Rule

$P(X|c)= \prod_i P(x_i|c)$ Assuming conditional independence

$P(x_i|c)$ & $P(c)$ Estimated from the training data

Linked Data:

x_i might be an object identifier (e.g. SSN) => $P(x_i|c)$ cannot be estimated
Let Ω_i be a set of k objects linked to x_i => $P(x_i|c) \sim P(\text{linked-to-}\Omega_i|c)$

$P(\Omega_i|c) \sim \prod_{O\in\Omega} P(O|c)$ Assume O is drawn independently

$P(\Omega_i|c) \sim \prod_{O\in\Omega} (\prod_j P(o_j|c))$ Assuming conditional independence

How to incorporate identifiers of related objects (in a nutshell)

1. Estimate from known data:
   - *class-conditional distributions* of related identifiers (say D^+ & D^-)
   - can be done, for example, assuming class-conditional independence in analogy to Naïve Bayes
   - save these as "meta-data" for use with particular cases
2. Any particular <u>case</u> C has its own "distribution" of related identifiers (say D_c)
3. Create features
   - $\delta(D_c,D^+)$, $\delta(D_c, D^-)$, $(\delta(D_c, D^+) - \delta(D_c, D^-))$
   - where δ is a distance metric between distributions
4. Add these features to target-node description(s) for learning/estimation

<u>Main idea</u>:

"Is the distribution of nodes to which this case is linked similar to that of a <whatever>?"

Density estimation for aggregation

1: Class-conditional distributions

Distr.	A	B
$D_{Class\ 1}$	0.75	0.25
$D_{Class\ 0}$	0.2	0.8

CID	*Class*
C1	0
C2	1
C3	1
C4	0

CID	*id*
C1	B
C2	A
C2	A
C2	B
C3	A
C4	B
C4	B
C4	B
C4	A

?

2: Case linkage distributions:

D_c	A	B
C1	0	1
C2	0.66	0.33
C3	1	0
C4	0.25	0.75

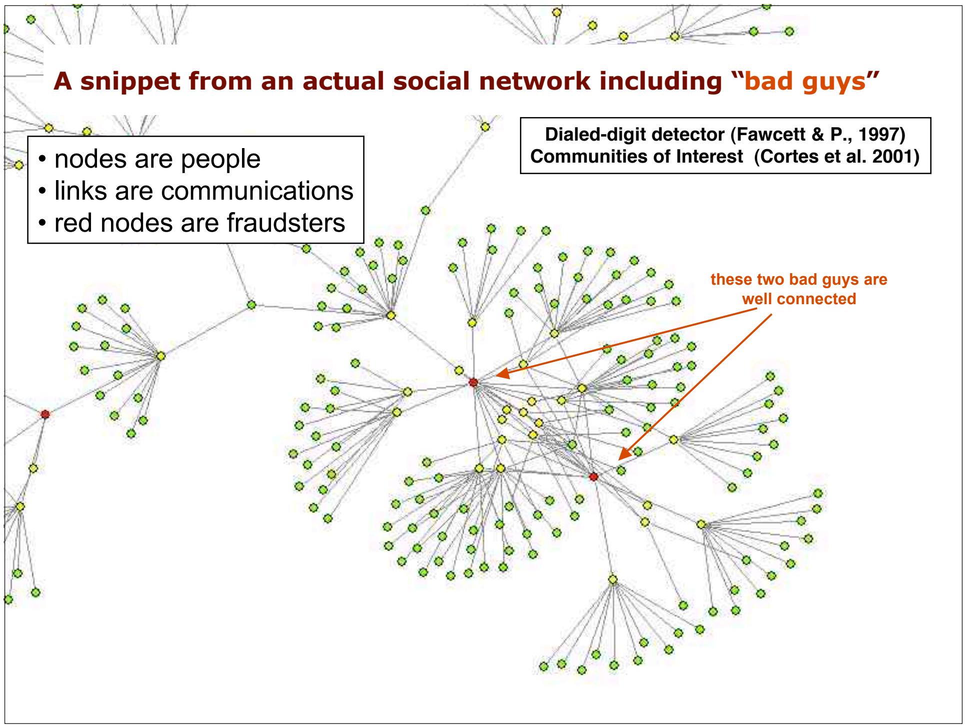
3: L2 distances for C1:

L2(C1, $D_{Class\ 1}$) = 1.125
L2(C1, $D_{Class\ 0}$) = 0.08

4: Extended feature vector:

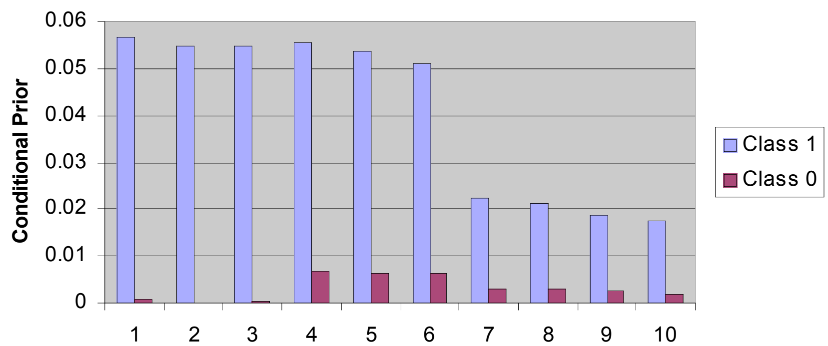
CID	$L2_1$	$L2_0$	$L2_1 - L2_0$	*Class*
C1	1.125	0.08	-1.045	0
C2	0.014	0.435	0.421	1
C3	0.125	1.28	1.155	1
C4	0.5	0.005	-0.495	0

A snippet from an actual social network including "bad guys"



Classify buyers of most-common title from a Korean E-Book retailer

Estimate whether or not customer will purchase the most-popular e-book: Accuracy=0.98 (AUC=0.96)



Class-conditional distributions across identifiers of 10 other popular books

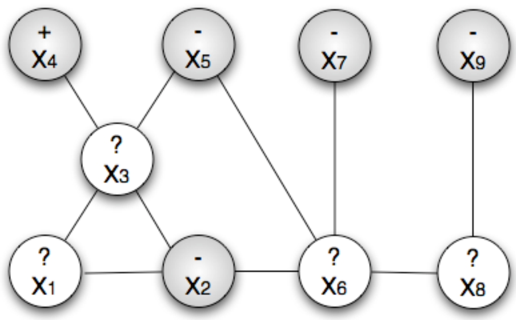
Watch for more results later

Models of network data

	Disjoint inference	
No learning	Basic NT, wvRN	
Disjoint learning	NT, ACORA, RBC, RPT, SLR	

An important, unique characteristic of networked data: one can perform *collective inference* across individuals

Collective inference

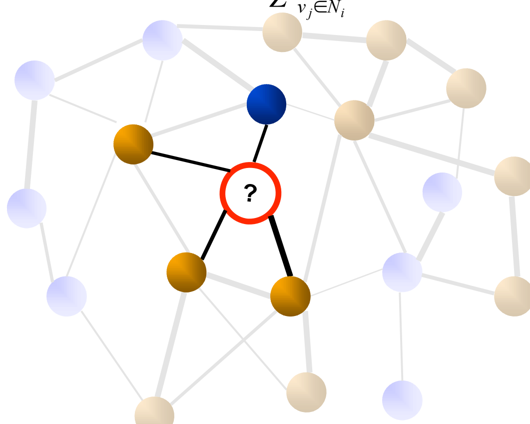


Use links among <u>unlabeled</u> nodes

Collective inference models

A particularly simple guilt-by-association model is that a value's probability is the average of its probabilities at the neighboring nodes

$$p(y_i = c \mid N_i) = \frac{1}{Z} \sum_{v_j \in N_i} w_{i,j} \cdot p(y_j = c \mid N_j)$$



- *Gaussian random field* (Besag 1975; Zhu et al. 2003)
- *"Relational neighbor" classifier - wvRN* (Macskassy & P. 2003)

Model partially-labeled network with a random field

Treat network as a random field
- a probability measure over a set of random variables $\{X_1, \ldots, X_n\}$ that gives non-zero probability to any configuration of values for all the variables.

Convenient for modeling network data:
- A Markov random field satisfies

$$p(X_i = x_i | X_j = x_j, i \neq j) = p(X_i = x_i | N_i)$$

- where N_i is the set of neighbors of X_i under some definition of neighbor.
- in other words, the probability of a variable taking on a value depends only on its neighbors
- probability of a configuration x of values for variables X the normalized product of the "potentials" of the states of the k maximal cliques in the network: $P(X = x) = \frac{1}{Z}\prod_k \phi_k(x_{(k)})$

(Dobrushin, 1968; Besag, 1974; Geman and Geman, 1984)

Markov random fields

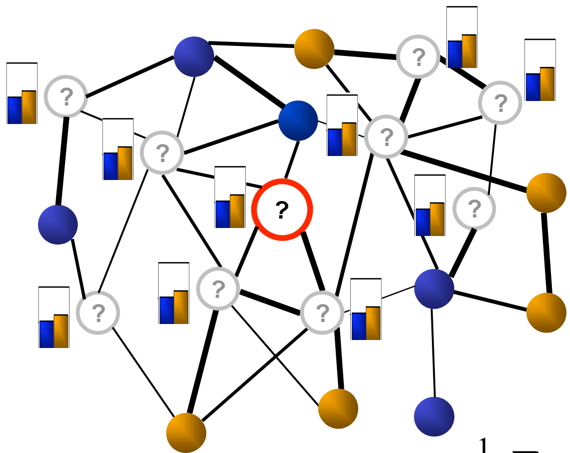
Random fields have a long history for modeling regular grid data
- in statistical physics, spatial statistics, image analysis
- see Besag (1974)

Besag (1975) applied such methods to what we would call networked data ("non-lattice data")

Some notable contemporary example applications:
- web-page classification (Chakrabarti et al. 1998)
- viral marketing (Domingos & Richardson 2001, R&D 2002)
- eBay auction fraud (Pandit et al. 2007)

Collective inference cartoon

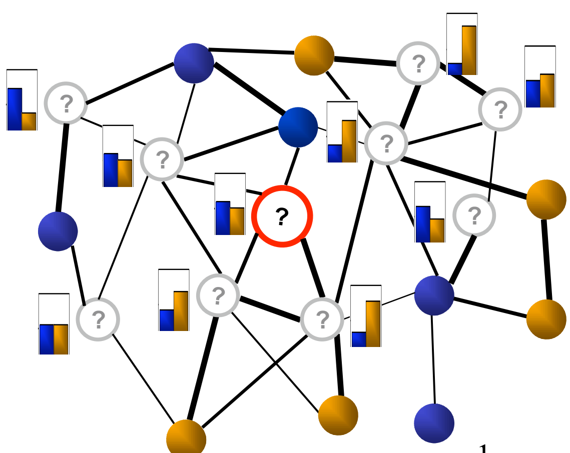
relaxation labeling – repeatedly estimate class distributions on all unknowns, based on current estimates



$$p(y_i = c \mid N_i) = \frac{1}{Z} \sum_{v_j \in N_i} w_{i,j} \cdot p(y_j = c \mid N_j)$$

Collective inference cartoon

relaxation labeling – repeatedly estimate class distributions on all unknowns, based on current estimates

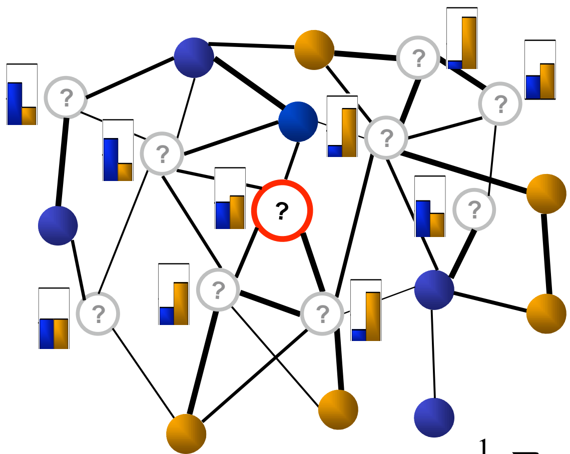


$$p(y_i = c \mid N_i) = \frac{1}{Z} \sum_{v_j \in N_i} w_{i,j} \cdot p(y_j = c \mid N_j)$$

© Neville & Provost 2001-2009

Collective inference cartoon

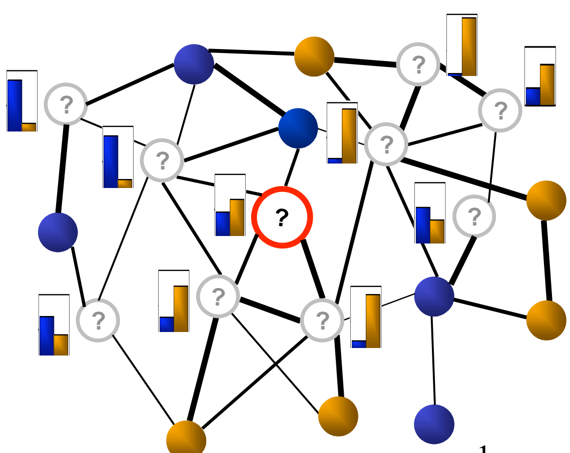
relaxation labeling – repeatedly estimate class distributions on all unknowns, based on current estimates



$$p(y_i = c \mid N_i) = \frac{1}{Z} \sum_{v_j \in N_i} w_{i,j} \cdot p(y_j = c \mid N_j)$$

Collective inference cartoon

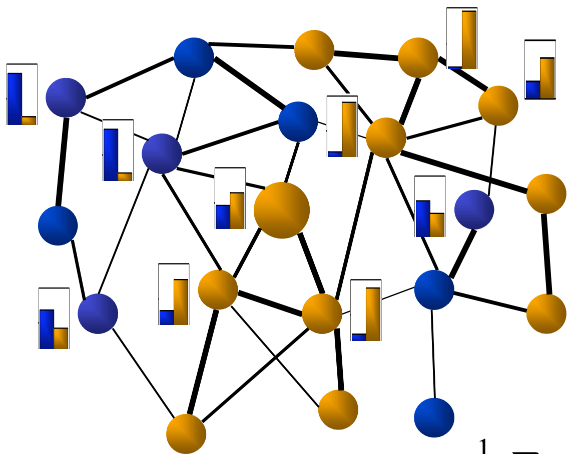
relaxation labeling – repeatedly estimate class distributions on all unknowns, based on current estimates



$$p(y_i = c \mid N_i) = \frac{1}{Z} \sum_{v_j \in N_i} w_{i,j} \cdot p(y_j = c \mid N_j)$$

Collective inference cartoon

relaxation labeling – repeatedly estimate class distributions on all unknowns, based on current estimates



$$p(y_i = c \mid N_i) = \frac{1}{Z} \sum_{v_j \in N_i} w_{i,j} \cdot p(y_j = c \mid N_j)$$

Various techniques for collective inference
(see also Jensen et al. KDD'04)

- MCMC, e.g., Gibbs sampling (Geman & Geman 1984)
- Iterative classification (Besag 1986; …)
- Relaxation labeling (Rosenfeld et al. 1976; …)
- Loopy belief propagation (Pearl 1988)
- Graph-cut methods (Greig et al. 1989; …)

Either:
- estimate the maximum a posteriori joint assignment to/distribution of all free parameters

or
- estimate the marginal distributions of some or all free parameters simultaneously (or some related likelihood-based scoring)

or
- just perform a heuristic procedure to reach a consistent state.

Models of network data

	Disjoint inference	Collective inference
No learning	Basic NT, wvRN	Random fields (Gaussian, Markov), wvRN
Disjoint learning	NT, ACORA, RBC, RPT, SLR	

Using wvRN/GRF and collective inference, we can ask:

How much "information" is in the network structure alone?

Network classification case study

12 data sets from 4 domains (previously used in ML research)
- IMDB (Internet Movie Database) (e.g., Jensen & Neville, 2002)
- Cora (e.g., Taskar et al., 2001) [McCallum et al., 2000]
- WebKB [Craven et al., 1998]
  - CS Depts of Texas, Wisconsin, Washington, Cornell
  - multiclass & binary (student page)
  - "cocitation" links
- Industry Classification [Bernstein et al., 2003]
  - yahoo data, prnewswire data

Homogeneous nodes & links
- one type, different classes/subtypes

Univariate classification
- only information: structure of network and (some) class labels
- guilt-by-association (wvRN) with collective inference
- plus several models
- that "learn" relational patterns

Macskassy, S. and F. P. "Classification in Networked Data: A toolkit and a univariate case study." *Journal of Machine Learning Research* 2007.

Local models to use for collective inference

(see Macskassy & Provost JMLR'07)

network-only Bayesian classifier nBC
- inspired by (Charabarti et al. 1998)
- multinomial naïve Bayes on the neighboring class labels

network-only link-based classifier
- inspired by (Lu & Getoor 2003)
- logistic regression based on a node's "distribution" of neighboring class labels, DN(vi) (multinomial over classes)

relational-neighbor classifier (weighted voting)
- (Macskassy & Provost 2003, 2007)

$$p(y_i = c \mid N_i) = \frac{1}{Z} \sum_{v_j \in N_i} w_{i,j} \cdot p(y_j = c \mid N_j)$$

relational-neighbor classifier (class distribution)
- Inspired by (Perlich & Provost 2003)

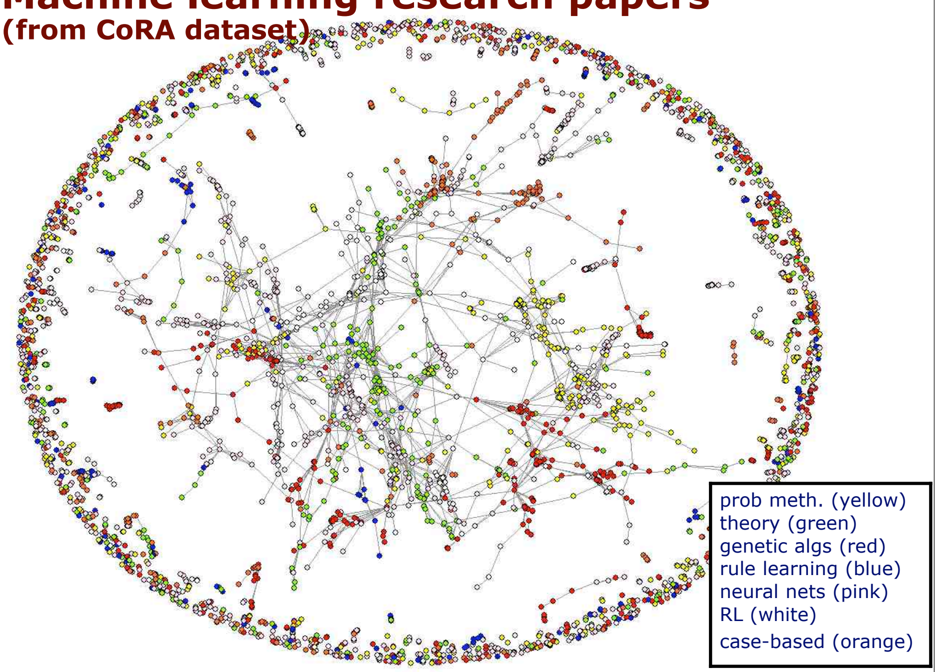
$$p(y_i = c \mid N_i) = sim(D_N(v_i), Dist(c))$$

How much information is in the network structure?

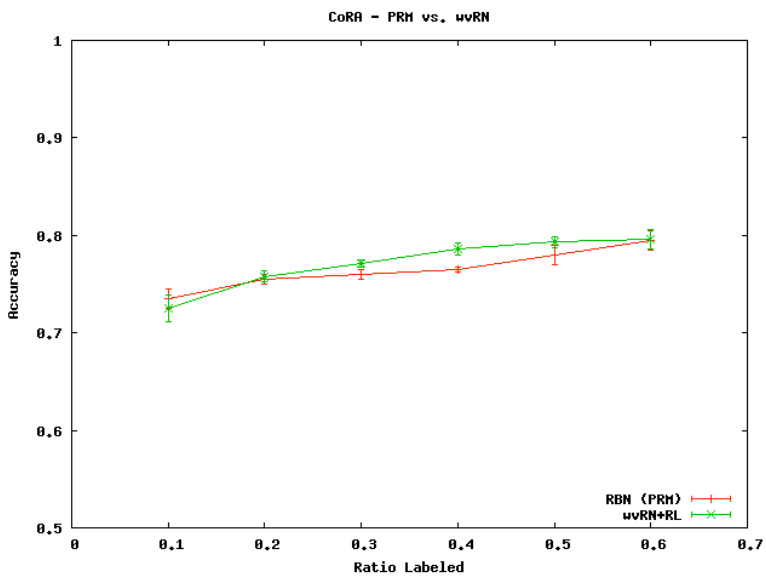
Data set	Accuracy	Relative error reduction over default prediction
wisconsin-student	0.94	86%
texas-student	0.93	86%
Cora	0.87	81%
wisconsin-multi	0.82	67%
cornell-student	0.85	65%
imdb	0.83	65%
wash-student	0.85	58%
wash-multi	0.71	52%
texas-multi	0.74	50%
industry-yahoo	0.64	49%
cornell-multi	0.68	45%
industry-pr	0.54	36%

- Labeling 90% of nodes
- Classifying remaining 10%
- Averaging over 10 runs

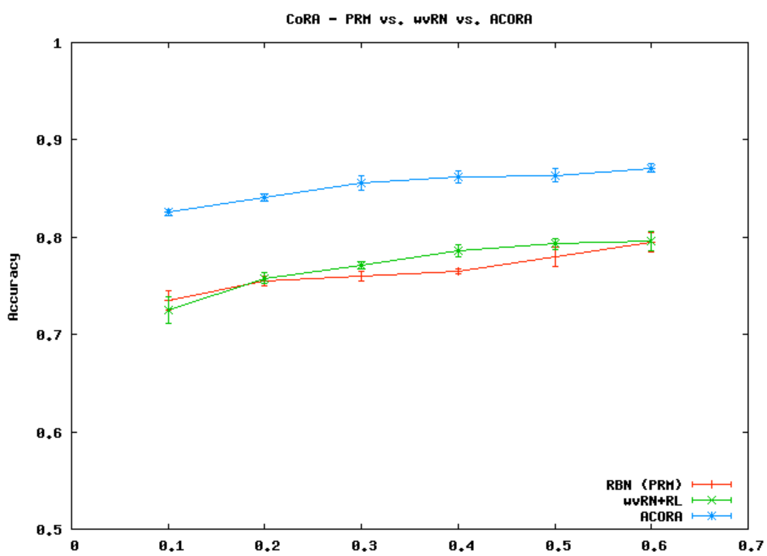
Machine learning research papers
(from CoRA dataset)



RBN vs wvRN *(Macskassy & Provost '07)*



Using identifiers *(Perlich & Provost '06)*



(compare: Hill & P. "The Myth of the Double-Blind Review", 2003)

Characteristics of network data

Single data graph

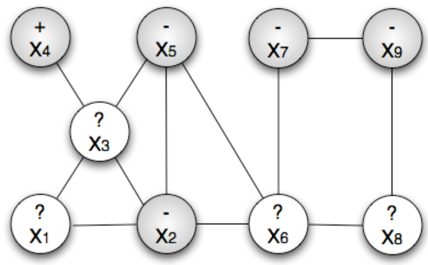
√ Partially labeled

- Widely varying link structure
- Often heterogeneous object and link types

Attribute dependencies

√ Homophily, autocorrelation among class labels

- Correlation among attributes of related entities
- Correlations between attribute values and link structure



Networks ≠ graphs?

Networked data can be much more complex than just sets of (labeled) vertices and edges.

- Vertices and edges can be heterogeneous
- Vertices and edges can have various attribute information associated with them

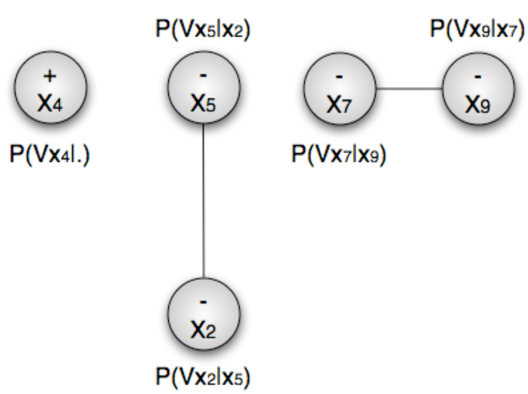
Various methods for learning statistical models that take advantage of attribute dependencies in relational data

- Probabilistic relational models (RBNs, RMNs, AMNs, RDNs, …)
- Probabilistic logic models (BLPs, MLNs, …)

Models of network data

	Disjoint inference	Collective inference
No learning	wvRN	Gaussian random fields, MRFs, wvRN
Disjoint learning	ACORA, RBC, RPT, SLR	MLN, RBN, RDN, RMN

Disjoint learning: part III



Assume training data are fully labeled (i.e., ignore missing labels) & model dependencies among linked entities

Relational learning

Let's consider briefly three approaches

- Model with inductive logic programming (ILP)
- Model with probabilistic relational model (graphical model+RDB)
- Model with probabilistic logic model (ILP+probabilities)

First-order logic modeling

The field of Inductive Logic Programming has extensively studied modeling data in first-order logic

Although it has been changing, traditionally ILP did not focus on representing uncertainty

- in the usual use of first-order logic, *each ground atom either is true or is not true* (cf., a Herbrand interpretation)

…one of the reasons for the modern rubric "statistical relational learning"

First-order logic for statistical modeling of network data?

- a strength is its ability to represent and facilitate the search for complex and deep patterns in the network
- a weakness is its relative lack of support for aggregations across nodes (beyond existence)
- more on this in a minute…

Network data in first-order logic

broker(Amit), broker(Bill), broker(Candice), …

works_for(Amit, Bigbank), works_for(Bill, E_broker), works_for(Candice, Bigbank), …

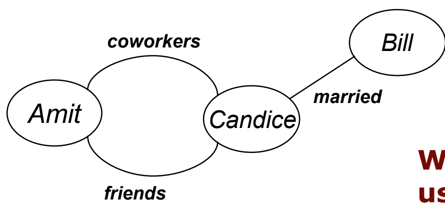
married(Candice, Bill)

smokes(Amit), smokes(Candice), …

works_for(X,F) & works_for(Y,F) -> coworkers(X,Y)

smokes(X) & smokes(Y) & coworkers(X,Y) -> friends(X,Y)

…



What's the problem with using FOL for our task?

Probabilistic graphical models

Probabilistic graphical models (PGMs) are convenient methods for representing probability distributions across a set of variables.

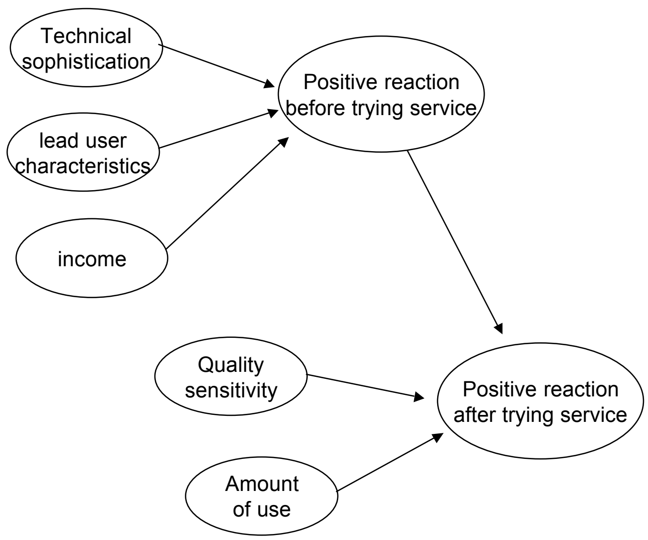
- Bayesian networks (BNs), Markov networks (MNs), Dependency networks (DNs)
- See Pearl (1988), Heckerman et al. (2000)

Typically BNs, MNs, DNs are used to represent a set of random variables describing independent instances.

- For example, the probabilistic dependencies among the descriptive features of a consumer—the same for different consumers

Example

A Bayesian network modeling consumer reaction to new service



Probabilistic relational models

The term "relational" recently has been used to distinguish the use of probabilistic graphical models to represent variables across a set of dependent, multivariate instances.

These methods model the full joint distribution over the attribute values in a network using a probabilistic graphical model (e.g., BN, MN)

- For example, the dependencies between the descriptive features of friends in a social network
- We saw a "relational" Markov network earlier when we discussed Markov random fields for univariate network data
  - although the usage is not consistent, "Markov random field" often is used for a MN over multiple instances of the "same" variable

In these *probabilistic relational models*, there are dependencies within instances and dependencies among instances

Key ideas for modeling network data:

- Learn from a single network by tying parameters across instances of same type
- Use aggregations to deal with heterogeneous network structure

Modeling the joint "network" distribution

- Relational Bayesian networks
  - Extend Bayes nets to network settings (Friedman et al. '99, Getoor et al. '01)
  - Efficient closed form parameter estimation, but acyclicity constraint limits representation of autocorrelation dependencies and makes application of guilt-by-association techniques difficult
- Relational Markov networks
  - Extension of Markov networks (Taskar et al '02)
  - No acyclicity constraint but feature selection is computationally intensive because parameter estimation requires approximate inference
  - Associative Markov networks are a restricted version designed for guilt-by-association settings, for which there are efficient inference algorithms (Taskar et al. '04)
- Relational dependency networks
  - Extension of dependency networks (Neville & Jensen '04)
  - No acyclicity constraint, efficient feature selection, but model is an approximation of the full joint and accuracy depends on size of training set

Example:
Can we estimate the likelihood that a stock broker is/will be engaged in activity that violates securities regulations?

newswise

Username:
Password:
Login
forgotten login
how to register

ABOUT NEWSWISE
Overview of Services
Media Subscribers
Source Institutions
What's New
Contact Us

LIBRARIES
Latest News
SciNews
MedNews
LifeNews
BizNews
Video/Audio
RSS Feeds
Search

CHANNELS
Breaking News
Features

RESOURCES
Expert Finder Tools
Contact Directory
Meetings Calendars
Awards for Journalists

Released: Thu 13-Oct-2005, 08:00 ET
Printer-friendly Version

Securities Fraud Targeted by New Computing Tool

Libraries
Business News

Keywords
SECURITIES FRAUD COMPUTER SCIENCE BROKERS

Contact Information
Available for logged-in reporters only

Description
The world's largest private-sector securities regulator, the National Association of Securities Dealers, has teamed with computer scientists to create a new tool for the world of securities fraud. By developing statistical models that assess data that most models can't manage, the scientists aim to help the NASD discover misconduct among brokers.

Newswise — The world's largest private-sector securities regulator, the National Association of Securities Dealers, has teamed with University of Massachusetts Amherst researchers to bring cutting-edge computer science to the world of securities fraud. By developing statistical models that assess data that most models can't manage, the scientists aim to help the NASD discover misconduct among brokers and concentrate regulatory attention on those who are most likely to misbehave.

Because broker malfeasance is often encouraged by the presence of those conspiring to commit fraud themselves, the researchers were given the task of developing statistical models that made use of this social aspect of rule-breaking. Such "relational" data is difficult for many models, which often assume independence among records.

David Jensen, computer science, likens the task to modeling medical diagnostics. When trying to predict the probability that an individual will catch a disease, information intrinsic to the individual—such as age or health history—can be critical. But clues can also be extracted from information about the person's social and professional network, such as where they've lived or worked, or with whom they've been in contact.

"Our methods are uniquely suited to analyze this kind of information," says Jensen. "They allow you to easily look at the characteristics of the surrounding network."

The work is part of an ongoing, joint project exploring fraud detection by UMass Amherst researchers and the NASD, and it was presented recently by doctoral student Jennifer Neville at the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

More than 600,000 brokers are engaged in securities transactions, making NASD examiners a valuable and finite resource. While these human examiners have the acuity to spot relational patterns that suggest a broker warrants further scrutiny, automating that sort of evaluation had proved difficult. But the relational probability trees (RPTs) developed by Neville and Jensen appear to make good use of this contextual information and they provide a ranking of risky brokers to boot.

Using data from past years supplied by the NASD, Jensen, Neville and doctoralstudent Ozgur Simsek applied their algorithms to the networks of organizational relationships in the securities world. For example, brokers are linked to the firms they work for, customer complaints are linked to the brokers they reference, and branches are linked to their parent firms. By analyzing records of brokers in the context of other records in their "neighborhood" the algorithms were able to predict which brokers would commit violations with surprising accuracy, says Jensen.

Detecting "bad brokers" for NASD

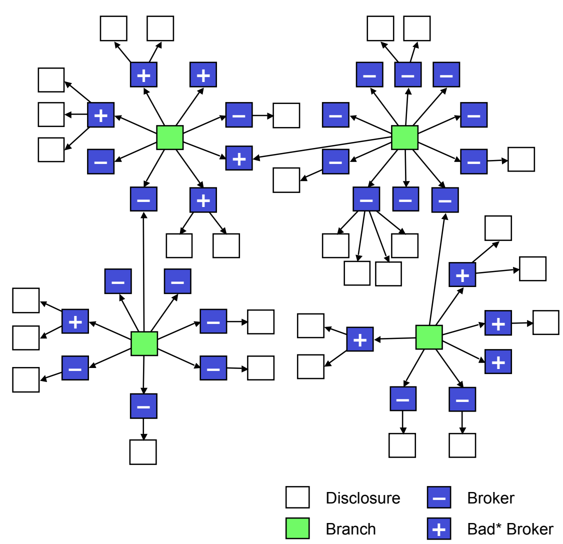
(Neville et al. KDD'05)

NASD (now FINRA) is the largest private-sector securities regulator

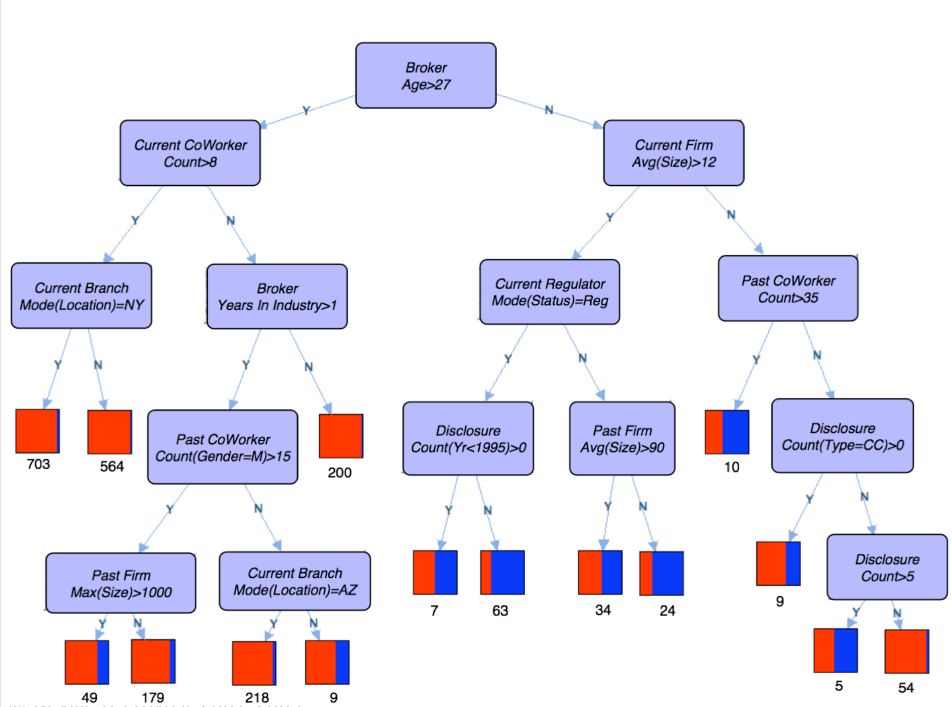
NASD's mission includes preventing and discovering misconduct among brokers (e.g., fraud)

Current approach: Hand-crafted rules that target brokers with a history of misconduct (HRB)

Task: Use relational learning techniques to automatically identify brokers likely to engage in misconduct based on network patterns

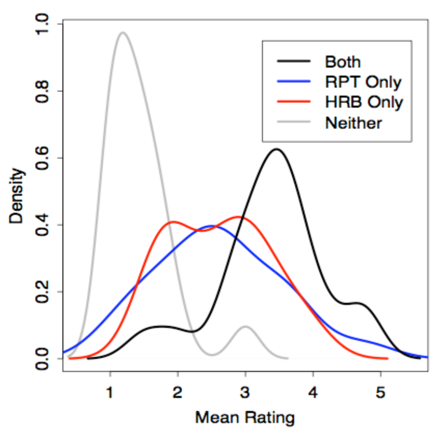


*"Bad" = having violated securities regulations



RPT identified additional brokers to target

(Neville et al. KDD'05)



"One broker I was highly confident in ranking as 5…

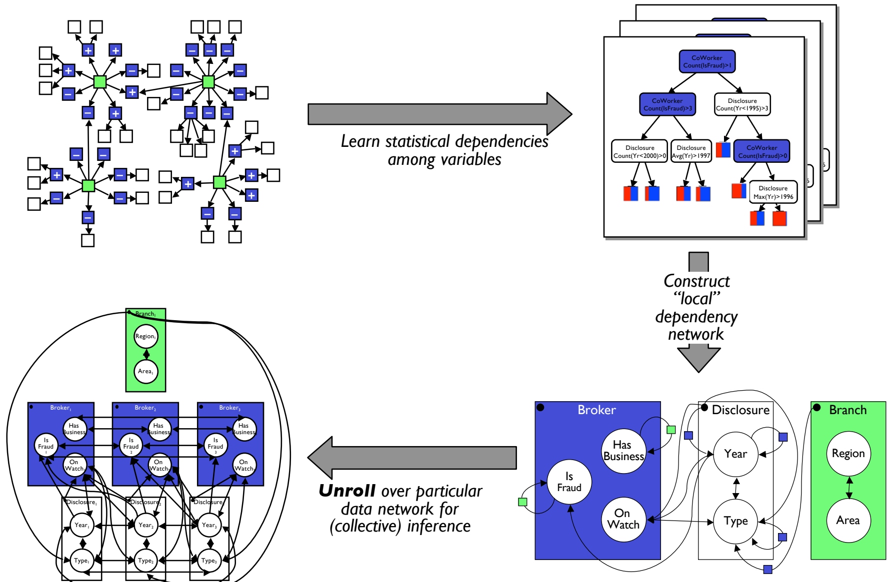
Not only did I have the pleasure of meeting him at a shady warehouse location, I also negotiated his bar from the industry...

This person actually used investors' funds to pay for personal expenses including his trip to attend a NASD compliance conference!

…If the model predicted this person, it would be right on target."

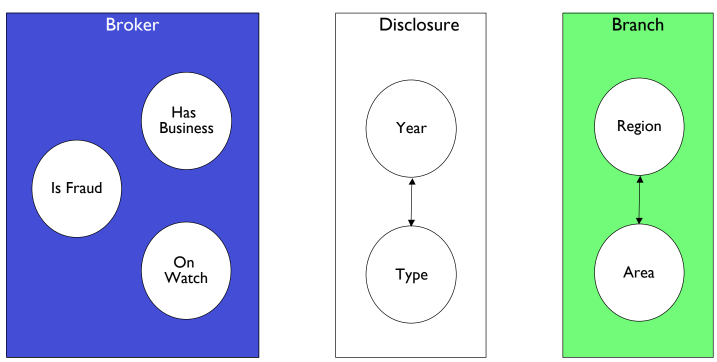
Informal examiner feedback

Learning a relational dependency network for the bad broker problem *(Neville & Jensen JMLR'07)*



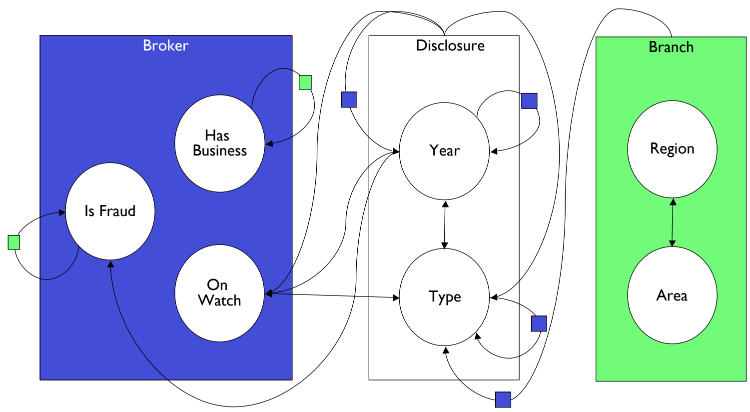
Data on brokers, branches, disclosures

(heterogeneous network)



Learned RDN for broker variables

(Neville & Jensen JMLR'07)



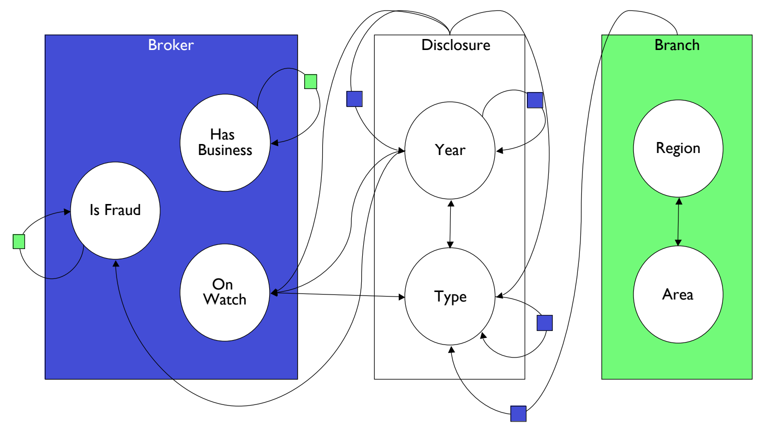
note: needs to be "unrolled" across network

Important concept!

The network of statistical dependencies does not necessarily correspond to the data network

Example on next three slides…

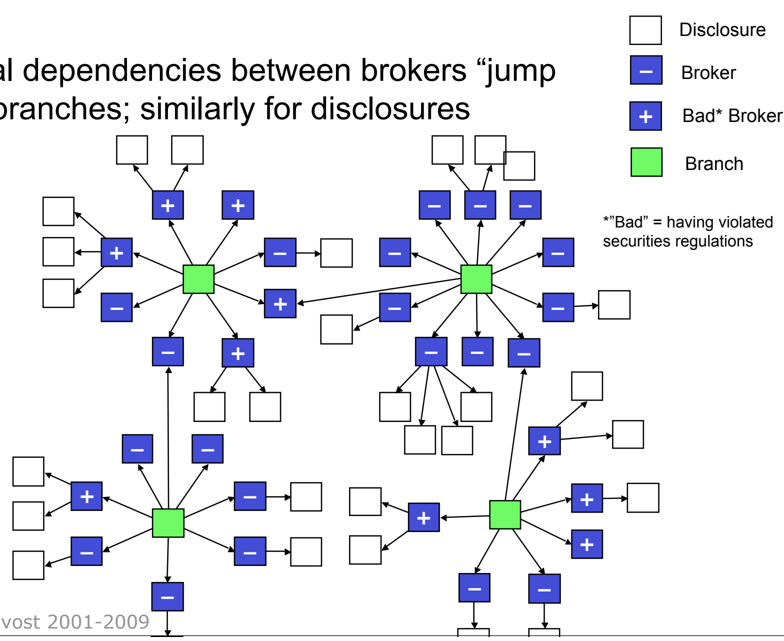
Recall: broker dependency network



note: this dependency network needs to be "unrolled" across the data network

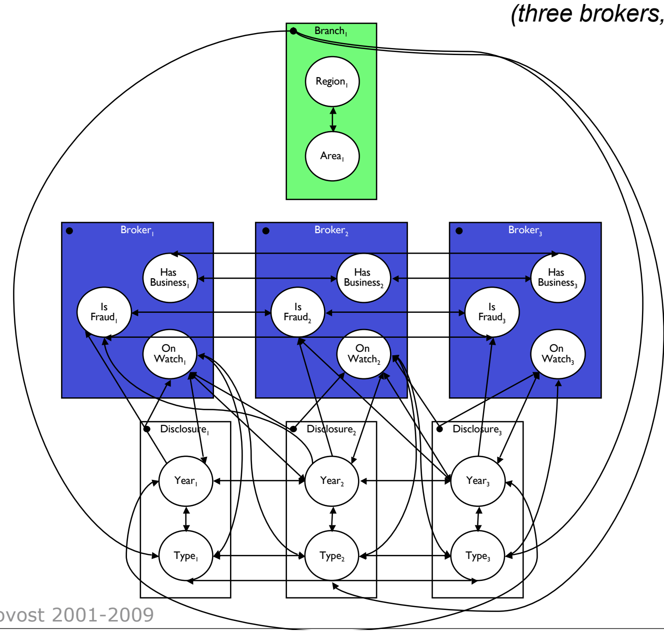
Broker data network

Statistical dependencies between brokers "jump across" branches; similarly for disclosures

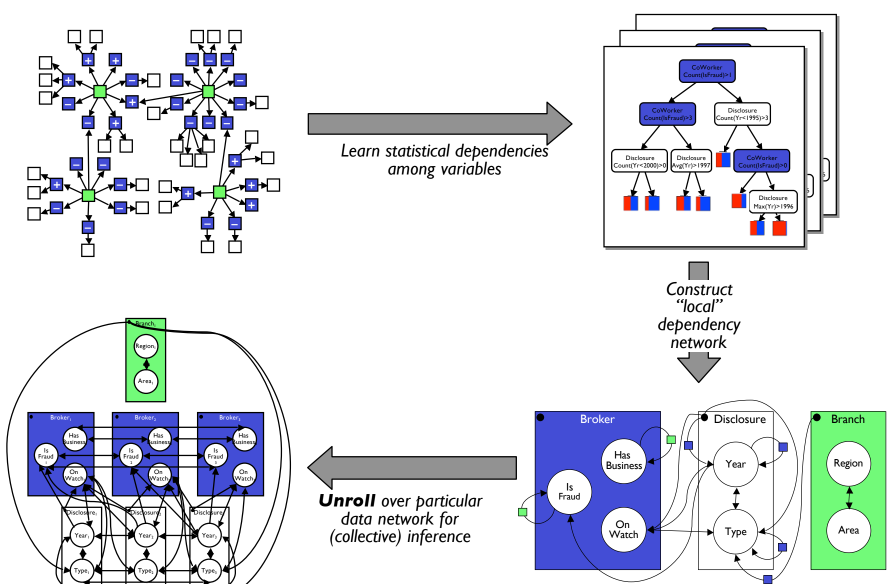


Model unrolled on (tiny) data network

(three brokers, one branch)



Putting it all together:
Relational dependency networks



Combining first-order logic and probabilistic graphical models

Recently there have been efforts to combine FOL and probabilistic graphical models

- e.g., Bayesian logic programs (Kersting and de Raedt '01), Markov logic networks (Richardson & Domingos MLJ'06)
- and see discussion & citations in (Richardson & Domingos '06)

For example: Markov logic networks

- A template for constructing Markov networks
  - Therefore, a model of the joint distribution over a set of variables
- A first-order knowledge base with a weight for each formula

Advantages:

- Markov network gives sound probabilistic foundation
- First-order logic allows compact representation of large networks and a wide variety of domain background knowledge

Markov logic networks

(Richardson & Domingos MLJ'06)

A Markov Logic Network (MLN) is a set of pairs (F, w):

- F is a formula in FOL
- w is a real number

Together with a finite set of constants, it defines a Markov network with:

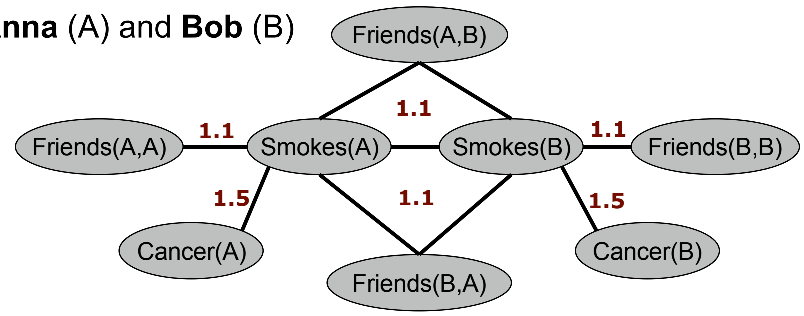
- One node for each grounding of each predicate in the MLN
- One feature for each grounding of each formula F in the MLN, with its corresponding weight w

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x,y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$

See Domingos' KDD'07 tutorial Statistical Modeling of Relational Data for more details

MLN details

Two constants:
Anna (A) and **Bob** (B)



1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x,y\ Friends(x,y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$

$$P(x) = \frac{1}{Z}\exp\left(\sum_i w_i n_i(x)\right)$$

w_i: weight of formula i
$n_i(x)$: # true groundings of formula i in x

Recall our network-based marketing example?

➔ collective inference can help for the nodes that are not neighbors of existing customers

➔ identify areas of the social network that are "dense" with customers

For targeting consumers, collective inference gives additional improvement, especially for non-network neighbors

(Hill et al. '07)

Model (network only)	Predictive Performance (Area under ROC curve/ Mann-Whitney Wilcoxon stat) NN	non-NN
All first-order network variables	0.61	0.71
All first-order + "oracle" (wvRN)	0.63	0.74
All first-order + collective inference* (wvRN)	**0.63**	**0.75**

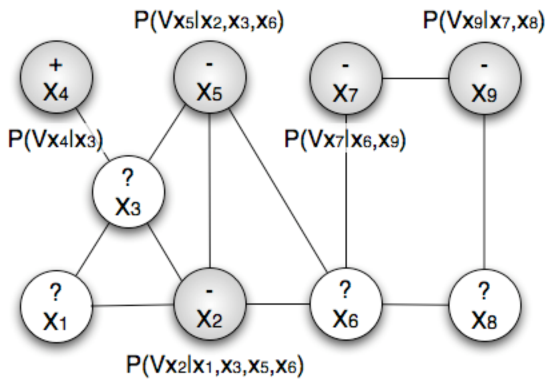
Model (with traditional variables)	Predictive Performance (Area under ROC curve/ Mann-Whitney Wilcoxon stat) NN	non-NN
All traditional variables	0.68	0.72
All trad + local network variables	0.69	0.73
All trad + local network + collective inference*	**0.72**	**0.77**

* with network sampling and pruning

Models of network data

	Disjoint inference	Collective inference
No learning	wvRN	Gaussian random fields, wvRN
Disjoint learning	ACORA, RBC, RPT, SLR	MLN, RBN, RDN, RMN
Collective learning	--	RBN w/EM, PL-EM, RGP

Collective learning



Consider links among <u>unlabeled</u> entities during learning

Collective learning is the network-data analog of semi-supervised learning

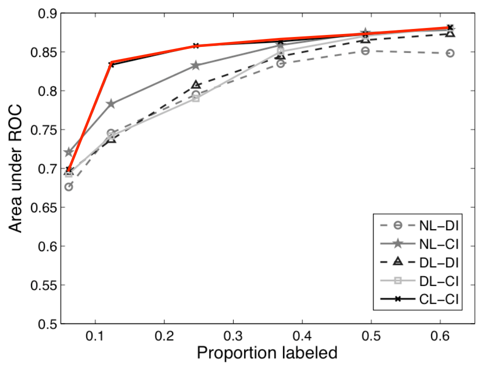
So far, network modeling techniques have focused on

1. exploiting links among unlabeled entities for inference (i.e., collective inference)
2. exploiting links between unlabeled and labeled for inference (e.g., identifiers)

Can we take into account links between unlabeled and labeled during <u>learning</u>?

- Ignoring missing data may be suboptimal, especially when lots of labels are missing and there is significant label autocorrelation
- Large body of related work on semi-supervised and transductive learning, but it has dealt primarily with i.i.d. data
- Exceptions:
  - PRMs w/EM (Taskar et al. ‘01)
  - Relational Gaussian Processes (Chu et al. ‘06)
  - Pseudolikelihood EM (Xiang and Neville ‘08)

Collective learning improves classification



Collective-learning/ collective-inference achieves equivalent or superior accuracy in all but sparsely labeled networks

The most significant gains occur when the network has a moderate amount of known labels

See Xiang & Neville ICDM'08 for details

Models of network data

	Disjoint inference	Collective inference
No learning	wvRN	Gaussian random fields, wvRN
Disjoint learning	ACORA, RBC, RPT, SLR	MLN, RBN, RDN, RMN
Collective learning	--	RBN w/EM, PL-EM, RGP

Collective learning, disjoint inference

Use unlabeled data for learning, but not for inference

- Open: No current methods do this
- However, disjoint inference is much more efficient
- May want to use unlabeled data to learn disjoint models (e.g., infer more labels to improve use of identifiers)

Recap

	Disjoint inference	Collective inference
No learning		
Disjoint learning		
Collective learning	--	

Conclusions: part I

1. Social network data often exhibit autocorrelation, which can provide considerable leverage for inference
2. "Labeled" entities link to "unlabeled" entities
   - Disjoint inference allows direct "guilt-by-association"
   - Disjoint learning can use correlations among attributes of related entities to improve accuracy
3. "Unlabeled" entities link among themselves
   - Inferences about entities can affect each other (e.g., indirect guilt by association)
   - Collective inference can improve accuracy
   - Results show that there is a lot of power for prediction just in the network structure
   - Collective learning can improve accuracy for datasets with a moderate number of labels or when labels are clustered in the graph

Conclusions: part II

5. The social network can be used to create variables that can be used in traditional ("flat") modeling
6. More sophisticated learning techniques exploit networks correlation in alternative ways
   - Node identifiers capture 2-hop autocorrelation patterns and linkage similarity
   - Models of the joint "network" distribution identify global attribute dependencies
   - These models can <u>learn</u> autocorrelation dependencies
7. There are many important methodological issues and open questions (see supplemental material)

By this point, hopefully, you are familiar with:

1. a wide-range of potential applications for predictive modeling in (social) networks
2. different approaches to network learning and inference
   - from simple to complex
   - a framework for organizing the ideas
3. various issues involved with each approach
4. when each approach is likely to perform well

See supplemental material for:

1. a large collection of related issues and research
2. potential difficulties for learning accurate network models
3. various methodological issues associated with analyzing network models
4. an extended social media example

Related network-analysis topics

Identifying groups in social networks

Predicting links

Entity resolution

Finding (sub)graph patterns

Generative graph models

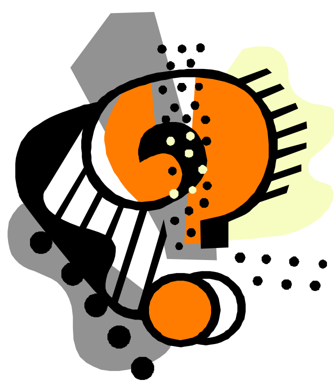
Social network analysis (SNA)

Preserving the privacy of social networks and SNA

Please see tutorial webpage for slides and additional pointers:
http://www.cs.purdue.edu/~neville/courses/icwsm09-tutorial.html

Thanks to…

Pelin Angin
Avi Bernstein
Scott Clearwater
Brian Dalessandro
Lisa Friedland
Brian Gallagher
Henry Goldberg
Michael Hay
Shawndra Hill
Rod Hook
David Jensen
John Komoroske
Kelly Palmer
Matthew Rattigan
Ozgur Simsek
Sofus Macskassy
Andrew McCallum
Alan Murray
Claudia Perlich
Ben Taskar
Chris Volinsky
Rongjing Xiang
Xiaohan Zhang
Rong Zheng



http://pages.stern.nyu.edu/~fprovost/
http://www.cs.purdue.edu/~neville

Google: foster provost | jennifer neville

Supplemental material

(see also resource list on tutorial web page)

Some other issues: part 0

Potential pathologies

- Statistical tests assume i.i.d data…
- Networks have a combination of widely varying linkage and autocorrelation …which can complicate application of conventional statistical tests

Methodology

- Within-network classification naturally implies dependent training and test sets
- How to evaluate models?
- How to understand model performance?
- How to accurately assess performance variance? (Open question)

Some other issues: part I

Propagating label information farther in the network
- Leverage other features (Gallagher & Eliassi-Rad SNA-KDD'08)
- Create "ghost" edges (Gallagher et al. KDD'08)
- Create "similarity" edges from other features (Macskassy AAAI'07)
- Leverage graph similarity of nodes (Fouss et al. TKDE'07)
- Latent group models (Neville & Jensen ICDM'05)

Do we know anything about the dynamics of label propagation?
- e.g., do true labels propagate faster than false ones?
- see (Galstyan & Cohen '05a,'05b,'06,'07)

What if labeling nodes is costly?
- Choose nodes that will improve collective inference (Rattigan et al. '07, Bilgic & Getoor KDD '08)

What if acquiring link data is costly?
- Acquire link data "actively" (Macskassy & Provost IA '05)

Some other issues: part II

What links you use makes a big difference
- Automatic link selection (Macskassy & Provost JMLR '07)
- Augment data graph w/2-hop paths (Gallagher et al. KDD '08)

How does propagating information with collective inference relate to using identifiers?
- open question

Can we identify the (causal) reason for the observed network correlation?
- Reasons might be:
  - Homophily: similar nodes link together
  - Social influence: linked nodes change attributes to similar values
  - External factor: causes both link existence and attribute similarity
- Manski '93; Hill et al. '06; Bramoulle '07; Burk et al. '07; Ostreicher-Singer & Sundararajan '08; Anagnostopoulos et al. KDD'08

Some other issues: part III

Computation and storage requirements can be prohibitive for data on real social networks -- how can we deal with massive (real) social networks?

- ignore most of the network (traditional method)
- use simple models/techniques! (e.g., Hill et al. 2007)
- reduce size of network via sampling/pruning of links and/or nodes, hopefully without reducing accuracy (much) (e.g., Cortes et al. ‘01; Singh et al. ‘05; Hill et al. ‘06b; Zheng et al. ‘07)

What are the effects of partial network data collection?

- one may not have access to or complete control over collection of nodes and/or links
- different sampling/pruning methods may induce different effects (e.g., Stumpf et al. '05, Lee et al. '06, Handcock and Gile '02, Borgatti et al. '05)
- can we improve accuracy by sampling/pruning?
  - irrelevant links/nodes can interfere with modeling (Hill et al. 2007)

Some other issues: part IV

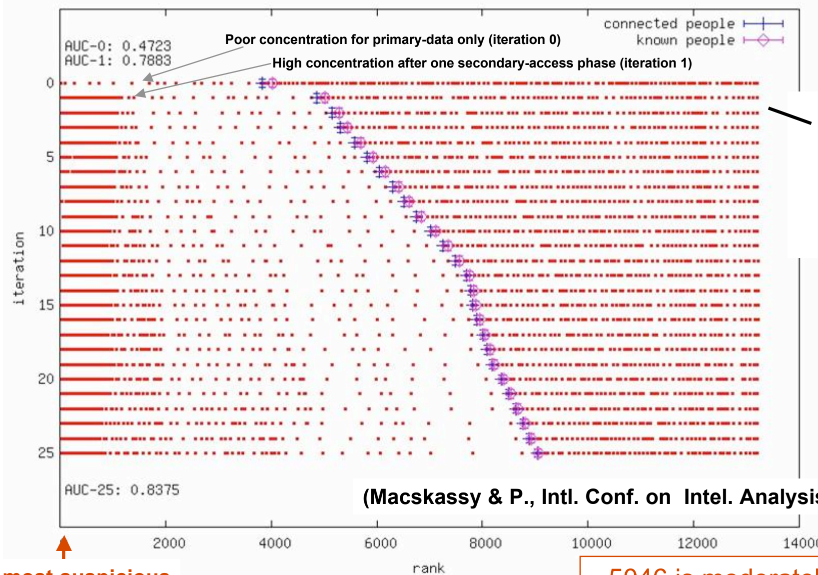
How to model networks changing over time?

- Summarize dynamic graph w/kernel smoothing (Cortes et al. ‘01, Sharan & Neville SNA-KDD’07)
- Sequential relational Markov models (Geustrin at al. IJCAI’03, Guo et al. ICML’07, Burk et al. ‘07)

How to jointly model attributes and link structure?

- RBNs with link uncertainty (Getoor et al. JMLR‘03)
- Model underlying group structure with both links and attributes (Kubica et al. AAAI’02, McCallum et al. IJCAI’05, Neville & Jensen ICDM’05)

A counter-terrorism application…



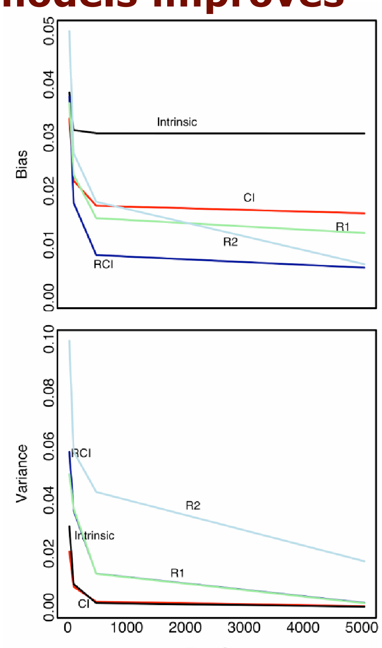
- high concentration of bad guys at "top" of suspicion ranking
- gets better with increased secondary-data access

5046 is moderately noisy: _ of "known" bad guys were mislabeled

Why learning collective models improves classification
(Jensen et al. KDD'04)

Why learn a joint model of class labels?

- Could use correlation between class labels and observed attributes on related instances instead
- But modeling correlation among unobserved class labels is a low-variance way of reducing model bias
- Collective inference achieves a large decrease in bias at the cost of a minimal increase in variance

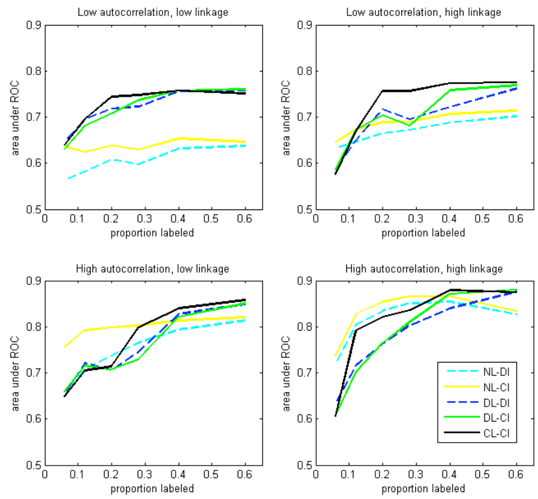


Comparing collective inference models

(Xiang & Neville SNA-KDD'08)

Learning helps when autocorrelation is low and there are other attributes dependencies

Learning helps when linkage is low and labeling is plentiful



Global vs. local autocorrelation

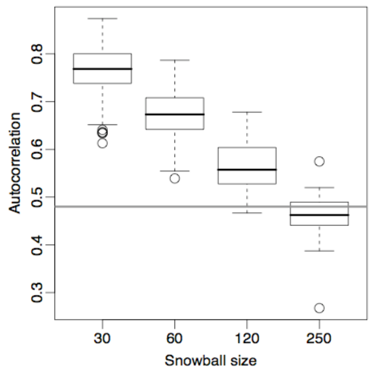
MLN/RDN/RMN:
- exploit global autocorrelation
- learning implicitly assumes training and test set are disjoint
- assumes autocorrelation is stationary throughout graph

ACORA with identifiers *(Perlich & Provost MLJ'06)*
- exploits local autocorrelation
- relies on overlap between training and test sets
- need sufficient data locally to estimate

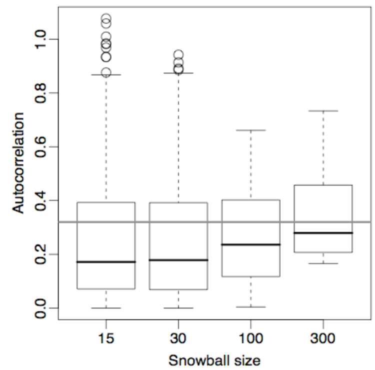
What about a combination of the two?
(open question)

Autocorrelation is non-stationary

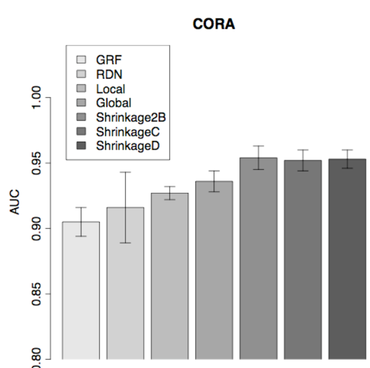
Cora: topics in coauthor graph



IMDb: receipts in codirector graph



Shrinkage models *(Angin & Neville SNA-KDD '08)*



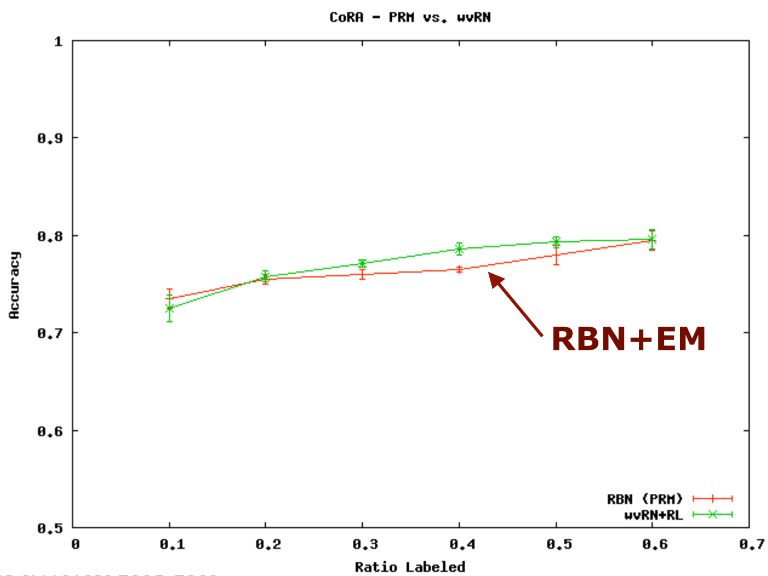
$$p(y^i|N(i)) \propto p(y) \prod_{j \in N(i)} p(y|j)$$

$$p_L(y|j) = \frac{\sum_{k \in N(j)} I_y(k)}{|N(j)|}$$

$$p_G(y|j) = \frac{|G_{yy^j}|}{\sum_{y' \in Y} |G_{y'y^j}|}$$

$$p_C(y|j) = c \cdot p_L(y|j) + (1-c) \cdot p_G(y|j)$$

Recall: RBN vs wvRN



Pseudolikelihood-EM

(Xiang & Neville KDD-SNA '08)

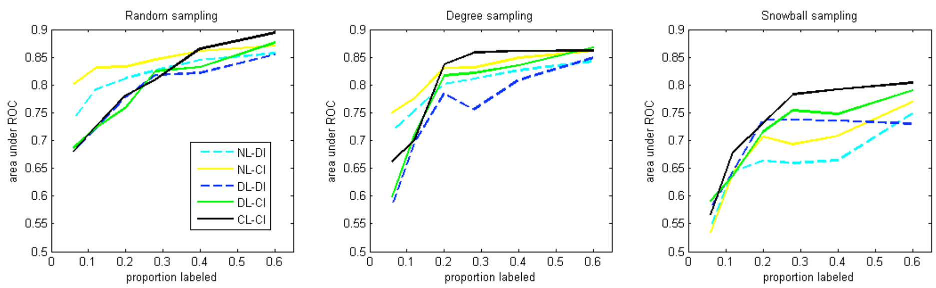
General approach to learning arbitrary autocorrelation dependencies in within-network domains

Combines RDN pseudolikelihood approach with mean-field approximate inference to learn a joint model of labeled and unlabeled instances

Algorithm

1. Learn an initial disjoint local classifier (with pseudolikelihood estimation) using only labeled instances
2. For each EM iteration:
   - **E-step**:
     apply current local classifier to unlabeled data with collective inference,
     use current expected values for neighboring labels;
     obtain new probability estimates for unlabeled instances;
   - **M-step**:
     re-train local classifier with updated label probabilities on unlabeled instances.

Comparison with other network models

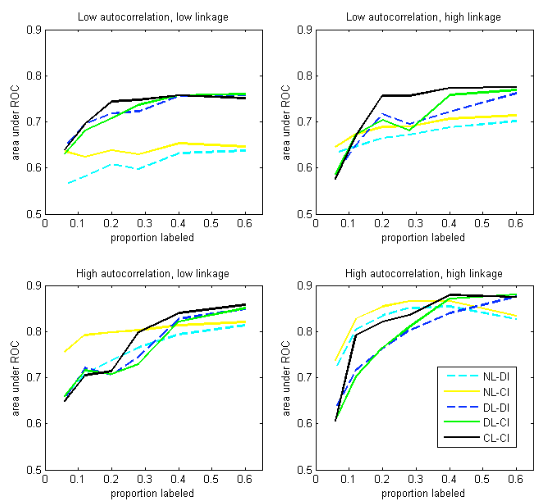


Collective learning improves performance when: (1) labeling is moderate, or (2) when labels are clustered in the network

Or when...

Learning helps when autocorrelation is low and there are other attributes dependencies

Learning helps when linkage is low and labeling is plentiful



Potential pathologies

Statistical tests assume i.i.d data…

Networks have a combination of widely varying linkage and autocorrelation

…which can complicate application of conventional statistical tests

- Naïve hypothesis testing can bias feature selection (Jensen & Neville ICML'02, Jensen et al. ICML'03)
- Naïve sampling methods can bias evaluation (Jensen & Neville ILP'03)

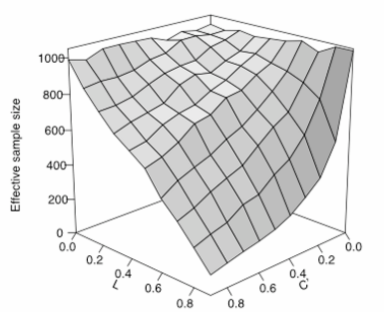
Bias in feature selection

(Jensen & Neville ICML'02)

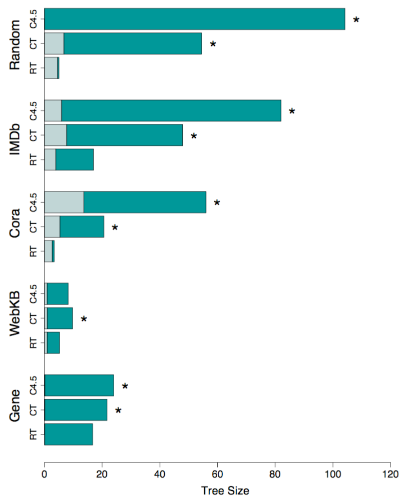
Relational classifiers can be biased toward features on some classes of objects (e.g., movie studios)

How?

- Autocorrelation and linkage reduce effective sample size
- Lower effective sample size increases variance of estimated feature scores
- Higher variance increases likelihood that features will be picked by chance alone
- Can also affect ordering among features deemed significant because impact varies among features (based on linkage)



Adjusting for bias:
Randomization tests



Randomization tests result in significantly smaller models *(Neville et al KDD'03)*

- Attribute values are randomized prior to feature score calculation
- Empirical sampling distribution approximates the distribution expected under the null hypothesis, given the linkage and autocorrelation

Metholodogy

Within-network classification naturally implies dependent training and test sets

How to evaluate models?

- Macskassy & Provost (JMLR'07) randomly choose labeled sets of varying proportions (e.g., 10%. 20%) and then test on remaining unlabeled nodes
- Xiang & Neville (KDD-SNA'08) choose nodes to label in various ways (e.g., random, degree, subgraph)
- See (Gallagher & Eliassi-Rad 2007) for further discussion

How to accurately assess performance variance? (Open question)

- Repeat multiple times to simulate independent trials, but…
  - Repeated training and test sets are dependent, which means that variance estimates could be biased (Dietterich '98)
- Graph structure is constant, which means performance estimates may not apply to different networks

Understanding model performance
(Neville & Jensen MLJ'08)

Collective inference is a new source of model error

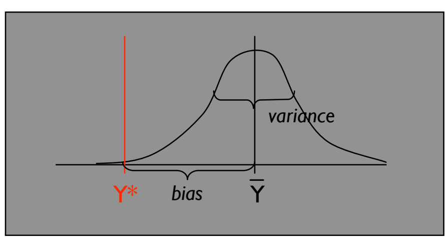
Potential sources of error:

- Approximate inference techniques
- Availability of test set information
- Location of test set information

Need a framework to analyze model *systems*

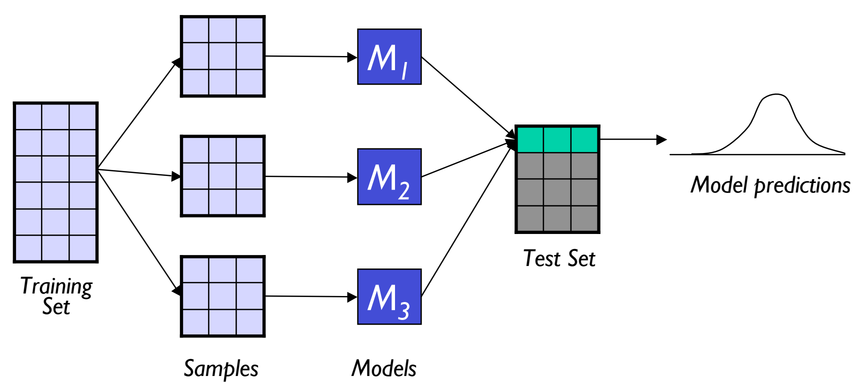
- Bias/variance analysis for collective inference models
- Can differentiate errors due to learning and inference processes

Conventional bias/variance analysis

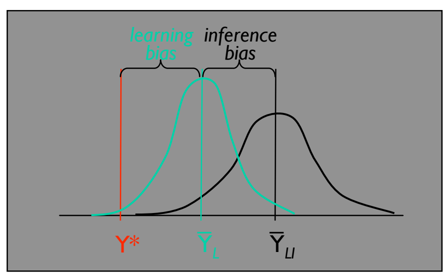


$$E_D[L_{sq}(t,y)] = \underbrace{E_D[(t - E_D[t])^2]}_{\text{noise}} + \underbrace{(E_D[t] - E_D[y])^2}_{\text{bias}} + \underbrace{E_D[(E_D[y] - y)^2]}_{\text{variance}}$$

Conventional bias/variance analysis



Bias/variance decomposition for relational data



$$E_{LI}[L_{sq}(t,y)] = \underbrace{E_L[(t-E_L[t])^2]}_{\text{noise}}$$

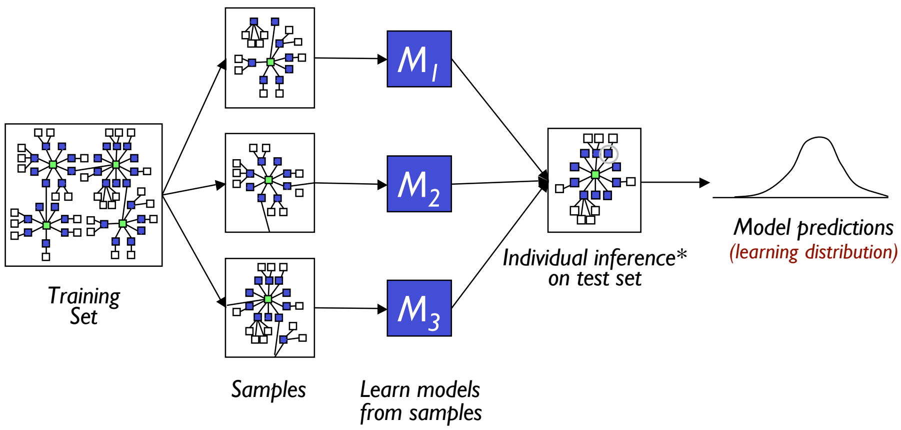
$$+\underbrace{(E_L[t]-E_L[y])^2}_{\text{learning bias}} + \underbrace{E_L[(E_L[y]-y)^2]}_{\text{learning variance}}$$

$$+\underbrace{(E_L[y]-E_{LI}[y])^2}_{\text{inference bias}} + \underbrace{E_{LI}[(E_L[y]-y)^2]-E_L[(E_{LI}[y]-y)^2]}_{\text{inference variance}}$$

$$+\underbrace{2(E_L[y]-E_L[t])(E_{LI}[y]-E_L[y])}_{\text{bias interaction term}}$$

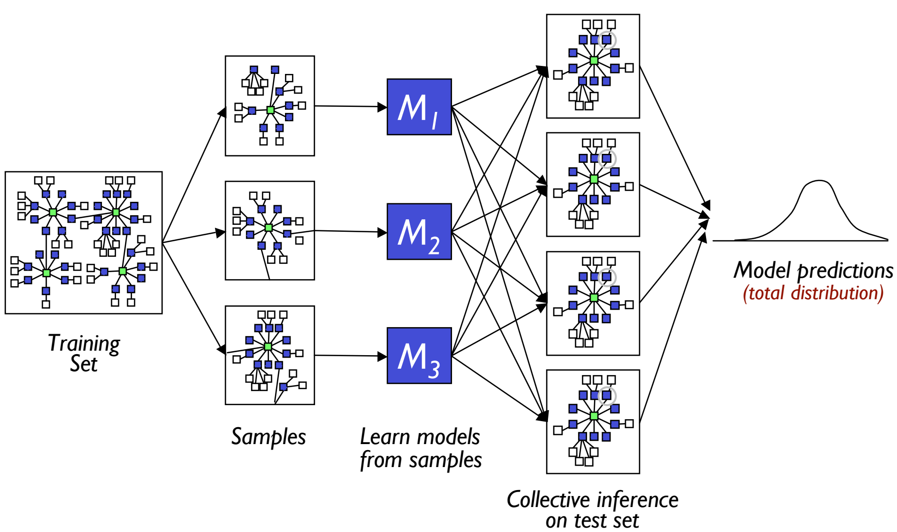
Expectation over learning and inference

Relational bias/variance analysis: part I

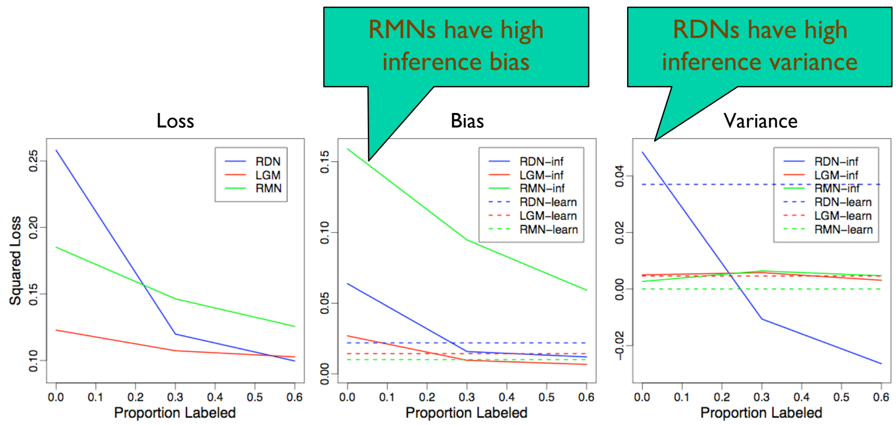


* Inference uses optimal probabilities for neighboring nodes' class labels

Relational bias/variance analysis: part II



Analysis shows that models exhibit different errors



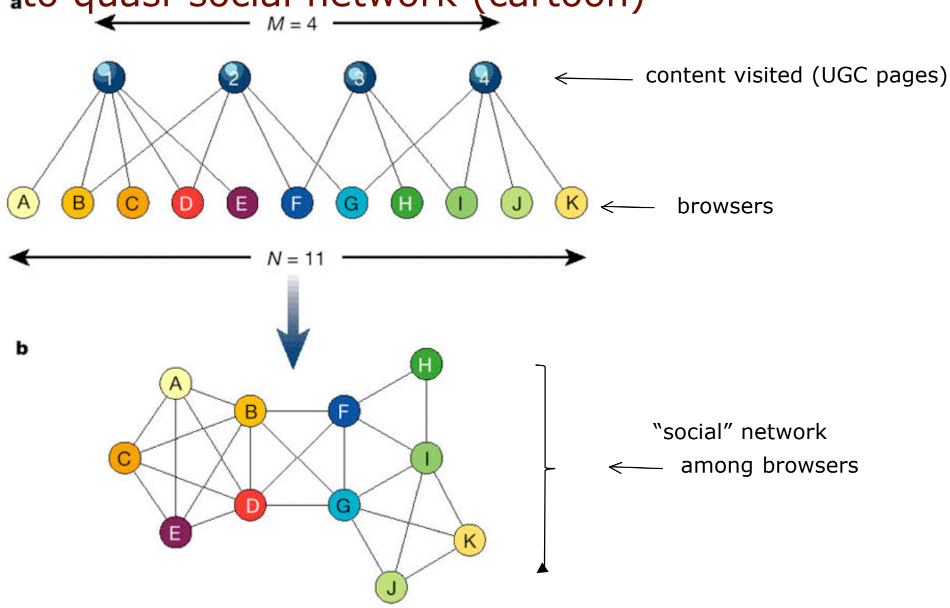
Another real-world example:

Mining data from social media for on-line brand advertising

Thanks to: media6°

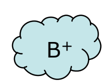
(See Provost et al. KDD 2009)

Social media example:
From bipartite content-affinity network to quasi-social network (cartoon)



Social media example:
On-line audience selection in a nutshell

- Advertiser indicates action showing brand affinity
  - visiting loyalty page, signing in to account, purchasing, visiting home page, etc.
- Collect brand action takers as *seed nodes*
  - call the set of seed nodes B^+



- Identify the set (N) of network neighbors of B^+
- Rank N based on "brand proximity" to B^+
  - using nearest-neighbor-style or more sophisticated models
  - brand proximity: a measure of similarity/distance between a node b and the set B+
- Choose audience A as the the top-ranked members of N
- Note: *This can be done without saving any PII: only random numbers for the browser and for the content*

Social media example:
Brand proximity measures

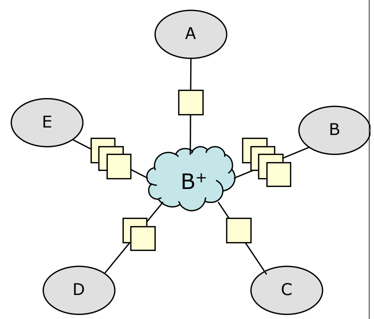
POSCNT
- number of unique content pieces connecting browser to B^+

MATL
- maximum number of content pieces through which paths connect browser to some particular action taker (i.e., seed node in B^+)

minEUD
- minimum Euclidean distance of normalized content vector to a seed node

maxCos
- maximum cosine similarity to a seed node

ATODD
- "odds" of a neighbor being an action taker (i.e., seed node in B^+).



Social media example:
The Social Network Data

(from a working ad network)

a sample of about 10 million anonymized browsers

all of their observed visits to social networking content over 90 days (from several of the largest SN sites)

bipartite graph:
- $10^7 \times 10^8$ with $\sim 2.5 \times 10^8$ non-zero entries

quasi-social network:
- 10^7 nodes with 20-40 neighbors each (on average)

Resultant audiences per brand
- on average ~100K seed nodes
- total network neighbor audience pool: 2-4 million

Social media example:
The Brand Data

More than a dozen well-known brands, separated into two groups:

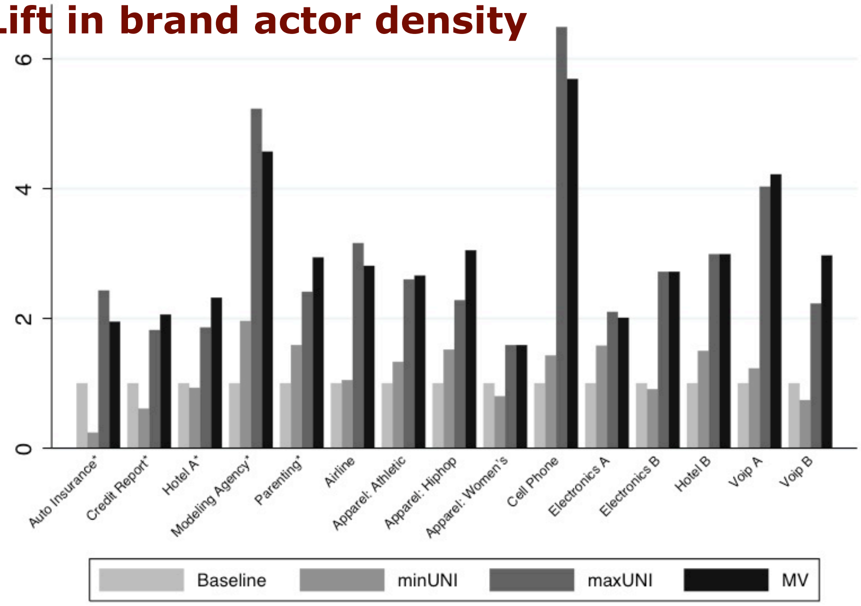
Group 1:

- Four brands where no advertising was done during experimental period (Hotel A, Modeling Agency, Credit Report, Auto Insurance)
- Plus a fifth "brand" comprising a sought-after demographic group (Parenting)

Group 2:

- 10 brands where some advertising was done during the experimental period
  - Apparel: HipHop, Voip A&B, Airline, Hotel B, Electronics A&B, Apparel: Athletic, Cell Phone, Apparel: Women's
- advertising uniform across network neighbors
- advertising does not lead directly to brand action

Social media example:
Lift in brand actor density



return

[For the top-10%, ATODD was usually the best]

Social media example:

In-vivo tests

Brand	Impressions of PSAs to top ranked	Impressions of PSAs to RON	Organic conversion lift
Electronic A	67	53,347	5.89
Apparel: Athletic	26,161	266,661	6.06
Apparel: Hiphop	5,757	223,509	64.65

We selected a small set of high-ranking network neighbors for three group-2 brands. In production we showed them only public service announcements (PSAs). We did the same (with the same campaign parameters) for a "run of network" campaign (bid on everyone).

We acquired from the ad exchange the rates of conversion -- here "organic" conversion.

return

Social media example:

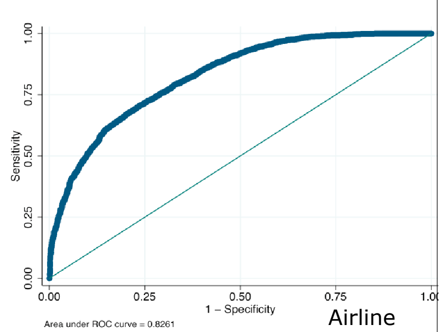
Social vs. Quasi-Social

return

The quasi-social network embeds a friends network?

- estimate each browser's home page based on techniques analogous to author id based on citations (Hill & Provost, 2003)
- estimate "friends" to be those who visit each other's home page
- do brand proximity measures rank brand actors' friends highly?

Brand	F-AUC on all B	F-AUC on *N* only
Hotel A	0.96	0.79
Modeling Agency	0.98	0.84
Credit Report	0.93	0.79
Parenting	0.94	0.80
Auto Insurance	0.97	0.81
…		
15 Brand Average	0.96	0.81



Airline

Social media example:
One more test

For one brand (Cell Phone) we asked Quantcast.com for demographic profiles of the seed nodes and their network neighbors:

Demographic	**Seeds**	**Neighbors**
Gender	Female	Female
Ethnicity	Hispanic	Hispanic
Age	Young	Young
Income	Low	Low
Education	No College	No College

return

Fun: Mining Facebook data (associations)

Birthday → School year, Status (yawn?)

Finance → Conservative
Economics → Moderate
Premed → Moderate
Politics → Moderate, Liberal or Very_Liberal
Theatre → Very_Liberal
Random_play → Apathetic

Marketing → Finance
Premed → Psychology
Politics → Economics

Finance → Interested_in_Women
Communications → Interested_in_Men
Drama → Like_Harry_Potter

Dating → A_Relationship, Interested_in_Men
Dating → A_Relationship, Interested_in_Women

Interested_in_Men&Women → Very_Liberal

Acronym guide

ACORA: Automatic construction of relational attributes (Perlich & Provost KDD'03)

AMN: Associative Markov network (Taskar ICML'04)

BN: Bayesian network

BLP: Bayesian logic program (Kersting & de Raedt '01)

DN: Dependency network (Heckerman et al. JMLR'00)

EM: Expectation maximization

GRF: Gaussian random field (Zhu et al. ICML'03)

ILP: Inductive logic programming

MLN: Markov logic network (Richardson & Domingos MLJ'06)

MN/MRF: Markov network/random field

NT: Network targeting (Hill et al.'06)

PGM: Probabilistic graphical models

PL: Pseudolikelihood

RBC: Relational Bayes classifier (Neville et al. ICDM'03)

RBN: Relational Bayesian network (aka probabilistic relational models) (Friedman et al. IJCAI'99)

RDB: Relational database

RDN: Relational dependency network (Neville & Jensen ICDM'04)

RGP: Relational Gaussian process (Chu et al. NIPS'06)

RMN: Relational Markov network (Taskar et al. UAI'02)

RPT: Relational probability trees (Neville et al. KDD'03)

SLR: Structural logistic regression (Popescul et al. ICDM'03)

wvRN: Weighted-vector relational neighbor (Macskassy & Provost JMLR'07)