Networks and Graphical Models

Big Data Lectures – Chapter 8

Lexin Li

Division of Biostatistics
University of California, Berkeley

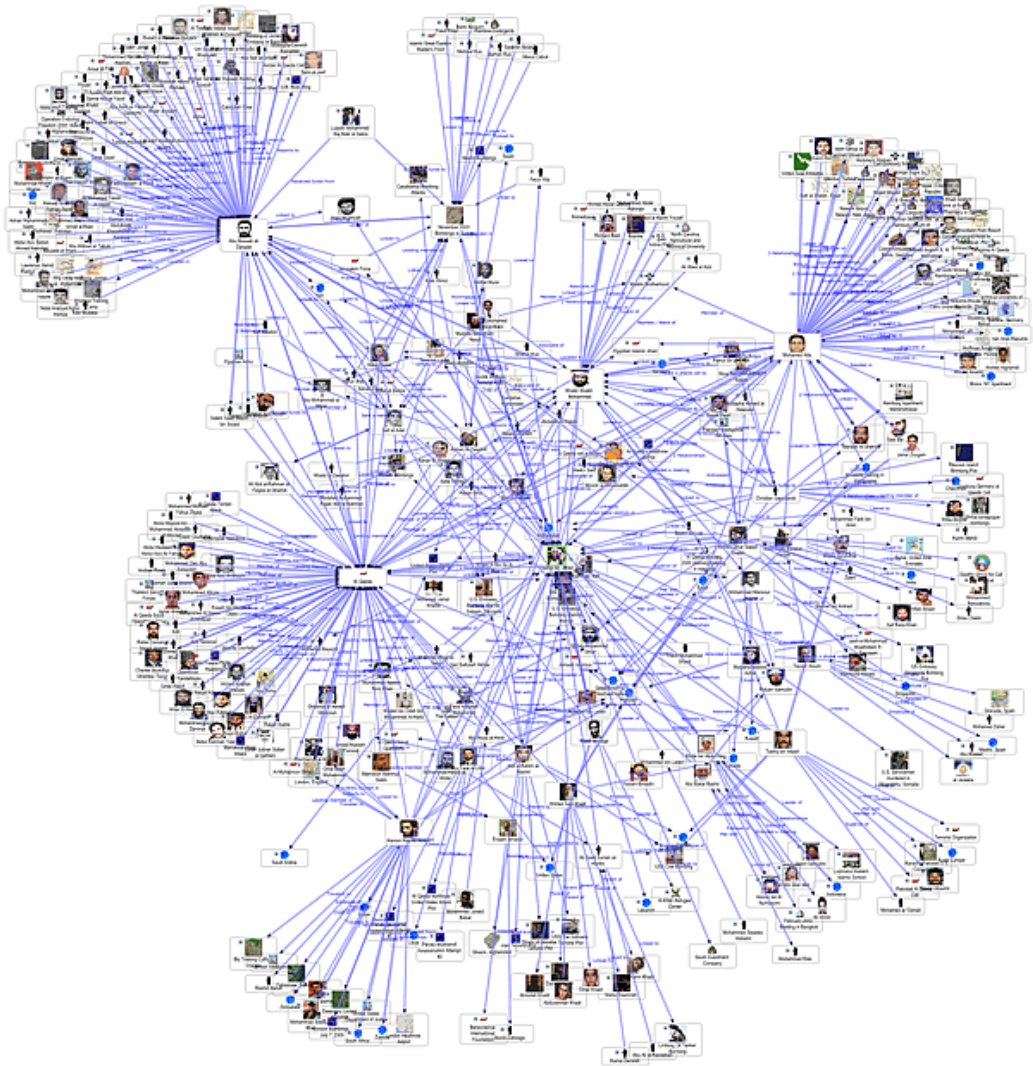
Outline

- list of topics:
  - introduction and basic concepts
  - network generative models
  - network enhanced analysis
  - network link analysis
  - Gaussian graphical models

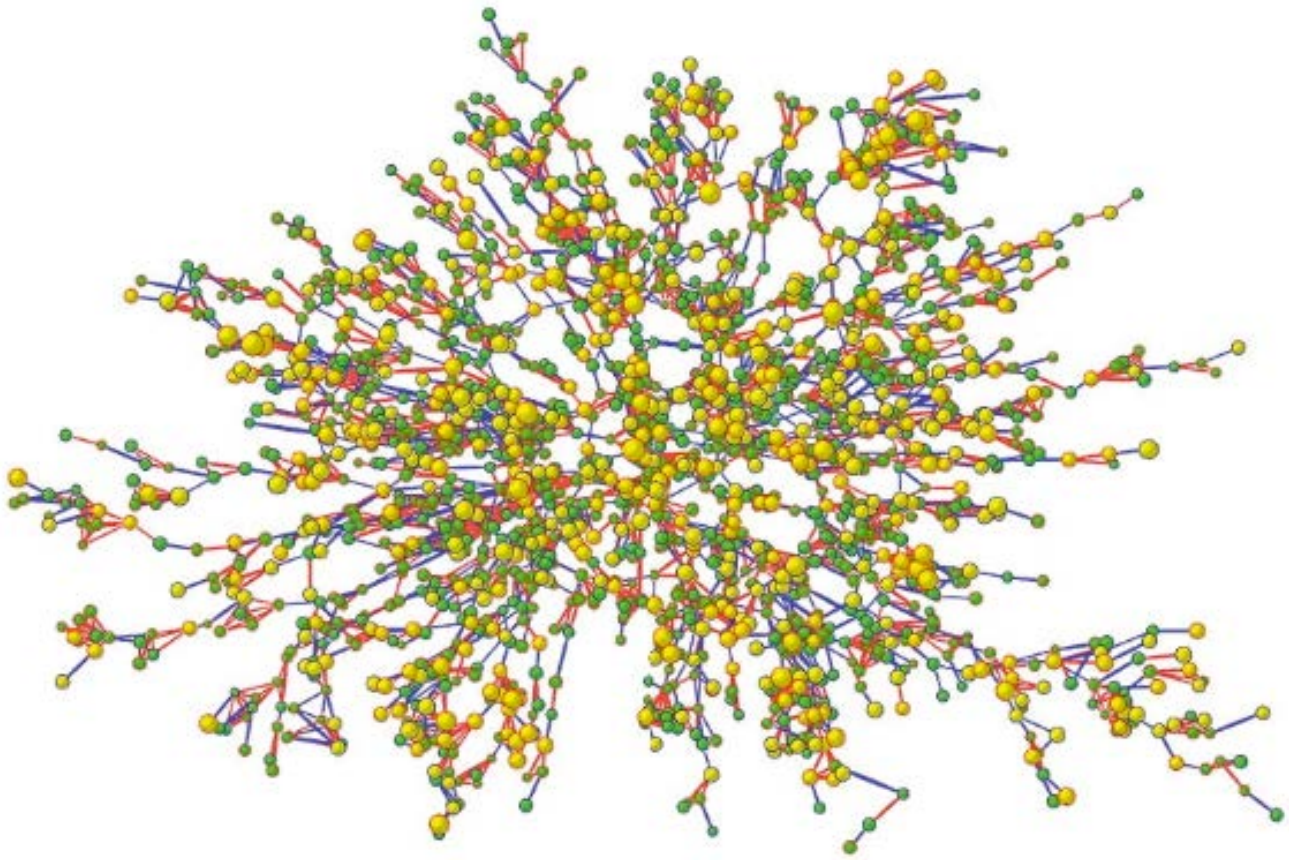
Networks: Basics

Examples

- networks data are now **everywhere**
  - Facebook, LinkedIn, Epinions, Flixster, last.fm, whrrl.com, ...

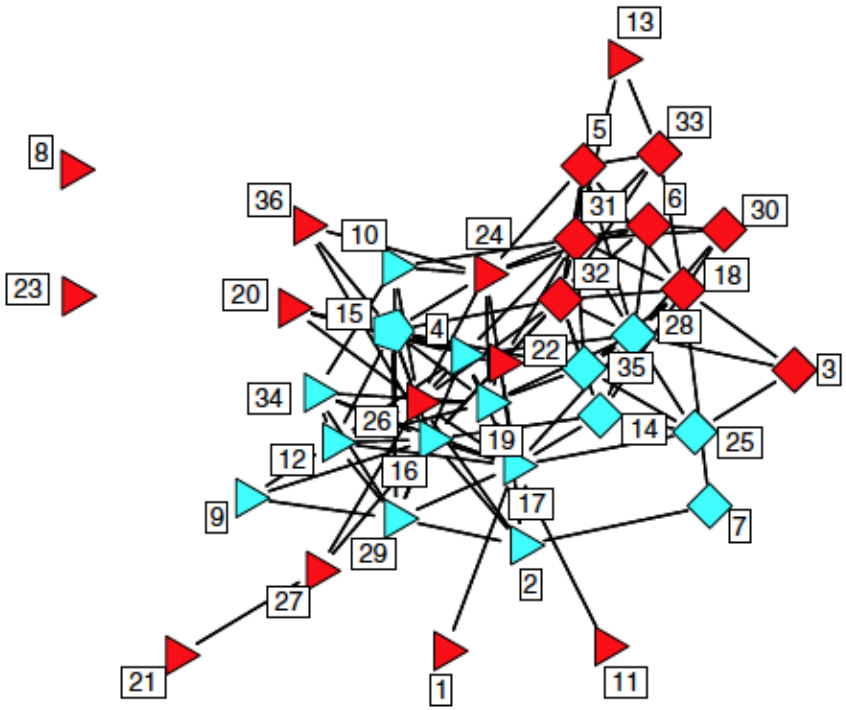


Examples

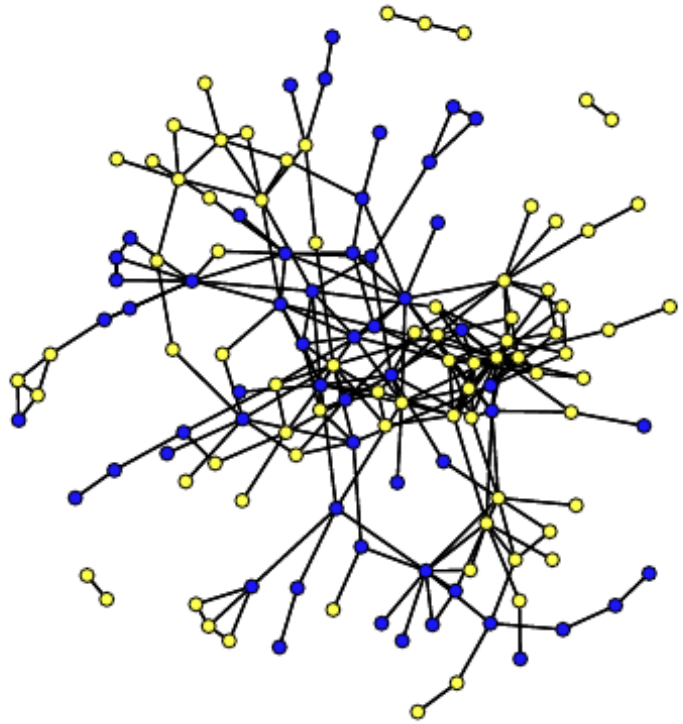


social network of 12,067 people from the Framingham Heart Study (Christakis et al., 2003). Yellow circle: obese, green: non-obese; circle size proportional to BMI. Purple link: friendship or marital tie, orange: family tie

Examples



lawyer collaboration



protein interaction

Key concepts

- graph:
  - $\mathcal{G} = \{V, E\}$, V the **vertex / node** set, and E the **edge / link** set
  - **directed graph** vs **undirected graph**
- **adjacent matrix**:
  - $\boldsymbol{A} = \{a_{ij}\}$, w_{ij} denotes "similarity" / inverse of "distance" between nodes i and j
  - $\boldsymbol{A}$ is an **unweighted** adjacent matrix if $a_{ij} = 0/1$
- random walk on $\mathcal{G}$:
  - **transition probability** for a random walk associated with $\mathcal{G}$:

$$P_{ij} = \frac{a_{ij}}{\sum_j a_{ij}} = \frac{a_{ij}}{a_{i+}}$$

  - **stationary probability** for an irreducible and aperiodic walk on $\mathcal{G}$: $\pi_i = \lim_{t\to\infty} P_{ij}^{(t)}$; for undirected graph,

$$\pi_i = \frac{\sum_j a_{ij}}{\sum_i \sum_j a_{ij}}$$

Key concepts

- degree:
  - for a directed graph:

$$\textbf{in-degree}: \ a_{+i} = \sum_k a_{ki}, \quad \textbf{out-degree}: \ a_{i+} = \sum_j a_{ij}$$

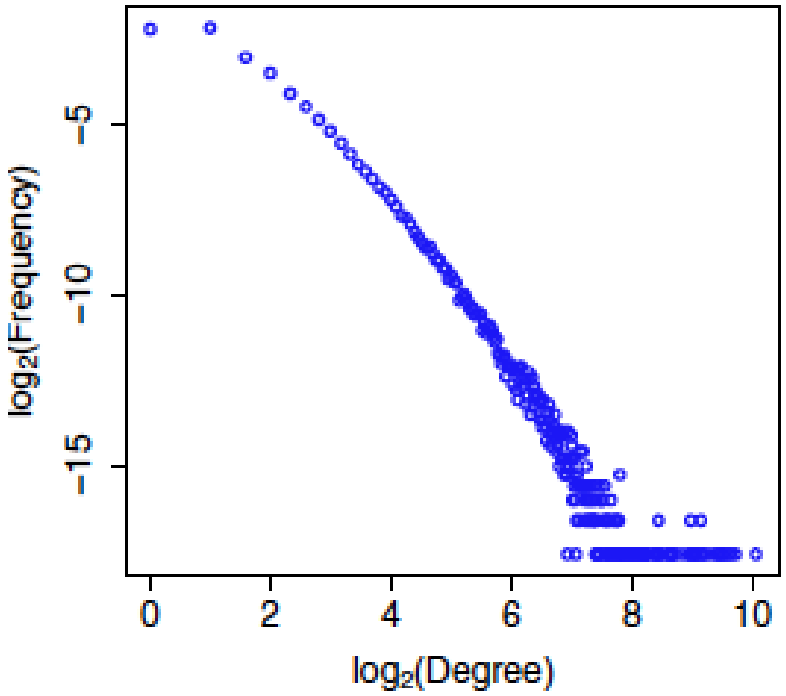
  - for an undirected graph: **vertex degree** = in-degree = out-degree

$$d_i = a_{+i} = a_{i+}$$

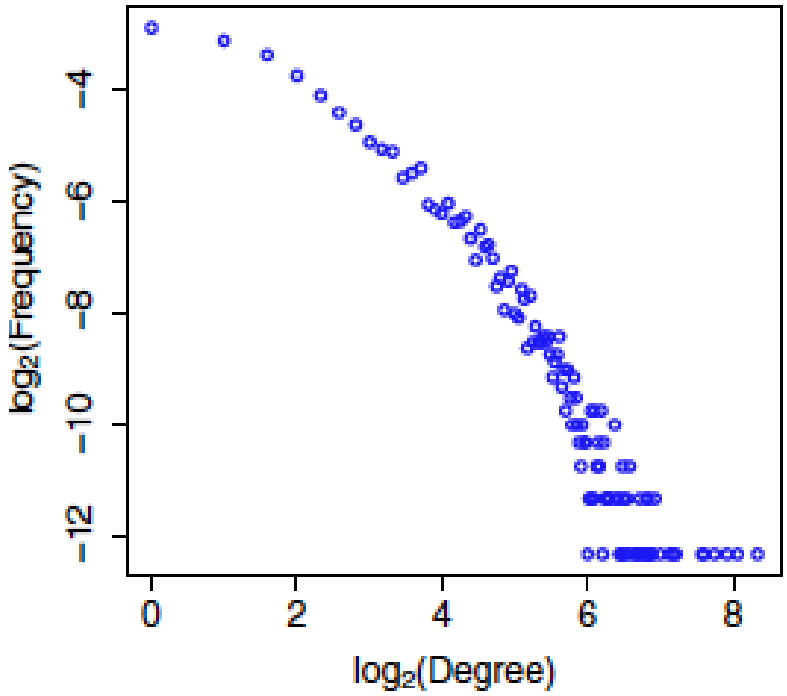
  - **degree matrix**: $\boldsymbol{D} = \mathrm{diag}(d_i)$

- power-law property:
  - in words: the majority of vertices are of very low degree, while a small number of vertices are of much higher degree (two to three orders of magnitude higher)
  - the power-law distribution for degrees: $f_d \propto d^{-\alpha}$
  - (approximate) power-law degree distributions appear to be **ubiquitous** in networks across many areas of the sciences

Power-law degree distribution



Internet network



proteins interaction network

Graph Laplacian

- **graph Laplacian matrix** for an undirected graph $\mathcal{G}$:
  - Laplacian matrix:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}; \quad \boldsymbol{y}^\top \boldsymbol{L}\boldsymbol{y} = \frac{1}{2}\sum_i \sum_j a_{ij}(y_i - y_j)^2$$

  - normalized Laplacian matrix:

$$\tilde{\boldsymbol{L}} = \boldsymbol{D}^{-1/2}(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{D}^{-1/2}; \quad \boldsymbol{y}^\top \tilde{\boldsymbol{L}}\boldsymbol{y} = \frac{1}{2}\sum_i \sum_j a_{ij}\left(\frac{y_i}{\sqrt{d_i}} - \frac{y_j}{\sqrt{d_j}}\right)^2$$

  - modified normalized Laplacian matrix:

$$\tilde{\boldsymbol{L}}' = \boldsymbol{D}'^{-1/2}(\boldsymbol{D}' - \boldsymbol{A}')\boldsymbol{D}'^{-1/2}; \quad \boldsymbol{y}^\top \tilde{\boldsymbol{L}}'\boldsymbol{y} = \frac{1}{2}\sum_i \sum_j \pi_i P_{ij}\left(\frac{y_i}{\sqrt{\pi_i}} - \frac{y_j}{\sqrt{\pi_j}}\right)^2$$

    - it is the normalized version of a Laplacian matrix with a_{ij} replaced with $a'_{ij} = (\pi_i P_{ij} + \pi_j P_{ji})/2$

Network metrics

- vertex centrality:
  - measures "importance" of a vertex in a network: e.g., deletion of which genes in a gene regulatory network is likely to be lethal to the corresponding organism; how critical is a given router in an Internet network to the flow of traffic...
  - three common centrality measures:
    - closeness / weighted degree centrality: $1/\sum_{u \in V} \mathrm{dist}(u, v)$, or $\sum_{u \in V} a_{uv}$; "central" means the vertex is "close" to many other vertices
    - betweenness: $\sum_{s,t \in V} \sigma(s, t|v)/\sigma(s, t)$, where $\sigma(s, t|v)$ is the total number of shortest paths between s and t that pass through v; measures the extent to which a vertex is located "between" other pairs of vertices
    - eigenvector centrality: typically can be expressed in terms of eigenvector solutions of appropriately defined linear systems of equations

Network metrics

- network cohesion:
  - measures the extent to which subsets of vertices are cohesive / stuck together; e.g., do friends of a given actor in a social network tend to be friends of one another as well; what collections of proteins in a cell appear to work closely together...
  - **clique**: subset of vertices all of which are connected; triangle
  - **clustering coefficient**: number of closed triplets divided by number of triplets (both open and close)
  - **modularity**: measure the strength of division of a network into modules (also called groups, clusters or communities); networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules
  - **characteristic path length**: average shortest path length between all pairs of nodes in the network
  - **global network efficiency**: mean of the inverse of characteristic path length between each pair of nodes within the network
  - **local network efficiency**

Network Generative Models

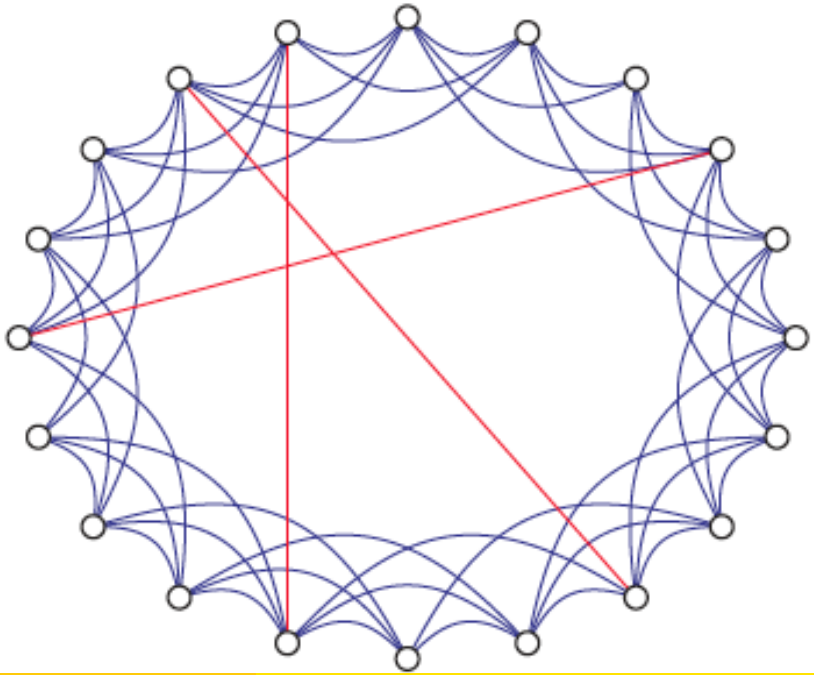
Network generative models

- network **generative** models:
  - random graph models as an appropriate frame of reference when testing significance of structural characteristics of an observed network
  - simple, parameterized generative models to possess and to understand some commonly observed properties in real-world networks, such as broad degree distributions or small-world effects

- classical random graph models:
  - **Erdös-Rényi model**: a graph of a given **order** (N_v, number of nodes) and **size** (N_e, number of edges), equal probability for each edge among every pair of nodes
  - **Gilbert model**: assign an edge independently with probability $p \in (0, 1)$
  - simulating random graphs: computational tricks for large N_v, noting that the graph is usually very sparse
  - (some other graph models later)

Network generative models

- network **growth** models:
  - typically a simple mechanism is specified for how the network changes at any given point in time; interest then centers on what properties emerge for the network in the limit of a large number of consecutive time periods; if certain properties are found to match those observed in real-world networks, this is often taken as suggestive that the specified mechanism is perhaps a reasonable **approximation** to a similar, real-world mechanism.
- **small-world** models:
  - many networks in the real world display high levels of clustering, but small distances between most nodes
  - Watts-Strogatz model: begin with a graph with lattice structure, and then randomly "rewiring" a small percentage of the edges



Network Enhanced Analysis

Network enhanced analysis

- problem set up:
  - the network structural (link) information is completely **known**
  - usually only **a single snapshot** of the network, i.e., only one network link structure
  - multiple samples, each has a vector of features, and **each feature** forms a node of the network
  - multiple samples, **each sample** itself forms a node of the network
- some key applications:
  - group comparison; regression estimation and prediction
  - prediction of an outcome variable on a node based on partially observed outcomes on other nodes — **network vertex modeling**
- case studies:
  - identification of differentially expressed genes
  - survival time prediction
  - protein function prediction
  - university research score prediction

Network enhanced analysis

- case study 1: Wei and Li (2007, Bioinformatics)
  - breast cancer data: 286 patients, 107 cases and 179 controls
  - each subject measures 1533 genes, belonging to 33 pathways
  - classical solution: two sample t-test; ignore potential correlation among the genes
  - **network enhanced analysis**: each gene forms a node of the network, and the link is determined by the pathway information
  - solution: define a latent state $\boldsymbol{D} = (D_j)$, with $D_j = 1$ if gene j differentially expressed and 0 o.w.

$$p(\boldsymbol{D}|\boldsymbol{y}) \propto p(\boldsymbol{D}|\boldsymbol{A})p(\boldsymbol{y}|\boldsymbol{D})$$

specify a model on $p(\boldsymbol{y}|\boldsymbol{D})$, a Markov random field model for $p(\boldsymbol{D}|\boldsymbol{A})$ given the **known** adjacent matrix $\boldsymbol{A}$, then optimize over $\boldsymbol{D}$

Network enhanced analysis

- case study 2: Li and Li (2008, Bioinformatics)
  - glioblastoma data: training set with 55 patients (5 alive), and testing set with 65 patients (4 alive)
  - each patient measures 1533 genes, belonging to 33 pathways
  - response: log of time to death; removed the alive patients so no censoring
  - **network enhanced analysis**: each gene forms a node of the network, and the link is determined by the pathway information
  - solution: use graph Laplacian as penalty; penalizes β_js that differ too much over similar nodes

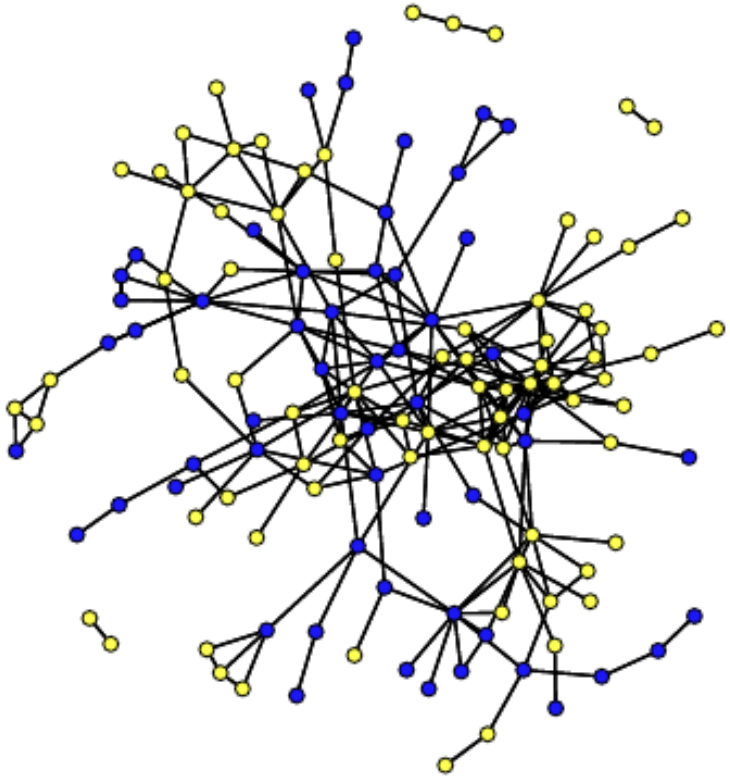
$$\sum_{i=1}^{n}(y_i - \beta_1 x_1 - \ldots \beta_p x_p)^2 + \lambda_1 \sum_{j=1}^{n} |\beta_j| + \lambda_2 \sum_{u,v} a_{uv} \left(\frac{\beta_u}{\sqrt{d_u}} - \frac{\beta_v}{\sqrt{d_v}} \right)^2$$

$$= \quad \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \boldsymbol{\beta}^{\mathsf{T}} \tilde{\boldsymbol{L}} \boldsymbol{\beta}$$

Modeling of network vertices

- case study 3:
  - proteins are fundamental to the complex molecular and biochemical processes taking place within organisms; understanding their roles – functions – are critical
  - data on Bakers yeast: 134 vertices – proteins, and 241 edges – protein interactions
  - let X denote the vertex process corresponding to the annotation, intracellular signaling cascade (ICSC), i.e., $X_i = 1$ if the protein corresponding to vertex i has the ICSC annotation, and $X_i = 0$ otherwise
  - goal: predict if a protein has ICSC annotation or not, based on the underlying belief that proteins that are nearer to each other in a network of protein interactions are to possess more similar functional roles
  - **network enhanced analysis**: each protein (sample) forms a node of the network, and the link is determined by the known protein-protein interaction

Modeling of network vertices



- network of interactions among proteins known to be responsible for cell communication in yeast
- yellow vertices denote proteins that are known to be involved in ICSD, a specific form of communication in the cell in yeast
- blue vertices denote remaining proteins

Modeling of network vertices

- nearest neighbor prediction:

$$\frac{\sum_{j \in \mathcal{N}_i} X_j}{|\mathcal{N}_i|}$$

  - prediction is simply the average of the observed vertex process in the neighborhood of i
  - for the protein example where X_js are binary, then for each vertex in the test set, if the fraction of neighbors in the training set annotated with ICSC was greater than a given threshold, that vertex is also assigned the ICSC annotation
  - while nearest neighbor methods may seem rather informal and simple, they often are found to be quite competitive with more formal and complex methods

Modeling of network vertices

- kernel regression:
  - fit a kernel regression, e.g., kernel least squares, kernel logistic regression and kernel support vector machines
  - the key is to design an appropriate **kernel on a network** – recall that the kernel function essentially quantifies the similarity among its arguments, and thus, a kernel on a network graph G should be designed to capture suspected similarity relationships among vertices in V
  - some commonly used kernels for graph:
    - Laplacian kernel: (pseudo)inverse of the graph Laplacian $K = L^{-}$
    - diffusion kernel: $K = \exp(\xi L) = \sum_{m=0}^{\infty} \frac{(-\xi)^m}{m!} L^m$
    - ...

Modeling of network vertices

- Markov random field model:
  - definition: let $X = (X_1, \ldots, X_{N_v})^\mathsf{T}$ be a collection of discrete random variables defined on V, then we say X is a **Markov random field** on G if
$$P(X_i = x_i | X_{-i} = x_{-i}) = P(X_i = x_i | X_{\mathcal{N}_i} = x_{\mathcal{N}_i})$$
and $P(X = x) > 0$ for any x. That is, X_i is conditionally independent of all other X_k , given the values of its neighbors
  - auto-logistic Markov random field model: for all binary X_i and some additional constraint
$$P(X_i = 1 | X_{\mathcal{N}_i} = x_{\mathcal{N}_i}) = \frac{1}{1 + \exp\left\{-(\alpha_i + \sum_{j \in \mathcal{N}_i} \beta_{ij} x_j)\right\}}$$
  - auto-Gaussian Markov random field model: for continuous X_i and some additional constraint
$$P(X_i | X_{\mathcal{N}_i} = x_{\mathcal{N}_i}) = \alpha_i + \sum_{j \in \mathcal{N}_i} \beta_{ij}(x_j - \alpha_j)$$

Modeling of network vertices

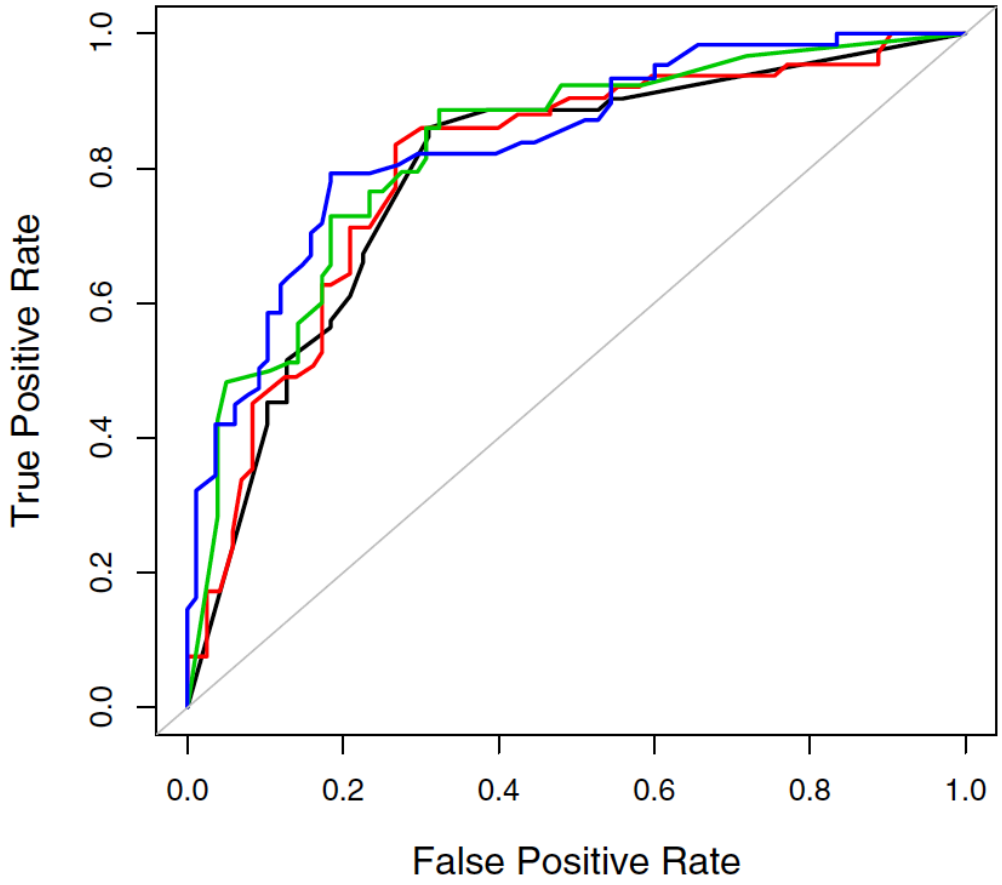


Figure: nearest neighbor (black), Markov random field (red), kernel logistic regression with Laplacian kernel (green), and with a more complicated kernel (blue)

Modeling of network vertices

- case study 4: Xu, Dyer, and Owen (2010, AoAS)
  - UK university data: 107 universities – vertices; the number of links from one university to another forms an asymmetric weighted adjacent matrix $\boldsymbol{A}$; the distribution of a_{ij} is heavily right-tailed, and follows roughly the power law; about 15% weights are zero (no link), 50% are less than 7, and the maximum is 2,130
  - response on vertex: a research score (RAE) for each university; only a subset of RAE are available as a training data
  - **network enhanced analysis**: each university (sample) forms a node of the network, and the link is determined by the known university-university interaction

Modeling of network vertices

- semi-supervised learning:

$$\min_{\boldsymbol{Z}} \boldsymbol{Z}^{\mathsf{T}} \boldsymbol{\Delta} \boldsymbol{Z} + (\boldsymbol{Z} - \boldsymbol{Y}^*)^{\mathsf{T}} \boldsymbol{\Lambda} (\boldsymbol{Z} - \boldsymbol{Y}^*)$$

  - only a subset of responses are observed, and the rest unobserved
  - $\boldsymbol{Y}^*$, imputed response, $Y_i^* = y_i$ or μ_y
  - first term: a smoothness penalty wrt $\boldsymbol{\Delta}$; penalizes vectors $\boldsymbol{Z}$ that differ too much over similar nodes
  - second term: error between the predicted and imputed response
  - an example: Tikhonov smoothing where $\boldsymbol{\Delta} = \boldsymbol{L}$

$$\boldsymbol{Z}^{\mathsf{T}} \boldsymbol{\Delta} \boldsymbol{Z} = \frac{1}{2} \sum_i \sum_j a_{ij} (z_i - z_j)^2$$

- various semi-supervised learning solutions:
  - random walk smoothing: $\boldsymbol{\Lambda} = \lambda \boldsymbol{I}$, $\boldsymbol{\Delta} = \tilde{\boldsymbol{L}}'$
  - Tikhonov smoothing: $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda \boldsymbol{I}_r, 0)$, $\boldsymbol{\Delta} = \boldsymbol{L}$
  - Undirected random walk smoothing: $\boldsymbol{\Lambda} = \lambda \boldsymbol{I}$, $\boldsymbol{\Delta} = \tilde{\boldsymbol{L}}$
  - kriging: (spatial) data adaptive way to estimate $\boldsymbol{\Delta}$, while the rest of solutions take a given function of the graph Laplacian

Network Link Analysis

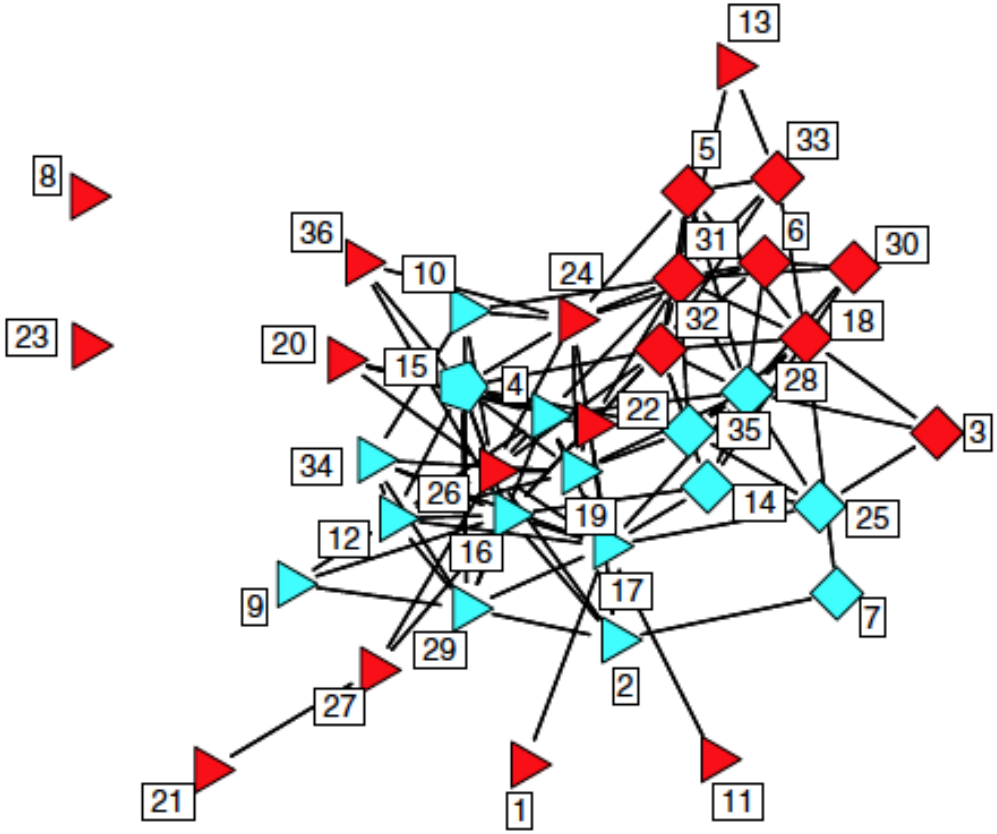
Network link analysis

- problem set up:
  - the network link information is only **partially known**, and the rest needs to be **inferred**
  - usually only **a single snapshot** of the network
- some key applications:
  - infer the link information among the nodes; usually determine if there is a link or not between a pair of nodes
  - covariance matrix / precision matrix estimation and inference — Gaussian graphical models
  - community detection
- case studies:
  - prediction of collaboration patterns among a group of lawyers and understand how are they affected by vertex and link features
  - prediction of protein-protein interactions

Modeling of network links

- case study 1:
  - data on lawyer collaboration in a law firm: 36 vertices / partners, and edges / $A = a$ worked together in a substantive manner
  - in addition, vertex features, including partner's seniority, gender, office location and type of practice (litigation or corporate law)
  - goal: understand collaborations among partners and the effect of vertex attributes while controlling for effects of network structure that might be anticipated
  - note that, $A = (A^{obs}; A^{miss})$

Modeling of network links



- shapes (triangle, diamond, or pentagon) indicate three office locations
- colors indicate the type of practice, litigation (red) or corporate (cyan)
- there are three female partners; the rest are male

Modeling of network links

- informal scoring method:
  - for each pair of vertices (i, j), compute a score $s(i, j)$, then apply a threshold, or rank
  - the larger the score value, the more likely for i and j to share an edge
  - some common score functions:

$$
\begin{array}{lcll}
s(i,j) & = & -\mathrm{dist}_{G^{obs}}(i,j) & \text{shortest-path distance} \\
s(i,j) & = & |\mathcal{N}_i^{ons} \cap \mathcal{N}_j^{obs}| & \text{number of common neighbors} \\
s(i,j) & = & \frac{|\mathcal{N}_i^{ons} \cap \mathcal{N}_j^{obs}|}{|\mathcal{N}_i^{ons} \cup \mathcal{N}_j^{obs}|} & \text{Jaccard coefficient}
\end{array}
$$

Modeling of network links

- logistic regression / classification model:
  - logistic regression:

$$\text{logic}(A_{ij}^{miss} = 1 | Z_{ij} = z) = \beta^{\mathsf{T}} z$$

    - Z_{ij} includes X_{ij}, combination of attributes on both nodes, and A^{obs}
    - dependencies among the observations (edges); missing value mechanism

  - logistic regression with latent variables (Hoff, 2005, 2007, 2008):

$$\text{logic}(A_{ij}^{miss} = 1 | Z_{ij} = z, M_{ij} = m) = \beta^{\mathsf{T}} z + m$$

    - M is an unknown, random, symmetric $N_v \times N_v$ matrix of latent variables
    - A_{ij} are conditionally independent, given Z_{ij} and M_{ij} , but are conditionally dependent given only Z_{ij}
    - Bayesian computations

Modeling of network links

- Markov random graph model:
  - a joint distribution for all elements in the random adjacency matrix A:

$$P(A = a) = \frac{1}{\kappa} \exp\left\{ \sum_{k=1}^{N_v - 1} \theta_k S_k(a) + \theta_\tau T(a) \right\}$$

    where $S_1(a) = N_e$, $S_k(a)$ is the number of k-stars, $T(a)$ is the number of triangles, and κ is the normalizing constant

  - to simplify and to take into account link attributes:

$$P(A = a) = \frac{1}{\kappa} \exp\left\{ \theta_1 S_1 + \theta_2 AKT_\lambda(a) + \beta^\mathsf{T} g(a, x) \right\}$$

    - $AKT_\lambda(a)$ is related to the number of triangles
    - $g(a, x) = \sum_{1 \le i < j \le N_v} a_{ij} h(x_i, x_j)$
    - for seniority, $h(x_i; x_j) = \text{seniority}_i + \text{seniority}_j$; for gender, office location, type of practice, $h(x_i; x_j) = I\{\text{gender}_i = \text{gender}_j\}$
    - likelihood based estimation; inference on β

Modeling of network links

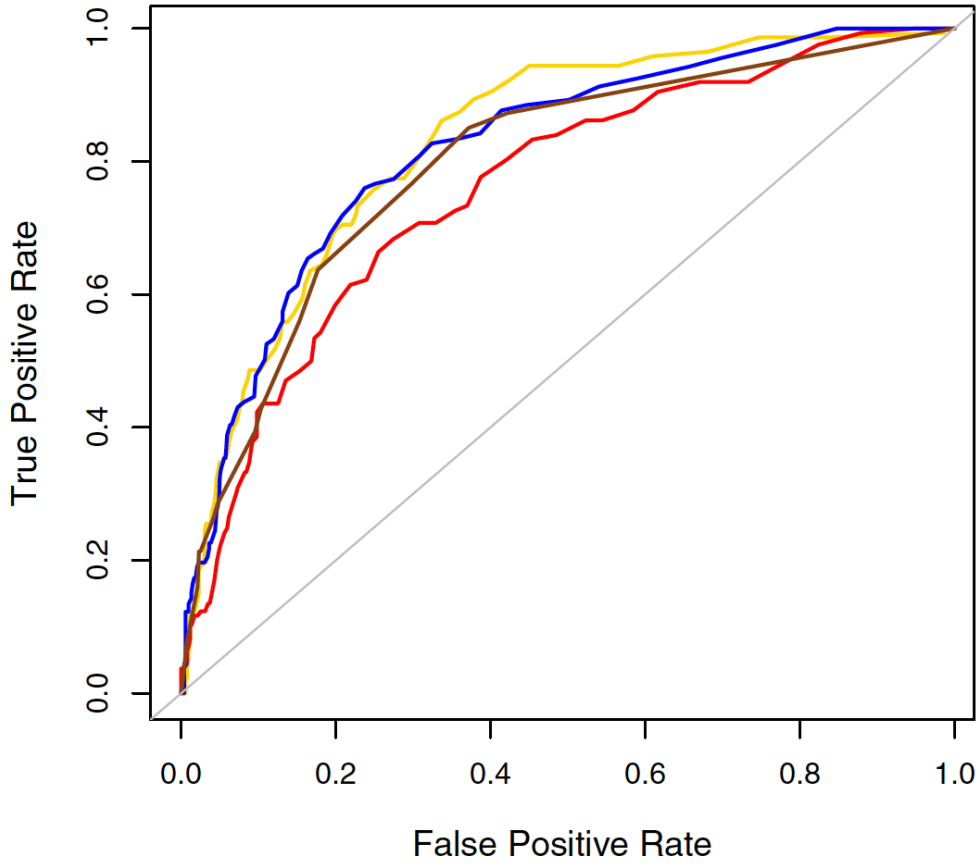


Figure: logistic regression (red), logistic regression with an addition covariate, the common number of neighbors $|\mathcal{N}_i^{ons} \cap \mathcal{N}_j^{obs}|$ (blue), informal scoring based on scores $s(i,j) = |\mathcal{N}_i^{ons} \cap \mathcal{N}_j^{obs}|$ (brown), and latent logistic regression (yellow)

Modeling of network links

- case study 2: Zhao et al. (2013, arXiv)
  - protein-protein interaction data
  - nodes: 984 proteins, each protein has a 325-dimensional (node covariate) vector of gene expression
  - links: partially observed 2438 (undirected) links
  - solution: obtain a relative ranking of potential links by their probabilities, utilizing information on node covariates as well as on network topology

$$\hat{f} = \arg\min_f \frac{1}{p^2}\sum_i\sum_j (a_{ij} - f_{ij})^2 + \lambda \frac{1}{p^4} W_{ii'} W_{jj'} \sum_i \sum_{i'} \sum_j \sum_{j'} (f_{ij} - f_{i'j'})^2,$$

where $\{W_{ij}\}_{i,j=1}^p$ is a similarity matrix, whose element measures the closeness between any pair of nodes. The links are predicted by choosing a threshold on the rank $\hat{f}_{ij}$

  - intuition: if (i,j) is close to (i',j'), then $W_{ii'}W_{jj'}$ will be large, forcing f_{ij} to be close to $f_{i'j'}$

Community detection

- **community detection**:
  - clustering on a single network, such that there are many connections within the community and relatively few connections between the communities
  - **stochastic block model**
  - **spectral clustering**: perform k-means clustering on a representation of the data; the representation is typically obtained by using the first eigenvectors of theLaplacian matrix of the graph that encodes the relationships between nodes

Gaussian Graphical Models

Gaussian graphical models

- problem set up:
  - the network link information is **completely unknown** and needs to be **inferred**
  - multiple samples, each has a vector of p features, and each feature forms a node of the network
- some key applications:
  - analysis of **covariance / correlation matrix** between nodes: $p \times p$; zero means the node pair are marginally independent under normality
  - analysis of **precision / partial correlation matrix** between nodes: $p \times p$; zero means the node pair are conditionally independent under normality given other nodes
- extensions:
  - multiple graphical models; dynamic graphical models
  - binary features; ordinal features; mixture features
  - semiparametric; nonparametric

Additional readings

- Kolaczyk, E.D. (2009). *Statistical Analysis of Network Data*. Chapters 2, 4, 6, 7, 8, 9, 10
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *Elements of Statistical Learning*. Springer. Chapter 17
- Rajaraman, A., Leskovec, J., and Ullman, J.D. (2012). *Mining of Massive Datasets*. Chapter 10
- Goldenberg, A., Zheng, A.X., Fienberg, S.E., and Airoldi, E.M. (2009). A survey of statistical network models. *Foundations and Trends in Machine Learning*, **2**, 129-233
- Jordan, M.I. (2004). Graphical models. *Statistical Sciences*, **19**, 140-155
- Zhao, Y., Levina, E. and Zhu, J. (2013), Link prediction for partially observed networks. *arXiv*.