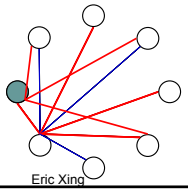
Advanced Machine Learning

Learning Graphical Models
Structure



Eric Xing

Lecture 22, April 12, 2010

Reading:

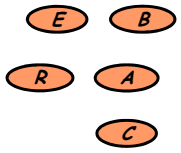
Inference and Learning

- A BN *M* describes a unique probability distribution *P*
- Typical tasks:
  - Task 1: How do we answer **queries** about P?
    - We use **inference** as a name for the process of computing answers to such queries
    - So far we have learned several algorithms for exact and approx. inference
  - Task 2: How do we estimate a **plausible model** M from data D?
    i. We use **learning** as a name for the process of obtaining point estimate of M.
    ii. But for *Bayesian*, they seek $p(M \mid D)$, which is actually an **inference** problem.
    iii. When not all variables are observable, even computing point estimate of M need to do **inference** to impute the *missing data*.

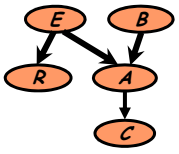
Learning Graphical Models

The goal:

Given set of independent samples (***assignments*** of random variables), find the ***best*** (the most likely?) graphical model (both the graph and the CPDs)



(B,E,A,C,R)=(T,F,F,T,F)
(B,E,A,C,R)=(T,F,T,T,F)
……..
(B,E,A,C,R)=(F,T,T,T,F)



E	B	P(A | E,B)	
e	b	0.9	0.1
e	$\bar{b}$	0.2	0.8
$\bar{e}$	b	0.9	0.1
$\bar{e}$	$\bar{b}$	0.01	0.99

Structural Search

- How many graphs over *n* nodes? $O(2^{n^2})$
- How many trees over *n* nodes? $O(n!)$
- But it turns out that we can find exact solution of an optimal tree (under MLE)!
  - Trick: in a tree each node has only one parent!
  - Chow-liu algorithm

Information Theoretic Interpretation of ML

$$
\begin{aligned}
\ell(\theta_G, G; D) &= \log p(D \mid \theta_G, G) \\
&= \log \prod_n \left(\prod_i p(x_{n,i} \mid \mathbf{x}_{n,\pi_i(G)}, \theta_{i\mid\pi_i(G)}) \right) \\
&= \sum_i \left(\sum_n \log p(x_{n,i} \mid \mathbf{x}_{n,\pi_i(G)}, \theta_{i\mid\pi_i(G)}) \right) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \frac{count(x_i, \mathbf{x}_{\pi_i(G)})}{M} \log p(x_i \mid \mathbf{x}_{\pi_i(G)}, \theta_{i\mid\pi_i(G)}) \right) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log p(x_i \mid \mathbf{x}_{\pi_i(G)}, \theta_{i\mid\pi_i(G)}) \right)
\end{aligned}
$$

From sum over data points to sum over count of variable states

Information Theoretic Interpretation of ML (con'd)

$$
\begin{aligned}
\ell(\theta_G, G; D) &= \log \hat{p}(D \mid \theta_G, G) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log \hat{p}(x_i \mid \mathbf{x}_{\pi_i(G)}, \theta_{i\mid\pi_i(G)}) \right) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log \frac{\hat{p}(x_i, \mathbf{x}_{\pi_i(G)}, \theta_{i\mid\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_i(G)})} \frac{\hat{p}(x_i)}{\hat{p}(x_i)} \right) \\
&= M \sum_i \left(\sum_{x_i, \mathbf{x}_{\pi_i(G)}} \hat{p}(x_i, \mathbf{x}_{\pi_i(G)}) \log \frac{\hat{p}(x_i, \mathbf{x}_{\pi_i(G)}, \theta_{i\mid\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_i(G)}) \hat{p}(x_i)} \right) - M \sum_i \left(\sum_{x_i} \hat{p}(x_i) \log \hat{p}(x_i) \right) \\
&= M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)}) - M \sum_i \hat{H}(x_i)
\end{aligned}
$$

Decomposable score and a function of the graph structure

Chow-Liu tree learning algorithm

- Objection function:

$$\ell(\theta_G, G; D) = \log \hat{p}(D \mid \theta_G, G) = M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)}) - M \sum_i \hat{H}(x_i) \quad \Rightarrow \quad \boxed{C(G) = M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)})}$$

- Chow-Liu:
  - For each pair of variable x_i and x_j
    - Compute empirical distribution: $\hat{p}(X_i, X_j) = \frac{count(x_i, x_j)}{M}$
    - Compute mutual information: $\hat{I}(X_i, X_j) = \sum_{x_i, x_j} \hat{p}(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)}$
  - Define a graph with node $x_1, \ldots, x_n$
    - Edge (I,j) gets weight $\hat{I}(X_i, X_j)$

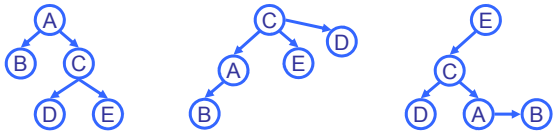
Chow-Liu algorithm (con'd)

- Objection function:

$$\ell(\theta_G, G; D) = \log \hat{p}(D \mid \theta_G, G) = M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)}) - M \sum_i \hat{H}(x_i) \quad \Rightarrow \quad \boxed{C(G) = M \sum_i \hat{I}(x_i, \mathbf{x}_{\pi_i(G)})}$$

- Chow-Liu:

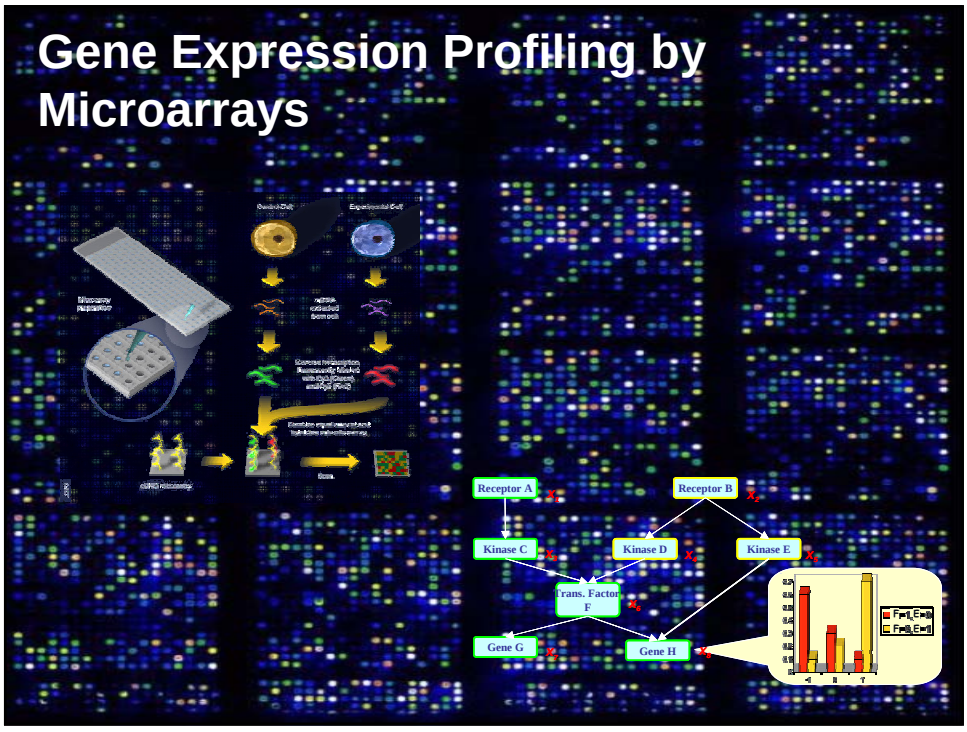
  Optimal tree BN
  - Compute maximum weight spanning tree
  - Direction in BN: pick any node as root, do breadth-first-search to define directions
  - I-equivalence:



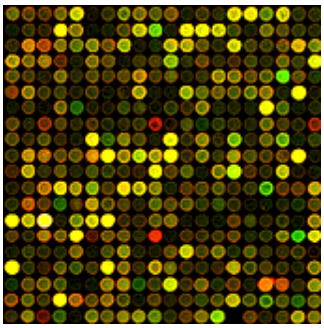
$$C(G) = I(A,B) + I(A,C) + I(C,D) + I(C,E)$$

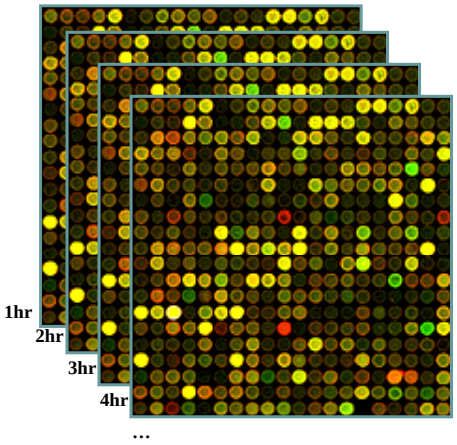
Structure Learning for general graphs

- Theorem:
  - The problem of learning a BN structure with at most *d* parents is NP-hard for any (fixed) *d*≥2
- Most structure learning approaches use heuristics
  - Exploit score decomposition
  - Two heuristics that exploit decomposition in different ways
    - Greedy search through space of node-orders
    - Local search of graph structures

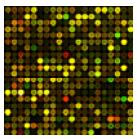


Microarray Data



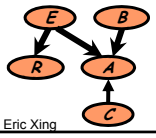


Structure Learning Algorithms



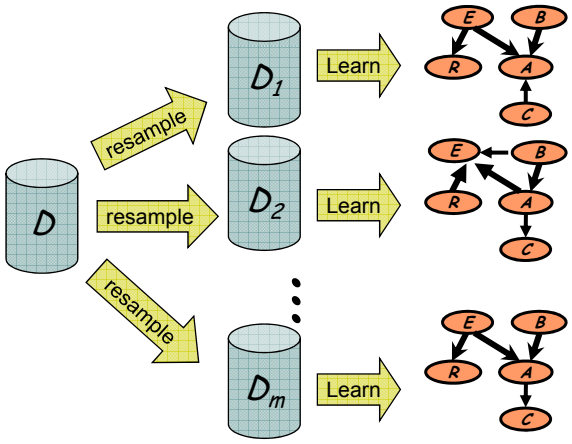
Expression data

Learning Algorithm



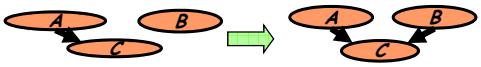
- Structural EM (Friedman 1998)
  - The original algorithm
- Sparse Candidate Algorithm (Friedman et al.)
  - Discretizing array signals
  - Hill-climbing search using local operators: add/delete/swap of a single edge
  - Feature extraction: Markov relations, order relations
  - Re-assemble high-confidence sub-networks from features
- Module network learning (Segal et al.)
  - Heuristic search of structure in a "module graph"
  - Module assignment
  - Parameter sharing
  - Prior knowledge: possible regulators (TF genes)

Scoring Networks



Learning GM structure

- Learning of best CPDs *given DAG* is easy
  - collect statistics of values of each node given specific assignment to its parents
- Learning of the graph topology (structure) is NP-hard
  - heuristic search must be applied, generally leads to a **locally** optimal network
- Overfitting
  - It turns out, that richer structures give higher likelihood P(D|G) to the data (adding an edge is always preferable)



$$P(C \mid A) \leq P(C \mid A, B)$$

  - more parameters to fit => more freedom => always exist more "optimal" CPD(C)
- We prefer *simpler* (more explanatory) networks
  - **Practical** scores **regularize** the likelihood improvement complex networks.

Gaussian Graphical Model

- Multivariate Gaussian density:

$$p(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left\{-\tfrac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right\}$$

- WOLG: let $\mu = 0 \quad Q = \Sigma^{-1}$

$$p(x_1, x_2, \cdots, x_p \mid \mu = 0, Q) = \frac{|Q|^{1/2}}{(2\pi)^{n/2}} \exp\left\{-\tfrac{1}{2}\sum_i q_{ii}(x_i)^2 - \sum_{i<j} q_{ij} x_i x_j\right\}$$

- We can view this as a continuous Markov Random Field with potentials defined on every node and edge:

The covariance and the precision matrices

- Covariance matrix Σ

$$\Sigma_{i,j} = 0 \quad \Rightarrow \quad X_i \perp X_j \quad \text{or} \quad p(X_i, X_j) = p(X_i)p(X_j)$$

  - Graphical model interpretation?

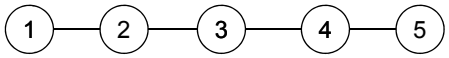
- Precision matrix $Q = \Sigma^{-1}$

$$Q_{i,j} = 0 \quad \Rightarrow \quad X_i \perp X_j | \mathbf{X}_{-ij} \quad \text{or} \quad p(Xi, X_j | \mathbf{X}_{-ij}) = p(X_i | \mathbf{X}_{-ij}) p(X_j | \mathbf{X}_{-ij})$$

  - Graphical model interpretation?

- How to prove the later?

Sparse precision vs. sparse covariance in GGM



$$\Sigma^{-1} = \begin{pmatrix} 1 & 6 & 0 & 0 & 0 \\ 6 & 2 & 7 & 0 & 0 \\ 0 & 7 & 3 & 8 & 0 \\ 0 & 0 & 8 & 4 & 9 \\ 0 & 0 & 0 & 9 & 5 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 0.10 & 0.15 & -0.13 & -0.08 & 0.15 \\ 0.15 & -0.03 & 0.02 & 0.01 & -0.03 \\ -0.13 & 0.02 & 0.10 & 0.07 & -0.12 \\ -0.08 & 0.01 & 0.07 & -0.04 & 0.07 \\ 0.15 & -0.03 & -0.12 & 0.07 & 0.08 \end{pmatrix}$$

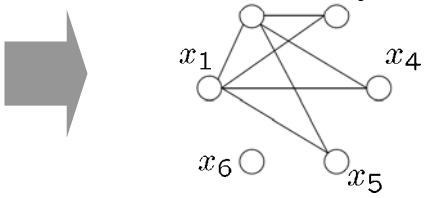
$$\Sigma^{-1}_{15} = 0 \Leftrightarrow X_1 \perp X_5 \,|\, X_{nbrs(1) \text{ or } nbrs(5)}$$

$$\not\Rightarrow$$

$$X_1 \perp X_5 \Leftrightarrow \Sigma_{15} = 0$$

Another example

$$Q = \begin{pmatrix} * & * & * & * & * & 0 \\ * & * & * & * & * & 0 \\ * & * & * & 0 & 0 & 0 \\ * & * & 0 & * & 0 & 0 \\ * & * & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & * \end{pmatrix}$$



- How to estimate this MRF?
- What if $p >> n$
  - MLE does not exist in general!
  - What about only learning a "sparse" graphical model?
    - This is possible when $s=o(n)$
    - Very often it is the structure of the GM that is more interesting …

Learning (sparse) GGM

- Multivariate Gaussian over all continuous expressions

$$p([x_1,...,x_n]) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left\{-\tfrac{1}{2}(\vec{x}-\mu)^T \Sigma^{-1}(\vec{x}-\mu)\right\}$$

- The precision matrix $K=\Sigma^{-1}$ reveals the topology of the (undirected) network

$$E(x_i \mid x_{-i}) = \sum_j (K_{ij} / K_{ii}) x_j$$

  - Edge ~ $|K_{ij}| > 0$

- Learning Algorithm: Covariance selection
  - Want a sparse matrix
    - Regression for each node with degree constraint (Dobra et al.)
    - Regression for each node with hierarchical Bayesian prior (Li, et al)
    - Graphical Lasso (we will describe it shortly)

Learning Ising Model (i.e. pairwise MRF)

- Assuming the nodes are discrete, and edges are weighted, then for a sample x_d, we have

$$P(\mathbf{x}_d|\Theta) = \exp\left(\sum_{i\in V}\theta_{ii}^t x_{d,i} + \sum_{(i,j)\in E}\theta_{ij}x_{d,i}x_{d,j} - A(\Theta)\right)$$

- **Graph lasso** has been used to obtain a sparse estimate of *E* with continuous *X*

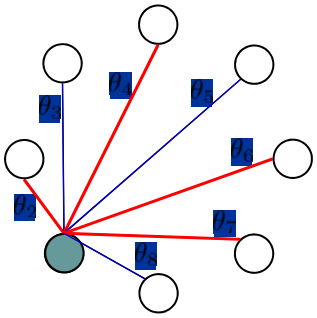
- We can use graphical L_1 regularized logistic regression to obtain a sparse estimate of with discrete *X*

Recall lasso

$$\hat{\theta}_i \quad = \quad \arg\min_{\theta_i} l(\theta_i) + \lambda_1 \| \ \theta_i \ \|_1$$

where $l(\theta_i) = \log P\big(y_i|\mathbf{x}_i, \theta_i\big).$

Graph Regression

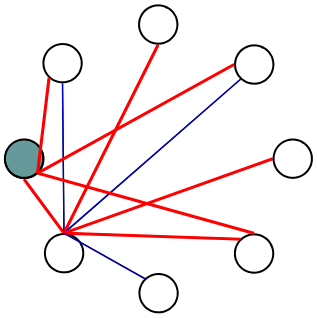


Neighborhood selection

Lasso:

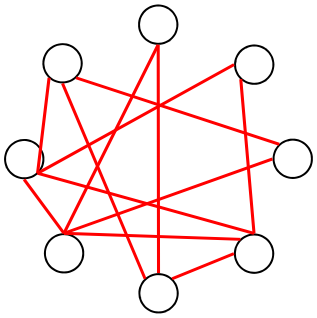
$$\hat{\theta} \quad = \quad \arg\min_{\theta} \sum_{i=1}^{T} l(\theta) + \lambda_1 \| \ \theta \ \|_1$$

Graph Regression



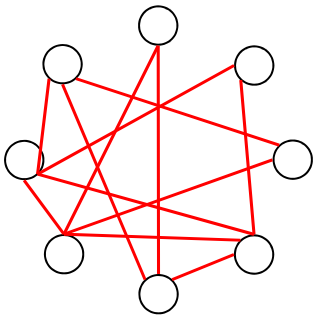
Neighborhood selection

Graph Regression



Neighborhood selection

Why this is reasonable?



Single-node Conditional

- The conditional dist. of a single node i given the rest of the nodes can be written as:

$$p(X_i|\mathbf{X}_{-i}) = \mathcal{N}\big(\mu_i + \Sigma_{X_i\mathbf{X}_{-i}}\Sigma^{-1}_{\mathbf{X}_{-i}\mathbf{X}_{-i}}(\mathbf{X}_{-i} - \mu_{\mathbf{x}_{-i}}), \\ \Sigma_{X_iX_i} - \Sigma_{X_i\mathbf{X}_{-i}}\Sigma^{-1}_{\mathbf{X}_{-i}\mathbf{X}_{-i}}\Sigma_{\mathbf{X}_{-i}X_i}\big)$$

- WOLG: let $\mu = 0$

$$\begin{aligned} p(X_i|\mathbf{X}_{-i}) &= \mathcal{N}\big(\Sigma_{X_i\mathbf{X}_{-i}}\Sigma^{-1}_{\mathbf{X}_{-i}\mathbf{X}_{-i}}\mathbf{X}_{-i}, \Sigma_{X_iX_i} - \Sigma_{X_i\mathbf{X}_{-i}}\Sigma^{-1}_{\mathbf{X}_{-i}\mathbf{X}_{-i}}\Sigma_{\mathbf{X}_{-i}X_i}\big) \\ &= \mathcal{N}\big(\vec{\sigma}_i^T\Sigma^{-1}_{-i}\mathbf{X}_{-i}, q_{ii}\big) \\ &= \mathcal{N}\big(\frac{\vec{q}_i^T}{-q_{ii}}\mathbf{X}_{-i}, q_{ii}\big) \end{aligned}$$

Conditional auto-regression

- From

$$p(X_i|\mathbf{X}_{-i}) = \mathcal{N}\left(\frac{\vec{q}_i^T}{-q_{ii}}\mathbf{X}_{-i}, q_{ii}\right)$$

- We can write the following conditional auto-regression function for each node:

- Neighborhood est. based on auto-regression coefficient

$$S_i \equiv \{j \quad : \quad j \neq i, \theta_{ij} \neq 0\}$$

Conditional independence

- From

$$p(X_i|\mathbf{X}_{-i}) = \mathcal{N}\left(\frac{\vec{q}_i^T}{-q_{ii}}\mathbf{X}_{-i}, q_{ii}\right)$$

- Given an estimate of the neighborhood s_i, we have:

$$p(X_i|\mathbf{X}_{-i}) = p(X_i|\mathbf{X}_s)$$

- Thus the neighborhood s_i defines the Markov blanket of node i

Consistency

- **Theorem**: for the graphical regression algorithm, under certain verifiable conditions (omitted here for simplicity):

$$\mathbb{P}\left[\hat{G}(\lambda_n) \neq G\right] = \mathcal{O}\left(\exp\left(-Cn^{\epsilon}\right)\right) \to 0$$

Note the from this theorem one should see that the regularizer is not actually used to introduce an "artificial" sparsity bias, but a devise to ensure consistency under finite data and high dimension condition.

Recent trends in GGM:

- Covariance selection (classical method)
  - Dempster [1972]:
    - Sequentially pruning smallest elements in precision matrix
  - Drton and Perlman [2008]:
    - Improved statistical tests for pruning

Serious limitations in practice: breaks down when covariance matrix is not invertible

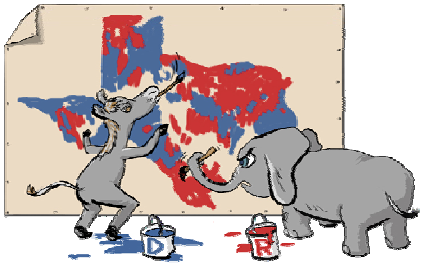
- L_1-regularization based method (*hot* !)
  - Meinshausen and Bühlmann [Ann. Stat. 06]:
    - Used LASSO regression for neighborhood selection
  - Banerjee [JMLR 08]:
    - Block sub-gradient algorithm for finding precision matrix
  - Friedman et al. [Biostatistics 08]:
    - Efficient fixed-point equations based on a sub-gradient algorithm
  - …

Structure learning is possible even when # variables > # samples

Learning GM

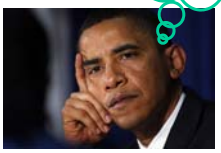
- Learning of best CPDs *given DAG* is easy
  - collect statistics of values of each node given specific assignment to its parents
- Learning of the graph topology (structure) is NP-hard
  - heuristic search must be applied, generally leads to a **locally** optimal network
- We prefer *simpler* (more explanatory) networks
  - **Regularized graph regression**

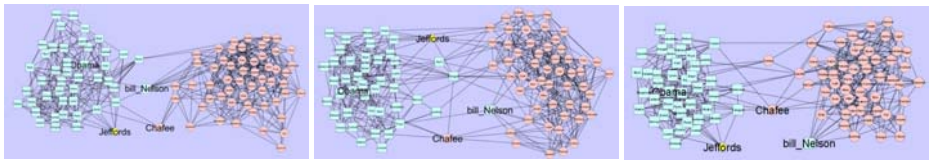
New Problem: Evolving Social Networks



Can I get his vote?

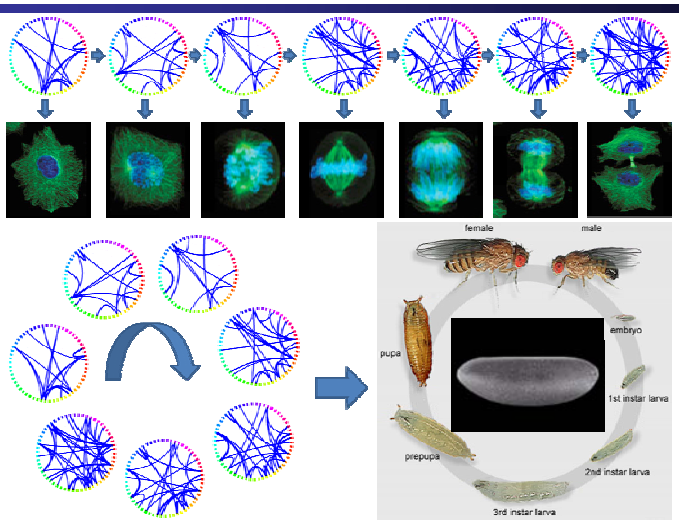
Corporativity,
Antagonism,
Cliques,
…
over time?





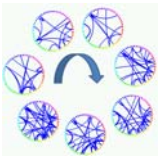
March 2005 January 2006 August 2006

Time-Varying Gene Regulations

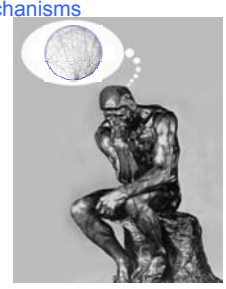


Departing from invariant GM est.

- Existing work:
  - Assuming networks or network time series are observable and given



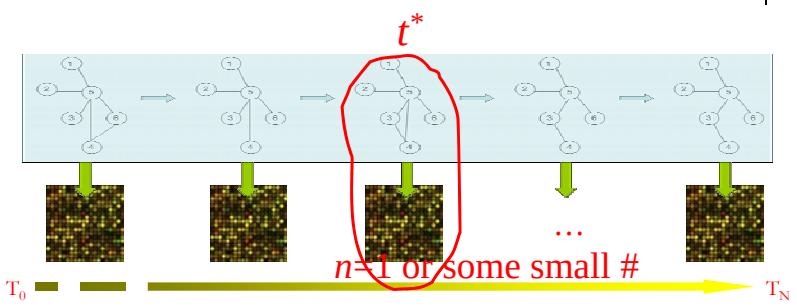
  - Then model/analyze the generative and/or dynamic mechanisms



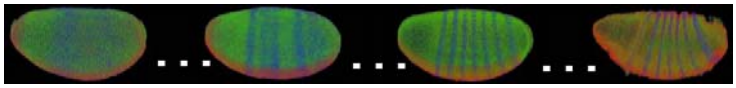
- We assume:
  - Networks are not observable
  - So we need to **INFER** the networks from nodal attributes before analyzing them

$$\mathcal{D} = \{x_1^i, \ldots, x_p^i\}_{i=1}^n \quad \Rightarrow \quad G_1, \ldots, G_n$$

Reverse engineer temporal/spatial-specific "rewiring" gene networks



Drosophila development



Challenges

- Very small sample size
  - observations are scarce and costly
- Noisy data
- Large dimensionality of the data
  - usually $p \gg n$
  - complexity regularization is required to avoid curse of dimensionality, e.g. sparsity
- And now the data are non-iid since underlying probability distribution is changing !

Inference I

[Song, Kolar and Xing, Bioinformatics 09]

- **KELLER**: Kernel Weighted L_1-regularized Logistic Regression

$$\hat{\theta}_i^t = \arg\min_{\theta_i^t} l_w(\theta_i^t) + \lambda_1 \| \theta_{-i}^t \|_1 \quad \forall t$$

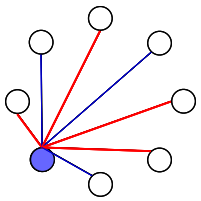
$$\text{where } l_w(\theta_i^t) = \sum_{t'=1}^{T} w(\mathbf{x}^{t'}; \mathbf{x}^t) \log P(x_i^{t'} | \mathbf{x}_{-i}^{t'}, \theta_i^t).$$

- Constrained convex optimization
  - Estimate time-specific one by one
  - Could scale to ~10^4 genes, but under stronger smoothness assumptions

Algorithm - neighborhood selection

- Conditional likelihood

$$\mathbb{P}_{\theta^t}(x_u^t | x_{\setminus u}^t) \propto \exp\left(2 x_u^t \left\langle \theta_{\setminus u}^t, x_{\setminus u}^t \right\rangle\right).$$

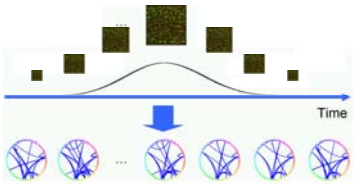


- Neighborhood: $S(x_u) = \{j \mid \theta_{u,j}^t \neq 0\}$
- Estimate at $t^* \in [0,1]$

$$\min_{\theta \in \mathbb{R}^{pn-1}} \left\{ -\sum_{t \in \mathcal{T}^n} w_t(t^*) \gamma(\theta; x^t) + \lambda_1 \|\theta\|_1 \right\}$$

Where $\gamma(\theta^t; x^t) = \log \mathbb{P}_{\theta^t}(x_u^t | x_{\setminus u}^t)$

and $w_t(t^*) = \dfrac{K_{h_n}(t - t^*)}{\sum_{t' \in \mathcal{T}^n} K_{h_n}(t' - t^*)}$



Structural consistency of KELLER

Assumptions

- Define: $Q_u^t := \mathbb{E}\left[\nabla^2 \log \mathbb{P}_{\theta^t}[X_u|X_{\backslash u}]\right], \quad \forall u \in V \qquad \Sigma_u^t := \mathbb{E}\left[X_{\backslash u}^t X_{\backslash u}^{t\,T}\right], \quad \forall u \in V$

  $s = \max_u \max_t |S_u^t|, \quad \theta_{\min} = \min_{e \in E} \max |\theta_e^t|$

- A1: Dependency Condition

  $\Lambda_{\min}(Q_{SS}^{t^*}) \geq C_{\min}, \quad \forall t \in [0,1]$

  $\Lambda_{\max}\left(\Sigma^{t^*}\right) \leq D_{\max}, \quad \forall t \in [0,1]$

- A2: Incoherence Condition $\exists \alpha \in (0,1]$ such that

  $\left\|Q_{S^cS}^{t^*}(Q_{SS}^{t^*})^{-1}\right\|_\infty \leq 1-\alpha, \quad \forall t^* \in [0,1]$

- A3: Smoothness Condition

  $\max_{u,v} \sup_{t^*} |\sigma'_{uv}(t^*)| \leq A_0, \quad \max_{u,v} \sup_{t^*} |\sigma''_{uv}(t^*)| \leq A$

  $\max_{u,v} \sup_{t^*} |\theta'_{uv}(t^*)| \leq B_0, \quad \max_{u,v} \sup_{t^*} |\theta''_{uv}(t^*)| \leq B$

- A4: Bounded Kernel

  $\exists M_k \geq 1 \qquad \max_{z \in \mathbb{R}} |K(z)| \leq M_k \quad \max_{z \in \mathbb{R}} K(z)^2 \leq M_k$

Theorem

[Kolar and Xing, 09]

Assume that A1, A2, A3, A4 hold. Furthermore, assume that the following conditions hold:

1. $h_n = \mathcal{O}(n^{-\frac{1}{3}})$
2. $s_n h_n = o(1)$,
3. $\frac{s_n^3 \log p_n}{n h_n} = o(1)$
4. $\lambda_1 = \mathcal{O}(\sqrt{\frac{\log p}{n h_n}})$
5. $\theta^*_{\min} = \Omega(\sqrt{\frac{s_n \log p_n}{n h_n}})$

then

$$\mathbb{P}\left[\hat{G}(\lambda_1, h_n, t^*) \neq G^{t^*}\right] = \mathcal{O}\left(\exp\left(-C\frac{n h_n}{s_n^3} + C' \log p\right)\right) \to 0$$

Inference II

[Ahmed and Xing, PNAS 09]

- **TESLA**: Temporally Smoothed L_1-regularized logistic regression

$$\hat{\theta}_i^1, \ldots, \hat{\theta}_i^T = \arg\min_{\theta_i^1,\ldots,\theta_i^T} \sum_{t=1}^{T} l_{avg}(\theta_i^t) + \lambda_1 \sum_{t=1}^{T} \| \theta_{-i}^t \|_1 + \lambda_2 \sum_{t=2}^{T} \| \theta_i^t - \theta_i^{t-1} \|_q^q,$$

$$\text{where } l_{avg}(\theta_{\mathbf{i}}^{\mathbf{t}}) = \frac{1}{N^t} \sum_{d=1}^{N^t} \log P\left(x_{d,i}^t | \mathbf{x}_{\mathbf{d},-\mathbf{i}}^{\mathbf{t}}, \theta_{\mathbf{i}}^{\mathbf{t}}\right).$$

- Constrained convex optimization
  - Scale to ~5000 nodes, does not need smoothness assumption, can accommodate abrupt changes.

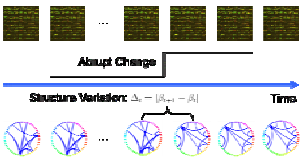
Modified estimation procedure

- estimate block partition on which the coefficient functions are constant

$$\min_{\beta} \sum_{i=1}^{n} (Y_i - \mathbf{X}_i \beta(t_i))^2 + 2\lambda_2 \sum_{k=1}^{p} ||\beta_k||_{\mathrm{TV}} \qquad (*)$$

- estimate the coefficient functions on each block of the partition

$$\min_{\gamma \in \mathbb{R}^p} \sum_{t_i \in \hat{j}} (Y_i - \mathbf{X}_i \gamma)^2 + 2\lambda_1 ||\gamma||_1 \qquad (**)$$



Structural Consistency of TESLA

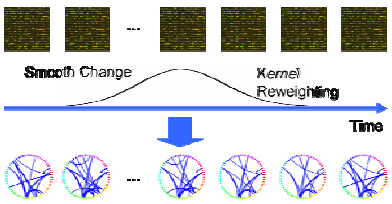
[Kolar and Xing, NIPS2009]

I. It can be shown that, by applying the results for model selection of the randomized Lasso on a *temporal difference transformation* of (*), **the block are estimated consistently**

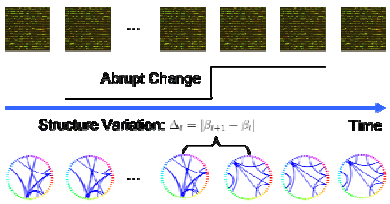
II. Then it can be further shown that, by applying Lasso on (**), **the neighborhood of each node on each of the estimated blocks consistently**

- Further advantages of the two step procedure
  - choosing parameters easier
  - faster optimization procedure

Two Scenarios

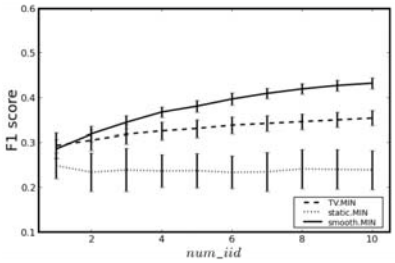


Smoothly evolving graphs

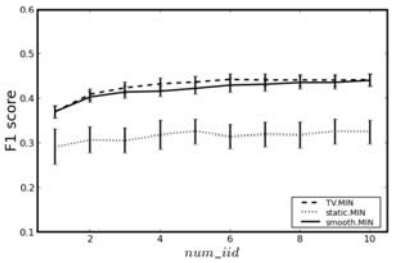


Abruptly evolving graphs

Comparison of KELLER and TESLA

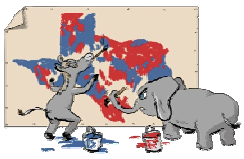


Smoothly varying



Abruptly varying

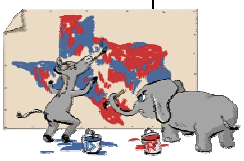
Senate network – 109th congress

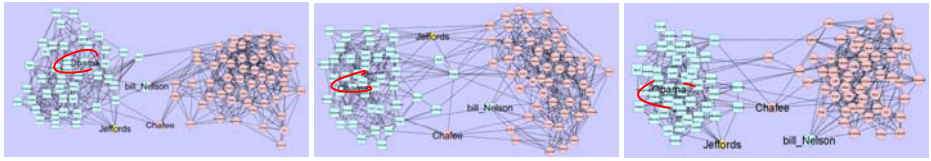


- Voting records from 109th congress (2005 - 2006)
- There are 100 senators whose votes were recorded on the 542 bills, each vote is a binary outcome

- Estimating parameters:
  - KELLER: bandwidth parameter to be $h_n = 0.174$, and the penalty parameter $\lambda_1 = 0.195$
  - TESLA: $\lambda_1 = 0.24$ and $\lambda_2 = 0.28$

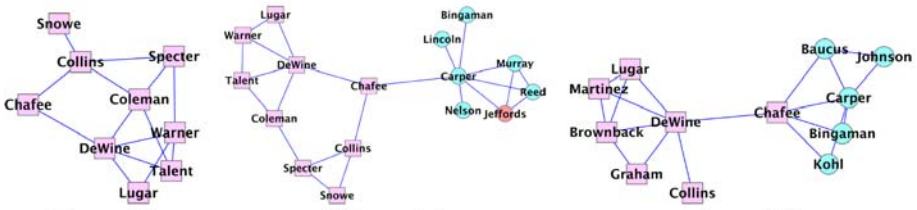
Senate network – 109th congress





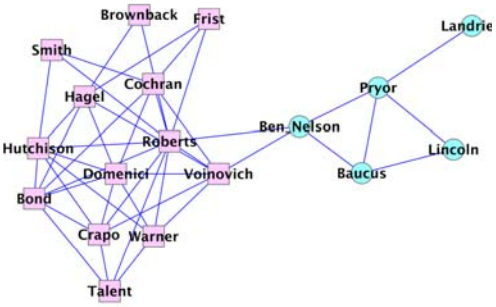
March 2005 January 2006 August 2006

Senator Chafee

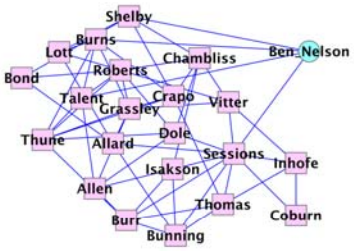


(a) t = 0.1 (b) t = 0.4 (c) t = 0.8

Senator Ben Nelson



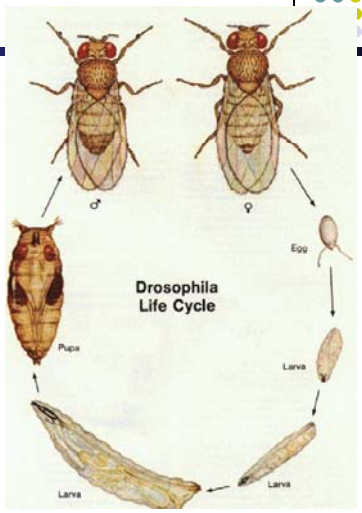
T=0.2



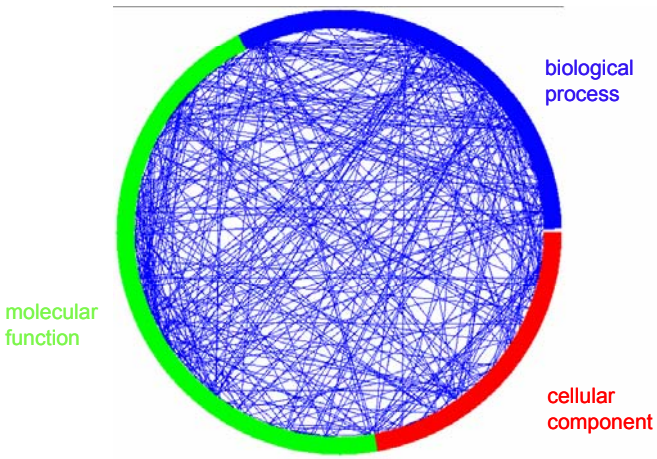
T=0.8

Drosophila life cycle

- From Arbeitman et al. (2002)
- Four stages:
  - embryo, larva, pupa, adult
- 66 microarray measured across full life cycle
- Focus on 588 development related genes



Dynamic Gene Interactions Networks of *Drosophila Melanogaster*



Summary

- Graphical Gaussian Model
  - The precision matrix encode structure
  - Not estimatable when p >> n
- Neighborhood selection:
  - Conditional dist under GGM
  - Graphical lasso
  - Sparsistency
- Time-varying GGM
  - Kernel reweighting est.
  - Total variation est.