7.90J 6.874J Computational functional genomics

(Spring 2005: Lecture 13)

David K. Gifford
(Adapted from a lecture by Tommi S. Jaakkola)
MIT CSAIL

Topics

- Modeling Biological Systems
  - A simple biological system
  - Model assumptions
- Discrete Bayesian networks
  - Discretizing data
  - Bayesian scoring functions
  - Edge scores

A simple biological system

- Data are observed from a biological subnetwork with four genes
- The genes might influence one anothers expression
- The structure of the network is hidden from us (we do not see the edges)
- We observe mRNA and active protein levels for each of the four genes
- We have hundreds of observations

Model assumptions

- There are no hidden variables (only A, B, C, and D can influence one another)
- There are no cycles in the unknown network
- "Sufficient conditions" are observed to perturb the expession of A, B, C, and D
- All observed data are continuous
- Data is complete (no missing variable observations)
- The underlying biological system can be modeled using discrete states
- Uniform population behavior (Why is this important?)
- We begin with 8 nodes...

Bayesian networks

- Nodes represent variables
- Each node has 0 or more parents
- The structure S of edges describes how the joint probability distribution of the observed variables can be factored
- S encodes the conditional independence of the observed variables
- To fully specify a network we need to specify how children depend on their parents
- This dependency is encoded in the parameters θ
- Given n variables, roughly how many structures are there?
- Less than $(2^n)^n = 2^{n^2}$ (Great!)

Bayesian network tasks

- Learn the structure of a Bayesian network (S) given observed data (Structure Learning)
- Learn a Bayesian network (S and θ) given observed data (Learning)
- Infer X_j when it is not observed given S and θ (Inference)

Discrete Bayesian Networks - Interval discretization

- Sort the observed values from smallest to largest
- Divide range of observed values into L intervals
- Policy vector

$$\Lambda = (-\infty, x_0 + \frac{(x_{N-1} - x_0)}{L}, x_0 + \frac{2(x_{N-1} - x_0)}{L}, \ldots, \quad (1)$$

$$x_0 + \frac{(L-1)(x_{N-1} - x_0)}{L}, \infty) \quad (2)$$

Quantile discretization

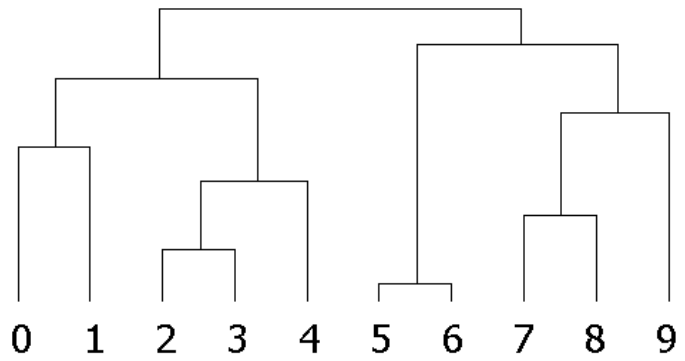
- Place an equal number of observations into L levels
- Policy vector

$$\Lambda = (-\infty, \frac{x_{\lfloor \frac{N}{L} \rfloor} + x_{\lfloor \frac{N}{L} \rfloor + 1}}{2}, \frac{x_{\lfloor \frac{2N}{L} \rfloor} + x_{\lfloor \frac{2N}{L} \rfloor + 1}}{2}, \ldots, \tag{3}$$

$$\frac{x_{\lfloor \frac{(L-1)N}{L} \rfloor} + x_{\lfloor \frac{(L-1)N}{L} \rfloor + 1}}{2}, \infty) \tag{4}$$

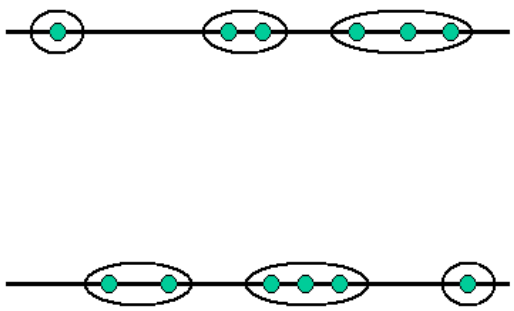
How do we decide on the number of levels?

- We can begin with one level for each unique observed value
- If we start with L levels of discretization, we can reduce this to $L - 1$ by coalescing levels
- Coalese two levels by adding the probabilities of the merged levels
- For example, we could start with 10 levels for 10 observations, and then reduce this to $L = 1$



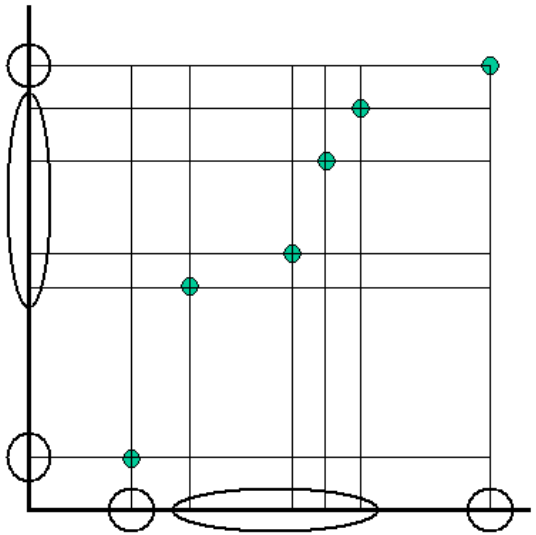
How should we merge levels?

- We could consider variables independently



How should we merge levels?

- Or consider preserving information between genes



Total mutual information between variables

- Let vector X_i^L be the discretization of variable X_i from all observations into L levels
- Define the total mutual information between all X_i^L at discretization level L as:

$$TMI(L) = \sum_{i,j} H(X_i^L) + H(X_j^L) - H(X_i^L, X_j^L) \tag{5}$$

- Mutual information is 0 when variables are independent
- $H(X_i^L)$ is a measure of the randomness of X_i^L

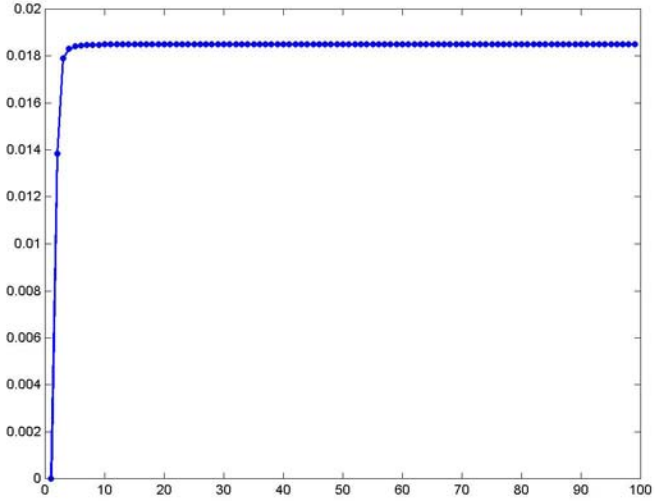
$$H(X_i^L) = -\sum_{X_i^L} p(X_i^L) log(p(X_i^L)) \tag{6}$$

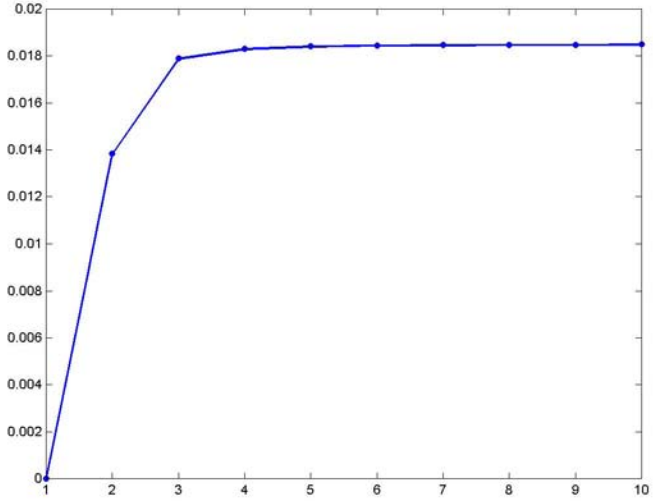
- $H(X_i^L, X_j^L)$ is the mutual entropy of X_i^L and X_j^L

$$H(X_i^L, X_j^L) = -\sum_{X_i^L, X_j^L} p(X_i^L, X_j^L) log(p(X_i^L, X_j^L)) \tag{7}$$

Total mutual information as a function of L

- When we go from L to $L-1$, pick the levels to merge to minimize $TMI(L) - TMI(L-1)$
- As we decrease L, $TMI(L)$ decreases:





- Pick an L that captures most of the information
- Why do we want to reduce L?

The Bayesian scoring metric

- The Bayesian score of model S given observed data D can be decomposed into a likelihood and a prior

$$\begin{aligned} BayesianScore(S) &= \log p(S|D) && (8) \\ &= \log p(S) + \log p(D|S) + c && (9) \end{aligned}$$

- The likelihood function is computed as follows

$$\begin{aligned} p(D|S) &= \int_\theta p(D,\theta|S)\mathrm{d}\theta && (10) \\ &= \int_\theta p(D|\theta,S)p(\theta|S)\mathrm{d}\theta && (11) \end{aligned}$$

Parameters for discrete Bayesian networks

- Index the n variables in the Bayesian network using the variable i
- Index the q_i parent configurations of variable i using the variable j
- Index the r_i states of variable i using the variable k
- θ_{ijk} is the probability of observing variable i in state k given parent configuration j

$$(\theta_{ij1}, \ldots, \theta_{ijr_i}) \sim \text{Dirichlet}(\alpha_{ij1}, \ldots, \alpha_{ijr_i}) \qquad \forall i, j \qquad (12)$$

$$\sim c \cdot \theta_{ij1}^{\alpha_{ij1}-1} \theta_{ij2}^{\alpha_{ij2}-1} \ldots \theta_{ijr_i}^{\alpha_{ijr_i}-1} \qquad (13)$$

- α are the hyperparameters

Scoring discrete Bayesian networks

- Assign each observation to a single level
- Let N_{ijk} be the number of occurrences in the data set D of variable i in state k given parent configuration j and

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \tag{14}$$

$$\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk} \tag{15}$$

- The Bayesian score of S (see Heckerman on the Web site) is:

$$\log p(S) + \log \left\{ \prod_{i=1}^{n} \prod_{j=1}^{q_i} \left(\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right) \right\} \tag{16}$$

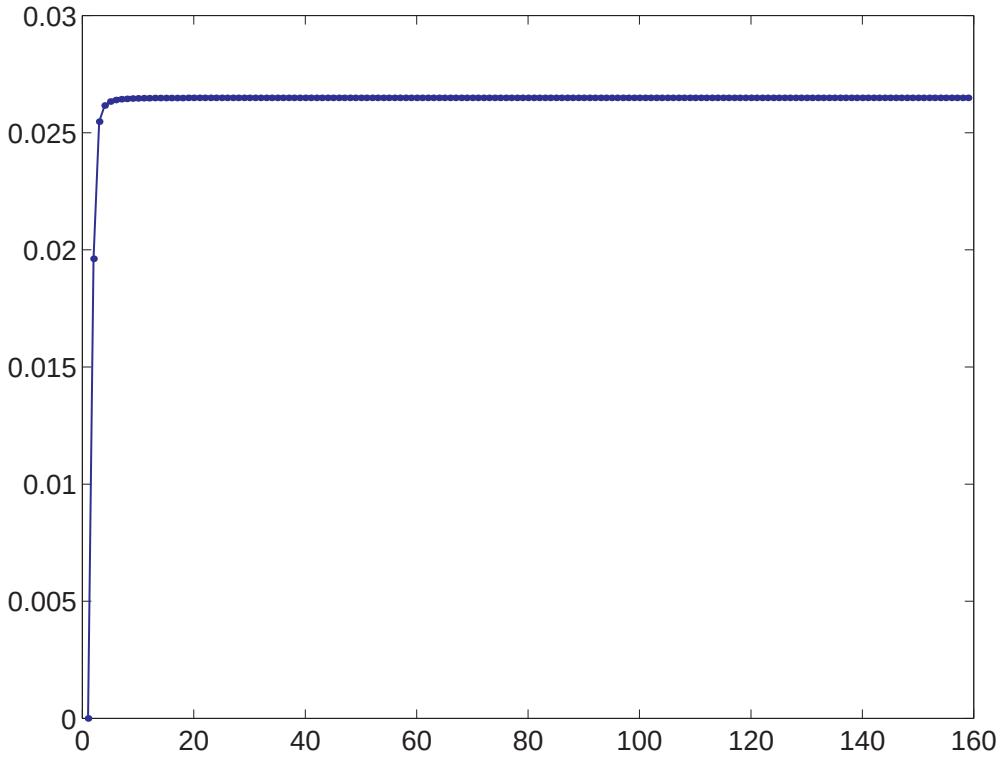
$$\log p(S) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left\{ \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right\} \tag{17}$$

Example: Yeast pheromone response pathway

Image removed for copyright reasons.

Total mutual information as a function of L

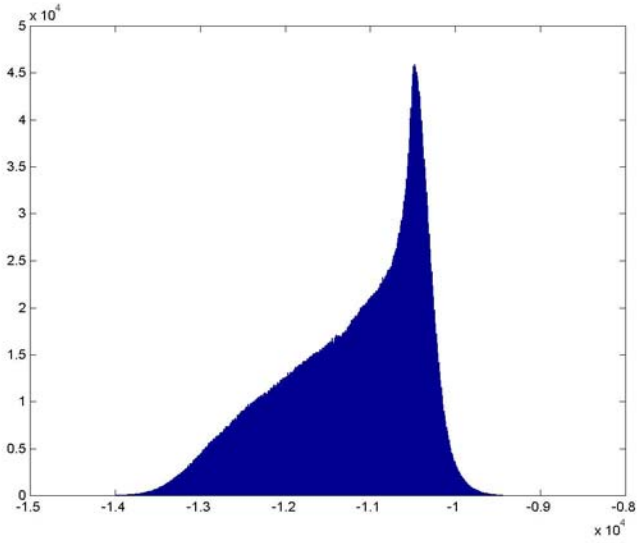
- We start with 320 experiments with $L = 160$ and run the level merging algorithm that minimizes the loss in total mutual information

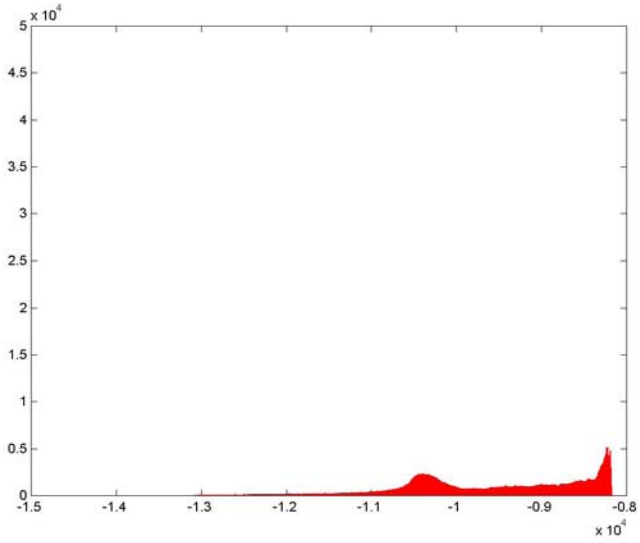


- $L = 4$ for scoring

Model search: distribution of model scores

Histogram of model scores using a random walk and simulated annealing. Note that simulated annealing does not get stuck as easily





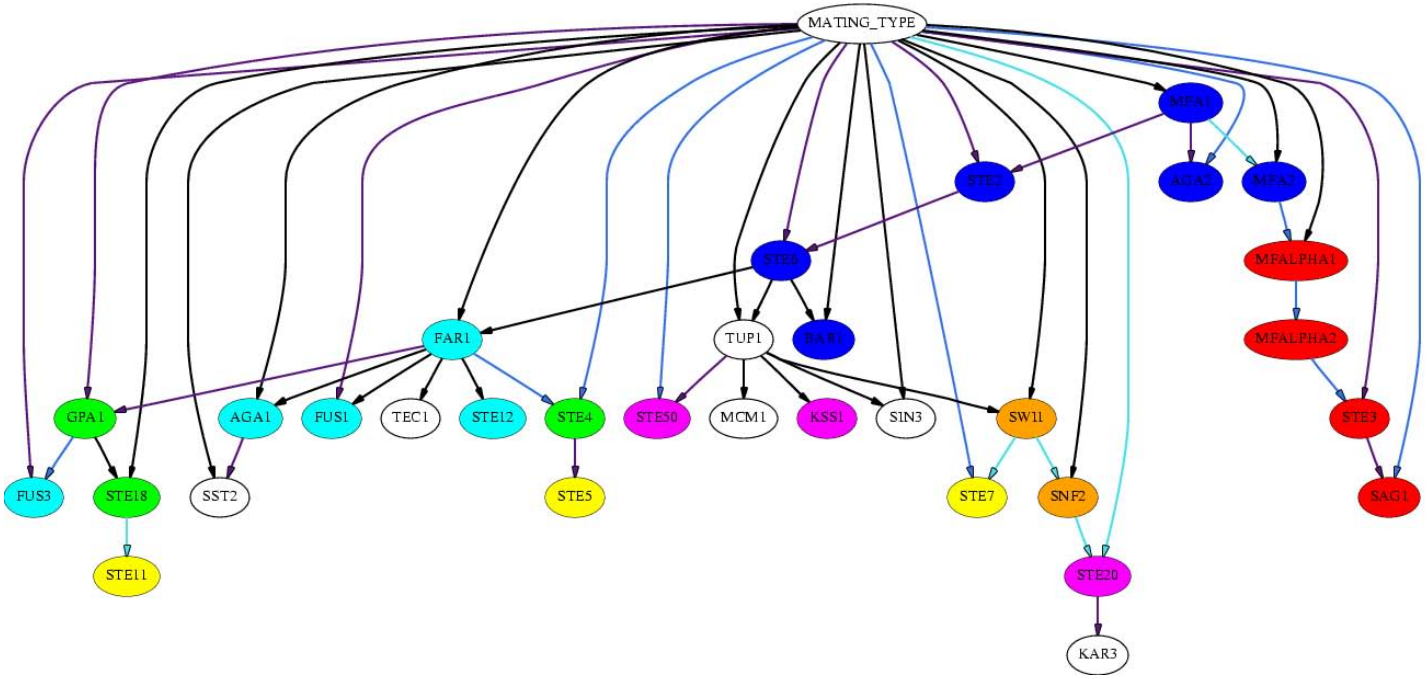
Model averaging

- Integrating over all possible parameters protects us from overfitting parameters
- We can provide some protection against overfitting model structure by averaging over the model posterior distribution

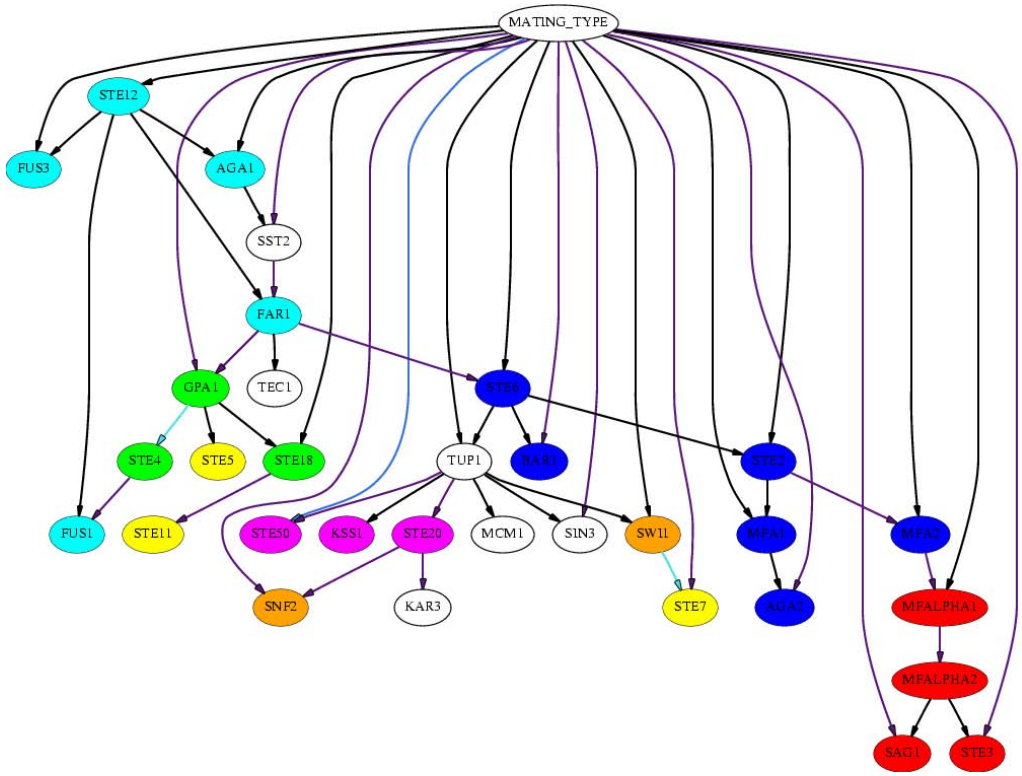
$$
\begin{aligned}
p(E_{XY}|D) &= \sum_S p(E_{XY}, S|D) && (18)\\
&= \sum_S p(E_{XY}|D, S) \cdot p(S|D) && (19)\\
&= \sum_S 1_{XY}(S) \cdot e^{BayesianScore(S)} && (20)
\end{aligned}
$$

Model search: edge consensus of top 50 models

Edge colors: black 10^9, purple $10^6 - 10^9$, dark blue $10^3 - 10^6$, light blue $1 - 10^3$



Model search: edge consensus of top 50 constrained models



How can we model these?

turns on	turns off	raises
lowers	activates	deactivates
represses	derepresses	inhibits
deinhibits	expresses	suppresses
methylates	demethylates	phosphorylates
dephosphorylates	protects	deprotects
reduces	oxidizes	transcribes
translates	regulates	controls
catalyzes	metabolizes	ligates
binds	initiates	enhances
silences	stimulates	induces
promotes	requires	elevates
is necessary for	is a component of	is a substitute for
is a factor in		

Idea - use the parameter prior!

- Recall the likelihood function is:

$$p(D|S) = \int_\theta p(D, \theta|S)\mathrm{d}\theta \quad (21)$$

$$= \int_\theta p(D|\theta, S)p(\theta|S)\mathrm{d}\theta \quad (22)$$

- We can use $p(\theta|S)$ to model relationships
- For example, to represent a positive edge from X to Y, for all values y of Y, and for all values $x_i < x_j$ of X we constrain θ so that:

$$p(Y > y|X = x_i) \leq p(Y > y|X = x_j) \quad (23)$$