Networks

Scoring Bayesian network structures and prior distributions

Outline

- Review of structure learning task
- Structure learning as search through graph space
- Structure learning scoring function
- Conjugate prior distributions
- Efficient computation of the structure scoring function

The Structure Learning Task

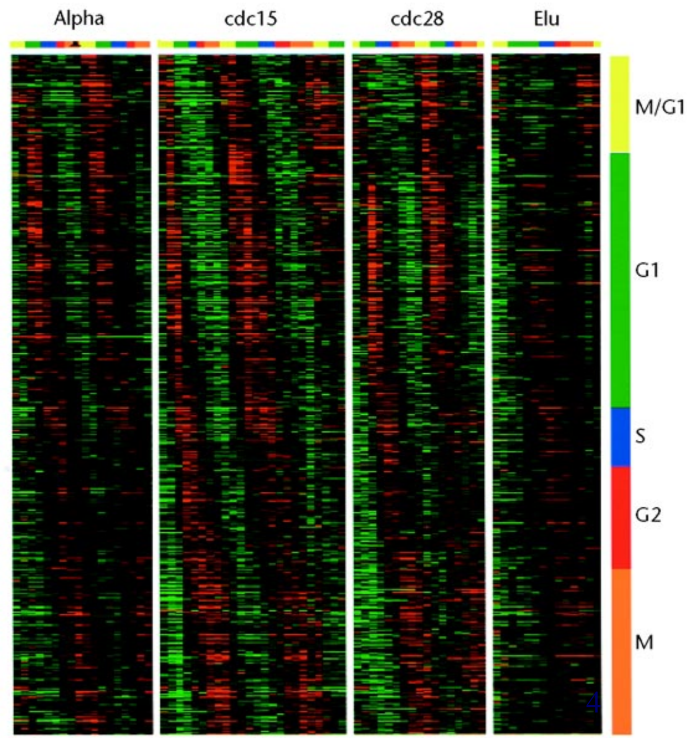
- **Given**: a set of training instances

L	G	I	C	lacI-unbound	CAP-bound	Z
present	present	present	present	true	false	low
present	present	present	present	true	false	absent
absent	present	present	present	false	false	high
			...			

- **Do**: infer the graph structure (and perhaps the parameters of the CPDs too)

Bayes Net Structure Learning Case Study:
Friedman et al., *JCB* 2000

- expression levels in populations of yeast cells
- 800 genes
- 76 experimental conditions



Learning Bayesian Network Structure

- given a function for scoring network structures, we can cast the structure-learning task as a search problem

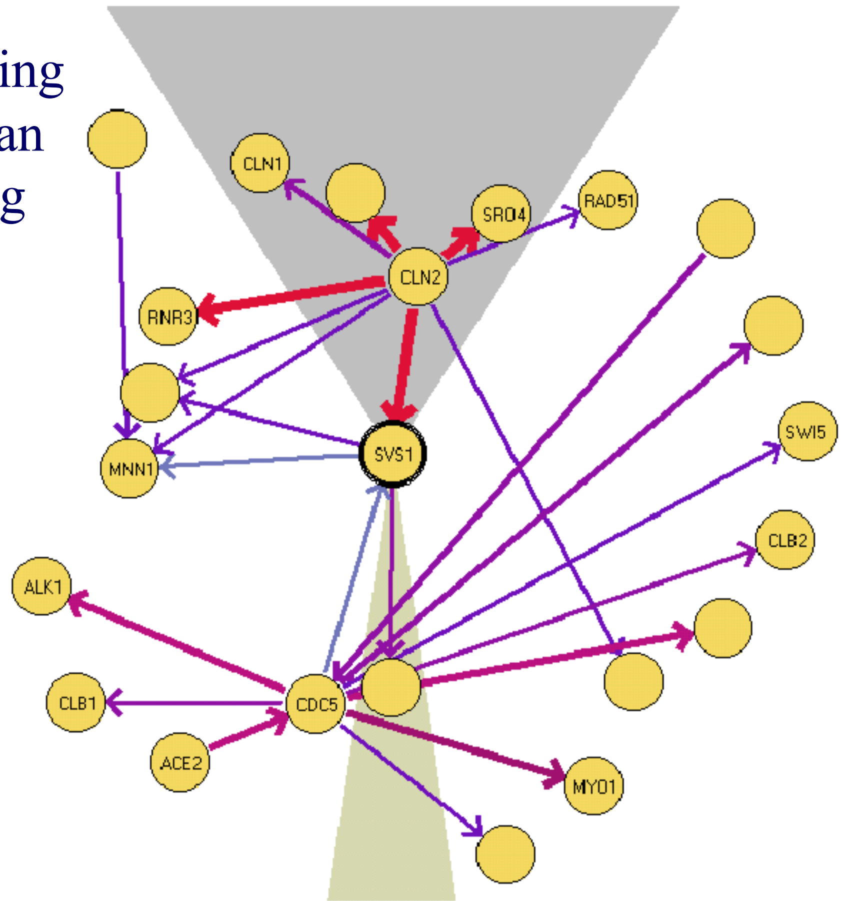
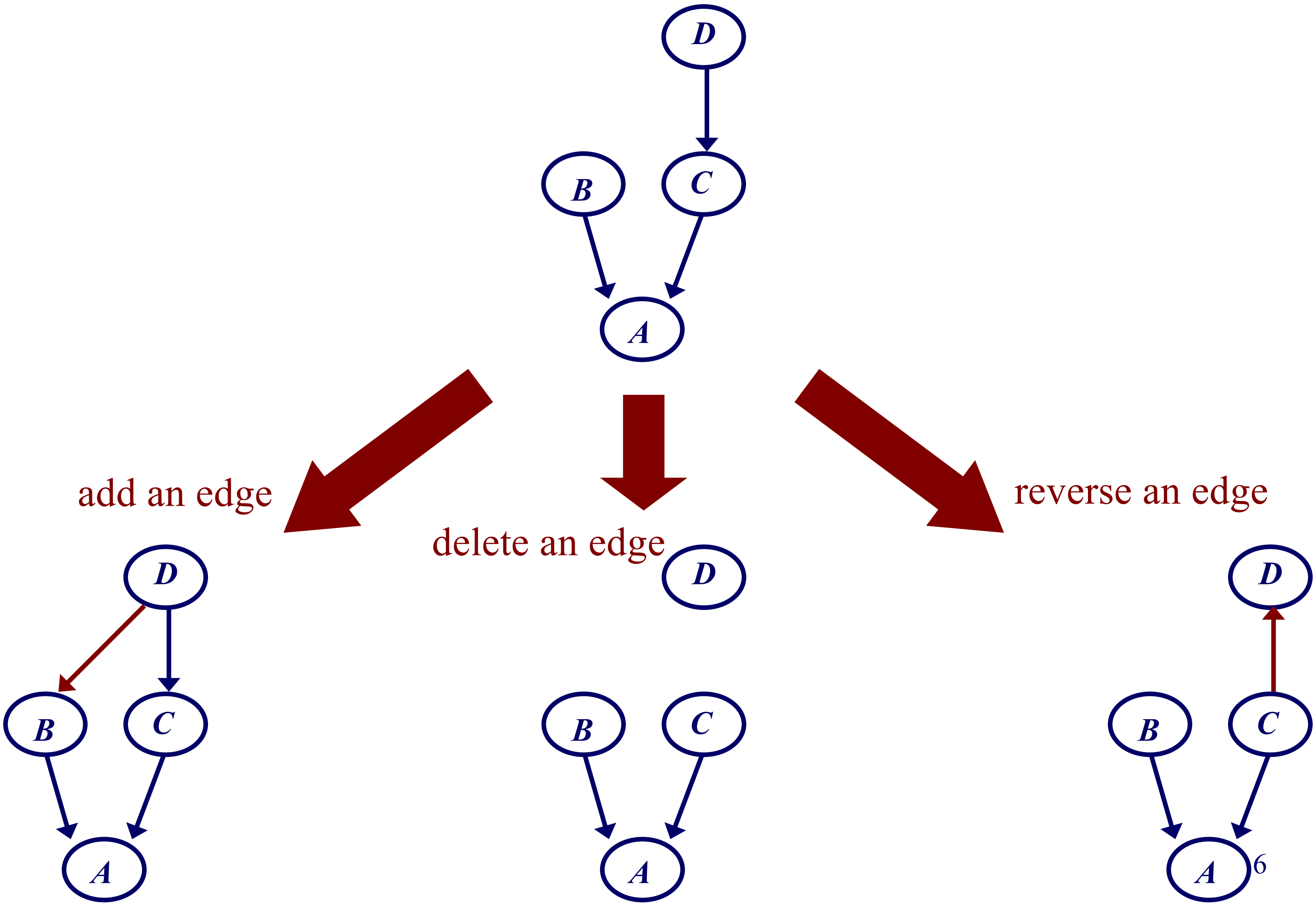


figure from Friedman et al., *Journal of Computational Biology*, 2000

Structure Search Operators



Bayesian Network Structure Learning

- we need a scoring function to evaluate candidate networks; Friedman et al. use one with the form

constant (depends on D)

$$
\begin{aligned}
\text{score}(G:D) &= \log \Pr(G \mid D) \\
&= \log \Pr(D \mid G) + \log \Pr(G) + C
\end{aligned}
$$

log probability of data D given graph G

log prior probability of graph G

- where they take a Bayesian approach to computing $\Pr(D \mid G)$

$$\Pr(D \mid G) = \int \Pr(D \mid G, \Theta) \Pr(\Theta \mid G) d\Theta$$

i.e. don't commit to particular parameters in the Bayes net

The Bayesian Approach to Structure Learning

- Friedman et al. take a Bayesian approach:

$$\Pr(D \mid G) = \int \Pr(D \mid G, \Theta) \Pr(\Theta \mid G) d\Theta$$

- How can we calculate the probability of the data without using specific parameters (i.e. probabilities in the CPDs)?

- Let's consider a simple case of estimating the parameter of a weighted coin…

The Beta Distribution

- suppose we're taking a Bayesian approach to estimating the parameter θ of a weighted coin
- the Beta distribution provides a convenient prior

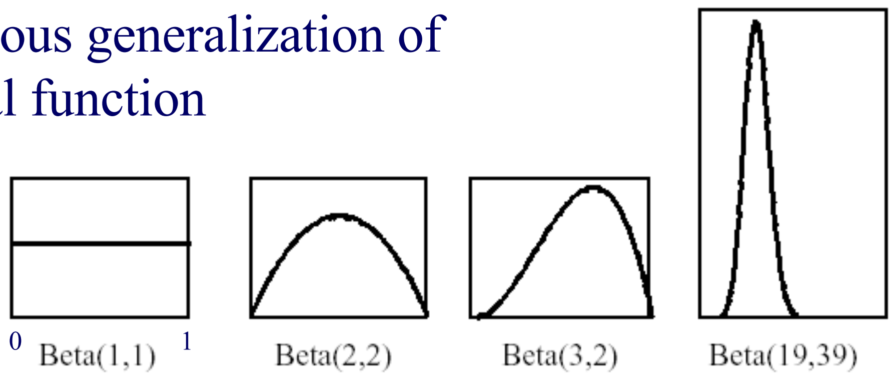
$$P(\theta) = \frac{\Gamma(\alpha_h + \alpha_t)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} \theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1}$$

where

α_h # of "imaginary" heads we have seen already

α_t # of "imaginary" tails we have seen already

Γ continuous generalization of factorial function



The Beta Distribution

- suppose now we're given a data set D in which we observe M_h heads and M_t tails

$$P(\theta \mid D) = \frac{\Gamma(\alpha + M_h + M_t)}{\Gamma(\alpha_h + M_h)\Gamma(\alpha_t + M_t)} \theta^{\alpha_h + M_h - 1}(1-\theta)^{\alpha_t + M_t - 1}$$

$$= \mathrm{Beta}(\alpha_h + M_h, \alpha_t + M_t)$$

- the posterior distribution is also Beta: we say that the set of Beta distributions is a *conjugate* family for binomial sampling

The Beta Distribution

- assume we have a distribution $P(\theta)$ that is Beta(α_h, α_t)
- what's the marginal probability (i.e. over all θ) that our next coin flip would be heads?

$$P(X = heads) = \int_0^1 P(X = heads \mid \theta)\ P(\theta)\ d\theta$$

$$= \int_0^1 \theta\, P(\theta)\ d\theta = \frac{\alpha_h}{\alpha_h + \alpha_t}$$

- what if we ask the same question after we've seen M actual coin flips?

$$P(X_{M+1} = heads \mid x_1, \ldots, x_M) = \frac{\alpha_h + M_h}{\alpha_h + \alpha_t + M}$$

Model evidence with a Beta prior

- For the purposes of scoring a Bayesian network structure, we are interested in computing P(D), which often referred to as the **model evidence**
- For our simple coin flipping example, if D consists of M_h heads and M_t tails, then

$$P(D) = \frac{\Gamma(\alpha_h + \alpha_t)\Gamma(M_h + \alpha_h)\Gamma(M_t + \alpha_t)}{\Gamma(\alpha_h)\Gamma(\alpha_t)\Gamma(M_h + \alpha_h + M_t + \alpha_t)}$$

The Dirichlet Distribution

- for discrete variables with more than two possible values, we can use *Dirichlet* priors

- Dirichlet priors are a *conjugate* family for multinomial data

$$P(\theta) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$$

- if $P(\theta)$ is Dirichlet($\alpha_1, \ldots, \alpha_K$), then $P(\theta|D)$ is Dirichlet($\alpha_1+M_1, \ldots, \alpha_K+M_K$), where M_i is the # occurrences of the i^{th} value

The Bayesian Approach to Scoring BN Network Structures

$$\Pr(D \mid G) = \int \Pr(D \mid G, \Theta) \Pr(\Theta \mid G) d\Theta$$

- we can evaluate this type of expression fairly easily because
  - *parameter independence*: the integral can be decomposed into a product of terms: one per variable
  - Beta/Dirichlet are conjugate families (i.e. if we start with Beta priors, we still have Beta distributions after updating with data)
  - the integrals have closed-form solutions

Scoring Bayesian Network Structures

- when the appropriate priors are used, and all instances in *D* are complete, the scoring function can be decomposed as follows

$$\text{score}(G : D) = \sum_i Score(X_i, Parents(X_i) : D)$$

- thus we can
  - score a network by summing terms (computed as just discussed) over the nodes in the network
  - efficiently score changes in a *local* search procedure

Summary

- Structure learning can be cast a search problem through through graph space
- By being Bayesian, a structure scoring function can be defined that does not depend on specific parameter values for the network
- Conjugate prior distributions allow for closed-form expressions of the structure scoring function
- The scoring function decomposes into a sum over the nodes of the network, allowing for efficient updates