Graphical model structure learning

Kevin Murphy

University of British Columbia

August, 2006

Some figures and slides are from Sam Roweis, Mike Jordan, David MacKay, Nir Friedman, Daphne Koller Dana Pe'er and Karen Sachs.

Outline

- Introduction
- Bayesian model selection: basics
- Bayesian model selection: tabular Bayes nets
- Tree-structured models
- Searching through DAGs
- Searching through variable orderings
- MCMC
- Latent variables
- Undirected models
- Causality
- Application: reconstructing a cell signalling pathway

Structure learning: why?

- We often want to learn the structure of the graphical model:
  - Scientific discovery (data mining)
  - Density estimation, for prediction, compression, classification etc.
- Often we might be uncertain about the right model (especially if the sample size is small)
  - Look for features that they all share
  - Average predictions over models

Structure learning: how?

- Constraint-based approach:
  - Assume some way of testing conditional independencies $X_1 \perp X_2 | X_3$
  - Then construct model consistent with these results
- Search-and-score approach:
  - Define a scoring function for measuring model quality (e.g., marginal likelihood or penalized likelihood)
  - Use a search algorithm to find a (local) maximum of the score
- We will mostly focus on the second method, using Bayesian scoring metrics.

Outline

- Introduction $\surd$
- Bayesian model selection: basics
- Bayesian model selection: tabular Bayes nets
- Tree-structured models
- Searching through DAGs
- Searching through variable orderings
- MCMC
- Latent variables
- Undirected models
- Causality
- Application: reconstructing a cell signalling pathway

Is the coin biased?

- "When spun on edge $N = 250$ times, a Belgian one-euro coin came up heads $Y = 140$ times and tails 110."
- We would like to distinguish two models, or hypotheses: H_0 means the coin is unbiased (so $p = 0.5$); H_1 means the coin is potentially biased (has probability of heads p).
- H_0 is a special case of H_1 (H_0 is "simpler"); these are nested hypotheses, not mutually exclusive. The corresponding events are p is clamped (H_0) or not (H_1).
- We want to compute the posterior ratio of the 2 hypotheses:

$$\frac{P(H_1|D)}{P(H_0|D)} = \frac{P(D|H_1)P(H_1)}{P(D|H_0)P(H_0)}$$

Bayes factors

- We want to compute the posterior ratio of the 2 hypotheses:

$$\frac{P(H_1|D)}{P(H_0|D)} = \frac{P(D|H_1)P(H_1)}{P(D|H_0)P(H_0)}$$

- If we assume a uniform prior $P(H_0) = P(H_1) = 0.5$, then we can just focus on the ratio of the marginal likelihoods:

$$\text{BayesFactor}(1,0) = \frac{P(D|H_1)}{P(D|H_0)}$$

- For H_0, there is no free parameter, so

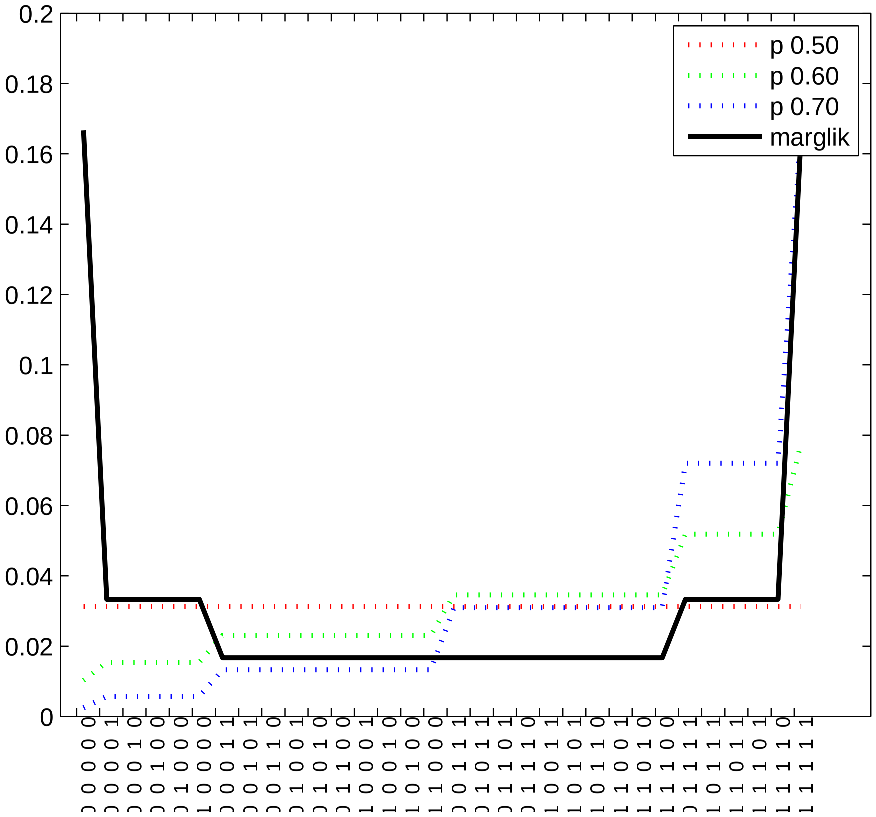
$$P(D|H_0) = 0.5^N$$

- For H_1, the marginal likelihood is given by

$$P(D|H_1) = \int_0^1 d\theta \;\; P(D|\theta, H_1)P(\theta|H_1)$$

Marginal likelihood integrates over all θ

$$p(D|H_1) = \int p(D|\theta)p(\theta)d\theta$$



Prior

- Let us assume a beta prior on the coin bias θ

$$P(\theta|\alpha, H_1) = Be(\theta; \alpha_h, \alpha_t) = \frac{1}{Z(\alpha_h, \alpha_t)} \theta^{\alpha_h - 1} (1-\theta)^{\alpha_t - 1}$$

where

$$Z(\alpha_h, \alpha_t) = \int_0^1 d\theta \quad \theta^{\alpha_h - 1}(1-\theta)^{\alpha_t - 1} = \frac{\Gamma(\alpha_h)\Gamma(\alpha_t)}{\Gamma(\alpha_h + \alpha_t)}$$

- $\Gamma(n) = (n-1)!$ for positive integers.
- Mean $E\theta = \frac{\alpha_h}{\alpha_h + \alpha_t}$.
- If we set $\alpha_h = \alpha_t = 1$, we get a uniform prior (and $Z = 1$).

Posterior

- Suppose we see N_h heads and N_t tails. The parameter posterior is

$$\begin{aligned} P(\theta|D,\alpha) &= \frac{p(\theta|\alpha)P(D|\theta,\alpha)}{P(D|\alpha)} \\ &= \frac{1}{P(D|\alpha)}\left[\frac{1}{Z(\alpha_h,\alpha_t)}\theta^{\alpha_h-1}(1-\theta)^{\alpha_t-1}\right]\theta^{D_h}(1-\theta)^{D_t} \\ &= Be(\theta;\alpha_h+N_h,\alpha_t+N_t) \end{aligned}$$

- Multiplying a beta prior by a binomial likelihood results in a beta posterior, so we say the beta prior is conjugate to the binomial likelihood.

Marginal likelihood (model evidence)

The posterior is

$$\begin{aligned}
P(\theta|D,\alpha) &= Be(\theta;\alpha_h+N_h,\alpha_t+N_t) = Be(\theta;\alpha'_h,\alpha'_t) \\
&= \frac{1}{Z(\alpha'_h,\alpha'_t)}\theta^{\alpha'_h-1}(1-\theta)^{\alpha'_t-1} \\
&= \frac{1}{P(D|\alpha)}\frac{1}{Z(\alpha_h,\alpha_t)}\theta^{\alpha'_h-1}(1-\theta)^{\alpha'_t-1}
\end{aligned}$$

Hence

$$\begin{aligned}
Z(\alpha'_h,\alpha'_t) &= P(D|\alpha)Z(\alpha_h,\alpha_t) \\
P(D|\alpha) &= \frac{Z(\alpha_h+N_h,\alpha_t+N_t)}{Z(\alpha_h,\alpha_t)} \\
&= \frac{\Gamma(\alpha)}{\Gamma(\alpha+N)}\cdot\frac{\Gamma(\alpha_h+N_h)}{\Gamma(\alpha+N)}\cdot\frac{\Gamma(\alpha_t+N_t)}{\Gamma(\alpha+N)}
\end{aligned}$$

An alternative derivation of $p(D)$

- By the chain rule of probability,

$$P(x_{1:N}) = P(x_1)P(x_2|x_1)P(x_3|x_{1:2})\ldots$$

- Also, after N data cases, $P(X|D_{1:N}) = Be(\vec{\alpha}+\vec{N})$, so

$$P(X=k|D_{1:N},\vec{\alpha}) = \frac{N_k+\alpha_k}{\sum_i N_i+\alpha_i} \overset{\text{def}}{=} \frac{N_k+\alpha_k}{N+\alpha}$$

- Suppose $D = H, T, T, H, H, H$. Then

$$\begin{aligned} P(D) &= \frac{\alpha_h}{\alpha}\cdot\frac{\alpha_t}{\alpha+1}\cdot\frac{\alpha_t+1}{\alpha+2}\cdot\frac{\alpha_h+1}{\alpha+3}\cdot\frac{\alpha_h+2}{\alpha+4} \\ &= \frac{[\alpha_h(\alpha_h+1)(\alpha_h+2)]\,[\alpha_t(\alpha_t+1)]}{\alpha(\alpha+1)\cdots(\alpha+4)} \\ &= \frac{[(\alpha_h)\cdots(\alpha_h+N_h-1)]\,[(\alpha_t)\cdots(\alpha_t+N_t-1)]}{(\alpha)\cdots(\alpha+N)} \end{aligned}$$

An alternative derivation of $p(D)$

- For integers,

$$
\begin{aligned}
&(\alpha)(\alpha+1)\cdots(\alpha+M-1)\\
&\quad=\frac{(\alpha+M-1)!}{(\alpha-1)!}\\
&\quad=\frac{(\alpha+M-1)(\alpha+M-2)\cdots(\alpha+M-M)(\alpha+M-M-1)\cdots 2\cdot}{(\alpha-1)(\alpha-2)\cdots 2\cdot 1}\\
&\quad=\frac{(\alpha+M-1)(\alpha+M-2)\cdots(\alpha)(\alpha-1)\cdots 2\cdot 1}{(\alpha-1)(\alpha-2)\cdots 2\cdot 1}
\end{aligned}
$$

- For reals, we replace $(\alpha-1)!$ with $\Gamma(\alpha)$.
- Hence

$$
\begin{aligned}
P(D) &= \frac{[(\alpha_h)\cdots(\alpha_h+N_h-1)]\,[(\alpha_t)\cdots(\alpha_t+N_t-1)]}{(\alpha)\cdots(\alpha+N)}\\
&= \frac{\Gamma(\alpha)}{\Gamma(\alpha+N)}\cdot\frac{\Gamma(\alpha_h+N_h)}{\Gamma(\alpha+N)}\cdot\frac{\Gamma(\alpha_t+N_t)}{\Gamma(\alpha+N)}
\end{aligned}
$$

Bayes factor for coin example

- Bayes factor = ratio of marginal likelihoods:

$$\frac{P(D|H_1)}{P(D|H_0)} = = \frac{Z(\alpha_h + N_h, \alpha_t + N_t)}{Z(\alpha_h, \alpha_t)} \frac{1}{0.5^N}$$

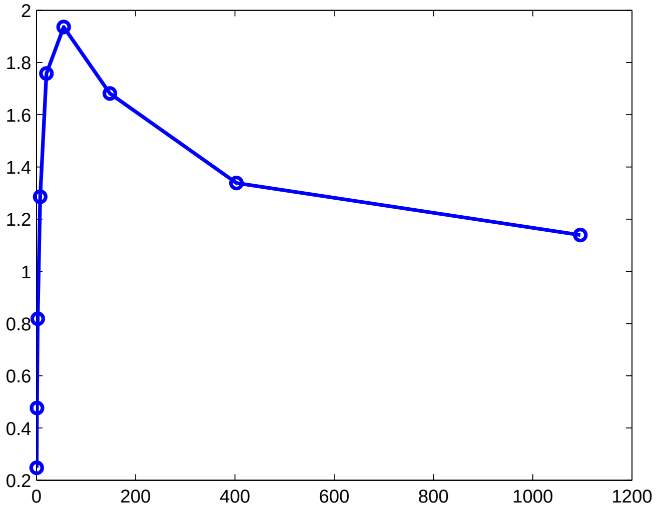
$$= \frac{\Gamma(140+\alpha)\Gamma(110+\alpha)}{\Gamma(250+2\alpha)} \times \frac{\Gamma(2\alpha)}{\Gamma(\alpha)\Gamma(\alpha)} \times 2^{250}$$

- Must work in log domain!

```
alphas = [0.37 1 2.7 7.4 20 55 148 403 1096];
Nh = 140; Nt = 110; N = Nh+Nt;
numer = gammaln(Nh+alphas) + gammaln(Nt+alphas) + gammaln
denom = gammaln(N+2*alphas) + 2*gammaln(alphas);
r = exp(numer ./ denom);
```

Robustness analysis (sensitivity to hyperparameter)

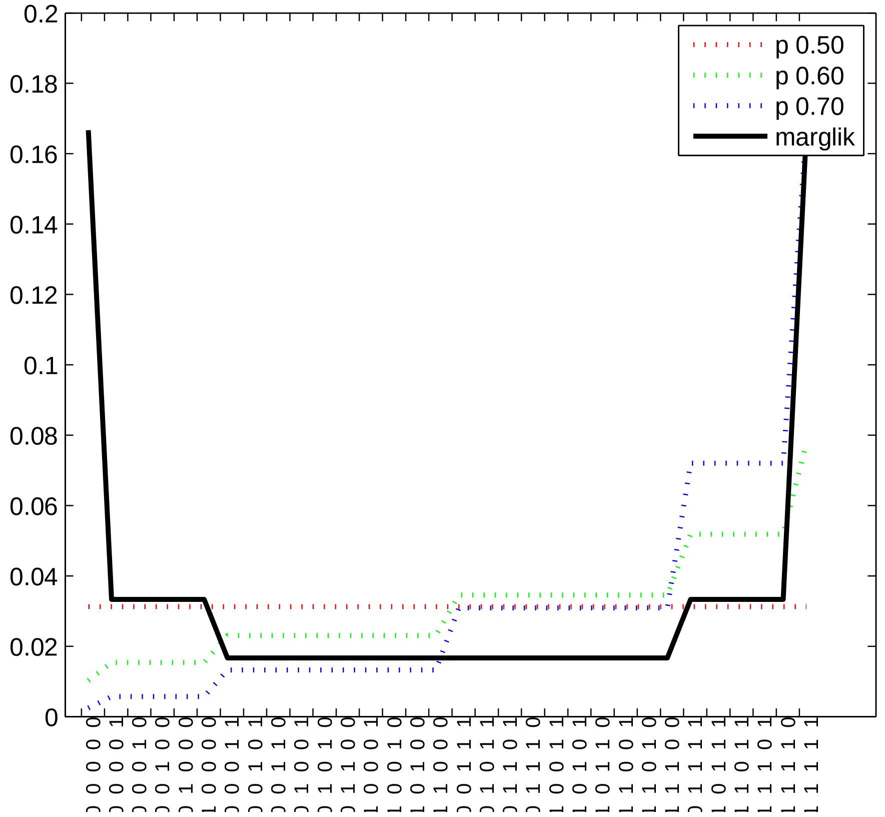
- We plot BayesFactor(1,0) vs hyperparameter α:



- For a uniform beta prior, $\frac{P(D|H_1)}{P(D|H_0)} = 0.48$, (weakly) favoring the fair coin hypothesis H_0!
- At best, for $\alpha = 50$, we can make the biased hypothesis twice as likely.
- Not as dramatic as saying "we reject the null hypothesis (fair coin) with significance 6.6%".

Marginal likelihood integrates over all θ

$$p(D) = \int p(D|\theta)p(\theta)d\theta$$

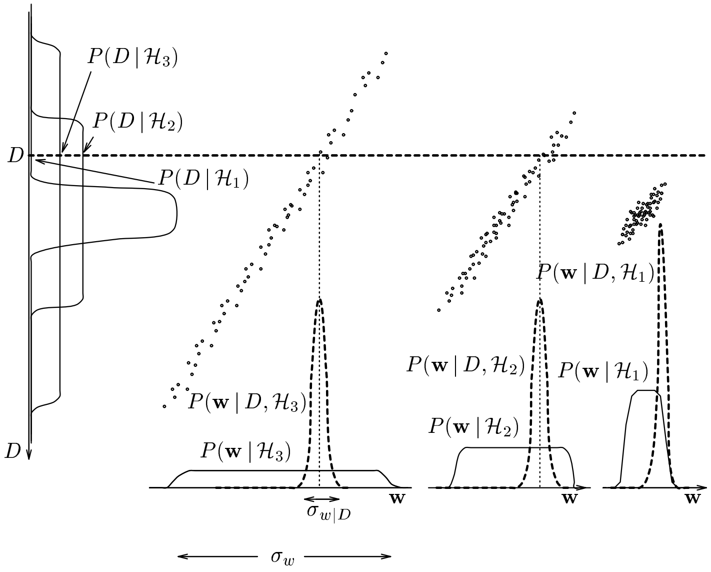


Bayesian Occam's razor

- Why is $P(H_0|D)$ higher when the coin sequences are more uniform?
- It is not because the prior explicitly favors simpler models $p(H_0) > p(H_1)$ (although this is possible).
- It because the evidence $P(D) = \int dw P(D|w)P(w)$, automatically penalizes complex models.
- Occam's razor says "If two models are equally predictive, prefer the simpler one".
- This is an automatic consequence of using Bayesian model selection.
- Maximum likelihood would always pick the most complex model, since it has more parameters, and hence can fit the training data better.
- Good test for a learning algorithm: feed it random noise, see if it "discovers" structure!

Bayesian Occam's razor

- $P(D|H_1)$ is smallest, since it is too simple a model.
- $P(D|H_3)$ is second smallest, since it is too complex, so it spreads its probability mass more thinly over the (D, θ) space (fewer dots on the horizontal line).
- We trust an expert who predicts a few *specific* (and correct!) things more than an expert who predicts many things.


$P(D\,|\,\mathcal{H}_3)$
$P(D\,|\,\mathcal{H}_2)$
D
$P(D\,|\,\mathcal{H}_1)$
$P(\mathbf{w}\,|\,D,\mathcal{H}_1)$
$P(\mathbf{w}\,|\,D,\mathcal{H}_2)$
$P(\mathbf{w}\,|\,\mathcal{H}_1)$
$P(\mathbf{w}\,|\,D,\mathcal{H}_3)$
$P(\mathbf{w}\,|\,\mathcal{H}_2)$
D
$P(\mathbf{w}\,|\,\mathcal{H}_3)$
$\mathbf{w}$
$\mathbf{w}$
$\mathbf{w}$
$\sigma_{w|D}$
σ_w

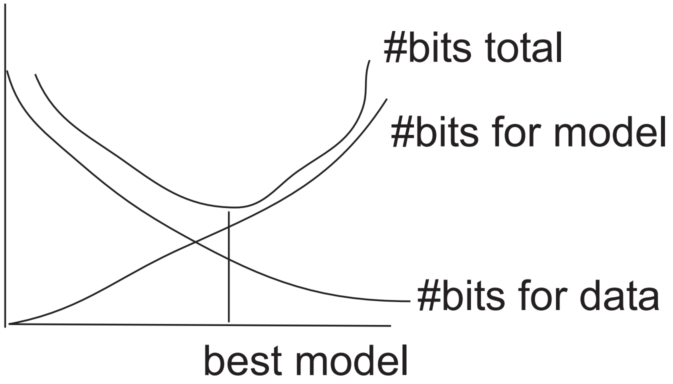
Minimum description length (MDL)

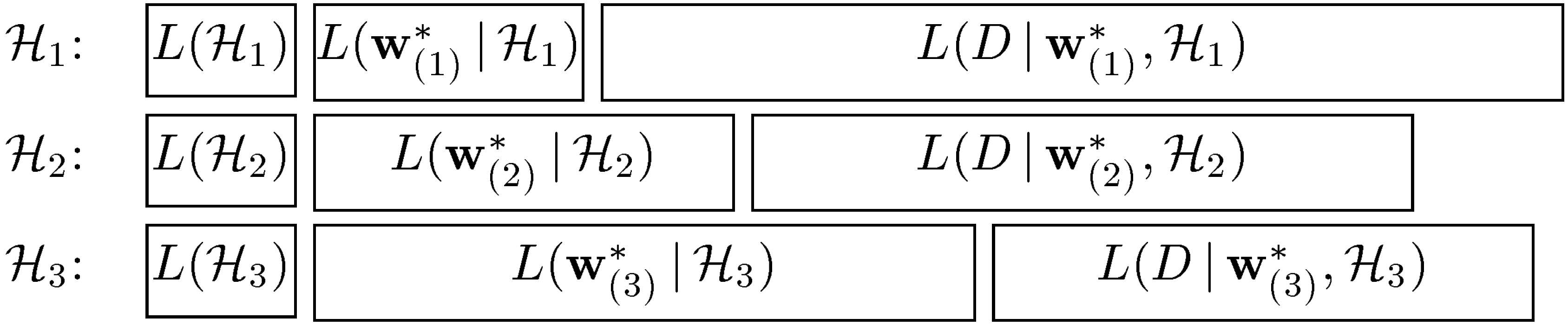
- Another way of thinking about Bayesian Occam's razor is in terms of information theory.
- To losslessly send a message about an event x with probability $P(x)$ takes $L(x) = -\log_2 P(x)$ bits.
- Suppose instead of sending the raw data, you send a model and then the residual errors (the parts of the data not predicted by the model).
- This takes $L(D, H)$ bits:

$$L(D, H) = -\log P(H) - \log P(D|H)$$

- The best model is the one with the overall shortest message.

Minimum description length (MDL)





$\mathcal{H}_1$: $L(\mathcal{H}_1)$ $L(\mathbf{w}^*_{(1)}\,|\,\mathcal{H}_1)$ $L(D\,|\,\mathbf{w}^*_{(1)},\mathcal{H}_1)$

$\mathcal{H}_2$: $L(\mathcal{H}_2)$ $L(\mathbf{w}^*_{(2)}\,|\,\mathcal{H}_2)$ $L(D\,|\,\mathbf{w}^*_{(2)},\mathcal{H}_2)$

$\mathcal{H}_3$: $L(\mathcal{H}_3)$ $L(\mathbf{w}^*_{(3)}\,|\,\mathcal{H}_3)$ $L(D\,|\,\mathbf{w}^*_{(3)},\mathcal{H}_3)$

Laplace approximation to the evidence

- Consider a large sample approximation, where the parameter posterior becomes peaked.
- Take a second order Taylor expansion around $\hat{\theta}_{MP}$:

$$\log P(\theta|D) \approx \log P(\hat{\theta}_{MP}|D) - \frac{1}{2}(\theta - \hat{\theta})^T H(\theta - \hat{\theta})$$

  where

$$H \overset{\text{def}}{=} -\frac{\partial^2 \log P(\theta|D)}{\partial\theta\partial\theta^T}|_{\hat{\theta}_{MP}}$$

  is the Hessian.
- By properties of Gaussian integrals,

$$\begin{aligned} P(D) &\approx \int d\theta \;\; P(D|\hat{\theta})P(\hat{\theta})e^{-\frac{1}{2}(\theta-\hat{\theta})^T H(\theta-\hat{\theta})} \\ &= P(D|\hat{\theta})P(\hat{\theta})(2\pi)^{d/2}|H|^{-\frac{1}{2}} \end{aligned}$$

Penalized likelihood

- Laplace approximation

$$P(D) \approx P(D|\hat{\theta})P(\hat{\theta})(2\pi)^{d/2}|H|^{-\frac{1}{2}}$$

- Taking logs

$$\log P(D) = \log P(D|\hat{\theta}) + \log P(\hat{\theta}) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|H|$$

- BIC (Bayesian Information Criterion): drop terms that are independent of N, and approximate $\log|H| \approx d \log N$. So

$$\log P(D) \approx \log P(D|\hat{\theta}_{ML}) - \frac{d}{2}\log N$$

  where d is the number of free parameters.

- AIC (Akaike Information Criterion):

$$\log P(D) \approx \log P(D|\hat{\theta}_{ML}) - d$$

Outline

- Introduction $\surd$
- Bayesian model selection: basics $\surd$
- Bayesian model selection: tabular Bayes nets
- Tree-structured models
- Searching through DAGs
- Searching through variable orderings
- MCMC
- Latent variables
- Undirected models
- Causality
- Application: reconstructing a cell signalling pathway

Bayesian model selection

- Suppose we observe a coin sequence $D = HHTHT$.
- What model generated it?
- A sequence of 5 independent tosses with fixed parameter θ?

$$p(x_{1:N}|indep) = \int [\prod_{i=1}^{N} p(x_i|\theta)]p(\theta)d\theta$$

- A first order Markov chain with stationary transition matrix θ?

$$p(x_{1:N}|MC) = \int [\prod_{i=2}^{N} p(x_i|x_{i-1}, \theta)]p(\theta)d\theta$$

- An HMM?

$$p(x_{1:N}|HMM) = \int [\sum_{h_{1:N}} \prod_{i=2}^{N} p(h_i|h_{i-1}, \theta)p(x_i|h_i, \theta)]p(\theta)d\theta$$

From coins to dice

- Likelihood: binomial → multinomial

$$P(D|\vec{\theta}) = \prod_i \theta_i^{N_i}$$

- Prior: beta → Dirichlet

$$P(\vec{\theta}|\vec{\alpha}) = \frac{1}{Z(\vec{\alpha})} \prod_i \theta_i^{\alpha_i - 1}$$

where

$$Z(\vec{\alpha}) = \frac{\prod_i \Gamma(\alpha_i)}{\Gamma(\sum_i \alpha_i)}$$

- Posterior: beta → Dirichlet

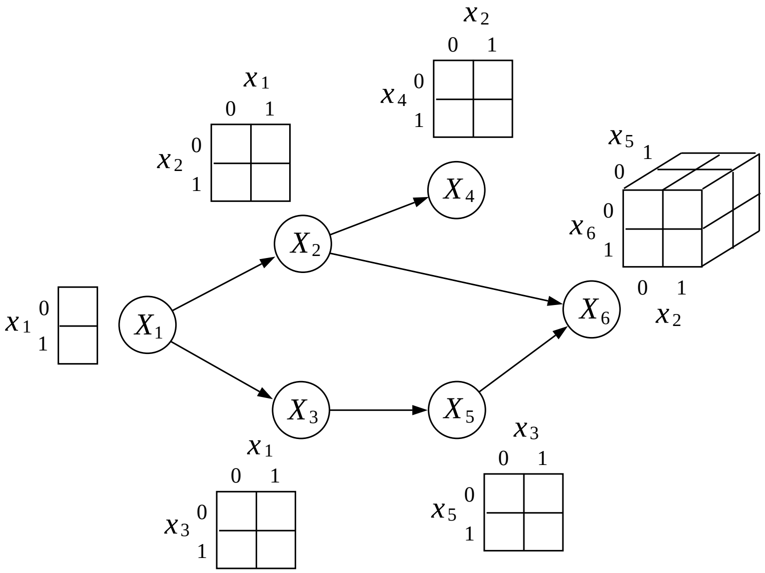
$$P(\vec{\theta}|D) = Dir(\vec{\alpha} + \vec{N})$$

- Evidence (marginal likelihood)

$$P(D|\vec{\alpha}) = \frac{Z(\vec{\alpha} + \vec{N})}{Z(\vec{\alpha})} = \frac{\prod_i \Gamma(\alpha_i + N_i)}{\prod_i \Gamma(\alpha_i)} \frac{\Gamma(\sum_i \alpha_i)}{\Gamma(\sum_i \alpha_i + N_i)}$$

From dice to tabular Bayes nets

- Each CPD $p(X_i = k | X_{\pi_i} = j) = \theta_{ijk}$ is a table, where $\sum_k \theta_{ijk} = 1$.



Assumptions

- Each CPD $p(X_i = k | X_{\pi_i} = j) = \theta_{ijk}$ is a table
- Global parameter independence: $p(\theta) = \prod_i p(\theta_i)$
- Local parameter independence: $p(\theta_i) = \prod_j p(\theta_{ij})$
- Conjugacy: $\theta_{ij\cdot} \sim Dir(\vec{\alpha})$
- Parameter modularity if $\pi_i^G = \pi_i^{G'}$ then

$$p(\theta_i | G) = p(\theta_i | G')$$

- Complete data (all nodes observed), iid data

$$p(D|\theta) = \prod_{n=1}^{N_D} p(X^n|\theta)$$

Marginal likelihood for a tabular Bayes nets

$$
\begin{aligned}
P(D|G) &= \prod_{i=1}^{N_G} \left[\int \prod_{n=1}^{N_G} p(x_i^n | x_{\pi_i}^n, \theta_i) p(\theta_i) \right] \\
&= \prod_{i=1}^{N_G} FamScore(x_i, x_{\pi_i}, D) \\
FamScore(x_i, x_{\pi_i}, D) &= \prod_{j \in Val(\pi_i)} \frac{Z(\vec{\alpha}_{i,j,.} + N_{i,j,.})}{Z(\vec{\alpha}_{i,j,.})} \\
&= \prod_{j \in Val(\pi_i)} \left[\prod_k \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right] \left[\frac{\Gamma(\sum_k \alpha_{ijk})}{\Gamma(\sum_k \alpha_{ijk} + N} \right.
\end{aligned}
$$

Model selection for a 2 node BN

- Suppose we generate data from $X \rightarrow Y$, where $P(X = 0) = P(X = 1) = 0.5$ and $P(Y = 1|X = 0) = 0.5 - \epsilon$, $P(Y = 1|X = 1) = 0.5 + \epsilon$.
- As we increase ϵ, we increase the dependence of Y on X.
- Let us consider 3 hypotheses: $H_0 = X \ \ Y$, $H_1 = X \rightarrow Y$, $H_2 = Y \leftarrow X$, and use uniform priors.
- We will plot model posteriors vs N for different ϵ and different random trials:

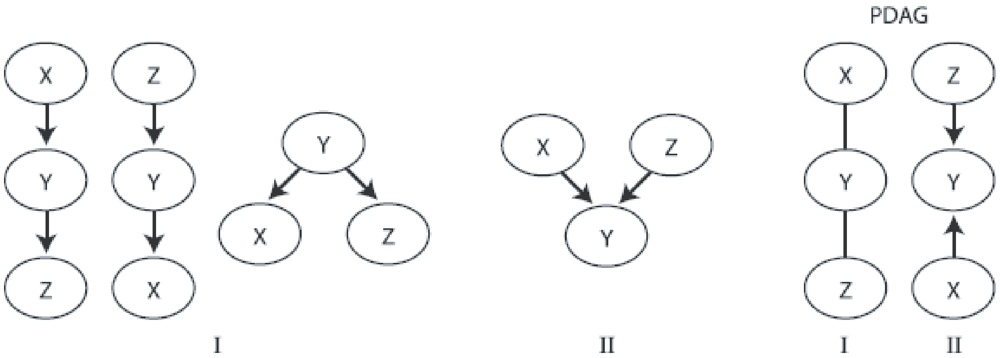
$$P(H_i|D_{1:N}) = \frac{P(D_{1:N}|H_i)P(H_i)}{\sum_j P(D_{1:N}|H_j)P(H_j)}$$

Markov equivalence

- $X \to Y$ and $X \leftarrow Y$ represent the same set of conditional independence statements (namely, none) and hence are called Markov equivalent.
- Hence we want $P(G_1|D) = P(G_2|D)$ (score equivalence), unless we interpret the models causally.
- Structure is only identifiable up to Markov equivalence.

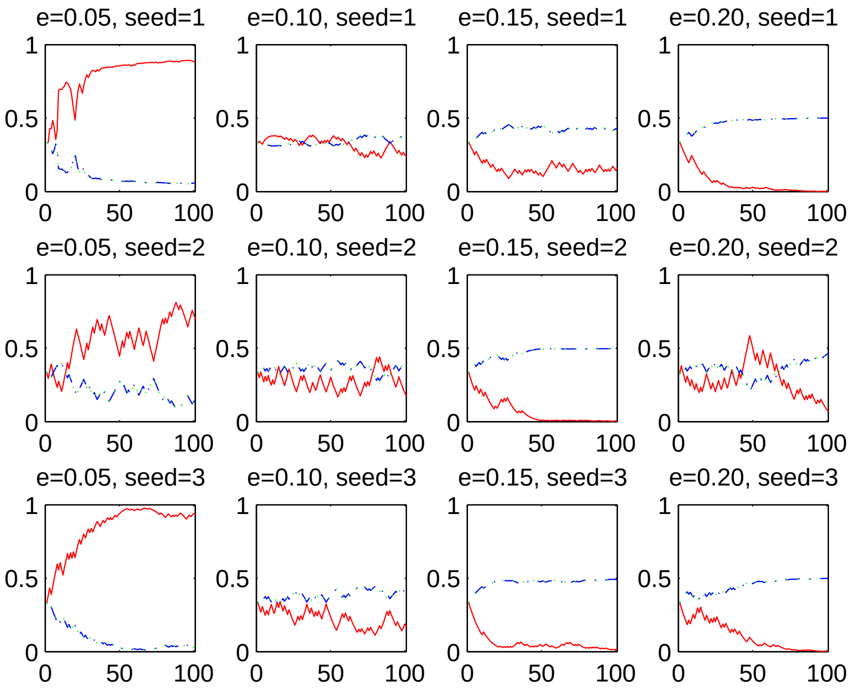
PDAGs (essential graphs)

- We can represent an equivalence class using a PDAG (partially directed acyclic graph), aka essential graph, in which compelled edges are directed, and reversible edges are undirected.
- eg PDAG $X - Y - Z$ represents $\{X \rightarrow Y \rightarrow Z, X \leftarrow Y \leftarrow Z, X \leftarrow Y \rightarrow Z\}$ which encodes $X \not\perp Z$ and $X \perp Z|Y$. The v-structure $X \rightarrow Y \leftarrow Z$ encodes $X \perp Z$ and $X \not\perp Z|Y$.
- Two structures are equivalent if they have the same undirected skeleton and the same set of v-structures.



Example of model selection

red $= H_0$ (independence), blue/green $= H_1/H_2$ (dependence).
See `BNT/examples/static/StructLearn/model-select1.m`.



As we increase the dependence ϵ, the more complex models win out.

Score equivalence

- $X \rightarrow Y$ and $X \leftarrow Y$ are I-equivalent (have the same likelihood).
- Suppose we use a uniform Dirichlet prior for each node in each graph, with equivalent sample size α (K2-prior):

$$P(\theta_X | H_1) = Dir(\alpha, \alpha), \;\; P(\theta_{X|Y=i} | H_2) = Dir(\alpha, \alpha)$$

- In H_1, the equivalent sample size for X is 2α, but in H_2 it is 4α (since two conditioning contexts). Hence the posterior probabilities are different.
- The BDe (Bayesian Dirichlet likelihood equivalent) prior is to use weights $\alpha_{X_i|X_{\pi_i}} = \alpha P'(X_i, X_{\pi_i})$ where P' could be represented by e.g., a Bayes net.
- The BDeu (uniform) prior is $P'(X_i, X_{\pi_i}) = \frac{1}{|X_i||X_{\pi_i}|}$.
- Using the BDeu prior, the curves for $X \rightarrow Y$ and $X \leftarrow Y$ are indistinguishable. Using the K2 prior, they are not.

BIC approximation to the evidence

- Recall BIC (Bayesian Information Criterion)

$$\log P(D) \approx \log P(D|\hat{\theta}_{ML}) - \frac{d}{2}\log N$$

where d is the number of free parameters.

- Note this is independent of the parameter prior $p(\theta)$.
- Let us derive this expression for a Bayes net.
- This is useful when we cannot compute $\int p(D|\theta)p(\theta)d\theta$ exactly.

Log-likelihood in information theoretic terms

$$
\begin{aligned}
\frac{1}{N}\ell &= \frac{1}{N}\sum_i \sum_j \sum_k N_{ijk} \log \theta_{ijk} \\
&= \sum_i \sum_j \sum_k \hat{P}(X_i = j, X_{\pi_i} = k) \log P(X_i = j | X_{\pi_i} = k) \\
&= \sum_{ijk} \hat{P}(X_i = j, X_{\pi_i} = k) \log \frac{P(X_i = j, X_{\pi_i} = k) P(X_i = j)}{P(X_{\pi_i} = k) P(X_i = j)} \\
&= \sum_i \sum_{jk} \hat{P}(X_i = j, X_{\pi_i} = k) \log \frac{P(X_i = j, X_{\pi_i} = k)}{P(X_{\pi_i} = k) P(X_i = j)} \\
&\quad + \sum_{ij} (\sum_k \hat{P}(X_i = j, X_{\pi_i} = k)) \log P(X_i = j) \\
&= \sum_i I(X_i, X_{\pi_i}) - H(X_i)
\end{aligned}
$$

BIC in information theoretic terms

$$\begin{aligned}\text{score}_{BIC}(G|D) &= \ell(\hat{\theta}) - \frac{d(G)}{2}\log N(D) \\ &= N\sum_i I(X_i, X_{\pi_i}) - N\sum_i H(X_i) - \frac{d}{2}\log N\end{aligned}$$

- The mutual information term grows linearly in N, the complexity penalty is logarithmic in N.
- So for large datasets, we pay more attention to fitting the data better.
- Also, the structural prior is independent of N, so does not matter very much.

Desirable properties of a scoring function

- Consistency: i.e., if the data is generated by G^*, then G^* and all I-equivalent models maximize the score.
- Decomposability:

$$\text{score}(G|D) = \sum_i \text{FamScore}(D(X_i, X_{\pi_i}))$$

which makes it cheap to compare score of G and G' if they only differ in a small number of families, e.g.

$$M_1 = (X \to Y \to Z), \quad M_2 = (X \leftarrow Y \to Z)$$

$$S(M_1)/S(M_2) = \frac{S(X)S(Y|X)S(Z|Y)}{S(Y)S(X|Y)S(Z|Y)}$$

- Bayesian score (evidence), likelihood and penalized likelihood (BIC) are all decomposable and consistent.

Maximizing the score

- Consider the family of DAGs G_d with maximum fan-in (number of parents) equal to d.
- Theorem: It is NP-hard to find

  $$G^* = \arg\max_{G \in G_d} \mathsf{score}(G, D)$$

  for any $d \geq 2$.
- The set of possible DAGs is super exponential in d. (The set of Markov equivalence classes is only about 4 times smaller.)

Approaches to structure search

- $d = 1$ (trees) - Chow-Liu algorithm takes $O(N_D N_G^2)$ time to find optimal structure.
- Heuristic search through DAG space.
- Heuristic search through variable orderings.
- MCMC

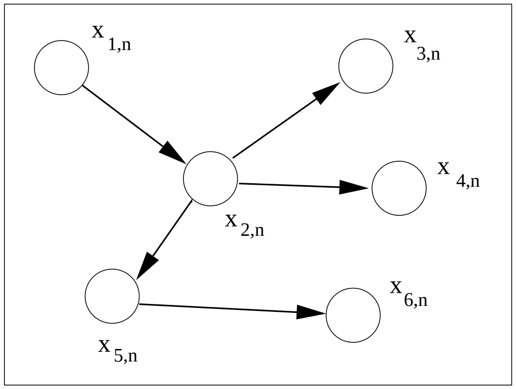
Outline

- Introduction $\surd$
- Bayesian model selection: basics $\surd$
- Bayesian model selection: tabular Bayes nets $\surd$
- Tree-structured models
- Searching through DAGs
- Searching through variable orderings
- MCMC
- Latent variables
- Undirected models
- Causality
- Application: reconstructing a cell signalling pathway

Simple directed trees

- In a simple directed tree, each node has at most one parent. Hence there is no "explaining away", so it does not matter whether we use directed or undirected graphs. It also doesn't matter which node we pick as root.

$$p(x) = p(x_r) \prod_{i \neq r} p(x_i | x_{\pi_i})$$

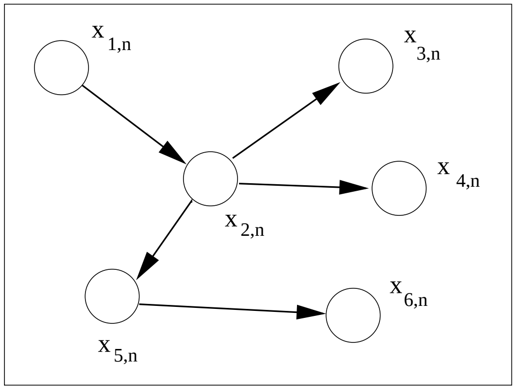


Undirected trees

- For undirected trees, the cliques are all pairs of connected nodes.

$$p(x) = \frac{1}{Z}\psi_i(x_i, x_{\pi_i})$$

where we can make $Z = 1$ with the choice $\psi_i = p(x_i|x_{\pi_i})$ except for one clique involving the root r and child j: $\psi_j = p(x_r)p(x_j|x_r)$.



Optimal Structure

- Let us write the likelihood function:

$$\ell(\theta; D) = \sum_{\vec{x}} N(\vec{x}) \log p(\vec{x})$$

$$= \sum_{\vec{x}} N(\vec{x}) \left(\log p(x_r) + \sum_{i \neq r} \log p(x_i | x_{\pi_i}) \right)$$

- Let $q(x) = N(x)/N$ be the observed counts. For the ML parameters

$$p(x_i | x_{\pi_i}, \hat{\theta}_i) = q(x_i, x_{\pi_i})/q(x_{\pi_i})$$

so

$$\frac{\ell^*}{N} = \sum_{\vec{x}} q(\vec{x}) \left(\log q(x_r) + \sum_{i \neq r} \log \frac{q(x_i, x_{\pi_i})}{q(x_{\pi_i})} \right)$$

$$= \sum_{\vec{x}} q(\vec{x}) \log q(x_r) + \sum_{\vec{x}} q(\vec{x}) \sum_{i \neq r} \log \frac{q(x_i, x_{\pi_i})}{q(x_i) q(x_{\pi_i})}$$

Edge Weights

- Each term in sum $i \neq r$ corresponds to an edge from i to its parent.

$$\begin{aligned}\frac{\ell^*}{N} &= \sum_{\vec{x}} q(\vec{x}) \sum_{i \neq r} \log \frac{q(x_i, x_{\pi_i})}{q(x_i) q(x_{\pi_i})} + C \\ &= \sum_{i \neq r} \sum_{x_i, x_{\pi_i}} q(x_i, x_{\pi_i}) \log \frac{q(x_i, x_{\pi_i})}{q(x_i) q(x_{\pi_i})} + C \\ &= \sum_{i \neq r} W(i; \pi_i) + C\end{aligned}$$

  where the edge weights W are defined by *mutual information*:

$$W(i;j) = \sum_{x_i, x_j} q(x_i, x_j) \log \frac{q(x_i, x_j)}{q(x_i) q(x_j)}$$

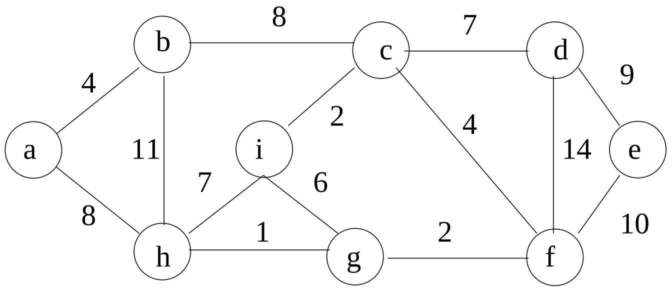
  and the constant $C = \sum_{\vec{x}} q(\vec{x}) \log q(x_r)$ is independent of the graph structure.

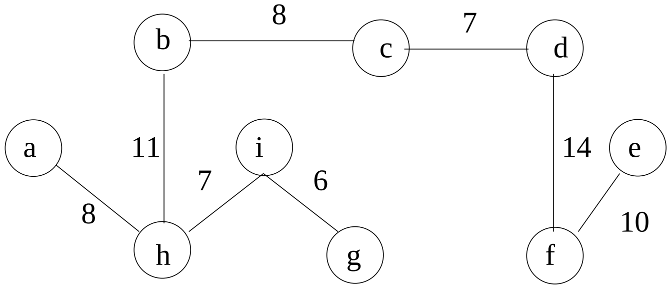
- So overall likelihood is sum of weights on edges that we use. We need the maximum weight spanning tree.

Kruskal's algorithm

- To find the maximum weight spanning tree A on a graph with nodes U and weighted edges E:

  1. $A \leftarrow \mathrm{empty}$
  2. Sort edges E by nonincreasing weight: $e_1, e_2, \ldots, e_K$.
  3. for $k = 1$ to K $\{A \mathrel{+}= e_k$ unless doing so creates a cycle$\}$





Global MLE tree for discrete data

We can now completely solve the tree learning problem:

1. Compute the marginal counts $q(x_i)$ for each node and pairwise counts $q(x_i, x_j)$ for all pairs of nodes.
2. Set the weights to the mutual informations:

$$W(i;j) = \sum_{x_i, x_j} q(x_i, x_j) \log \frac{q(x_i, x_j)}{q(x_i) q(x_j)}$$

3. Find the maximum weight spanning tree A=MWST(W).
4. Using the undirected tree A chosen by MWST, pick a root arbitrarily and orient the edges away from the root.
   Set the conditional functions to the observed frequencies:

$$p(x_i | x_{\pi_i}) = \frac{q(x_i, x_{\pi_i})}{\sum_{x_i} q(x_i, x_{\pi_i})} = \frac{q(x_i, x_{\pi_i})}{q(x_{\pi_i})}$$

Global MLE tree for Gaussian data

- One can use exactly the same algorithm for jointly Gaussian random variables (the CPDs are fit by linear regression).
- The mutual information between X and Y, where $P(X, Y)$ is Gaussian with covariance

  $$\Sigma = \begin{pmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{pmatrix}$$

  is given by

  $$I(X;Y) = -\frac{1}{2} \log \frac{\det \Sigma}{\det \Sigma_{XX} \det \Sigma_{YY}}$$

- For scalars, this becomes

  $$I(X;Y) = -\frac{1}{2} \log(1 - r^2(X, Y))$$

  where the correlation coefficient is

  $$r(X, Y) \overset{\text{def}}{=} \frac{\mathsf{cov}(X, Y)}{\sqrt{\mathsf{var}(X)\mathsf{var}(Y)}}$$

Mixtures of trees

- One can use EM to learn a mixture of tree structured graphical models.
- We use MWST in the M step, applied to the expected mutual information.

Outline

- Introduction $\surd$
- Bayesian model selection: basics $\surd$
- Bayesian model selection: tabular Bayes nets $\surd$
- Tree-structured models $\surd$
- Searching through DAGs
- Searching through variable orderings
- MCMC
- Latent variables
- Undirected models
- Causality
- Application: reconstructing a cell signalling pathway

Searching in DAG space

- Can use greedy hill climbing, tabu search, beam search, simulated annealing, genetic algorithms, etc.
- Typical search operators:
  - Add an edge
  - Delete an edge
  - Reverse an edge
- We can get from any graph to any other graph in at most $O(n^2)$ moves (the diameter of the search space).
- Moves are reversable.
- We can only apply a search operator o to the current graph G if the resulting graph $o(G)$ satisfies the constraints, e.g., acyclicity, indegree bound, induced treewidth bound ("thin junction trees"), hard prior knowledge.

Cost of evaluating moves

- There are $O(n^2)$ operators we could apply at each step.
- For each operator, we need to check if $o(G)$ is acylic.
- We can check acyclicity in $O(e)$ time, where $e = O(n \times deg)$ is the number of edges.
- For local moves, we can check acyclicity in amortized $O(1)$ time using the ancestor matrix.
- If $o(G)$ is acyclic, we need to evaluate its quality. This requires computing sufficient statistics for every family, which takes $O(Mn)$ time, for M training cases.
- Suppose there are K steps to convergence. (We expect $K \ll n^2$, since the diameter is n^2.)
- Hence total time is $O(K \cdot n^2 \cdot Mn)$.

Exploiting decomposable score

- If the operator is valid, we need to evaluate its quality. Define

$$\delta_G(o) = \textsf{score}(o(G)|D) - \textsf{score}(G|D)$$

- If the score is decomposable, and we want to modify an edge involving X and Y, we only need to look at the sufficient statistics for X and Y's families.
- e.g., if $o =$ add $X \rightarrow Y$:

$$\delta_G(o) = \textsf{FamScore}(Y, Pa(Y,G) \cup X|D) - \textsf{FamScore}(Y, Pa(Y,G)|D)$$

- So we can evaluate quality in $O(M)$ time by extracting sufficient statistics for the columns related to X, Y and their parents.
- This reduces the time from $O(Kn^3M)$ to $O(Kn^2M)$.

Exploiting decomposable score

- After eg adding $X \to Y$, we only need to update $\delta(o)$ for the $O(n)$ operators that involve X or Y.
- Also, we can update a heap in $O(n \log n)$ time and thereby find the best o in $O(1)$ time at each step.
- So total cost goes from $O(Kn^2M)$ to $O(K(nM + n \log n))$.
- kd-trees can help for large M.

Local maxima

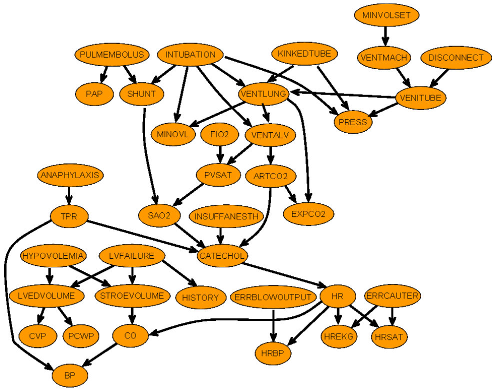
- Greedy hill climbing will stop when it reaches a local maximum or a plateau (a set of neighboring networks that have the same score).
- Unfortunately, plateaux are common, since equivalence classes form contiguous regions of search space, and such classes can be exponentially large.
- Solutions:
  - Random restarts
  - TABU search (prevent the algorithm from undoing an operator applied in the last L steps, thereby forcing it to explore new terrain).
  - Data perturbation (dynamic local search): reweight the data and take step.
  - Simulated annealing: if $\delta(o) > 0$, take move, else accept with probability $e^{\frac{\delta(o)}{t}}$, where t is the temperature. Slow!

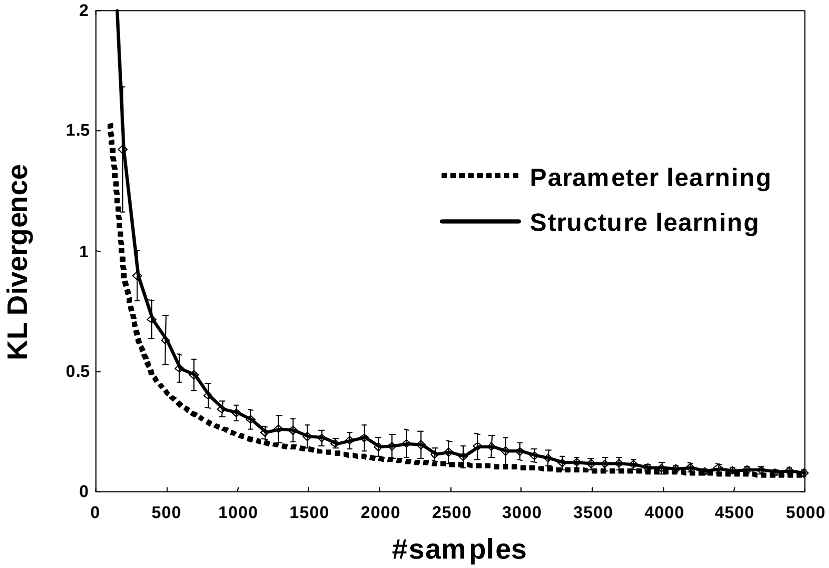
Searching in space of equivalence classes

- The space of class PDAGs is smaller.
- We avoid many of the plateux of I-equivalent DAGs.
- Operators are more complicated to implement and evaluate, but can still be done locally (see paper by Max Chickering).
- Cannot exploit causal/ interventional data (which can distinguish members of an equivalence class).
- Currently less common than searching in DAG space.

Learning the ICU-Alarm network with TABU search

- Learned structures often simpler than “true” model (fewer edges), but predict just as well.
- Can only recover structure up to Markov equivalence.
- 10 minutes to learn structure for 100 variables and 5000 cases.
- (From Friedman and Koller’s book)





Outline

- Introduction $\surd$
- Bayesian model selection: basics $\surd$
- Bayesian model selection: tabular Bayes nets $\surd$
- Tree-structured models $\surd$
- Searching through DAGs $\surd$
- Searching through variable orderings
- MCMC
- Latent variables
- Undirected models
- Causality
- Application: reconstructing a cell signalling pathway

Known order (K2 algorithm)

- Suppose we know a total ordering of the nodes $X_1 \prec X_2 \ldots \prec X_n$ and want to find the best DAG consistent with this.
- The choice of parents for X_i, from $Pa_i \subseteq \{X_1, \ldots, X_{i-1}\}$, is independent of the choice for X_j: since we obey the ordering, we cannot create a cycle.
- Hence we can pick the best set of parents for each node independently.
- For X_i, we need to search all $\binom{i-1}{d}$ subsets of size up to d for the set which maximizes FamScore.
- We can use greedy techniques for this,e.g. using a decision tree or using LASSO for GLMs.

What if order isn't known?

- Search in the space of orderings, then conditioned on $\prec$, pick best graph.
- One possible move is to flip 2 variables in the order, leaving the rest unchanged:

$$(X_{i_1}, \ldots, \mathbf{X_{i_j}}, \ldots, \mathbf{X_{i_k}}, \ldots, X_{i_n}) \rightarrow (X_{i_1}, \ldots, \mathbf{X_{i_k}}, \ldots, \mathbf{X_{i_j}}, \ldots, X_{i_n})$$

- Using score decomposability, only family scores for nodes inside the bold range need to be recomputed.
- The space of orderings is "only" $n!$, and each move is more global.
- This is currently considered the state of the art method for learning Bayes net structure (M. Teyssier and D. Koller, UAI 2005).

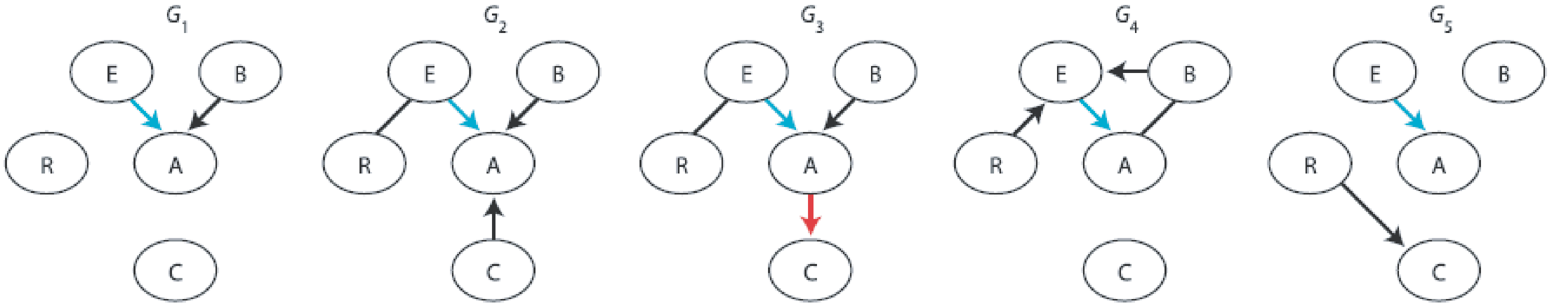
Outline

- Introduction $\surd$
- Bayesian model selection: basics $\surd$
- Bayesian model selection: tabular Bayes nets $\surd$
- Tree-structured models $\surd$
- Searching through DAGs $\surd$
- Searching through variable orderings $\surd$
- MCMC
- Latent variables
- Undirected models
- Causality
- Application: reconstructing a cell signalling pathway

Computing the posterior over models

- So far, we have just tried to find the mode of $P(G|D)$, i.e., the best scoring network.
- But there might be many other models that are almost as good.
- Suppose instead of recovering the graph, we try to determine features, such as: is there an edge or path $X \rightarrow Y$?
- We can compute the probability of features like this using

$$P(f|D) = \sum_G f(G) P(G|D)$$



$P(G_1|D) = 30.41$ $P(G_2|D) = 30.43$ $P(G_3|D) = 30.44$ $P(G_4|D) = 30.42$ $P(G_5|D) = 30.40$

Monte Carlo methods

- The problem is we cannot sum over all graphs

$$P(f|D) = \sum_G f(G)P(G|D)$$

- If we can uniformly sample graphs from $P(G|D)$, we can approximate this using

$$P(f|D) \approx \frac{1}{T}\sum_t f(G_t)$$

  where G_k is the k'th sample.

- Markov chain Monte Carlo (MCMC) provides a way of sampling from distributions such as $P(G|D) = P(G,D)/P(D)$ without having to compute the normalizing constant $p(D) = \sum_G p(G,D)$.

MCMC

- We define a Markov chain on graph structures (in this case) with transition probability given by the Metropolis-Hastings rule
$$P(G'|G) = \min\left(1, \frac{P(G'|D)Q(G'|G)}{P(G|D)Q(G|Q')}\right)$$
where $Q(G'|G)$ is the *proposal probability* and the ratio is the *acceptance probability*.
- The proposal Q has to be such that the Markov chain is ergodic, i.e., we can get to any state from any other state.
- We start the chain off in some inital state and then perform a random walk according to the above dynamics.
- Theory shows the stationary distribution of such a Markov chain is $P(G|D)$.

MCMC convergence

- The *mixing time* is how long it takes the chain to converge from a random starting point.
- Once the chain has converged (after the *burnin*), we can draw (correlated) samples from $P(G|D)$.
- We can diagnose convergence by running the chain from multiple starting points and comparing the results. (Diagnosing convergence is an open problem.)

MCMC for DAG structure

- Suppose the proposal Q picks randomly from the following operators (where legal): add an edge, delete an edge, reverse an edge.
- The MH acceptance probability requires computing the Bayes factor $P(G'|D)/P(G|D)$, which is efficient for decomposable scores.
- However, small changes to the graph can result in large changes to the score, resulting in a jagged landscape.
- So the chain does not mix rapidly (it gets stuck in local optima).

Rao-Blackwellised MCMC

- An alternative idea is to do MCMC sampling in the space of node orderings $\prec$, which "only" has size $n!$.
- Given an ordering, we can sum over all graphs efficiently (see below). Hence
$$P(f|D) \approx \frac{1}{T} \sum_t P(f|D, \prec_t)$$
- This combination of sampling and exact integration/ marginalization is called Rao-Blackwellised sampling.
- This is named after the Rao-Blackwell theorem, which says (roughly) that variance is reduced if you use stratified sampling:
$$\text{Var}E\left[E[f(G)| \prec]\right] \leq \text{Var}E[f(G)]$$

MCMC over orderings

- We use Metropolis-Hastings as before.
- One proposal is to flip 2 variables in the order, leaving the rest unchanged:

$$(X_{i_1}, \ldots, \mathbf{X_{i_j}}, \ldots, \mathbf{X_{i_k}}, \ldots, X_{i_n}) \rightarrow (X_{i_1}, \ldots, \mathbf{X_{i_k}}, \ldots, \mathbf{X_{i_j}}, \ldots, X_{i_n})$$

- Using score decomposability, only family scores for nodes inside the bold range need to be recomputed.
- This is much more expensive than MCMC in DAG space, but each move is much more powerful, and the space is much smaller.

Marginal likelihood given known node ordering

- If we know the ordering (eg. temporal), we have

$$P(D|\prec) = \sum_{G \in G_{d,\prec}} P(G|\prec) P(D|G)$$

- Given $\prec$, we can pick the parents for each node independently. Let $U_{i,\prec} = \{U : U \prec X_i, |U| \leq d\}$. Assuming $P(G|\prec)$ is uniform for legal graphs,

$$\begin{aligned} P(D|\prec) &= \sum_{G \in G_{d,\prec}} \prod_i \exp \mathsf{FamScore}(D(X_i, \pi_i)) \\ &= \prod_i \sum_{U_i \in U_{i,\prec}} \exp \mathsf{FamScore}(D(X_i, \pi_i)) \end{aligned}$$

- We marginalize out parameters θ and graph structures G.
- This is what we need to evaluate the MH acceptance probability.

Prob. feature given known node ordering

- Given a sampled ordering, we can compute the probability of a parent set

$$P(\pi_i^G = U | D, \prec) = \frac{\exp \mathsf{FamScore}(D(X_i, U))}{\sum_{U' \in U_{i,\prec}} \exp \mathsf{FamScore}(D(X_i, U'))}$$

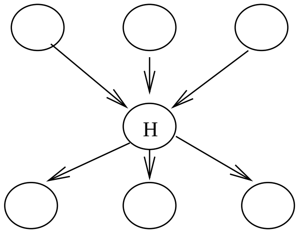
- From this, we can sample parents and hence graphs compatible with $\prec$.
- From this, we can compute probability of features such as "There is a directed path from X_i to X_j".
- Useful for determining features of biological networks from small sets of data (so the posterior is highly multimodal).

Outline

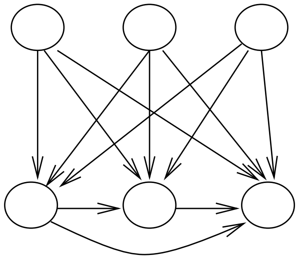
- Introduction $\surd$
- Bayesian model selection: basics $\surd$
- Bayesian model selection: tabular Bayes nets $\surd$
- Tree-structured models $\surd$
- Searching through DAGs $\surd$
- Searching through variable orderings $\surd$
- MCMC $\surd$
- Latent variables
- Undirected models
- Causality
- Application: reconstructing a cell signalling pathway

Hidden variables

- So far, we have assumed all variables have been observed.
- In this case, we can compute the Bayesian score (evidence) exactly.
- But hidden variables can simplify a model a lot
  eg. mixture models, HMMs.



17 parameters

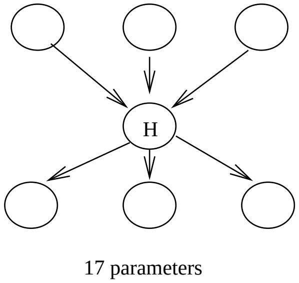


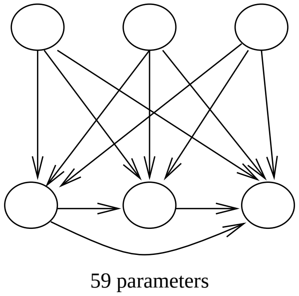
59 parameters

- Can still run local search to pick best model.
- But hidden variables raise various problems:
  - Efficiently computing the score from partially observed data.
  - Detecting the presence of latent (confounding) factors.
  - Inferring the dimensionality/ cardinality of latent factors.

Detecting presence of hidden variables

- One idea is to look for dense semi-cliques.





- Then insert a hidden variable “in the middle”, and let the search algorithm figure out the detailed “wiring”.
- Unfortunately, many scoring criteria (e.g., BIC) produce very sparse graphs, which makes such semi-cliques rare.
- Constraint-based methods sometimes can be used to detect confounding.
- In general, this is an open problem.

Approximating $p(D)$ in latent variable models

- When there are many hidden variables, the parameter posterior has an exponential number of modes, so computing the marginal likelihood is intractable

$$p(D) = \int_\theta [\prod_n \sum_h p(h, x_n|\theta)] p(\theta)$$

- There are various possible approximations:
  - BIC
  - Cheeseman-Stutz (CS) lower bound
  - Variational Bayes EM lower bound
  - Sampling

BIC approximation

- Use EM to compute $\hat{\theta}$, then compute

$$\begin{aligned} \text{score}_{BIC}(G|D) &= \log P(D|G,\hat{\theta}) - \frac{d(G)}{2}\log N_D \\ &= \sum_i \sum_{jk} N_{ijk} \log \hat{\theta}_{ijk} - \frac{d_i}{2}\log N_D \end{aligned}$$

  where $d_i = q_i(r_i - 1)$ is the number of parameters in X_i's CPT, $\hat{\theta}$ are the MLE parameters derived from N_{ijk}.

- In general, the effective dimensionality of a latent variable model $d(G)$ can be hard to compute.

Structural EM algorithm

- We can do local search, and run EM inside each step to evaluate the BIC score, but this is slow.
- Idea behind structural EM: run EM in the outer loop, and perform the structural search in the M step.
- We use the expected BIC score

$$\text{EBIC-score}(G) = \sum_i \sum_{jk} \langle N_{ijk} \rangle \log \hat{\theta}_{ijk} - \frac{d_i}{2} \log N_D$$

where

$$\langle N_{ijk} \rangle = \sum_m P(X_i = k, X_{\pi_i} = j | D_m, \theta, G)$$

Structural EM algorithm

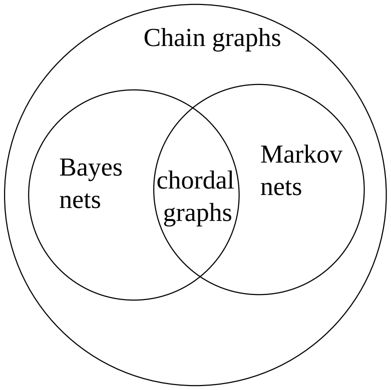
```
Choose $G$ somehow
Choose $\theta$ somehow
While not converged
        Improve $\theta$ using parametric EM
        For each $G'$ in nbd$(G)$
             Compute ESS$(G')$ using $G, \theta$ [E step]
             Compute $\hat{\theta}(G')$ using ESS$(G')$
             Compute Escore$(G')$ using ESS$(G'), \hat{\theta}(G')$
        $G^* := \arg\max_{G'}$ Escore$(G')$
        If Escore$(G^*) >$ Escore$(G)$
             then $G := G^*$ [structural M step]
                   $\theta := \theta(G^*)$ [parametric M step]
             else converged $:=$ true
```

Outline

- Introduction $\surd$
- Bayesian model selection: basics $\surd$
- Bayesian model selection: tabular Bayes nets $\surd$
- Tree-structured models $\surd$
- Searching through DAGs $\surd$
- Searching through variable orderings $\surd$
- MCMC $\surd$
- Latent variables $\surd$
- Undirected models
- Causality
- Application: reconstructing a cell signalling pathway

Learning MRF structure

- Essentially all of the same techniques used for learning BN structure (local search, MCMC, etc) can be used for learning MRF structure.
- The problem is how to compute $p(D|G)$.
- This is hard because parameter estimation does not decouple due to the partition function Z.
- An exception is decomposable graphical models, for which closed-form formulae can be written for the MLE (in the tabular and Gaussian cases).

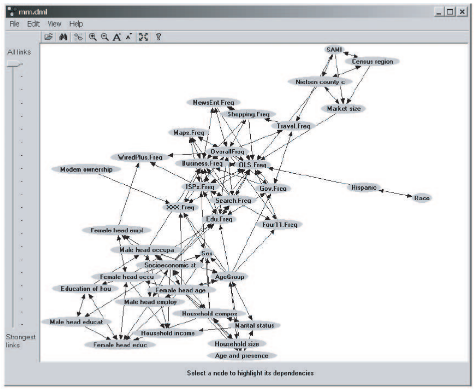


MRFs vs BNs

- MRF advantages
  - Cycles allowed
  - No need to search for ordering
- MRF disadvantages
  - Can be harder to learn
  - Cannot encode causality
  - Hard to define informative priors

Dependency networks

- A dependency network is a directed cyclic graph, consisting of a set of models $p(X_i|X_{-i})$.
- This implicitly defines a joint distribution via Gibbs sampling, but the resulting stationary distribution is not unique - it depends on the update order. (In the case of Gaussian networks, there are conditions which specify when a product of local conditionals is consistent with a global joint.)
- However, it is very easy to learn such models.



Select a node to highlight its dependencies

Outline

- Introduction $\surd$
- Bayesian model selection: basics $\surd$
- Bayesian model selection: tabular Bayes nets $\surd$
- Tree-structured models $\surd$
- Searching through DAGs $\surd$
- Searching through variable orderings $\surd$
- MCMC $\surd$
- Latent variables $\surd$
- Undirected models $\surd$
- Causality
- Application: reconstructing a cell signalling pathway

Causality

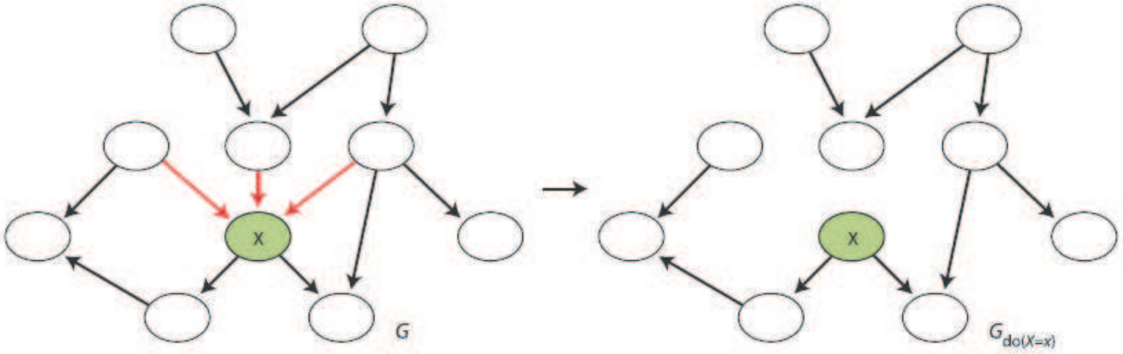
- If we interpret a Bayes net merely as encoding conditional independence, then $X \rightarrow Y$ and $X \leftarrow Y$ are Markov equivalent, and hence are indistinguishable from observational data.
- However, if we intervened and “wiggled” X, we could see if Y changes. Hence experimental data (where we “manipulate” certain variables) can be used to distinguish between members of an equivalence class.
- This is useful for domains like molecular biology where we can “knock out” genes, etc.

Pearl's do calculus

- Conditioning on observations is not the same as conditioning on events you have caused.
- e.g., suppose smoking → yellow-fingers.
- Let C = smokes and E = has yellow fingers.
- $P(C|E) > P(\neg C|E)$. However, $\neg(P(C|do(E)) > P(\neg C|do(E)))$.

Manipulation theorem (graph surgery)

- To reason about the effects of interventions, sever all incoming arcs from nodes that have been set. Then apply usual BN inference rules.
- To learn structure from interventional data, apply surgery to graph for each case as appropriate, then use usual scoring function (need not satisfy score equivalence).



Simpson's paradox

- We will show a dramatic example of the dangers of not thinking causally.
- Suppose taking a drug (cause C) decreases recovery rate (effect E) in females (F) and males ($\neg F$).
- How can it be possible that in the combined population, the drug apparently increases recovery rate?

	Combined				Male				Female			
	E	$\neg E$	Total	Rate	E	$\neg E$	Total	Rate	E	$\neg E$	Total	Rate
C	20	20	40	50%	18	12	30	60%	2	8	10	20%
$\neg C$	17	24	40	40%	7	3	10	70%	9	21	30	30%
Total	36	44	80		25	15	40		11	29	40	

Simpson's paradox

- Suppose taking a drug (cause C) decreases recovery rate (effect E) in females (F) and males ($\neg F$)

$$P(E|C,F) < P(E|\neg C,F)$$
$$P(E|C,\neg F) < P(E|\neg C,\neg F)$$

- but in the combined population, the drug increases recovery rate

$$P(E|C) > P(E|\neg C)$$

- By the rules of probability, this is perfectly possible.
- But it goes counter to intuition. Why?
- Put another way: given a new patient, do we use the drug or not?

> The apparent answer is that when we know the gender of the patient, we do not use the drug, but if the gender is unknown, we should use the drug. Obviously that conclusion is ridiculous.
> — Novick, 1983.

Paradox resolved

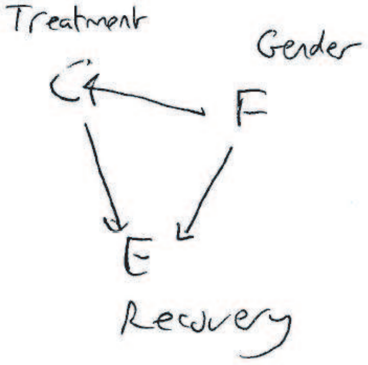
- The statement that the drug C causes recovery E is

$$P(E|do(C)) > P(E|do(\neg C))$$

whereas the data merely tell us

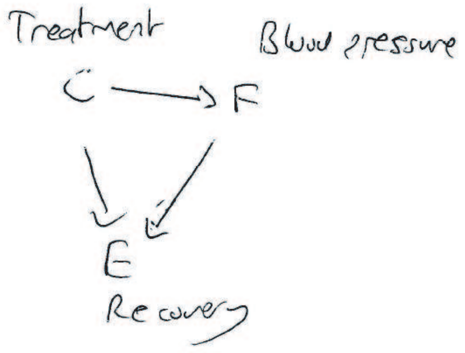
$$P(E|C) > P(E|\neg C)$$

- This is not a contradiction. Observing C is positive evidence for E, since more males than females take the drug, and the male recovery rate is higher (regardless of the drug).
- Gender is a confounding factor. We should not use the drug.



A different cover story

- Suppose we keep the data the same but interpret F as something that is affected by C, such as blood pressure.
- Now we see that we should not condition on F (since that would block one of the causal pathways), and instead should use the combined table to infer that we should use the drug.
- Different causal assumptions (which are statistically indistinguishable) lead to different actions.
- We need prior domain knowledge to distinguish these models.



Inferring causality from observational data

- It is clearly impossible to distinguish members of an equivalence class without interventional data: "no causation without manipulation".
- However, we can distinguish the v-structure $X \rightarrow Y \leftarrow Z$ from $\{X \rightarrow Y \rightarrow Z, X$ (PDAG $X - Y - Z$) using observational data alone, since in the former, $X \perp Z$ and $X \not\perp Z|Y$, but in the latter, $X \not\perp Z$ and $X \perp Z|Y$.
- Pearl and Verma (1991) and Spirtes, Glymour and Scheines (1993) constructed various algorithms (IC/PC/IC*/FCI - implemented in Tetrad) that they proved will identify the "true" PDAG - up to Markov equivalence - even in the presence of confounding factors.
- We assume the model is faithful/ stable, i.e., no "accidental" (non structural) independencies. e.g. if we have a v-structure $X \rightarrow Z \leftarrow Y$, then Z always depends on both X and Y (no context specific independence allowed).

Constraint-based approach: PC algorithm

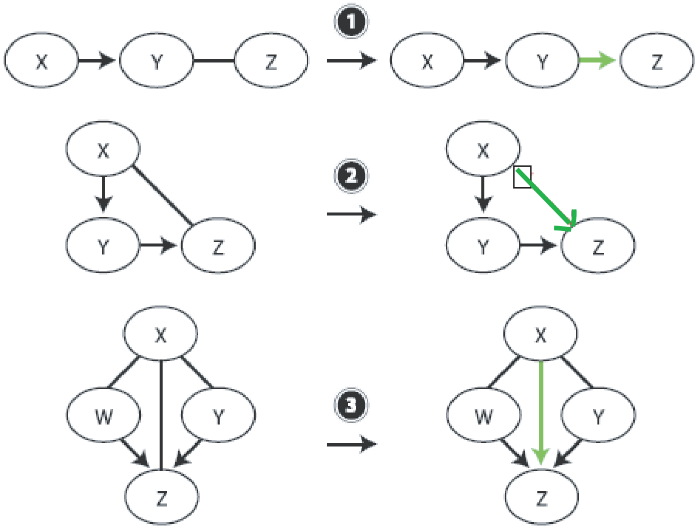
- IC (Inference of Causation) algorithm, Pearl and Verma 1991; PC (Peter and Clark) algorithm, Spirtes and Glymour 1993.
- Algorithm idea

```
Initialize $G$ to fully connected graph
For $k = 0, 1, \ldots$
    For each $i - j$ in $G$ s.t. $|nbd(i) \setminus \{j\}| > k$
        For each set $S \subseteq nbd(i) \setminus \{j\}$ s.t. $|S| = k$
        If $X_i \perp X_j | X_S$ then remove $i - j$ from $G$
Orient as many edges as possible and return resulting PDAG
```

- Widely applied to social science data.

Orienting edges

- We can identify v-structures because of their unique statistical signature.
- Then we can reason about the orientation of some of the other edges, and propagate these constraints to get a PDAG.



Problems with the constraint based approach

- The constraint-based approach first uses statistical tests (e.g., χ^2) to detect statistical dependencies amongst variables, and then reasons deductively about causal structures that could account for these dependencies.
- The problem is that the statistical tests impose an arbitrary threshold on the evidence that data provide for a causal relationship.
- Thus these methods cannot combine weak sources of evidence, or maintain graded degrees of belief.
- In realistic domains, sample sizes will be small, so we will need strong priors to overcome the huge amounts of uncertainty.

Occam's razor

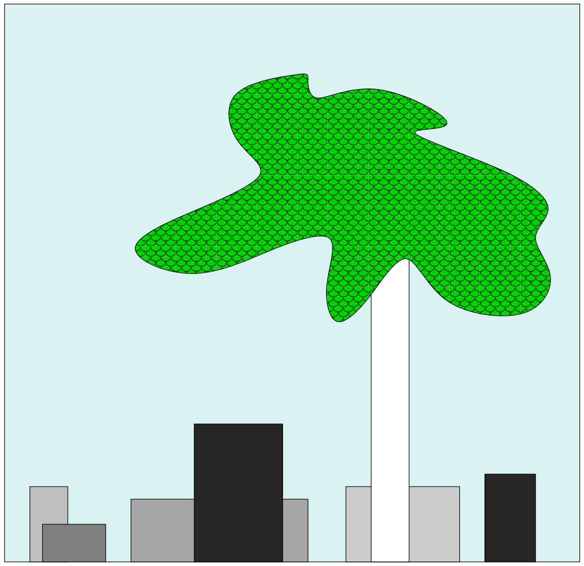
- It is always possible that a given correlation is due to a hidden common cause.
- However, Occam's razor argues that it may be more parsimonious to explain certain statistical signatures in terms of causality (e.g., due to v-structures) than in terms of hidden causes.
- Pearl writes

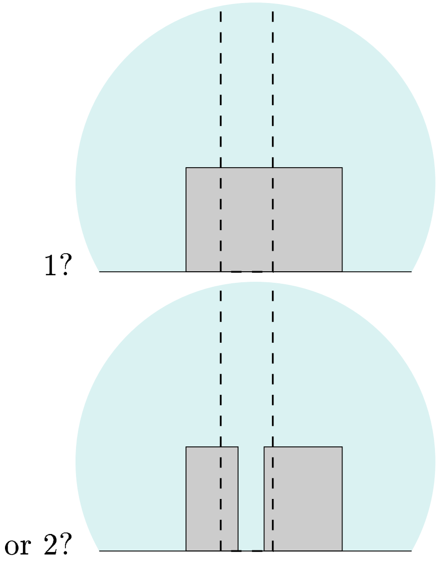
  > How safe are the causal relationships inferred by [these] algorithms? We may equally well ask: how safe are our predictions when we recognize 3D objects from their 2D appearance?

- In other words, it may be optimal to believe in causation from a Bayesian decision theoretic viewpoint, even if you are not guaranteed to always be right.

Bayesian image interpretation

- How many boxes behind the tree?
- The intrepretation that the tree is in front of one box is much more probable than there being 2 boxes which happen to have the same height and color (suspicious coincidence).
- This can be formalized by assuming (uniform) priors on the box parameters, and computing the Bayes factors.





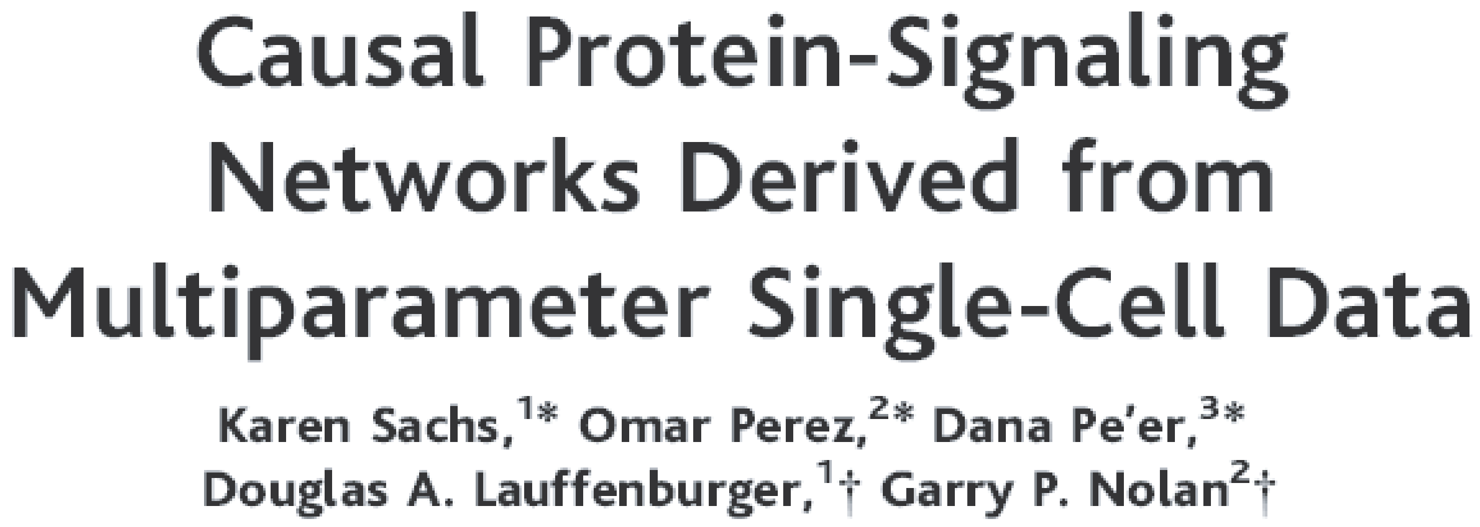
Causality: further reading

- There are many books (mostly non-Bayesian) on causal inference
  - *Causality*, Judea Pearl, 2000.
  - *Causation, Prediction and Search*, Spirtes, Glymour and Scheines, 2000 (2nd edn)
  - *Computation, Causation and Discovery*, eds Glymour, Cooper, 1999.
  - *Cause and Correlation in Biology*, Bill Shipley, 2000
- For a Bayesian approach, applied to cognitive psychology, see the papers by Josh Tenenbaum et al
  - "Structure learning in human causal induction", Josh Tenenbaum and Tom Griffiths, NIPS 2001.
  - "Causes, coincidences and theories", Tom Griffiths, PhD thesis, MIT 2005

OUTLINE

- Introduction $\surd$
- Bayesian model selection: basics $\surd$
- Bayesian model selection: tabular Bayes nets $\surd$
- Tree-structured models $\surd$
- Searching through DAGs $\surd$
- Searching through variable orderings $\surd$
- MCMC $\surd$
- Latent variables $\surd$
- Undirected models $\surd$
- Causality $\surd$
- Application: reconstructing a cell signalling pathway

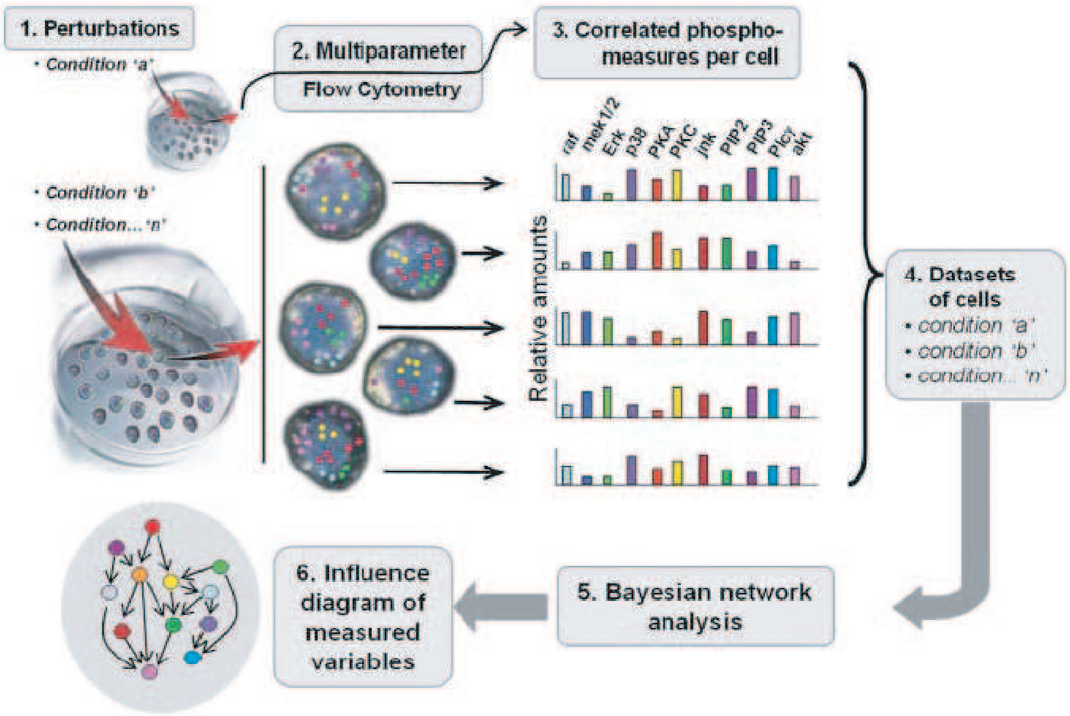
Case study (Sachs et al, Science, April 2005)



Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data

Karen Sachs,[1]* Omar Perez,[2]* Dana Pe'er,[3]*
Douglas A. Lauffenburger,[1]† Garry P. Nolan[2]†

Measurement of human T cell signalling system



1. Perturbations
- Condition 'a'
- Condition 'b'
- Condition... 'n'

2. Multiparameter Flow Cytometry

3. Correlated phospho-measures per cell

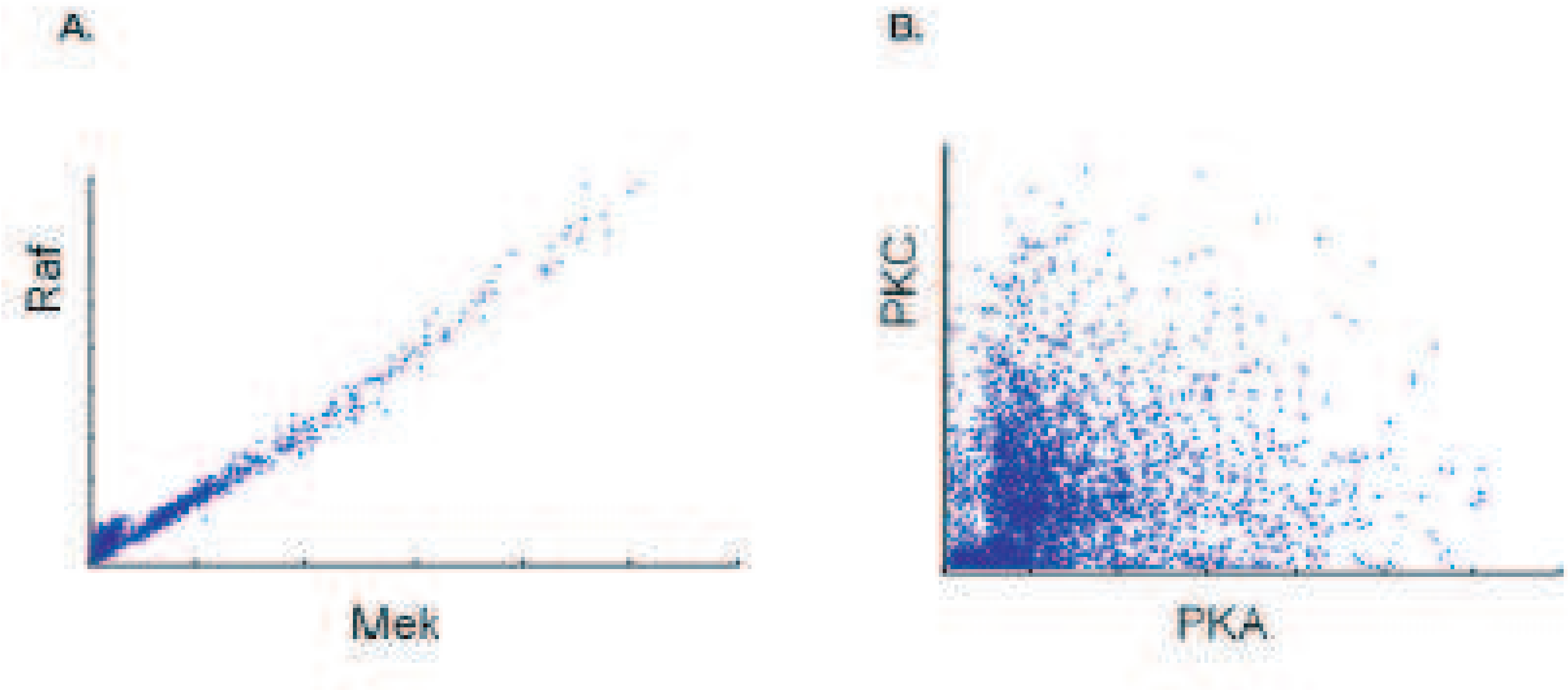
4. Datasets of cells
- condition 'a'
- condition 'b'
- condition... 'n'

5. Bayesian network analysis

6. Influence diagram of measured variables

- Measured 11 protein phosphorylation levels in 600 individual cells (no population averaging) using flow cytometry
- Performed 9 perturbations (so total sample size is $N_D = 600 \times 9 = 5400$)

Data

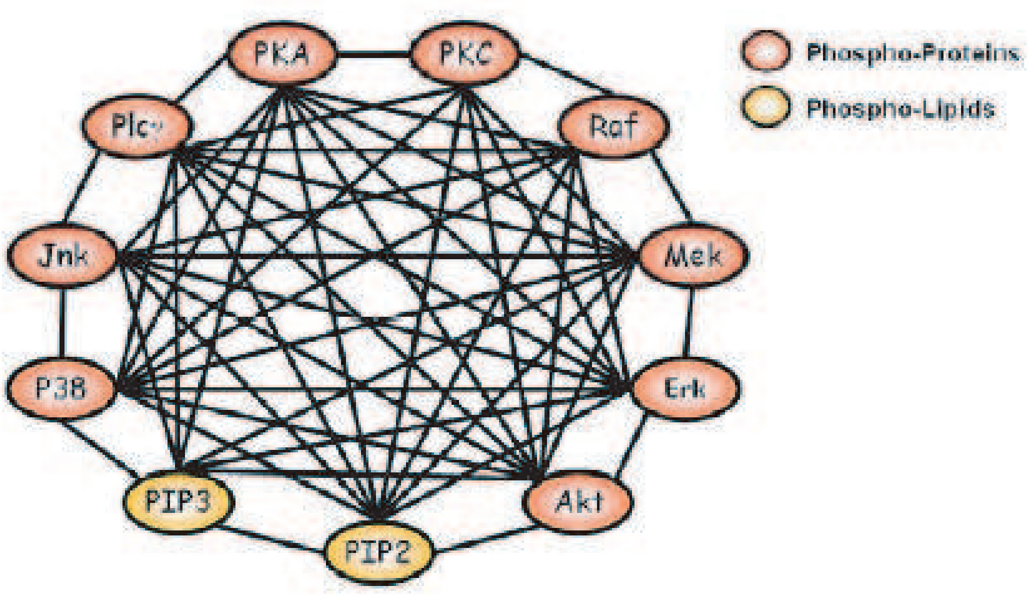


- Scatterplots of two sets of pairs. Raf causes Mek, and PKC (weakly) causes PKA.

Method

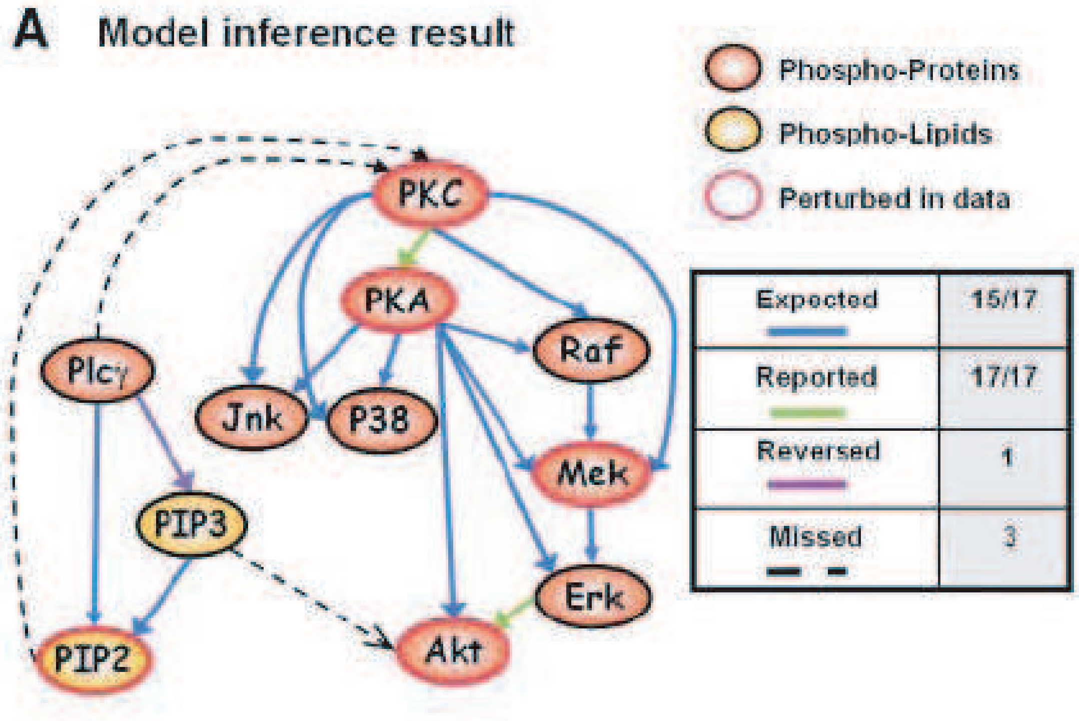
- Data points that were more than 3σ from the mean were thrown out.
- Data were discretized into 3 levels (low, medium and high) using an agglomerative clustering technique.
- Multiple restart simulated annealing was used to get a sample of 500 high scoring graphs.
- The BDe score was used (modified for interventions).
- The final inferred network contains arcs that occur in at least 85% of the high scoring networks.

Correlation analysis



- 52 of the 55 possible pairs are significantly correlated (using a Bonferroni corrected p-value).

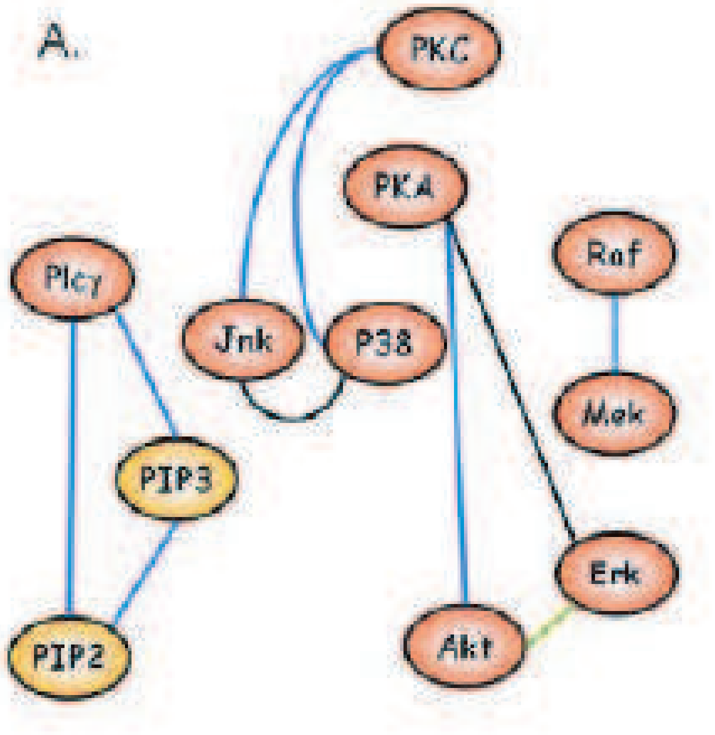
Bayes net analysis ($N = 5400$)



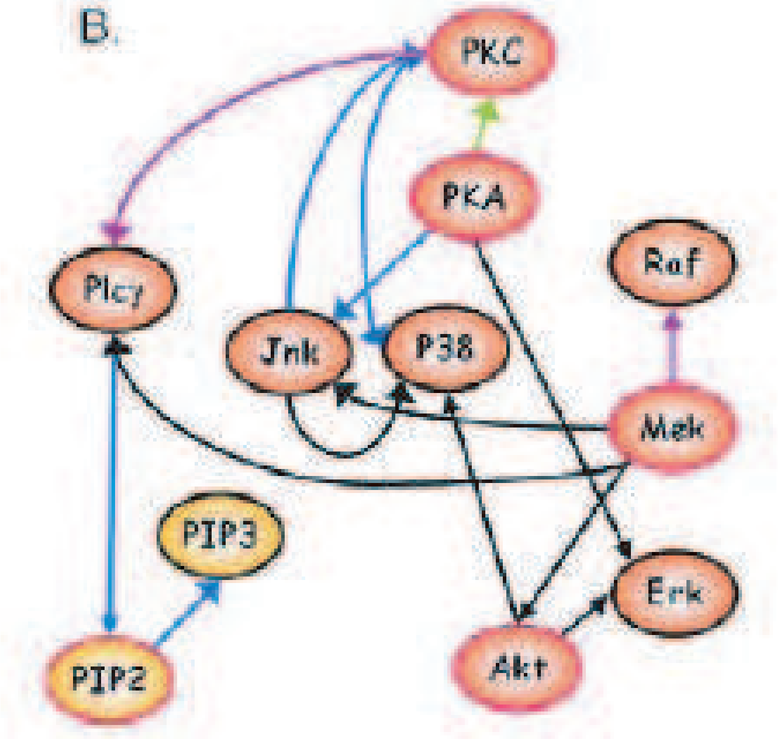
Expected	15/17
Reported	17/17
Reversed	1
Missed	3

- Of the 17 arcs in the model, 15 were expected (well known), and 2 had been reported (no false positives); 3 known edges were missed (3 false negatives). All but one edge directions were correct.

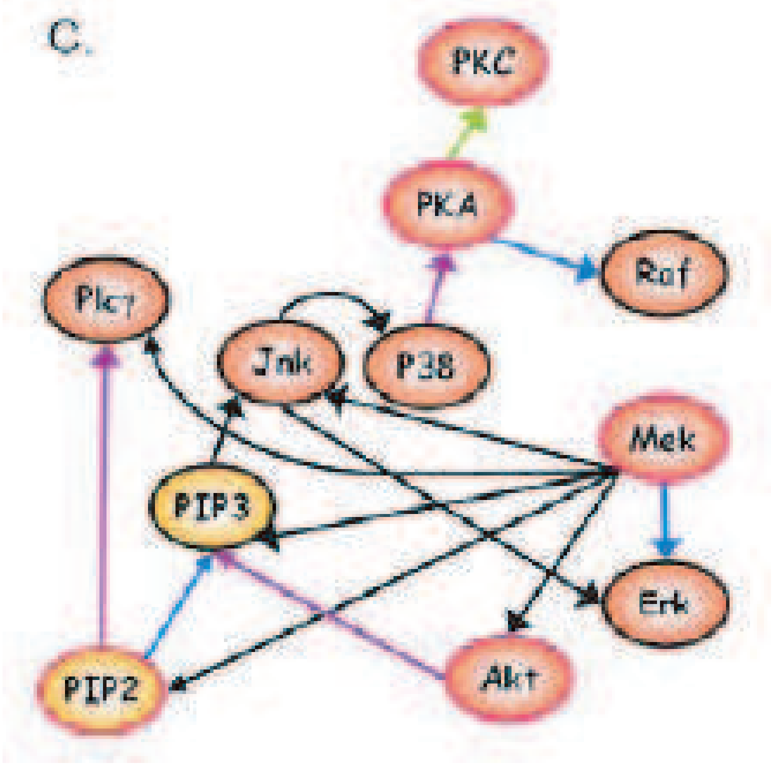
EFFECTS OF NO INTERVENTIONS ($N = 1200$)



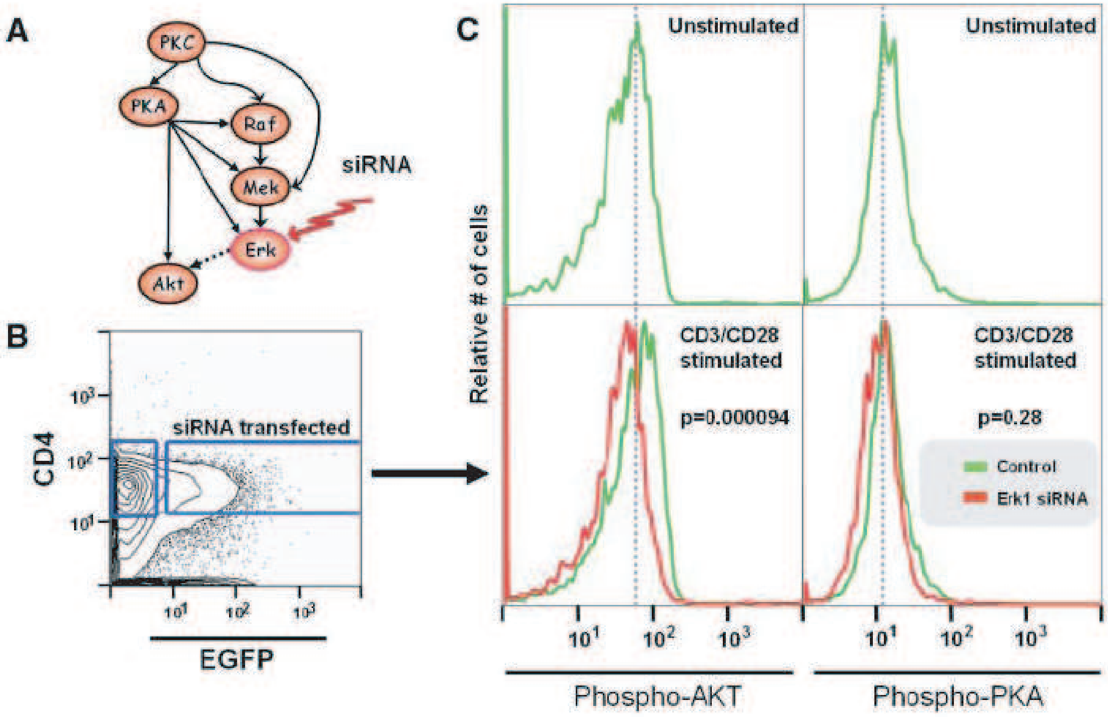
Effects of small sample size ($N = 420$)



Effects of population averaging ($N = 420$)



Verification of inferred causal links



- By inhibiting ERK (using small interfering RNAs) they verified that Akt is reduced to background levels, but PKA is unaffected.

Open problems

- Efficiently computing $p(D)$ for models with latent variables, non conjugate priors, etc.
- Devising more efficient optimization methods than local search.
- Active learning.
- Automatically discovering/ inventing latent variables.
- Using rich prior knowledge (e.g., domain theories) to aid inference.
- Automatically abstracting from ground network instances into general theories (c.f., ILP).