Probabilistic Graphical Models

Lecture 5 – Bayesian Learning of Bayesian Networks

CS/CNS/EE 155

Andreas Krause

Announcements

- Recitations: Every Tuesday 4-5:30 in 243 Annenberg
- Homework 1 out. Due in class Wed Oct 21
- Project proposals due Monday Oct 19

Project proposal

- At most 2 pages. One proposal per project
- due Monday Oct 19
- Please clearly specify
  - What is the idea of this project?
  - Who will be on the team?
  - What data will you use? Will you need time "cleaning up" the data?
  - What code will you need to write? What existing code are you planning to use?
  - What references are relevant? Mention 1-3 related papers.
  - What are you planning to accomplish by the Nov 9 milestone?

Project ideas

- Ideally, do graphical model project related to your research (and, e.g., data that you're working with)
  - Must be a new project started for the class!
- Website has examples for
  - Project ideas
  - Data sets
  - Code

Project ideas

- All projects should involve using PGMs for some data set, and then doing some experiments
- Learning related
  - Experiment with different algorithms for structure / parameter learning
- Inference related
  - Compare different algorithms for exact or approximate inference
- Algorithmic / decision making
  - Experiment with algorithms for value of information, MAP assignment, …
- Application related
  - Attempt to answer interesting domain-related question using graphical modeling techniques

Data sets

- Some cool data sets made available specifically for this course!!
  ➔ Contact TAs to get access to data.
- Exercise physiological data (collected by John Doyle's group)
  - E.g., do model identification / Bayesian filtering
- Fly data (by Pietro Perona and Michael Dickinson et al.)
  - "Activity recognition" – what are the patterns in fly behavior? Clustering / segmentation of trajectories?
- Urban challenge data (GPS data + LADAR + Vision) by Richard Murray et al.
  - Sensor fusion using DBNs; SLAM
- JPL MER data by Larry Matthies et al.
  - Predict slip based on orbital imagery + GPS tracks
  - Segment images to identify dangerous areas for rover
- LDPC decoding
  - Compare new approximate inference techniques with Loopy-BP
- Other open data sets mentioned on course webpage

Code

- Libraries for graphical modeling by Intel, Microsoft, …
- Toolboxes
  - computer vision image manipulations
  - Topic modeling
  - Nonparametric Bayesian modeling (Dirichlet processes / Gaussian processes / …)

Learning general BNs

	Known structure	Unknown structure
Fully observable	Easy :)	hard 2.
Missing data	hard 3. (EM)	very hard (last)

Algorithm for BN MLE

Given BN structure G

For each variable X_i

learn $\hat{\theta}_{X_i|Pa_i} = \frac{\text{Count}(X_i, Pa_i)}{\text{Count}(Pa_i)}$

$\Rightarrow$ globally maximum likelihood estimate for fixed structure G

Structure learning

- Two main classes of approaches:
- Constraint based
  - Search for P-map (if one exists):
  - Identify PDAG
  - Turn PDAG into BN (using algorithm in reading)
  - **Key problem**: Perform independence tests
- Optimization based ← coming up!
  - Define scoring function (e.g., likelihood of data)
  - Think about structure as parameters
  - More common; can solve simple cases exactly

MLE for structure learning

- For fixed structure, can compute likelihood of data

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = \sum_{\ell} \sum_{i} \log P(X_i = x_i^{(\ell)} \mid \mathbf{Pa}_i = \mathbf{pa}_i^{(\ell)} ; \theta_G ; G)$$

$$\overset{MLE}{=} \sum_i \sum_{x_i} \sum_{pa_i} \mathrm{Count}(x_i, pa_i) \log \frac{\hat{P}(x_i, pa_i)}{\hat{P}(pa_i)}$$

$$= m \sum_i \sum_{x_i} \sum_{pa_i} \hat{P}(x_i, pa_i) \log \frac{\hat{P}(x_i, pa_i)\,\hat{P}(x_i)}{\hat{P}(pa_i)\,\hat{P}(x_i)}$$

$$= m \sum_i \sum_{x_i} \sum_{pa_i} \hat{P}(x_i, pa_i) \log \frac{\hat{P}(x_i, pa_i)}{\hat{P}(x_i)\hat{P}(pa_i)} + m \sum_i \sum_{x_i} \sum_{pa_i} \hat{P}(x_i, pa_i) \log \hat{P}(x_i)$$

$$= m \sum_i \hat{I}(X_i ; Pa_i) - m \sum_i \hat{H}(X_i)$$

Decomposable score

- Log-data likelihood

$$\log \widehat{P}(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) = m \sum_i \underline{\widehat{I}(X_i, \mathbf{Pa}_i)} - m \sum_i \widehat{H}(X_i)$$

independent of graph structure!

- MLE score decomposes over families of the BN (nodes + parents)
- Score(G ; D) = $\sum_i$ FamScore(X_i | Pa_i; D)
- Can exploit for computational efficiency!

Finding the optimal MLE structure

- Log-likelihood score:

$$\mathrm{Score}(\mathcal{G};\mathcal{D}) = \sum_i \widehat{I}(X_i, \mathbf{Pa}_i)$$

- Want G^* = argmax_G Score(G ; D)
- Lemma: $G \subseteq G'$ ➔ Score(G; D) $\leq$ Score(G'; D)

Complete graph
maximizes log data likelihood!

"Information never hurts"
RV X, $A \subset B$
$H(X|A) \geq H(X|B)$
$I(X;A) = H(X) - H(X|A)$
$\Rightarrow I(X;B) \geq I(X;A)$

Finding the optimal MLE structure

- Optimal solution for MLE is always the fully connected graph!!! ☹
  - → Non-compact representation; Overfitting!!
- Solutions:
  - Priors over parameters / structures (later)
  - Constraint optimization (e.g., bound #parents)

Chow-Liu algorithm

- For each pair X_i, X_j of variables compute

$$\widehat{P}(x_i, x_j) = \frac{\mathrm{Count}(x_i, x_j)}{m}$$

- Compute mutual information

$$\widehat{I}(X_i, X_j) = \sum_{x_i, x_j} \widehat{P}(x_i, x_j) \log \frac{\widehat{P}(x_i, x_j)}{\widehat{P}(x_i)\widehat{P}(x_j)}$$

- Define complete graph with weight of edge (X_i,X_i) given by the mutual information
- Find maximum spanning tree ➔ skeleton
- Orient the skeleton using breadth-first search

Today: Bayesian learning

- X Bernoulli variable
- Which is better:
  - Observe 1 H and 2 T $\hat{\theta} = \frac{1}{3}$
  - Observe 10 H and 20 T $\hat{\theta} = \frac{1}{3}$
  - Observe 100 H and 200 T $\hat{\theta} = \frac{1}{3}$

- MLE is same in all three cases
- However, should be much more “confident” about MLE if we have more data

  ➔ Want to model distributions over parameters

Bayesian learning

- Make prior assumptions about parameters P(θ)
- Compute posterior

$$P(\theta \mid D) = \frac{P(\theta)\,P(D \mid \theta)}{P(D)} \propto P(\theta)\,P(D \mid \theta)$$

Given data D want to predict

$$P(X \mid D) = \int P(\theta \mid D)\,\underline{P(X \mid \theta)}\,d\theta$$

In MLE

$$P(X \mid D) \approx P(X \mid \hat{\theta}) \qquad \hat{\theta} = \arg\max P(D \mid \theta)$$

Bayesian Learning for Binomial

$$P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta) P(\theta)$$

- Likelihood function:

$$P(\mathcal{D} \mid \theta) = \theta^{m_H} (1 - \theta)^{m_T}$$

- How do we choose prior?
  - Many possible answers…
  - Pragmatic approach: Want computationally "simple" (and still flexible) prior

Conjugate priors

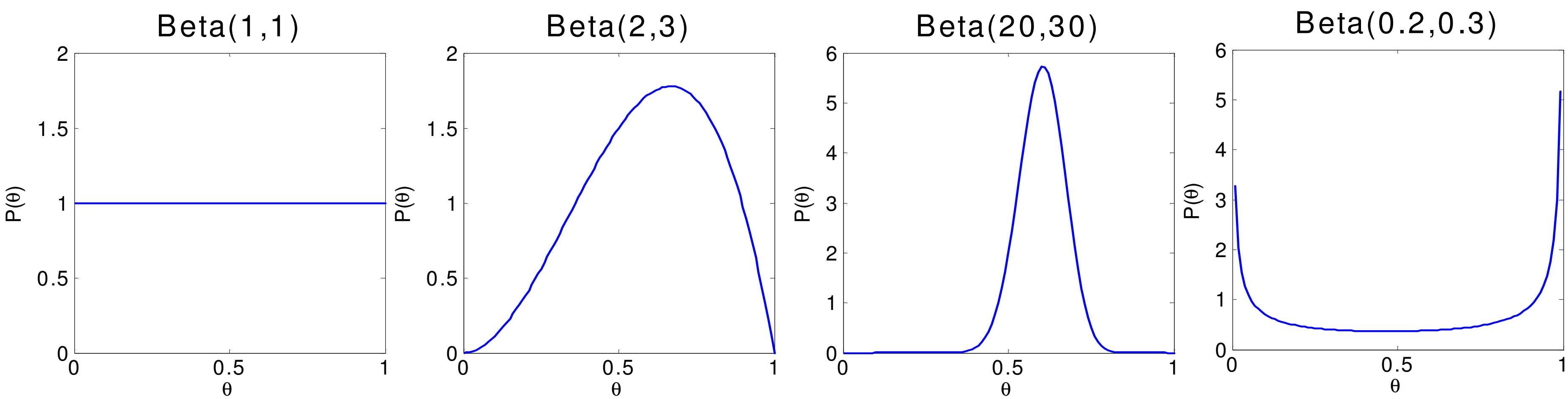
- Consider parametric families of prior distributions:
  - $P(\theta) = f(\theta; \alpha)$
  - α is called “hyperparameters” of prior
- A prior $P(\theta) = f(\theta; \alpha)$ is called **conjugate** for a likelihood function $P(D \mid \theta)$ if $P(\theta \mid D) = f(\theta; \alpha')$
  - Posterior has same parametric form
  - Hyperparameters are updated based on data D

- Obvious questions (answered later):
  - How to choose hyperparameters??
  - Why limit ourselves to conjugate priors??

Conjugate prior for Binomial

- Beta distribution

$$\mathrm{Beta}(\theta; \alpha_H, \alpha_T) = \frac{\theta^{\alpha_H - 1}(1-\theta)^{\alpha_T - 1}}{\underbrace{B(\alpha_H, \alpha_T)}_{\text{Normalization constant}}}$$



Posterior for Beta prior

- Beta distribution

$$P(\theta) = \mathrm{Beta}(\theta; \alpha_H, \alpha_T) = \frac{\theta^{\alpha_H - 1}(1-\theta)^{\alpha_T - 1}}{B(\alpha_H, \alpha_T)}$$

- Likelihood:

$$P(\mathcal{D} \mid \theta) = \theta^{m_H}(1-\theta)^{m_T}$$

- Posterior:

$$P(\theta|D) \propto P(\theta) P(D|\theta) \propto \theta^{\alpha_H + m_H - 1} (1-\theta)^{\alpha_T + m_T - 1}$$

$$P(\theta|D) = \mathrm{Beta}(\theta; \alpha_H + m_H, \alpha_T + m_T)$$

Bayesian prediction

- Prior P(θ) = Beta(α_H,α_T}) Bernoulli: $P(X=H)=\theta$
- Suppose we observe D= {m_H heads, and m_T tails}
- What's P(X=H | D), i.e., prob. that next flip is heads?

$$P(X=H \mid D) = \underbrace{\int \theta \, P(\theta \mid D)\, d\theta} = \mathbb{E}[\theta \mid D] = \frac{\alpha_H + m_H}{\alpha_H + \alpha_T + m_H + m_T}$$

Prior = Smoothing

$$\mathbb{E}[\theta] = \frac{m_H + \alpha_H}{\underbrace{m_H + m_T}_{m} + \underbrace{\alpha_H + \alpha_T}_{m'}} = \frac{m_H + \gamma m'}{\underbrace{m + m'}_{(*)}}$$

- Where m' = $\alpha_H + \alpha_T$, and $\gamma = \alpha_H$ / m' , $0 \le \gamma \le 1$
- m' is called "equivalent sample size"
  ➔ "hallucinated" coin flips

$$E[\theta] = \frac{m}{m+m'} \underbrace{\frac{m_H}{m}}_{\text{MLE}} + \frac{m'}{m+m'} \underbrace{\gamma}_{\text{prior mean}}$$

$m \to \infty$ $E[\theta] \to$ MLE   Forget prior

$m = 0$ prior

➔ Interpolate between MLE and prior mean

Conjugate for multinomial

- If X$\in${1,…,k} has k states:
- Multinomial likelihood

$$P(\mathcal{D} \mid \theta) = \theta_1^{m_1} \theta_2^{m_2} \dots \theta_k^{m_k}$$

where $\sum_i \theta_i = 1$, $\theta_i \geq 0$

- Conjugate prior: Dirichlet distribution

$$P(\theta) = \mathrm{Dir}(\theta; \alpha_1, \dots, \alpha_k) = \frac{1}{Z} \prod_i \theta_i^{\alpha_i - 1}$$

- If observe D={m_1 1s, m_2 2s, … m_k ks}, then

$$P(\theta \mid \mathcal{D}) = \mathrm{Dir}(\theta; \alpha_1 + m_1, \dots, \alpha_k + m_k)$$

Parameter learning for CPDs

- Parameters $P(X \mid Pa_X)$
- Have one parameter $\theta_{X \mid pa_X}$ for each value of parents pa_X

$$P(\theta_{X \mid Pa_X = u}) = Dir(\alpha_1 \ldots \alpha_k)$$

$$P(\theta_{X \mid Pa_X = u_1}, \ldots, \theta_{X \mid Pa_X = u_N}) = \prod_u P(\theta_{X \mid Pa_X = u})$$

"local parameter independence"

Parameter learning for BNs

- Each CPD P(X | Pa_X; $\theta_{X|Pa_X}$) has its own sets of parameters P($\theta_{X|pa_X}$)

  ➔ Dirichlet distribution
- Want to compute posterior over all parameters

$$P(\theta_{X_1|\mathbf{Pa}_{X_1}}, \ldots, \theta_{X_n|\mathbf{Pa}_{X_n}} \mid \mathcal{D})$$

- How can we do this??
- **Crucial assumption**: Prior distribution over parameters factorizes (“global parameter independence”)

$$P(\theta_{X_1|\mathbf{Pa}_{X_1}}, \ldots, \theta_{X_n|\mathbf{Pa}_{X_n}}) = \prod_i P(\theta_{X_i|\mathbf{Pa}_{X_i}})$$

Parameter Independence

- Assume

$$P(\theta_{X_1|\mathbf{Pa}_{X_1}}, \ldots, \theta_{X_n|\mathbf{Pa}_{X_n}}) = \prod_i P(\theta_{X_i|\mathbf{Pa}_{X_i}})$$

- Why useful?
- If data is fully observed, then

$$P(\theta_{X_1|\mathbf{Pa}_{X_1}}, \ldots, \theta_{X_n|\mathbf{Pa}_{X_n}} \mid \mathcal{D}) = \prod_i P(\theta_{X_i|\mathbf{Pa}_{X_i}} \mid \mathcal{D})$$

- I.e., posterior still independent. Why??

Meta-BN with parameters

Meta-BN

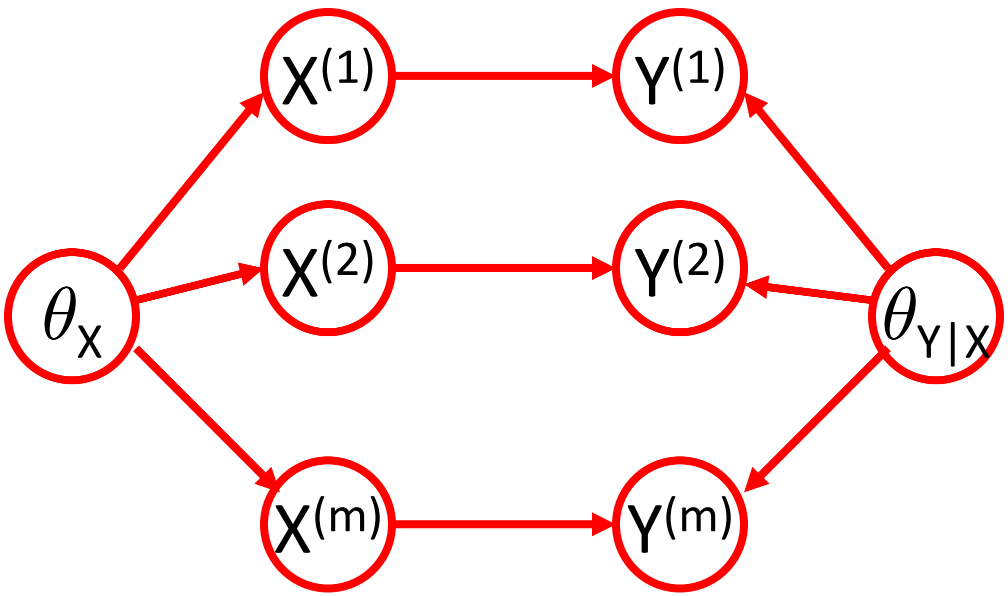
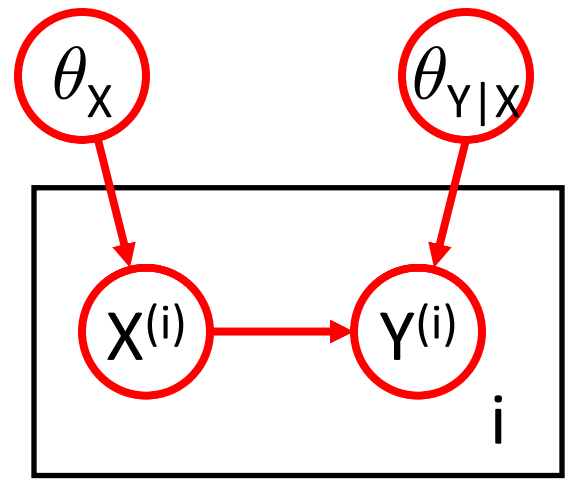


Plate notation



Meta BN contains one copy of original BN per data sample, and one variable for each parameter

Under parameter-independences, data d-separates parameters

Also: Parameters d-separate copies of BN: $P(D, X \mid \theta) = P(D \mid \theta)\, P(X \mid \theta)$

Bayesian learning of Bayesian Networks

- Specifying priors helps overfitting
  - Do not commit to fixed parameter estimate, but maintain distribution
- So far: Know how to specify priors over parameters for fixed structure.
- Why should we commit to fixed structure??
- Fully Bayesian inference

$$P(\underline{\mathbf{X} \mid \mathcal{D}}) \propto \sum_{\mathcal{G}} \underbrace{P(\mathcal{G})}_{\text{Prior over structures}} \int \underbrace{P(\theta_{\mathcal{G}} \mid \mathcal{G})}_{\text{Prior over param.}} \underbrace{P(\mathcal{D} \mid \mathcal{G}, \theta_{\mathcal{G}})}_{\text{Likelihood of data}} \underbrace{P(\mathbf{X} \mid \mathcal{D}, \mathcal{G}, \theta_{\mathcal{G}})}_{\substack{\text{likelihood of pred. vars} \\ = P(X \mid G, \theta_G)}} d\theta$$

Fully Bayesian inference

$$P(\mathbf{X} \mid \mathcal{D}) \propto \sum_{\mathcal{G}} P(\mathcal{G}) \int P(\theta_{\mathcal{G}} \mid \mathcal{G}) P(\mathcal{D} \mid \mathcal{G}, \theta_{\mathcal{G}}) P(\mathbf{X} \mid \mathcal{G}, \theta_{\mathcal{G}}) d\theta$$

- P(G): Prior over graphs
  - E.g.: P(G) = exp(-c Dim(G))

Dim (G) = # free params

- Called "Bayesian Model Averaging"
- Hopelessly intractable for larger models
- Often: want to pick most likely structure:

$$\mathcal{G}^* = \underset{\mathcal{G}}{\operatorname{argmax}}\, P(\mathcal{G} \mid \mathcal{D}) = \underset{\mathcal{G}}{\operatorname{argmax}} \log P(\mathcal{G}) + \log P(\mathcal{D} \mid \mathcal{G})$$

Why do priors help overfitting?

$$P(\mathcal{D} \mid \mathcal{G}) = \int P(\mathcal{D} \mid \mathcal{G}, \theta_{\mathcal{G}}) dP(\theta_{\mathcal{G}} \mid \mathcal{G})$$

- This Bayesian Score is tricky to analyze. Instead use:

$$\log P(\mathcal{D} \mid \mathcal{G}) \approx \log P(\mathcal{D} \mid \mathcal{G}, \widehat{\theta_{\mathcal{G}}}) - \frac{\log m}{2} \mathrm{Dim}(\mathcal{G})$$

- Why??
- **Theorem**: For Dirichlet priors, and for m$\rightarrow\infty$:

$$\log P(\mathcal{D} \mid \mathcal{G}) \rightarrow \log P(\mathcal{D} \mid \mathcal{G}, \widehat{\theta_{\mathcal{G}}}) - \frac{\log m}{2} \mathrm{Dim}(\mathcal{G}) + \mathcal{O}(1)$$

BIC score

$$\log P(\mathcal{D} \mid \mathcal{G}) \approx \log P(\mathcal{D} \mid \mathcal{G}, \widehat{\theta_{\mathcal{G}}}) - \frac{\log m}{2} \operatorname{Dim}(\mathcal{G})$$

- This approximation is known as **Bayesian Information Criterion** (related to Minimum Description Length)

$$\log P(\mathcal{D} \mid \mathcal{G}) \approx m \sum_i \Big(\widehat{I}(X_i; \mathbf{Pa}_i) - \widehat{H}(X_i) \Big) - \frac{\log m}{2} \operatorname{Dim}(\mathcal{G})$$

- Trades goodness-of-fit and structure complexity!
- Decomposes along families (computational efficiency!)
- Independent of hyperparameters! (Why??)

Consistency of BIC

- Suppose true distribution has P-map G*
- A scoring function Score(G ; D) is called **consistent**, if, as m → ∞ and probability → 1 over D:
  - G* maximizes the score
  - All non-I-equivalent structures have strictly lower score
- **Theorem**: BIC Score is consistent!
- Consistency requires m → ∞. For finite samples, priors matter!

Parameter priors

- How should we choose priors for discrete CPDs?
- Dirichlet (computational reasons). But how do we specify hyperparameters??
- K2 prior:
  - Fix α
  - $P(\theta_{X \mid Pa_X}) = Dir(\alpha,\ldots,\alpha)$
- Is this a good choice?

X   Y

$P(\theta_Y) = Dir(\alpha, \alpha)$

$\Rightarrow$ Equiv. sample size 2α

$X \rightarrow Y$

$P(\theta_{Y \mid X=H}) = Dir(\alpha, \alpha)$

$P(\theta_{Y \mid X=T}) = Dir(\alpha, \alpha)$

$\Rightarrow$ Equiv sample size 4α

BDe prior

- Want to ensure "equivalent sample size" m' is constant
- Idea:
  - Define P'(X_1,…,X_n)

    For example: P'(X_1,…,X_n) = $\prod_i$ Uniform(Val(X_i))
  - Choose equivalent sample size m'
  - Set $\alpha_{x_i \mid pa_i} = m'$ P'(x_i, pa_i)

$X \quad Y \qquad\qquad X \rightarrow Y$

$$\alpha_y = m' P'(y) = m' \sum_x P'(x,y) = \sum_x \alpha_{y|x}$$

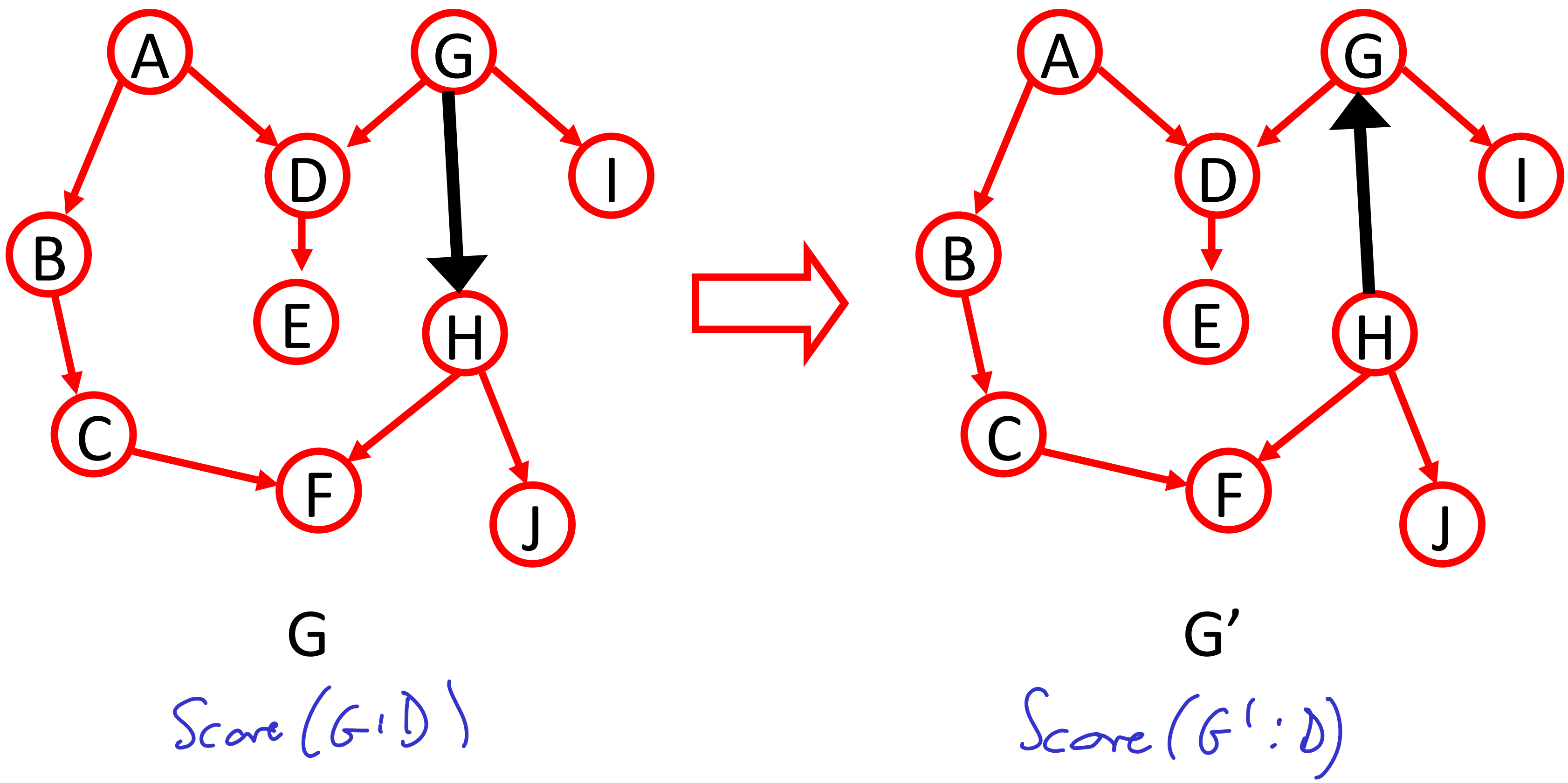
Bayesian structure search

- Given consistent scoring function Score(G : D), want to find to find graph G* that maximizes the score
- Finding the optimal structure is **NP-hard** in most interesting cases (details in reading). ☹
- Can find optimal tree/forest efficiently (Chow-Liu) ☺
- Want practical algorithm for learning structure of more general graphs..

Local search algorithms

- Start with empty graph (better: Chow-Liu tree)
- Iteratively modify graph by
  - Edge addition
  - Edge removal
  - Edge reversal
- Need to guarantee acyclicity (can be checked efficiently)
- Be careful with I-equivalence (can search over equivalence classes directly!)
- May want to use simulated annealing to avoid local maxima

Efficient local search



- Want to avoid recomputing the score after each modification!

Score decomposability

- Proposition: Suppose we have
  - **Parameter independence**
  - **Parameter modularity**: if X has same parents in G, G', then same prior.
  - **Structure modularity**: P(G) is product over factors defined over families (e.g.: $P(G) = \exp(-c|G|)$)
- Then Score(D : G) **decomposes** over the graph:

$$\text{Score}(G \,;\, D) = \sum_i \text{FamScore}(X_i \mid Pa_i;\, D)$$

- If G' results from G by modifying a single edge, only need to recompute the score of the affected families!!

What you need to know

- Conjugate priors
  - Beta / Dirichlet
  - Predictions, updating of hyperparameters
- Meta-BN encoding parameters as variables
- Choice of hyperparameters
  - BDe prior
- Decomposability of scores and implications
- Local search

Tasks

- Read Koller & Friedman Chapter 17.4, 18.3-5
- Project proposal due Monday Oct 19 (contact TAs or instructor to discuss ideas)
- Homework 1 due Wednesday Oct 21