

Bayesian learning: Bayes' rule

- Given some model space (set of hypotheses h_i) and evidence (data D):
 - $P(h_i|D) = \alpha P(D|h_i) P(h_i)$
- We assume that observations are independent of each other, given a model (hypothesis), so:
 - $P(h_i|D) = \alpha \prod_j P(d_j|h_i) P(h_i)$
- To predict the value of some unknown quantity, X (e.g., the class label for a future observation):

$$- P(X|D) = \sum_i P(X|D, h_i) P(h_i|D) = \sum_i P(X|h_i) P(h_i|D)$$

These are equal by our independence assumption

Bayesian learning

- We can apply Bayesian learning in three basic ways:
 - **BMA (Bayesian Model Averaging):** Don't just choose one hypothesis; instead, make predictions based on the weighted average of all hypotheses (or some set of best hypotheses)
 - **MAP (Maximum A Posteriori) hypothesis:** Choose the hypothesis with the highest *a posteriori* probability, given the data
 - **MLE (Maximum Likelihood Estimate):** Assume that all hypotheses are equally likely *a priori*; then the best hypothesis is just the one that maximizes the likelihood (i.e., the probability of the data given the hypothesis)
- **MDL (Minimum Description Length) principle:** Use some encoding to model the complexity of the hypothesis, and the fit of the data to the hypothesis, then minimize the overall description of $h_i + D$

Parameter estimation

- Assume known structure
- Goal: estimate BN parameters θ
 - entries in local probability models, $P(X \mid \text{Parents}(X))$
- A parameterization θ is good if it is likely to generate the observed data:

$$L(\theta : D) = P(D \mid \theta) = \prod_m P(x[m] \mid \theta)$$

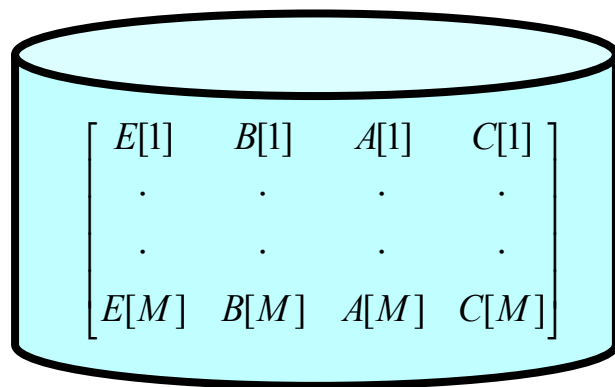


i.i.d. samples

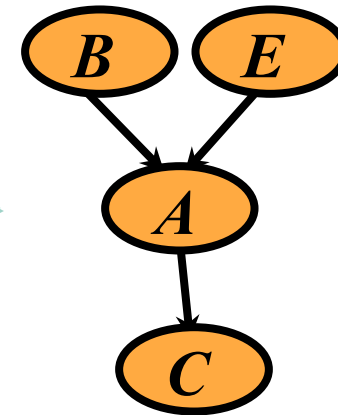
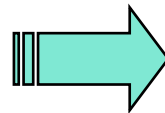
- Maximum Likelihood Estimation (MLE) Principle: Choose θ^* so as to maximize L

Learning Bayesian networks

- Given training set $D = \{\mathbf{x}[1], \dots, \mathbf{x}[M]\}$
- Find B that best matches D
 - model selection
 - parameter estimation



Data D



Model selection

Goal: Select the best network structure, given the data

Input:

- Training data
- Scoring function

Output:

- A network that maximizes the score

Structure selection: Scoring

- Bayesian: prior over parameters and structure
 - get balance between model complexity and fit to data as a byproduct

Marginal likelihood

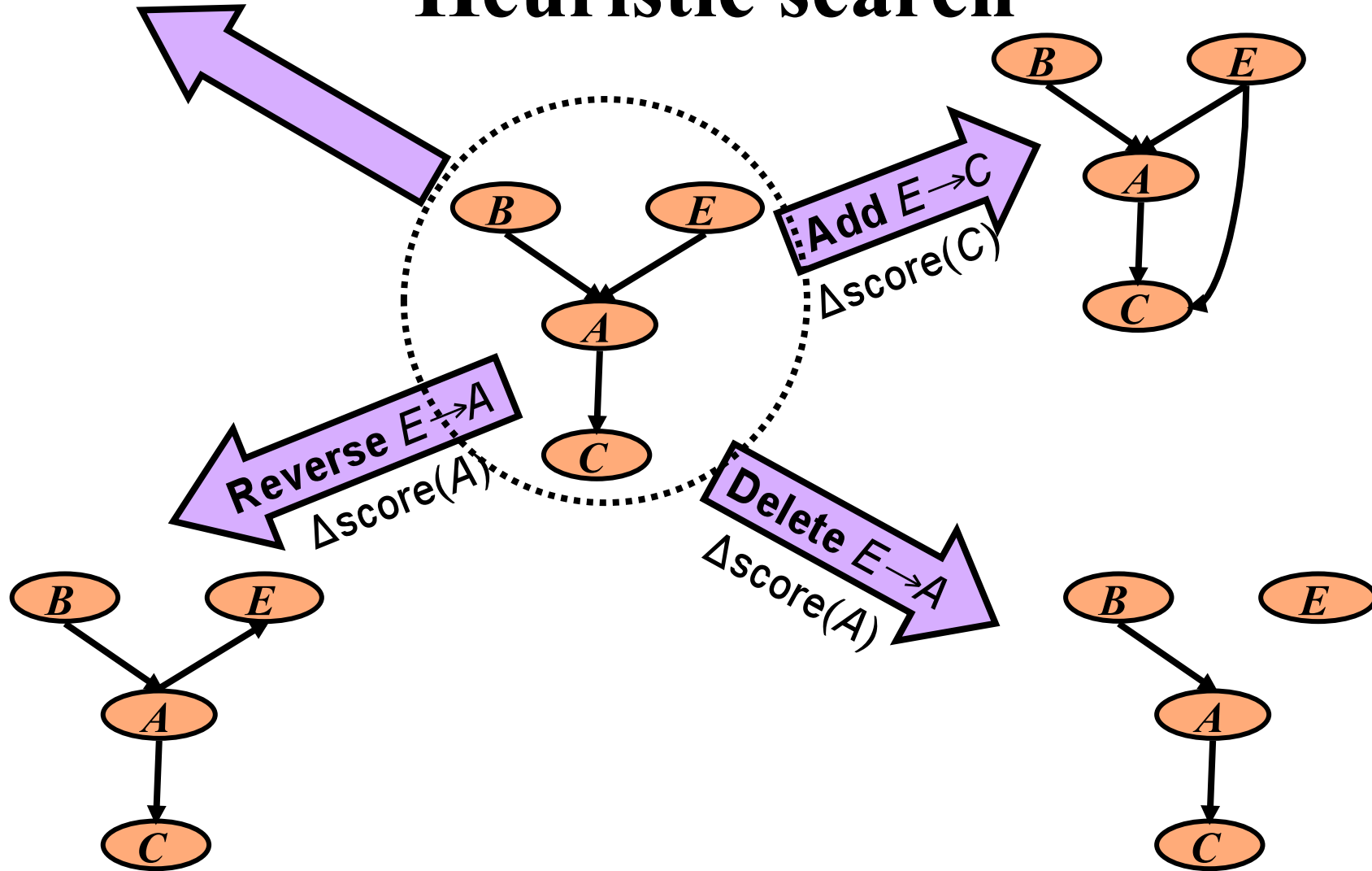
Prior

- Score (G:D) = $\log P(G|D) \propto \log [P(D|G) P(G)]$
- Marginal likelihood just comes from our parameter estimates
- Prior on structure can be any measure we want; typically a function of the network complexity

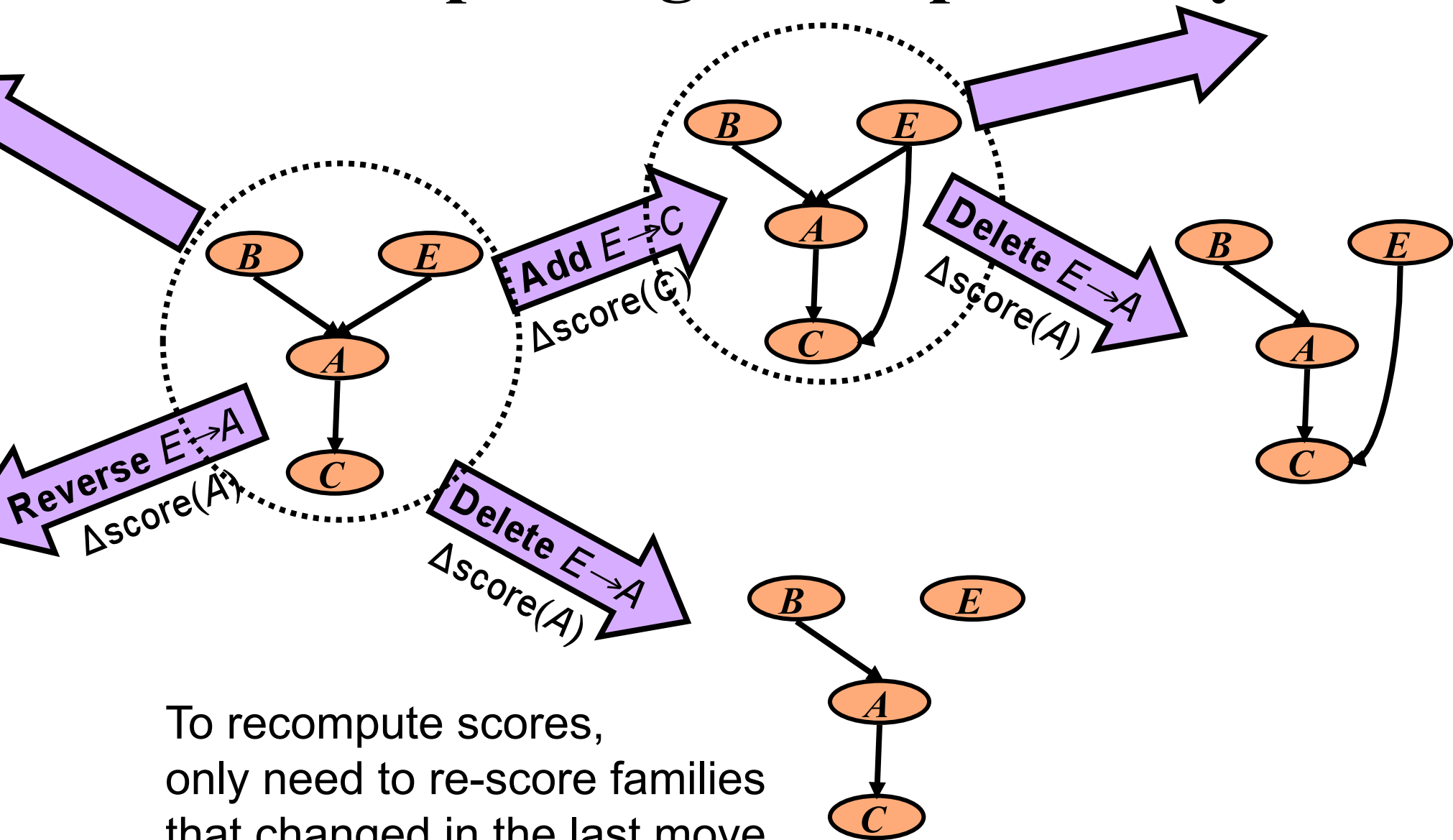
Same key property: Decomposability

$$\text{Score}(\text{structure}) = \sum_i \text{Score}(\text{family of } X_i)$$

Heuristic search



Exploiting decomposability



To recompute scores, only need to re-score families that changed in the last move

Variations on a theme

- **Known structure, fully observable:** only need to do parameter estimation
- **Known structure, hidden variables:**
use expectation maximization (EM) to estimate parameters
- **Unknown structure, fully observable:** do heuristic search through structure space, then parameter estimation
- **Unknown structure, hidden variables:** too hard to solve!