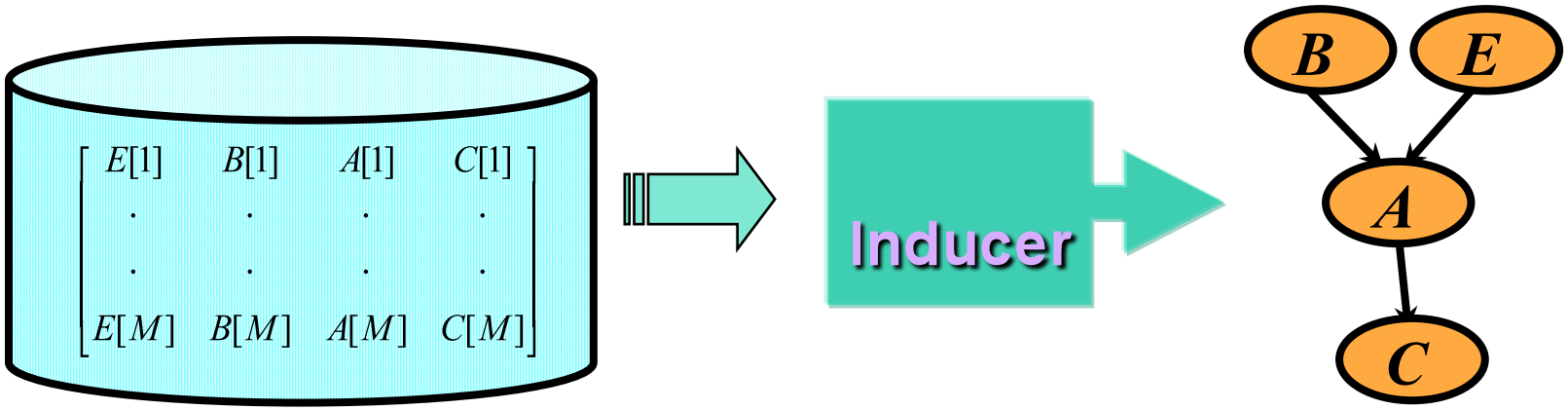
Learning Bayesian networks

- Given training set $\boldsymbol{D} = \{\boldsymbol{x}[1],\ldots,\boldsymbol{x}[\boldsymbol{M}]\}$
- Find Model that best matches $\boldsymbol{D}$
  - 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

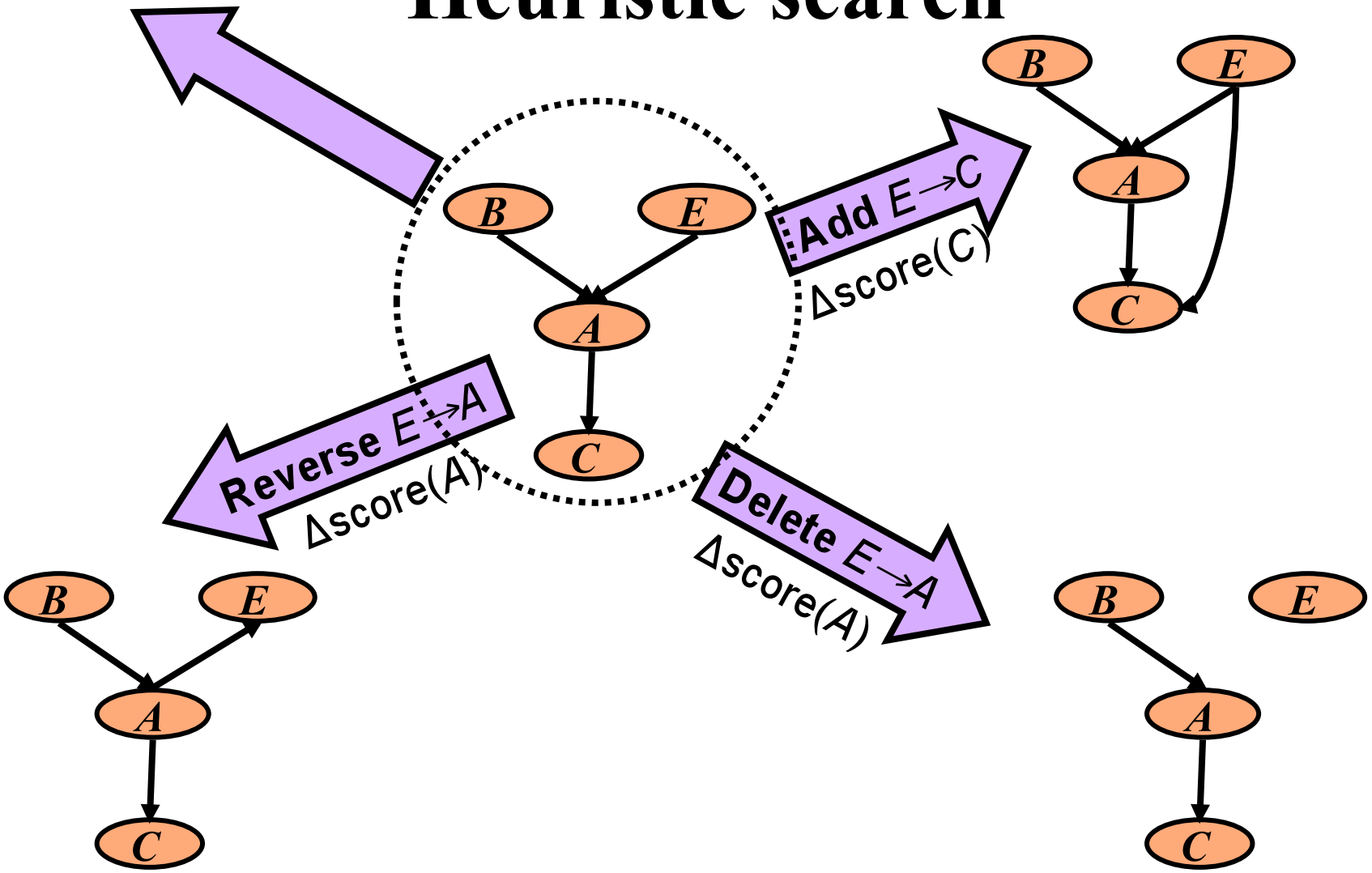
Marginal likelihood ***Prior***

- Score (G:D) = log P(G|D) α 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

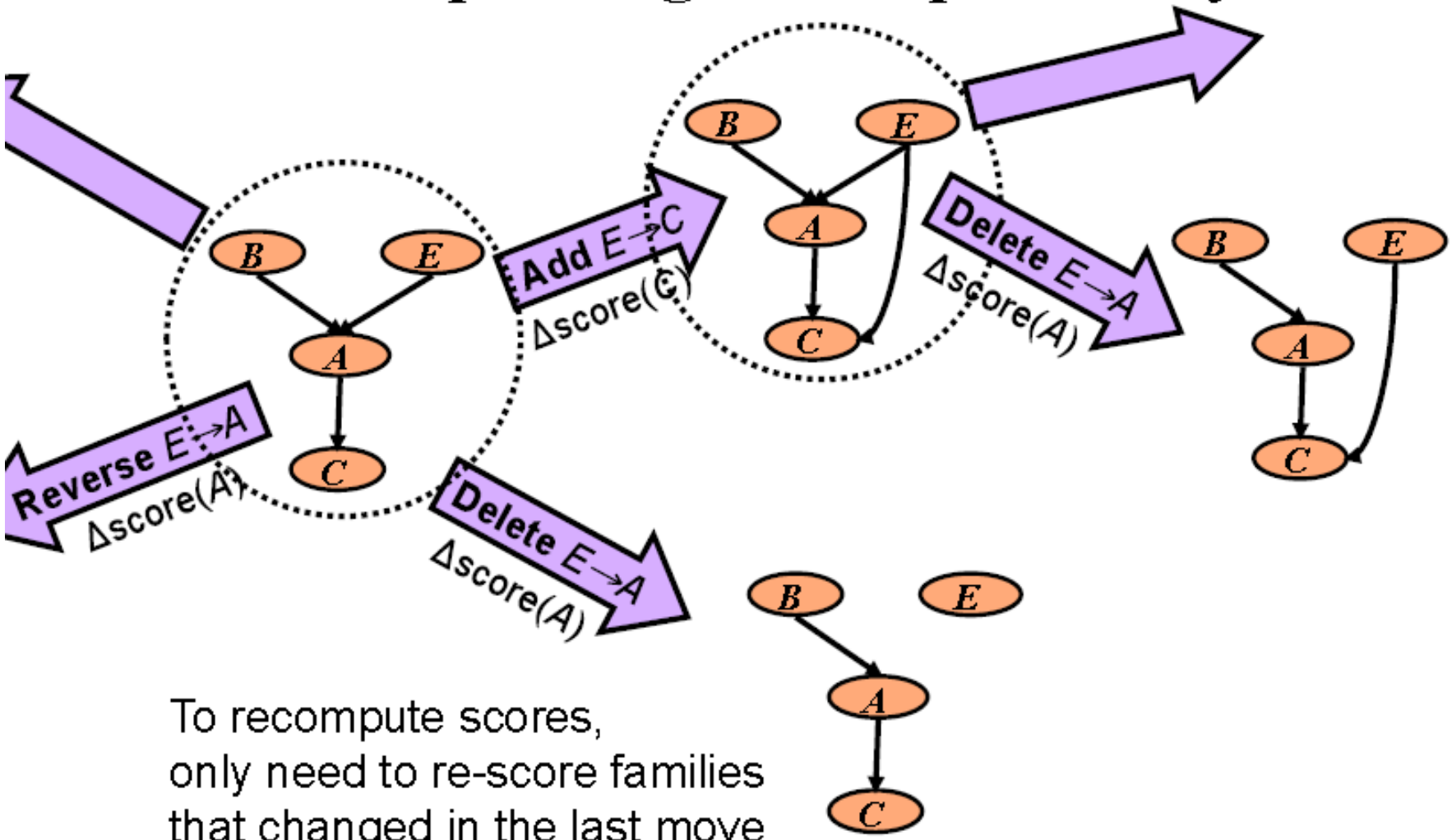
Same key property: Decomposability

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

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!