Northeastern

CS 7140: ADVANCED MACHINE LEARNING

Maximum Likelihood Estimation

- Finding a hypothesis that fits the data well
  $\theta^\star = \text{argmax}_\theta \log P(D \mid \theta, H)$
- Work with the *logarithm* of the likelihood
  - products of probabilities tends too be small
  - likelihood multiples, log likelihood adds
- MLE is equivalent to minimize the relative entropy
  $KL(P(x \mid \theta^\star) \mid\mid P(x \mid \theta)) = \mathbb{E}[\log P(x \mid \theta^\star)] - H[P(x \mid \theta^\star)]$

Maximum a Posterior (MAP)

- $P(\theta|D) \propto P(D|\theta)P(\theta)$

  posterior likelihood prior

- Conjugate distributions

  $P(\theta|D) \propto P(D|\theta)P(\theta)$

  same distribution family

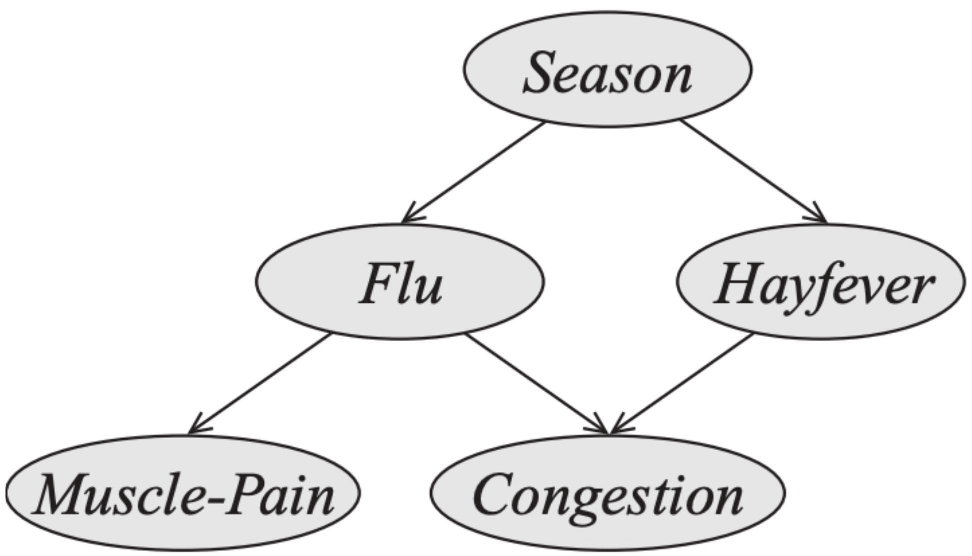
- Example: Dirichlet prior is conjugate to multinomial

  if $P(\theta) = \text{Dir}(\alpha_1, \cdots, \alpha_K)$ then

  $P(\theta|D) = \text{Dir}(M_1 + \alpha_1, \cdots, M_K + \alpha_K)$

STRUCTURE LEARNING

Structure Learning

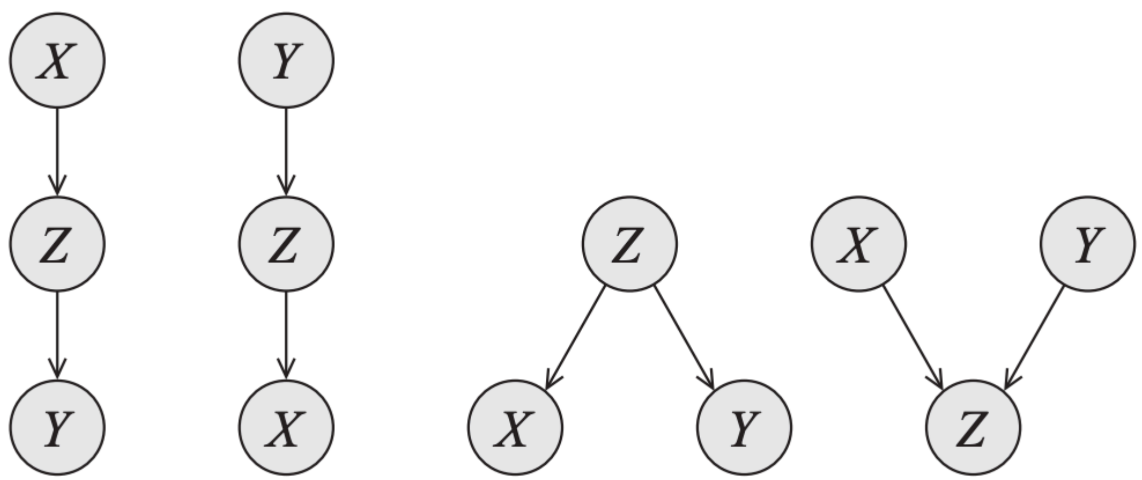


- Reconstruct the structure of Bayesian networks $G^\star$
- Find independencies in variables
- $G^\star$ is **not** identifiable: the class of all I-equivalent networks

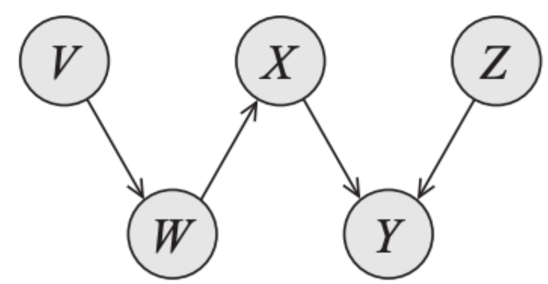
I-Equivalence

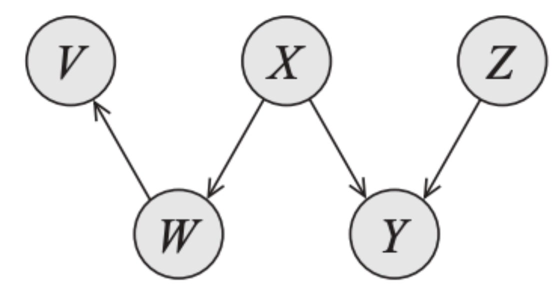
- Two Bayesian networks are I-equivalent if they encode precisely the same conditional independence assertions.

- Are these I-equivalent?



(a) (b) (c) (d)





Two Bayesian networks are I-equivalent if they have the same **skeletons** and same **V-structures**.

Methods Overview

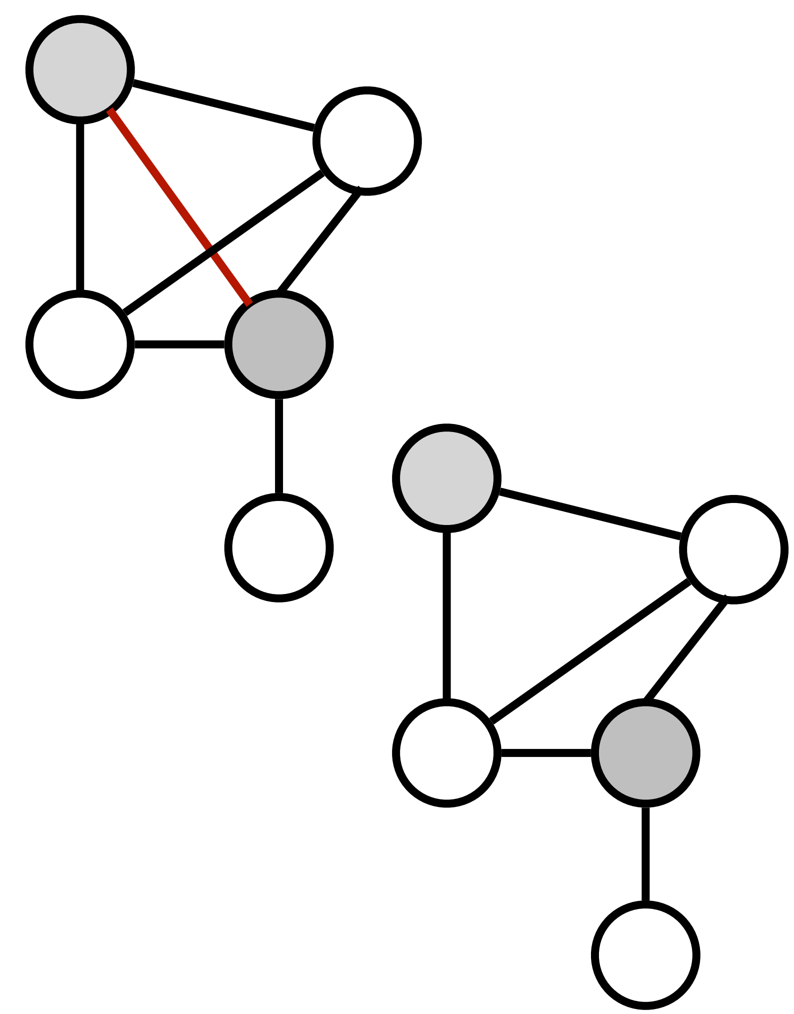
- **Constraint-based** : reconstruct a network structure that best captures the independencies
- **Score-based**: find a network structure that best fits the observed data
- **Bayesian model averaging**: average the prediction of all possible structures.

Constraint-Based Structure Learning

Learn an I-Equivalent class and find minimal I-Map

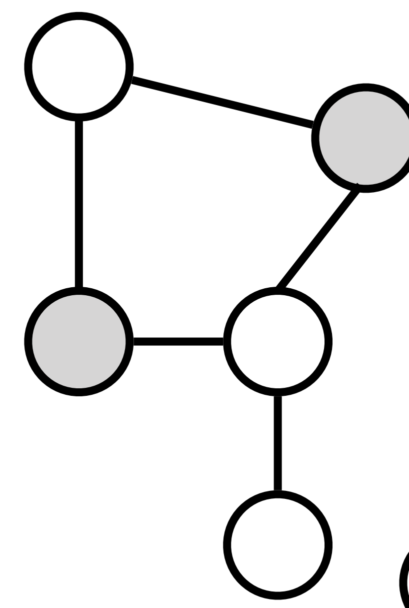
Algorithm input $X_1, \cdots, X_n$, output $G^\star$

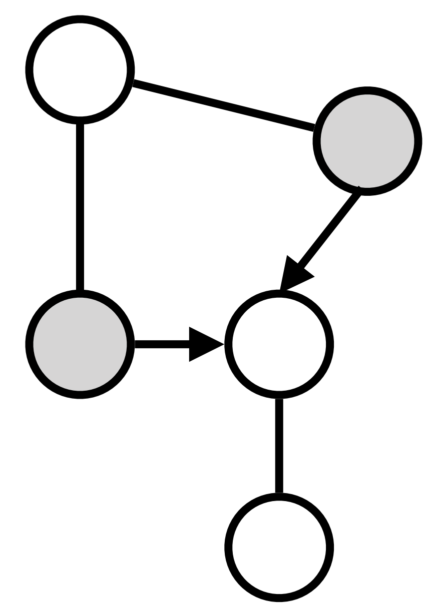
1. Build a complete graph for $X_1, \cdots, X_n$
2. For every pair X_i, X_j, find the witness set U such that $X_i \perp X_j \mid U$
3. Remove edge $X_i - X_j$ if U is not empty
4. Identify triplets that are *immoralities*
5. Orient edges by propagating a set of *constraints*



Identify Immoralities (V-Structures)

- for every triplets that satisfies $X_i - X_j - X_k$ but $X_i \neg X_k$
- Find the witness set of X_i, X_k such that $X_i \perp X_k \mid U_{X_i,X_k}$
- If $X_j \notin U_{X_i,X_k'}$, add $X_i \rightarrow X_j$ and $X_j \leftarrow X_k$

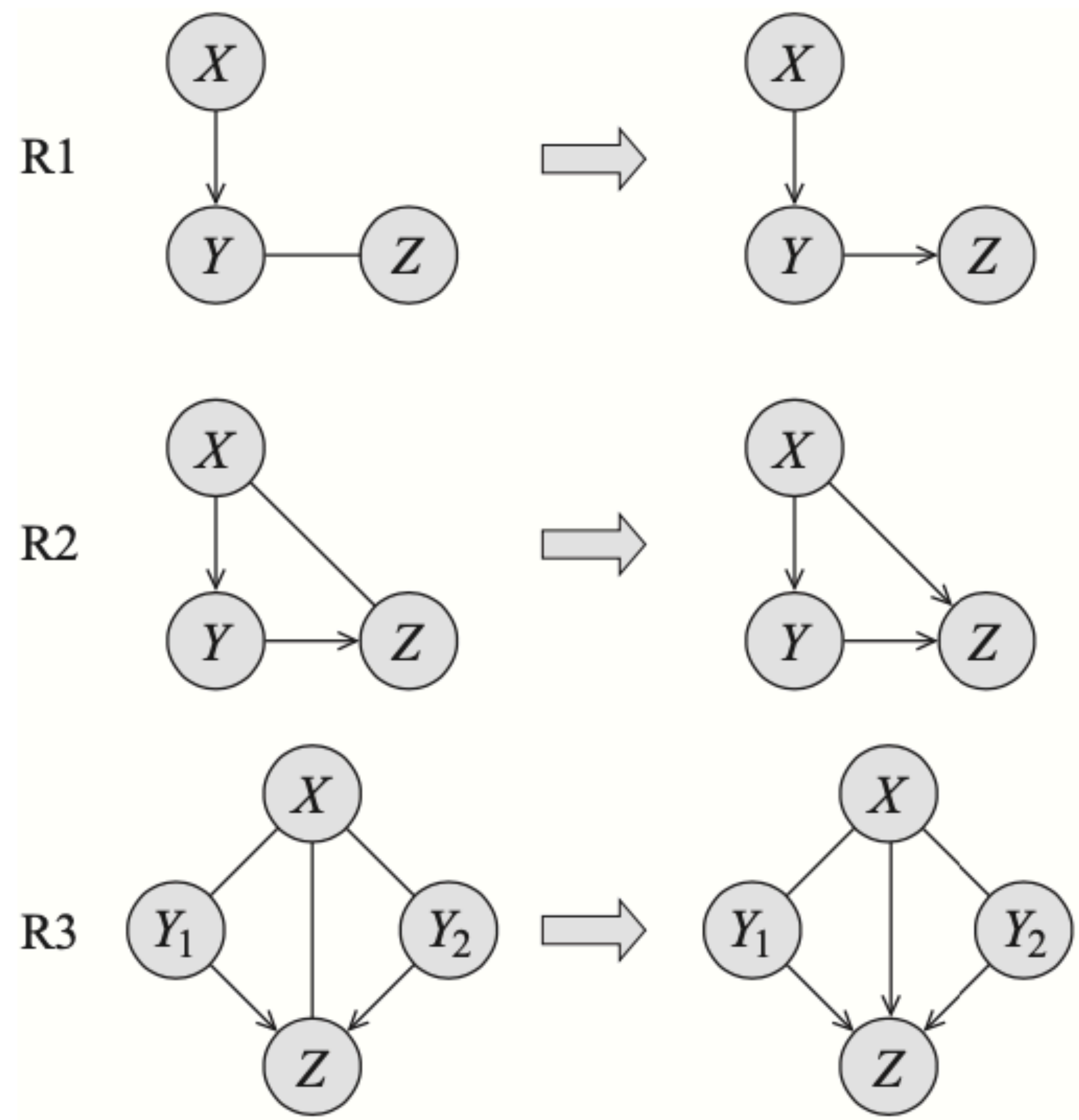




Independence Test hypothesis testing $P(X_1, X_2) = P(X_1)P(X_2)$

Rules for Orienting Edges

- Preserve the V-structures
- Acyclic graph
- Allow undirected edges to have both directions



Guaranteed Recovery

The algorithm reconstructs the network with *polynomial* number of *independence tests*

Assumptions

- The network $G^\star$ has bounded indegree d
- The independence procedure can perfectly answer any independence query that involves up to $2d+2$ variables.
- The underlying distribution $P^\star$ is faithful to $G^\star$

Score-Based Structure Learning

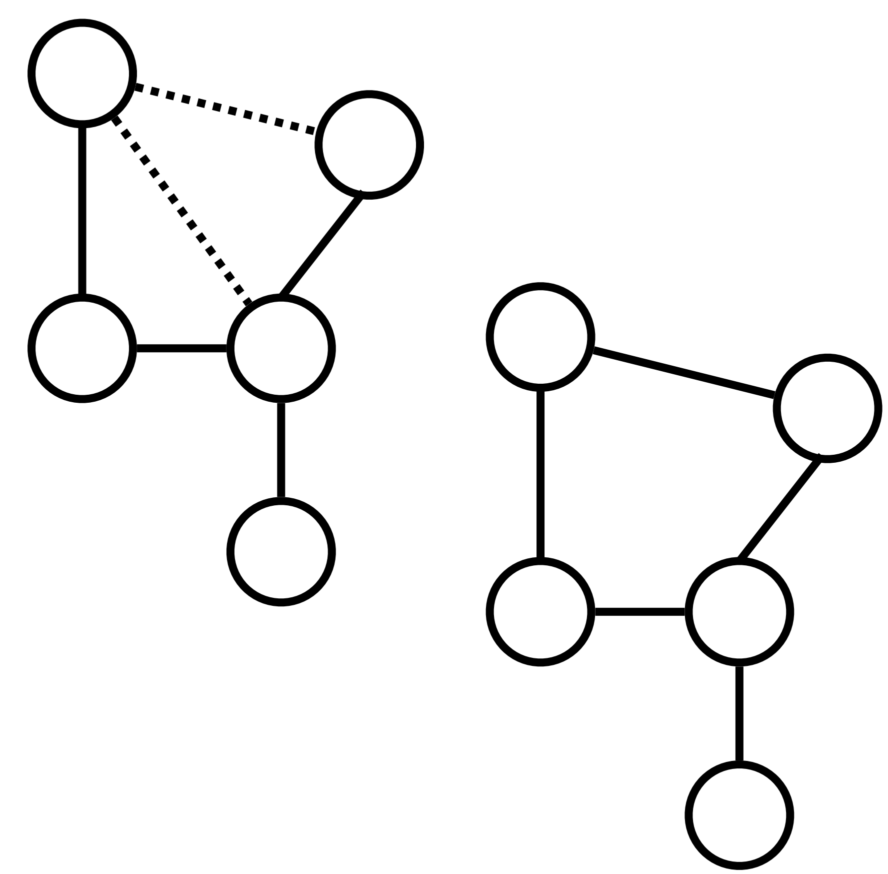
Optimize the network structure score that best fits the data

Algorithm input $X_1, \cdots, X_n$, output $G^\star$, a scoring function

1. Generate a set of possible network structures with bounded indegree

2. Search the space of graphs

3. Return the high-scoring one

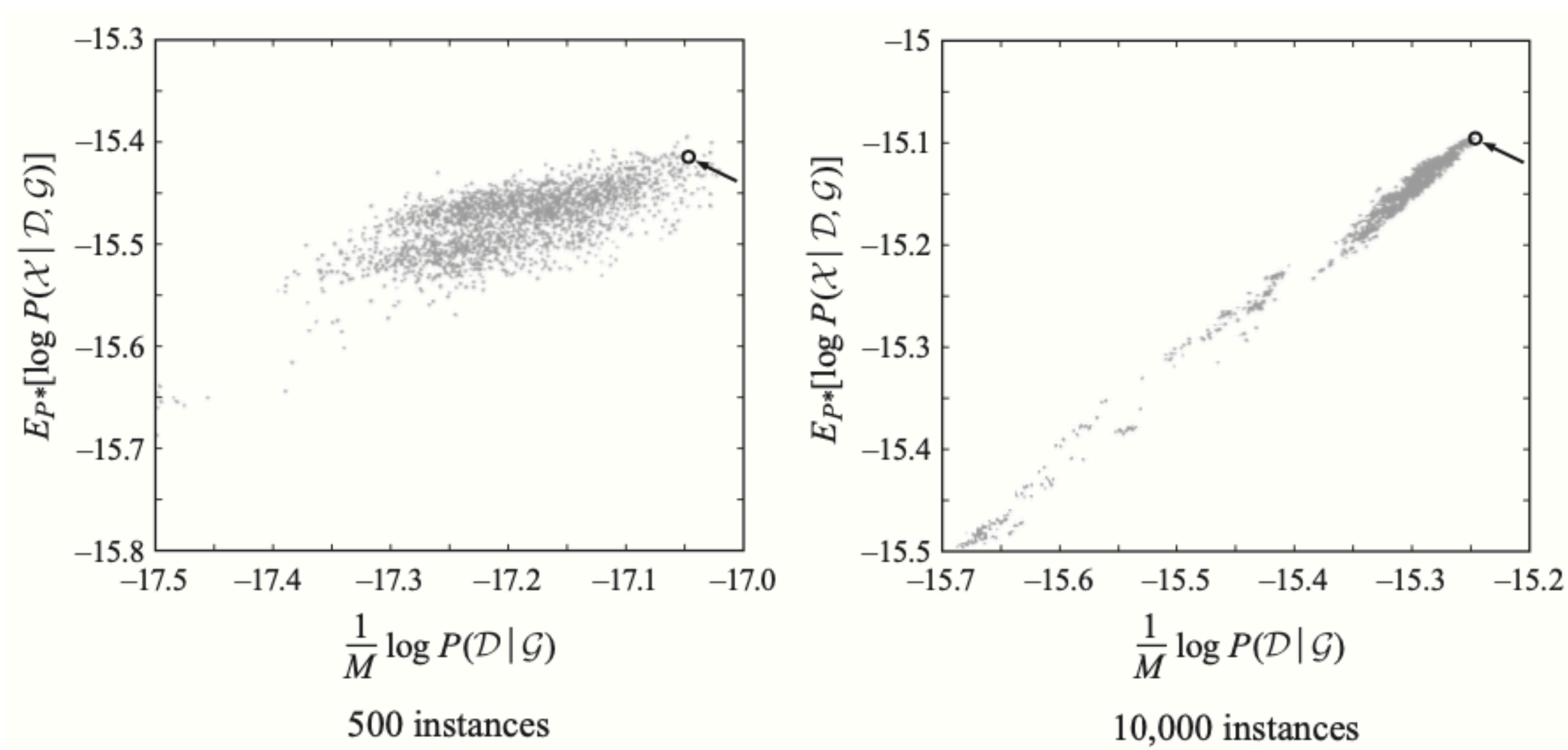


Maximum Likelihood Score

Find the model that has the highest likelihood

$$\max_{G} \max_{\theta_G} L(D \mid G, \theta_G)$$

- Never prefers simpler network
- Add an edge will never decrease the score, overfit the training data



Bayesian Score

Place a prior whenever we have uncertainty

$$P(G|D)$$

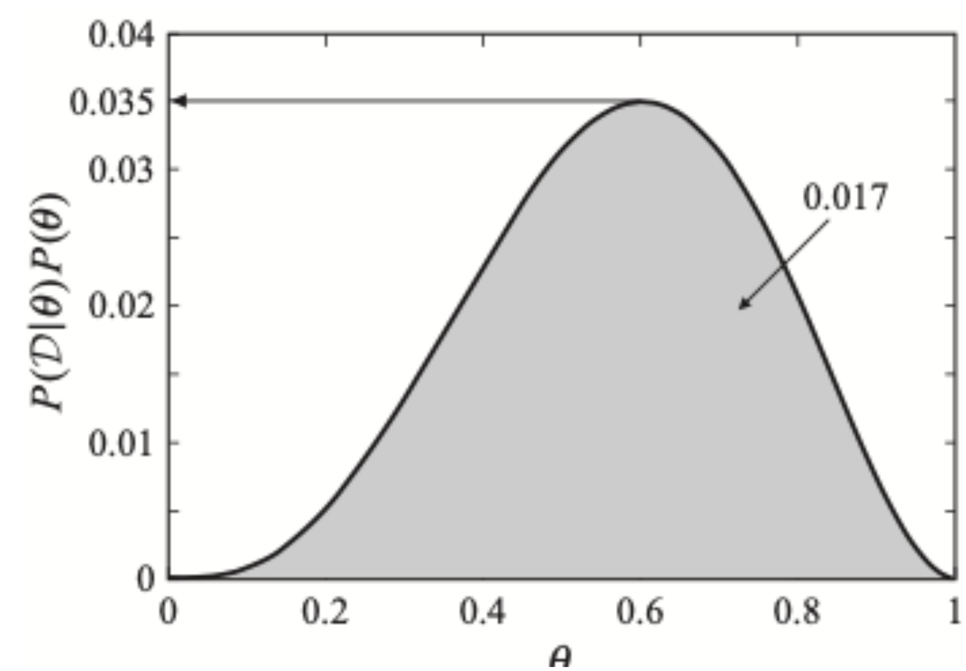
- By Bayesian rule: $P(G|D) = \dfrac{P(D|G)P(G)}{P(D)}$

- Score: $\log P(D|G) + \log P(G)$

  $\boxed{\text{marginal likelihood}}$

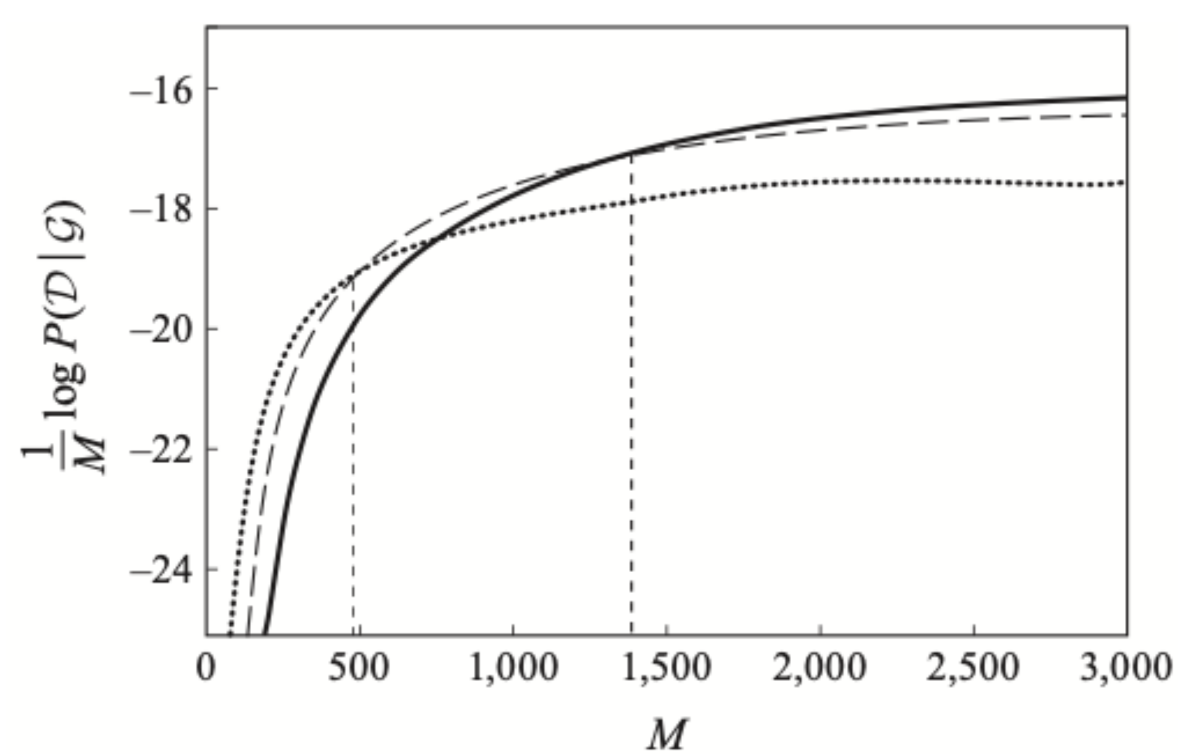
- Intuition:

$$\frac{1}{M}\log P(D|G) \approx \mathbb{E}_P[\log P(X|G,D)]$$



Bayesian Information Criteria

- Biased toward simpler structures, willing to recognize more complex structures with more data.
- Trade off the likelihood — fit to data — and some notion of model complexity , reducing the extent of *overfitting*.
- $\text{score}_{BIC}(G : D) = L(D\,|\,\theta) - \dfrac{\log M}{2}\text{Dim}(G)$

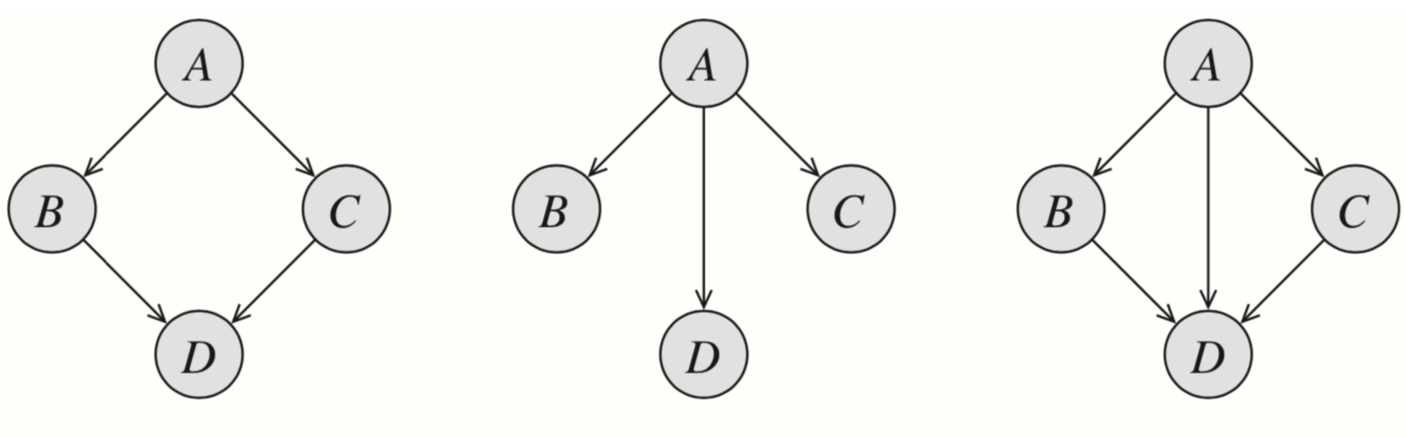


Score Decomposition

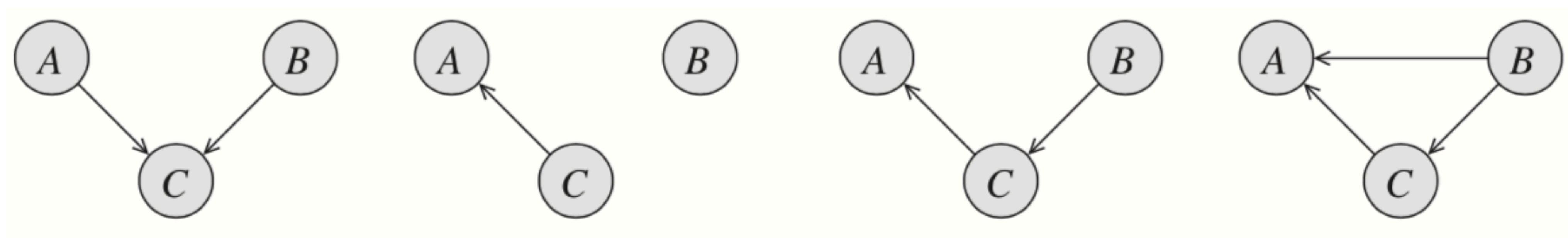
- Specify prior over structure $P(G) \propto \prod_i P(\mathrm{Pa}_{X_i} = \mathrm{Pa}_{X_i}^G)$
- Decomposable score: $\mathrm{score}(G : D) = \sum_i \mathrm{score}(X_i \,|\, \mathrm{Pa}_{X_i}^G : D)$
- A local change in the structure (such as adding an edge) does not change the score of other parts of the structure that remained the same.
- Reduce dramatically the computational overhead of evaluating different structures during search.

Structure Search

- **State**: a network structure of variables
- **Action**: edge addition, edge deletion, edge reversal



(a) (b) (c)



(a) (b) (c) (d)

Bayesian Model Averaging

Compute the average prediction over candidate network structures

$$\mathbb{E}_{P(G|D)}[f(G)] = \sum_G f(G)P(G|D)$$

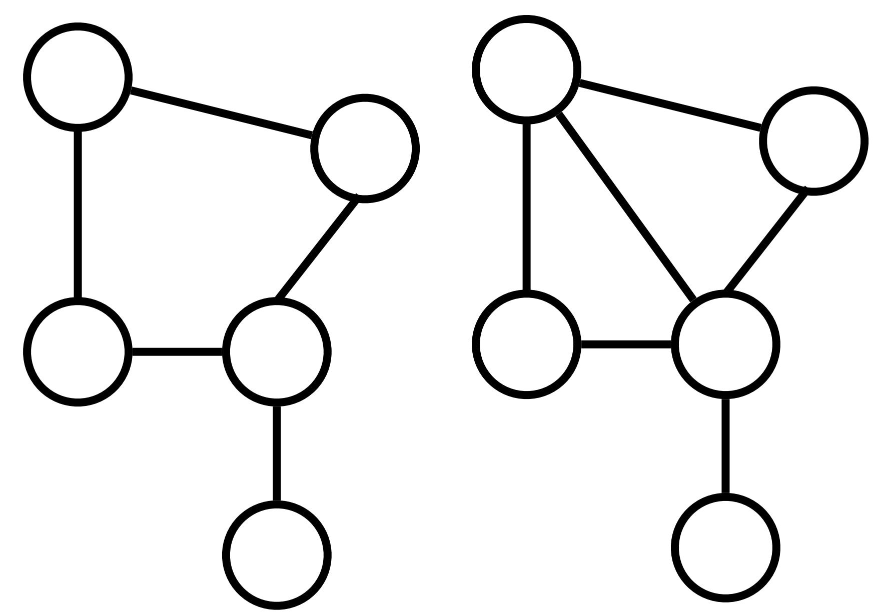
Algorithm input $X_1, \cdots, X_n$, output $G^\star$, a prediction function

1. Find a set $\mathcal{G}'$ of high scoring structures

2. Estimate the probability of the structure in $\mathcal{G}'$

3. Compute average prediction

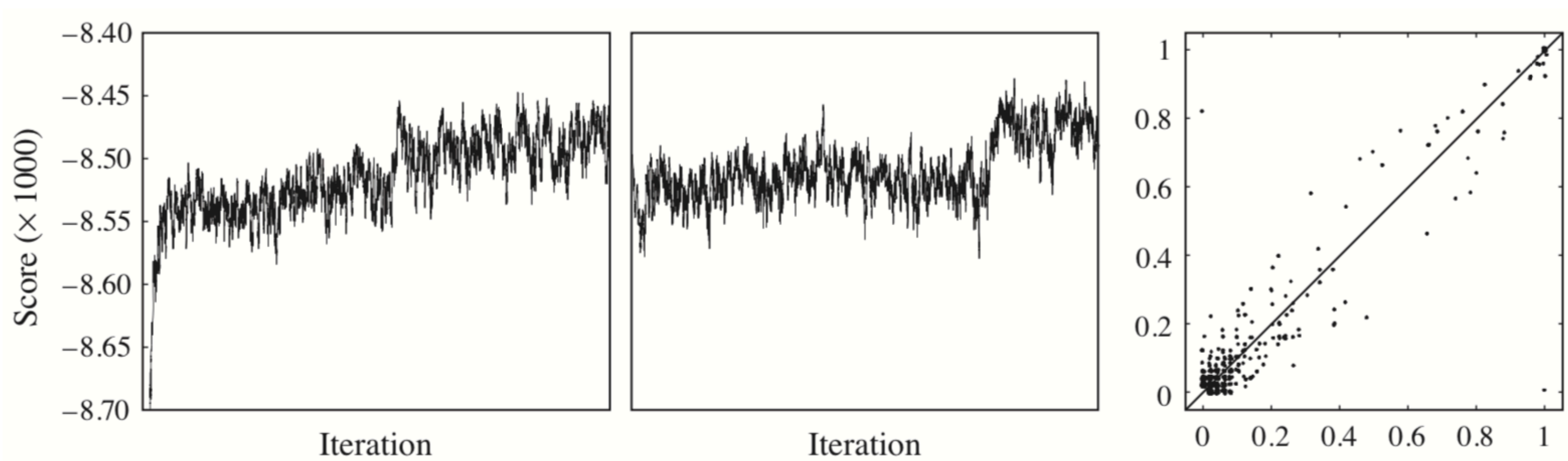
$$P(f|D) \approx \frac{\sum_{G\in\mathcal{G}'} P(G|D)f(G)}{\sum_{G\in\mathcal{G}'} P(G|D)}$$



$f(G_1) + f(G_2)$

Find Candidate Structures

- Large data: a single high-scoring structure gives a good approximation of the prediction
- Small data: a large number of high-scoring models
- Alternative: MCMC over structures



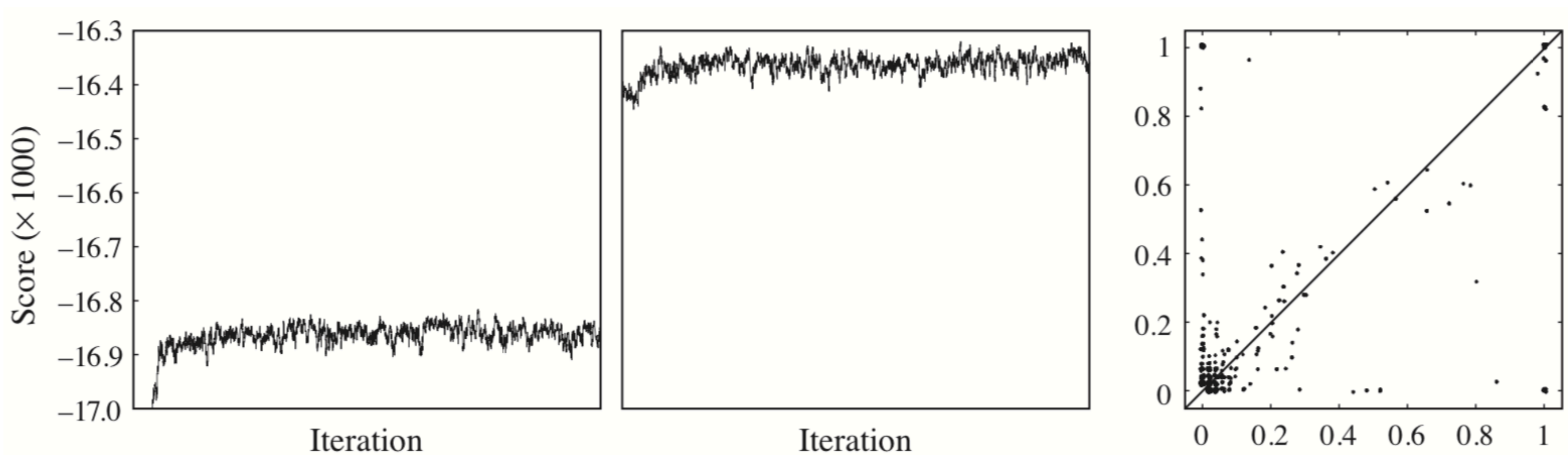
(a) (b) (c)

MCMC over Structures

- Use Metropolis-Hasting as an example
- The current state is G, sample G' from the proposal distribution
- Accept the transition with probability

uniform distribution

$$\min\left[1, \frac{P(G', D)T^Q(G' \to G)}{P(G, D)T^Q(G \to G')}\right]$$



(a) (b) (c)