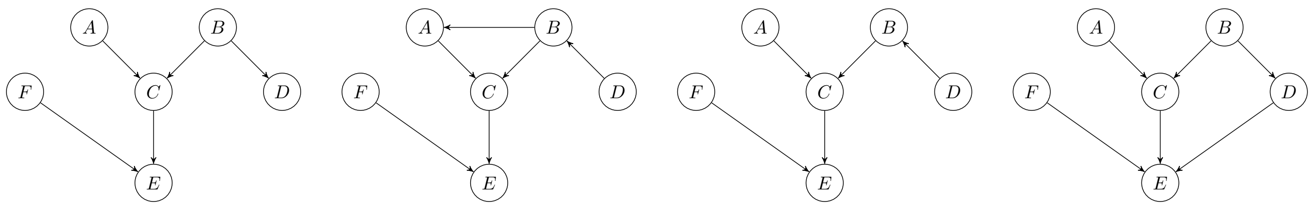


(a) Graph 1 (b) Graph 2 (c) Graph 3 (d) Graph 4

Figure 1: Different BNs

1 Structure scores

Exercise 1.1 *Likelihood score. Consider the graphs of Figure 1. Which statement about the likelihood scores of the different graphs is true? You may select 1 or more options, or none of them.*

a) $\mathit{Score}_L(G_1; D) = \mathit{Score}_L(G_3; D)$, *for every dataset* D

b) $\mathit{Score}_L(G_1; D) \geq \mathit{Score}_L(G_4; D)$, *for every dataset* D

c) $\mathit{Score}_L(G_2; D) \geq \mathit{Score}_L(G_3; D)$, *for every dataset* D

d) $\mathit{Score}_L(G_4; D) \geq \mathit{Score}_L(G_2; D)$, *for every dataset* D

Exercise 1.2 *BIC score. Consider the same graphs of Figure 1, but now think about the BIC score. Which statement is true? You may select 1 or more options, or none of them.*

a) $\mathit{Score}_{BIC}(G_1; D) = \mathit{Score}_{BIC}(G_3; D)$, *for every dataset* D

b) $\mathit{Score}_{BIC}(G_1; D) \geq \mathit{Score}_{BIC}(G_4; D)$, *for every dataset* D

c) $\mathit{Score}_{BIC}(G_2; D) \neq \mathit{Score}_{BIC}(G_3; D)$, *for every dataset* D

d) $\mathit{Score}_{BIC}(G_1; D) \geq \mathit{Score}_{BIC}(G_2; D)$, *for every dataset* D

Exercise 1.3 *Hidden variables. Consider the case where the generating distribution is a naive Bayes model, with an unobserved class variable* C *and its binary-valued children* $\{X_1, \ldots, X_{100}\}$. *Assume that* C *is strongly correlated with each of its children (that is, distinct classes are associated with fairly different distributions over each* X_i*). Now suppose we try to learn a network structure directly on* $\{X_1, \ldots, X_{100}\}$, *without including* C *in the network. What network structure are we likely to learn if we have* $10,000$ *data instances, and we are using table CPDs with the likelihood score as the structure learning criterion?*

a) The empty network, i.e., a network consisting of only the variables but no edges between them.

b) Some connected network over $\{X_1, \ldots, X_{100}\}$ *which is not fully connected nor empty.*

c) A fully connected network, i.e., one with an edge between every pair of nodes.

Exercise 1.4 *Hidden variables. Using the same setup of the previous exercise, now suppose that we use the BIC score. Which network structure are we likely to learn?*

a) The empty network, i.e., a network consisting of only the variables but no edges between them.

b) Some connected network over $\{X_1, \ldots, X_{100}\}$ *which is not fully connected nor empty.*

c) A fully connected network, i.e., one with an edge between every pair of nodes.

2 Tree learning

Exercise 2.1 *Tree construction. We want to build a tree-structured network (any node has at most a single parent). We have a structure score that satisfies score decomposability and score equivalence. Which graph algorithm would you use to find the optimal tree-structured network?*

a) Finding an undirected spanning forest with the largest diameter (i.e., the longest distance between any pair of nodes).

b) Finding a directed spanning forest with the largest diameter (i.e., the longest distance between any pair of nodes).

c) Finding the maximum-weight undirected spanning forest (i.e., a set of undirected edges such that there is at most one path between any pair of nodes).

d) Finding the maximum flow through the graph, using any pair of observed nodes as source and sink.

Exercise 2.2 *Recovering directionality. Following the previous example, if after the optimization we find an undirected spanning forest, which is the most efficient way to recover to transform it into a directed spanning forest?*

a) Evaluate all possible directions for the edges by iterating over them. This takes $O(2^n)$ time, since there are at most 2^n possible sets of edge directions in the spanning forest.

b) Evaluate all possible directions for the edges. While there are at most 2^n possible sets of edge directions, we can exploit score decomposability to find the best directed spanning forest in $O(n)$ time.

c) Pick any arbitrary direction for each edge, which takes $O(n)$ time. Because of score equivalence, all possible directed versions of the optimal undirected spanning forest have the same score, so this is valid.

d) Pick any arbitrary root, and direct all edges away from it. This takes $O(n)$ time.

Exercise 2.3 *Trees vs. forests. If the algorithm we used to learn tree-structured networks, rather than a single tree, can produce a forest (a set of two or more disconnected trees), why don't use them? Assume that we use the likelihood score, and that the algorithm breaks ties between equal-scoring trees arbitrarily. Which of the following is true?*

a) This algorithm will never produce a forest, since there will always be a tree that has strictly higher score.

b) It's theoretically possible for the algorithm to produce a forest. However, this will only occur in very contrived and unrealistic circumstances, not in practice.

c) It's possible for the algorithm to produce a forest, since there are cases in which a forest will have a higher score than any tree.

d) It's possible for the algorithm to produce a forest even though trees will always score more highly, since the algorithm need not find the structure that is globally optimal (relative to the likelihood score).

Exercise 2.4 *Augmenting Trees. Following the previous exercises, we now want to learn a hybrid naive-Bayes/tree-structured network (Tree-augmented naive Bayes), where we have a single class variable C as well as the variables $\{X_1, \ldots, X_n\}$, and each X_i has C as a parent as well as there is also a tree connecting the X_i's (each X_i may have at most 2 parents, C and other X_j).*

Which is the correct edge weight to use for $j \to i$ in the spanning tree finding algorithm if we want to optimize the likelihood score?

a) $N \cdot \mathrm{MI}_{\hat{p}}(X_i; X_j, C)$, where N is the dataset size, and $\mathrm{MI}_{\hat{p}}(\boldsymbol{A}, \boldsymbol{B})$ is the mutual information in the empirical distribution $\hat{p}$ of variables $\boldsymbol{A}$ with variables $\boldsymbol{B}$.

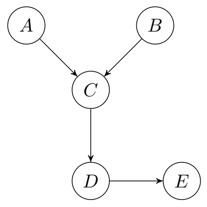
b) $N \cdot \mathrm{MI}_{\hat{p}}(X_i; X_j, C) - \mathrm{H}_{\hat{p}}(X_i)$

c) $N \cdot (\mathrm{MI}_{\hat{p}}(X_i; X_j, C) - \mathrm{MI}_{\hat{p}}(X_i; C)) - \mathrm{H}_{\hat{p}}(X_i)$

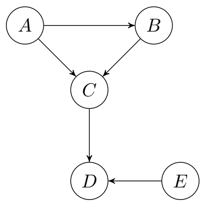
d) $N \cdot (\mathrm{MI}_{\hat{p}}(X_i; X_j, C) - \mathrm{MI}_{\hat{p}}(X_i; C))$

e) $N \cdot (\mathrm{MI}_{\hat{p}}(X_i; X_j, C) - \mathrm{MI}_{\hat{p}}(X_i; X_j)) - \mathrm{H}_{\hat{p}}(X_i)$

f) $N \cdot (\mathrm{MI}_{\hat{p}}(X_i; X_j, C) - \mathrm{MI}_{\hat{p}}(X_i; X_j))$



(a) Graph 1



(b) Graph 2

Figure 2: Different BNs (II)

3 Hill climbing

Exercise 3.1 *Calculating Likelihood Differences. While doing a hill-climbing search, you run into the two graphs of Figure 2, and need to choose between them using the likelihood score. What is the difference in likelihood scores,* $Score_L(G_1; D) - Score_L(G_2; D)$*, given a dataset* D *of size* N*?*

a) $N \cdot [\mathrm{MI}_{\hat{p}}(C; A, B) + \mathrm{MI}_{\hat{p}}(D; C) + \mathrm{MI}_{\hat{p}}(E; D) - \mathrm{MI}_{\hat{p}}(B; A) - \mathrm{MI}_{\hat{p}}(D; C, E)]$ *where* N *is the dataset size,* $\mathrm{MI}_{\hat{p}}(\boldsymbol{A}, \boldsymbol{B})$ *is the mutual information in the empirical distribution* $\hat{p}$ *of variables* $\boldsymbol{A}$ *with variables* $\boldsymbol{B}$*, and* $\mathrm{H}_{\hat{p}}(\boldsymbol{A})$ *is the empirical entropy in the empirical distribution* $\hat{p}$ *of variables* $\boldsymbol{A}$*.*

b) $N \cdot [\mathrm{MI}_{\hat{p}}(D; C) + \mathrm{MI}_{\hat{p}}(E; D) - \mathrm{MI}_{\hat{p}}(A; B) - \mathrm{MI}_{\hat{p}}(D; C, E) - \mathrm{H}_{\hat{p}}(A, B, C, D, E)]$

c) $N \cdot [\mathrm{MI}_{\hat{p}}(D; C) + \mathrm{MI}_{\hat{p}}(E; D) - \mathrm{MI}_{\hat{p}}(B; A) - \mathrm{MI}_{\hat{p}}(D; C, E)]$,

d) $N \cdot [\mathrm{MI}_{\hat{p}}(A; B) - \mathrm{H}_{\hat{p}}(A, B)]$

e) $N \cdot \mathrm{MI}_{\hat{p}}(A; B)$

Exercise 3.2 *Optimality of Hill Climbing. You use Hill Climbing to improve your graph structure, by considering edge deletions, reversals, and additions at each step, until no change gives you a higher-scoring structure. Are you guaranteed that the last graph is the best graph structure? Assume that the training dataset is sufficiently large.*

a) *Yes, but only if we extend our range of available moves to allow for pairs of edges to be changed simultaneously.*

b) *No - greedy hill-climbing will find only local maxima of the scoring function with respect to our available moves, and we cannot guarantee that it is the true graph structure.*

c) *Yes - greedy hill-climbing provably finds the true graph structure, provided our dataset is large enough.*

d) *No - greedy hill-climbing will only find the true graph structure if we restrict the number of parents for each node to at most 2.*

e) *Yes, but only if we use random restarts and tabu search.*

f) *No - greedy hill-climbing can never find the true graph structure, only local maxima of the scoring function with respect to our available moves.*

Answers

Ex. 1.1: a,c

Ex. 1.2: a

Ex. 1.3: c

Ex. 1.4: b

Ex. 2.1: c

Ex. 2.2: d

Ex. 2.3: b

Ex. 2.4: d

Ex. 3.1: c

Ex. 3.2: b