Tree Learning and Hill Climbing

4/4 得分（100%）

测验通过！

返回第 #3 周课程

1 / 1 分

1。

Detective! You are a detective tracking down a serial robber who has already stolen from 1,000 victims, thereby giving you a large enough training set.

No one else has been able to catch him (or her), but you are certain that there is a method (specifically, a Bayesian network) in this madness.

You decide to model the robber's activity with a **tree-structured network** (meaning that each node has **at most** one parent): this network has (observed) variables such as the location of the previous crime, the gender and occupation of the previous victim, the current day of the week, etc., and a single unobserved variable, which is the location of the next robbery. The aim is to predict the location of the next robbery.

Unfortunately, you have forgotten all of your classical graph algorithms, but fortunately, you have a copy of The Art of Computer Programming (volume 4B) next to you.

Which graph algorithm do you look up to help you find the optimal tree-structured network?

Assume that the structure score we are using satisfies score decomposability and score equivalence.

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

正确

The tree-structured Bayesian network that we eventually want to construct is directed,

However, if we have score equivalence, finding the maximum-weight undirected spanning forest is equivalent to finding the maximum-weight directed spanning forest and is easier to implement.

- ○ Finding the maximum flow through the graph, using any pair of observed nodes as source and sink.
- ○ Finding the maximum-weight bipartite matching between observed nodes and hidden nodes.

---

1 / 1 分

2。

***Recovering Directionality.** Once again, assume that our structure score satisfies score decomposability and score equivalence. After we find the optimal undirected spanning forest (containing n nodes), how can we recover the optimal directed spanning forest (and catch the robber)?

If more than one option is correct, pick the faster option; if the options take the same amount of time, pick the more general option.

- ○ 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.

- ○ 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.

- ○ 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.

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

正确

No matter which root we pick, the resulting trees are in the same I-equivalence class; in fact, there are no valid directed trees that cannot be obtained with this procedure.

Because of score equivalence, it does not matter which root we pick.

---

1 / 1 分

3。

***Augmenting Trees.** It turns out that the tree-structured network we learnt in the preceding questions was not sufficient to apprehend the robber, allowing him to claim his 1001th victim.

Not one to be discouraged, you decide to increase the expressiveness of your network.

Assume that we now want to learn a hybrid naive-Bayes/tree-structured network, where we have a single class variable C as well as the variables $X_1, \ldots, X_n$.

In this model, each X_i has C as a parent, and there is also a tree connecting the X_i 's; that is, each X_i , in addition to C , may also have up to one other parent X_j .

For our baseline network G_0 , we are going to use the naive Bayes network, in which each X_i has only C as a parent.

We are thus aiming to optimize the difference in likelihood scores

$\text{Score}_L(G : \mathcal{D}) - \text{Score}_L(G_0 : \mathcal{D})$, where D is the training dataset.

If we use the appropriate spanning tree algorithm to find the optimal forest structure, what is the correct edge weight to use for $w_{j \to i}$?

In these options, $M = 1001$ is the size of our training dataset, and $I_{\hat{P}}(\mathbf{A}, \mathbf{B})$ is the mutual information in the empirical distribution of the variables in set $\mathbf{A}$ with the variables in set $\mathbf{B}$.

- ○ $M \cdot (I_{\hat{P}}(X_i; X_j, C) - I_{\hat{P}}(X_i; C)) - H_{\hat{P}}(X_i)$

$M \cdot (I_{\hat{P}}(X_i; X_j, C) - I_{\hat{P}}(X_i; C))$

正确
$w_{j \to i} = \mathrm{FamScore}_L(X_i | X_j, C : D) - \mathrm{FamScore}_L(X_i | C : D)$
.

In the case of the likelihood score, this gives us $M(I_{\hat{P}}(X_i; X_j, C) - I_{\hat{P}}(X_i; C))$, since the entropy terms only depend on X_i and cancel each other out.

$M \cdot I_{\hat{P}}(X_i; X_j, C)$

$M \cdot (I_{\hat{P}}(X_i; X_j, C) - I_{\hat{P}}(X_i; X_j)) - H_{\hat{P}}(X_i)$

$M \cdot I_{\hat{P}}(X_i; X_j, C) - H_{\hat{P}}(X_i)$

$M \cdot (I_{\hat{P}}(X_i; X_j, C) - I_{\hat{P}}(X_i; X_j))$

1 / 1 分

4。
Trees vs. Forests. Congratulations! Your hybrid naive-Bayes/tree-structured network managed to correctly predict where the criminal would be next, allowing the police to catch him (or her) before the 1002th victim got robbed.

The grateful populace beg you to return to studying probabilistic graphical models.

While re-watching the video lectures, you begin to wonder if the algorithm we have been using to learn tree-structured networks can produce a forest, rather than a single tree.

Assume that we use the likelihood score, and also assume that the maximum spanning forest algorithm breaks ties (between equal-scoring trees) arbitrarily.

Which of the following is true? In this question, interpret "forest" to mean a set of two or more disconnected trees.

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

- [ ] 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.

- [ ] 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).

- [x] 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.

正确

A forest will be produced only if we can partition the variables into two disjoint sets A and B ,

such that all edges $X_j \to X_i$ with either $X_i \in A, X_j \in B$ or $X_i \in B, X_j \in A$ have weight 0.

This will be the case only if all variables in A are independent of the variables in B in the empirical distribution.

While this is not impossible, it is very unlikely to happen in practice.