

AI exam, 2018. 05. 11.

NAME: Hailey Lynn James TOTAL POINTS:60 (minimum to pass: 24)

NEPTUN ID:

- 1) Suggest an admissible heuristic function for the Steiner tree problem (prove its admissibility).

(15p)

**Heuristic:**

I propose the following heuristic: for each multicast node not currently included in the tree, find the shortest edge connected to this node. Sum the lengths of all of these edges.

**Proving admissibility:**

First, we know that if we have reached the optimal solution, the heuristic function should return zero. This is consistent with my heuristic function, since all of the multicast nodes would be included in the tree, so the sum of shortest edges of multicast nodes not in the tree must be zero.

Second, the heuristic function should never overestimate the cost. We must use at least one edge to add a missing multicast node to the tree, and in the best case scenario, this is the shortest edge connected to the node. Certainly, the optimal solution might not include the shortest edge connected to every multicast node. However, if we don't use this edge, any edge we do use will be longer, so we have effectively constructed a lower bound, and the heuristic function will not overestimate the cost.

- 2) Describe a first-order logic definition that a subgraph of a given graph could be a solution for the Steiner tree problem (i.e., it is the sub-tree connecting all the prespecified nodes, but ignore the minimality of its weight). (15p)

To be a valid solution, for all nodes  $x$  and  $y$ , either they are nodes in the graph and are connected, or they are not members of the multicast set.

$$OneComponent(G) \leftrightarrow \forall x, y : Node(x) \& Node(y) \& Connected(x, y, G) \vee (\neg Member(x) \& \neg Member(y))$$

To be connected, there is either an edge between the nodes or a node (nodes) that connect to the two nodes.

$$Connected(x, y, G) \leftrightarrow \exists e : Edge(e) \& Node(x) \& Node(y) \& EdgeBetween(x, y) \vee \exists z Connected(z, x) \& Connected(z, y)$$

I assume we can tell if a node is in the graph (Node), if the node is a multicast node (Member) and if there exists a direct edge between two nodes (EdgeBetween).

- 3) Prove that the value of information is nonnegative (see AIMA: Section 16.6: The value of information). (15p)

We seek to prove the following:

$$e, E_j VPI_e(E_j) \geq 0$$

I proceed with proof by contradiction.

Suppose

$$\exists e, E_j VPI(E_j) < 0$$

Using the definition of  $VPI$ .

$$\exists e, E_j (\sum_k P(E_j = e_{jk}|e) EU(\alpha_{e_{jk}}|e, E_j = e_{jk}) - EU(\alpha|e) < 0$$

$$(\sum_k P(E_j = e_{jk}|e) EU(\alpha_{e_{jk}}|e, E_j = e_{jk}) < EU(\alpha|e)$$

By removing the sum, we preserve the nature of the inequality.

$$\sum_k P(E_j = e_{jk}|e) \max_a \sum P(RESULT(a) = s'|a, e, e_j) U(s') < \max_a \sum P(RESULT(a) = s'|a, e) U(s')$$

$$\max_a \sum P(RESULT(a) = s'|a, e, e_j) U(s') < \max_a \sum P(RESULT(a) = s'|a, e) U(s')$$

This states that that best action given the additional piece of evidence is less than the best action without the additional piece of evidence, which cannot be true. Thus we have a contradiction.

Additionally, we can use the property that the value of further information is irrespective of its ordering. While the probability of information being significantly helpful decreases with the amount of information we have amassed, we know that this increment is never negative. Regardless of the ordering of the information, the added information cannot cause us to make a less-informed decision, even if the information is not particularly helpful or relevant.

- 4) For the Bayesian network in the 2nd major homework and following the data generation described in the 3rd major homework, apply the Steiner tree problem solver for the whole graph with edge weights defined by the pairwise mutual information. The learned tree for all the nodes/variables is a maximum weight spanning tree, where the weights are the mutual information. Give a formula for the entropy of your original network and the entropy of this tree-shaped Bayesian network. Calculate the entropy of this tree-shaped Bayesian network and discuss its relation to the entropy of your original network. (15p)

(see learnedtree.py)

Since my original network didn't have many nodes and was already relatively optimized, we wouldn't expect a significantly different entropy.