

Exam Data Mining

November 4, 2020, 17.00-20.00 hours

General Instructions

1. Write your name and student number on every sheet of paper you hand in.
2. You are allowed to consult 1 A4 sheet with notes written (or printed) on both sides.
3. You are allowed to use a (graphical) calculator.
4. Always show how you arrived at the result of your calculations. Otherwise you can not get partial credit for incorrect final answers.
5. This exam contains five questions for which you can earn 100 points.

Question 1: Mixed Short Questions (20 points)

Answer the following questions:

- (a) In classification trees, what is the purpose of pruning?
- (b) In ensemble methods for trees, what is the difference between bagging and random forests?
- (c) We use the following string representation of a labeled rooted ordered tree: list the labels according to the depth-first pre-order traversal of the tree, and use the special symbol \uparrow to indicate that we go up one level in the tree.

Consider the labeled rooted ordered trees $T_1 = ab \uparrow c$ and $T_2 = abc \uparrow\uparrow cc \uparrow\uparrow b$. How many times does T_1 occur as an embedded subtree of T_2 ?

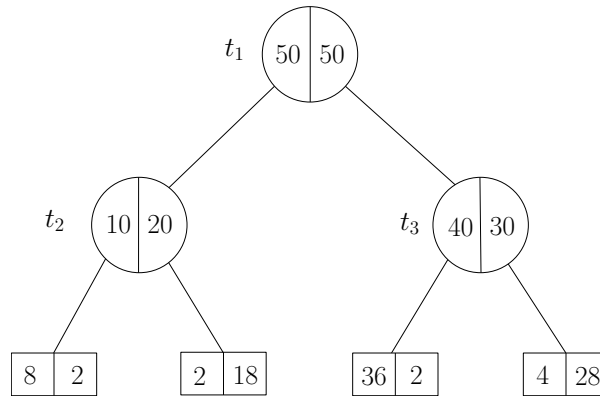
- (d) In text mining, list the bigrams that occur in:

`never say never again`

- (e) In link-based classification (article of Lu and Getoor), what problem do we run in to when we want to use the trained model to predict the node labels on a test set with all node labels unknown? How do Lu and Getoor address this problem?

Question 2: Classification Trees (20 points)

The tree T_{\max} given below has been grown on the training sample.



In each node, the number of observations with class 0 is given in the left part, and the number of observations with class 1 in the right part. The leaf nodes have been drawn as rectangles.

- (a) Determine the impurity of nodes t_1 , t_2 , and t_3 . Use the gini-index.
- (b) Give the impurity reduction achieved by the split in the root node.
- (c) Give the cost-complexity pruning sequence $T_1 > \dots > \{t_1\}$. For each tree in the sequence, give the interval of α values for which it is the smallest minimizing subtree of T_{\max} .

Question 3: Frequent Sequence Mining (20 points)

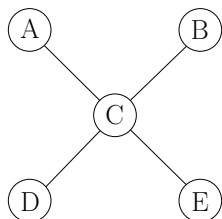
Consider the following database of sequences over alphabet $\{M,N,O,R\}$:

sid	sequence
1	MOON
2	MONO
3	MORON

Use the GSP algorithm to find all frequent sequences with $\text{minsup}=2$. For each level, give a table listing all candidates (and only the candidates) and their support. Indicate whether or not a candidate sequence is frequent.

Question 4: Undirected Graphical Models (20 points)

Consider a graphical log-linear model with the following independence graph:



- (a) Give a formula for the maximum likelihood fitted counts for this model. You are not required to show how you derived the formula.
- (b) Argue that the directed independence graph (Bayesian network structure) that has directed edges going from C to every other node (and no other edges) is equivalent to the given undirected graph.
- (c) Show that the formula for the maximum likelihood fitted counts of the Bayesian network given under (b) is indeed equivalent to the formula for the undirected graphical model.

Question 5: Bayesian Networks (20 points)

To find a good Bayesian network structure on four binary variables A, B, C, D we perform a hill-climbing local search starting from the empty graph (the mutual independence model). Neighbor models are obtained by adding, deleting, or reversing an edge. In iteration 1 of the search we compute the Δ scores of all possible operations in the initial model. Answer the following questions.

- (a) The table of counts on variables A and D is given by:

	D	0	1
A			
0		40	20
1		10	30

Compute the change in log-likelihood score if we add the edge $A \rightarrow D$ in the initial model. Always use the natural logarithm in your computations.

- (b) Compute the change in BIC score if we add the edge $A \rightarrow D$ in the initial model.

PLEASE TURN PAGE OVER!

(c) Suppose we find that $\Delta \text{ add } (A \rightarrow D)$ is largest, so in iteration 1, we add an edge from A to D . Assume that Δ scores of operations that have been computed in previous iterations and that are still valid, are not recomputed, but retrieved from memory. For which of the following operations do we need to compute the Δ score in iteration 2? (0 or more answers may be correct)

1. $\text{add}(B \rightarrow A)$.
2. $\text{add}(B \rightarrow D)$.
3. $\text{add}(C \rightarrow D)$.
4. $\text{add}(A \rightarrow C)$.