Exam Data Mining
January 6, 2021, 15.15-18.15 hours

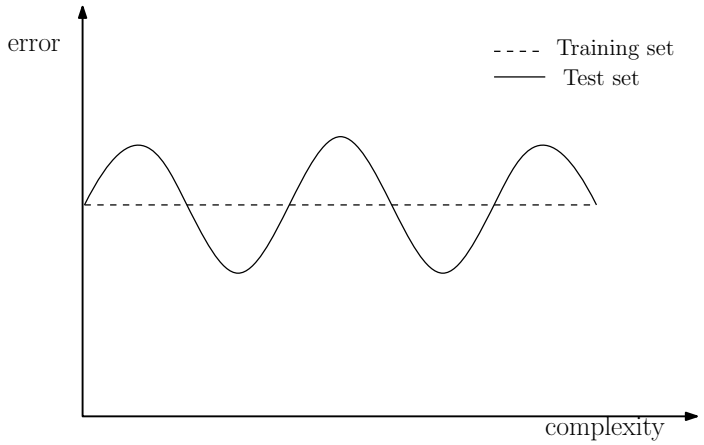
General Instructions

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 (25 points)

Answer the following questions:

(a) Sketch the typical behaviour of the error rate of a classifier (e.g. a classification tree) on the training set and on the test set as a function of its complexity. An example answer is given below (hint: the example answer is not correct).



(b) In random forests we aim to reduce the prediction error of the ensemble of trees by only allowing splits on a random subset of the features at each node. How could that reduce the prediction error?

(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 $T_2 = aac \uparrow b \uparrow\uparrow cb \uparrow\uparrow bc$. How many times does T_1 occur as an embedded subtree of T_2?

(d) In text mining, list the bigrams that occur in: `you only live twice`

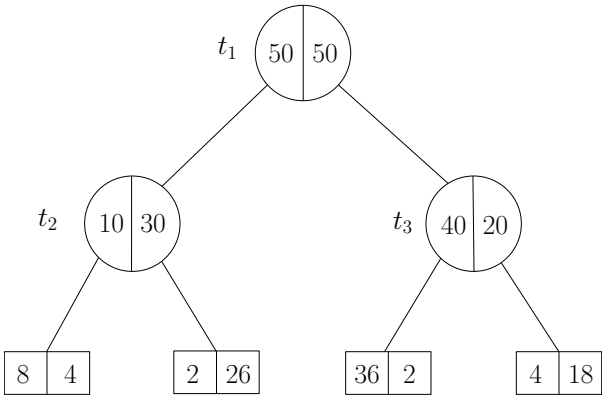
(e) In link prediction, consider the following feature:

$$f(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|},$$

where x, y and z are nodes in the graph, and $\Gamma(x)$ is the set of nodes directly connected to x. Explain in words what this feature tries to capture. You may give a concrete link prediction problem to illustrate its meaning.

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 $T_1 > \ldots > \{t_1\}$. 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)

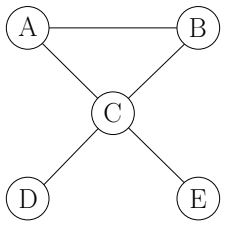
Consider the following database of sequences over alphabet {A,D,R,T}:

sid	sequence
1	DATA
2	TADA
3	DRATRA

Use the GSP algorithm to find all frequent sequences with 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 (15 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) Give a directed independence graph (Bayesian network structure) that is equivalent to the given undirected graph.

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 and 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 B and C is given by:

B \ C	0	1
0	40	5
1	10	45

Compute the change in log-likelihood score if we add the edge $B \to C$ in the initial model. Always use the natural logarithm in your computations.

(b) Compute the change in BIC score if we add the edge $B \to C$ in the initial model.

(c) Suppose we find that Δ add $(B \to C)$ is largest, so in iteration 1, we add an edge from B to C. 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. add$(A \to C)$.
2. add$(B \to D)$.
3. add$(A \to B)$.
4. add$(D \to C)$.