

Problem Set I

October 3, 2009

We consider objects x with p binary features. We will write $x.f$ for the f th feature of object x where we have $x.f \in \{0, 1\}$. A decision tree is a binary tree where every internal node is labeled with a feature and every leaf node is labeled with either 0 or 1. A tree with root node labeled with f and with left and right subtrees T_L and T_R respectively will be written as $f ? T_L : T_R$ (by analogy with C conditional expressions). For example we might have the following tree.

$$f ? (g ? 0 : 1) : (h ? 1 : 0)$$

For a tree T we write $T[x]$ for the value of the tree on the object x . This value is defined by the following equations.

$$(f ? T_L : T_R)[x] = \begin{cases} T_L[x] & \text{if } x.f = 1 \\ T_R[x] & \text{if } x.f = 0 \end{cases}$$

$$1[x] = 1$$

$$0[x] = 0$$

We let $|T|$ be the number of bits it takes to represent T . A code for representing trees as bit strings can be devised such that we have the following where n is the number of nodes in T .¹

$$|T| = (1 + (1 + \lceil \log_2 p \rceil)/2)n + (1 - (1 + \lceil \log_2 p \rceil)/2)$$

You have been provided with training data and test data each of which is a set of pairs (x, y) with $y \in \{0, 1\}$ and where each object x has p binary features. You have also been given a learning program for constructing a decision tree in a way that heuristically minimizes the following objective where C is a parameter of the learner and $I[\Phi]$ is the indicator function for the statement Φ , i.e., $I[\Phi] = 1$ if Φ is true and 0 otherwise.

$$T^* = \operatorname{argmin}_T C \sum_{i=1}^N I[y_i \neq T(x_i)] + |T|$$

¹We use one bit to flag whether a node is an internal node or a leaf node. If the node is a leaf node we use one more bit to indicate the value at that node. For internal nodes we code the feature used at that node with $\lceil \log_2 p \rceil$ bits. The code for a tree is the code for the root node followed by the code for left subtree followed by the code for the right subtree. The number of nodes of a binary tree is always odd. A tree with n nodes has $(n + 1)/2$ leaf nodes and $(n - 1)/2$ internal nodes.

1. Use the given learning code to learn a tree from the training data with $C = 1$ as suggested by the (2.20) of the Occam's razor notes.

- a.** Compute the error rate of T on the training data.
- b.** Compute the error rate of T on the test data.
- c.** Compute the 99% percent confident upper bound on the error rate of T by applying (2.14) from the notes on Occam's razor.
- d.** Use Hoeffding's confidence interval, equation (2.7) from the Occam razor note, to give the 99% confidence interval for the generalization error of T as measured by the test data.
- e.** Explain why it is not valid to apply Hoeffding's confidence interval (2.7) to the error rate of T as measured on the training data.

2. We will now explore the empirical dependence on the parameter C .

a. Graph $|T|$, the training error rate, and test error rate, as a function C for trees trained from the training data for values of C from $1/8$ to 8 with steps increasing by a factor of $\sqrt{2}$, i.e. $1/8, \sqrt{2}/8, 1/4, \dots, 4\sqrt{2}, 8$.

b. Divide the training data into three sets of equal size. Repeat **a.** but for training on just the first third of the training data and repeat again training on the first two thirds of the training data. Observe how the optimal value of C and the optimal value $|T|$ (as measured by minimizing test error) depend on the sample size m .

c. Repeat **a.** but training on the first two thirds of the training data and reporting the error on the last third of the training data rather than the error on the test data. Select the value of C minimizing the error on the last third of the test data. This is called tuning C with hold out data. Test the tree corresponding to this value of C (as trained on the first two thirds of the training data) on the test data. Compare the result with **1.b** and **2.a** above.