Statistical Learning, Second Exam, April 23rd, 2013

You get 1 point for free, a maximum of 2 points for each of questions 1, 3 and 4, and a maximum of 3 points for question 2. Good luck!

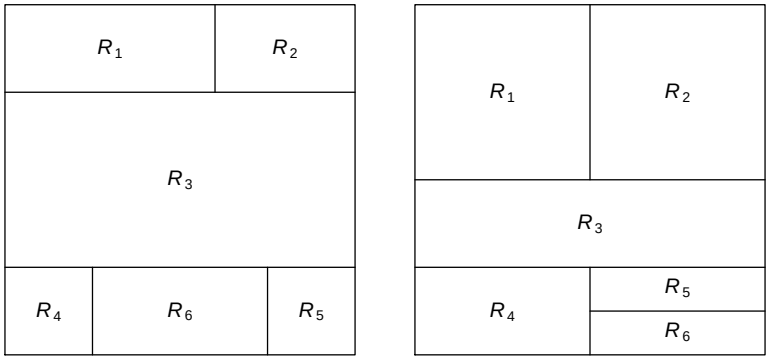

Figure 1: Ordinary classification tree (left) and dyadic classification tree (right).

1. *Classification trees*. A *dyadic* classification tree recursively partitions the feature space into two subregions along one of the axes, similarly to the ordinary classification trees that were discussed during the lectures. However, while ordinary trees subsequently test each possible split-point on each axis, dyadic trees can only make a partition exactly in the middle of the current region. Figure 1 shows possible partitions obtained using the two kinds of trees for a two-dimensional feature space. Give the answer to the following questions:

   a) What do you think are the main advantages and main disadvantages of the dyadic tree over the ordinary tree? Consider:
      - computational issues,
      - interpretability,
      - statistical issues (generalization ability); can you characterize the two kinds of trees in terms of the bias and variance (it is sufficient to give a qualitative answer, we do not expect you to give a formula here)?

   b) Are ordinary classification trees more general than dyadic ones? More precisely, if we think of a tree as a function $f: X \to Y$ assigning the label (output) $y \in Y$ to each input vector $x \in X$, can there be an ordinary tree f which is not expressible as a dyadic tree as well (assume that you analyze data on a computer, so that all x-values are recorded up the a fixed, finite precision)?

   c) Assume we control the complexity of the tree by fixing the maximal number of partitions to the same number (for concreteness, say 5) for both ordinary and dyadic trees. In such a case, which trees – ordinary or dyadic – would probably be more advantageous when combining them using boosting? Please motivate your answer.

2. *Variable Selection in Regression* Consider a linear regression problem in which we fit a model

$$E[Y \mid X] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_k X_k. \tag{1}$$

   Our main goal is to find out which of the variables $X_1, \ldots, X_k$ are relevant for predicting Y. To this end, for each subset $\mathcal{J}$ of $\{1, \ldots, k\}$, define $\mathcal{P}_{\mathcal{J}}$ as the model in which all variables

outside $\mathcal{J}$ are set to 0, i.e. $E[Y \mid X]$ is of the form $\beta_0 + \sum_{j\in\mathcal{J}} \beta_j X_j$ where the $|\mathcal{J}| + 1$ parameters are all real numbers.

To determine the 'best' subset $\mathcal{J}$ based on training sample $(x_1, y_1), \ldots, (x_n, y_n)$, we use three methods: (1) OLS+CV (ordinary least squares plus cross-validation); (2) Ridge Regression; (3) Lasso. In method 1, we split the training set in two equal parts and we fit, for each $\mathcal{J} \subseteq \{1, \ldots, k\}$, the corresponding model $\mathcal{P}_\mathcal{J}$ using ordinary least squares on the first sample. We then test the inferred parameter values by using them to predict the second sample and noting the mean squared error they achieve on that sample. In the end, we select the submodel $\mathcal{P}_\mathcal{J}$ that gave the best fit on the second sample. In method (2) and (3), we simply fit the full model using ridge regression and Lasso, respectively, and we select the submodel consisting of all variables X_j for which the corresponding β_j was not set to zero by the fitting procedure.

a) Compare (approximately) the computational efficiency of the three methods. Is there a clear winner/loser?

b) Compare the methods in terms of how well they work in practice for determining what variables are relevant for predicting Y (assuming enough computation time is available for each method). Is there a clear winner/loser?

c) Suppose that (without the statistician knowing this) the data are preprocessed: in each data point (x_i, y_i), $x_i = (x_{i1}, \ldots, x_{ik})$), for all $j = 1, \ldots, k$, x_{ij} is replaced by $x_{ij} + 5$. Will this affect the results of method (1)? Of method (2)? Of method (3)? Explain your answer.

d) Suppose that the data are preprocessed: in each data point (x_i, y_i), $x_i = (x_{i1}, \ldots, x_{ik})$), for all $j = 1, \ldots, k$, x_{ij} is replaced by $5 \cdot x_{ij}$. Will this affect the results of method (1)? Of method (2)? Of method (3)? Explain your answer.

e) Suppose that the data are preprocessed: in each data point (x_i, y_i), $x_i = (x_{i1}, \ldots, x_{ik})$), for all $j = 1, \ldots, k$, x_{ij} is replaced by $x_{ij} + 5j$. Will this affect the results of method (1)? Of method (2)? Of method (3)? Explain your answer.

f) Suppose that the data are preprocessed: in each data point (x_i, y_i), $x_i = (x_{i1}, \ldots, x_{ik})$), for all $j = 1, \ldots, k$, x_{ij} is replaced by x_{ij}^2. Will this affect the results of method (1)? Of method (2)? Of method (3)? Explain your answer.

g) Suppose that we really have only one input variable U, and we want to model Y as a polynomial of U, i.e, for some $d \geq 0$, $E[Y|U] = \beta_0 + \sum_{j=1}^{d} \beta_j U^j$. Describe how we can map this model to the model given by (1), and how we can then use the methods above to find a 'best' degree d based on the data. How do the computational requirements of the three methods compare in this new setting?

h) Suppose that we really have only one input variable U, and we want to model Y as a 'power law' of U, i.e. for some β_1, β_2, $E[Y \mid U] = \beta_1 U^{\beta_2}$. Can we still map this model to (1)? Why (not)? And can we still map the extended model $E[Y \mid U] = \beta_0 + \beta_1 U^{\beta_2}$ to (1)? Why (not)?

3. *Cross-validation*. The following questions are about estimation of the expected prediction error (EPE) using K-fold cross-validation (CV) in two-class classification problems. EPE is defined relative to the 0/1-loss function. Please motivate your answer in each case.

We will apply cross validation in combination with a very simple learning algorithm, the so-called *majority rule*. The majority rule neglects the input variables X_i in the training set $(X_1, Y_1), \ldots, (X_N, Y_N)$ and predicts according to the class that has occurred most frequently

in this training set (in case of a tie, the class is chosen at random). Thus, if the majority of the training instances have $Y_i = 1$, then, given a new training instance X_{NEW}, we predict the corresponding Y_{NEW} as 1 (irrespective of the value of X_{NEW}); if the majority has $Y_i = -1$, then we predict Y_{NEW} as -1, irrespective of X_{NEW}.

a. What is the true EPE of the majority rule if we assume the data are sampled independently from distribution $\Pr(X, Y)$ with equal class priors $\Pr(Y = 1) = \Pr(Y = -1)$?

b. Consider the case of CV with $K = N$ ("leave-one-out"). Suppose we have a training data set, *exactly* half of which $(N/2)$ belongs to the class $Y = 1$, while the other half belongs to the class $Y = -1$. What estimate of the EPE would we get from leave-one-out CV with this data set? What estimate would we (approximately) get from K-fold CV with a smaller value of K?

c. Can you come up with a simple classification rule/learning algorithm (just as simple as the majority rule) which would get a zero leave-one-out CV error for the considered data set? Would it be a reasonable classification rule in general?

4. *Linear Classification* Let $Y_i \in \{-1, 1\}$ and let the X_i be p-dimensional vectors of categorical (discrete) or real-valued attributes: $X_i = (X_{i,1}, \ldots, X_{i,p})$. Are the following statements true or false? Please motivate your answer in each case.

a. For every training set $(X_1, Y_1), \ldots, (X_N, Y_N)$ that is linearly separable, the decision boundary learned by logistic regression for that training set perfectly separates the data.

b. For every training set $(X_1, Y_1), \ldots, (X_N, Y_N)$ that is linearly separable, the decision boundary learned by using the Naive Bayes model with maximum likelihood, perfectly separates the data (naive Bayes is used here with a multinomial model for the categorical attributes, and a Gaussian model for the real-valued attributes).

c. For every training set $(X_1, Y_1), \ldots, (X_N, Y_N)$ that is linearly separable, the decision boundary learned by using the maximum-margin hyperplane (support vector machine) algorithm with a cost parameter $C > 0$, perfectly separates the data.