Statistical Learning Exam 7 January 2014

- You are allowed to use the book and your private notes. If you do not have a paper copy, you are allowed to read the book on your laptop if you disable all communications (wifi, bluetooth, etc.), you close all other windows and you have *nothing else on your screen* at any time.
- Every subquestion (1a, 1b, etc.) counts for the same number of points.
- You have to hand in the Appendix. Do not forget to write your name and student number!

1. **[Classification]** See the data set in Figure 1 in the Appendix.
   (a) What is the smallest number of mistakes on this training data that can be achieved by a linear classifier *without* an intercept? (That is, set $f_\beta(x) := \beta_1 x_1 + \beta_2 x_2$ and predict with $\text{sign}(f_\beta(x))$.) Draw the corresponding decision boundary in Figure 1a.
   (b) What is the smallest number of mistakes on this training data that can be achieved *with* an intercept? (That is, $f_\beta(x) := \beta_0 + \beta_1 x_1 + \beta_2 x_2$.)
   (c) Draw the decision boundary for an SVM (with intercept) in Figure 1b, and explain/illustrate why you drew it that way. For the SVM, assume we simply use the inner product as our kernel. (This is called the *support vector classifier* in the book.)
   (d) Suppose we run logistic regression (with intercept), but we restrict all three parameters $\beta_0, \beta_1, \beta_2$ to values between -10 and $+10$. Let $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2)$ be the parameters that maximize the conditional likelihood of the training set under this restriction. Now we multiply all parameters in $\hat{\beta}$ by 1000. What happens to the corresponding decision boundary?
   (e) Continuing from the previous question: What happens to the conditional likelihood $\Pr_{\hat{\beta}}(Y_i|X_i)$ for each of the points (X_i, Y_i) in the training set from Figure 1 when we multiply $\hat{\beta}$ by 1000? You do not have to give an exact number; just describe the qualitative effect.
   (f) Suppose we run logistic regression without restricting the parameters, where we again fit the parameters by maximizing the conditional likelihood on the training data. Explain why some parameters can go to infinity on the training data in Figure 1.
   (g) Suppose we use L1-penalized logistic regression instead. Explain why in this case parameters will not go to infinity.
   (h) Does L1-penalized logistic regression achieve the smallest possible number of mistakes on the training data from Figure 1 for penalty multiplier $\lambda = 0.01$? And for $\lambda = 10\,000$?

2. **[Bagging and Boosting]** Consider the training data from Figure 1 again. Recall that, for the case of two classes $+1$ and -1, a *decision stump* is a classifier that selects a single feature x_j, a threshold a and a class label $c \in \{-1, +1\}$; then it outputs class c if $x_j \geq a$ and class $-c$ if $x_j < a$. Assume that the feature $j \in \{1, 2\}$, the threshold a and the class label c are chosen to minimize the 0/1-loss on the training data (breaking ties in some arbitrary way).
   (a) Suppose we run AdaBoost (called *Adaboost.M1* in the book) with decision stumps only for a single iteration ($M = 1$). Draw the resulting decision boundary in Figure 1c. What is the corresponding number of mistakes?
   (b) Suppose we run AdaBoost with decision stumps for more iterations. Which examples will get increased weights in the second iteration ($m = 2$)? (Circle them in Figure 1c and explain.)
   (c) Suppose we run either bagging with decision stumps or AdaBoost with decision stumps on the training data in Figure 1. Which of the two would you expect to work best, and why?

3. [**Cross-validation for regression**]. The machine learning company *Earning by Learning* sells you its famous super-trade-mark regression learning algorithm. However, you find out by accident that the algorithm *does not really learn*: the output of the learning algorithm is always the same regression function, no matter what the training set is. You email the company to complain about this. In response, they send you 10 000 new learning algorithms, and they tell you that one of these will definitely perform well on your data; you can simply find a good one for your data by cross-validation. However, after some testing you find that the 10 000 new 'algorithms' still do not learn: they each output a different but fixed regression function, which does not change if you change the training set.

   (a) Suppose you are given a training set $\mathcal{T} = (X_1, Y_1), \ldots, (X_N, Y_N)$. Explain why, irrespective of the value of K, the K-fold cross-validation error for an algorithm that does not learn is equal to its training set error, and why this means that running K-fold cross validation for the 10 000 new learning algorithms amounts to empirical risk minimization.

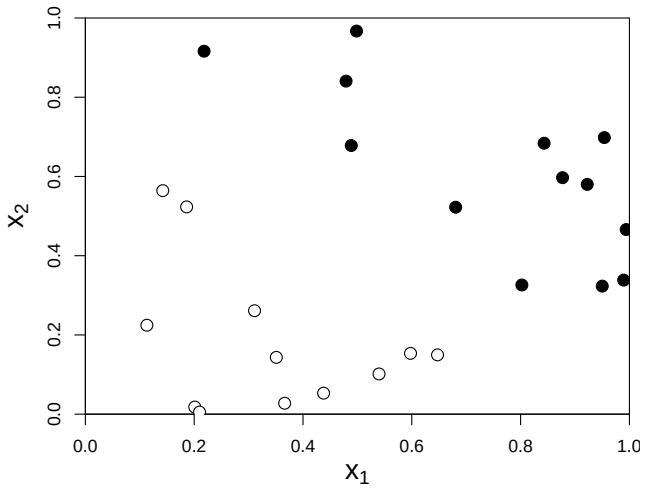
   (b) Frustrated with the company, you decide to learn a regression function for $\mathcal{T}$ by ridge regression instead. Suppose you first determine the linear regression parameters $\hat{\beta}_\lambda$ using ridge regression on the full set $\mathcal{T}$ for each value of the penalty multiplier λ in the set $\{0.00, 0.01, 0.02, \ldots, 100.00\}$. Then you use cross-validation, again on the full set $\mathcal{T}$, to choose between the various $\hat{\beta}_\lambda$. In what sense is this procedure different from the standard cross-validation method to learn λ in ridge regression? Can you say anything about the λ that you will tend to choose and explain whether it is a good choice?

   (c) Suppose that I now use the standard cross-validation method to learn λ in ridge regression. Now I repeat the same procedure after a rescaling, where I multiplied all features by 5. Will this change the λ I find by cross-validation? And will it change the predicted values for Y given X according to the fitted $\hat{\beta}_\lambda$? What happens if I use the Lasso instead of ridge regression? Explain your answers.

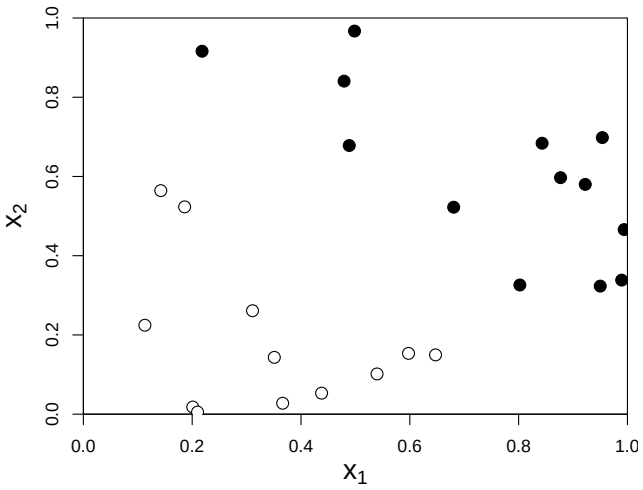
Appendix to the Statistical Learning Exam 7 January 2014

Name:

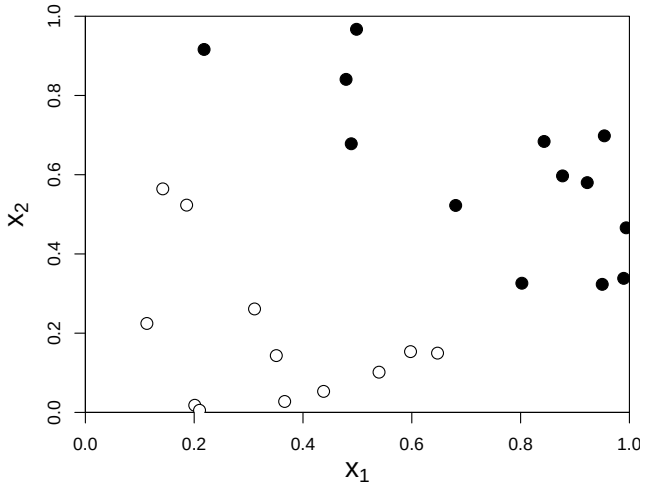
Student number:



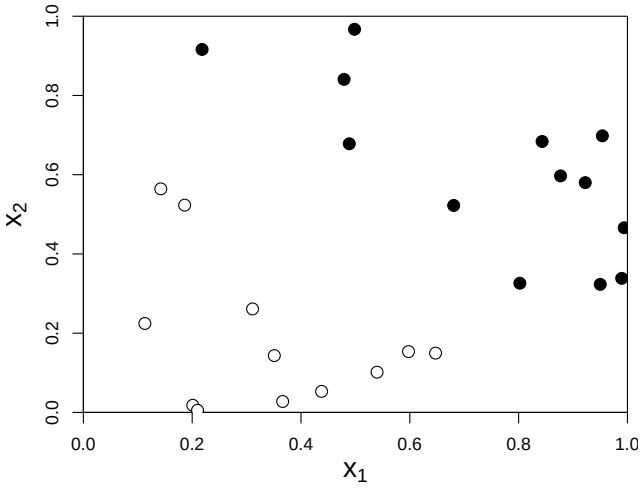
(a) Smallest nr. of mistakes



(b) SVM



(c) AdaBoost



(d)

Figure 1: Four times the same classification data set for binary classification with two features. For concreteness, let us say that the filled black dots have label $+1$ and the circles have label -1. (You can use Figure 1d as a backup in case you make a drawing mistake in one of the other figures.)