582631 Introduction to Machine Learning, Autumn 2013
Exercise set32 (based on second week of lectures)

To be discussed in the exercise session 15 November.

Credit is given based on the written solutions turned in to the course assistant. Extra credit (0.5 points per problem) is given for willingness to present the solution to the pen-and-paper problems at the exercise session.

The deadline for turning in your solutions is **9:00am on Wednesday, 13 November.**

Send your solutions to the course assistant (Yuan.Zou@cs.helsinki.fi) by e-mail. You should send one PDF file including your solutions to the pen-and-paper problems and the report for the programming problem, and a single compressed file including the code for the programming problem.

Pen-and-paper problems

Problem 1 (3 points)

Consider a binary classification task with two classes, '+' and '–'. The desired prediction is either one of the two classes, or alternatively answering 'don't know', with the following cost matrix:

		predicted class		
		–	+	don't know
actual class	–	$0	$10	$2
	+	$3	$0	$1

The goal is to minimize the expected cost (which is also the long-term average cost). You are given a classification method that returns a probabilistic prediction, i.e. for each new object the existing method gives you the probability α that the object belongs to class '+' (and hence $1-\alpha$ is the probability that the object belongs to class '-').

Give the optimal policy of answering '+', '–', or 'don't know' for each object, as a function of the value of α. That is, for any given value of α (for $0 \leq \alpha \leq 1$) what is the best answer of the three possibilities?

Problem 2 (3 points)

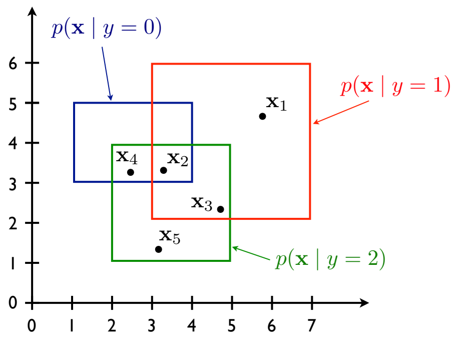
In a binary classification task we are trying to classify the objects $\mathbf{x}$ into classes $y = 1$ and $y = 0$. We are here using probabilistic predictions, so for each new object we are required to output our probability $b = P(y = 1 \mid \mathbf{x})$, with $0 \leq b \leq 1$. We are using the following quadratic cost function:

$$C(y, b) = \begin{cases} (b-1)^2, & \text{if } y = 1 \\ b^2, & \text{if } y = 0 \end{cases} \tag{1}$$

Obviously, if the value of y is 1, the lowest cost (0) is obtained when $b = 1$. Similarly, if the value of y is 0, the lowest cost (0) is obtained when $b = 0$, so the cost function seems to encourage appropriate behaviour. In fact, the cost function is *proper*. Show this. (Hint: Assume that the true probability that $y = 1$ is a (so the true probability that $y = 0$ is $1 - a$), and show that the expected cost is minimized for $b = a$.)

Problem 3 (3 points)

The figure below shows the class conditional probability densities $p(\mathbf{x} \mid y)$ of 3 different classes ($y = 0$, $y = 1$, and $y = 2$): Each of the class conditionals is a uniform density over the corresponding rectangle. (For example, all vectors $\mathbf{x}$ belonging to class $y = 0$ have attribute values $x_1 \in [1, 4]$ and $x_2 \in [3, 5]$, and these vectors are uniformly distributed within this rectangle.) Additionally we know that the marginal probabilities of the classes are $P(y = 0) = 0.2$, $P(y = 1) = 0.7$, $P(y = 2) = 0.1$. For each of the indicated points $\mathbf{x}_1, \ldots, \mathbf{x}_5$, compute the posterior probability distribution over the three classes. (That is, when observing a new vector $\mathbf{x}_i$, what is the probability that it belongs to class $y = 0$, to class $y = 1$, or to class $y = 2$? Remember that these probabilities should sum to one.)



Computer problem

General instructions:

- Return a brief written report (as PDF, included in the same file as your pen-and-paper solutions) and one directory (as a zip or tar.gz file) including all your code.
- Do not include any of the data files in your solution file.
- Always mention your name and student ID in the report.
- We use the report as the main basis for grading: All your results should be in the report. We also look at the code, but we won't however go fishing for results in your code.
- The code needs to be submitted as a runnable file or set of files (command to run it given in the report).
- In your report, the results will be mostly either in the form of a figure or program output. In both cases, add some sentences which explain what you are showing and why the results are the answer to the question.
- If we ask you to test whether an algorithm works, always give a few examples showing that the algorithm indeed works

Problem 4 (15 points)

In this exercise we implement an extremely simple prototype-based classifier to classify handwritten digits from the MNIST dataset, and compare that to a nearest-neighbor classifier.

(a) Download the MNIST data from the course web page. In addition to the actual data, the package contains some functions for easily loading the data into Matlab/Octave/R and for displaying digits. See the README files for details. Load the first N=5,000 images using the provided function.

(b) Use the provided functions to plot a random sample of 100 handwritten digits, and show the associated labels. Verify that the labels match the digit images. (This is a sanity check that you have the data is in the right format.)

(c) Divide the data into two parts: A 'training set' consisting of the first 2,500 images (and associated labels), and a 'test set' containing the remaining 2,500 images (and their associated labels).

(d) For each of the ten classes (digits 0-9), compute a class *prototype* given by the mean of all the images in the training set that belong to this class. That is, select from the training set all images of class '0' and compute the mean image of these; this should look sort of like a zero. Do this for all ten classes, and plot the resulting images. Do they look like what you would expect?

Continues on the next page!

(e) For each of the images in the test set, compute the Euclidean distance of the image to all 10 prototypes, and classify the test image into the class for which the distance to the prototype is the smallest. So, if a test image is closer to the prototype for '3' than it is to the prototypes for any of the other digits, predict its class to be '3'. Compute and display the resulting *confusion matrix.*

(f) Classify each of the test images with a nearest neighbor classifer: For each of the test images, compute its Euclidean distance to all (2,500) of the training images, and let the predicted class be the class of the closest training image. Compute and display the resulting confusion matrix.

(g) Compute and compare the error rates of both classifiers (the prototype-based classifier and the nearest neighbor classifier). Which is working better? Based on the confusion matrix, which digits are confused with each other? Why do you think this is?