

**Homework Set One**  
ECE 271A  
Department of Computer and Electrical Engineering  
University of California, San Diego  
Nuno Vasconcelos

The purpose of this assignment is to give you experience with Bayesian decision theory. The first four problems are of the pen-and-paper type, while problem 5 is a computer problem. Some of the problems simply require the straightforward application of the principles covered in class. These are intended to give you some practice on the Bayesian manipulations that are usually involved in decision theory. Others involve somewhat deeper thinking and extend in some way what we have seen in class. The problems are not ordered by difficulty.

**1.** In this problem we will consider the traditional probability scenario of coin tossing. However, we will consider two variations. First, the coin is not fair. Denoting by  $S$  the outcome of the coin toss we have

$$P_S(\text{heads}) = \alpha, \alpha \in [0, 1].$$

Second, you do not observe the coin directly but have to rely on a friend that reports the outcome of the toss. Unfortunately your friend is unreliable, he will sometimes report heads when the outcome was tails and vice-versa. Denoting the report by  $R$  we have

$$P_{R|S}(\text{tails}|\text{heads}) = \theta_1 \tag{1}$$

$$P_{R|S}(\text{heads}|\text{tails}) = \theta_2 \tag{2}$$

where  $\theta_1, \theta_2 \in [0, 1]$ . Your job is to, given the report from your friend, guess the outcome of the toss.

**a)** Given that your friend reports heads, what is the optimal decision function in the minimum probability of error sense. That is, when should you guess heads, and when should you guess tails?

**b)** Consider the case  $\theta_1 = \theta_2$ . Can you give an intuitive interpretation to the rule derived in **a)**?

**c)** You figured out that if you ask your friend to report the outcome of the toss various times, he will produce reports that are statistically independent. You then decide to ask him to report the outcome  $n$  times, in the hope that this will reduce the uncertainty. (Note: there is still only one coin toss, but the outcome gets reported  $n$  times). What is the new minimum probability of error decision rule?

**d)** Consider the case  $\theta_1 = \theta_2$  and assume that the report sequence is *all heads*. Can you give an intuitive interpretation to the rule derived in **c)**?

**2.** Problem 2.2.2 in DHS.

**3.** Problem 2.5.23 in DHS (feel free to use MATLAB here to compute eigenvalues, matrix vector products, and so forth. The goal is not evaluate your mastery of the mechanics of linear algebra, but to give you an intuitive understanding of what whitening is).

**4.** Problem 2.9.43 in DHS

**5. (computer)** This is the first in a series of computer problems where we will get a feel for the difficulty of practical pattern recognition problems, such as computer vision. The goal is to segment the “cheetah” image (shown below in the left) into its two components, cheetah (foreground) and grass (background).

To formulate this as a pattern recognition problem, we need to decide on an observation space. Here we will be using the space of  $8 \times 8$  image blocks, i.e. we view each image as a collection of  $8 \times 8$  blocks. For each block we compute the discrete cosine transform (function `dct2` on MATLAB) and obtain an array of  $8 \times 8$  frequency coefficients. We do this because the cheetah and the grass have different textures, with different frequency decompositions and the two classes should be better separated in the frequency domain. We then convert each  $8 \times 8$  array into a 64 dimensional vector because it is easier to work with vectors than with arrays. The file `Zig-Zag Pattern.txt` contains the position (in the 1D vector) of each coefficient in the  $8 \times 8$  array. The file `TrainingSamplesDCT_8.mat` contains a *training set* of vectors obtained from a similar image (stored as a matrix, each row is a training vector) for each of the classes. There are two matrices, `TrainsampleDCT_BG` and `TrainsampleDCT_FG` for foreground and background samples respectively.

To make the task of estimating the class conditional densities easier, we are going to reduce each vector to a scalar. For this, for each vector, we compute the index (position within the vector) of the coefficient that has the  $2^{nd}$  largest energy value (absolute value). This is our observation or feature  $X$ . (The reason we do not use the coefficient with the largest energy is that it is always the so-called “DC” coefficient, which contains the mean of the block). By building an histogram of these indexes we obtain the class-conditionals for the two classes  $P_{X|Y}(x|cheetah)$  and  $P_{X|Y}(x|grass)$ . The priors  $P_Y(cheetah)$  and  $P_Y(grass)$  should also be estimated from the training set.

**a)** using the training data in `TrainingSamplesDCT_8.mat`, what are reasonable estimates for the prior probabilities?

**b)** using the training data in `TrainingSamplesDCT_8.mat`, compute and plot the index histograms  $P_{X|Y}(x|cheetah)$  and  $P_{X|Y}(x|grass)$ .

**c)** for each block in the image `cheetah.bmp`, compute the feature  $X$  (index of the DCT coefficient with  $2^{nd}$  greatest energy). Compute the state variable  $Y$  using the minimum probability of error rule based on the probabilities obtained in a) and b). Store the state in an array  $A$ . Using the commands `imagesc` and `colormap(gray(255))` create a picture of that array.

**d)** The array  $A$  contains a mask that indicates which blocks contain grass and which contain the cheetah. Compare it with the ground truth provided in image `cheetah_mask.bmp` (shown below on the right) and compute the probability of error of your algorithm.

Notes:

- in `TrainingSamplesDCT_8.mat` each matrix row contains the “zig-zag scanned” vector of coefficients. So, for **a)** and **b)** you do not need to compute the DCT, etc.
- the training data that we provide was created by representing images as doubles in the range  $[0, 1]$ . You need to represent the image in the same way, in order to have consistency between the train and test sets. This can be done by using the commands `im2double(img)` or `double(img)/255` on MATLAB (where `img` is the array that contains the image as read from `cheetah.bmp`).
- in `Zig-Zag Pattern.txt` the zig-zag pattern goes from 0 to 63, not 1 to 64. Please remember

this as MATLAB starts indexes from 1. (You can just add 1 to the numbers in the file to get to the MATLAB coordinates).

