CS 434a-654b

Assignment 2 (*Due Feb. 22)*

Instructions: Hand in all the work you do. This should include matlab code, print outs of graphs, if any, and written answers. Also, the matlab code should be emailed to me as scripts or function files. Make sure your email includes your name in the subject, and explanation which scripts go with which problem in the body of the email. The paper copy of the homework should be put in the locker #87 in the basement of the Middlesex college building by the midnight of the due date. Please note that for the undergraduate students, the maximum score is 100, and for graduate students, the maximum score is 120. (Of course, a score of 100 for undergrads is 100% on the homework, and for graduate students the score of 100 is just 83%)

Problem 1 (15% for undergraduates, 20% for graduate students): Do problem 1 on p. 140. Undergraduate students do parts (a) and (b). Graduate students do all parts.

1. Let x have an exponential density

$$p(x|\theta) = \begin{cases} \theta e^{-\theta x} & x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

(a) Plot $p(x|\theta)$ versus x for $\theta = 1$. Plot $p(x|\theta)$ versus θ, $(0 \leq \theta \leq 5)$, for $x = 2$.

(b) Suppose that n samples $x_1, ..., x_n$ are drawn independently according to $p(x|\theta)$. Show that the maximum likelihood estimate for θ is given by

$$\hat{\theta} = \frac{1}{\frac{1}{n}\sum_{k=1}^{n} x_k}.$$

(c) On your graph generated with $\theta = 1$ in part (a), mark the maximum likelihood estimate $\hat{\theta}$ for large n.

Problem 2 (10%): Consider a one dimensional two class classification problem, where we have collected the following data for each class: $\boldsymbol{D_1}$ = {-3,-2,8,3,6,7} and $\boldsymbol{D_2}$ = {-5,-2,3,4,8}. Suppose we decided to use Parzen windows with window width $\boldsymbol{h = 2}$ and $\boldsymbol{\varphi(x)}$ defined as

$$\boldsymbol{\varphi(u)} = \begin{cases} \boldsymbol{4e^{-4u}} & \boldsymbol{u \geq 0} \\ \boldsymbol{0} & \boldsymbol{\text{otherwise}} \end{cases}$$

(a) Classify sample $\boldsymbol{x = 4}$ using the ML classifier
(b) Plot the density estimate $\mathbf{p}_{\boldsymbol{\varphi}}(\mathbf{x})$ in matlab

Problem 3 (40%): In this problem we will try to build a simple face recognition system. You will have 58 face images for training, 58 non-face images for training, 57 face images for testing, 57 non-face images for testing. Each image is of size 24 by 24, but will be stored as a feature vector of size 24*24=576. To download this data into Matlab, use the provided file A2.mat. Simply use command "load A2" in matlab. When you type who, you will see 4 matrices:

faceTrain --- is a 58 by 576 array of face examples for training
faceTest --- is a 57 by 576 array of face examples for training
nonfaceTrain --- is a 58 by 576 array of non-face examples for training
nonfaceTest --- is a 57 by 576 array of non-face examples for training

The examples are stored as rows in each of the matrix above. I have provided a function display_image to visualize the images. To see what the 2nd training image for faces looks like, use in matlab display_image(faceTrain(2,:),24,24). The first parameter just takes the 2nd row of faceTrain matrix, which is just the 2nd image for face training, and the last 2 parameters specify the image dimensions.

(a) (5%) Assume that both the face and nonface class can be modeled by a multivariate gaussian distribution. Use ML estimation (see lecture notes for the mean and covariance ML estimates, you do not need to derive them!), to find the parameters for class conditional densities for both classes. Warning, this is a tricky question, try this but you will run into problems. Explain the problems you are having and take your best guess of why problems arise.

(b) (5%) For this part, let's still assume now that both the face and nonface class can be modeled by a multivariate gaussian distribution. However, now assume that the sample features are independent, that is the covariance matrix is diagonal. Find ML estimates of the mean and the variances for the multivariate distribution. (The MLE estimate of σ_i^2, that is the ith entry on the diagonal of the covariance matrix is just the sample variance of the ith feature). Visualize the mean and variance of the face class and the mean and the variance of the nonface class. Discuss your results.

(c) (20%) Write a matlab function ClassifyNormal which takes 4 arguments as an input, namely faceTrain, faceTest,nonfaceTrain ,nonfaceTest and outputs a 2 by 2 confusion matrix, where the (i,j)th entry of the confusion matrix is the number of samples of class i that have been classified as class j. Notice that trace(confusion matrix) gives the number of correctly classified examples. The function ClassifyNormal should classify faceTest and nonfaceTest examples assuming that classes have normal distributions with independent features.

(d) (10%) Now write a matlab function ClassifyNormalNormalized which takes 4 arguments as an input, namely faceTrain, faceTest,nonfaceTrain ,nonfaceTest and outputs a 2 by 2 confusion matrix. It will do the exactly the same thing as the function ClassifyNormal above, but before estimating MLE parameters and classifying, the features in all the 4 matrices are normalized, that is each sample in the 4 arrays is normalized so that it has mean 0 and variance 1. Compare the performance of this classifier with the one in (d) and discuss any difference in performance.

Problem 5 (35%): Use the same dataset for this problem as the previous problem.

(a) (20%) Write a matlab function ClassifyKnn which takes 5 arguments as an input, namely k (an integer), faceTrain, faceTest,nonfaceTrain ,nonfaceTest , and outputs a 2 by 2 confusion matrix, where the (i,j)th entry of the confusion matrix is the number of samples of class i that have been classified as class j. Notice that trace(confusion matrix) gives the number of correctly classified examples. The function ClassifyNormal should classify faceTest and nonfaceTest examples using the kNN classifier.

(b) (5%) Write a matlab function ClassifyKnnNormalized which does the exactly the same thing as the function ClassifyKnnl above, but before classifying all the features in all the 4 matrices are normalized, that is each sample in the 4 arrays is normalized so that it has mean 0 and variance 1. Compare the performance of this classifier with the one in (a) and discuss any difference in performance.

(c) (5%) Run the ClassifyKnn for different values of k and discuss the difference in performance as k grows from 1 to 10.

(d) Discuss the difference in performance between the kNN classifier and the classifier based on assumption on Normal density in problem 4

Problem 6 (15%): For graduate students.

Choose the best classifier for faces from problems 4 and 5 and run it on the .tif images provided which contain faces. To do this, you will have to apply your classifier to windows of different sizes in the input picture, at different scales. Choose a scaling factor that you think is appropriate. Mark correctly detected faces with red color and points falsely classified as faces with blue color. Hand in your code and printout of images.