

Problem Set 4

MAS 622J/1.126J: Pattern Recognition and Analysis

Due Wednesday, 25 October 2006

[Note: As usual, please use Python or Matlab when asked to plot data or write a program.]

Problem 1: ML Estimation After Dimensionality Reduction

Download this problem set's data set files from the course webpage. All these data sets consist of 3-dimensional data. There are data sets for two classes, class 0 and class 1. For each class, there are two training data sets, 'A' and 'B,' and one testing data set.

- a. Use Matlab or Python to reduce the dimensionality of the 'A' training data set for both classes from 3-dimensional to 1-dimensional using Principal Component Analysis (PCA). As usual, include your program in your answer.
- b. Use Matlab or Python to reduce the dimensionality of the 'A' training data set for both classes from 3-dimensional to 1-dimensional using Fisher Linear Discriminant (FLD). As usual, include your program in your answer.
- c. Use Matlab or Python to compute the maximum-likelihood mean and variance of the dimension-reduced 'A' training data set for both classes. Do this for both the data reduced by PCA and the data reduced by FLD. As usual, include your program in your answer.
- d. Assume the two classes are equally likely (i.e., $p(\omega_1) = p(\omega_2)$) and the likelihoods of observation given a class are Gaussian (i.e., $p(x|\omega_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$). Write a Matlab or Python program to classify the testing.dat data for both classes using the estimated means and variances using a 0-1 cost criterion. Report the recognition results for both the PCA-reduced and the FLD-reduced training sets. As usual, include your program in your answer.
- e. Repeat the above steps using the 'B' training data sets. Report only the final recognition results for the two methods of dimension reduction and comment on how the two training data sets might differ.

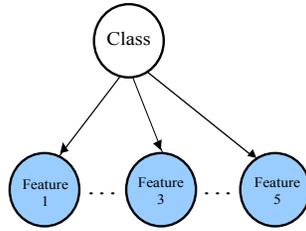


Figure 1: BN representing the hidden (unshaded) class and the (shaded) observed features 1 to 5.

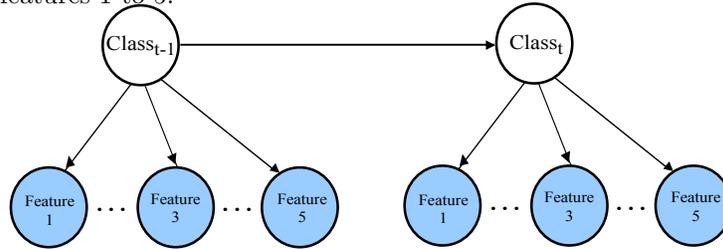


Figure 2: DBN representing the hidden (unshaded) class and the (shaded) observed features 1 to 5.

Problem 2: Bayesian Networks

- a. First, install the Bayes Net Toolkit for Matlab. You can download the toolkit and find the install instructions here:

<http://bnt.sourceforge.net/usage.html#install>

- b. Define the Bayes Net Structure: Create a MATLAB m-file that defines the Naive Bayesian classifier in Figure 1: five observed features (1–5) that are conditionally independent and are discrete and binary (i.e. can assume the values 0 or 1), and one root node that represents the class. This can take the value 1, 2 or 3.
- c. Learning the parameters: The conditional density functions are determined from the accompanying text file (this will constitute the observation matrix). Each row in the data file corresponds to a sample, the first column contains the correct class of the sample and the next columns contain the actual features that have been measured. Divide the data into two distinct sets: one for learning and one for testing (use 75% for training).
- d. Inference: Create a junction tree inference engine for inference (filtering). Use the portion of the data set aside for testing to test the classifier. Enter each row as evidence in to the inference engine, then calculate the probability of the class values given the observed feature values. Show a confusion matrix of the results.

- e. Dynamic Bayesian Networks: Modify the Bayesian Network classifier such that it is a Dynamic Bayesian network as shown in Figure 2 (the class value at time $t-1$ affects the class value at time t). Use the following as the conditional probability table for the transitions:

$$P(t = 1 | t - 1 = 1) = 0.1$$

$$P(t = 2 | t - 1 = 1) = 0.5$$

$$P(t = 3 | t - 1 = 1) = 0.4$$

$$P(t = 1 | t - 1 = 2) = 0.25$$

$$P(t = 2 | t - 1 = 2) = 0.5$$

$$P(t = 3 | t - 1 = 2) = 0.25$$

$$P(t = 1 | t - 1 = 3) = 0.25$$

$$P(t = 2 | t - 1 = 3) = 0.5$$

$$P(t = 3 | t - 1 = 3) = 0.25$$

Define the parameters of the DBN. Show the confusion matrix of the results using a DBN junction tree inference engine.

Problem 3: Hidden Markov Models

Use the HMM depicted in Fig. 3 to work out the following questions. Please work them out by hand, don't write any programs for this problem.

- In the given HMM, what is the probability that an observation sequence {AAB} was generated?
- Given that this HMM produced an observation sequence {AAB}, what is the most likely sequence of hidden states that led to those observations?
- If instead of being equally likely, the prior probabilities of the states is $\pi_1 = .75$ and $\pi_2 = .25$, how does this change your answers (a) and (b)?

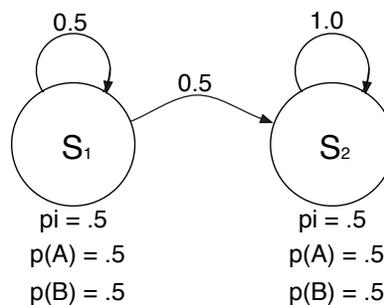


Figure 3: Two-state HMM.