Assignment: Supervised learning for digit classification

September 26, 2012

In this assignment, you will develop a classifier for handwritten digits. You will use the supervised learning paradigm, in which you are given some examples of two different classes: handwritten 3's and handwritten 5's.

You will be expected to develop a classifier, using the "training samples", that classifies the "test samples" as well as possible.

1. Download the file digits.mat from the course web page. Put the file in your current working directory. Use the command 'load digits.mat' to load it into matlab. Type 'whos' to see the variables that are defined in it. You should see four variables named train_threes, train_fives, test_threes, and test_fives. They are each a group of 50 images stored in a three-dimensional array. Try plotting a few of the images using imagesc to make sure they appear as you expect.

2. You will be developing a classifier based on maximum a posteriori (MAP) classification. To simplify matters, we will consider the class of threes to be "class 1" and the class of fives to be "class 2".

3. Write a function called likeFromTraining.m which estimates the likelihoods for a particular pixel in each class, using the training data. That is, given a particular pixel position, like $(23, 56)$, it should give the frequency of that pixel being "on" (meaning that its value is 255.5) in the set of training threes and the set of training fives. The function should return a column vector of 2 values, one of which is the likelihood for that pixel in the set of threes and the other for the set of fives.

4. Now write a function called BayesRule.m which takes 3 arguments: the value of a feature, the likelihoods for each class as a vector, and a vector of prior probabilities (one for each class), and returns the posterior probability for each class as a vector. In other words, your function should return a vector whose first element is the posterior probability of class 1 given the feature value and whose second value is the posterior probability of class 2 given the feature value.

5. Next, write a function called classifyTestData which takes the two sets of training images, two sets of test images, a prior probability for each class (as a vector), and a pixel location as arguments, and returns whether each image was classified correctly or incorrectly. It should call your likeFromTraining function and your BayesRule function.

6. Using a prior probability of 0.5 for each class, try training and testing your classifier for various different pixel positions. Try to find the position that gives the highest accuracy. Choose five different pixel positions to show that some pixel positions are better than others.

7. Now report your results for each of the five pixel positions using each of the following 11 different values for the prior probability of "3": $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. Report the percentage of correct 3's, the percentage of correct 5's, and the overall percentage correct for all 5 pixel positions and all 11 different priors. You should have $3 * 5 * 11 = 165$ values in your results. Make it clear whether you are reporting the positions using the (row,column) convention or the (x,y) convention for coordinates. Notice that you don't have to re-calculate the likelihoods when you use a different prior.

8. As an optional challenge, try to produce a two-dimensional map which shows the accuracy for every position in the image when that position is used as the pixel value. The code for this is easy to write, but the problem is in making the code fast enough to finish.

What to turn in

- Three matlab files with the following names: likeFromTraining.m, BayesRule.m, classifyTestData.m
- Whatever images you would like us to have, like the map of accuracies.
- a pdf file called nameLASTNAME.pdf with all the code and with everything else you have to report, like accuracies etc. The TA will print this and give you feedback directly on this sheet.