

MATLAB exercise

Detection

Part 1, Radar Detection.

Consider a radar detection system, where we want to make a determination if a target is present or absent. If the target is present, $Y = A + X$, where A is a known constant. If the target is not present, $Y = X$. X is a zero mean Gaussian with variance σ^2 . The a priori probability that the target is not present is .8. The signal to noise ratio is the ratio of A to σ^2 , you must choose values for this exercise that give insightful results.

- Derive and implement in MATLAB the MAP rule for detecting the target. Run 1000 iterations of your detector; compare the probability of error with the theoretical probability of error.
- Implement a simulation that plots the receiver operating curve for this detector. Plot the receiver operating curve for several signal to noise ratios.
- Assume that missing the target is 10 times worse than falsely detecting the target. What is the decision rule that minimizes the conditional risk? Mark this point on your receiver operating curve for at least one SNR value.
- Using the cost structure in part c), Select one SNR value and plot the value of the expected cost for a range of a priori target present probabilities from 0 to 1.
- Now, repeat parts a and b, but change the model such that the target present remains $Y = A + X$ but the target not present model is now $Y = A + Z$ where Z is a zero mean Gaussian random variable with $\sigma_z^2 > \sigma^2$. Plot a few receiver operating curves for different ratios of σ_z^2 to σ^2 .

Part 2, Introduction to pattern classification and machine learning.

This exercise combines ML estimation from the previous assignment, and detection theory. A common use of detection techniques is to classify objects into categories based on observations of certain features. In this exercise, you will build a maximum a posteriori classifier to classify a type of Iris plant based on a classic pattern recognition from the UCI data repository [1]. This data set has 4 features, and 3 classes. The features are sepal length/width and petal/length width. This data has been uploaded to the course website as the Iris MAT file.

A MAP classifier is an M-ary classifier, meaning it can distinguish between an arbitrary number of classes. It is closely related to the likelihood ratio test. The main difference is there are more classes, and thus more likelihoods to compute. The MAP rule simply means that the cost for selecting any correct class is 0, and the cost for any mistake is 1. When you have these conditions, you simply compute the

likelihood that a particular sample came from each class, and then pick the most likely one. This is MAP classification in a nutshell.

However, you do not know the means, variances, or anything really about the data. This is where machine learning comes in. You have to figure out the parameters from the data. The simplest model is Gaussian. Since you have 4 features, you'll need to compute the pdf of a 4 dimensional Gaussian, for each class in the data set. How do you do this? Recall, a multivariate Gaussian is completely specified by its mean vector and covariance matrix. Also remember that as we showed in class, the max-likelihood estimates for the means and variance of Gaussians are just the sample mean vector and covariance matrices. With your estimates of the mean and covariance, you can compute the likelihood of any sample using MVNPDF.

Now, one last thing, it is cheating to estimate the mean and covariance from all the data, and then use that same data to test your classifier. So, divide the data, at random, into 2 halves. One is for training and the other is for testing. In this case, training your classifier simply means estimating the mean and covariance matrix of the Gaussian random variables.

Evaluate your classifier using the test data, compute a total probability of error, and plot a confusion matrix [2]. If you want to experiment with some other simple things, try making a Naïve Bayesian classifier, or see if you can reduce the dimensionality of the dataset and still get good results using.

1. <http://archive.ics.uci.edu/ml/datasets/Iris>
2. http://en.wikipedia.org/wiki/Confusion_matrix