ECE662 Spring 2010
Statistical Pattern Recognition and
Decision Making Processes

Homework Assignment 2
4/21/2010

Abstract

The objective of this assignment is to intuitively get a sense for the effect of Maximum Likelihood Estimation of parameters (MLE) on Bayesian decision making processes. In all experiments, the following occurs:

1. Two sets of Normally distributed, synthetic data are generated for two classes
2. A MLE estimate is calculated on a number of samples (The "Training Samples") for each of the two classes.
3. The remaining synthetic data is classified according to a generalized Discriminant, formed by the MLE estimate of each class.
4. An error rate is determined and plotted.

Two experiments are considered. In the first, the number of training samples used to estimate the original density is fixed, and the distance between the mean of each normal distribution is varied. In the second experiment, the distance between distribution means is fixed, and the number of training samples is varied. In both experiments, the number of features is varied as well. For simplicity, every random variable is independent and identically distributed, with equal priors.

Experiment One

For this experiment, data is synthetically generated for two classes, and a decision hypersurface is used according to a generalized discriminant of the form:

$$g(\vec{x}) = -\frac{1}{2}\left(\vec{x} - \vec{\mu}_{i,MLE}\right)^T \sum_i{}^{-1}\left(\vec{x} - \vec{\mu}_{i,MLE}\right) + \ln(P(\omega_i)) - \frac{1}{2}\ln|\sum_i|$$

Class 1 is Normal, with the mean in every dimension equal to zero, and sigma equal to the identity matrix. That is, every feature is independent, with equal scatter. Similarly, class 1 is Normal with the mean in every dimension equal to the same value (The x axis as it varies), and sigma is also equal to the identity matrix. The experiment is run several times, in N=2, 8, 14, and 20 dimensions. Mean values for Class 2 are varied from .125 to 5. An example distribution of the two classes is shown in Figure 1.

From Figure 1, it is clear that a good separation is not possible, which is reflected in the experiment results. At low separations, the classifier achieves near 50% accuracy, which is no better than blindly guessing. As the means drift apart, much higher accuracies are achieved.

Figure 2 shows the results of several runs of the test, with the MLE parameters using 100 training samples. In the figure, the shaded regions correspond to actual error rates of the classifier for a given set MLE parameters used in the classifier. (For Normal distributions, the MLE method determines the mean and standard deviation.) However, the shaded region is constructed by selecting only 2 sets of MLE parameters, chosen using the following formulas over 10,000 iterations:

$$\max(norm(\mu_{1,MLE} - \mu_{2,MLE}))$$
$$\min(norm(\mu_{1,MLE} - \mu_{2,MLE}))$$

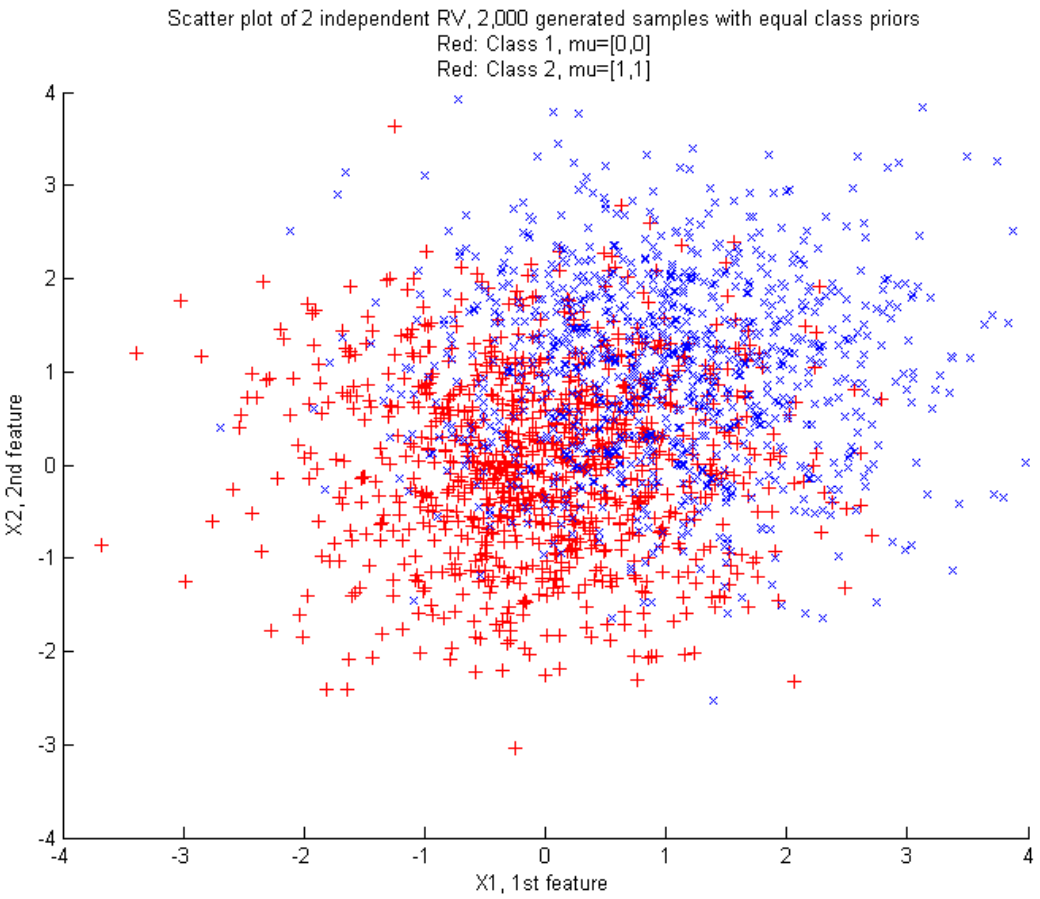

Figure 1. Example plot of samples generated from class 1 and 2, N=2 Dimensions

This method of selection for the MLE values does not guarantee a good upper or lower bound (especially since the min and max values are stochastic already), but it gives a good approximation of the range of possible MLE parameter values.

In Figure 2, the dashed line of the same color corresponds to the error rate of the same classifier with a perfect, non-MLE parameter estimate. In the 2 Dimensional case, the error rate remains relatively close to that of an ideal Bayesian classifier, but higher dimensions suffer from a significantly increased error rate due to the MLE method.

Note also that increasing the number of independent features adds more information to each decision, but adding a new feature to a high-dimensional feature space has a smaller effect than adding a dimension to a lower-dimensional feature space. That is, the accuracy improvement from 2 to 8 dimensions is much higher than from 8 to 14.

However, it is also important to note that the higher dimensional feature spaces converge to near-100% accuracy at a much lower mean-separation than the 2-dimensional case.

Figure 3 shows a second run of the test, which is identical to the first run, except that the number of samples used to calculate the MLE parameters is fixed at 50, instead of 100. In the figure, one can see

graphically that the vertical range of each shaded region is greater (for the most part). This makes sense, since each set of MLE parameters uses less training data. One can also see that the "lower bound" follows the ideal, non-MLE case almost as closely in the 2-Dimensional case. Unfortunately, in the higher-dimensional cases (specifically N=20) the error rate's pseudo-lower bound is significantly higher than in the 100-training-sample version in Figure 2. One can conclude that a poorly-constructed set of MLE parameters will hamper a higher dimensional classifier more than a 2-dimensional classifier.

Figure 3 also shows that clearly, with 20-dimensional case rising above the 14-dimensional case error rates, the min/max method used to select the so-called upper and lower bound values is somewhat flawed. Again, notice that every value chosen for these "bounds" is a valid MLE-determined estimate. This method gives consistent results in a much lower computational cost.

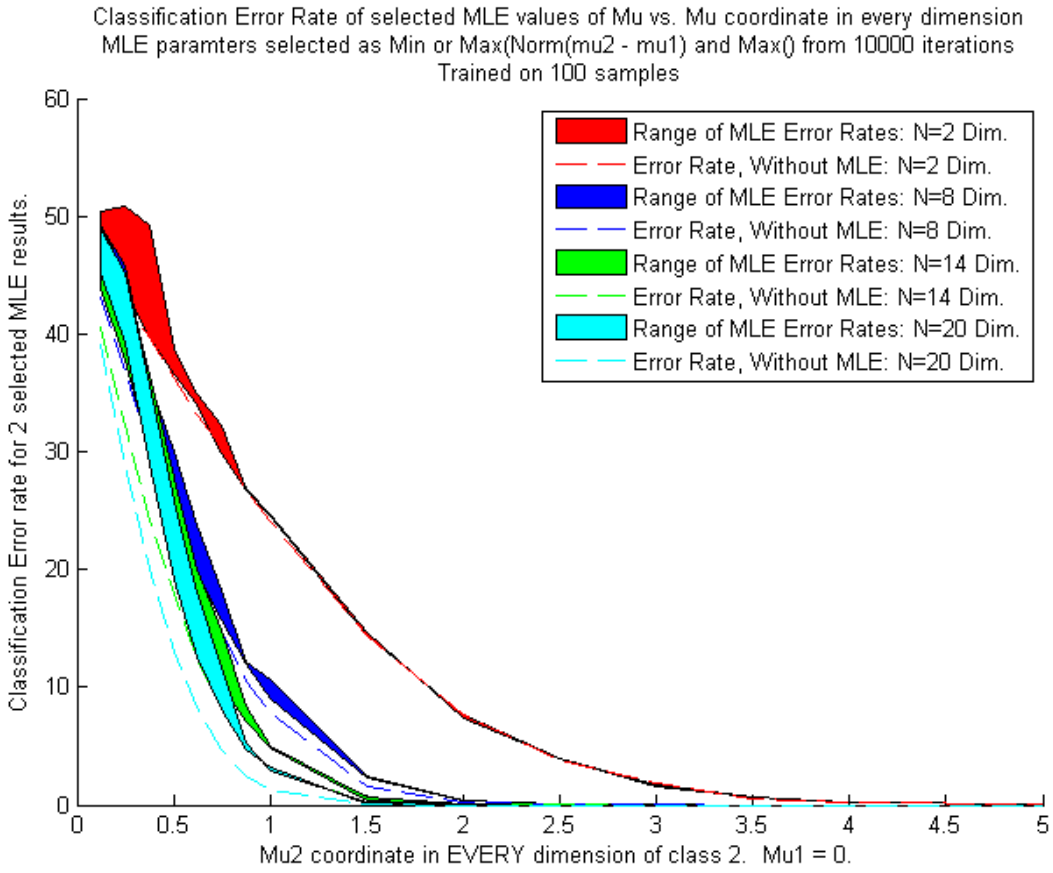

Figure 2. Several runs of experiment 1 in increasing dimensions, using 100 training samples

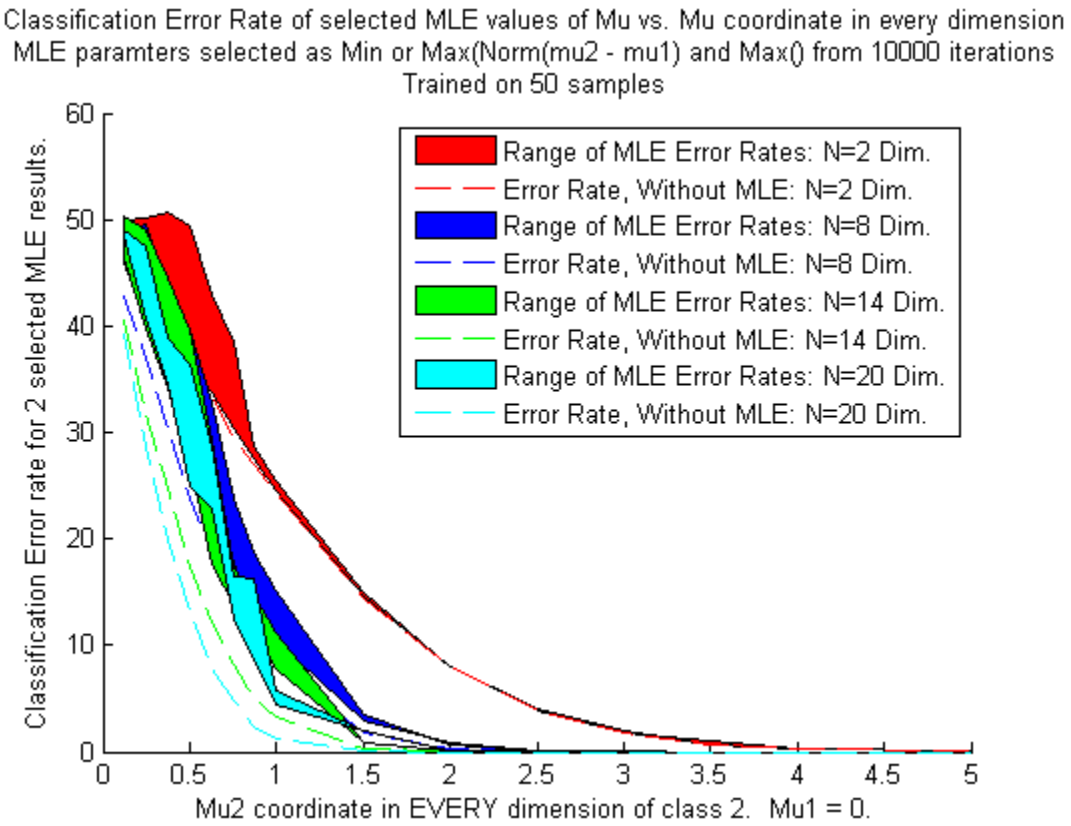

Figure 3. Several runs of experiment 1 in increasing dimensions, using 50 training samples

Experiment Two, Part A

In this experiment, the number of samples required to achieve a good MLE estimate is investigated. Instead of varying the mean separation of Class 1 and Class 2, the mean of Class 2 is fixed while the number of samples used to calculate the MLE parameters is varied. Otherwise, the method (and literally the MATLAB code) is almost identical.

Figure 4 shows several runs of the experiment, again using the same increasing dimensions of the feature space. Realizing this, one can see that the points in Figures 2 and 3 where Mu2=1 correspond directly to the points in Figure 4 where the No. of Training Samples is 100 and 50, respectively.

Increasing the number of training samples above approximately 100 will not, in this experiment, greatly affect the accuracy of a MLE parameter estimate. That is, using 100 training samples in an MLE classifier will probably give reasonable success in $N \leq 20$ dimensions. Also note that in lower numbers of training samples, the classifier resulting from these estimates is much less likely to reach the ideal, non-MLE error rate (the dashed line). In N=2 dimensional feature space, this is close to being at least possible (if not likely), but in higher dimensions with few training samples, the MLE estimate is nowhere near as good as the non-MLE error rate.

The biggest take-away here is that higher-dimensional classifiers require more training samples to reach a pseudo-steady state. In the N=2 dimensional case, only about 100 samples are required to get a relatively good classifier, and as low as 60 may work.

A final note on the high-dimensional feature space cases: while 100 training samples will give a reasonable classifier, a much greater number of training samples is required to give near-ideal results. When the last 0.1% of accuracy is required, more training samples may do the trick. The N=8 Dimensional case needed more than 400 training samples to (graphically) reach the non-MLE error rate. In no case, however, will increasing the number of training samples give results better than an Ideal Bayesian classifier.

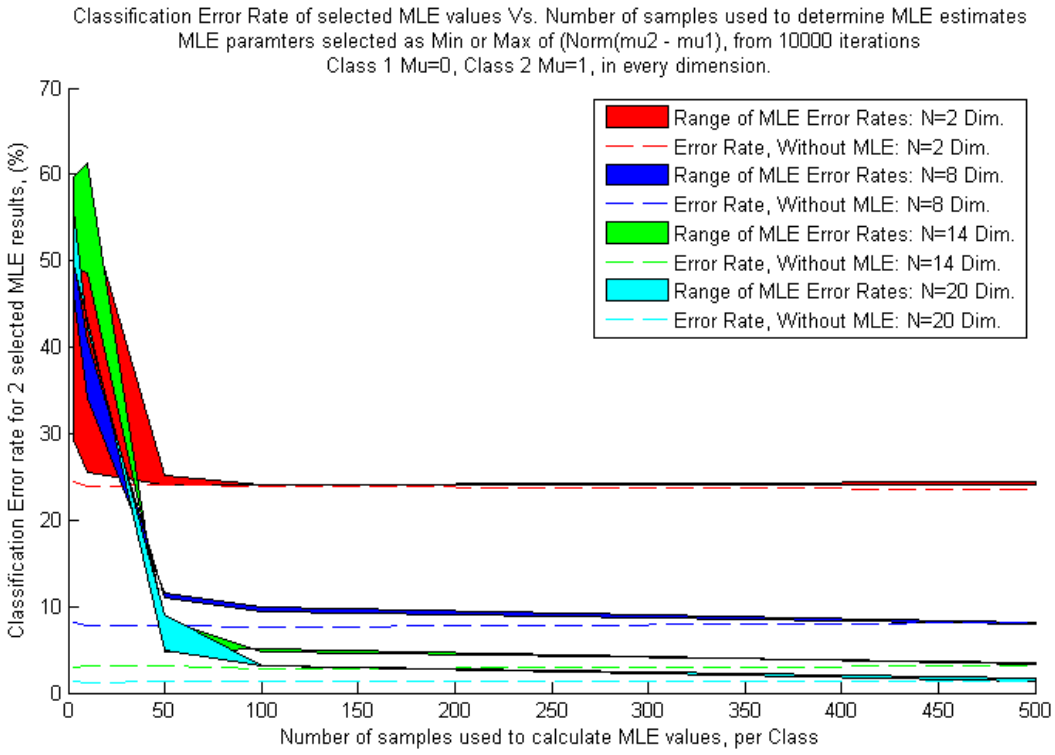

Figure 4. Several runs of experiment 2 in increasing dimensions, with Class 2 Mean=1

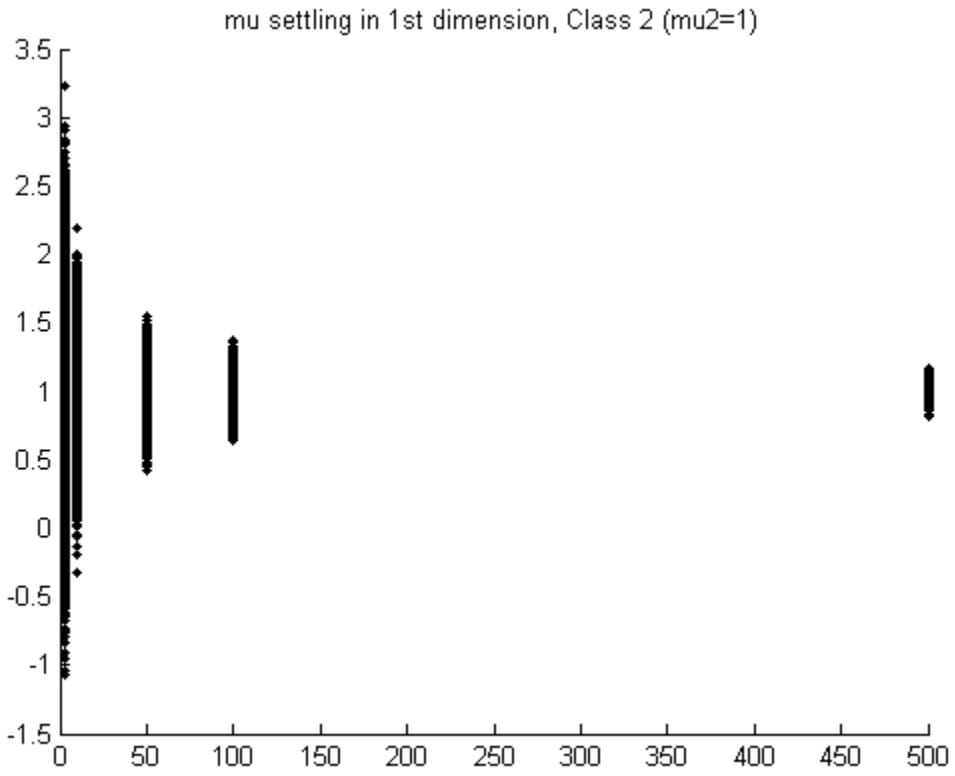

Figure 5. Scatter plot of the calculated Mean for class 2 in every MLE result, N=2 Dimensions

Experiment Two, Part B

In each run of Experiment 2, 10,000 sets of MLE parameters are calculated on a given number of training samples. Although only 2 of these sets are used in Figure 4, Figure 5 shows a scatter plot of every single MLE value used for N=2 dimensions.

Clearly, an abrupt settling of the estimate occurs when the number of training samples used in MLE increases from 3 to 10 samples, and again from 10 to 50 samples. The odds of generating a horribly inaccurate MLE parameter are very low after about 100 samples, but do not reach the ideal, non-MLE case consistently even after 500 training samples are used. In a real world application, more valid training data is always a good thing.

To reinforce this, Figure 6 shows the corresponding Covariance values which were calculated using MLE. The Covariance matrix is 2x2 in N=2 Dimensions, so the determinant was used a measure of accuracy to the ideal case, where determinant($\sum$)=1. Again, one can see an abrupt decrease in the number of outliers when moving from 3 training samples to 10 samples, and again from 10 to 50. And even after 500 samples are used, one still don't appear to see 99% accuracy.

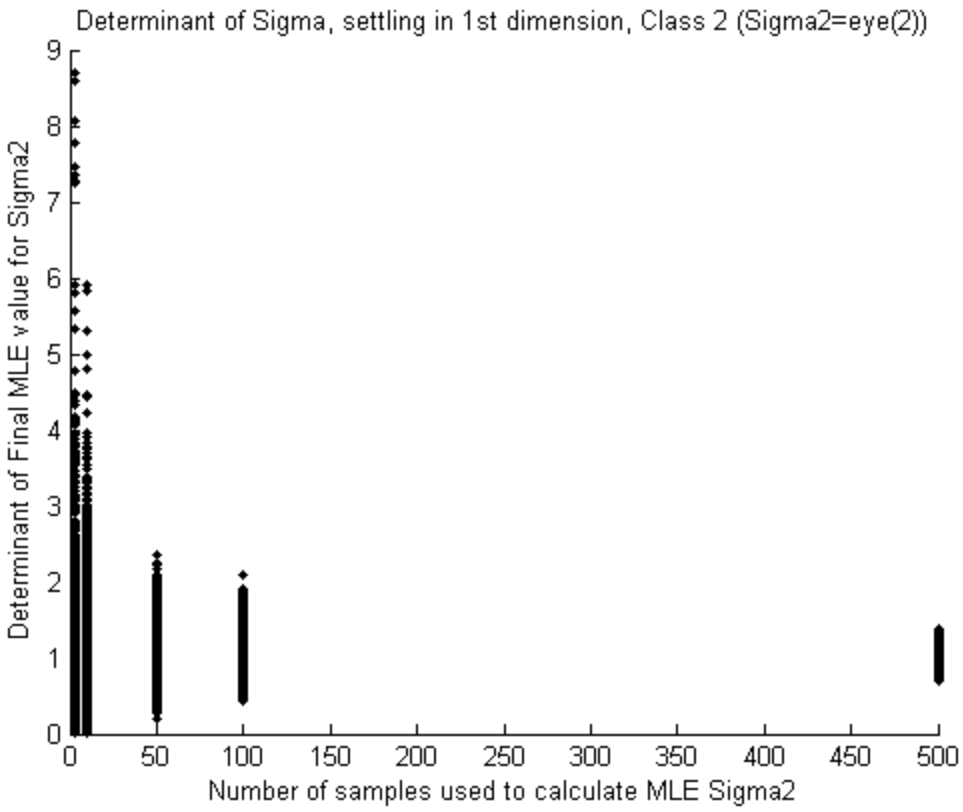

Figure 6. Scatter plot of the determinant of the calculated Variance for class 2, using every MLE result, N=2 Dimensions

Conclusion

In General, using Maximum Likelihood Estimation of parameters to classify Normally-distributed data can give results very close to Bayes' Rule. In no case did the MLE estimate perform better than Bayes', but a minimum number of training samples is required to guarantee good results. That number depends on the number of dimensions, as well as the original separation of the data. Using a minimum of 100 training samples for $N \leq 20$ independent features will give relatively good results for the experiment above, where μ_2-μ_1=1, σ^2=1.

Further Study

Given more time, plotting every single error rate for every single calculated MLE parameter would be ideal, and give more consistent results. In addition, varying the scatter (Variance) of the data, as well as using unequal prior probabilities, may give a more complete picture. Lastly, an analysis of using MLE on the wrong type of distribution (i.e. guessing Normal on a Uniform distribution) would give a comprehensive discussion, and would introduce the elephant in the room: an analysis of using Bayesian Parameter Estimation, which does not apply well for synthetic data.