Diabetic Testing Classification Utilizing PCA and FLD Dimensionality Reduction and Performance Evaluation

Timothy Ryan Lovvorn

March 12, 2017
ECE 471 Pattern Recognition
tlovvorn@vols.utk.edu

Abstract

Gathered data sets have a degree of uncertainty regardless of gathering methods, and as a result, gathered data must be evaluated within given confidence intervals. Doing so is possible using both Baysian based parametric decision rules, as well as nonparametric based methods in the kth nearest neighbor method. This report designs decision rules based upon these methods utilizing the raw, normalized data, as well as Utilizing the dimensionality reduction methods of FLD and PCA to determine to what degree they affect the overall accuracy of data. All data sets are evaluated using the 3 case method based upon the Gaussian PDF as well as the kNN method generated from the training data and utilizing on the testing data to determine class. The performance of the decision methods is determined by their average effectiveness With both equal and dynamic prior probabilities as well as variant k values for the kth nearest neighbor method. While a high degree of variance occurred, all methods were within approximately a range of 65 to 90% accuracy.

1. Introduction

Millions, if not billions, of individuals annually visit a healthcare provider for routine or serious testing for everything from mild such as blood type and pregnancy testing, to the severe such as genetic defects and serious illness screening. As the result of such testing can be extremely relevant to lifestyle changes, a diabetic must now be always conscientious of their glucose levels, it is rather critical our testing methods be extremely accurate. In order to do so it is necessary we are both able to create methods by which we may answer a question, typically a binary decision rule of negative or positive, as well as have the ability to quantify the effectiveness of such a rule. It is also quite common we may have a tremendous amount of gathered data by which we wish to make our decision rule, therefore it would be prudent if we had a method by which we determine how "useful" each feature of said data sets actually are. We perform these calculations within the report using pattern classification and performance evaluation.

Pattern recognition is extremely useful in many real world applications, as it allows us to make classifications functions useful for a variety of classification and prediction purposes. Analysis of patterns in seismic activity may be used to predict possible future volcanic or tectonic activity. Analysis of DNA sequences may be utilized to determine heredity traits and possible identify unique genetic defects or signatures. Deep pixel analysis of imagery may be utilized for image recognition software in a variety of consumer, production, and military applications. Through the power of these classifiers, questions which, while trivial to humans, are very hard to quantify to computation machines such as, "what type of expression does the person in this image have?" may be reduced to mere arithmetic computations and probabilistic comparisons.[2]

Through pattern classification we are able to take the raw data we have gathered concerning certain qualities, or features, of a given set of data, and use such data to determine a generalized classification method for such data. In addition, we my reduce the overall number of features within our initial data set to ease data aggregation and sifting issues. After all data sets are initialized, our different classification methods may be performed upon the data set to determine the classification of the data within our testing set. This data may then be compared to known values of class for such data, as all our learning in this except PCA reduction, is supervised learning, to determine the overall effectiveness of our classification functions.[3]

All of our recognition work is based upon Bayes Decision Theory and the Bayesian formula for determining posteriori probability. Originally determined by the 18th century Rev. Thomas Bayes and later refined, Bayes formula allows us to determine the posteriori probability based upon the probability density function of their data distribution, we utilized Gaussian, the prior probability, and some known evidence about the data set. The formula is represented as the following:

$$P(w_i|x) = \frac{P(x|w_i)P(w_i)}{P(x)}$$

Where $P(w_i|x)$ is the posteriori probability, $P(x|w_i)$ is the probability density function, $P(w_i)$ is the prior probability, and $P(x)$ is the evidence for the variable x. [3]

We utilize this formula, in addition to the Gaussian Distribution, to create our maximum likelihood estimation classier as well as our kth Nearest Neighbor classier. In addition, our data is reduced using Principal Component Analysis and Fisher's Linear Determinant to map the data into a lower dimension to allow us to determine how such reduction affects the overall reliability of our classifier function. With the maximum likelihood method we use the maximum ratio to determine the class of the data based upon 3 cases making varying assumptions about the variance of the data. Kth Nearest Neighbor method utilizes varying k values between a certain range to determine the class based upon the kth number of nearest neighbors within the given range. The overall effectiveness of all methods is evaluated based on the correct number of classifiers with discriminant functions are able to determine for testing data sets, vs the overall amount of data. All this is achieved through a C++ program. The source code, relevant files, and an README for usage of said program is appended to this report.

2. Technical Approach

2.1 Maximum Likelihood Parameters

To determine the maximum likelihood of an given sample, it was first necessary to determine the covariance and mean for a given value. As I want the maximized values for these parameters, this is performed by determining at what point the Gaussian pdf derivate is equal to zero for each possible variable. This gives the following two equations:

$$\overline{\mu} = \frac{1}{n}\sum_{i=1}^{n}\overline{x} \qquad \overline{\Sigma} = \frac{1}{n}\sum_{i=i}^{n}(\overline{x}-\overline{\mu})(\overline{x}-\overline{\mu})^T$$

2.2 Maximum Likelihood Determination

As aforementioned, Baysian Theory is utilized to create our determinate function based upon the maximum likelihood ratio. For this method we may discount the evidence, P(x), as it will be identical for all cases and therefore only a scaling factor, leaving us with the following:

$$P(w_i|x) = P(x|w_i)P(w_i)$$

As we have two classes in our case, 0 for 'no' and 1 for 'yes' we have two possible values of w. From this we determine the value with a greater product of both it's prior probability, and probability of x occurring as a case of w_i, assuming a Gaussian distribution, we can create the following assumption:

$$if(P(x|w_0)P(w_0) > P(x|w_1)P(w_1))\ then\ case\ 0\ else\ case\ 1$$

Which we formalize as:

$$\varepsilon(x) = \begin{cases} \text{if} \quad P(x|w_0)P(w_0) > P(x|w_0)P(w_0)\ \text{case 0} \\ \text{else case 1} \end{cases}$$

This is the formula by which we will determine cases for all data points. As previously stated, this will be done through a multivariate gaussian equation to determine the $P(x|w_i)$ for each data set as follows:

$$P(x|w_i) = \frac{1}{2\pi|\overline{\Sigma}|^{\frac{d}{2}}} \exp(-\frac{1}{2}(\overline{x}-\overline{\mu})^T\overline{\Sigma}^{-1}(\overline{x}-\overline{\mu}))$$

This will be further broken into 3 specific cases based upon our assumptions about data distribution:

Case I: All features are statistically independent, meaning they have the same variance represented as □ = □^{2}**I**. Our value may be determined using the euclidean distance between each point as in the following:

$$-(\overline{x}-\overline{\mu})(\overline{x}-\overline{\mu})^T + \ln(P(w_i))$$

Case II: Covariance between all matrices for all classes is identical, but not the product of a scalar to the identity matrix. This gives the following:

$$-\frac{1}{2}(\overline{x}-\overline{\mu})\Sigma^{-1}(\overline{x}-\overline{\mu})^T + \ln(P(w_i))$$

Case III*:* The covariance for each class is different. Giving the following:

$$-\frac{1}{2}(\overline{x}-\overline{\mu})\Sigma^{-1}(\overline{x}-\overline{\mu})^T - \frac{1}{2}\ln(|\Sigma|) + \ln(P(w_i))$$

2.3 Kth Nearest Neighbor Determination

Taking Bayes Formula, we can now consider the following. Probability of x occurring in w_i can be thought of as the ratio of occurences of the w_i in the overall sample in a given volume of space. Prior probability is the number of occurences of w_i in a given data set, and evidence is the ratio of neighbors used versus the total number of neighbors over the entire volume of space. Thus the following:

$$P(x|w_i) = \frac{\frac{k_m}{n_m}}{V} \qquad P(w_i) = \frac{n_m}{n} \qquad P(x) = \frac{\frac{k}{n}}{v}$$

By applying this to Bayes formula, we generate the following:

$$P(w_i|x) = \frac{k_m}{k}$$

This works when we consider the prior probability as already determined. For later use in the dynamic data section, I will consider this formulas without the $P(w_i)$ section and multiply by the prior probability to note the changes dependent on how the prior probability changes.

2.4 Feature Reduction

For Fisher's Linear Determinant, in order to map each variable in our data set to a lower dimension, it was necessary to maximize the distance projected means and variance, generalized in the Jacobi equation:

$$J(\overline{w}) = \frac{\overline{w}^T S_B \overline{w}^T}{\overline{w}^T S_w \overline{w}^T}$$

Which we reduce to the following:

$$\overline{w} = {S_w}^{-1}(\overline{m_1} - \overline{m_2})$$

Where S_w is the summation of the switch matrices:

$$S_w = S_0 + S_1$$

For Principal Component Analysis, we perform an eigen value decomposition upon the entire data set as we are trying to reduce our set by d >> m where m is a dimension far reduced from d while still within our acceptable error range of .10:

$$\varepsilon^2(m) = \sum_{i=m+1}^{d} \lambda_i \qquad \text{Where} \qquad \frac{\sum_{i=m+1}^{d} \lambda_i}{\sum_{i=1}^{d} \lambda_i} < .10$$

Our eigen value decomposition yielded the following eigen values:

Eigen Value	Eigen Vector
2108.81	[0.0109922,0.972629,0.117432,0.161946,-0.0986168,0.063696,-0.00852519]
18.1552	[-0.000588058,0.0220603,-0.0403138,-0.0984647,0.0577461,-0.055302,-0.990858]
5.87762	[0.00127281,0.0858758,-0.154931,-0.791794,-0.572707,0.106902,0.0475545]
1.5074	[-0.0025056,0.0503225,-0.366595,-0.109408,0.412416,0.82522,0.00488703]
0.850508	[-0.00956907,0.0766225,-0.906705,0.240747,-0.131902,-0.309801,0.0242814]
0.639956	[-0.999888,0.0103203,0.0105866,-0.00224104,-0.000234755,0.000854614,0.000553784]
0.162881	[0.00201471,0.193962,-0.0629528,-0.51692,0.686631,-0.452207,0.123501]

Table 1: Eigen value decomposition of Training Data

Based upon this, I concluded I was able to remove the bottom 6 dimensions, and only suffer an error of approximately 1.3%, well within my desired range of 10%

2.5 Normalization and Data Mapping

Because our data features each varied greatly in their ranges, anywhere between 0 to 1 or 1 to 1,000, it was prudent to first normalize all the data sets. This was done by determining the mean and variance of each data set using the following Gaussian Equation for each jth feature:

$$\mu_j = \frac{1}{n}\sum_{i=0}^{n} x_j \qquad \sigma_j^2 = \frac{1}{n-1}\sum_{i=0}^{n}(\mu_j - x_j)$$

Then each data point x was normalized as follows:

$$x' = \frac{(x - \mu_j)}{\sigma_j}$$

Finally, for our PCA and FLD reductions, the following method was used to map our x data points to the lower dimension using the derived w and e vectors.

$$FLD = \overline{w}^T\overline{x} \qquad PCA = \overline{e}^T\overline{x}$$

3. Experiments and Results

3.1 The Data Sets

Both the training set, pima.tr, and the testing set, pima.te utilized were taken from the Cambridge University website provided for this project [1]. Both data sets contained lits of 7 features of given patients where were, the yes or 1 class, or were not, the no or 0 class, classified as diabetic. The training set contained 200 samples while the testing set contained 332 samples. These samples represented an excellent example of the practical applications of our given discriminant function, as they are hyperdimensional sets which defy the conventional ability to plot in a 2 or 3 dimensional plane. The training set was utilized to create our classifiers for our testing set.

3.2 Equal Prior Probability

3.2.1 Likelihood Ratio Estimation

For determining our values based upon the discrimination function we utilized our classifier derived from the training set on the normalized training set, the testing set, and PCA reduction, and the FLD reduction. For each category, case 1, case 2, and case 3 was performed. By diving the number of correctly determined cases, by the total number of samples, accuracy is able to be determined. All of the data for such comparison is given in Table 1 below:

Data Type	Case	Accuracy(%)
Normalized Training Data	Case 1	61.5
	Case 2	75.5
	Case 3	79
Normalized Testing Data	Case 1	62.05
	Case 2	76.81
	Case 3	74.1
Training PCA Data	Case 1	61
	Case 2	61
	Case 3	63.5
Testing PCA Data	Case 1	61.75

Case 2	61.75	
Case 3	67.77	
Training FLD Data	Case 1	76
	Case 2	76
	Case 3	76
Testing FLD Data	Case 1	76.20
	Case 2	76.20
	Case 3	76.81

Table 2: Ratio Estimation Data from all cases

Despite the seemingly small loss of accuracy we determined by our PCA eigen value decomposition, there is a marked reduction in the overall accuracy in the PCA cases. This does, however, match our expectations, as PCA is most concerned with the most overall accurate representation of the data, not with best classification. Consequently, we would expect a higher of very similar rate of accuracy in the FLD reduction, as it is concerned with best classification of the data, which is indeed the case. FLD can be noted to have consistently an accuracy rate of nearly 77%, which seems to imply this is the highest possible rate of accuracy possible, which is supported by the data from our normalized testing data. In addition, from the normalized testing data we also notice it peaks in accuracy at the Case 2 scenario. This is suggestive that while there is some relationship between the features of the samples, but there are still some levels of independence as well. The data also suggest the FLD reduction eliminates the slight error introduced into the 3rd case by this relationship as it has the consistently highest accuracy rate of all distinctions. Overall, case 1 is consistently the worst performing, with Case 3 being consistently the best performing.

3.2.2 Kth Nearest Neighbor Estimation

For determining the classifier based upon the nearest neighbor I utilized a derived table of distances for each point, as well as the given classes for all values in the testing data set for comparison. This was performed with the range recommended by Dr.Qi, between 1 and 52 for each data sample of training, testing, PCA, and FLD. The distribution of their accuracy relative to their current k value, is displayed below in the following figures:

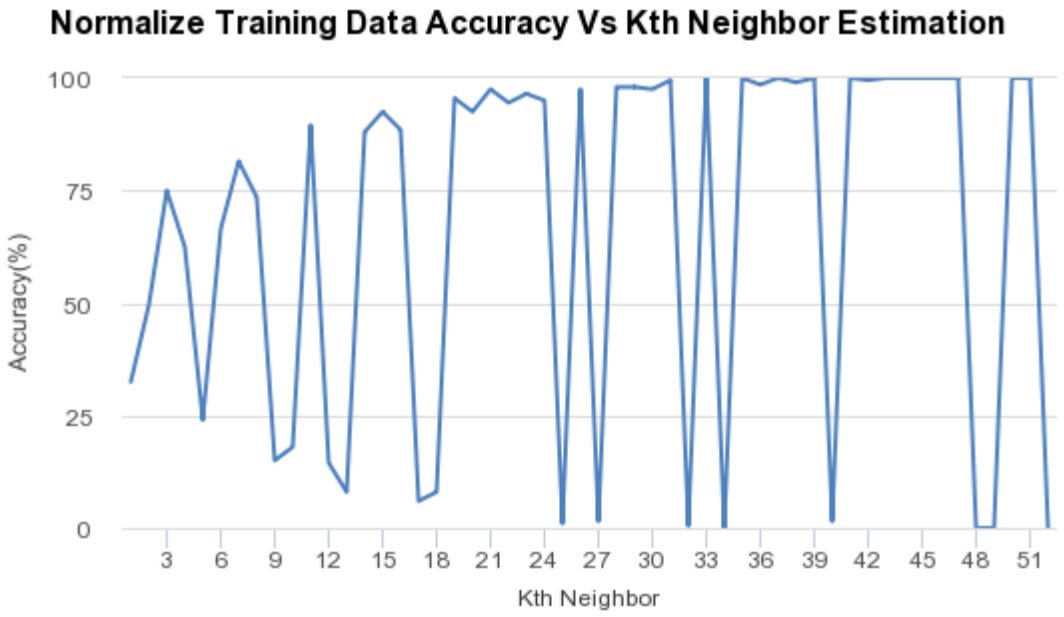

Figure 1: Normalized Training Data between 1-52 Kth Neighbor Iterations

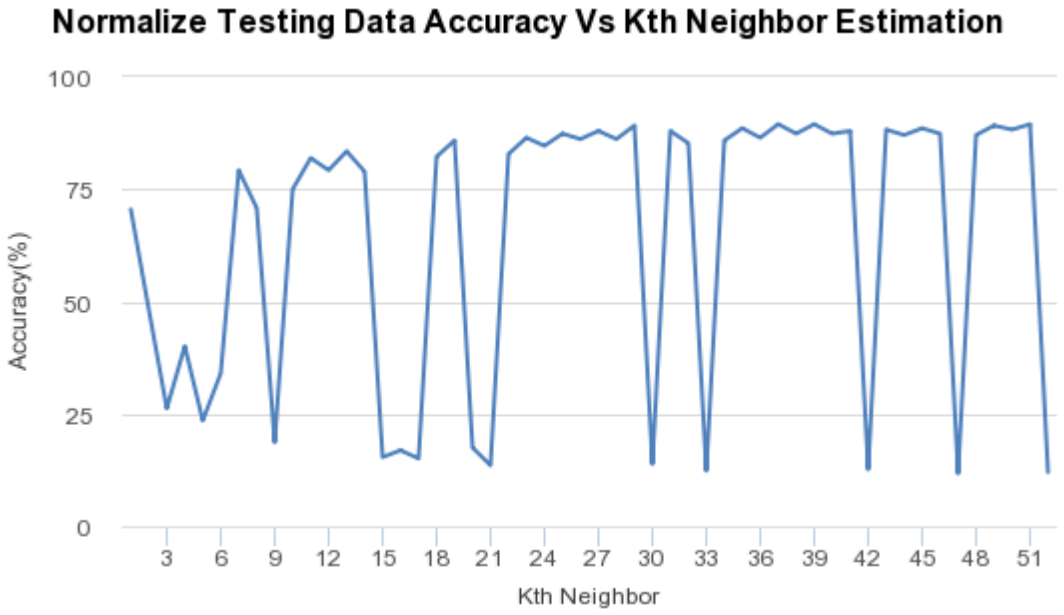

Figure 2: Normalized Testing Data between 1-52 Kth Neighbor Iterations

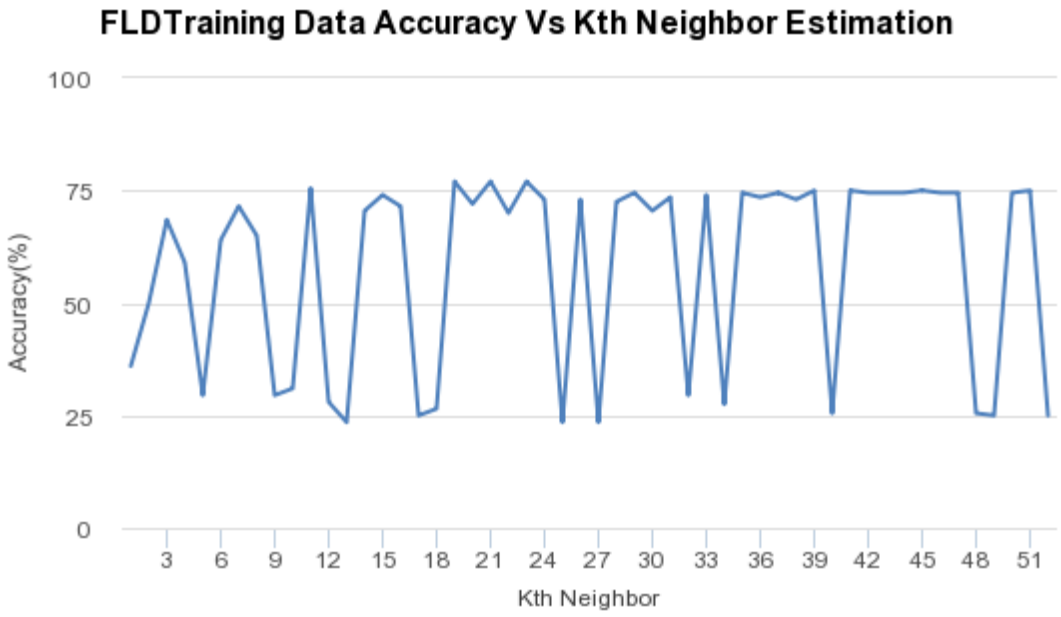

Figure 3: FLD Training Data between 1-52 Kth Neighbor Iterations

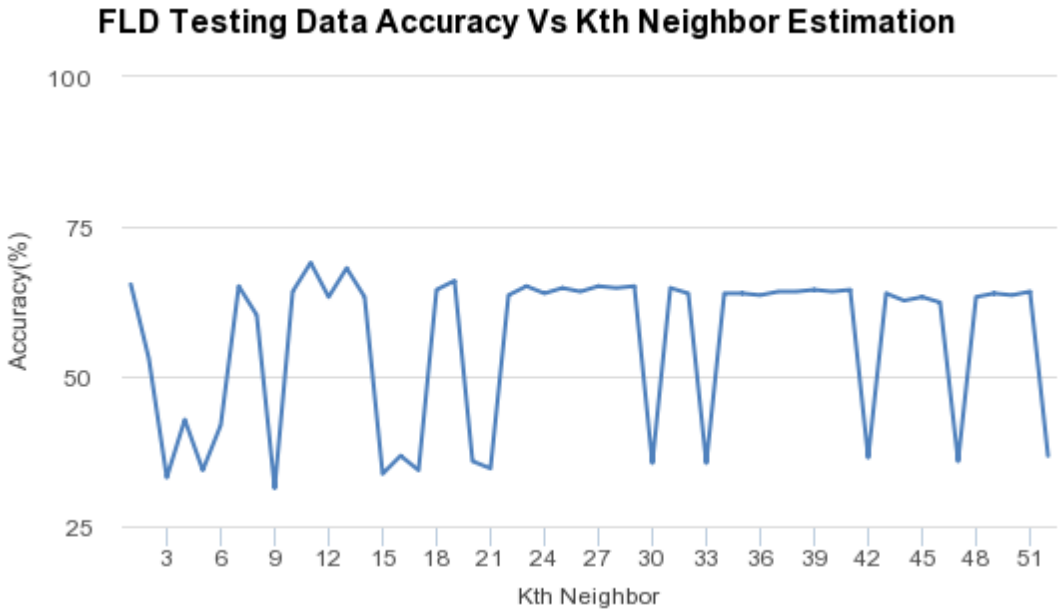

Figure 4: FLD Testing Data between 1-52 Kth Neighbor Iterations

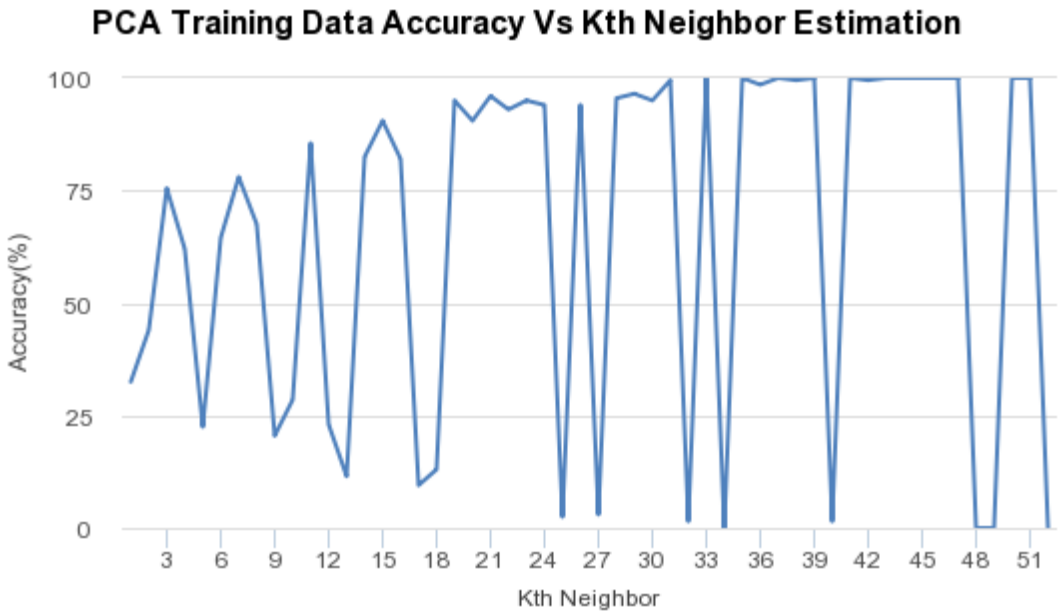

Figure 5: PCA Training Data between 1-52 Kth Neighbor Iterations

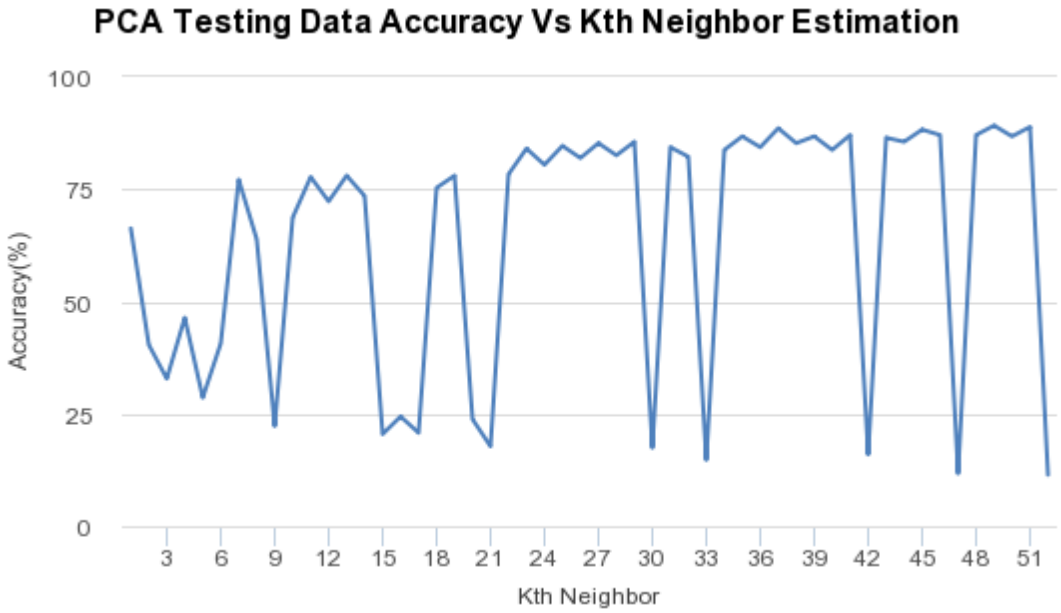

Figure 6: PCA Testing Data between 1-52 Kth Neighbor Iterations

Data Set	Data Type	Best Kth Value	Accuracy(%)
Training	Normalized	33	100
	PCA	33	100
	FLD	19	77
Testing	Normalized	37	89.46
	PCA	49	81.16
	FLD	11	68.98

Table 4: Best Estimates of Kth Nearest Neighbor For each Data set

As noted by the above Figures and Table 4, our generally our best accuracy for number of neighbors was in the mid 30's to late 40's range. Of particular interest is the notable drop off of both accuracy and k with the FLD case. While accuracy for both the PCA and Normalized cases of the Testing data were well within the 80 to 90 percent range, the FLD was signficantly lower. This is an expected result, as we know PCA strives to best represent the entire data set, it is unsupervised, while FLD seeks to only classify, it is supervised. For PCA we would expect it to improve as it utilized a larger portion of the data set as a whole, as it does with the highest value being at the 49th iteration. We also see FLD takes a very low value, again this makes sense as it is concerned with classification. The more values it gains of opposite classes, the worse it's ability to represent data will become. This would seem to indicate then most data points 11 nearest neighbors are most likely related directly to it's correct class, with a accuracy of approximately 69%. Normalized data also performed well, but this result also coincides with expectations as normalization took the entire data set into consideration for a given feature, regardless of class. As it was more directed, only considering the specific features values for a given set separate of other features in the overall data set, we would expect it to reach a lower maximum PCA. PCA is more generalized in its considerations of the data set as a whole and the relations between all features, instead of between different features subsets. This conclusion is reflected in our data set as well, shown by Table 4.

3.3 Dynamic Prior Probability

3.3.1 Likelihood Ratio Estimation

For further work on the likelihood estimation method, I took the original equation, and began to alter the prior probability for each case to determine the overall effect it would have to determine the best prior probability. I alternated the prior probabilities for each case between 0 and 1. The

results are shown in the below figure and table:

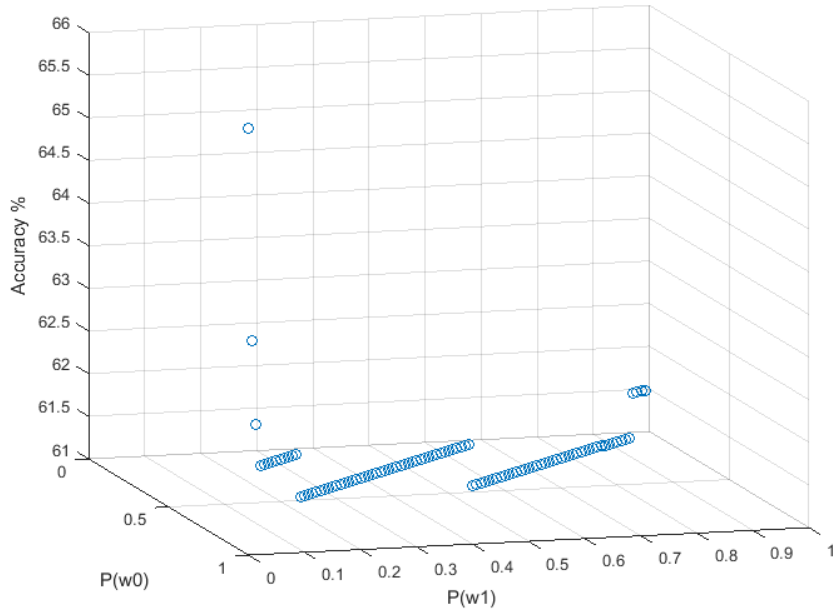

Figure 7: Normalized Training Set Case 1 with variant Prior Probability

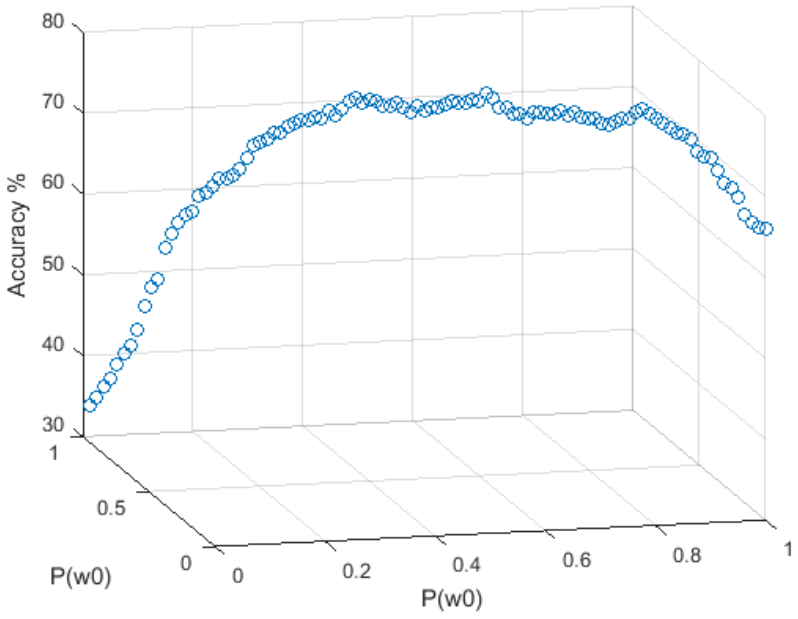

Figure 8: Normalized Training Set Case 2 with variant Prior Probability

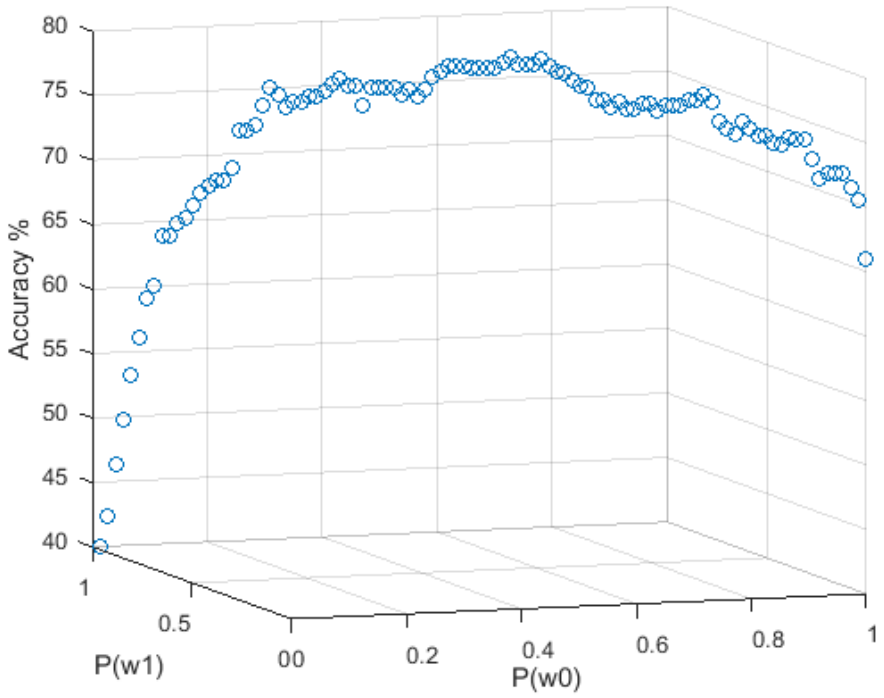

Figure 9: Normalized Training Set Case 3 with variant Prior Probability

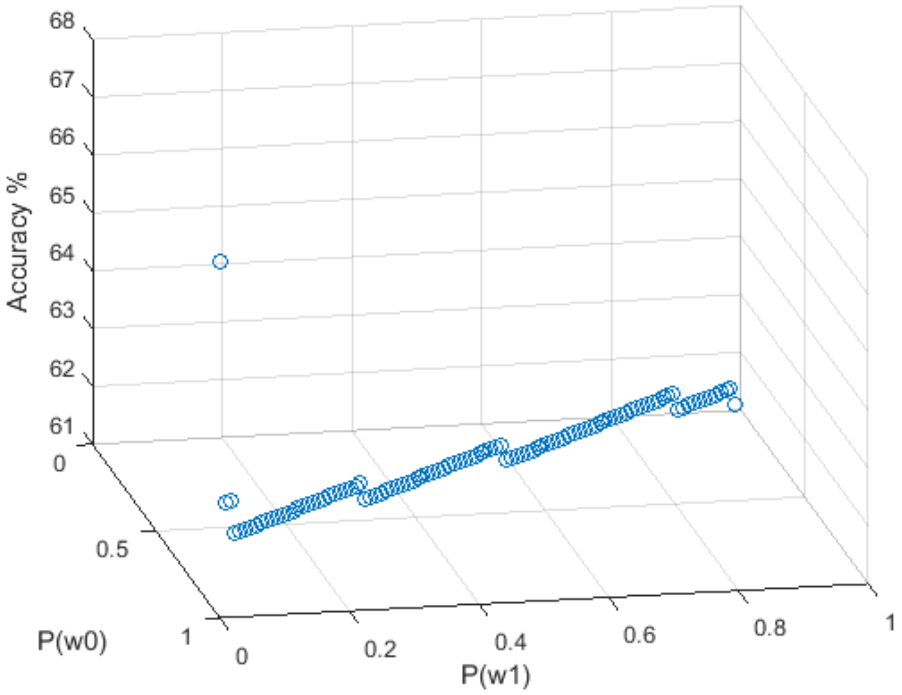

Figure 10: Normalized Training Set Case 1 with variant Prior Probability

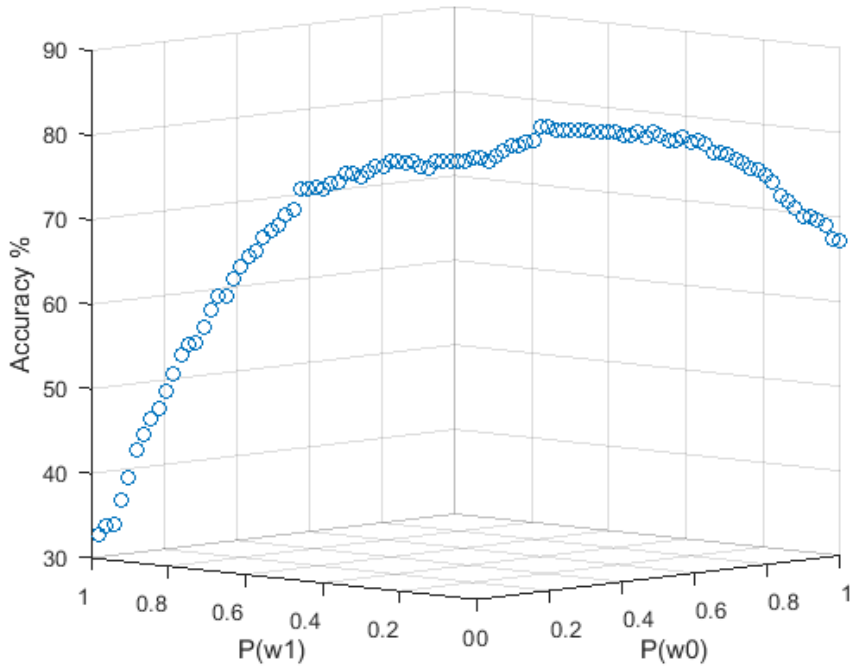

Figure 11: Normalized Testing Set Case 2 with variant Prior Probability

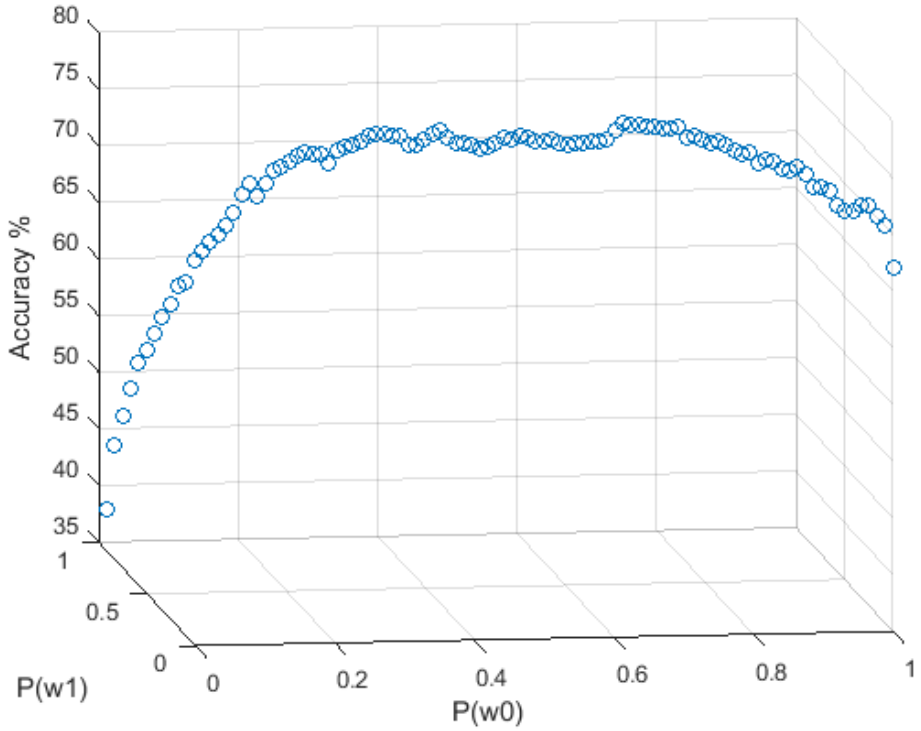

Figure 12: Normalized Testing Set Case 3 with variant Prior Probability

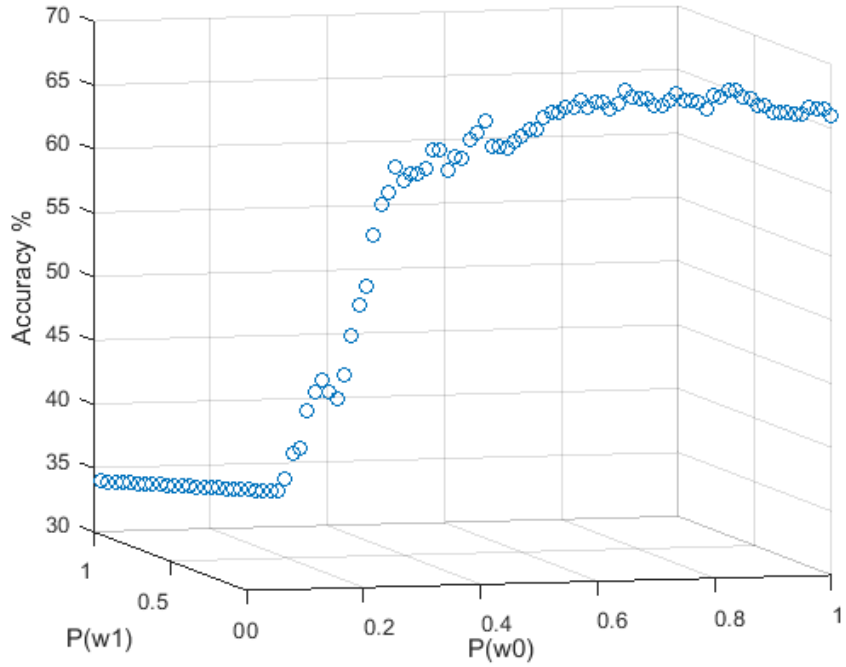

Figure 13: PCA Training Set Case 1 with variant Prior Probability

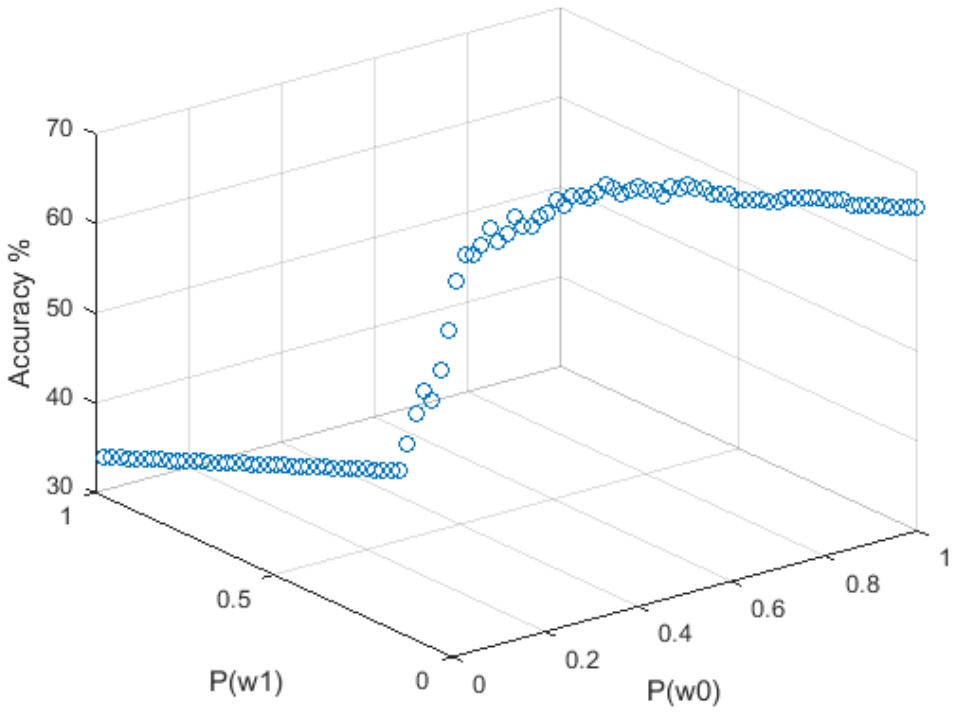

Figure 14: PCA Training Set Case 2 with variant Prior Probability

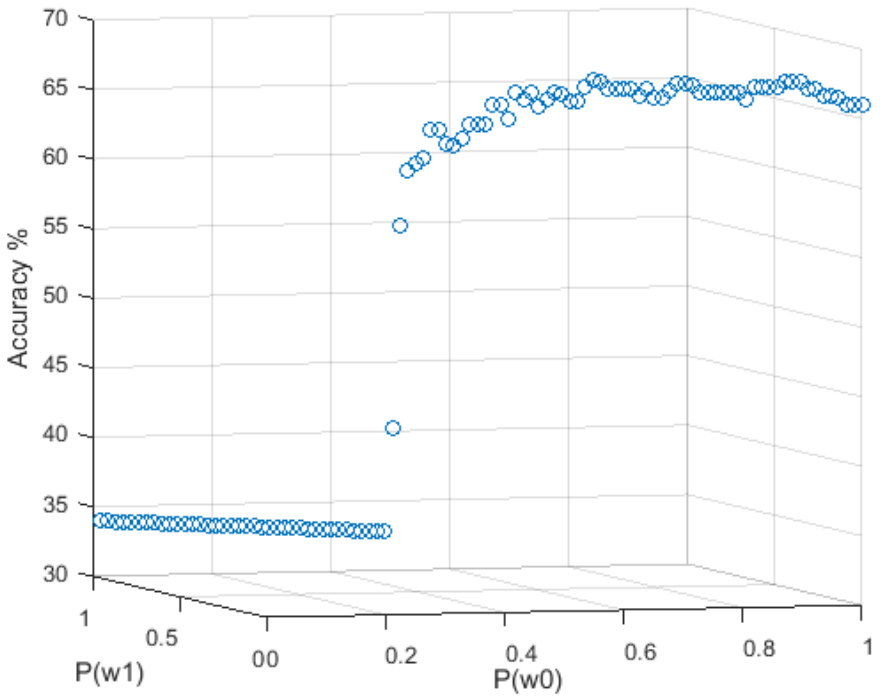

Figure 15: PCA Training Set Case 3 with variant Prior Probability

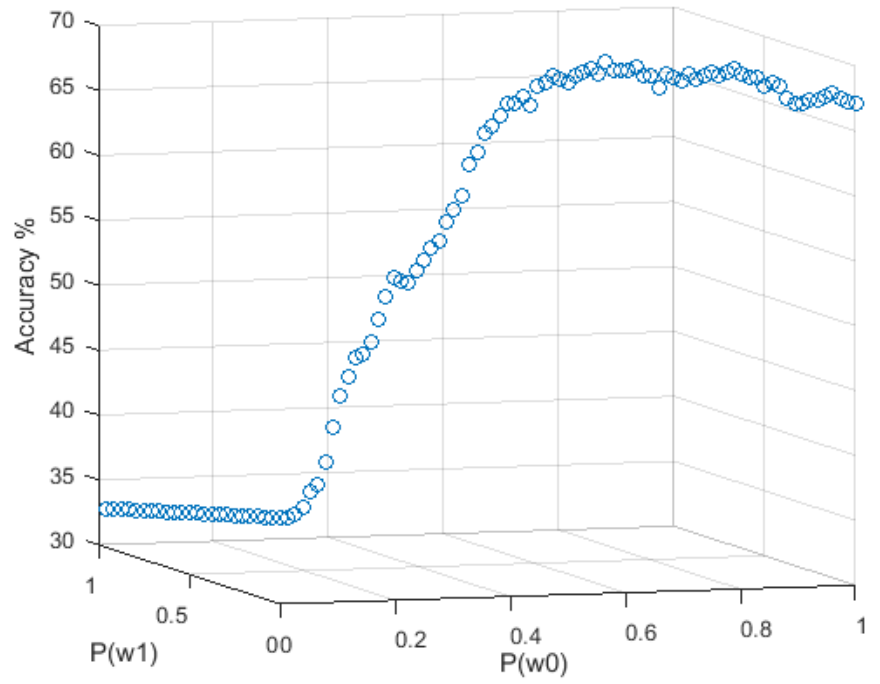

Figure 16: PCA Testing Set Case 1 with variant Prior Probability

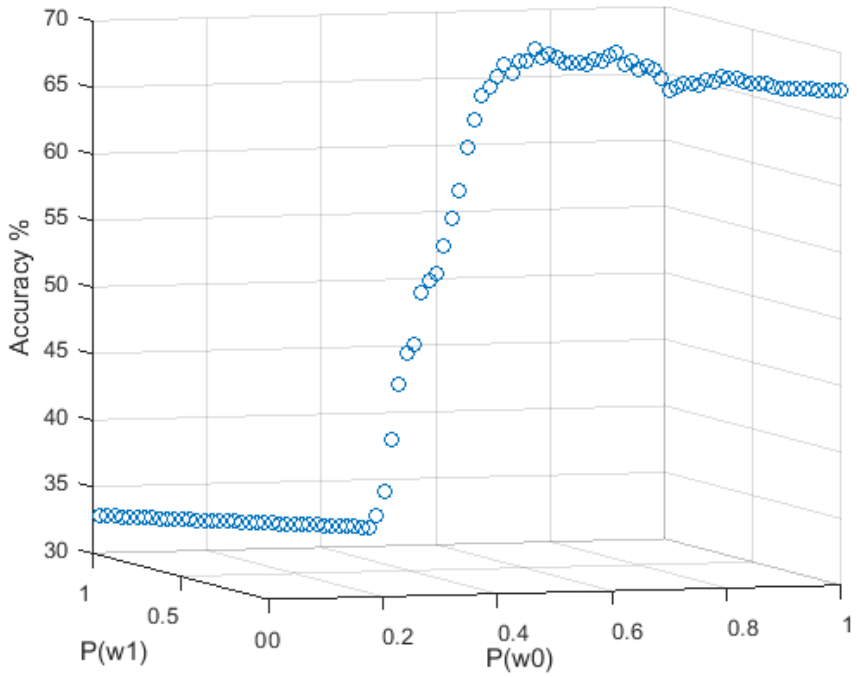

Figure 17: PCA Testing Set Case 2 with variant Prior Probability

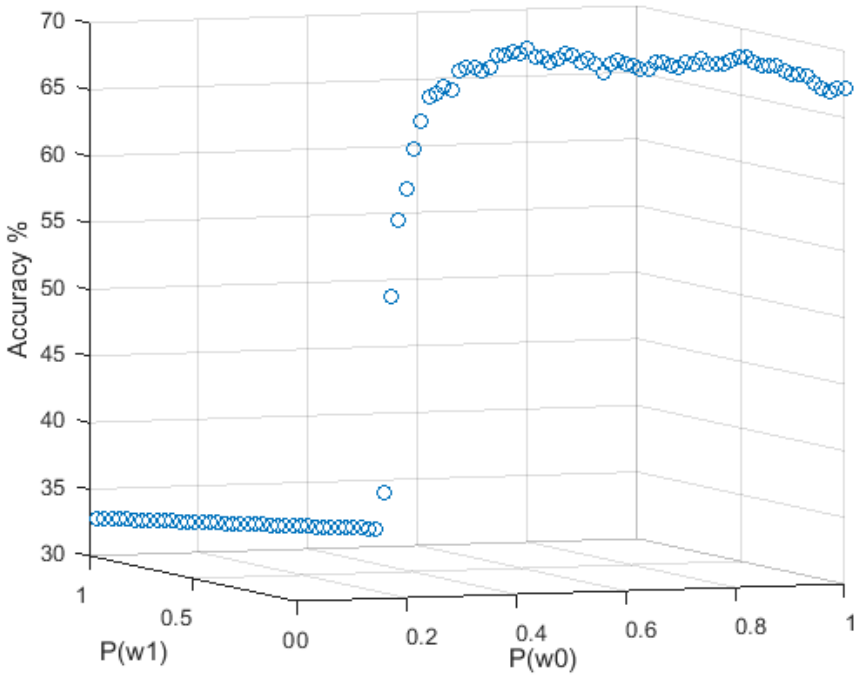

Figure 18: PCA Testing Set Case 3 with variant Prior Probability

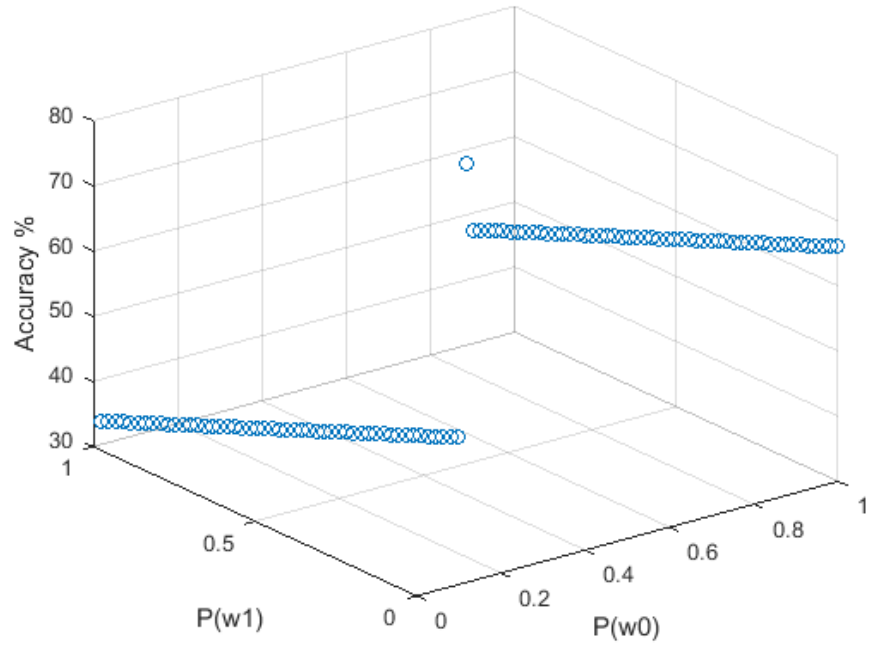

Figure 19: FLD Training Set Case 1 with variant Prior Probability

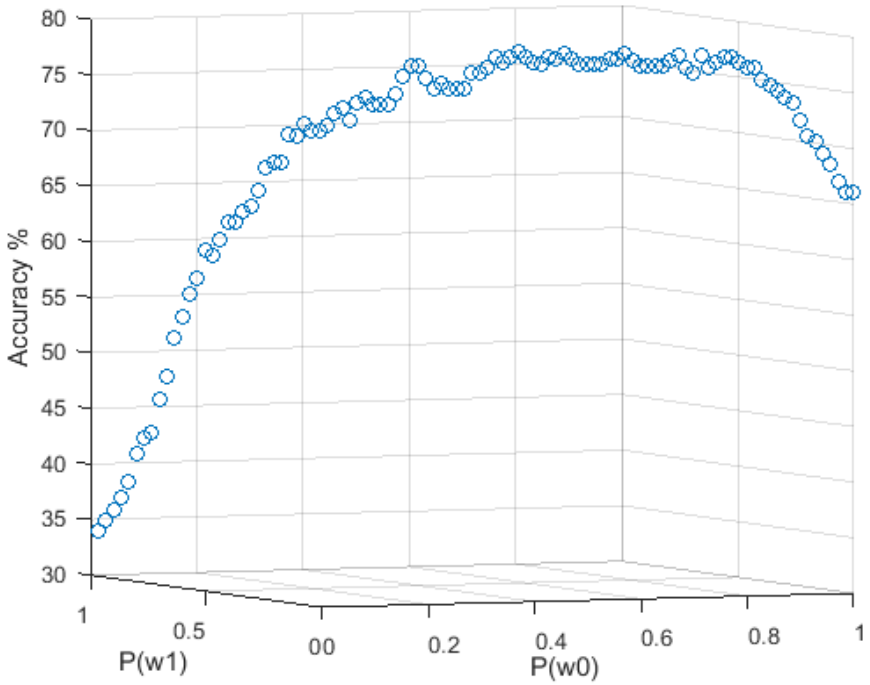

Figure 20: FLD Training Set Case 2 with variant Prior Probability

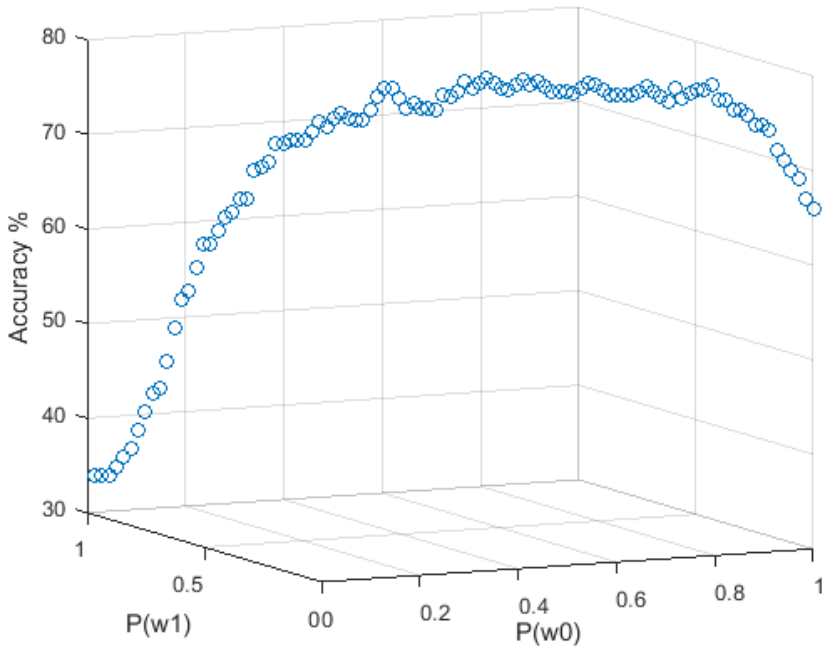

Figure 21: FLD Training Set Case 3 with variant Prior Probability

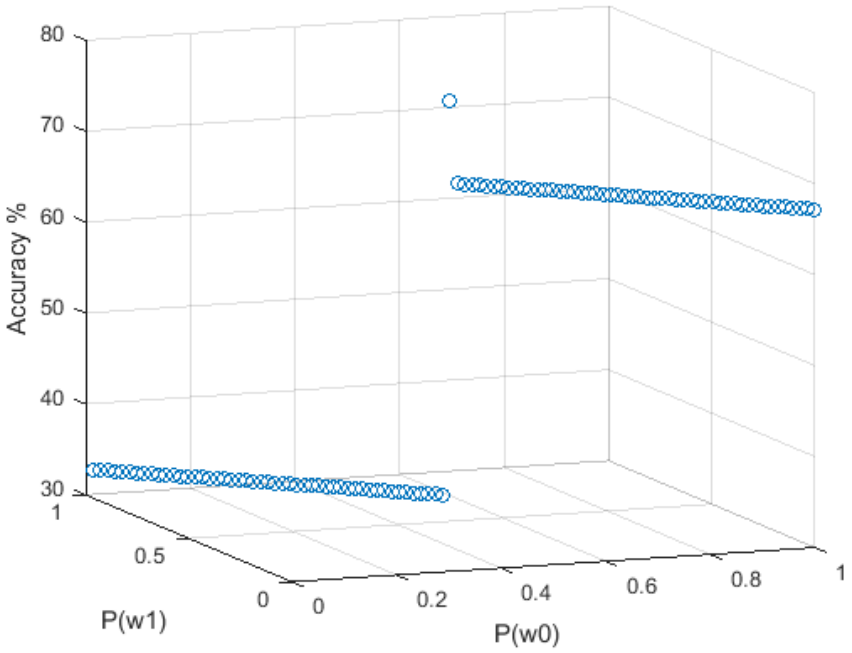

Figure 22: FLD Testing Set Case 1 with variant Prior Probability

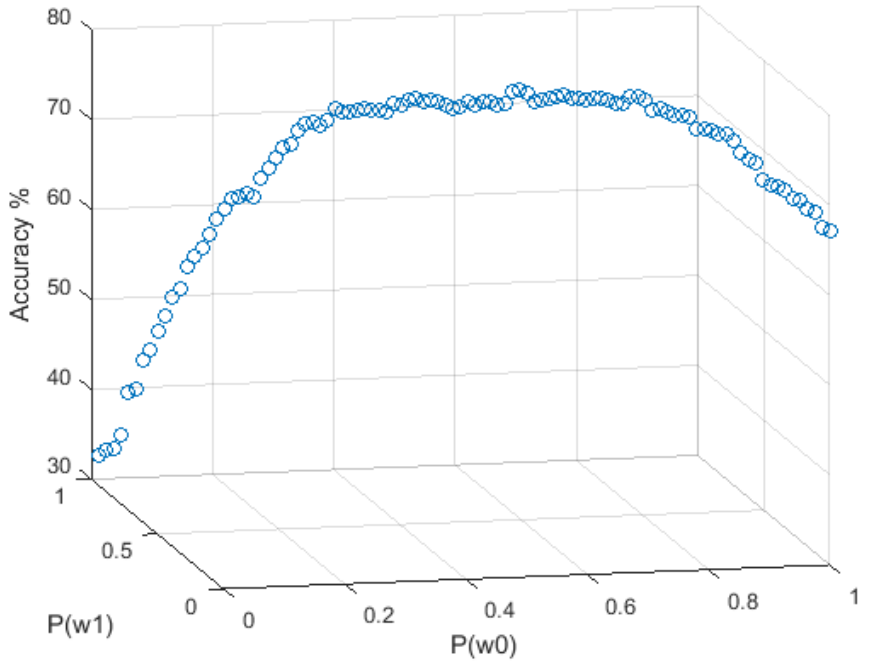

Figure 23: FLD Testing Set Case 2 with variant Prior Probability

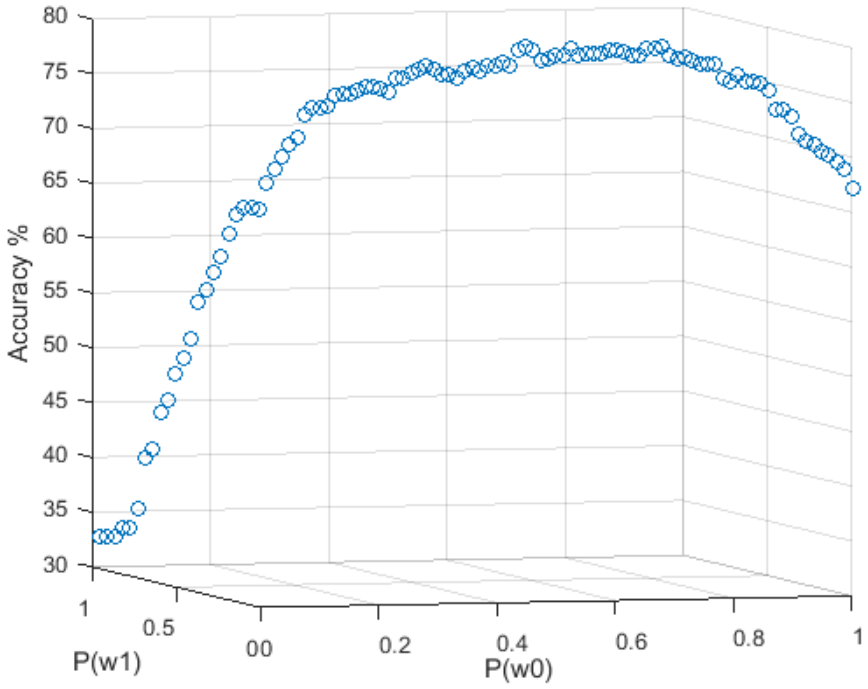

Figure 24: FLD Testing Set Case 3 with variant Prior Probability

Data Set	Data Type	Case	Best P(w0)	Best P(w1)	Accuracy %
Training Set	Normalized	Case 1	1	0	66
		Case 2	.82	.18	79
		Case 3	.54	.46	80
	PCA	Case 1	.87	.13	67.5
		Case 2	.72	.28	67.5
		Case 3	.92	.08	67.5
	FLD	Case 1	.5	.5	76
		Case 2	.84	.16	78
		Case 3	.86	.15	78.5
Testing Set	Normalized	Case 1	1	0	67.17
		Case 2	.61	.39	80.72
		Case 3	.73	.27	77.41
	PCA	Case 1	.84	.16	69.28
		Case 2	.7	.3	69.28
		Case 3	.87	.13	69.28
	FLD	Case 1	.5	.5	76.20
		Case 2	.74	.26	79.52
		Case 3	.74	.26	79.52

Table 5: Variant Prior Probability determining each data set

Near universally across all data sets, their is a clear trend to favor the 0 class as having a higher prior probability. This is suggestive of such a case, non diabetic, being the typical trend among society as a whole. In addition. When compared to data in Table 3 where prior probability was held in equality, this seems to be supported as there was an average of 3 to 7 percent increase in accuracy across all cases as well. The same trends we noticed prior, such as Case 2 for the normalized test data having the highest accuracy, are continued as well.

3.3.2 Kth Nearest Neighbor Estimation

Final work was done with the changing of values for prior probability on kth neighbor estimation. For each simulation, the kth value which gave the highest probability in the prior static trials was utilized while prior probability was held in flux between 0 and 1 for either case. The results are shown in the below figures:

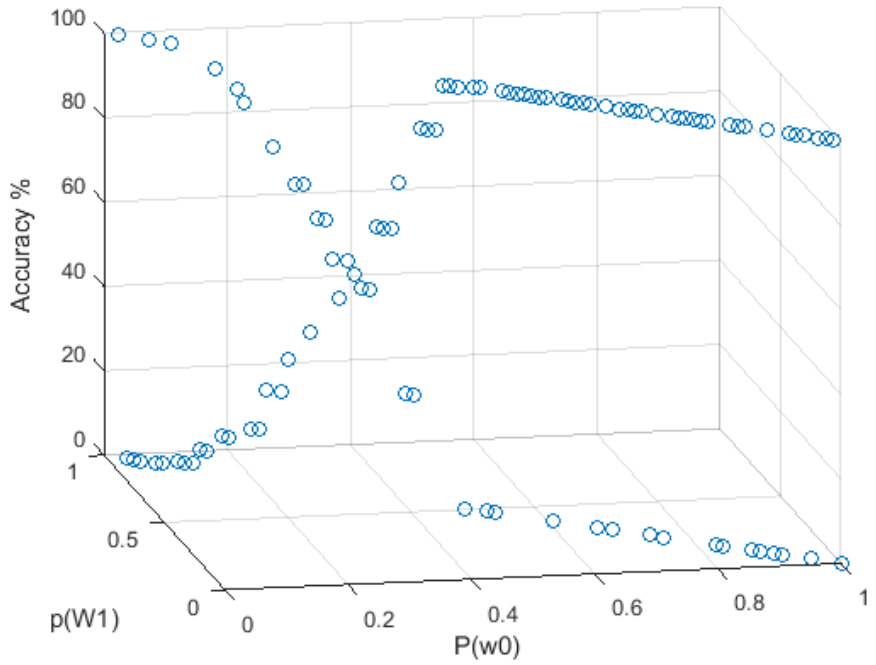

Figure 25: K = 33 Normalized Training Set with variant Prior Probability

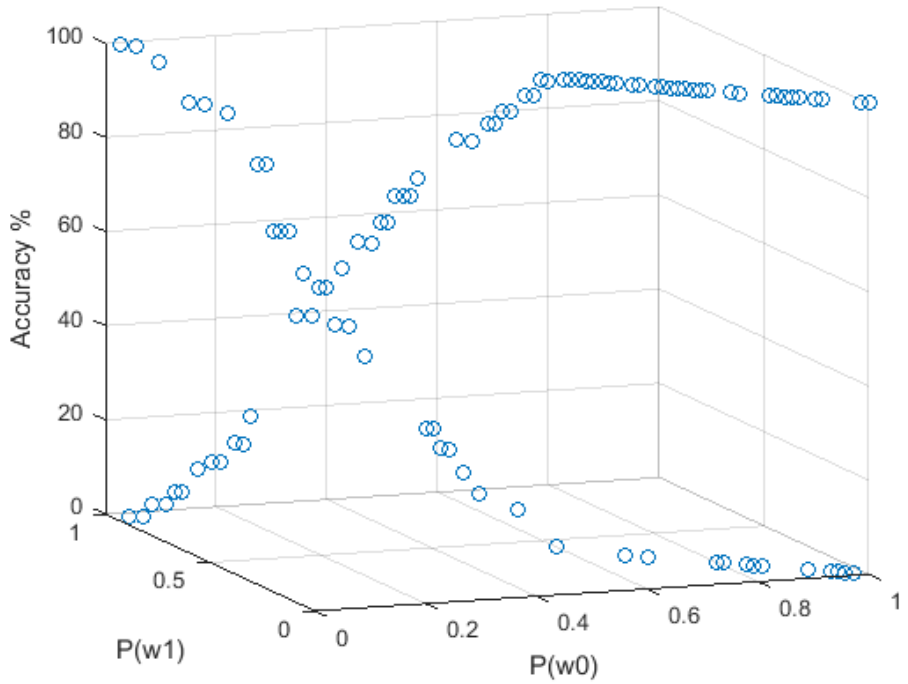

Figure 26: K = 33 PCA Training Set with variant Prior Probability

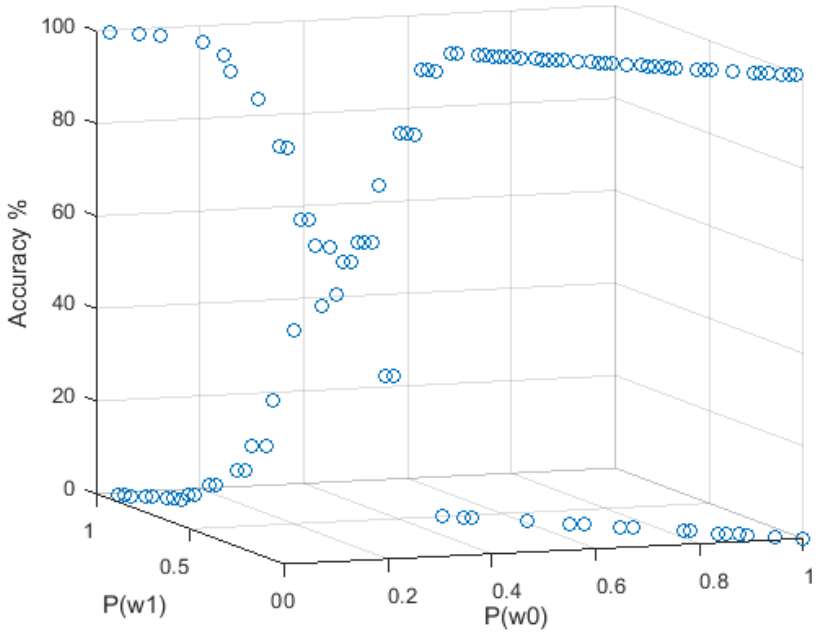

Figure 27: K = 19 FLD Training Set with variant Prior Probability

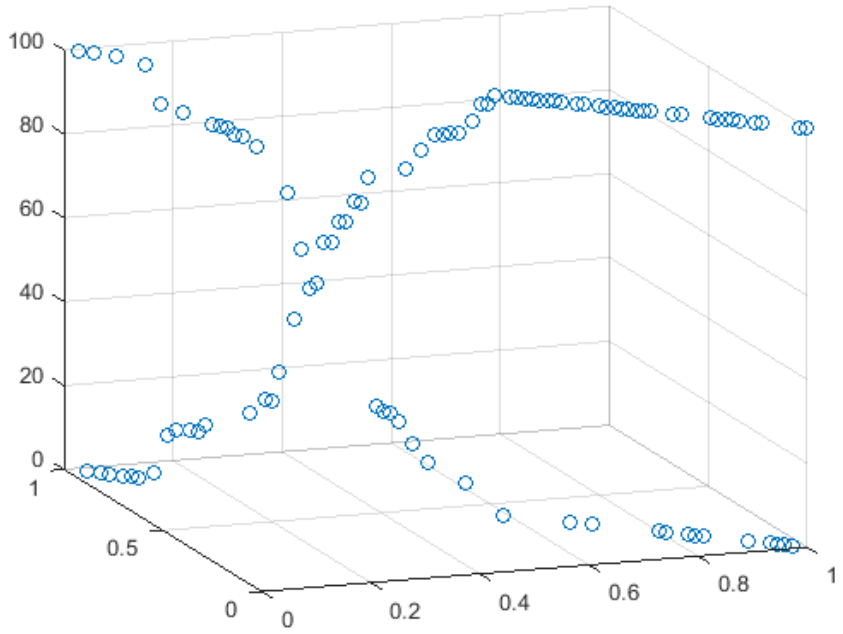

Figure 28: K = 37 Normalized Testing Set with variant Prior Probability

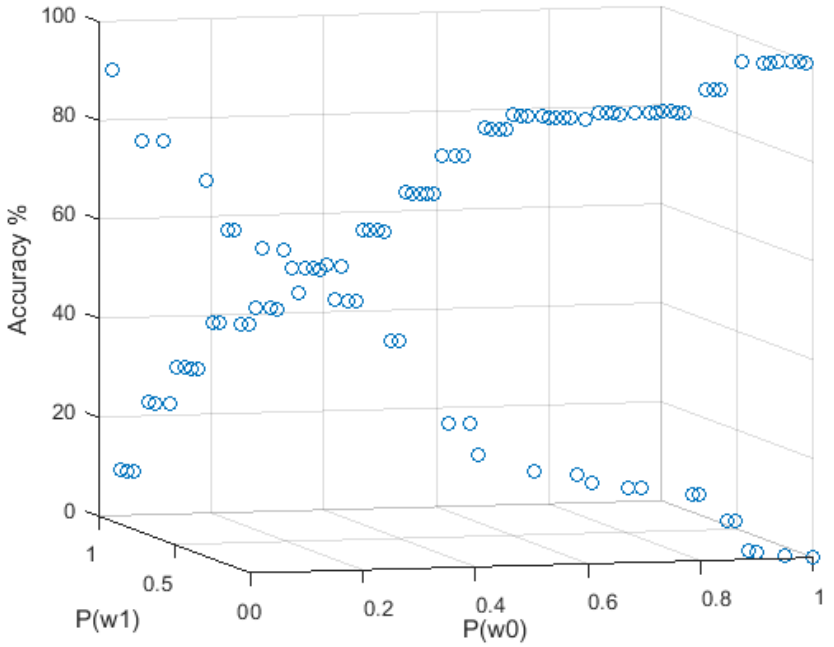

Figure 29: PCA Testing Set with variant Prior Probability

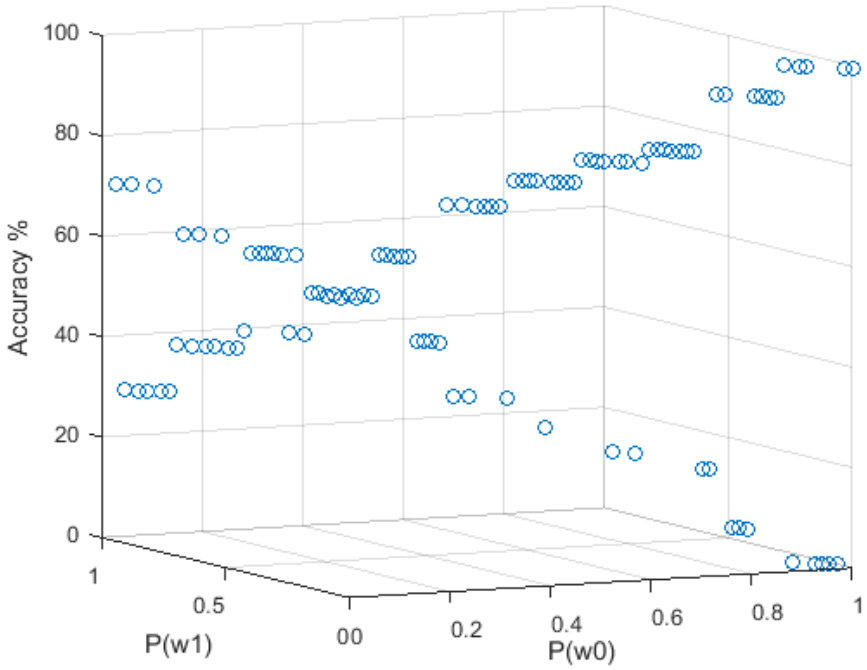

Figure 30: FLD Testing Set with variant Prior Probability

These results are much harder to quantify as universally with every testing parameter utilized there was a cross-like pattern which emerged. A very high or very low prior probabilities for each case their was a near perfect scoring. Upon further consideration this actually matches perfectly with expectations. Data is usually clustered closely to those within its own class. At very low k values it will nearly universally pick the correct class, with the exception of a few outliers. The same will occur as k approaches a very high value. The class with has a higher concentration of sample within the overall space will eventually begin to win over by sheer numbers. Notice for all cases with the testing data, the class 0, which is more represented in the overall space, generally has the highest overall average correctness over different prior probabilities. This falls within all the same conclusions we have drawn from our previous data sets. The grander picture from this is there is a inherent unreliability to using variant prior probability with kNN determinations. This is likely a result of the fact there are two dimensions of change in these cases, in both the prior probability, as well as the kth size of the space.

4. Summary

Using applied Bayes Theory in conjunction with Gaussian distributions we have constructed discriminant functions to determine class based upon likelihood ratios and temporal relation to neighboring samples. Our work has concluded for normalized data, case II is best for determining data, utilizing the prior probabilities of .61 for class 0 and .39 for class 1. We have also determined FLD to be the most effective reduction methods using either class II or II with the prior probabilities of .74 for class 0 and .26 for class 1. For the kNN utilizing the normalized data with k = 37 yielded the best results, with the PCA reduction being the best reduction form, only losing approximately 9% accuracy with k = 49 neighbors required. A clear disadvantage of using the FLD decomposition was also demonstrated. While there was no clear seeming advantage to varying prior probabilities with the kNN method, it did reveal specific behaviors of accuracy relative to high and low prior probability values.

The objective of this paper was met without undue difficulty. Through creation of my simulation program my overall understanding our the mechanism of these methods, in particular the kNN estimation method, has improved.In addition, the methods utilized could easily be used in a plethora of other possible pattern classification methods, now that the general behavior is well understood. Working with this program also afforded me greater practice and understanding of the MATLAB 3D scatter plotting mechanics. Based upon the data it is tempting to conclude the kNN method is generally a more accurate method as it did indeed produce the highest overall accuracy, but this is not universal. As previous work in earlier projects has shown, your method

of creating a determinant function is highly dependent upon your data set. While the more generalized approach of kNN does assume no prior knowledge of the data set to make it generic, this is not always the case. When data is known to distribute in a particular manner, such as in a normalized manner, utilizing the appropriate distribution will likely yield better results, such as the high accuracies achieved in project 1 using the Gaussian Distribution on normally distributed data. Further work could also be performed using the error rate to generate data within a confidence interval. As we now know the relative accuracy of several methods of classifying data, it could now be possible to take trends within the classified data an create confidence intervals by which we may further create classification methods. Based upon which feature values are within a particular range, we may designate all features within a particular range to be of a particular class, with probability which we have derived for that given feature to appear. In this manner, we can use these methods here to further expand and develop our classification methods.

Development of the simulator necessary to create this project was not terribly arduous, however it could use refactoring as it is written nearly entirely monolithically. Refactoring could easily lend itself to the creation of a static library for probabilistic classification methods which could have great usage in the future. In addition, the bash script created to run this program could easily be expanded to create more simultaneous programs as well as further develop additional simulations of this existing program on new data sets. As alluded to at the outset of this program, pattern classification is highly useful because of its ability to quantify question to a computing device with are not very computationally intuitive like handwriting or facial expressions. It also have a further usage of a great predictor based upon historical data. By collecting data on trends over time, say the climatological data for a particular region of the planet over the course of millenia, we could make predictions about future cosmetological changes for that area with a certain probabilistic range. The current simulator for this project could be expanded to this end. To take the derived probabilities and then draw a forecast of sorts for future possible trends, possibly even with particular portions of the simulation being able to be manipulated by the user for a more dynamic model. This could have grand applications within the real world fields of metrology, geology, and economics to name a few. The power of probabilities is in effect a near ability to predict the very future, within a given range of confidence of course. The results of this project further develop our ability to do just that.

5. References

1. B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, 1996
2. Dutt, Vinita, Vikas Chaudhry, and Imran Khan. "Pattern Recognition: An Overview." American Journal of Intelligent Systems, 2012. Web. 10 Mar. 2017.
3. The Editors of Encyclopedia Britannic.a. (2017). Thomas Bayes. Retrieved February 12, 2017, from https://www.britannica.com/biography/Thomas-Bayes