2020-01-01

Development Of Advanced Statistical Methods For Multivariate Classification

Mario Cardenas Jr
University of Texas at El Paso

DEVELOPMENT OF ADVANCED STATISTICAL METHODS FOR MULTIVARIATE CLASSIFICATION

MARIO CARDENAS JR

Master's Program in Physics

APPROVED:

Marian Manciu, Ph.D., Chair

Felicia Manciu, Ph.D.

Giulio Francia, Ph.D.

Stephen L. Crites, Jr., Ph.D.
Dean of the Graduate School

DEVELOPMENT OF ADVANCED STATISTICAL METHODS FOR MULTIVARIATE CLASSIFICATION

by

MARIO CARDENAS JR

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Physics

THE UNIVERSITY OF TEXAS AT EL PASO

May 2020

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Research into the diagnosis and treatment of a variety of diseases has been a longstanding area of interest. The use of biomarkers is one way in which this area is explored. Biomarkers have several potential clinical applications some of which include treatment response predictions, risk assessment, and class identification [3]. A biomarker is defined as "any substance, structure, or process that can be measured in the body or its products and influence or predict the incidence of outcome of disease" [5]. Biomarkers are also used to track disease progression, serve as surrogate clinical endpoints, and measure and detect the effects of a drug [1,2]. Thus, biomarkers have the potential to improve the early detection of a disease present in a subject and lead to an improved life expectancy. As an example, the early detection of the presence of a disease may lead to shorter treatment response and may ultimately result in a lower mortality rate [2]. Another example of where lower mortality can occur is with risk assessment. If a risk assessment can be made based on biomarkers, a patient can take certain actions to reduce their risk of developing a particular disease by taking preventive measures, like that of a lifestyle changes [3].

In biospectroscopy one can make use of multivariate and univariate methods for biomarker identification [2]. The field of biospectroscopy provides a wide range of spectra data via techniques like IR Spectroscopy, Fourier-transform IR, and Raman Spectroscopy. Raman spectroscopy uses Raman scattering where one measures the vibrational energy of chemical bonds present in cell or tissue samples. Here, a beam of monochromatic light is directed at a sample of interest typically within the mid-Infra-red range (λ=5-25 μm). During this process, the incident photons are polarizing the present molecules' electron cloud and promote them into excited states. These states lie above the molecule's ground state and are considered unstable or short-lived. Because of the instability of this virtual state, the incident photons are quickly scattered (re-emitted) and then captured using a detector. Whereas most of the scattering is elastic (and therefore carries no information about the scattering material), Raman spectroscopy is concerned with the inelastic scattering, which provides information about the chemical structure of the scatterer [1]. Raman Confocal Microscopy is an experimental imaging technique, which provides individual Raman spectra on a (usually) 250x250 map, where this spectra is employed for the image reconstruction. In this process and other similar techniques typically result in large data sets providing ample information regarding the molecular composition of the sample measured. For example, Raman measurement discussed in the last part of the thesis contains 250x250x1024 data points. To analyze such large data sets, one can turn to principal component analysis, linear discriminant analysis, and other multivariate analysis methods to aid in the extraction of information [3]. The combination of biospectroscopy and multivariate methods can result in cell type identification, biomarker identification, and other information regarding the measured sample(s). Methods like principal component analysis are of particular interest because of their dimensionality reduction capabilities which can reduce the computation power required used to analyze them.

With the help of microarray technology, gene expression profiles, like spectra data sets, can be used for disease classification. In fact, an important aspect of microarray analysis is cancer classification. This is because cancer may be a genetic disease, and so the analysis of gene mutations may lead to the identification of the gene(s) responsible for cancer [4]. Gene expressions data sets typically consist of low sample (observations) size in the order of tens and a high number of genes (variables), in the order of $\sim10^4$. The dimensionality of the data sets presents researchers with a problem commonly referred to as the “curse of dimensionality”. This can lead to data overfitting in microarray cancer classification [4].

Class prediction (classification) and feature selection are two important types of analysis employed when analyzing gene expressions. In feature or gene selection the analysis focuses on finding the most informative genes. To do this, three types of approaches can be used: the filter, wrapper, and hybrid approach. The filer approach uses techniques like that of Random Forest Ranking, where features are ranked based on decision trees. The wrapper approach typically involves bio-inspired algorithms like the Genetic Algorithm, Ant Colony Optimization, and others. Finally, the hybrid method combines the two mentioned approaches by reducing the features present in a data set followed by feature optimization of the reduced data set or subset. In classification, supervised learning techniques can be used by creating classifiers based on learning data sets. A classifier can then be used for class prediction when applied to other data sets not used in the training phase. Some examples of algorithms currently used include Support Vector Machines, Neural Networks, K Nearest Neighbor, and others [4].

In the following chapter, I will give a brief description of the commonly used supervised and unsupervised learning techniques which will serve as an overview of the novel method proposed in Chapter 3 for the classification of data sets. Chapter 4 details another novel alternative classification method, which was shown to provide an unprecedented accuracy for the classification of a particular disease, using a reduced set of data points [paper].

Chapter 2: Multivariate Methods

2.1 Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised statistical data analysis tool commonly used to process genomic datasets [2]. This is due to the high dimensionality of the matrices obtained when, for example, you measure the gene expression levels for a given cell or tissue sample. This dimensionality reduction technique used in the field of multivariate analysis where one can benefit from reducing the computational cost or time of analysis. PCA analysis a dataset via orthogonal transformations using Singular Value Decomposition (SVD).

Singular Value Decomposition provides us with a manner to calculate the Principal Components of a matrix without having to compute the covariance matrix [14]. Using Singular value decomposition to find the Principal Components of a matrix has been regarded as the best computational approach to finding Principal Components [12]. Finding Principal Components of a given matrix, as well as the covariance matrix, serve a very important role in multivariate classification which I will briefly describe in the following pages.

Here matrices will be denoted in bold upper-case letters and the transpose of said matrices with an upper case T superscript. Elements of a matrix will be denoted by the lowercase letter of the corresponding matrix uppercase letter with subscripts defining the element index. The matrix that is to be processed, matrix **A**, will be separated into three main matrices when subjected to Singular Value Decomposition:

A= **SVD**^T

Where **A** represents an (n x p) matrix composed of n observations and p variables. **S** is an (n x r) matrix referred to as the left singular matrix, **V** is an (r x r) matrix referred to as the diagonal matrix, and **D** is a (p x r) matrix referred to as the right singular matrix. The diagonal matrix **D** is of particular interest because it contains the square root of the eigenvalues of the matrix **A**^T**A**. The singular values lie on the diagonal elements of the matrix **D** with zeroes on the remaining elements or the off-diagonal elements.

The columns of matrix **S** contain eigenvectors computed from the **AA**^T matrix and the columns of the **D** matrix eigenvectors of the **A**^T**A** matrix [14]. With the left and right singular matrices, one can compute the coefficients and standard deviation of the principal components of the covariance matrix.

To find the principal component scores we multiply **A** by **D** (Note that we are multiplying the original matrix by **D** not the transpose of **D**) as follows:

P = AD = (**SVD**^T)**D = SV**

Where **D**^T**D = I** which is an identity matrix with dimensions (r x r). **P** is then a matrix whose columns contain the Principal Component scores, or loadings, with dimensions (n x r). The row elements on the **P** matrix contain the projection scores of each of the n variables along each of the Principal Components.

The Principal Component scores are a measure of the variance of the matrix **A**. The geometrical interpretations is that when a matrix is subject to Principal Component Analysis procedure the original matrix is rotated in such a manner that the variance is maximized. This is seen when analyzing the Principal components of the processed matrix. The resulting projection scores obtained from the **P** matrix are the projections of the n variables onto an r^{th} dimensional space in which each of the r axes spans the space corresponding to the new set of r Principal Axes. In a sense, the r^{th} dimensional coordinate space spanning the original matrix is rotated to that of the one provided by the new set of orthogonal Principal Axes corresponding to the Principal Components. The new set of axes are ordered such that the first Principal Axes corresponds to that of the highest variance in the matrix **A**. This means that the first few principal components contain the most variance. Often one can find that the first three principal components are sufficient to describe the majority of the variance, containing up to 99% of the variance of the matrix or dataset [2]. Figure 2.1 [15], displays a plot of words found in a dictionary relating the number of letters in a word (vertical axis, red) vs the number of lines in the definition (horizontal axis, red). The green axes labeled one and two correspond to the first and second principal components of the data, respectively. The second set of axes has been superimposed to demonstrate the geometrical meaning of an orthogonal rotation of the matrix.

Whenever necessary, one can determine the number of principal components required to properly represent the amount of variance needed for classification. This amount is, of course, arbitrary as the amount of variance per principal component can vary. By determining the appropriate number of components required to describe the variance of **A**, we can then reduce the dimensionality of the matrix or data set we need to analyze the data set. The following three methods have been proven to work and have a good intuitive standing [12].

Figure 2.1 Words on a dictionary

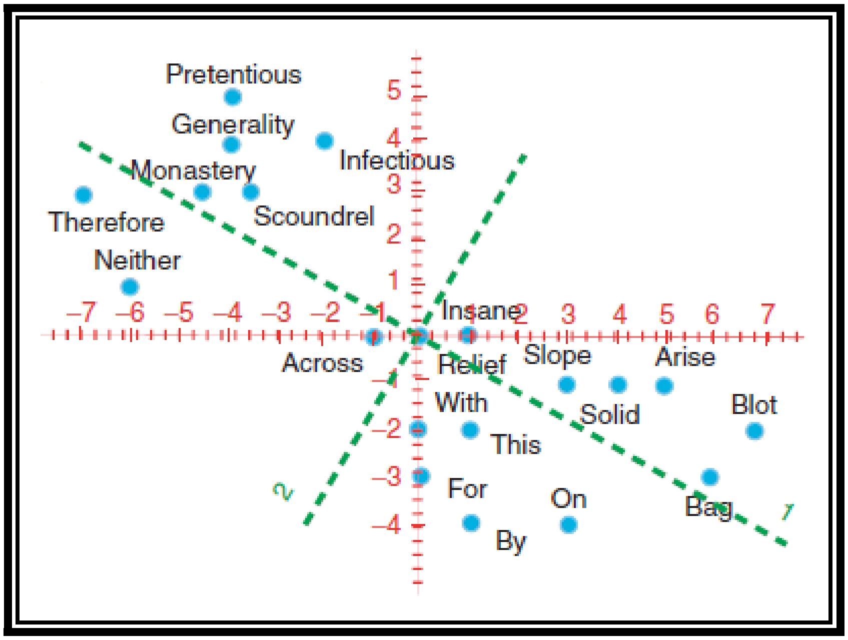

Figure 2.1: Plot of words found in a dictionary. Here the number of words has been graphed vs. the number of lines in the definition of said words.

Cumulative Percentage of Total Variation: Given that the sum of the variances of the elements of **A** is the sum of the variances of the principal components in **P**, we can simply sum the column elements of Z to find the amount of variance held by each principal component. Once the degree of variance is determined, the following task is then to determine the amount of variance desired and to choose the set of principal components required to match that degree. The degree of variance chosen can be determined based on the characteristics of each dataset with an appropriate analysis of the principal components. Namely, identifying the source of variation per principal component then choosing the appropriate set of components.

The second method utilized is that which evaluates the *Size of Variances of Principal Components*. Here the relevant principal components are determined by how close the variance held by each component is to a value of one. In a matrix where all the elements of **A** are independent of each other, i.e. uncorrelated, the principal components will contain a variance of one. This means that those components whose variance is less than that hold very little information regarding the matrix and is of no interest to the classification process. This is condition is referred to as Kaiser's rule, where only the components having a variance greater than one are chosen. In general, only a single principal component will be retained per variable group.

The Scree Graph and the Log-Eigenvalue Diagram: In this method, we make use of graphing utilities by plotting the eigenvalues of the matrix **AA**^T vs the principal component number. A line is then drawn joining each eigenvalue to the following eigenvalue corresponding to the next principal component.

Because the eigenvalues are ordered according to their component number one can see that the 'steepness' of the lines tends to diminish from left to right. The component number at which the slope begins to level-off determines the number of components that should be retained. This point is referred to as the 'elbow' in the graph. A similar procedure is employed for a Log-Eigenvalue Diagram, where the logarithm of the eigenvalues are plotted against their component number and joining the log of the eigenvalues as in the Scree Graph. The number of components retained will also be determined by the component at which the slopes approach a straight line.

The advantages of the dimensionality reduction of a dataset can also be appreciated when using graphical representations of the principal components. Due to the majority of the variance being held along the first three principal components, the data set can be represented in a two or three-dimensional plane [12]. This allows for additional visual aids when interpreting the variance of a dataset or matrix.

Figure 2.2 Scree Graph Example

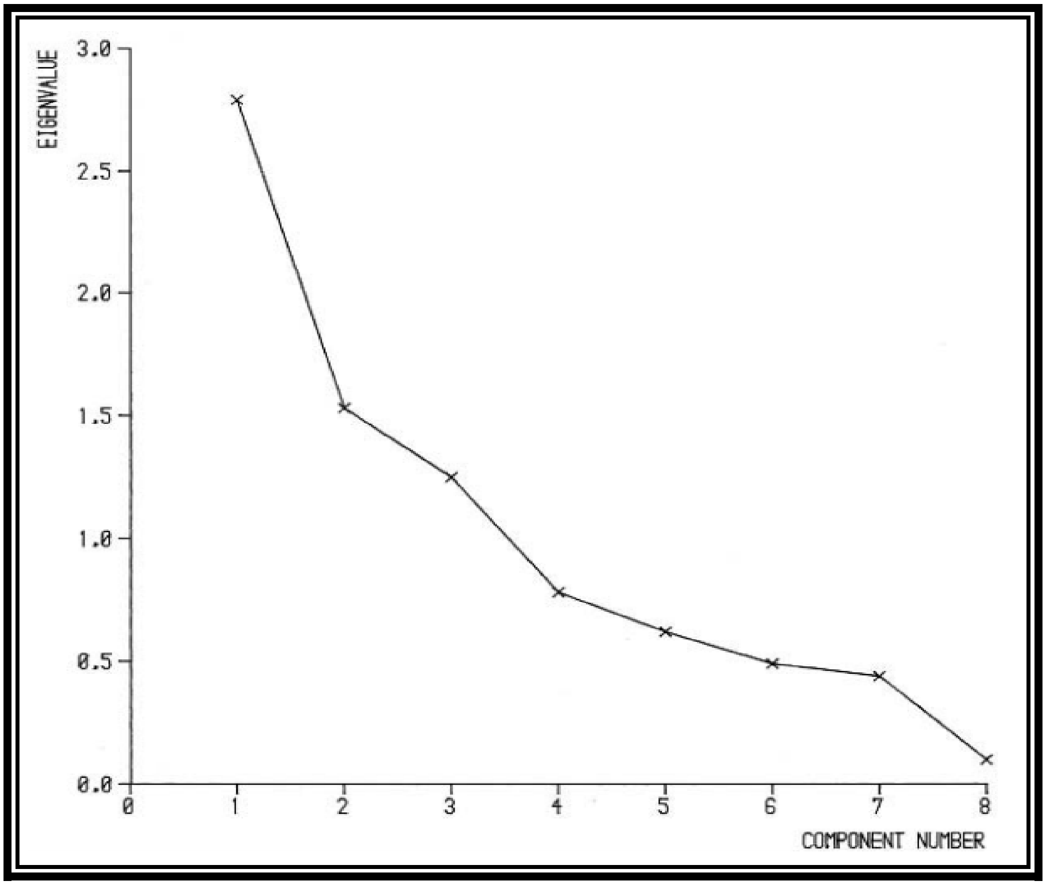

Figure 2.2: Example of a Scree graph found in Jolliffe, 2002.

In figure 2.3 [12], a plot is made of 50 observations and two variables displayed with high amounts of correlation along each axis. However, one can see that on the second component the variables are more dispersed. Figure 2.4 [12], is a graph where the observations are plotted with respect to their principal components. Here the variance is more expressed along the first component on account of this first component having the most variance as mentioned earlier.

Principal Component Analysis is a helpful tool when a dimensionality reduction is desired, and an unsupervised technique is of interest. One can see why it is amongst the most popular multivariate statistical technique [15].

Figure 2.3 Observations vs components

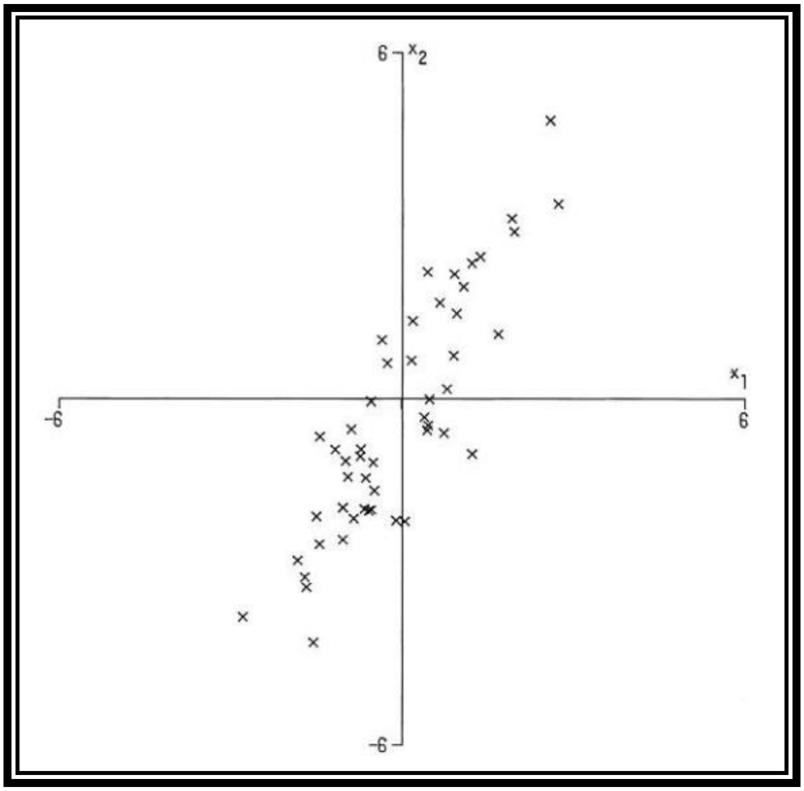

Figure 2.3: Plot of 50 observation vs two component

Figure 2.4 Principal components two-dimensional plot

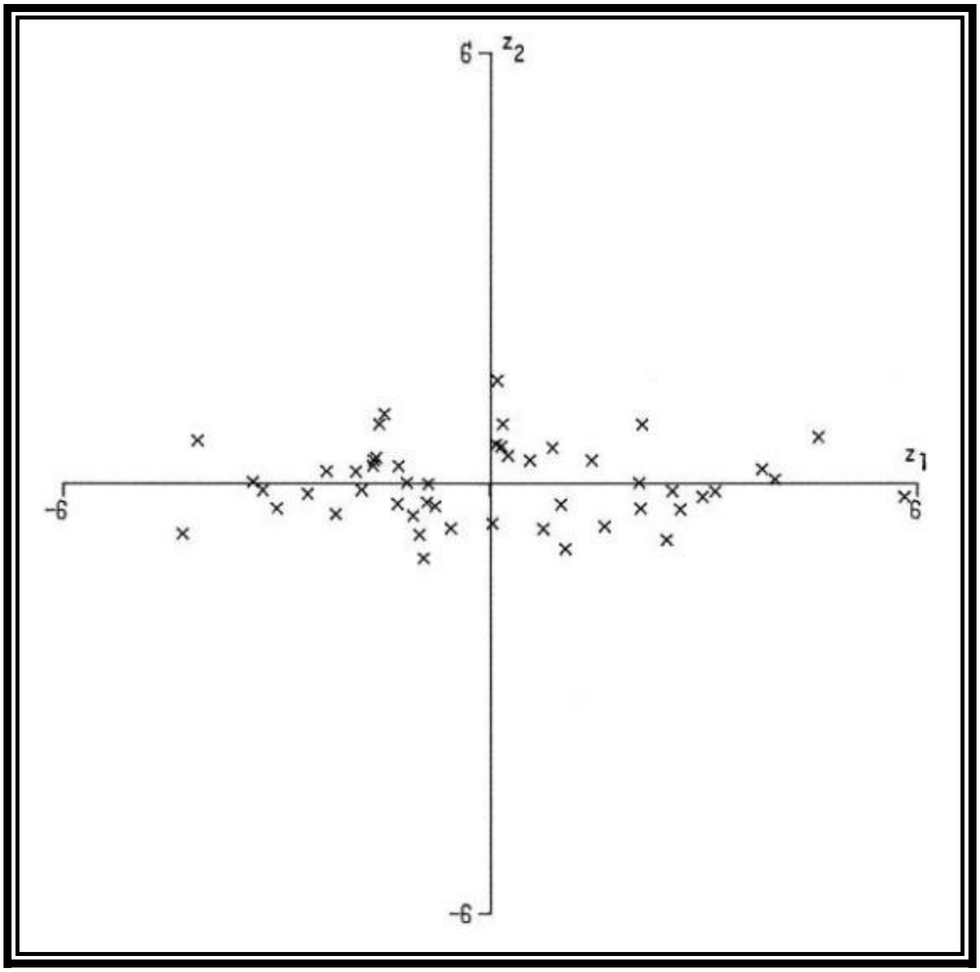

Figure 2.4: Plot of 50 observations with respect to principal components

2.2 Linear Discriminant Analysis

Linear Discriminant Analysis is another technique used for data classification and dimensionality reduction. However, this technique is a supervised learning technique, as opposed to principal component analysis which is unsupervised [19]. Linear Discriminant Analysis has two basic objectives: *Discrimination* and *Classification*. When undergoing the task of discrimination, a learning data set containing multivariate observations are used to build a classifier. This classifier obtains the largest separability between a set of *n* predefined or known classes. This defines this method as a supervised learning technique. Once the classifier is obtained we can use it to predict or classify a new set of observations for which we have no prior class knowledge. To do this we must construct a linear discriminant function or in case *n* = 2, Fisher's linear discriminant function [19]. This function is derived via a linear combination of the learning set. The application of the linear discriminant function is regarded as a linear transformation of some data set **D**.

There are two transformations that a data set can undergo when determining the linear combination of a training set, a class-dependent, and a class-independent transformation. The former maximizes the ratio between-class variance to within-class variance, therefore obtaining the optimal class separability. The latter maximizes the overall variance within-class variance, forgoing class identity when performed [16] [20]. To make use of either transformation, a data set, or in our case a matrix, must be supplied for each of the classes that are to be analyzed. That is, the data set supplied must be separated into individual subsets according to class type which they belong to. The classes, in this case, must be known before discriminant analysis is used.

$$\boldsymbol{D} = [\boldsymbol{S_1}, \boldsymbol{S_2}, .\,., \boldsymbol{S_n}]$$

Where **D** is the data set that is to be analyzed and $\boldsymbol{S_1}, \boldsymbol{S_2}, .., \boldsymbol{S_n}$ the sets for each of n classes all in matrix form. The mean m_n of each data set is then computed individually for each class data sets and an additional mean $m_{all}$is computed for the entire dataset **D**.

$$m_{all} = \sum_{i=i}^{n} p_i \cdot m_i$$

Where p_i is the a priori probability of the i^{th} class. For both transformations, we define the class separability using the within-class and between-class scatter as follows:

$$within - class\ scatter: \boldsymbol{S}_w = \sum_{i=1}^{n} p_i \cdot (\boldsymbol{D} - m_i) \cdot (\boldsymbol{D} - m_i)^T$$

$$between - class\ scatter: \boldsymbol{S}_b = \sum_{i=1}^{n} p_i \cdot (m_i - m_{all})(m_i - m_{all})^T$$

With this, we can define the criteria to optimize the class-dependent and class-independent transformation by transforming the scatters to set of numbers as follows [16][20]:

$$for\ class - dependent\ transformation:$$

$$\boldsymbol{C}_{c-d} = \sum_{i=1}^{n} C_i = \ [(\boldsymbol{D} - m_i) \cdot (\boldsymbol{D} - m_i)^T]^{-1} \times \boldsymbol{S}_b$$

$$for\ class - independent\ transformation:$$

$$C_{\boldsymbol{c-ind}} = \ [\boldsymbol{S}_w]^{-1} \times \boldsymbol{S}_b$$

The linear discriminant functions are then,

$$[\boldsymbol{U}_{c-d}]_i = \ \boldsymbol{C}_{cd}{}^T \times \boldsymbol{S}_i$$

$$[\boldsymbol{U}_{c-ind}]_i = \boldsymbol{C}_{c-ind}{}^{T} \times \boldsymbol{S}_i$$

Where $\boldsymbol{U}_{cd}$ and $\boldsymbol{U}_{c-ind}$ correspond to the discriminant function for class-dependent and class-independent transformation criteria, respectively. These transformations result in discriminant scores which when applied to a specific $\boldsymbol{S}_i$ data set. These scores are then used to determine the amount of separation obtained using the discriminant function, typically displayed as a Gaussian distribution.

As an example of this classification procedure, suppose we have two data sets $\boldsymbol{S}_A$ and $\boldsymbol{S}_b$ whose between-class separation we wish to optimize. We follow the procedure above using the class-dependent criteria to obtain the linear transformation $\boldsymbol{S}_A$ and $\boldsymbol{S}_B$ and use a Gaussian distribution to display the linear transformation for each class along the new axis defined by the discriminant function. A plot of the two example data sets is provided below, where both classes are projected along X_1 and X_2 axes using Gaussian distributions, Figure 2.5. We can see a considerable amount of overlap along each of the axes. However, the projection score obtained from the linear discriminant function is exhibiting a larger separation.

Figure 2.5 Two class LDA

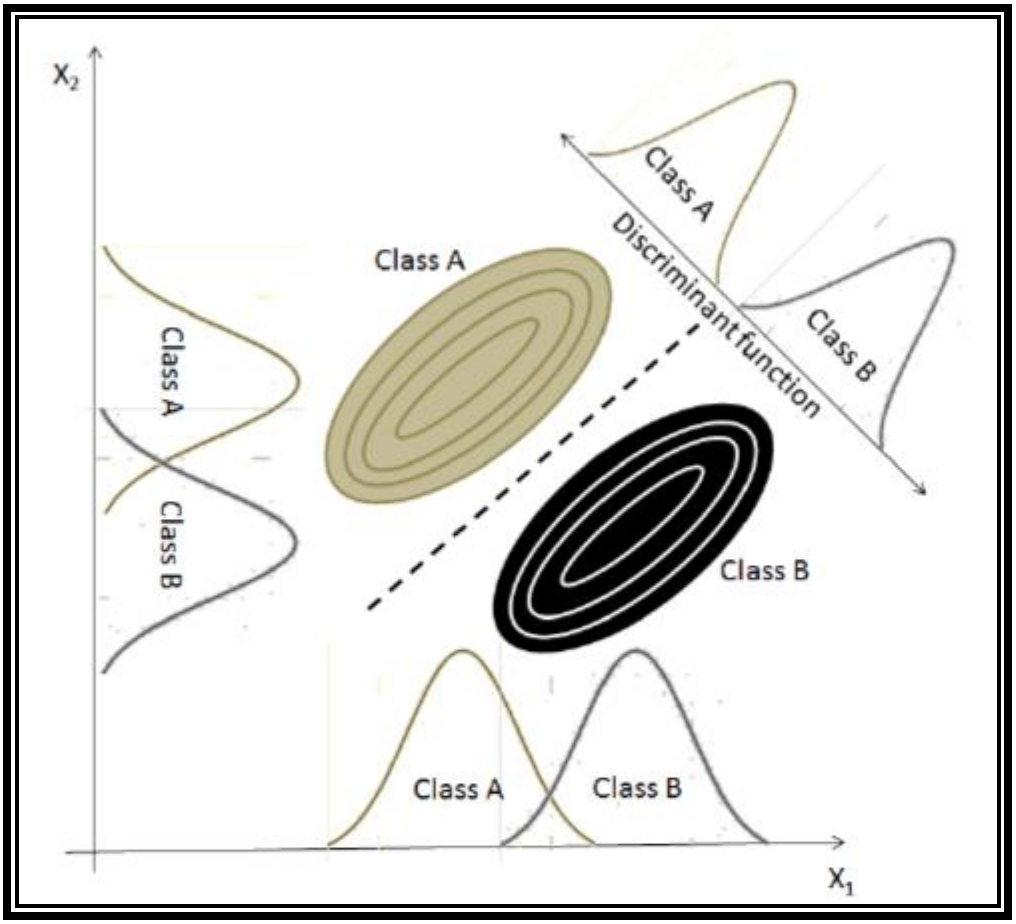

Figure 2.5: Is an example of two classes exhibiting an overlap along the axes X_1 and X_2 axis, but full separation along the discriminant function, [18].

With this discriminant function, we can calculate the projections score for a test dataset, $\boldsymbol{D}_{test}$ for which we have no prior knowledge of classification status.

$$[\boldsymbol{U}_{c-d}]_i = \boldsymbol{C}_{cd}{}^T \times \boldsymbol{D}_{test}$$

In this manner, the linear discriminant function obtained from our optimized class separation can be used to determine which class the elements of $\boldsymbol{D}_{test}$ belong to according to their scores. We can also note that our system is no longer described in two dimensions $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. Instead, it is described in one, along that of the discriminant function.

2.3 Linear Support Vector Machines

Support Vector Machines is a method used in Machine Learning classification. The goal of support vector machines is to establish a hyperplane or line that separates a set of n classes. The quality of the separation is obtained by maximizing the margin of separation between classes [21]. This method requires the use of a training set to establish the classification parameters, making it a supervised learning technique [21] [22]. Two cases of interest serve as an introduction: the Separable Case and the Non-Separable case (for this thesis, I will focus on the separable case). The Separable case also referred to as the Maximal Margin Classifier, establishes the condition that variables must lie on one side or the other of the hyperplane. To show this we begin by obtaining a training sample *S*. This sample must be known to be linearly separable to train the classifier. The training set must take the following form:

$$S = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\} \qquad \text{where } i = 1, \dots, l$$

Here $\boldsymbol{x}_i$ represents a *p*-dimensional vector or a variable with *p*-observations. $\boldsymbol{y}_i$ is the assigned class or "truth" and will take the value of either *1 or -1* [22]. To define the hyperplane that best separates the two classes, namely class *1* and *-1*, we first define the point lying on the hyperplane, H, with the following condition:

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

Where $\boldsymbol{w}$ is the vector normal to the hyperplane and the distance from the hyperplane to the origin is defined as $\frac{|b|}{||\boldsymbol{w}||}$. We now define the margin of this hyperplane as the minimal distance between the hyperplane and the closest set of $\boldsymbol{x}_i$. If, for example, we denote a set of variables $\boldsymbol{x}_i$ as $+\boldsymbol{x}$ corresponding to $y = 1$, $-\boldsymbol{x}$ corresponding to $y = -1$, we can assign them a corresponding distance $+d$ and $-d$ from the hyperplane. With this, we can then define the margin as $(+d) +$

$(-d)$. The margin results in two additional hyperplanes H_1 and H_2. The first hyperplane is found by the following condition:

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b_1 = 1 \text{ where } b_1 = \frac{|1-b|}{||\boldsymbol{w}||}$$

Similarly, for the second hyperplane we have:

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b_2 = -1 \text{ where } b_2 = \frac{|-1-b|}{||\boldsymbol{w}||}$$

Notice that both hyperplanes H_1 and H_2 are parallel to H. The margin can then be defined as $\frac{2}{||\boldsymbol{w}||}$ which can be maximized by minimizing $||\boldsymbol{w}||$. The training data is then assumed to be subject to the following constraint:

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \geq 1 \qquad \text{for } \boldsymbol{x}_i \text{ values corresponding to } y_i = 1$$

$$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \leq -1 \qquad \text{for } \boldsymbol{x}_i \text{ values corresponding to } y_i = -1$$

We are now left with the task of minimizing $||\boldsymbol{w}||$, this can be done with the use of Lagrange multipliers in the following fashion [21][22]:

$$L = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{l} \alpha_i y_i (\boldsymbol{x}_i \cdot \boldsymbol{w} + b) + \sum_{i=1}^{l} \alpha_i$$

The derivative is then taken with respect to $\boldsymbol{w}$ and b, resulting in the two conditions below.

$$\boldsymbol{w} = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{x}_i \qquad \text{and} \qquad \sum_{i=1}^{l} \alpha_i y_i = 0$$

With this, we can make use of Karush-Kuhn-Tucker's complementary conditions to find the solutions. There will exist a Lagrange multiplier for each $\boldsymbol{x}_i$. However, the support vector will be characterized as those whose $\alpha_i > 0$ which lie on one of the hyperplanes H_1 or H_2 [22]. In the image that follows a solid line black line denotes the hyperplane H, and the dashed lines hyperplanes H_1 and H_2. The points whose Lagrange multiplier is greater than zero are circle.

Figure 2.6 Example of LSVM for a Seperable Case

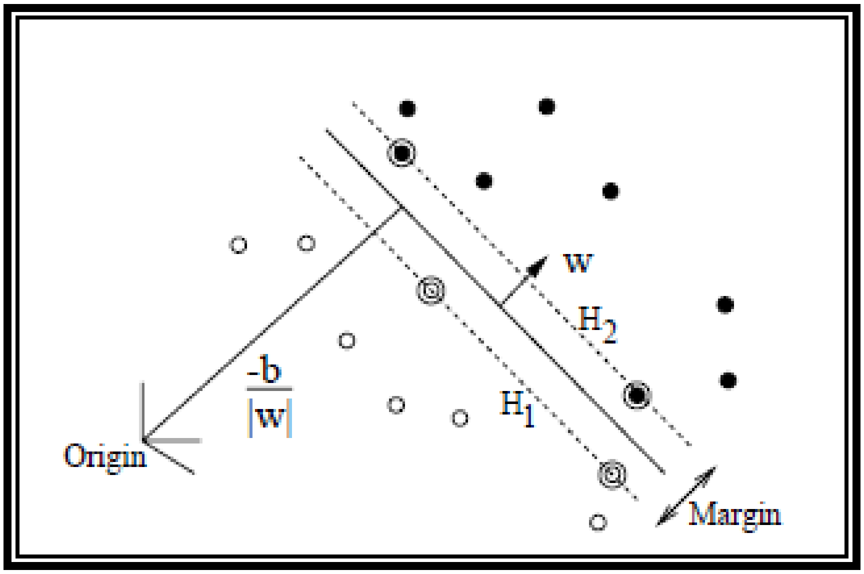

Figure 2.6: An example of a linearly separable data set. Here the support vectors have are denoted by a circle around them [22].

It is important to note that no point lies between the hyperplanes H_1 and H_2, this is due to the conditions set forth by the Maximal Margin Classifier.

Similarly, we use a training data set for a Non-Separable Case. This time, however, the constraints will change as follows:

$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \ \geq 1 - \ \varepsilon_i$ for $\boldsymbol{x}_i$ values corresponding to $y_i = 1$

$\boldsymbol{x}_i \cdot \boldsymbol{w} + b \ \leq -1 + \ \varepsilon_i$ for $\boldsymbol{x}_i$ values corresponding to $y_i = -1$

Here, ε_i corresponds to an error made at $\boldsymbol{x}_i$, This allows for some misclassification that is to be determined by the user. The Non-Separable case is also referred to as a Soft Margin Classifier because of ε_i that was introduced [21] [23].

Linear Support Vector Machines are known to be effective classifiers with an additive advantage of being able to handle high dimensionality data [4]. To test such a classifier we are only left with determining on which side of the hyperplane H (or decision boundary, our test data is on. It should also be noted that the Non-Separable case or the use of Non-linear Support Vector Machines are more likely to be used in real-world applications [21][22].

2.4 Random Forest

Random Forest is yet another common method used for classification and regression, within the realm of supervised statistical learning (for this thesis I will focus on classification). A random forest is constructed by a set of random decision trees or classification trees. A classification tree is "grown" by asking a series of ordered questions, where each successive question depends on the answers to the previous question. Decision trees begin with a root node containing the entire set of variables found in the learning set L. Nodes can take the form of a terminal or nonterminal node. For a nonterminal node, a binary split can occur when the Boolean question is used to determine if the condition is satisfied or not.

Example of Boolean question: is $x_i \leq \propto_i$?

Where $\propto_i$ is some threshold value determined by the user. The binary split then results in two daughter nodes. A terminal node, on the other hand, is a node that cannot be split and results in a classification label. An example of a classification tree is given below, Figure 2.7, where the terminal nodes (5) are marked with $\tau_1 - \tau_5$.

Figure 2.7 Decision Tree map

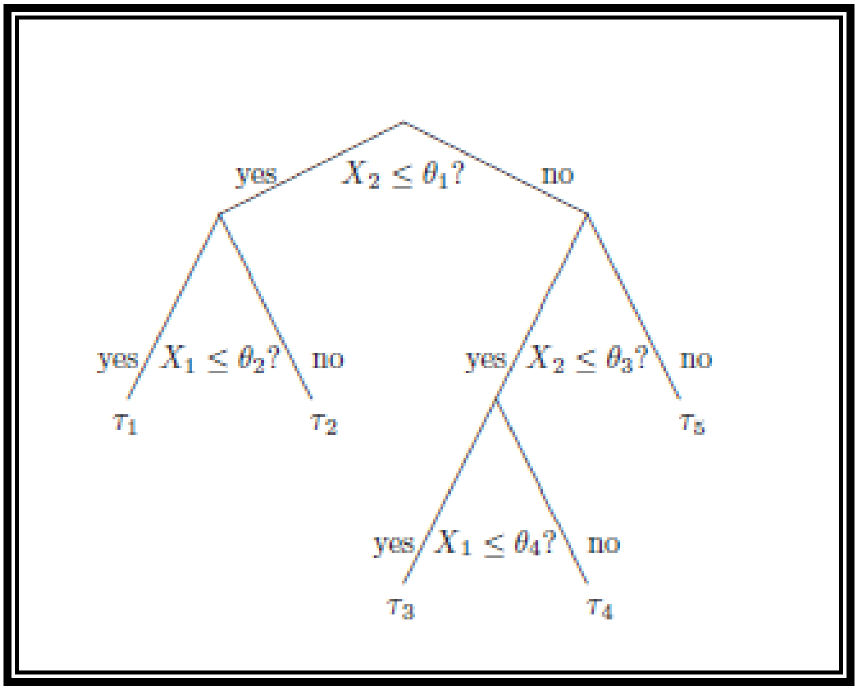

Figure 2.7: Example of a decision tree [19].

These decision trees are obtained taking bootstrapped training samples from the learning set, as in the Bagging technique [11]. This method, however, improves upon the Bagging method by the decorrelation and averaging of the decision trees [26]. The general outline for the implementation of a random forest is to first obtain a bootstrap sample from the learning set $\boldsymbol{L}$. From this sample, a tree classifier is grown using a set of m randomly chosen predictors [19] [11]. For classification purposes, m takes the value of $\sqrt{p}$ or a minimum of one, where p is the number of inputs. For regression, m takes the value of $\frac{p}{3}$ or a minimum of five [26]. The predictors serve as split candidates from which one is chosen to split the node or tree. It is important to note that when the set of m values is selected they are not removed from the learning set. This means that the m value (determined according to the intended purpose of classification or regression) remains constant throughout the process [18]. From the m candidates, the best spit is determined by the

Gini or entropy index. A series of random vectors $\boldsymbol{v}_i$ is generated from each classification tree which is independent of previous vectors. This independence is a result of using the entire learning set to generate the trees (m values chosen are not removed from L). A classifier $h(\boldsymbol{x}, \boldsymbol{v}_i)$ is then generated by the vector $\boldsymbol{v}_i$, where $\boldsymbol{x}$ is an observation. The set of tree-like structures $\{h(\boldsymbol{x}, \boldsymbol{v}_i)\ \}$ is called a random forest, which determines the class assigned to $\boldsymbol{x}$. To complete the random forest, the Out-of-Bag error is estimated by averaging the error frequency found for observations predicted using the trees. This out-of-bag error is regarded as an internal validation because, in principle, not all the variables chosen from the learning set were also used to create the trees [29]. Once the forest is grown to the desired size and the out-of-bag error is determined, it can be applied to test data for classification.

Chapter 3: Proposed Classification Method

3.1 Overview of Method

Here I will propose an unsupervised "observation" reduction technique intended for the classification of two classes. This method relies on the systematic "observation" or subject reduction in a data set based on variance held by the principal components of a matrix.

I begin with some data set $\boldsymbol{A}$, consisting of n observations and p variables. Then decompose this matrix using singular value decomposition, as described in the previous chapter, to compute the principal components of this matrix. Then extract m principal components that contain approximately 90% of the variance in the matrix.

Once the first m principal components have been extracted, the components are then sorted in descending order while simultaneously indexing them. Performing the indexing along the rows of $\boldsymbol{P}$ because they correspond to the subjects or observations, meaning that the observations with the highest variance would be at the top of the sorted matrix and the observations with the least variance at the bottom. This is a critical point in the analysis as I propose that the observation(s) with the least amount of variance should be removed from the dataset to improve the classification. The reasoning is as follows because a low variance is associated with poor separability, eliminating the observation(s) based on the principal components should, in turn, improve the separability and the classification of the data set. This results in the identification and elimination of outliers.

Once the principal component(s) with the lowest variance is identified the indexing can then be used to determine what observation(s) should be removed from the original data set. This process is iterated until the improvement no longer occurs.

3.2 Experimental Design

The data set employed in the following analysis has been obtained from the *Genome Expression Omnibus*. Here one can find gene expression data on a variety of subjects and diseases. Data sets obtained from this database are MIAME and MINSEQ compliant, which means they contain experimentally relevant information about the subjects. Because of this, my initial task is to remove information that is not relevant for classification purposes. The gene expression data comes in a form of matrix arrays where the columns represent subjects or observable and the row contained the expressions which were measured. To verify the classification improvement, I chose a data set that included the subject's or observable's classification. For example, I would retain a "classification" row that stated whether the sample was a tumor or non-tumor tissue sample. I opted for a numerical representation of "1" and "0" for tumor and non-tumor classes, respectively. This is to simplify the identification of subjects when determining the classification accuracy. This row, however, is removed before the principal components are calculated so that this technique remains *unsupervised*. I will label this matrix as $\boldsymbol{A}'$ to distinguish it from the matrix I will be processing. This matrix, $\boldsymbol{A}'$, is an $(p+1) \times n$ containing n observables as columns and p variables as rows with the additional class row.

Once the gene expression matrix is ready, I then proceed to import the matrix to MATLAB, the software I will be using to perform the numerical calculations. Before the matrix is processed I take the transpose of $\boldsymbol{A}'$, which results in a $n \times (p+1)$ matrix which now has the form required by PCA. I then remove the column vector containing the observation's class values resulting in the previously mentioned $\boldsymbol{A}$ matrix. Now that the gene expression matrix is properly formatted I calculated the principal component matrix $\boldsymbol{P}$**,** as described in the first section of Multivariate Methods using MATLAB. To determine the number of m principal components required to describe the system I used the *Cumulative Percentage of Total Variation* method. Where my variance threshold was between 70 – 90% of variance as suggested by Jolliffe, 1986 [12]. This resulted value of m = 1 principal component(s), where the first principal component accounted for approximately 89.65% of the variance in the data set tested. With this principle component, I determined the observation(s) with the least amount of variance. I did this by extracting the first column of the matrix $\boldsymbol{P}$ corresponding to the first component and sorting the column in descending order. When using the sorting command, I also included an indexing feature to identify the observation with the least amount of variance. Using the index value for the last principal component I then removed that observation from matrix $\boldsymbol{A}$. I repeated this process until the classification is no longer improved.

Sorting the $\boldsymbol{P}$ matrix is a critical part of this proposed method as I argue that the observation(s) with the least amount of variance can be treated as *outliers* which if removed should, in turn, improve the separability of the data set and consequently improve the classification.

To determine the validity of observation reduction I compared the classification results with the one provided by a standard Linear Support Vector Machine application found in MATLAB. Using the first two principal components of the $\boldsymbol{P}$ matrix and the class column of the transposed matrix $\boldsymbol{A'}$, I set up a new matrix $\boldsymbol{Z}_i$ for each iteration, $i = 1, 2, ...$, performed. Each matrix $\boldsymbol{Z}_i$ was classified using the application mentioned above to determine the accuracy improvement. This is was possible because the support vector machine application provides an accuracy score for each $\boldsymbol{Z}_i$. In the following section, I will provide my experimental results where an improvement can be seen after a certain number of iterations.

3.3 Experimental Results

To test the validity of the method I have used the data set GDS4336 obtained from the Genome Expression Omnibus. This set contains the human gene expression profiles for 45 matching pairs of Pancreatic Ductal Adenocarcinoma tumor and adjacent non-tumor tissue samples, for a total of 90 samples. Using the first 1,000 gene expressions, I set up a matrix $\boldsymbol{A}'$ with dimensions (1001×90) in line with the experimental design described in the previous section. Using the procedure in the previous section, I iterated the process a total of 10 times and used Linear Support Vector Machines to determine the accuracy of the classification after each iteration. Throughout the iteration process, the first two principal components maintained an average variance of approximately 89.6%, maintaining $m = 1$ value consistent throughout the observation reduction. Table 3.1 contains my findings.

Table 3.1 GDS4336

Table 3.1: GDS4336 dataset

Iteration	Accuracy
1	81.1%
2	84.3%
3	85.2%
4	87.4%
5	86.0%
6	87.1%
7	85.7%
8	84.3%
9	85.4%
10	85.2%

From the table, an improvement can be seen as the process is repeated until a maximum classification score is reached at iteration 4. When the process continues to be repeated there is fluctuation in the scores in a decreasing manner. The following three figures correspond to the first, fourth, and tenth iterations plotted using linear support vector machines. The axis labeled column_2 and column_3 correspond to the first and second principal components of each iteration respectively.

Figure 3.1 LSVM plot for $i = 1$

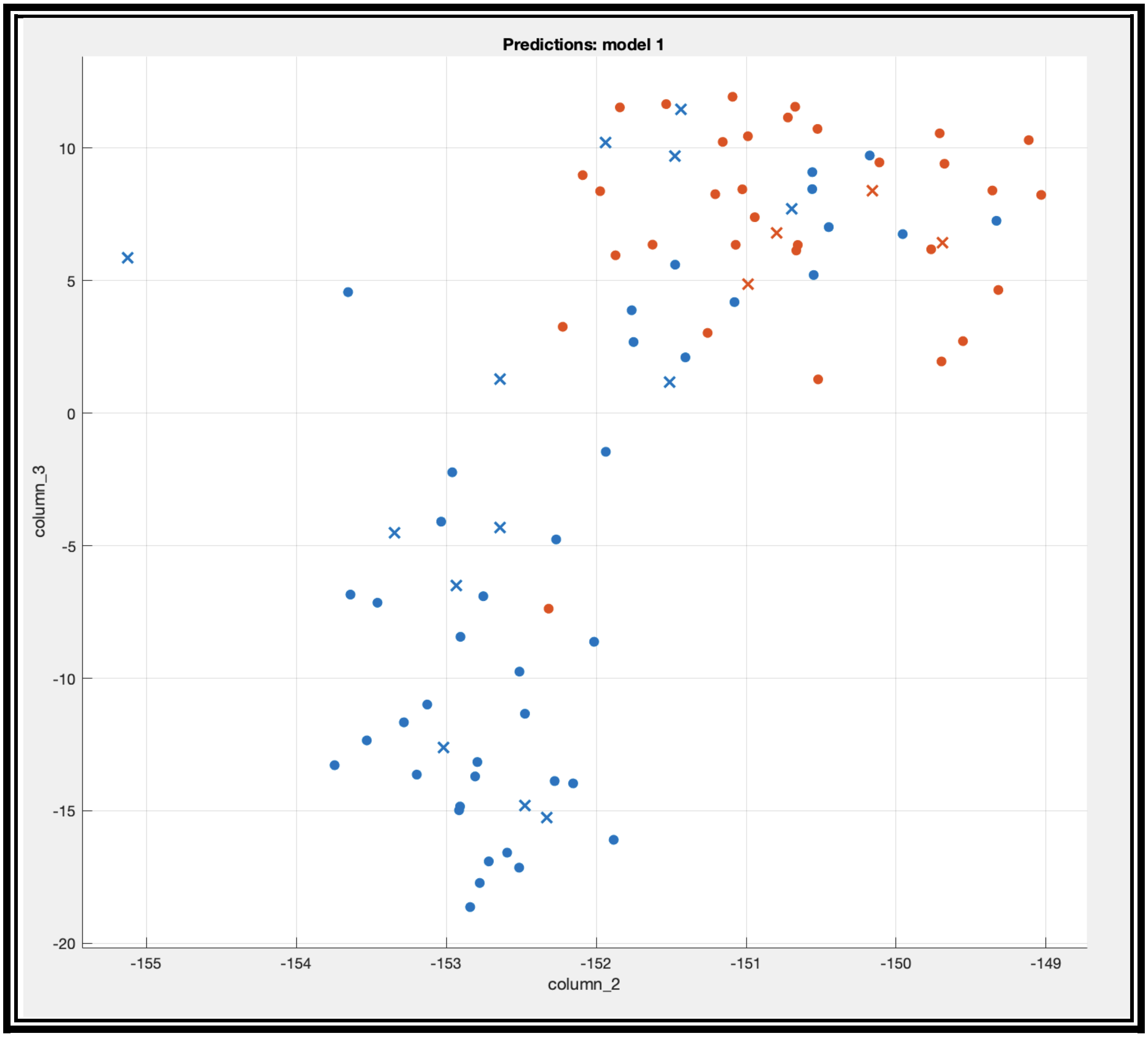

Figure 3.1: Plot containing the first two principal components for $i = 1$

Figure 3.2 LSVM plot for $i = 4$

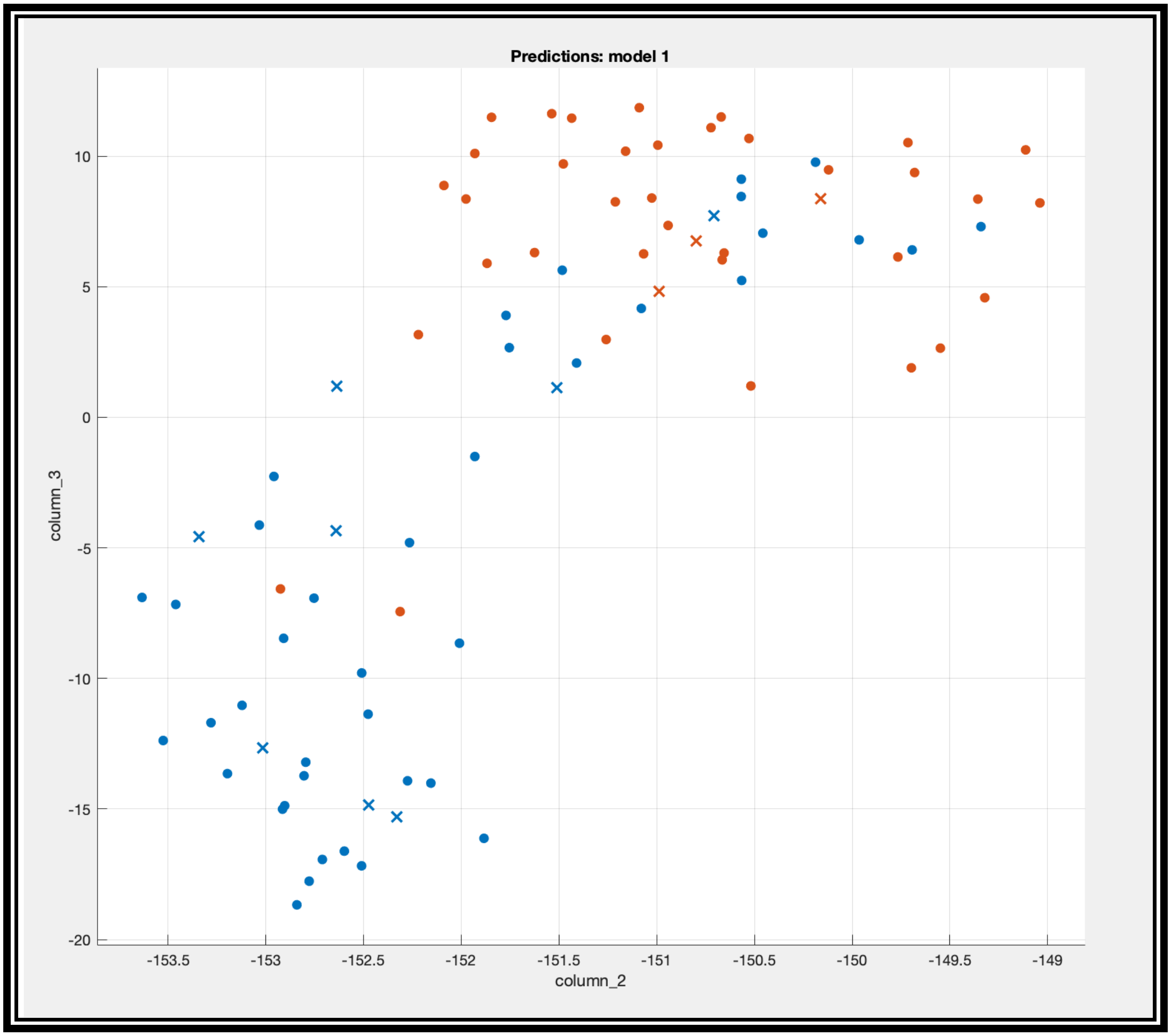

Figure 3.1: Plot containing the first two principal components for $i = 4$

Figure 3.3 LSVM plot for $i = 10$

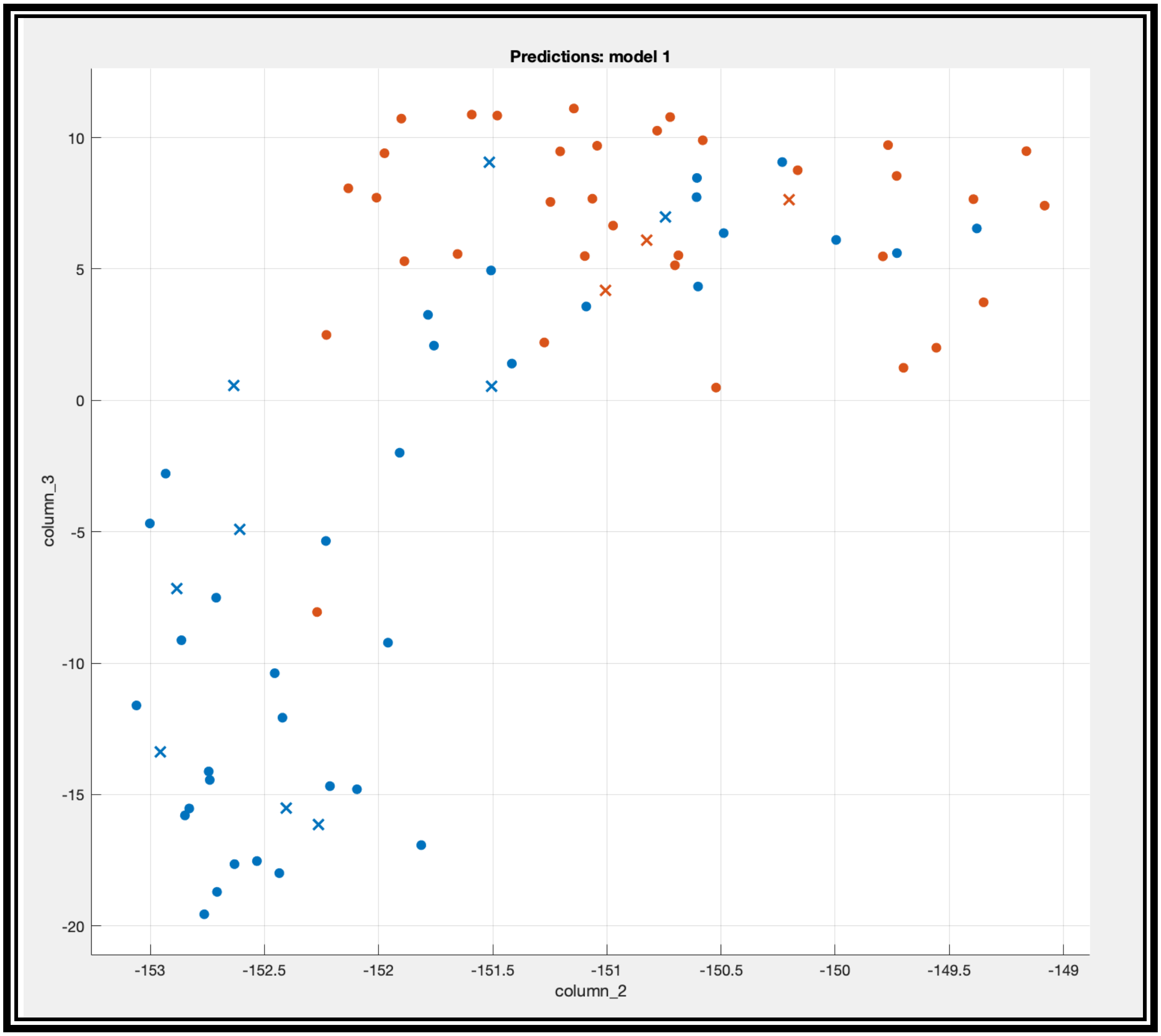

Figure 3.1: Plot containing the first two principal components for $i = 10$

Although not entirely appreciable in the figures above, the values of the elements of $\boldsymbol{P}$ change after each iteration. This was expected as the variance in the system is altered as the observations are removed.

Chapter 4: Accurate classification of normal/disease sample using Raman Confocal Microscopy

4.1 MOTIVATION

In a recent paper, our group showed that the bone samples of patients with renal osteodystrophy (ROD) exhibit an overall increase in phenylalanine and a decrease in calcium content, mineral to matrix and carbonate to matrix ratios, which can be measured by proper ratios of the area of individual Raman spectra [28]. Although one Raman spectra is clearly not sufficient to assign at statistically significant levels the samples as being either "Normal" or "ROD", the peculiarity of Raman Microscopy to collect many independent spectra from the same sample (~22500) allowed us to identify the samples with excellent accuracy ($p < 10^{-300}$). Power analysis showed that using one suitable ratio, a relatively low number of spectra (of the order of 20-50) is required to identify the ROD samples at the typically desired level of accuracy ($p=0.05$). The main goal of the work presented in this chapter is to show that the simultaneous use of all four ratios mentioned (the ones proportional with phenylalanine and calcium content, mineral to matrix and carbonate to matrix) for each individual spectra, as well as the classification methods developed by us, allows one to classify the sample at a good accuracy from only a few spectra, which raises the possibility to of in vivo detection of ROD, with a biosensor formed by a cluster of optical fibers with multiplexed Raman signal detection. The single spectra classification is performed initially by a standard Linear Discriminant Analysis with 10-fold cross-validation, and a score for each spectra is attributed based on a logit transform. The resulting confusion matrix shows that the probability of correct assignment (Normal or ROD) from a randomly selected spectra is about 80%. Statistical analysis shows that by employing only a reasonably small number of randomly selected spectra from any sample (i.e., spectra recorded at different positions from the same sample) the classification of the sample as "Normal" or "ROD" can be obtained at any desired degree of accuracy. The advantage of this procedure is that it takes into account explicitly the known physical differences between the "Normal" and "ROD" samples, and the classification is performed using only four variables, which reduce the potential impact of multicomparison correction analysis (e.g., Bonferoni [] or Hochberg-Israeli []) on the final p-value. Finally, all the information contained in the spectra is used in the classification using a statistical learning algorithm; the dimensionality is reduced using Principal Component Analysis and 20 directions of most variations are classified via Support Vector Machines algorithm implemented in MATLAB. The later classification, although employing much more information (1024 variables), it is only marginally superior to the 4-variables approach.

4.2 Preliminary data analysis

Each spectra has a linear background individually subtracted (between 377 cm $^{-1}$ and 1720 cm^{-1}) and is normalized to the laser line (each of the laser lines is normalized to have the same integral area for all the spectra after the background subtraction). The integral intensity of the relevant bands are calculated as follows: between 395 and 469 cm^{-1} for the $\nu_2PO_4^3$ band centered at about 430 cm^{-1}, between 907 and 990 cm^{-1} for the $\nu_1PO_4^3$ band centered at about 960 cm^{-1}, between 970 and 1040 cm^{-1} and also between 1574 and 1543 cm^{-1} for the two phenylalanine bands centered at 1005 and 1609 cm^{-1}, respectively, between 395 and 469 cm^{-1} for the $\nu_2PO_4^3$ band centered at about 430 cm^{-1}.

The four ratios involved in the analysis are calculated for each of the individual bone samples (three corresponding to "Normal" cells and four corresponding to "ROD"). A common supervised learning statistical analysis, Linear Discriminant Analysis using logit classification is performed for each spectra. Alternatively, all the information contained in spectra is performed by dimensionality reduction to the most relevant 20 variables, using Principal Component Analysis, followed by Linear Support Vector Machine classification using 10 fold validation. The reason for the prior dimensionality reduction is to reduce the computing time devoted to the SVM algorithm.

4.3. Results and Discussion

The integral spectra (after background subtraction and laser line normalization) for the "Normal" and "ROD" samples are presented in Figure 4.1. Although differences (particularly in the spectral region corresponding to phenylalanine) can be observed, it should be noted that each integral spectra is an average over 22500 individual spectra measured for each sample and therefore have excellent statistics.

Figure 4.1 Integral Raman Spectra

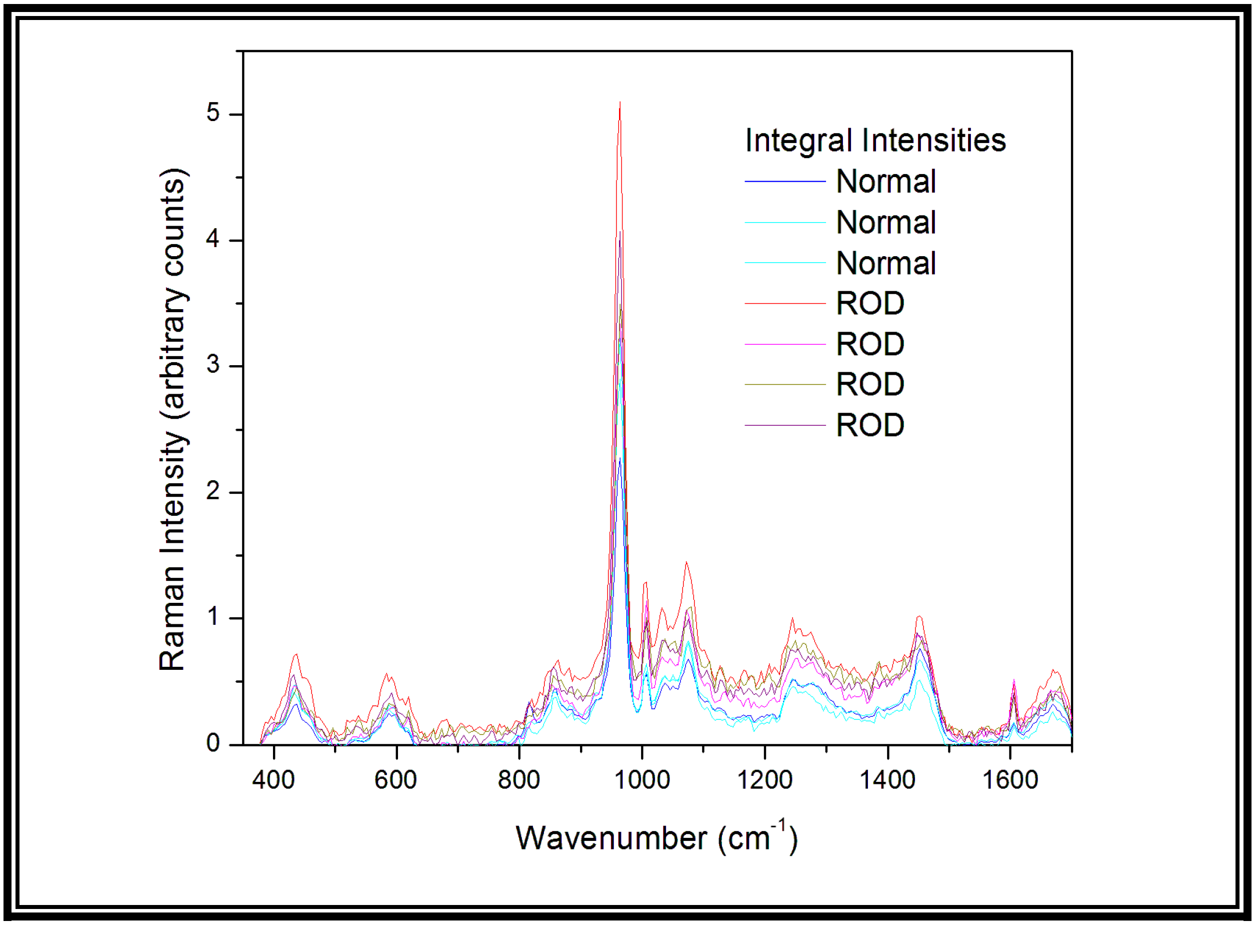

Figure 4.1: Integral Raman spectra (each averaged over 22500 spectra with the background individually subtracted) of 7 samples (three "Normal" and four "ROD")

The situation can be understood easier from the Figure 4.2a and 4.2b, in which the average, as well as 1-sigma ellipsoids, are plotted for the Ratio 1 vs Ratio 2, and Ratio 3 vs Ratio 4, respectively. It is clear that the average over 22500 spectra (the filled circles) can be easily used for the discrimination between "Normal" and "ROD" samples, particularly when looking at phenylalanine Ratio 4; however, by using only individual spectra, the classification is much poorer.

Figure 4.2a Ratio 2 vs. Ratio 1

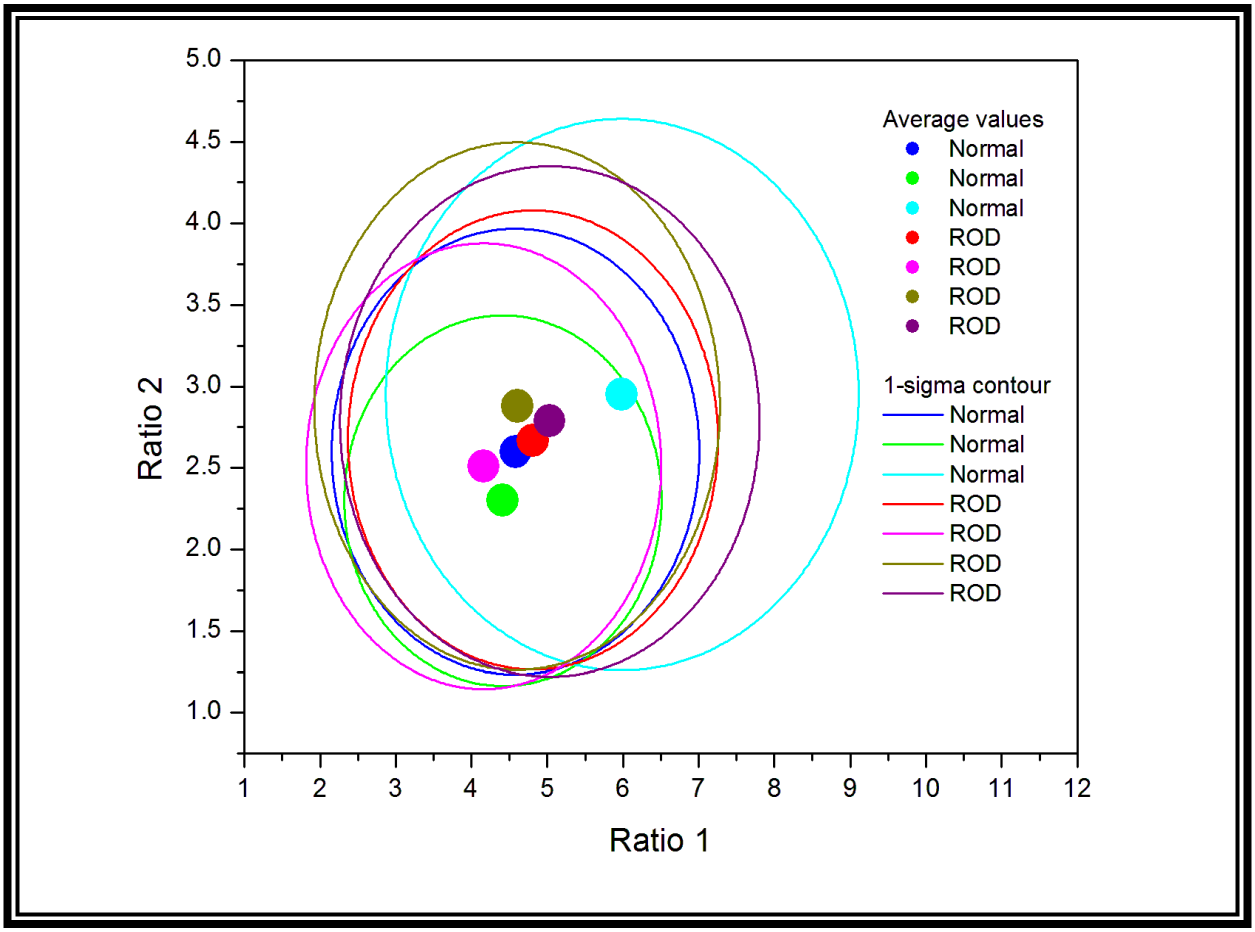

Figure 4.2a: Ratio 2 of the individual Raman spectra vs. Ratio1.

Ratio 1 is the ratio between the integral areas of the bands centered around 960 cm^{-1} ($\nu_1 PO_4^{3}$) and 1660 cm^{-1} (amide I) respectively, shown to be associated with the mineral-to-matrix content of the samples. Ratio 2 is the ratio between the integral areas of the bands centered around 1074 cm^{-1} (carbonate) and 1660 cm^{-1} (amide I), proportional to the carbonate-to-matrix-content of the sample, associated with bone modeling and turnover rate. The circles are the average values for each sample and the semi-axes of the ellipsoids are equal to one standard deviation.

Figure 4.2b Ratio 4 vs Ratio 3

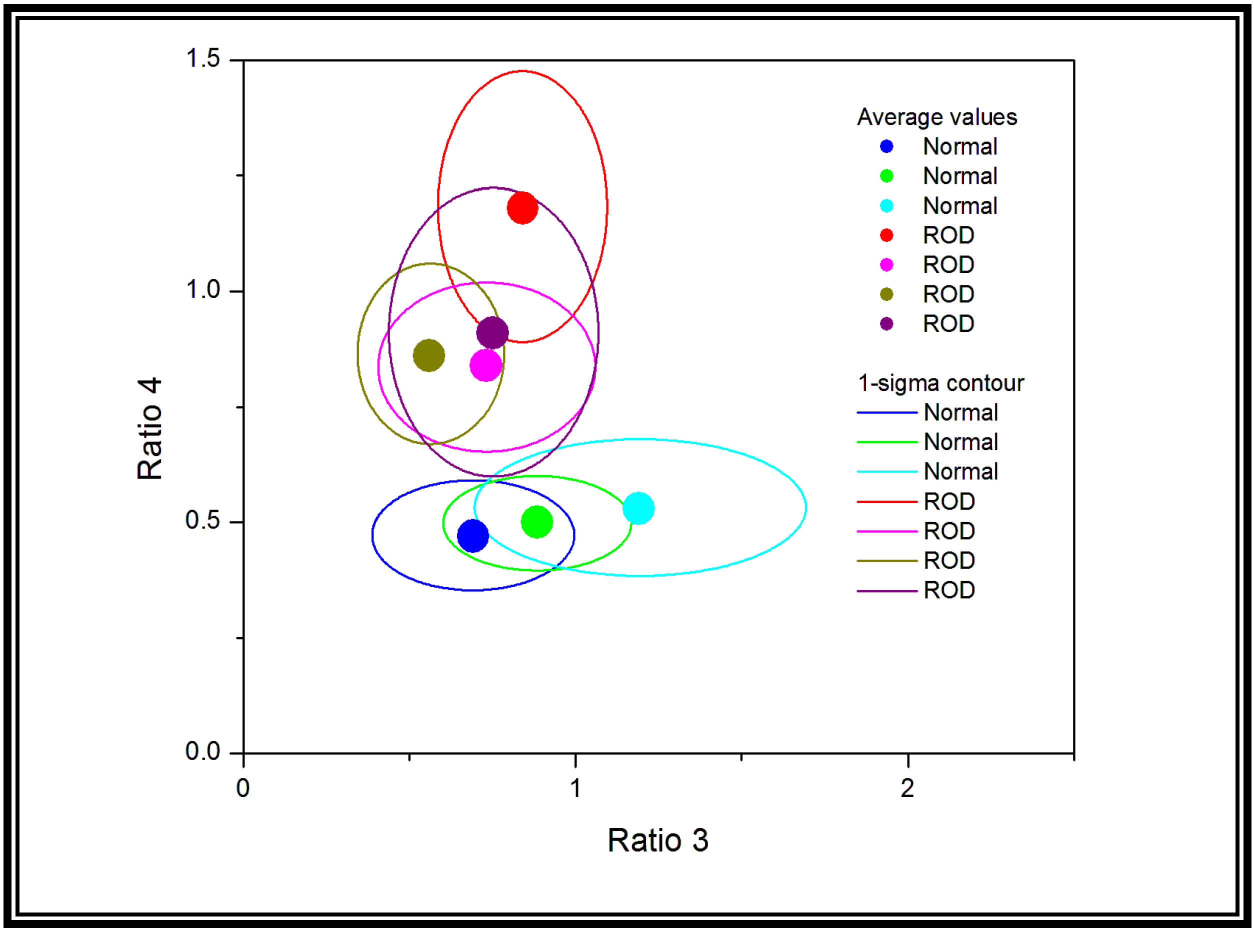

Figure 4.2b: Ratio 4 of the individual Raman spectra vs. Ratio3.

Ratio 3 is the ratio between the integral areas of the bands centered around 430 cm^{-1} ($\nu_2PO_4^3$) and 1275 cm^{-1} (amide III) respectively, shown to be associated with the calcium content of the samples. Ratio 4 is the ratio between the integral areas of the bands centered around 1005 cm^{-1} and 1609 cm^{-1} (phenylalanine) and 1660 cm^{-1} (amide I), proportional with the phenylalanine content of the sample. The circles are the average values for each sample and the semi-axes of the ellipsoids are equal to one standard deviation.

To visualize this difficulty, In Figures 4.3a and 4.3b the same ratios are plotted for individual spectra, and it is much more difficult to assign a spectra as belonging to a "Normal" or a "ROD" cell with reasonable accuracy. Therefore, whereas it is apparent that 22500 measurements for each cell are more than enough to assign the cell to "Normal" or "ROD" status with excellent accuracy, the main focus of this work is to determine the minimum number of spectra required to have the sufficient (desired) accuracy.

Figure 4.3a Ratio 2 vs Ratio 1 for individual spectra

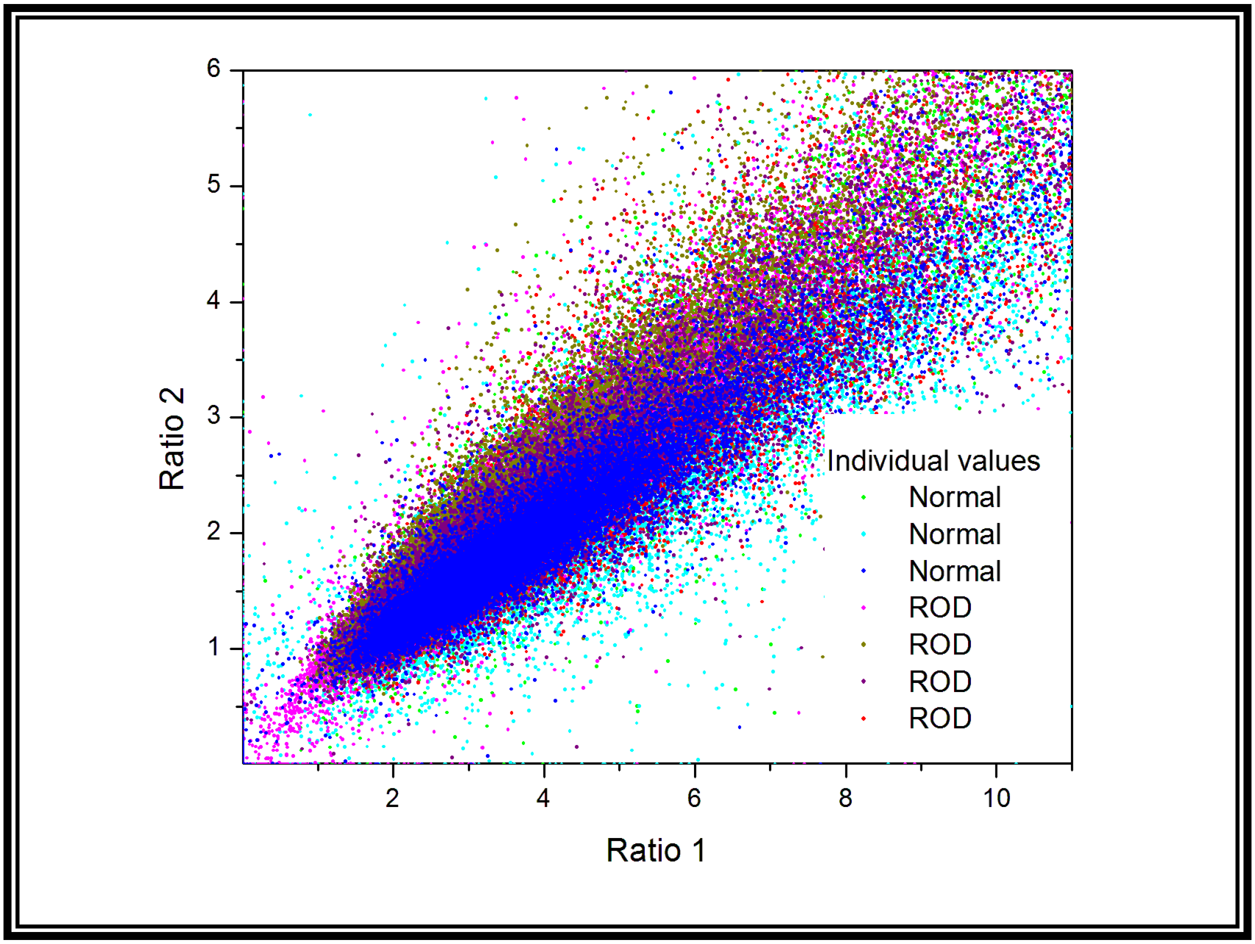

Figure 4.3a: Values of Ratio 2 vs. Ratio 1 for individual spectra.

Figure 4.3b Ratio 4 vs Ratio 3 for individual spectra

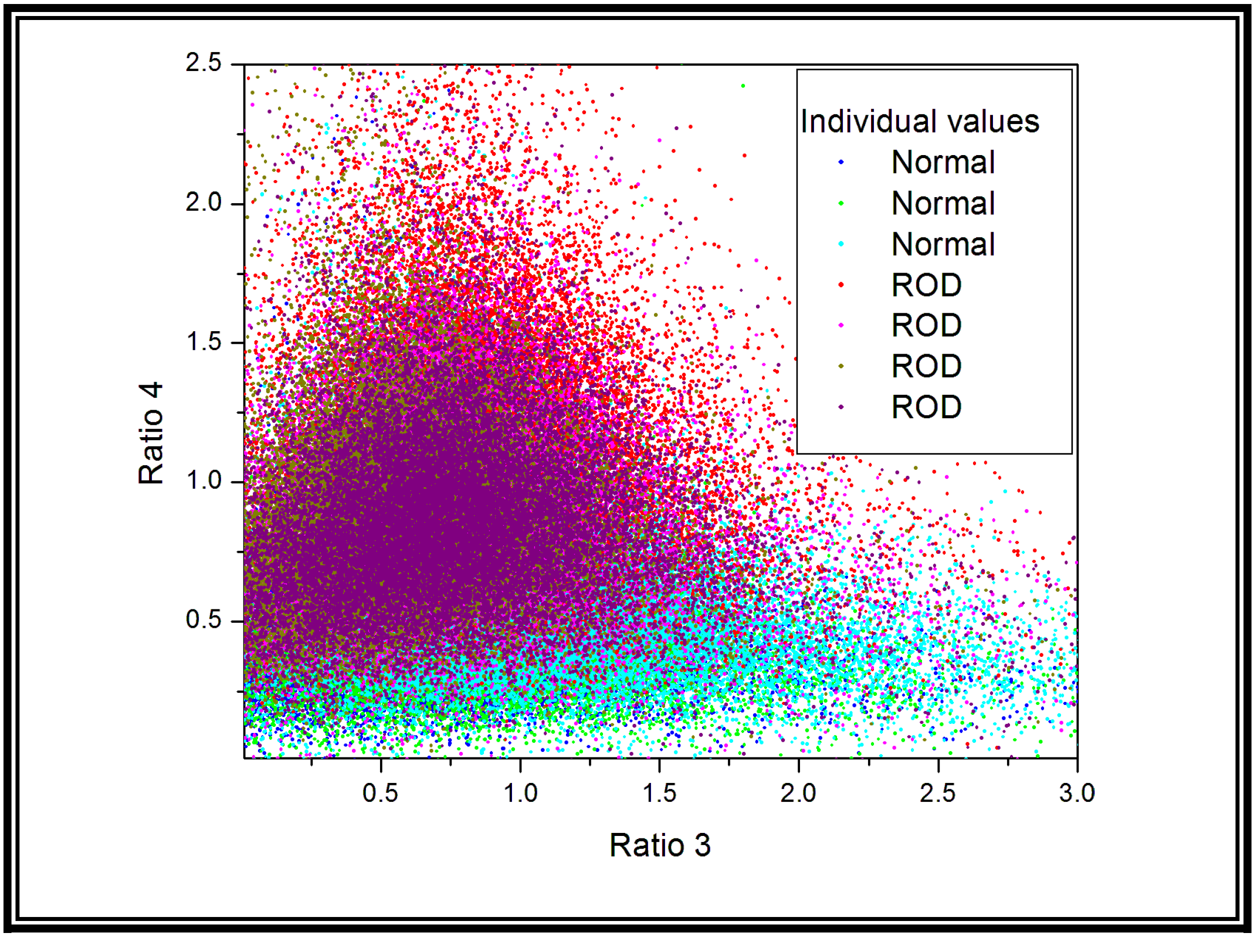

Figure 4.3b: Values of Ratio 4 vs. Ratio 3 for individual spectra.

Linear discriminant analysis with 10-fold cross-validation of the training data was employed on all the individual spectra (using as variables the four mentioned ratios), and the prediction classification has been performed using a logistic score transformation (a score less than one is likely to correspond to a spectra from a "Normal" cell and larger than one to a spectra from an "ROD" cell". The histogram of the results is presented in Figure 4.4 and the confusion matrix with the usual parameters related to prediction ability (based on only one spectra selected at random) are shown in Table 1.

Figure 4.4 Distribution Score for “Normal” and “ROD”

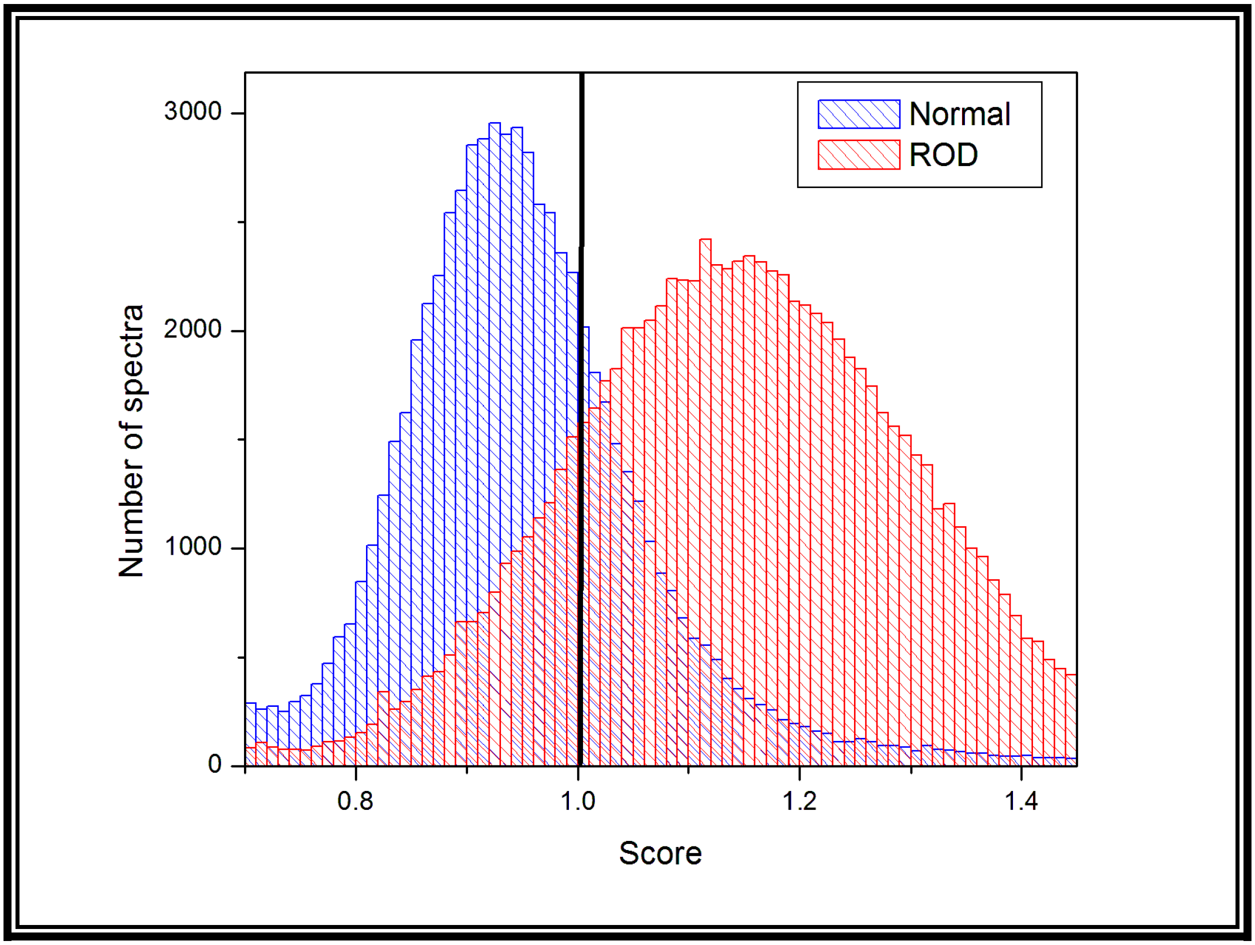

Figure 4.4: Distribution of scores for "Normal" and "ROD" individual spectra; classification assumes a score of less than one for "Normal" and larger than one for "ROD" samples. The complete details of the confusion matrix and related parameters for individual spectra are provided in Table 4.1.

Table 4.1. Confusion matrix for single spectrum LDA classification (4 variables).

	Condition Positive	Condition Negative	Prevalence 57.14%	Accuracy 80.5%
Prediction positive	70470	11205	Precision 78.3%	FDR (false discovery rate) 16.6%
Prediction negative	19530	56295	FOR (false omission rate) 21.7%	NPV (negative predictive value) 78.3%
	Sensitivity 78.3%	Specificity 83.7%	FPR (false positive rate) 16.6%	FNR (false negative rate) 38.9%

An alternative statistics approach is to employ a Support Vector Machine algorithm for classification (by employing all the variables contained in the Raman spectra, namely each photon counts recorded at all the frequency measured). The corresponding confusion matrix and the related probabilities are revealed in Table 4.2. Whereas this approach involves a number of independent variables larger by about two orders of magnitude that the preceding approach, the classification is only marginally improved, which point to the fact that the four variables chosen from physical reasoning contain most of the differences between "Normal" and "ROD" spectra.

Table 4. 2. Confusion matrix for single spectrum LSVM classification (~300 variables).

	Condition Positive	Condition Negative	Prevalence 57.1%	Accuracy 87.5%
Prediction positive	70470	9112	Precision 88.3%	FDR (false discovery rate) 13.5%
Prediction negative	10530	58388	FOR (false omission rate) 11.7%	NPV (negative predictive value) 88.3%
	Sensitivity 88.3%	Specificity 86.5%	FPR (false positive rate) 13.5%	FNR (false negative rate) 15.6%

To examine the minimum number of spectra required to classify an unknown sample, we assume that N spectra are measured (with N being an odd integer); if n>N/2 spectra have a score larger than one, the sample is assigned to "ROD", and in the other case to the normal. Given the probability p_1 than a "Normal" spectra has the score less than 1 and p_2 than a ROD spectra has a score larger than 1 (see Table 4.1), the probabilities for Type I (rejection of a true null hypothesis, or false positive), Q_I (N), and Type II error (non-rejection of a false null hypothesis, or false negative) Q_{II} (N) can be calculated from:

$$Q_1(N) = 1 - P_1(N) \quad = \sum_{k=0}^{k<\frac{N}{2}} \binom{N}{k} (1-p_1)^{N-k} {p_1}^k \qquad (1)$$

$$Q_2(N) = 1 - P_2(N) \quad = \sum_{k=0}^{k<\frac{N}{2}} \binom{N}{k} (1-p_1)^{N-k} {p_1}^k \qquad (2)$$

which represents the probability that a number of k=N, N-1, ...k< N/2 spectra obtained from a "Normal" sample are wrongfully assigned, respectively that the k=N, N-1, ...k< N/2 obtained from a "ROD" sample are wrongfully assigned.

In Figure 4.5, the probability of Type 1 and Type II assignment error are plotted as functions of the number of independent spectra recorded; whereas the large figure indicates that the probability of assignment error can be made as small as desired if a sufficient number of sampling point are used, the inset shows that for a typically desired precision, a number of about 10 independent spectra are sufficient.

Figure 4.5 Probability of Type I and Type II erors vs. Number of Spectra

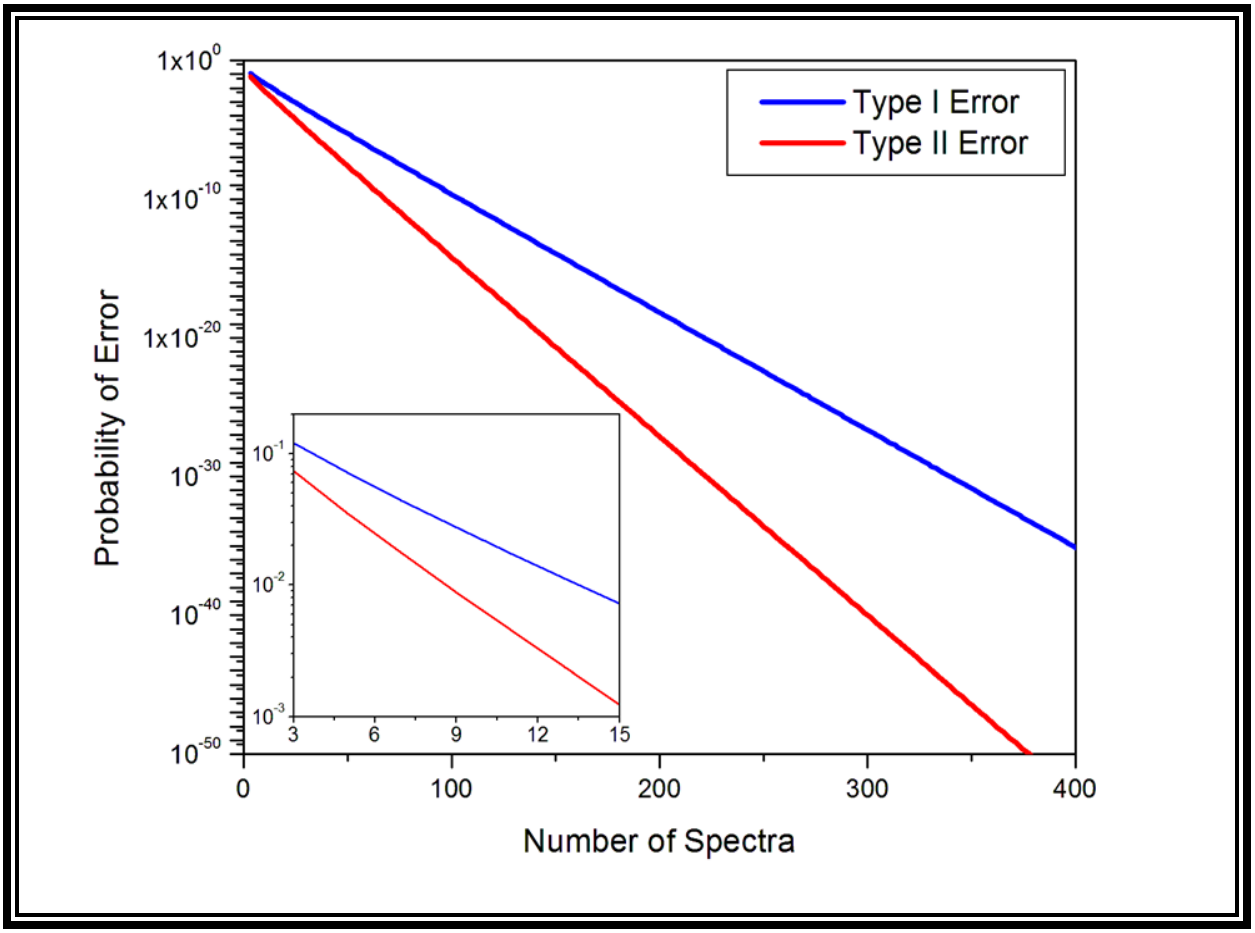

Figure 4.5: Probability of type I and type II errors vs. the number of randomly chosen spectra employed in classification.

In the inset, it is shown that a relatively small set of spectra measured at different positions of a sample is sufficient for classification with typical accuracy (e.g., $p<0.05$).

The confusion matrix and the related classification parameters for using 11 independent spectra (taken from different positions) are presented in Table 4.3.

Table 4.3. Confusion matrix for 11 spectra classification.

	Condition Positive	**Condition Negative**	**Prevalence 57.1%**	**Accuracy 98.8%**
Prediction positive	**98.3%**	**0.5%**	**Precision 98.3%**	**FDR (false discovery rate) 0.5%**
Prediction negative	**1.7%**	**99.5%**	**FOR (false omission rate) 0.0174%**	**NPV (negative predictive value) 98.3%**
	Sensitivity 78.3%	**Specificity 99.5%**	**FPR (false positive rate) 0.5%**	**FNR (false negative rate) 2.3%**

Since a sensor that can record such a small number of independent spectra is in principle feasible (e.g. by multiplexing the recorded Raman signal on a bunch of optical fiber), the present work supports the utility of creating such a sensor.

Chapter 5: Conclusions

The main goal of this work was to develop statistical approaches that improve the classification, particularly for low sample number data. In Chapter 3, it was shown that our method, to use a restricted set of vectors for orthogonal projections, has the ability to improve significantly over traditional PCA and LDA analysis. The method has been extended and applied to classify Raman Confocal Microscopy data. It was shown that, whereas traditional methods can offer excellent classification for large sample numbers (of the order of 20000 spectra), our method can provide a very good classification for a much smaller data set (of the order of 10 spectra). This raises the possibility of constructing Raman sensors for accurate in vivo detection of various diseases.

References

[1] Jain, Kewal K., and Kewal K. Jain. The handbook of biomarkers. New York: Springer, 2010.

[2] Kelly, Jemma G., et al. "Biospectroscopy to metabolically profile biomolecular structure: a multistage approach linking computational analysis with biomarkers." Journal of proteome research 10.4 (2011): 1437-1448.

[3] Henry, N. Lynn, and Daniel F. Hayes. "Cancer biomarkers." *Molecular oncology* 6.2 (2012): 140-146.

[4] Almugren, Nada, and Hala Alshamlan. "A survey on hybrid feature selection methods in microarray gene expression data for cancer classification." *IEEE Access* 7 (2019): 78533-78548.

[5] Strimbu, Kyle, and Jorge A. Tavel. "What are biomarkers?." *Current Opinion in HIV and AIDS* 5.6 (2010): 463.

[6] Jombart, T., D. Pontier, and Anne-Béatrice Dufour. "Genetic markers in the playground of multivariate analysis." Heredity 102.4 (2009): 330-341.

[7] Tinker, Anna V., Alex Boussioutas, and David DL Bowtell. "The challenges of gene expression microarrays for the study of human cancer." Cancer cell 9.5 (2006): 333-339.

[8] Mwangi, Benson, Tian Siva Tian, and Jair C. Soares. "A review of feature reduction techniques in neuroimaging." Neuroinformatics 12.2 (2014): 229-244.

[9] Smith, Ewen, and Geoffrey Dent. Modern Raman spectroscopy: a practical approach. John Wiley & Sons, 2019.

[10] Golub, Todd R., et al. "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring." science 286.5439 (1999): 531-537.

[11] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An introduction to statistical learning. Vol. 112. New York: springer, 2013.

[12] Jolliffe, I. T. "Principal component analysis. 1986."

[13] Fisher, Ronald A. "The use of multiple measurements in taxonomic problems." Annals of eugenics 7, no. 2 (1936): 179-188.; Xanthopoulos, Petros, Panos M. Pardalos, and Theodore B. Trafalis. "Linear discriminant analysis." In Robust data mining, pp. 27-33. Springer, New York, NY, 2013.

[14] Martinez, Wendy L., et al. Exploratory data analysis with MATLAB. Crc Press, 2010.

[15] Abdi, Hervé, and Lynne J. Williams. "Principal component analysis." Wiley interdisciplinary reviews: computational statistics 2.4 (2010): 433-459.

[16] Balakrishnama, Suresh, and Aravind Ganapathiraju. "Linear discriminant analysis-a brief tutorial." Institute for Signal and information Processing 18 (1998): 1-8.

[17] Hair, Joseph F., et al. Multivariate data analysis. Vol. 5. No. 3. Upper Saddle River, NJ: Prentice hall, 1998.

[18] Robotti, Elisa, Marcello Manfredi, and Emilio Marengo. "Biomarkers discovery through multivariate statistical methods: a review of recently developed methods and applications in proteomics." J Proteomics Bioinform 3 (2014): 003.

[19] Izenman, Alan Julian. "Modern multivariate statistical techniques." Regression, classification and manifold learning 10 (2008): 978-0.

[20] Fukunaga, Keinosuke. Introduction to statistical pattern recognition. Elsevier, 2013.

[21] Mammone, Alessia, Marco Turchi, and Nello Cristianini. "Support vector machines." Wiley Interdisciplinary Reviews: Computational Statistics 1.3 (2009): 283-289.

[22] Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." Data mining and knowledge discovery 2.2 (1998): 121-167.

[23] Jakkula, Vikramaditya. "Tutorial on support vector machine (svm)." School of EECS, Washington State University 37 (2006).

[24] Goel, Eesha, et al. "Random forest: A review." International Journal of Advanced Research in Computer Science and Software Engineering 7.1 (2017).

[25] Berk, Richard A. Statistical learning from a regression perspective. Vol. 14. New York: Springer, 2008.

[26] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. The elements of statistical learning. Vol. 1. No. 10. New York: Springer series in statistics, 2001.

[27] Boulesteix, Anne-Laure, et al. "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2.6 (2012): 493-507

[28] Ciubuc, J.D.; Manciu, M.; Maran, A.; Yaszemski, M.J.; Sundin, E.M.; Bennet, K.E.; Manciu, F.S. Raman Spectroscopic and Microscopic Analysis for Monitoring Renal Osteodystrophy Signatures. Biosensors 2018, 8, 38. [CrossRef] [PubMed]

[29] Manciu, Marian, et al. "Assessment of Renal Osteodystrophy via Computational Analysis of Label-free Raman Detection of Multiple Biomarkers." Diagnostics 10.2 (2020): 79.

Vita

Mario Cardenas Jr was born and raised in El Paso, Texas, where he attended The University of Texas at El Paso and earned a Bachelor of Science in Physics. During this time, he worked as a Teaching Assistant and participated in events organized by the Student Physics Society and other volunteer activities. Mario has since then co-authored a published paper titled "Assessment of Renal Osteodystrophy via Computational Analysis of Label-free Raman Detection of Multiple Biomarkers" and is currently pursuing a Ph.D. in Physics.