2014

MINING MULTI-GRANULAR MULTIVARIATE MEDICAL MEASUREMENTS

Conrad Sykes
Eastern Washington University

MINING MULTI-GRANULAR MULTIVARIATE MEDICAL MEASUREMENTS

A Thesis

Presented To

Eastern Washington University

Cheney, Washington

In Partial Fulfilment of the Requirements

For the Degree

Master of Science

By

Conrad Sykes

Spring 2014

THESIS OF CONRAD SYKES APPROVED BY

______________________________________DATE_______________

DAN LI, GRADUATE STUDY COMMITTEE

______________________________________DATE_______________

STU STEINER, GRADUATE STUDY COMMITTEE

MASTER'S THESIS

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Eastern Washington University, I agree that the JFK Library shall make copies freely available for inspection. I further agree that copying of this project in whole or in part is allowable only for scholarly purposes. It is understood, however, that any copying or publication of this thesis for commercial purposes, or for financial gain, shall not be allowed without my written permission.

Signature________________________________

Date________________________________

Abstract

This thesis is motivated by the need to predict the mortality of patients in the Intensive Care Unit. The heart of this problem revolves around being able to accurately classify multivariate, multi-granular time series patient data. The approach ultimately taken in this thesis involves using Z-Score normalization to make variables comparable, Single Value Decomposition to reduce the number of features, and a Support Vector Machine to classify patient tuples. This approach proves to outperform other classification models such as k-Nearest Neighbor and demonstrates that SVM is a viable model for this project. The hope is that going forward other work can build off of this research and one day make an impact in the medical community.

Acknowledgements

I would like to thank several people for their assistance with this thesis. First I would like to thank Dr. Li for inspiring me to take on this challenge; and for the guidance and wisdom she has provided throughout this endeavor. Without her help and insight this thesis would never have been possible. Next, I would like to thank Stu Steiner for his help navigating all the bureaucracy, paperwork, deadlines, and other politics involved with college. Without him I would have never started Grad School and certainly never finished.

Table of Contents

List of Figures

1 INTRODUCTON

The field of medical research is rich with data. Through the use of medical technology massive amounts of information about every facet of a patient's health can be recorded. This data is vital in understanding disease, developing new drugs, and preventing illness. Pure unprocessed data by itself is meaningless. Data Mining takes large data sets, finds interesting patterns and relationships, and transforms raw data into something meaningful [1]. One very useful form of data mining is "Classification". Classification takes a training data set and builds a model which can then be used to accurately classify other previously unseen data into groups [1]. Often medical data is collected over a period of time; this type of information is called "Time Series" data. Time Series data adds a level of complexity because not only must the values of the data be considered, but also the timestamp when data is collected, and the frequency of collection. Mining this data involves several steps including data selection, pre-processing, model creation, and results analysis [2].

Medical data is massive and complicated but through the process of data mining and classification this mountain of data can be transformed into meaningful information. This thesis attempts to data mine time series data as to classify mortalities in the Intensive Care Unit (ICU) at hospitals. Thirty-six (36) unique time series measurement variables and six (6) general description variables, collected during patients stay in the ICU, are used to accurately classify the survival and mortality of patients. The data being examined is multivariate and multi-granular. 'Multivariate' refers to patients having multiple variables recorded and 'multi-granular' refers to the time series variables that are recorded multiple times. The multivariate nature of the data adds a level of complexity because finding similarity measures between data becomes exponentially complicated as more dimensions (patient variables) are considered.

To complete the goal of multivariate multi-granular classification there are several objectives defined.

1. Determine if all the variables have equal significance, or some are more important than others.
2. Find a model for classifying multivariate, multi-granular time series data.
3. Decide what similarity measure should be used and what is an appropriate level of time segmentation.
4. Discover the maximum level of accuracy in classification that can be achieved with the given data set.

The contribution of this thesis is important because strategies involved with classifying data like this can be applied to similar projects in the future. Information that can be gained from this data holds the potential for giving medical professionals deeper insight into a patients stay in the ICU.

2 BACKGROUND

With the advancement in medical technology, sophisticated patient monitoring systems are now commonplace in hospitals and these systems are typically first installed into the ICU ward where patients require the closest observation [3]. While these systems are sophisticated and each records a large amount of data, the real benefit is derived from being able to mine the information gathered which can be used to increase patient care. Thus, the goal of this thesis is to classify mortalities in the ICU at hospitals, based off of statistics collected during the first 48 hours of a patient's stay. This research has been motivated by the desire to compare how well different medications work, hospital and care guidelines, and other methods of treatment [4]. Using data mining techniques and classification methods the goal is to predict with a high level of precision and sensitivity which patients will die in the ICU. This research is based off the "PhysioNet Computing In Cardiology" challenge of 2012. PhyisoNet's goal was "develop methods for patient-specific prediction of in-hospital mortality" [4].

The data used in this challenge is from 12,000 medical records of adult patients during the first 48 hours in the ICU. The patients had a wide variety of maladies including "cardiac, medical, surgical, and trauma" [4]. Forty-two variables are included in this data set although not all the data points are recorded for every patient. Additionally, six of these variables are "general descriptors" such as age, gender, and weight, and are collected on patient's entry into the ICU as shown in Table 2-1; while the remaining 36 variables are time series based as shown in Table 2-2. In addition to the 42 variables are 5 outcome-related descriptors that describe the final state of the patient including if the patient survived as shown in Table 2-3.

All of the data is provided from the Computing in Cardiology challenge and is broken into three sets. Set A is the training set and includes the outcomes, while sets B and C do not contain the outcomes therefore are used as Test Sets. Sets A, B and C each contain 4,000 records although only A and B are given to competition contestants and set C is used for judging.

Table 2-1: General Descriptors

RecordID	A unique integer for each ICU stay
Age	Years
Gender	0: female, or 1: male
Height	Cm
ICUType	1: Coronary Care Unit, 2: Cardiac Surgery Recovery Unit, 3: Medical ICU, or 4: Surgical ICU
Weight	Kg

Table 2-2: Time Series Variables

Albumin	g/Dl
ALP	Alkaline phosphatase (IU/L)
ALT	Alanine transaminase (IU/L)
AST	Aspartate transaminase (IU/L)
Bilirubin	mg/dl
BUN	Blood ureanitrogen
Cholesterol	Mg/dl
Creatinine	Serum creatinine (mg/dL)
DiasABP	Invasive diastolic arterial blood pressure (mmHg)
FiO2	Fractional inspired O2 (0-1)
GCS	Glasgow Coma Score (3-15)
Glucose	Serum glucose (mg/dL)
HCO3	Serum bicarbonate (mmol/L)
HCT	Hematocrit (%)
HR	Heart rate (bpm)
K	Serum potassium (mEq/L)
Lactate	mmol/L
Mg	Serum magnesium (mmol/L)
MAP	Invasive mean arterial blood pressure (mmHg)
MechVent	Mechanical ventilation respiration (0:false, or 1:true)
Na	Serum sodium (mEq/L)
NIDiasABP	Non-invasive diastolic arterial blood pressure (mmHg)
NIMAP	Non-invasive mean arterial blood pressure (mmHg)
NISysABP	Non-invasive systolic arterial blood pressure (mmHg)
PaCO2	partial pressure of arterial CO2 (mmHg)

PaO2	Partial pressure of arterial O2 (mmHg)
pH	Arterial pH (0-14)
Platelets	cells/nL
RespRate	Respiration rate (bpm)
SaO2	O2 saturation in hemoglobin (%)
SysABP	Invasive systolic arterial blood pressure (mmHg)
Temp	Temperature (°C)
TropI	Troponin-I (μg/L)
TropT	Troponin-T (μg/L)
Urine	Urine output (mL)
Weight	(kg)*

Table 2-3: Outcome Related Descriptors

SAPS-I score	Simplified Acute Physiology Score
SOFA score	Sequential Organ Failure Assessment
Length of stay	Days. Includes time outside of the ICU
Survival	Days after ICU stay that the patient died (if recorded and if dead)
In-hospital death	0: survivor, or 1: died in-hospital

3 RELATED WORK

The medical community uses different scoring systems which can be used to assign patients a score to access a patient's health and severity of condition. These scoring systems are a good starting point to examine the current methodologies and to determine if a more effective solution is possible. From a data mining perspective, classification models must be examined to determine the appropriate algorithm for a multivariate multi-granular data set. Finally, given the large amount of features that this data set has and the level of complexity due to its time series features, it is necessary to investigate existing feature reduction techniques.

3.1 Existing Medical Solutions

A predictive scoring system is a system which assigns a patient a score that relates to severity of the patient's illness and/or probability of death. Some common predictive scoring systems used by the medical community are Simplified Acute Physiology Score (SAPS), and Sequential Organ Failure Assessment (SOFA) [5].

A patients SAPS value is found by taking the worst values of certain variables (within a 24 time period), inputting these values into the SAPS model which was originally defined using logistic regression, and finding the appropriate weights to assign variables [5]. There are 17 different health-related variables that SAPS takes into consideration and how "worst" is defined depends on each specific variable. For example, a patients "worst" temperature is the patients highest temperature over the preceding 24-hour period [5]. The scale defined by SAPS can be used by doctors and other medical staff to gain insight into how a patients health is doing.

SOFA works in a similar manner to SAPS but it is focused specifically on organ dysfunction [6]. The SOFA score is based on six parameters Respirationa, Coagulation, Liver, Cardiovascular, CNS, and Renal. Each of these is graded on a 0 to 4 scale where 0 represents normal functionality and 4 represents abnormal behavior or failure [6]. Each variable has a predefined value range which specifies what value (0 through 4) that variables value should correlate to. After tallying all of the different parameters a minimum score of 0 and maximum score of 24 are possible. SOFA provides an easy scoring system that can give medical staff a clear view of how a patient's organs are performing.

While scoring systems like SAPS and SOFA are useful, they both utilize relatively few patient parameters, and do not take into consideration a patients change in health over time. The data used in this thesis has many different variables which SAPS and SOFA do not consider as well as time series information which allows for a much more intricate classification model to be created.

3.2 Existing Classification Methods

The challenge placed virtually no limits on how to classify this problem and thus a variety of different techniques were pursued. Since the contest already concluded it is possible to see how well these different methodologies compared.

3.2.1 Support Vector Machine

From the challenge, the team that had the best overall results used both time series and general descriptors as input to a quadratic Support Vector Machine (SVM). Since the number of patients who live after staying in the ICU greatly outweighs the number of patients who die, the winning team divided their training set into six parts. All of the dies examples from the data set were matched with an equal number of lives patient examples [7]. SVM uses a nonlinear mapping to transform data into a higher level dimension and then searches for a decision boundary, called a hyper-plane, which separates the classes [1]. SVM is known to handle noisy data and is well suited for large data sets. SVM is also excellent at handling data sets that have uninformative or redundant features [7].

The winning team created a unique idea for handling the time aspect of the data. The team broke up the 48 hour period into two 24 hour chunks. Within each 24 hour period; the minimum, mean, and maximum of every variable was calculated, and then normalized on a scale from -1 to 1 [7]. This interesting solution allows for the improvement or decline of a patient to be tracked over time, and is at a level of resolution that helps minimize the effects of missing data. Missing data is a problem, but the team alleviated this problem by filling in missing daily information from either the proceeding or following day.

3.2.2 Neural Network

Other teams took an approach of implementing a neural network [3]. A neural network is a set of nodes with associated weights. A neural network is trained by adjusting the weight of these nodes to maximize the correctness of inputs to the class labels assigned as outputs [1]. While neural networks are good at handling noisy and complicated data sets, apparently a neural network does not perform as well with this data set. One reason for this underperformance is that while training the neural network the "optimization could often be stuck in local minima and result in very poor classification accuracy" [8]. There are strategies that address this issue such as training multiple neural networks and then using a "voting strategy"; but based on the findings of participants from the challenge the neural network tended to underperform other models [3][8].

3.3 Existing Feature Reduction Methods

This data set is extremely complex and it is necessary to transform the data in some way to make it more manageable. Thirty-six of the 42 variables that each patient could potentially have are time series variables which have been sampled at various rates. The best method to transform and interpret this complex data set can be derived from examining the existing methodologies.

Discrete Wavelet Transform (DWT) takes the time/amplitude domain of data and transforms it into a time/frequency domain. “Mathematically speaking, the wavelet transform is a convolution of the wavelet function with the signal” [9]. Frequency information about a signal reveals important detail about the signal which provides a concise method for describing what is occurring. One common method for discovering frequency information about a signal is the Fourier Transform. The Fourier transform makes it possible to tell what frequencies occur within a single. If a signal is stationary, meaning that its frequency doesn’t change, then the Fourier transform is perfectly fine to use. On the other hand, if the frequencies within a signal change over the lifetime of the signal the Fourier transform will be of limited use. Unfortunately, the Fourier Transform does not give any information as to when a specific frequency occurred within a signals timeline [10].

The Short Time Fourier Transform (STFT) provides the ability to distinguish both when and what frequencies occur in a signal. STFT combines the concepts of the Fourier Transform with discrete windows of time [11]. The Heisenberg uncertainty principle states that it is impossible to know both the momentum and position of a particle in motion simultaneously, meaning it is impossible to know the precise frequency and time information of a signal at a given point. What can be found is what frequencies exist within a time range instead of at an exact moment [11]. STFT breaks the signal into windows of time and applies the Fourier Transform on those windows. Through this process of using windows a more precise idea of what signals occur during which time periods can be calculated.

While the STFT is useful it does have its drawbacks. The biggest drawback of the STFT is the window it uses for its calculations is of fixed size. The window size matters because the larger the window the better the frequency resolution, and the smaller the window the better the time resolution [11]. With a fixed sized window resolution can become a serious problem. The benefit of the DWT is that it provides Multi Resolution Analysis (MRA). With DWT filters of different frequencies are used to analyze the signal at different scales (window sizes). This enables only the important parts of a signals to be viewed and the irrelevant frequencies can be discarded; therefore simplifying the complexity [11]. By using a system with MRA, accurate information about both what and when frequencies within a signal occur can be obtained.

Before it is possible to understand how the Wavelet Transform works, a definition of what a wavelet is must be determined. A wavelet is just a function which integrates to zero. This means that if the function were graphed that as much of the function would exist above the x-axis as it does below, or in other words it's "waving" above and below the x-axis [12]. The most basic example of a wavelet is the Harr wavelet shown in Figure 3-1.

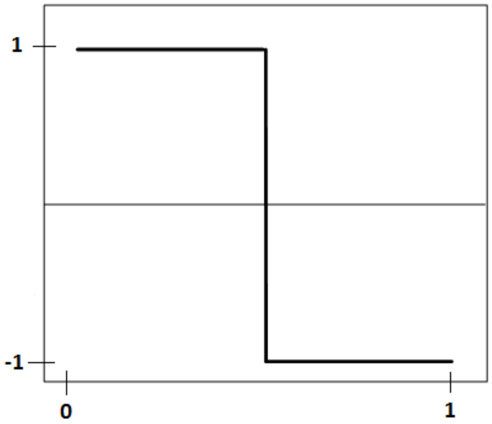

Figure 3-1: Harr wavelet

Wavelets are used as basis functions, meaning that they can approximate or represent other functions. The Harr wavelet is an example of a Mother Wavelet. A Mother Wavelet acts as a starting point for representing a function. This mother wavelet can be translated and dilated depending on what function is trying to be approximated. Through translations, dilations, and different combinations any continuous function can be approximated by Haar functions [12].

The Wavelet in the Wavelet Transform acts like the window used in the STFT. Once a Mother Wavelet is chosen it is translated across the length of entire signal giving a specific resolution.

Next the Mother Wavelet is dilated and once again shifted across the entire signal. This process continues and as the wavelet shifts and changes size-different frequencies are detected within the window created by the wavelet. This process is what gives the Wavelet Transform its MRA [11].

While DWT is a powerful tool it does have a major drawback. DWT is meant to be used on regularly sampled data. For example DWT would work on a patients temperature that is taken every five minutes. Conversely, DWT has issues when the data is being applied to has gaps or missing intervals [13]. To overcome the problem of gaps, interpolation can be used, but unfortunately the data used in this thesis has large missing intervals and it is also completely irregularly sampled thereby reducing the efficacy of these solutions or completely invalidating them.

3.4 Summary of Previous Work

- When dealing with time series data the level of resolution that can be accomplished is dependent on the amount of missing data, which in this instance is quite high.
- The data is high dimensional, with each of these dimensions having a different scale, it is necessary to normalize each variable in some manner.
- The data is irregularly sampled which makes using a transform such as DWT nearly impossible.
- The classification model chosen is very important. Several teams that chose to use a neural network consistently underperformed while other models such as SVM performed well.

4 METHODS USED

The methods used in this thesis are discussed below. With 42 potential different variable types per patient it is necessary to comprise methods to use the variables in a model without variable scale altering the applied weights for that variable. This is where the technique of normalization comes into play. The curse of dimensionality is an issue, therefore some feature reduction is important to utilize. A classification model must be chosen which can accurately predict the morality of patients such as K-Nearest Neighbor or Support Vector Machine.

4.1 Normalization

With 42 different types of patient variables being recorded and all 42 variables having different scales it is important to compare the variables without the scale of the variables interfering with the comparison. For example both a patient's cholesterol and temperature are recorded but since the variables have different scales if a similarity measure such as Euclidean distance is used to compare variables then the variable with the greater scale will have a larger weight. Normalization is used to rescale the variables so that all the variables have a common scale.

Previous work illustrates the use of Gaussian normalization which scaled all of the variables between negative one and one and transformed all the variables so that their distributions would match that of the standard Gaussian distribution [7]. To accomplish this, the outliers of each variable were clipped between the first (1st) and ninety-ninth (99th) percentiles. This process involves sorting each occurrence of a specific variable from least to greatest. Then, the empirical quantile of each value is found with equation 4-1.

$$q_i = \frac{i - \frac{1}{2}}{N} \; for \; 1 \leq i \leq N \qquad \textbf{(Equation 4-1)}$$

Where q_i is the empirical quantile value, i is the index of the sorted values, and N is the total number of occurrences of a variable. The index of the 1st percentile can be found by determining the smallest empirical quantile value that is greater than .001; which would therefore correlate the first percentile (i_L). The index of the 99th percentile (i_U), corresponds to the largest empirical quantile that is smaller than .99 as illustrated in equation 4-2.

$$\begin{gathered} i_L = \min\{i \mid q_i > .01\} \\ i_U = \max\{i \mid q_i < .99\} \end{gathered} \quad \text{(Equation 4-2)}$$

The variable value of the 1st percentile, $X_L = X_{i_L}$, and the variable value of the 99th percentile $X_U = X_{i_U}$ are found and saved to be used in the next step. A Nx3 matrix of reggressors R and a Nx1 vector Y are found as illustrated in equation 4-3.

$$R = [1, X_i, \log(1 + X_i)] \quad and \quad Y = \frac{\beta^{-1}(q_i)}{3} \quad for\, i_L < i < i_U \qquad \text{(Equation 4-3)}$$

Where $\beta(.)$is the Cumulative Distribution Function (CDF) of the standard Gaussian Distribution.

The CDF gives the accumulated probability from negative infinite up to a point X of a distribution [14]. This means that CDF represents the probability of a random point falling between negative infinite and X of a given distribution. In the case of the standard Gaussian distribution this can be represented mathematically as illustrated in equation 4-4.

$$\beta(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}}^{(\frac{X^2}{2})} dx \qquad \text{(Equation 4-4)}$$

Where for the purposes of this thesis $z = q_i$.

For each variable class a vector W (W is a 1x3 vector) of weighting coefficients was found as illustrated in equation 4-5 [7].

$$W = (R^T R)^{-1} (R^T Y) \qquad \text{(Equation 4-5)}$$

To normalize each variable occurrence x the corresponding values of X_L, X_U and vector W are retrieved and x is clipped between X_L, X_U. The clipped version of x is then passed into the final equation where it is transformed using the weighted values of W and its normalized value, z, is returned as illustrated in equation 4-6.

$$z = W_1 + W_2 + W_3 \, Log(1 + x) \qquad \textbf{(Equation 4-6)}$$

Once the normalization process is complete all the normalized values now fall between a range of -1 to +1. With all the variables on the same scale they can now be compared.

After implementing the Gaussian normalization process and using it with the classification model chosen for this thesis the classification results were poor. One possible explanation is a limiting factor of the programming language. The language chosen was python version 2.7. When taking the integral of $\beta(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}}^{(\frac{X^2}{2})} \, dx$ Python could not handle the complexity of taking the integral from negative infinite to z. Since the range being considered was from the 1st percentile to the 99th percentile the decision was made to change the lower bound on the integral from negative infinite to the value X_L. While this solved the problem of python not being able to handle negative invitee it is a deviation from the original equation and could be a possible reason that the transformation performed poorly.

Another possible reason that Gaussian transformation did not perform as well as anticipated is when solving for W it is often impossible to take the inverse of $(R^T \, R)$. This is because not all matrices have an inverse [15]. For a matrix to have an inverse it must meet the following requirement as illustrated in equation 4-7:

The matrix A is invertible if there exists a matrix A^{-1} such that

$$A^{-1}A = I \;\; and \; AA^{-1} = I \qquad \textbf{(Equation 4-7)}$$

Where I is the identity matrix. A matrix inverse is similar to an inverse of a number. For instance if the number 2 is multiplied by the number 8 the result is 16. If that result is multiplied by the inverse of 8, 1/8, then it returns the original number 2. The same is true for a matrix inverse. If the identity matrix is transformed by matrix A then transforming that result by A^{-1} should return the original identity matrix [15].

Unfortunately the paper on which this Gaussian transformation procedure was based on was vague and did not cover what to do in these instances. Thus the decision was made that when taking the inverse $(R^T R)$ proved impossible the pseudo inverse would be taken. The pseudo inverse of a matrix is an approximation of A^{-1} [15]. When multiplying A by its pseudo inverse the identity matrix is not returned but instead a best approximation matrix to the identity matrix is returned. The use of the pseudo inverse was a logical choice however it did not allow the Gaussian transformation to perform as intended.

While the Gaussian transformation implementation was unsuccessful, the outlier clipping was useful. Instead of using a Gaussian transformation the more commonly used Z-score transformation (also known as "zero-mean normalization) was used in conjunction with the variable clipping from the 1st to 99th percentiles. Z-score transformation works by transforming a number into a value that represents how many standard deviations the original value is away from the mean of that variables population [1]. The transformed data has a mean value of zero and a standard deviation of one. Thus if a transformed value has a result of zero, this means that the original value was equal to the mean, and if instead the value was two then the value was two standard deviations above the mean. The equation for the z-score is illustrated in equation 4-8:

$$Z_i = \frac{(X_i - \mu)}{\sigma} \qquad \textbf{(Equation 4-8)}$$

Where X_i is the value being normalized, μ is the mean of the values population, and σ is the standard deviation. The z-score normalization worked well at normalization for the variables discussed in this thesis.

4.2 Feature Reduction - PCA & SVD

Dimensionality is a problem that needs to be addressed. Originally there were approximately 40 variables which represented patient variables. To help reduce the dimensionality of the data both Principle Component Analysis (PCA) and Singular Value Decomposition (SVD) were considered [16]. Both techniques accomplish the same goal of taking high dimensional data and reducing it to a lower dimensional space. Both techniques transform the data into new dimensions which better define the data based on eigenvalues and eigenvectors. The two techniques are so similar that the names are often used interchangeably and some refer to SVD as a form of PCA [17]. While both traditional PCA and SVD accomplish the same goals, the math behind them is a bit different. In order to discuss PCA and SVD it is important to understand the fundamental mathematical concepts applied in PCA and SVD.

An important concept of PCA and SVD are eigenvectors and eigenvalues. An eigenvector is simply a direction and an eigenvalue is a value which describes the amount of variance in the data in that direction [16]. For every eigenvector there is a corresponding eigenvalue. The number of eigenvectors that a set of data can have is equal to the number of dimensions the data is in. Eigenvectors are used to project the original data into a new set of orthogonal dimensions [16]. The mathematical definition of an eigenvector is illustrated in equation 4-9.

$Let\ A\ be\ a\ square\ matrix\ NxN.\ \lambda\ is\ an\ eigenvalue\ of\ A\ if\ there\ exists\ a\ nonzero\ vector\ \vec{v}\ such\ that$:

$$A\vec{v} = \lambda\vec{v} \qquad \textbf{(Equation 4-9)}$$

This equation states that A is a square matrix, $\vec{v}$ is the eigenvector, and λ is a scalar which is the corresponding eigenvalue to the eigenvector. To find the eigenvalues and eigenvectors of a square matrix, treat the matrix as a system of linear equations and solve for the variables within the linear equation [18]. Using this technique an NxN matrix will result in N eigenvectors being found.

PCA works by finding the covariance matrix between all the preexisting dimensions of the dataset and then calculating the eigenvectors from this matrix and ranking them by eigenvalues [19]. SVD breaks a matrix down into the product of matrices U, S, and V where U is an orthogonal matrix, S is a diagonal matrix, and V is the transpose of an orthogonal matrix [18]. The equation for SVD is illustrated in equation 4-10:

$$A_{mn} = U_{mm}S_{mn}V_{nn}^{T} \qquad \textbf{(Equation 4-10)}$$

"Where $U^TU = I, V^TV = I$; the columns of U are orthonormal eigenvectors of AA^T, the columns of V are orthonormal eigenvectors of A^TA, and S is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order" [18].

PCA and SVD both use the same eigenvectors and eigenvalues to define new dimensions and rank these dimensions by importance. The main difference between these two techniques is SVD allows for the use of sparse matrices since it can operate directly on the data; In contrast PCA requires a covariance matrix to be created first [20]. Thus, SVD was chosen for this thesis since much of the data is extremely sparse.

Figures 4-1 to 4-5 demonstrate a step by step example of a SVD transform of data points from their original dimensions into a new eigenvector space.

Step 1: The data is normalized so all dimensions use the same scale. The data is plotted along its original dimensions. Figure 4-1 illustrates this sample data set initially has two dimension denoted by and y.

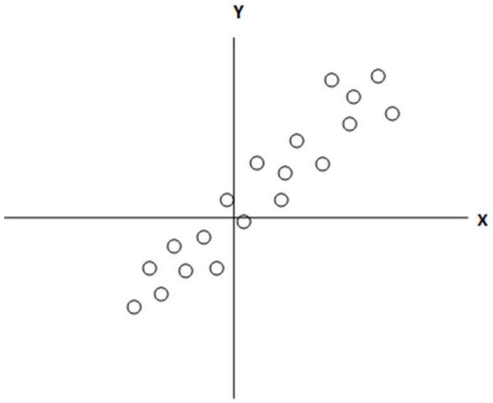

Figure 4-1: SVD Example Part 1

Step 2: Find the first eigenvector. This eigenvector has the highest eigenvalue indicating that dimension has the most amount of variance. As illustrated in Figure 4-2 the solid line represents the eigenvector/eigenvalue, where the eigenvector determines the direction of the line and the eigenvalue determines its scale.

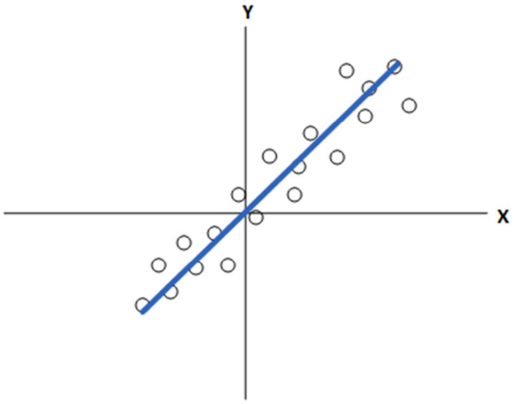

Figure 4-2: SVD Example Part 2

Step 3: Figure 4-3 illustrates the second eigenvector after it is found. This second eigenvector, presented by the dotted line, is orthogonal to the first one and has a smaller amount of variation.

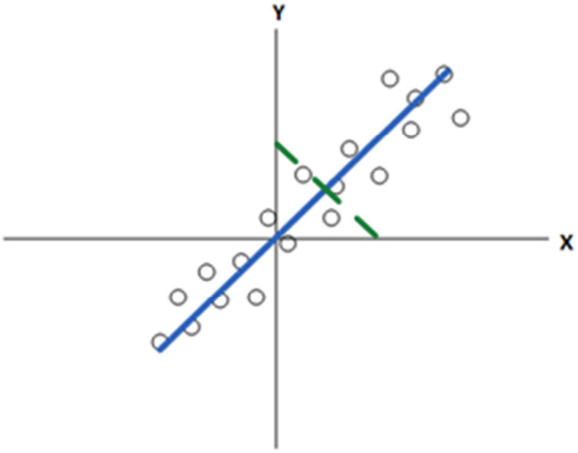

Figure 4-3: SVD Example Part 3

Step 4: Using all the eigenvectors/eigenvalues the data is projected onto the new eigenvector space. Figure 4-4 illustrates the example data set after its projection onto the eigenvector space.

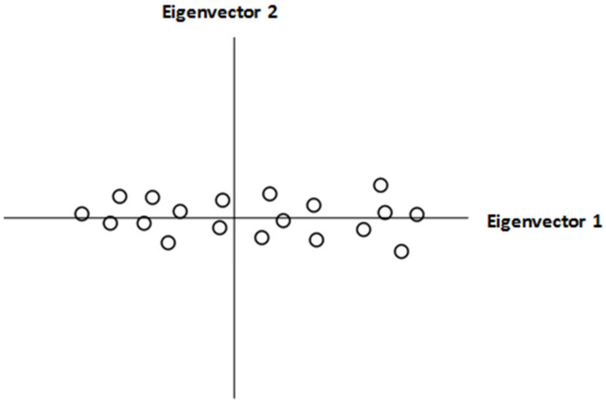

Figure 4-4: SVD Example Part 4

Step 5: The data is in the new dimensional space and the dimensions are ranked by importance with regards to variation. To reduce the dimensionality of original data the less variation dimensions can be removed. From the example, most of the variation comes from eigenvector1 thus eigenvector2 could be removed and the data points are projected into one dimensional points along the eigenvector1 dimension as illustrated in Figure 4-5.

Eigenvector 1

Figure 4-5: SVD Example Part 5

Using SVD the dimensional complexity of a data set can be greatly reduced while at the same time maintaining most of the original structure [18]. Discussed in Section 5, by using SVD the number of features is dramatically reduced with only a small loss in accuracy. SVD was a useful technique for feature reduction that can greatly helped alleviate the problems of dimensionality.

4.3 Classification - k-Nearest-Neighbor

After completing the preprocessing steps of data selection and transformation, the next step in the knowledge mining process is to choose an appropriate data mining model. The first classification method used in this thesis was the k-Nearest-Neighbor (KNN). KNN is a Lazy Learner which means that when the algorithm receives training tuples it simple stores those tuples and doesn't apply calculations until after a test tuple is provided [1]. A Lazy Learner completes all the calculations on the fly when presented with a test tuple that needs to be classified. KNN works by searching for the *k* closest training tuples to a given test tuple. These closest tuples are called the "nearest neighbors" [1]. After the *k* nearest neighbors are found a vote is taken to determine into which category the test tuple is classified.

To determine the "closeness" of tuple the similarity measure Euclidean distance was used. Euclidean distance finds the distance between two points X_1 and $X_2$of *n* dimensions by using the formula as illustrated in equation 4-11.

$$\boldsymbol{dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}} \qquad \text{(Equation 4-11)}$$

KNN didn’t provide the accuracy desired and because lazy learners are typically slow and computationally expensive it was decided that an “Eager Learner” solution would be used. An Eager Learner takes training tuples and then builds a model that can be used to classify new testing tuples. This approach allows for a single model to be built and then used to classify any number of testing tuples. Since there are thousands of training and testing tuples, an Eager Learner allowed for a single prebuilt model resulting in calculations that were not computationally expensive.

4.4 Classification - SVM

The challenge was a binary classification problem in which patients must be classified into either “lies” or “dies”. To accomplish this a supervised learning algorithm is appropriate since the classes are known for the training tuples. Since a lazy learner algorithm has been ruled not acceptable the classification model for this thesis will be an eager learner. Additionally, the solution chosen must be well suited to multi-dimensional data. Therefore, the model chosen is “Support Vector Machine” (SVM).

The SVM algorithm was designed by Boser, Guyon, and Vapnik in 1992. Some of SVM’s strengths include the ability to handle high-dimensional and diverse sources of data with high classification accuracy [21]. SVM involves the “optimization of convex function” meaning that it has no false minima like a neural network [21] It is able to handle multidimensional data better than other models [1]. SVM is a model that is easily understandable as compared to other models such as neural networks [21].

In order to understand SVM imagine a data set in which all the inputs, X, have class value $Y = -1$ or $+1$ as show in the Figure 4-6. Furthermore, imagine that this data set is linearly separable; meaning that the two classes allow a straight line drawn between them for two dimensional data, or hyperplane for multidimensional data [22]. For multidimensional data there are several hyperplanes that are potential solutions. SVM finds the “Maximum Marginal Hyperplane” (MMH) [1], which is the hyperplane with the maximum distance from itself to the nearest points of each class. These nearest points are called “Support Vectors” (SV) and are used to define the boundaries of the MMH with the hyperplane being in the center.

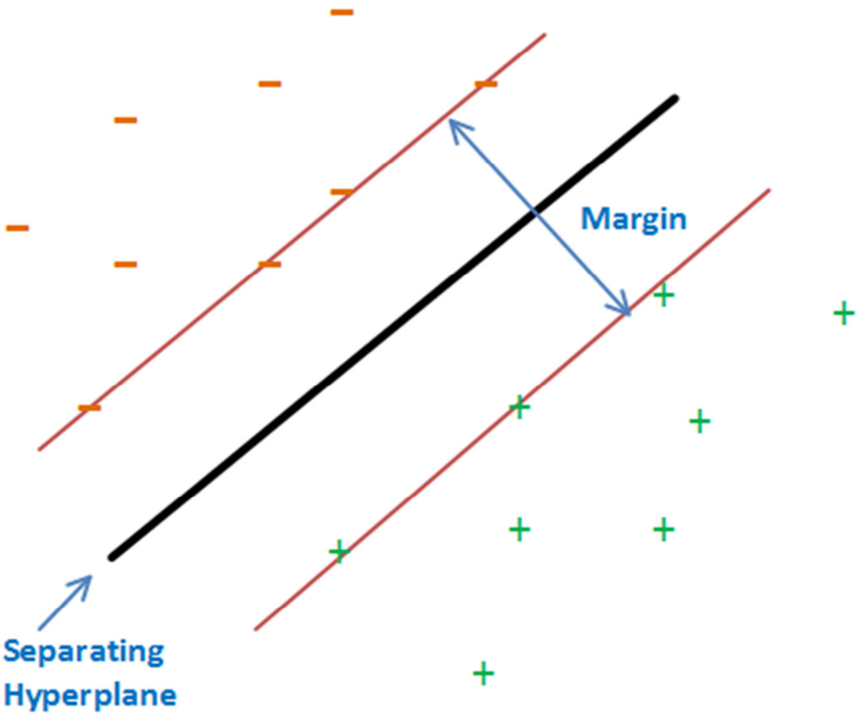

Figure 4-6: SVM Hyperplane Example

Support vectors are unique and in the SVM model and are the only points that matter because any non-support vector points can be removed without changing the MMH. This SVM feature is very useful because the complexity of the model is not based on the number of inputs but based on the number of support vectors.

Up to this point it has been assumed that the data is linearly separable, but with real world data sets this is often not the case. To classify non-linearly separable data a SVM model can take two approaches, soft margins or shifting the data to a higher dimension. Soft margins allow points to be on the wrong side of the margin. For each point a penalty is assigned. This penalty is calculated by the number of points that violate the margin and the distance those points are from the margin. For soft-margin constant, C, on the decision boundary, a small value of C means the SVM will have a larger margin and will allow more points to be ignored that are near the decision boundary [21]. Since C controls the balance between the margin size and penalty received, this is a tuning parameter that can be used to maximize performance and handle outliers.

Shifting data into a higher dimension in which a decision boundary can be found is an alternative means to handling non-linearly separable data. Unfortunately, explicitly computing these non-linear features can prove to be quite computationally expensive. When shifting into a higher dimension there is a quadratic increase in both the time and memory required to complete the computations [21]. Based on the quadratic increase in time and the memory required explicitly shifting the data is not feasible for this thesis.

To handle non-linearly separable data, a Kernel function was used. A Kernel function acts as a scalar product on the higher dimension which implicitly trains a SVM on the space defined by the kernel [23]. The Kernel function maps the data points into an alternative feature space through a replacement: $X_i * X_j \rightarrow \phi(X_i) * \phi(X_j)$ [21]. This replacement is the inner product between pairs of points in the higher dimensional feature space [21]. Once in the new higher dimensional space it is possible to find a decision boundary between two classes which was impossible at the lower dimension.

For example presume there is a one dimensional data set that contains data points ranging from -10 to 10 and all the points are sorted in ascending order. Furthermore let the squares represent one class of data while the triangles represent another class. As illustrated in figure 4-7 the data points of the two classes are impossible to separate.

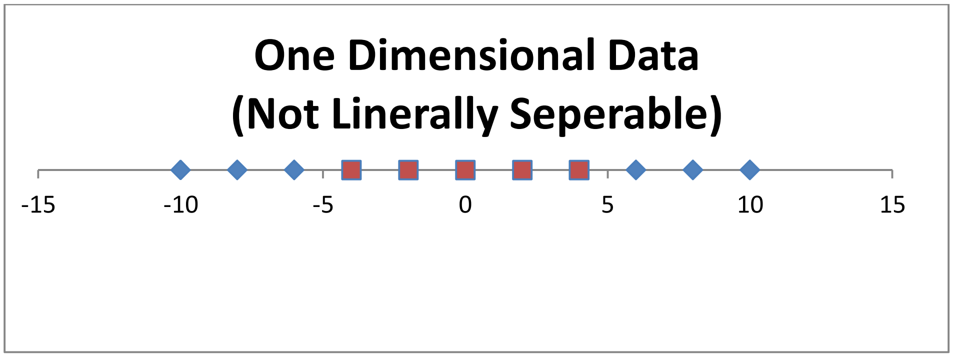

Figure 4-7: One Dimensional Data

By using a Kernel function and mapping the data into a higher dimensional space the data is separable. As illustrated in figure 4-8 the X dimension remains unchanged; however the Y dimension is transformed by squaring each X value.

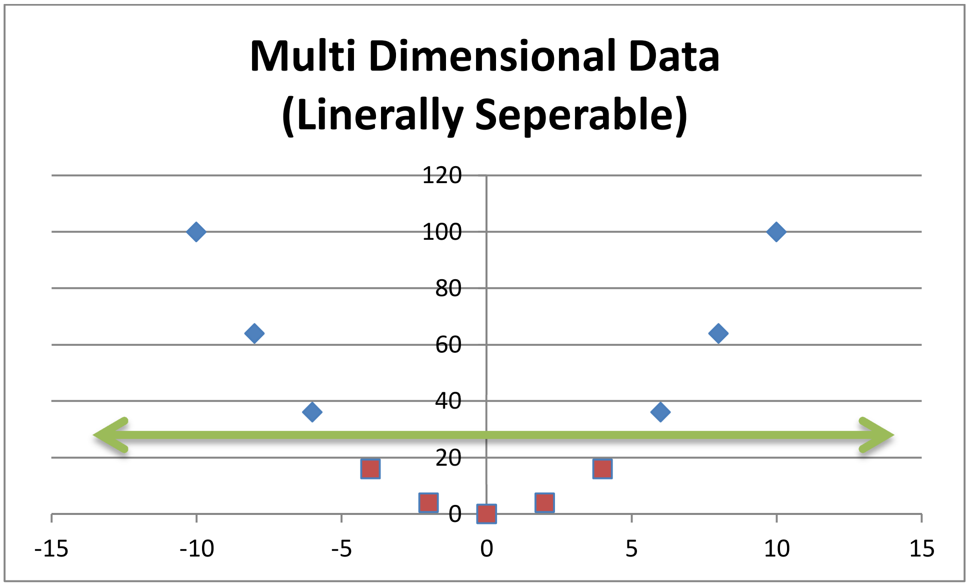

Figure 4-8: Multi Dimensional Data

The kernel function for each SVM model depends on the type and distribution of the data. A kernel is chosen through a process of trial and error in which cross-validation is used to discover which kernel elicits the best results. Popular kernels include linear, polynomial, radial basis function (RBF), and sigmoid [23]. A kernel acts as a similarity measure and the points belonging to the same class will have a higher kernel level and points belonging to different classes will have a lower kernel level [23].

Once a kernel has been chosen the SVM model is constructed. The accuracy of the model depends upon many factors including the chosen Kernel and the values for the tuning parameters. The dimension in which the SVM is trained and the size of the margin determine the key characteristics of the SVM. In summary, the SVM is a versatile classification model that is well suited for multi-dimensional data and binary classification problems. The SVM model was chosen for this thesis.

5 RESULTS

"Cross-validation & scoring" is used to ensure that the results collected are reliable and not biased. Specific changes to the overall model are examined to see how the outcome is affected. The different kernel choices are analyzed to compare the impact on performance of the support vector machine. Different time slicing techniques are investigated based on the time series aspect of the data. Single Value Decomposition is studied for the impact it has on reducing the number of features.

5.1 Cross Validation & Scoring

For results to be meaningful, it is important to guarantee the results collected are accurate unbiased. *K*-fold cross validation was used in this thesis when doing any testing. *K*-fold cross validation breaks the data set up into *K* parts (folds). One fold is designated to be the test set and the other *K* - 1 folds are designated to be the training sets [1]. The experiments are run and the results are collected. Once again, the process is repeated with a new fold chosen to be the test set and the remaining folds used for training. This process continues K times and then the results are averaged to give the final score. Using K-fold cross validation this helps to prevent overfitting and bias [1]. For this thesis a 10-fold cross validation was chosen. Meaning 10 percent of the data at any one time is used for testing and the other 90 percent is used for training. A 10-fold cross validation technique is very common in data mining and it has proven to be reliable and return more accurate results [1].

The overall goal for this challenge is to maximize the Score1 result which is the minimum of the sensitivity (Se) and precision (+P). Se represents the fraction of in-hospital deaths that are predicted and precision +P represents the fraction of correct predictions of in-hospital deaths. Se, +P, and Score 1 are determined with the following equations where TN = True Negatives, TP =

True Positives, FN = False Negatives, and FP = False Positives as illustrated in equation 5-1, 5-2, and 5-3.

$$Se = \frac{TP}{TP + FN} \qquad \textbf{(Equation 5-1)}$$

$$+P = \frac{TP}{TP + FP} \qquad \textbf{(Equation 5-2)}$$

$$Score\ 1 = \min(Se, +P) \qquad \textbf{(Equation 5-3)}$$

5.2 Choosing a Kernel

Before handling the time series aspect of the data the Support Vector Machine needs to be tuned to work well with the data set. The most important choice to make when using a SVM is what type of Kernel to use. The Kernel acts like a similarity measure for SVM [23] and the kernel choice is derived from the type and distribution of data. Using the “Kernel Trick” it is possible to replace the dot product between two vectors with the kernel function. This substitution allows values can to mapped to higher dimensions without having to compute the mapping explicitly [21].

The first kernel tested was the Linear Kernel. The Linear Kernel is different from the other Kernels because it doesn’t actually change the dimension. The Linear Kernel is only useful in cases where the data is linearly separable within the current dimension. While the Linear Kernel is of limited use its performance can be improved by using the SVM tuning parameter "c". c sets how fuzzy the decision surface is between classes [24]. Equation 5-4 is the Linear Kernel where *x* and *y* are the two vectors under consideration.

$$k(x, y) = x^T y \qquad \textbf{(Equation 5-4)}$$

The data in this thesis is of high dimensionality therefor the the Linear Kernel performed poorly and returned a constant 0.0 Score1 result no matter what c value was chosen for the decision margin.

The next kernel tested was the Hyperbolic Tangent (Sigmoid Kernel). The Sigmoid Kernel is also used for neural networks and when using the Sigmoid Kernel with SVM it is similar to using a "two-layer, perceptron neural network" [25]. The equation for the Sigmoid Kernel is illustrated in equation 5-5 where γ determines how much influence an individual training example has on the determination of which class it and its surrounding data points are classified as.

$$k(x, y) = \tanh(\gamma x^T y) \qquad \textbf{(Equation 5-5)}$$

After testing the kernel and adjusting both γ and c it was found that the best γ value is 0.1 and the best value for c is 1000 and returns a Score1 value of 0.226. These results are illustrated in Figure 5-1 and 5-2.

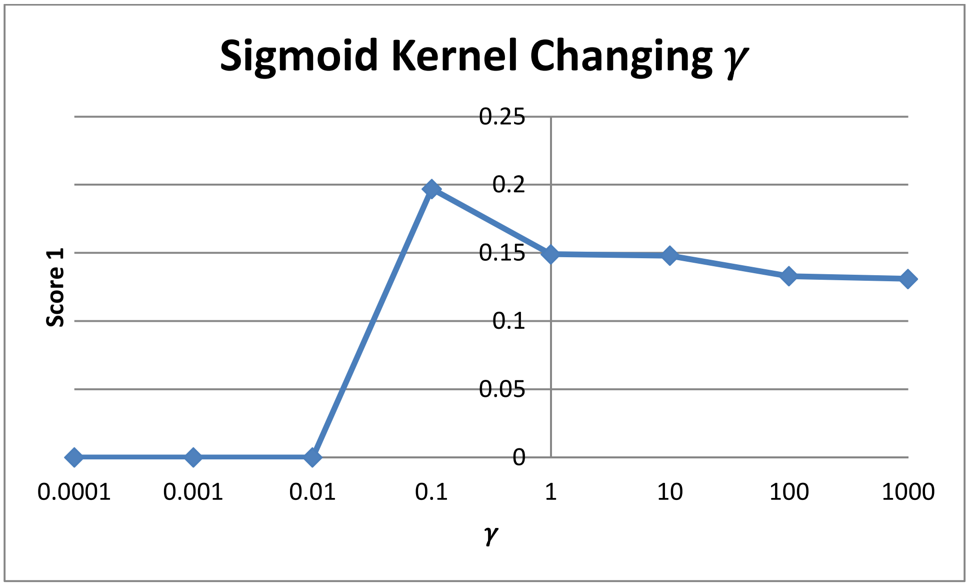

Figure 5-1: Sigmoid Kernel Changing γ

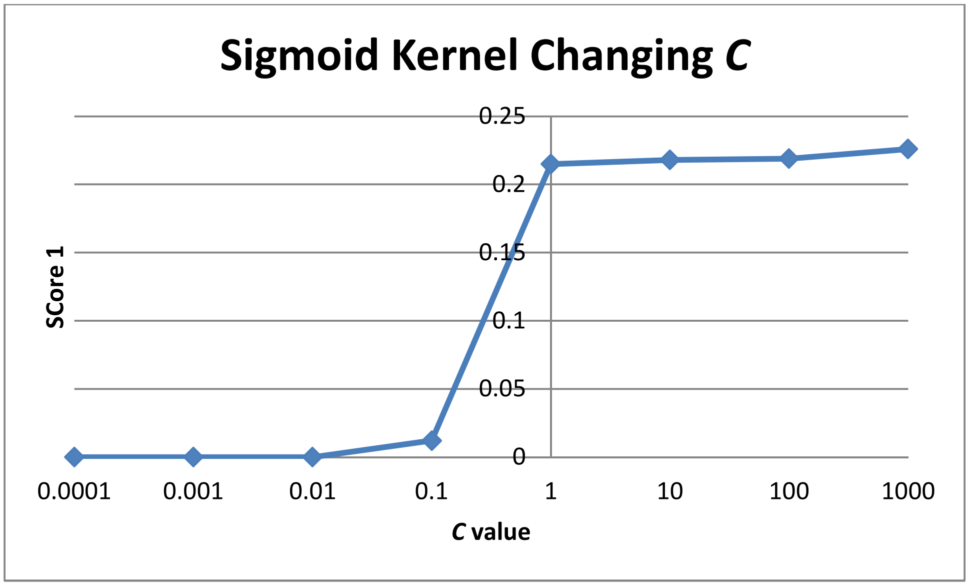

Figure 5-2: Sigmoid Kernel Changing C

The Sigmoid Kernel performed as poorly as the K-Nearest Neighbor.

The polynomial kernel was chosen next. The polynomial kernel can be raised to any degree d that is higher than the number of dimensions of the data as illustrated in equation 5-6 [21]. The SVM tuning parameter c value can also be manipulated to increase performance. The polynomial kernel returned very poor Score1 results. Using a dimension value d of 2 to 5 degrees higher than the number of data dimensions performed better, however the results were still poor.

$$k(x, y) = (x^T y)^d \qquad \textbf{(Equation 5-6)}$$

The best results were from the Gaussian Radial Basis Function (RBF). RBF is a versatile kernel function that adapts well to many different types of data sets. Its flexibility stems from the use of an infinite-dimensional feature space which allows the data to be separable [23]. RBF behaves similarly to a density based clustering algorithm allowing it to find multiple groupings of unique shapes. The equation for RBF is illustrated in equation 5-7.

$$K(x, y) = \exp(-\gamma||x - y||^2) \qquad \textbf{(Equation 5-7)}$$

After adjusting the values of γ and c within the SVM and RBF kernel it was found that a γ value of 0.01 and c value of 1000 worked best and returned an average score 1 value of .324 as illustrated in Figures 5-3 and 5-4.

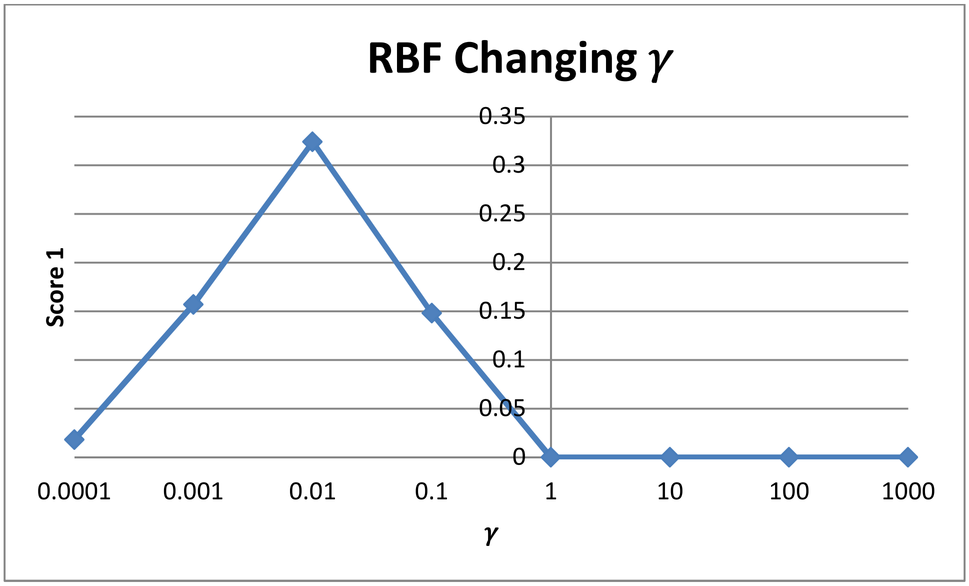

Figure 5-3: RBF Changing γ

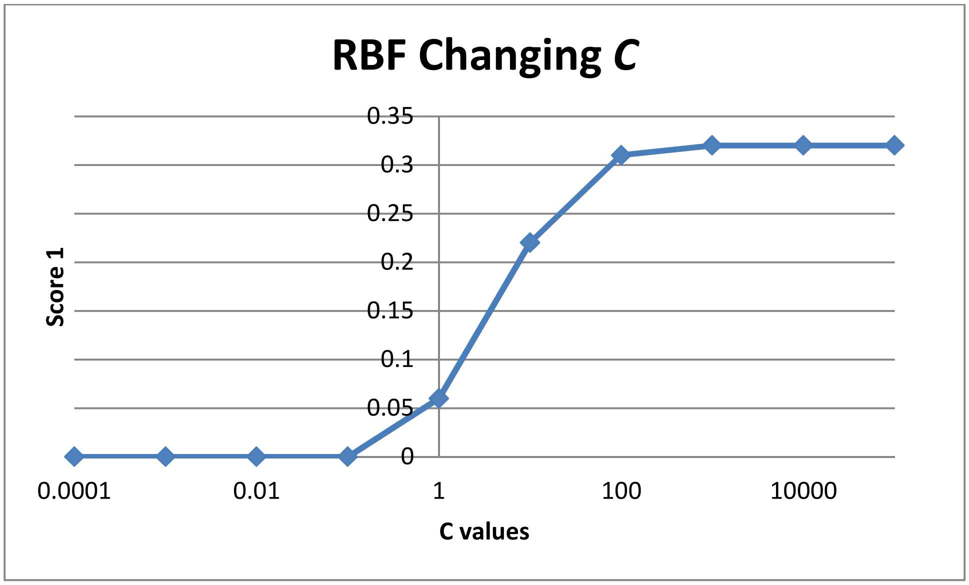

Figure 5-4: RBF Changing C

After comparing the results of using different kernels it was determined the transformed data feature space affected the SVM performance. Figure 5-5 illustrates a comparison of kernel performance as it relates to the SVM. The Liner Kernel performed poorest and the RBF kernel performed best.

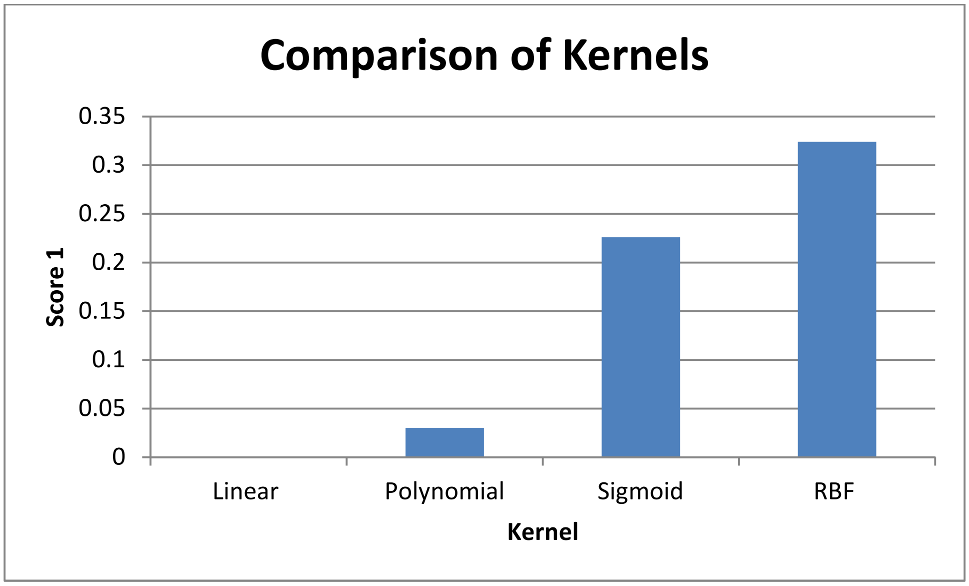

Figure 5-5: Comparison of Kernels

5.3 Time Segments

Now that a kernel has been chosen the time aspect of the data must be considered. One approach is to break the time into segments of equal length. After breaking the data into equal time segments the z-score normalization was applied. To reduce the dimensionality SVD was used and classification predictions were made for each time segment using SVM. This approach used a majority voting system amongst the time segments to determine the final overall patient prediction. In the case of a tie the outcome would be randomly chosen with 7 to 1 odds of living vs dying respectively since this was close to actual distribution outcome of the data.

After applying the time segment majority vote to the time segments it is clear that as the number of time segments were increased the overall accuracy of the model decreased as figure 5-6 illustrates. The best Score1 value for the different time segments was from 1 time segment.

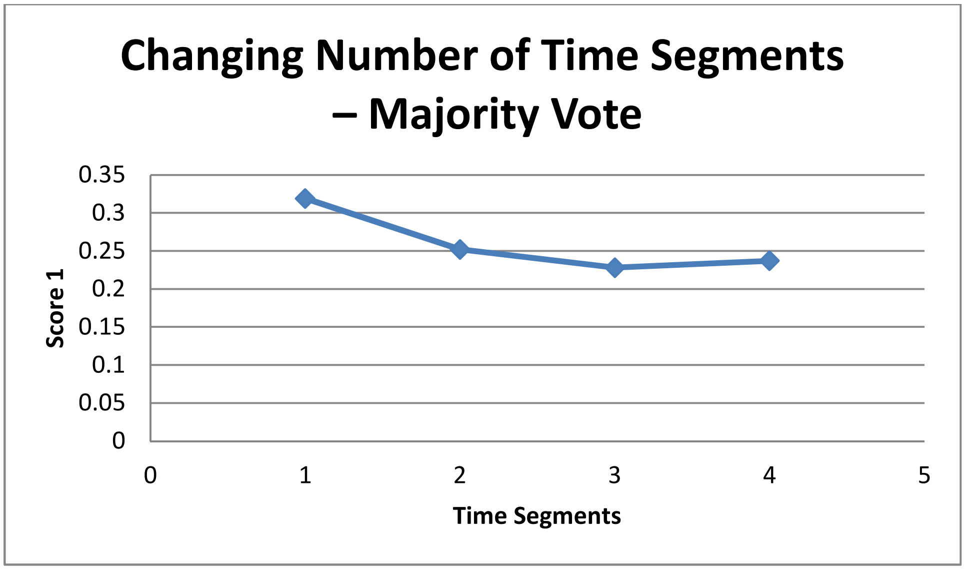

Figure 5-6: Changing Number of Time Segments –Majority Vote

The next attempt to improve performance involved changing the voting between segments. Instead of giving equal weight to all the segments each segment was assigned a weight according to its accuracy. Every time segment was individually tested to determine its particular overall individual accuracy at classifying patient outcomes. This determined accuracy was recorded and used for weighting the time segments. With the new weighted data for the time segments the tests were re-run with the new voting scheme. The results illustrated that as the data was segmented into smaller time chunks the overall accuracy of the mod declined.

In the previous attempts, each time segment was treated completely independently. Every time segment had its own z-score normalization, and PCA was applied before making a prediction based on its own SVM model. This approach meant that depending on the distribution of data within a time segment the components transformed by SVD for one time segment could be different from those within another time segment. The SVM model was built upon only the limited information within each time segment and thus as the number of time segments increased and the window width of the time segment decreased the overall accuracy decreased.

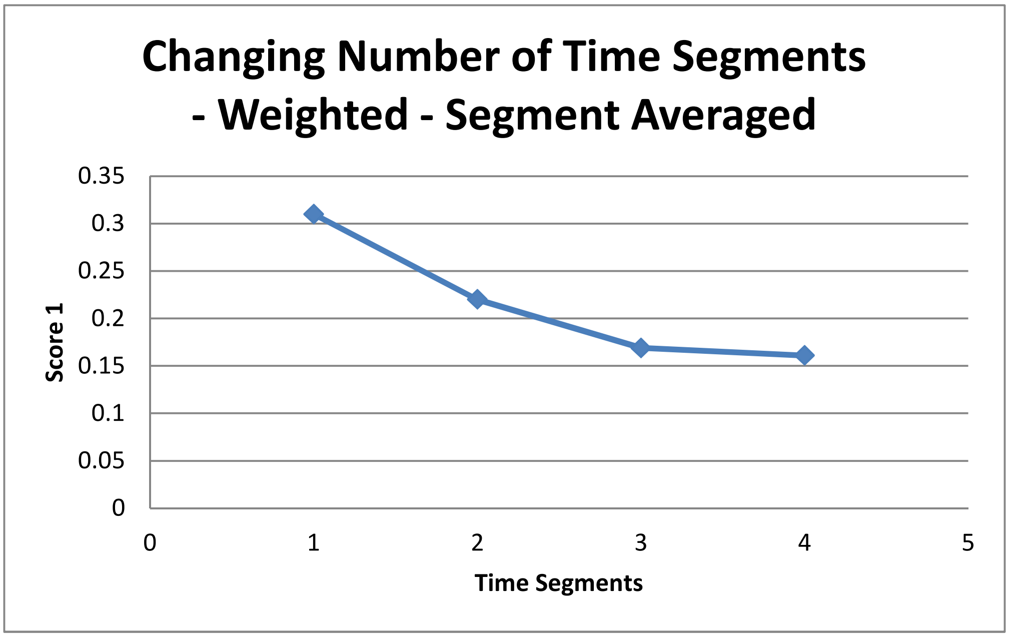

Figure 5-7: Changing Number of Time Segments - Weighted - Segment Averaged

To solve the problem of independent time segments, the data was broken up into individual time segments and normalized using the standard z-score normalization technique; however the z-score normalization the average value used in the formula was based on the overall average from all the time segments for each variable. This allowed the components for SVD to be based on the data holistically instead of single time segments. These changes allowed all of the components from the different time segments to be the same transformed spaces after applying SVD. Stated concisely the SVD would determine the most important features of the data, as a whole, and then apply these features to each time segment. The SVM now uses the most important features over the entire length of time. For example, if the number of components from PCA was 15 and the number of time segments was 3, then the SVM would receive 45 data points for each patient where values 1, 16, and 31 would all represent the same variable but at different points in time.

The approach of recombining the data before applying the SVM did show an improvement however it also showed a decline in accuracy with an increase in time segments as illustrated in Figure 5-8.

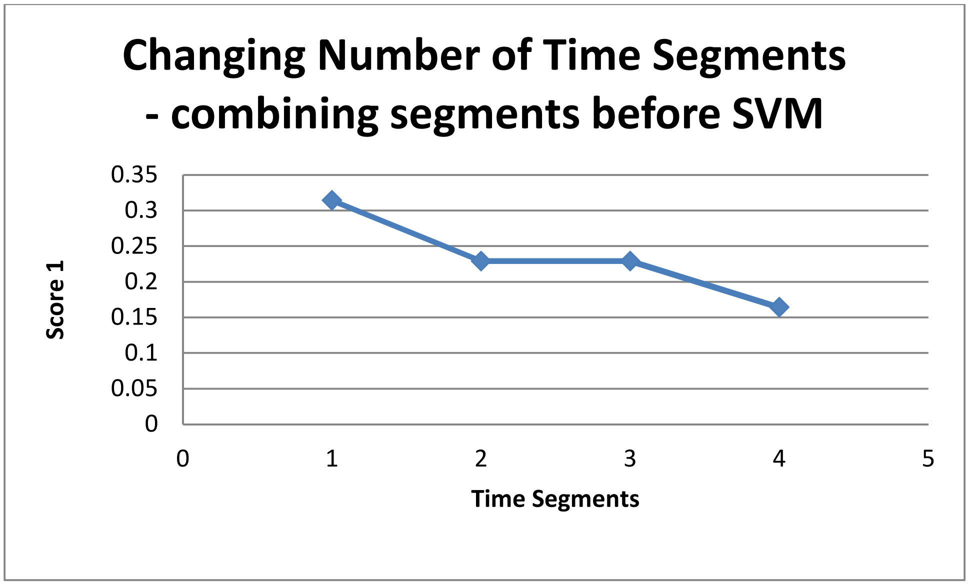

Figure 5-8: Changing Number of Time Segments - combining segments before SVM

All the techniques showed a decrease in Score1 value as the number of windows increased as illustrated in Figure 5-9.

Among the three methods used for time segmentation, a simple majority vote produced the best results. While this technique performed better than the others it wasn't able to increase the overall performance of treating the data as a whole. SVM provided the best results when it was applied on the training data set in its entirety. When the training data is segmented into time chunks the hyperplane the SVM finds is not as accurate and thus when applying the test set the overall Score 1 declines as well.

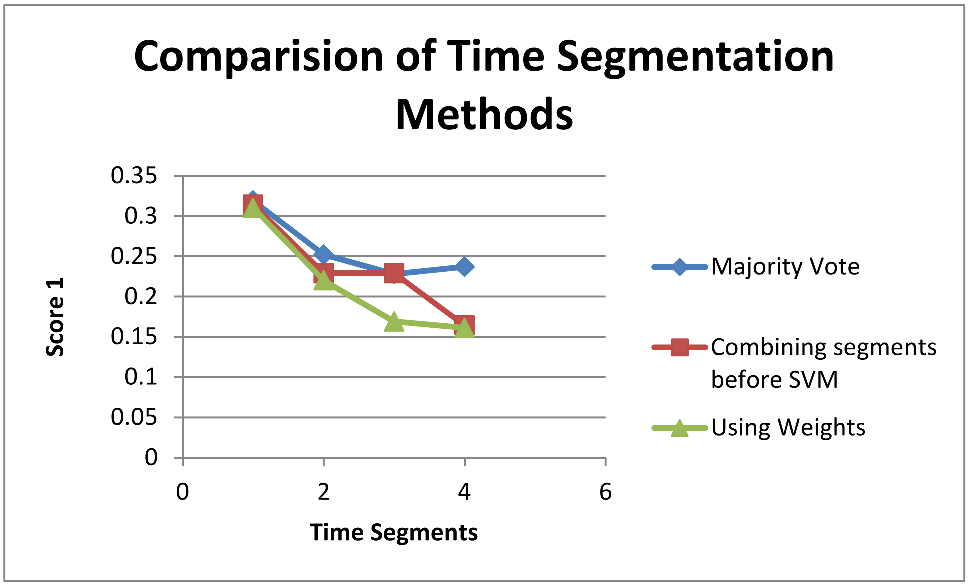

Figure 5-9: Comparison of Time Segmentation Methods

5.4 Single Value Decomposition

Recall there are 42 variables that describe a patient. After removing the patient ID and MechVent there were 40 variables remaining. With 40 different variables and thousands of patients dimensionality is a problem. To reduce the number of variables the SVM must handle, Singular Value Decomposition (SVD) was applied.

SVD is able to define new dimensions to describe the data and then rank those dimensions by variation [18]. Using this technique, high dimensional data can be described using fewer dimensions while still maintaining a high level of accuracy. SVD was applied to the normalized data set and several tests were conducted and the data set was reduced. As previously stated the Score1 values were considered for SVM with a RBF kernel and SVD with a varying number of components. These results are illustrated in Figure 5-10.

As the number of components increased so did the overall Score1 value. Once the component level reached 15 the increase in performance was minimal no matter how many more components were added. There was only a 3 percent increase in Score1 value from 15 to 35 components. For this thesis a SVD reduction down to 15 components was chosen.

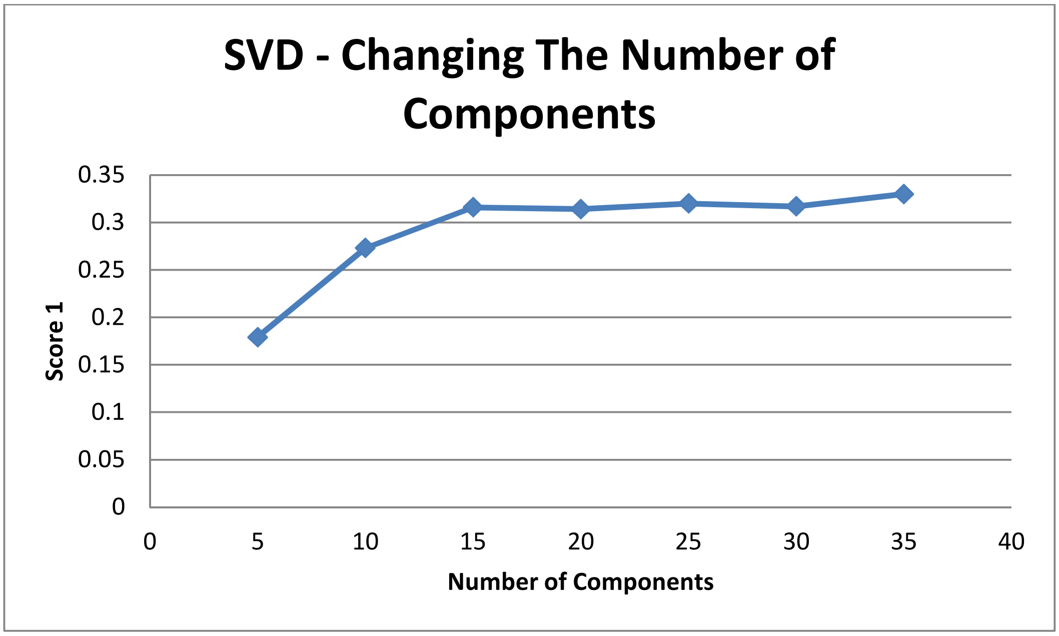

Figure 5-10: SVD - Changing The Number of Components

Figure 5-11 illustrates the “comparative performance” which refers to the percentage difference between using SVM with SVD compared to using SVM without SVD. When viewing the Score1 accuracy with SVD applied to the same model as without SVD it is apparent that the optimal tradeoff between dimensionality reduction and accuracy was 15 components. By using 15 components the overall comparative accuracy is 97%. With only a 3% accuracy sacrifice the dimensionality reduction of 62.5%. The 62.5% value is derived by taking the 25 discarded components divided by the 40 original components.

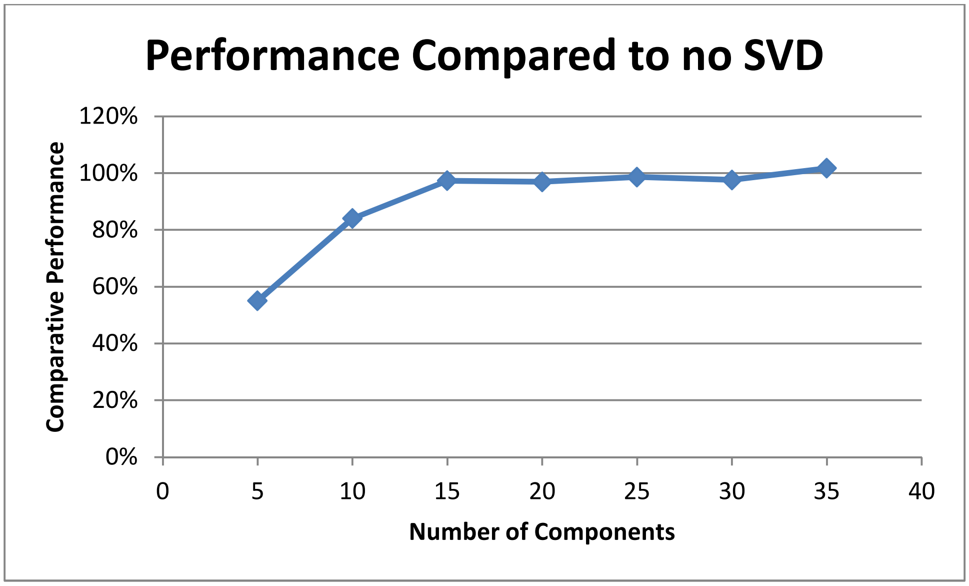

Figure 5-11: Accuracy Compared to no SVD

5.5 Undersampling

After applying SVM and the time segmentation techniques, the Score1 value was still rather low. After reexamining the approach as taken in this thesis, it was determined that the underperformance was caused by the class imbalance between patients who lived vs who died. The SVM was originally trained on the natural distribution of the data which was a little over a 6 to 1 live to die ratio. To overcome this imbalance, undersampling was applied. With undersampling an equal number of tuples from the two different classes of lives and dies are used to form a new data sub set [1]. In the original data set there were 554 patients who died. These 554 deaths were matched with an equal number of living patients providing a new training set of 1108 records. Using this new data set, the classification process was conducted 6 times; each time switching out the living patient tuples for a different set of 554 living patient tuples. The results were averaged together and a Score1 value of 0.6752 was found. This Score1 value more than doubled the previous highest Score1 value of .324. The results of using an undersampled data set are illustrated in Figure 5-12.

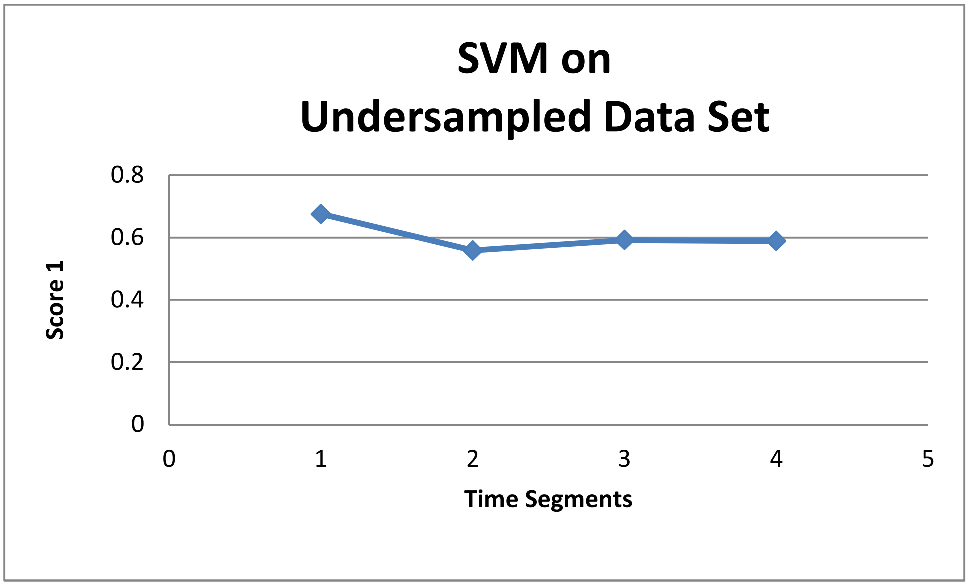

Figure 5-12: Balanced Training Set

As evidenced in Figure 5-13, SVM with an undersampled data set did not have the same continual decline in Score1 value as SVM with the normal data distribution. Instead, after an initial drop the Score1 value levels out at approximately .60. These results indicate that using a balanced data set has a significant improvement on classification performance.

While the time segmentation still showed no improvement over treating the data holistically the work was still considered a success based on the Score1 results of using SVM with SVD and undersampling.

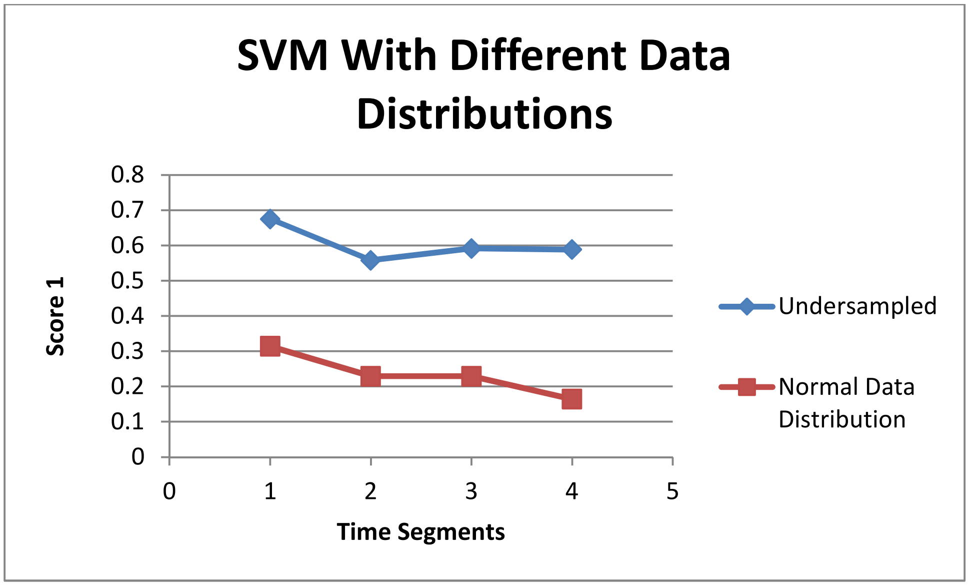

Figure 5-13: SVM With Different Data Distributions

6 CONCLUSION & FUTURE WORK

Today's modern Intensive Care Units (ICUs) are full of sophisticated machinery capable of monitoring and recording a wide variety of patient statistics. Despite having a great deal of data available it is possible to be "data rich but information poor" [1]. To solve the data rich vs information poor dilemma, knowledge mining can be applied. Knowledge Mining uses "raw data" and mines useful information from that data**.** This thesis focuses on a particular type of Knowledge Mining called "Classification". Classification finds a model that can then be used to predict the class of previously unseen data [1]. The data in this thesis is comprised of thousands of patient records from the ICU and the goal is to use a classification model applied to previously unseen patient data to predict mortality.

The abilities to predict the mortality of patients is a challenging task. The magnitude of diversity amongst patients and the ailments that afflict them only compounds the challenges. Within this data set there were 42 variables used to describe patients. There was no guarantee of the number of variables per patient meaning two patients could have completely different variables with no overlap. The only guaranteed variable that the patient had was the "Record ID" which uniquely identified the patient. Furthermore 6 of the variables were "general variables" such as age, gender, etc, with the rest of the variables being "time series" meaning that they were sampled over time adding an additional layer of complexity. The time series variables were not guaranteed to be sampled at regular intervals. If a particular variable was sampled at a certain interval, that same variable for another patient could be sampled at another rate or completely sporadically.

There are four main areas where significant insight was made in regards to the problem presented by this thesis: data reduction and cleaning, selecting an appropriate classification model, selecting a time series technique, and using a balanced data set.

“Data reduction and cleaning” is extremely important to this thesis because given the number of patients and number of variables each patient has; there is a problem posed by dimensionality. When dealing with a data set of this size the data is dirty meaning there are errors which cause outliers that shouldn’t exist and potentially skew the classification process. To overcome these challenges this thesis used a clipping process which bounded each variable to the 1st and 99th percentiles of that variables group. Z-score normalization was used so that variables with different scales could be compared to one another. To reduce the number of overall features Single Value Decomposition a form of Principle Component Analysis was used to transform the data and find the components that represented the greatest deviation in the set. After removing uninformative features and applying SVD the overall number of components was reduced from 42 to 15 while maintaining a reduction of 3 percent accuracy in classification.

Choosing an appropriate classification model is critical in predicting the mortality of patients. Several classifications models were considered and tested. The “k-Nearest-Neighbor” (KNN) model which relies on clustering and similarity measures to equate how similar an unseen tuple is, to previously seen tuples, and base its class on its nearest neighbors was tested. KNN was able to successfully classify patients; however its accuracy was not desired and because KNN is a “lazy learner” not performing calculations until the test tuple is given it indicated that it would not scale well to larger data sets [1].

The Support Vector Machine classification model was also tested. SVM is an eager learner and handles multi-dimensional data well. SVM works by finding decision boundaries between different classes. SVM transforms the data into a new feature space, higher dimension, where the data can be separated. Shifting data to higher dimension can be computationally expensive but SVM uses a "Kernel Trick" to bypass having to explicitly map each tuple and thus solves the complexity [1]. The "Kernel Trick" allows points to be compared in a feature space through each points inner products without having to explicitly map them into that space [21]. The kernel chosen for the SVM determines how SVM performs. By testing different kernels it was determined the Linear Kernel and Polynomial Kernel performed subpar while the Radial Basis Function Kernel (RBF) performed well. Each kernel had the SVM tuning parameters γ and c tested which affect the weight of a tuple and the fuzziness of the margin respectively. After comparing SVM to other models the decision was that SVM was appropriate model for this thesis.

Choosing the correct time series handling technique was critical to this thesis. The Discrete Wavelet Transform (DWT) was considered for use. DWT takes the time/amplitude domain of data and transforms it into a time/frequency domain. DWT gives multi-resolution analysis making it a powerful tool for transforming time series information and a better option than related methods such as the Short Time Fourier Transform [11]. The drawback of DWT was it required the time series data to be regularly sampled. As previously stated the data stated the data in this thesis was not regularly sampled.

The technique for handling time series data of use by "windows" was considered. By segmenting the data into time chunks of a determined size it is possible to see the changes between windows. For this thesis several different windowing techniques were used. Including majority voting, weighted time chunks, and reassembly of data before passing it to the SVM. While some of the tested techniques provided some working data each suffered from degradation as time segmentation was applied.

A balanced data set was considered which dramatically improved the classification performance. Using undersampling to balance the number of patients who lived versus who died, when building the classification model, more than doubled the Score1 value as compared to a unbalanced data set. While time segmentation still suffered degradation, this degradation was not continuous; where an unbalanced data set was continually degrading.

After completing the research the best results were achieved from creating a balanced data subset, removing the outliers by clipping each variable's range between its 1st and 99th percentiles, and using a Standard Z-score normalization. A Single Value Decomposition was applied to reduce the number of variables to 15 and a Support Vector Machine using a Radial Basis Function Kernel with a γ value of 0.01 and c value of 1000. The success of the classification model was based on the highest Score1 value received. Score1 is the minimum of the sensitivity (Se) and precision (+P). Se represents the fraction of in-hospital deaths that are correctly predicted, and +P represents the fraction of correct predictions of in-hospital death among all the predicted death cases. By using this methodology a Score1 value of .675 was achieved.

Although a great deal has been accomplished there is future work to be done. Focus needs to be placed on effectively segmenting the time series data. Support Vector Regression with Trend Analysis to determine how a patient's health fluctuates over time can build off this research [23]. An additional option is an Ensemble Method where multiple classification methods of various types can be used to classify data and then vote on the outcome [1]. In conclusion, the work done here illustrates using SVD and SVM on medical data can be an effective method for classification.

Bibliography

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems)*, 3rd ed. San Francisco, 2011, p. 744.

[2] Y. Jiang, T. Lan, and L. Wu, "A Comparison Study of Missing Value Processing Methods in time series data mining," in *Computational Intelligence and Software Engineering. CiSE 2009. International Conference on*, 2009, no. 50604012, pp. 0–3.

[3] T. Pollard, L. Harra, D. Williams, S. Harris, D. Martinez, and K. Fong, "2012 PhysioNet Challenge: An artificial neural network to predict mortality in ICU patients and application of solar physics analysis methods," *Computing In Cardiology (CinC), 2012*, pp. 485–488, 2012.

[4] "Predicting Mortality of ICU Patients: the PhysioNet/Computing in Cardiology Challenge 2012," *Computing In Cardiology (CinC)*, 2012. [Online]. Available: http://physionet.org/challenge/2012/. [Accessed: 19-Feb-2013].

[5] M. T. Gilani, M. Razavi, and A. M. Azad, "A comparison of Simplified Acute Physiology Score II, Acute Physiology and Chronic Health Evaluation II and Acute Physiology and Chronic Health Evaluation III scoring system in predicting mortality and length of stay at surgical intensive care unit.," *Nigerian medical journal : journal of the Nigeria Medical Association*, vol. 55, no. 2, pp. 144–7, Mar. 2014.

[6] F. Ferreira, D. Bota, and A. Bross, "Serial evaluation of the SOFA score to predict outcome in critically ill patients," *Jama*, vol. 286, no. 14, 2001.

[7] L. Citi and R. Barbieri, "PhysioNet 2012 Challenge: Predicting mortality of ICU patients using a cascaded SVM-GLM paradigm," *Computing In Cardiology (CinC)*, vol. 39, p. 257,260, 2012.

[8] H. Xia, B. J. Daley, A. Petrie, and X. Zhao, "A Neural Network Model for Mortality Prediction in ICU," *Computing In Cardiology (CinC)*, no. 39, pp. 261–264, 2012.

[9] P. Addison, *The Illustrated Wavelet Transform Handbook*, 1st ed. London: Institute of Physics Publishing, 2002, p. 351.

[10] D. M. R. Devi, V. Maheswari, and P. Thambidurai, "Similarity search in Recent Biased time series databases using Vari-DWT and Polar wavelets," *Interact-2010*, pp. 398–404, Dec. 2010.

[11] R. Polikar, “The Engineer’s Ulitmate Guide to Wavelet Analysis.” Rowan University, Glassboro, p. 19, 2006.

[12] B. C. Si and E. Richard, “Scale-Dependent Relationship between Wheat Yield and Topographic Indices : A Wavelet Approach,” *Soil Science Society of American Journal*, vol. 68, no. 2, pp. 577–587, 2004.

[13] D. Mondal and D. B. Percival, “Wavelet variance analysis for gappy time series,” *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 5, pp. 943–966, Sep. 2008.

[14] J. Lee, P. Zhu, and J. C. Principe, “A parameter-free kernel design based on cumulative distribution function for correntropy,” in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–5.

[15] A. Fill and D. Fishkind, “The Moore--Penrose Generalized Inverse for Sums of Matrices,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 2, pp. 629–635, Jan. 2000.

[16] D. Tufts and C. Melissinos, “Simple, effective computation of principal eigenvectors and their eigenvalues and application to high-resolution estimation of frequencies,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1046–1053, Oct. 1986.

[17] J. Zhang and K. H. Lim, “Implmentation of a covariance-based principal component analysis algorithm for hyperspectral imaging applications with multi-threading in both CPU and GPU,” in *2012 IEEE International Geoscience and Remote Sensing Symposium*, 2012, pp. 4264–4266.

[18] K. Baker, “Singular value decomposition tutorial,” *The Ohio State University*, vol. 2005. Ohio State University, Columbus, pp. 1–24, 2005.

[19] G. Heo, P. Gader, and H. Frigui, “Robust kernel PCA using fuzzy membership,” in *2009 International Joint Conference on Neural Networks*, 2009, pp. 1213–1220.

[20] H.-B. Yan and Y.-S. Liu, “Image retrieval in data stream using principle component analysis,” *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pp. 2634–2637, Apr. 2012.

[21] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, “An introduction to kernel-based learning algorithms.,” *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 12, no. 2, pp. 181–201, Jan. 2001.

[22] K. B. Schebesch and R. Stecking, “Support vector machines for classifying and describing credit applicants: detecting typical and critical regions,” *Journal of the Operational Research Society*, vol. 56, no. 9, pp. 1082–1088, Jun. 2005.

[23] D. Representation, “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, p. 781, May 2005.

[24] T. Mu and A. Nandi, “EKF Based Multiple Parameter Tuning System for a L2-SVM Classifier,” in *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, 2006, pp. 229–233.

[25] B. Baesens, S. Viaene, T. Van Gestel, J. A. K. Suykens, G. Dedene, B. De Moor, and J. Vanthienen, “An empirical assessment of kernel type performance for least squares support vector machine classifiers,” in *KES’2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No.00TH8516)*, 2000, vol. 1, pp. 313–316.

Appendix A

The source code and test results are provided on the attached CD. Note: the source code was written in Python 2.7

VITA

Author: Conrad Sykes

Place of Birth: Spokane, Washington

Undergraduate Schools Attended: Eastern Washington University

Degrees Awarded: Masters of Computer Science, 2014, Eastern Washington University (In Progress)

Bachelor of Computer Science, 2012, Eastern Washington University

Honors and Awards: Graduate Assistantship, Computer Science Department 2012 – 2014, Eastern Washington University

EWU Linux Users Group President 2013 – 2014

EWU Linux Users Group Vice President 2012 – 2013

Graduated Cum Laude, Eastern Washington University, 2012

Graduated Valedictorian, Lewis & Clark High School, 2007

Professional Experience: Graduate Assistant, System Administrator, Eastern Washington University, Cheney, Washington, 2012 – 2014

Work Study, Spokane Virtual Learning, District 81, Spokane, Washington, 2007