

J S S Mahavidyapeetha

Sri Jayachamarajendra College of Engineering (SJCE), Mysore - 570 006

An Autonomous Institute Affiliated to
Visvesvaraya Technological University, Belgaum



“Support Vector Machine Approach to Automated Cardiac SPECT diagnosis”

Thesis submitted in partial fulfillment of curriculum prescribed for the
award of the degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING

by

Dharmatheja Bhat (4JC08CS023)

Varun B Patil (4JC08CS118)

Supreeth Nag V.P (4JC08CS107)

Vinay M (4JC08CS124)

Under the guidance of

Dr. M.P Pushpalatha

Associate Professor,

Department of Computer Science & Engineering,

SJCE, Mysore

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

June 2012

J S S Mahavidyapeetha
Sri Jayachamarajendra College of Engineering (SJCE)
Mysore - 570 006
An Autonomous Institute Affiliated to
Visvesvaraya Technological University, Belgaum



CERTIFICATE

This is to certify that the work entitled “**Support Vector Machine Approach To Automated Cardiac SPECT Diagnosis**” is a bonafied work carried out by **Dharmatheja Bhat, Supreeth Nag V.P, Varun B Patil, Vinay M** in partial fulfillment of the award of the degree of **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum during the year 2012**. It is certified that all corrections / suggestions indicated during CIE have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering degree.

Guide

Dr. M.P Pushpalatha

Associate Professor,

Dept. of CS & E,

S.J.C.E, Mysore

Head of the Department

Dr. C.N. Ravikumar

Professor & Head,

Dept. of CS & E,

S.J.C.E, Mysore

Place: Mysore

Date:

Examiners: 1.

2.

3.

DECLARATION

We hearby declare that the project work entitled “**Support Vector Machine Approach to Automated Cardiac SPECT diagnosis**” submitted to **Sri Jayachamarajendra College of Engineering, Mysore** is a record of original work done by me under the guidance of Dr. M.P Pushpalatha, Associate Professor, Dept. of Computer Science and Engineering, S.J.C.E, Mysore, and this project work has not performed the basis for the award of any Degree or Diplomo / associateship / fellowship and similar project if any.

Dharmatheja Bhat
Supreeth Nag V.P
Varun B Patil
Vinay M

Abstract

This project aims to develop an automated system for detection of cardiac disease given pre-labelled SPECT (Single photon emission computed tomography) data to train the system and then provide it with unclassified SPECT data for the purpose of classification as indicating possible cardiac disease or not. For the purpose of this classification, we have used a well known and highly used Machine Learning classification algorithm known as Support Vector Machine. Popularly abbreviated as SVM, Support Vector Machines are a class of supervised machine learning algorithms meaning that in the training stage of the classifier, we provide the system with SPECT data pre-classified as indicating cardiac disease and not indicating cardiac disease. That is, you are teaching the system; the system is learning from the inputs you provide. Once the learning is complete, you can provide the system with SPECT data that has not been labelled and the system will classify them automatically for you. The accuracy of this classification will obviously depend on the the accuracy of the training set images, the parameters of the Support Vector Machine, the complexity of the training algorithm and many more factors which will be detailed later. This project has a lot of relevance in todays world especially when corporations are providing million dollar prize money to anyone who can build an automated system that can detect virtually any disease Star Wars style (a.k.a Transcoder) and improvements in machine learning is getting us closer to that goal.

Acknowledgement

Apart from our efforts, the success of this project depends largely on the encouragements and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We extend our deep regards to **Dr. B.G Sangameshwara, Honorable principal of Sri Jayachamarajendra College of Engineering** for providing an excellent environment for our education and his encouragement throughout our stay in college.

We would like to show our greatest appreciation to **Professor and Head of the Department of Computer Science and Engineering Dr. C.N Ravikumar** for his tremendous support and help and most importantly the time and opportunity to work on such a fulfilling project. Without his encouragement and guidance, this project would not have materialized. We would also like to sincerely thank our **Project Guide and Associate Professor Dr. M.P Pushpalatha** for her constant encouragement, valuable insight and her experienced suggestions.

Finally, we would like to thank our friends for providing numerous insightful suggestions and our sincere thanks to all those who have contributed to this learning opportunity at every step of this project.

Dharmatheja Bhat
Supreeth Nag V.P
Varun B Patil
Vinay M

Contents

Abstract	i
Acknowledgement	ii
1 Introduction	1
1.1 Objective	1
1.2 Existing Solution	1
1.3 Proposed Solution	1
1.4 Applications	2
1.5 Project Timeline	3
2 System Requirements and Analysis	4
2.1 Functional Requirements	4
2.1.1 System Features	4
2.1.2 Use Cases	5
2.2 Hardware Requirements	6
2.3 Software Requirements	6
3 Tools and Technologies Used	7
3.1 Tools	7
3.2 Technologies	7
4 Literature Survey	9
5 System Design	11
5.1 Supporting Systems	11
5.2 SVM Structure	11

5.2.1	Inputs	11
5.2.2	SVM Algorithm	13
5.2.3	Choices in algorithm design	13
5.2.4	SVM parameters C and σ	14
5.2.5	Optimization Problem in SVM	15
5.2.6	Outputs	16
6	Alternative Designs	17
7	System Implementation	18
7.1	Pseudocode	18
8	System Testing and Results	22
8.1	SVM classifier using Gaussian Kernel	22
8.2	SVM classifier using Linear Kernel	24
8.3	aNN classifier	25
9	Applications	27
10	Conclusions and Future Work	29
	Bibliography	31

List of Figures

1.1	Project Timeline	3
7.1	State diagram for classifier working	20
7.2	Flow chart for SVM classifier	21
8.1	Accuracy vs. number of instances for SVM(gaussian kernel) .	22
8.2	Accuracy vs. number of instances for SVM(linear kernel) . .	24
8.3	Accuracy vs. number of instances for aNN	25

Chapter 1

Introduction

1.1 Objective

The objective of this project work is to build an automated system for automatically classifying SPECT heart data as possibly indicating cardiac disease or not.

1.2 Existing Solution

The existing solution is to have an experienced doctors trained eye and finely honed intuition recognize as possibly indicating a cardiac disease or not. Very few automated solutions exist, because they are not very accurate compared to an experienced human being and also because patients do not trust such systems with serious diseases relating to the heart. And it is not that a doctor is confronted with hundreds of patients with cardiac disease everyday. Thus, manually examining data is more efficient than spending considerable time and effort training an automated system that will be used only on a couple of patients a day.

1.3 Proposed Solution

The proposed solution is to use state-of-the-art in machine learning to build an accurate classifier that can do the job of a doctor in deciding whether a particular instance of SPECT heart data indicated cardiac disease or not. Though it is not built to replace a highly paid cardiac

specialist in hospitals anytime soon, the technology could well be used someday to dramatically reduce treatment costs and also improve accuracy of detection. We are actually moving closer to a day when the Star wars style handheld device (a.k.a Transcoder) which can detect any disease in an instant will become reality.

1.4 Applications

An automated classification system like the one we are building could be used by doctors as an auxiliary tool which can assist their finely honed skills. It can also be used to collect statistics from hospitals about cardiac diseases just from the instances of SPECT data instead of a doctor providing all the statistics which can be a laborious task and is un-necessary. Such an automated system can even one day be a part of a magic device that can help detect any disease. Most importantly information obtained from such a system can help less experienced doctors make informed decisions not that we are comfortable taking the opinion of a less experienced cardiologist.

1.5 Project Timeline

DETECTION OF HEART DISEASE FROM SPECT DATA
USING SUPPORT VECTOR MACHINES

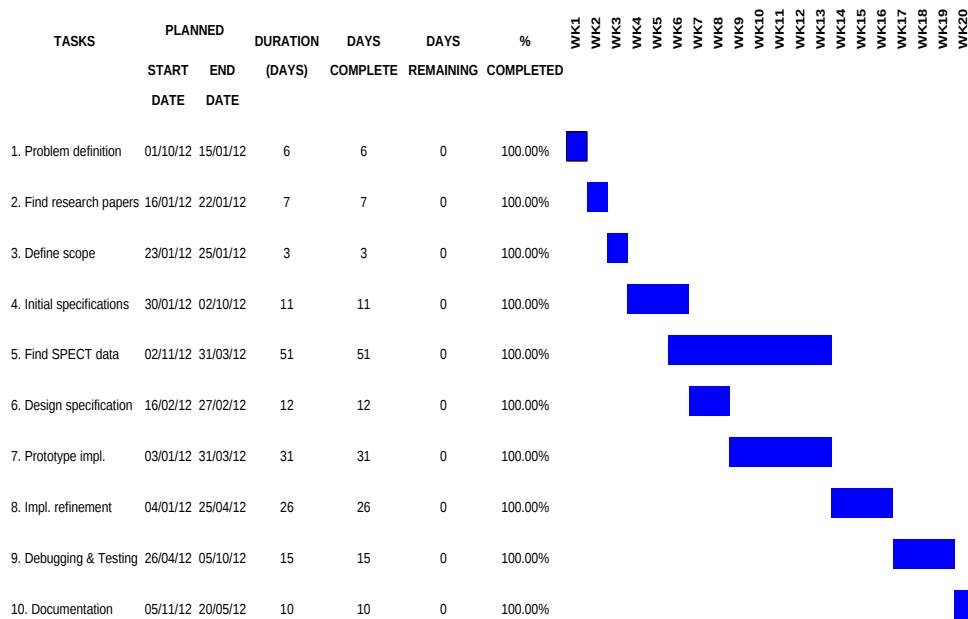


Figure 1.1: Project Timeline

Chapter 2

System Requirements and Analysis

2.1 Functional Requirements

2.1.1 System Features

- Purely open source because it uses GNU Octave as the implementation platform. This also makes it very portable.
- Allows classification of any data using SVM, but the current implementation is tailored towards classifying only SPECT data (specifically, only binary classification).
- Allows execution from command line. This is important because the program can be included as a part of any other script as a submodule. This makes it easy to be integrated with any other systems as a subsystem. For example, an automated system to identify all kinds of diseases can use this program to identify cardiac diseases.
- The program supports many kernels. Currently, we have implemented only two kernels namely Linear Kernel and Gaussian Kernel that allows us to switch between kernels on the fly depending on the particular problem at hand. We have found out that for our particular problem the Gaussian kernel (or non-linear kernel) is more effective for classification. However, the rule of thumb is that linear kernel is used when the number of features in the

dataset far exceed the number of available inputs. However, Gaussian kernel is more suitable when the number of inputs are very large compared to the number of features.

- SVM is a very good algorithm for this purpose, because it is one of those large margin classifier, which is capable of separating the classes to the maximum extent.
- SVM is a supervised(inductive) learning algorithm which means that you have to train it prior to using it for classification.

2.1.2 Use Cases

Use Case 1: Classify a given set of data from an input file.

Primary Actors: Cardiologist.

Pre-condition: GNU Octave and x-terminal installed.

Main Success Scenario:

1. Fire up a x-terminal.
2. cd to the directory in which the program is stored.
3. Launch the Octave program by typing in \$octave.
4. Then run the program as \$svm_g or \$svm_l based on whether you want to use gaussian kernel or linear kernel.
5. The program then reads from the file specified in the program, first in order to train the program(i.e, Training set), and then to classify (i.e, Test set).
6. Once prediction is complete, the prediction accuracy and other numerical precision measures are reported to the user.

Exception Scenario

1. If the data specified by the user is not in .csv format, the program won't be able to parse it and load the data into the matrix.
2. If the dataset is very small, especially for training, it can lead to suboptimal classification due to problem like overfitting and underfitting.

2.2 Hardware Requirements

- 512 MB RAM although 1GB RAM is preferable for better performance.
- 2.0 GHz processor speed.
- About 1GB of free storage space.

2.3 Software Requirements

- Any Linux distro preferably Ubuntu 12.04 or Fedora 16 or Linux Mint 12 which are the latest versions at the time of this writing (either 32bit or 64bit) or any version of Microsoft Windows operating system (Linux is preferred).
- Core library files for Octave (open source alternative to Matlab) should be installed on the system although Matlab can also be used (Octave is preferred).
- An X-window terminal should be installed to run the program and view the results.

Chapter 3

Tools and Technologies Used

3.1 Tools

The most important tool used to develop the program for this project is called Octave. Many people know it as an opensource alternative to Matlab. Apart from Octave being freely available and the core libraries occupying much less space than the Matlab software (which is a good thing for computers with limited disk space), it is known for being extremely efficient for numerical computations like the ones this project makes use of largely owing to expertly written numerical libraries for almost every advanced mathematical computation known to mankind. This saves you from re-inventing the wheel and also saves you a lot of time by allowing you to concentrate on refining the core concepts and not worry about the internal computations. But what makes Octave such a worthy alternative to Matlab is that it supports the exact same syntax as Matlab and can perform virtually anything you can do on Matlab in exactly the same way. Also, you will not need the bulk of the Matlab software to perform the computations for this project; the core libraries in Octave are more than sufficient.

3.2 Technologies

This project implements an advanced machine learning algorithm known as Support Vector Machines (SVM) with Sequential Minimal Optimization as the learning algorithm. SVM's have the capability to produce high accuracy classifiers that are fast as well. Ofcourse, there

are many variations in the parameters of the Support Vector Machine and the inputs that will ultimately decide the accuracy of the classifier. Nonetheless, Support Vector Machines are the state-of-the-art in supervised machine learning and is only fitting that such an important application is able to make use of it.

Chapter 4

Literature Survey

The main literature behind the idea and implementation of this project is the paper titled “*Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis*” [1]. This paper focusses on how data mining approaches can be used to mine for patterns in vast quantities of medical data to help professionals make informed decisions faster and also helps to discover new knowledge that would otherwise have gone unnoticed in the vast expanse of data. The writers claim that using highly optimized versions of Support Vector Machines can achieve the highest possible accuracy of all the other classification methods available.

The paper titled “*Improving the Classification of Multiple Disorders using Problem Decomposition*” [2] talks about using AIM Abductive Networks which is a supervised machine learning algorithms much like Support Vector Machines, but have an accuracy which is not very far from a well designed and highly tuned Support Vector Machine given that AIM abductive networks are substantially more complex than Support Vector Machines. The paper also hints about the GMDH type algorithms for Self-organizing methods in modelling which are advancements to Support Vector Machines (SVM).

A very recent paper titled “*Hierarchical Neural Networks for Partial Diagnosis in Medicine*” [7] makes use of very advanced neural network models like inductive neural networks to achieve higher accuracy than our simple Support Vector Machine based classifier.

Of course there is one other class of algorithms that is widely used to solve problems in medicine. The paper titled “*Differential Diagnosis of diseases using Genetic Algorithms and Support Vector Machines*” by Kim B.Y, Park K.S et al., gives a very broad overview of how Genetic algorithms can be used in conjunction with Support Vector Machines to help in computer-aided disease identification.

We have also looked at the following papers for a deeper and richer understanding of SPECT imaging and classification techniques – “Quantitative analysis in single photon emission tomography (SPECT)” [3], “A novel algorithm for classification of SPECT images of a human heart” [4], “Cardiac SPECT Imaging” [5], “Issues in Automating Cardiac SPECT Diagnosis” [8], “Analysing and improving the diagnosis of ischaemic heart disease with machine learning, Artificial Intelligence in Medicine” [6], “Diagnosing Myocardial Perfusion from SPECT Bulls-eye Maps - A Knowledge Discovery Approach”.

Chapter 5

System Design

The following describes the design of the SVM classifier running on Octave on Ubuntu 12.04LTS OS.

5.1 Supporting Systems

The classifier is written in Octave and thus requires the GNU Octave software to be installed on the system. Note that, even though an Octave program looks very much like a Matlab program, an Octave program cannot be executed on Matlab inside Windows. To install GNU Octave on Ubuntu do the following:

```
$ sudo apt-get install octave
```

5.2 SVM Structure

5.2.1 Inputs

- SPECT is actually images. It has to be preprocessed to generate numerical data that the program can work on.
- SPECT images are few and far between. For all computational machine learning purposes, UCI maintains a comprehensive database of numerical SPECT data collected from some of the best medical institutions around the world that we will be using.

- The data set we will be using was obtained from <http://archive.ics.uci.edu/ml/datasets/SPECT+Heart>.
- The UCI data set is a comma-separated-value file (CSV) which we read into the octave program as a two dimensional matrix using a built in function called `csvread()`.
- There are binary values for about 23 features indicating the presence or absence of a particular feature in that particular SPECT image instance. These features include left ventricular ejection fraction, stroke volume, myocardial perfusion, etc.
- The data is provided in two parts: Training set used to train the machine learning classifier and Test set used to test the classifiers accuracy.
- The Test Set itself is further divided into a Cross Validation set and a Test set. The CV consists of random samples from the Test Set.
- The cross validation(CV) set is primarily used to tune the classifier program for best performance; it is not used to gauge the accuracy of the classifier. For example, the CV set is used to select the best possible values for C and σ used in the SVM algorithm.
- The remaining part of the Test Set is the one that is actually used to measure the accuracy of the classifier after it has been trained on the Training Set.

- Usually the data is divided as training set(60%), CV set(20%) and test set(20%). The distribution between the sets is done randomly. Then, the hypothesis that minimizes the CV set error(fraction of CV set instances that are wrongly classified) is chosen and the error on the test set is reported as the generalized error.

5.2.2 SVM Algorithm

- Our aim in designing the SVM algorithm was to find best relationship between the features and its class so as to minimize the error in classification.
- The algorithm we employ is called SMO(Sequential Minimal Optimization) and it finds the best possible class to which a particular instance of SPECT data belongs to. The term “best class” here means the class which minimizes the classification error or likewise improves the classification accuracy.
- The two classes possible for any SPECT instance are: indicates possible cardiac disease, does not indicate cardiac disease(future enhancements to the project will include more fine grained classification or multiclass classification).

5.2.3 Choices in algorithm design

- The SVM algorithm we have implemented can either use a Linear Kernel or a Gaussian Kernel.

- A Linear Kernel can be imagined to be a straight-line separation between the classes (or a planar separation in the case of multidimensional data).
- The Gaussian Kernel on the other hand can be visualized as a non-linear separation between classes which also means a non-planar or curved separation between classes of multidimensional data.

$$K_{linear}(x^i, x^j) = x^i * x^j \quad (5.1)$$

$$K_{gaussian}(x^i, x^j) = \exp\left(-\frac{\|x^i - x^j\|^2}{2\sigma^2}\right) \quad (5.2)$$

We can switch the kernel on the fly meaning that the kernel can be changed for any particular execution of the algorithm. Now, the question arises as to which kernel to use, given that we have two choices.

- A Linear Kernel is usually employed when the number of features is more than the number of samples and a Gaussian Kernel is employed when the number of samples is more than the number of features.
- And also, the kernel used depends on whether the data itself is linearly-separable or not.

5.2.4 SVM parameters C and σ

We noted earlier that C and σ are two important parameters that need to be decided by examining the Cross Validation set.

Informally, the C parameter is a positive value that controls the penalty for misclassified training examples. A large C parameter tells the SVM to try to classify all the examples correctly (large margin). However, large C means that it is more susceptible to outliers and does not give a natural fit for the data.

σ is a parameter of the Gaussian Kernel which determines how fast the “similarity measure” approaches 0 for data points that are further apart. If σ is large, the similarity changes slowly and can cause underfitting. If σ is small, the similarity changes fast and can cause overfitting.

In practice, C and σ are selected from a set of highly possible values for both C and σ where the values vary approximately by a factor of 10, by testing all possible combinations of C and σ for the one that gives the least error in classification of the CV set.

5.2.5 Optimization Problem in SVM

The optimization problem in SVM is that of minimizing the following equation:

$$J = (C * A) + B \quad (5.3)$$

where,

$$A = \sum (y^i * cost_1(z) + (1 - y^i) * cost_0(z)) \quad (5.4)$$

$$B = \frac{1}{2} \sum (\Theta^2) \quad (5.5)$$

$$cost_1(z) = -\log(z) \quad (5.6)$$

$$cost_0(z) = -\log(1 - z) \quad (5.7)$$

Once we know Θ , we can predict the class as follows:

$$if(\Theta^T x) > 0, \text{predict class} = 1 \quad (5.8)$$

$$if(\Theta^T x) < 0, \text{predict class} = 0 \quad (5.9)$$

5.2.6 Outputs

When all is said and done, we expect the classifier to tell you the class to which a particular SPECT instance belongs to. One way to check whether what the classifier predicts is “upto the mark” is to compare the result with already known values. There are several ways to gauge this “upto the mark” notion in terms of numerical values.

- Prediction Accuracy is the fraction of the instances in the Test set that are correctly classified.
- Sensitivity is the fraction of positive instances in the Test Set that are correctly classified as belonging to the positive class.
- Specificity is the fraction of negative instances in the Test Set that are correctly classified as belonging to the negative class.

$$\text{Accuracy} = \frac{\text{Test set instances correctly classified}}{\text{total Test set instances}} \quad (5.10)$$

$$\text{Sensitivity} = \frac{\text{positive instances correctly classified}}{\text{total postive instances in Test set}} \quad (5.11)$$

$$\text{Specificity} = \frac{\text{negative instances correctly classified}}{\text{total negative instances in Test set}} \quad (5.12)$$

When you have high values for all of the above numerical measures, you know that your classifier is “upto the mark”.

Chapter 6

Alternative Designs

One of the most important alternative designs that we have prototyped prior to our full scale implementation of the SVM classifier is the aNN (artificial neural networks) version of the classifier because of its comparable complexity to the SVM algorithm.

The first thing we observed with the neural networks prototype is that its accuracy heavily depends on its structure which is almost decided on “intuition” alone. However, machine learning experts will testify to the fact that trusting your intuition is not a good idea. The upshot is that a neural network structure that seems perfect for one problem might perform terribly on another problem. It is just not possible to stick to one structure.

On the contrary, SVM does not rely on “intuition”. It is able to decide on the best parameters using information from the problem itself. In our case, the best possible values for C and σ were determined from the problem itself(CV set).

Ofcourse several advanced algorithms can perform better and faster, but our goal has been to demonstrate the use of machine learning and data mining to improve and ease disease diagnosis and not designing the ultimate cardiologist substitute.

A complete graphical comparison of the results of using SVM algorithm and the aNN algorithm will be presented in the testing and results section of this report along with an in-depth comparison of the Linear and Gaussian kernels for the SVM algorithm.

Chapter 7

System Implementation

Our SVM implementation for the SPECT Heart classifier involves a completely linear execution style which can be put in summary as:

- Take input and store it as a two dimensional Matrix.
- Train the classifier on the training set.
- Run the classifier on the test set.
- Report the accuracy.

The above four steps are expatiated in the pseudocode below.

7.1 Pseudocode

- **Step 1 : Accept Inputs**

Description : As described previously, the file containing the features, their values and the class to which the instance belongs is in a comma-separated-value (.csv) format. But the program expects the inputs in the form of a 2-Dimensional matrix. The Octave function that does this conversion is called “csvread()”

which takes as input a csv formatted file and outputs a two dimensional matrix.

- **Step 2 : Train the classifier**

Description : Now that the training data is available, we can start running the SVM algorithm on this training set. The purpose of this training phase is to identify those values of C and σ that give the least error on the training set. C and σ are vectors with a set of possible values which increase exponentially or increase as factors of 10. The training phase uses a combination of all these values of C and σ in a loop and use the kernel function specified in the algorithm to train the classifier for the least possible error. In our implementation, we have the following set of values for C and σ .

$$C = [0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30] \quad (7.1)$$

$$\sigma = [0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30] \quad (7.2)$$

- **Step 3 : Classify the test set**

Description : Now that we have the correct values of C and σ , we can proceed to classify the test set. Note that, we already have the classes of these test set instances. We compare these classes with the classes reported by the classifier algorithm.

- **Step 4 : Report accuracy**

Description : Now that we have the classes reported by our classifier and the classes mentioned in the test set, we can see how well our classifier performed by using certain numerical measures such as accuracy, sensitivity and specificity that have been described previously.

A detailed flowchart of the above steps is shown in Figure 7.2 below.

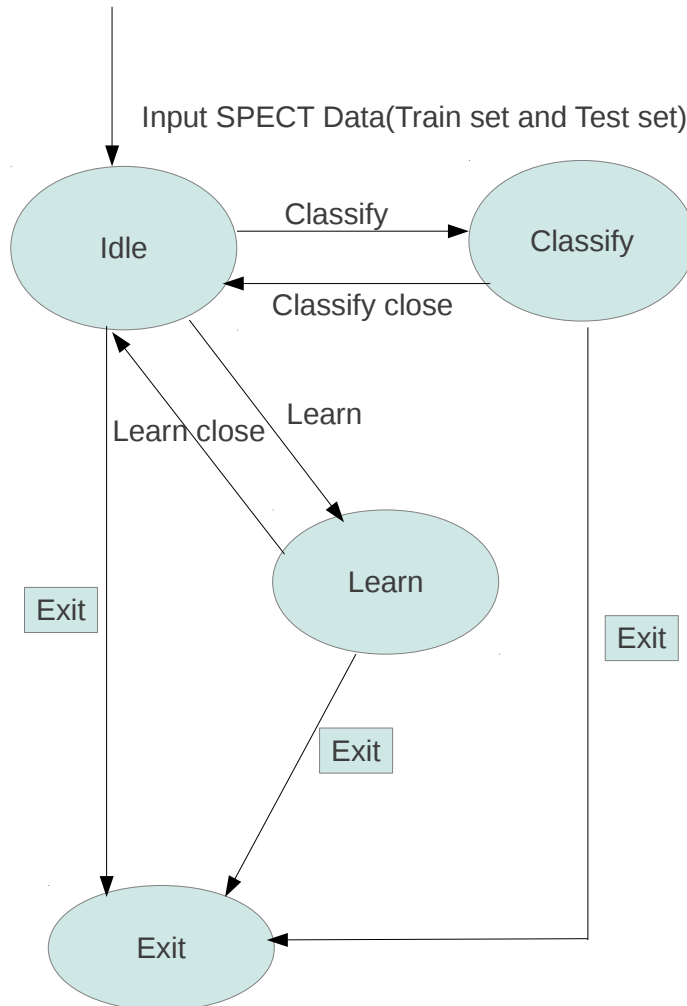


Figure 7.1: State diagram for classifier working

Figure 7.1 shows a simple state diagram for the working of our classifier. The “idle” state implies that state when the classifier is still unusable, meaning that it has not yet been trained. The “Learn” stage indicates that the classifier is in the process of being trained on the training set. Once the classifier has been trained, it can enter the “Classify” stage to begin classifying the test dataset.

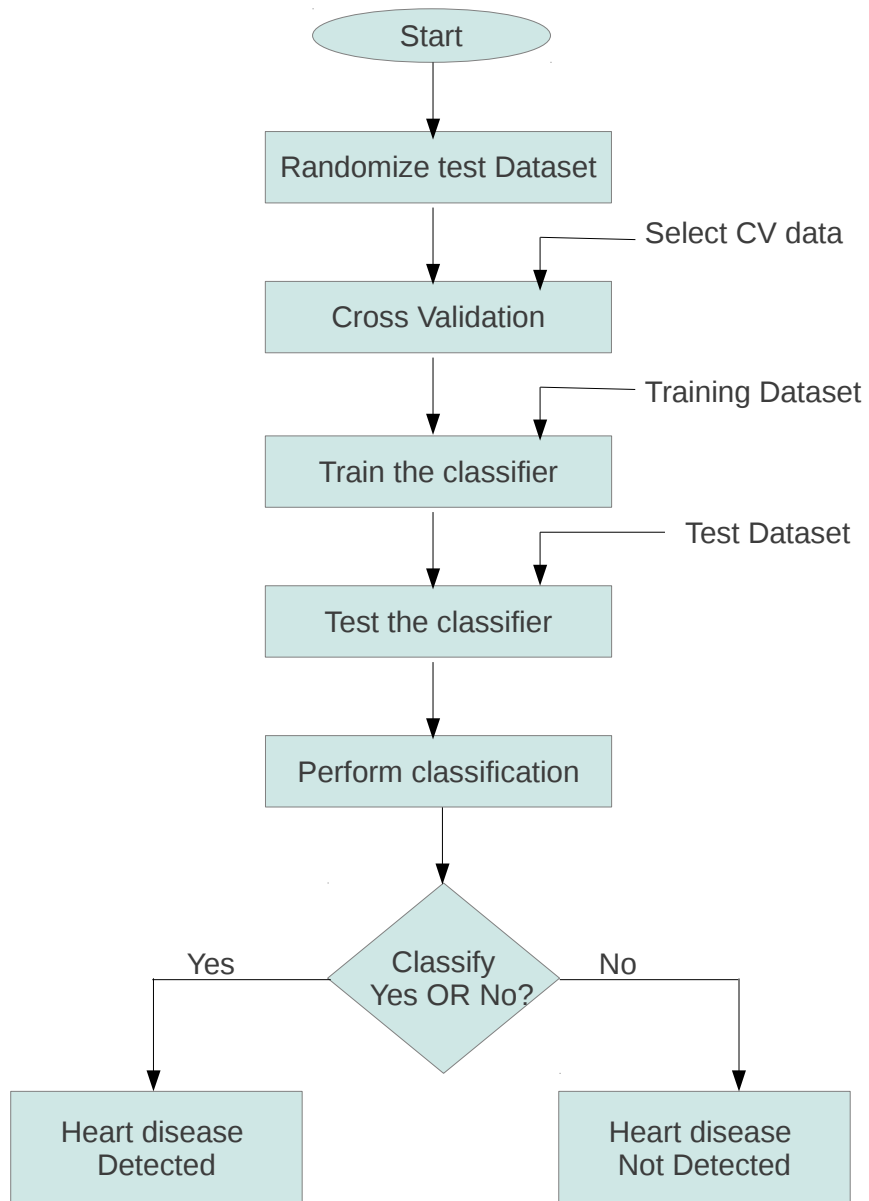


Figure 7.2: Flow chart for SVM classifier

Chapter 8

System Testing and Results

The results in our algorithm are mainly predictions accuracies. In this section I will provide graphical representation of results for the SVM implementation (both the Linear and Gaussian kernel versions) and then compare them with the results obtained by the aNN implementation.

8.1 SVM classifier using Gaussian Kernel

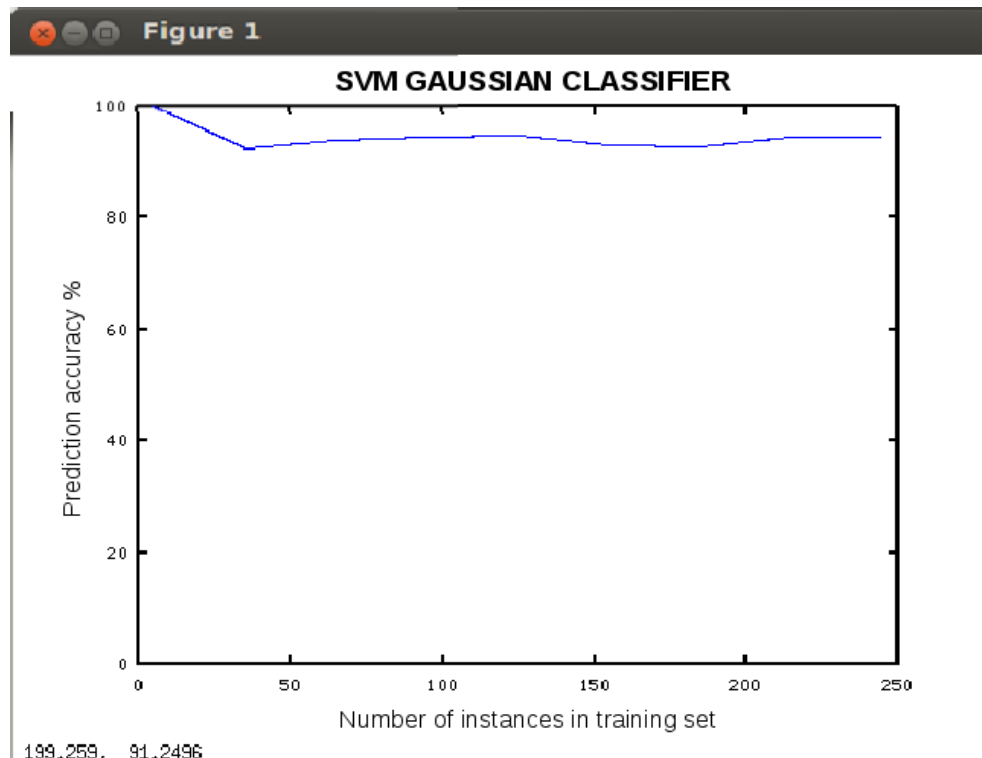


Figure 8.1: Accuracy vs. number of instances for SVM(gaussian kernel)

Figure 8.1 shows the prediction accuracy for the SVM implementation using a non-linear (also called Gaussian) kernel. We can see that the accuracy is a constantly above 90%. One important thing to notice that the accuracy is almost 100% for a small number of instances. This is an idiosyncrasy in the data set. The data set consists of a huge number of samples that belong to class 1 (i.e, heart disease present), whereas there are very few examples in the data set that correspond to class 0 (heart disease not present), and hence when only a few instances are considered, it will most likely contain only instances that belong to class 1 and no instances that belong to class 0 which explains the reported accuracy of 100%. This is not a true indication of the accuracy of the classifier. The true accuracy only becomes evident for a large number of training samples. Another thing to notice is that the graph is not smooth, rather it is spikey. The reason for this is that the accuracy is measured for an increase in number of instances by 30. This is because, if the graph was plotted in Octave for an increase in the number of instances of 1, it would take an unimaginably long amount of time to run through the algorithm with a total of close to 300 instances.

8.2 SVM classifier using Linear Kernel

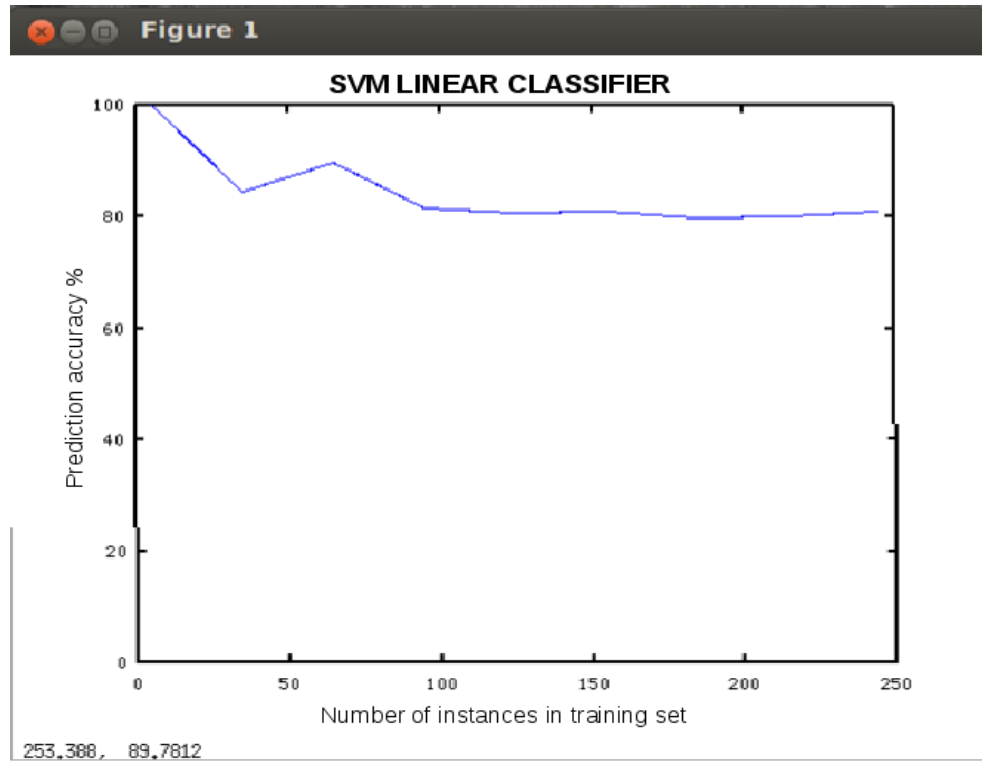


Figure 8.2: Accuracy vs. number of instances for SVM(linear kernel)

Figure 8.2 shows the prediction accuracy for the SVM implementation using a linear kernel. Clearly, the accuracy is lesser when compared to that produced by the Gaussian Kernel version of the SVM classifier. The reason for this is simple: The dataset is just not linearly separable. i.e., there does not exist a straight line or a plane that perfectly separates the dataset instances into two classes. Only a non-linear separator (like the one produced by the Gaussian Kernel) can produce a good separation of classes. Also, with a very few number of instances, the reported accuracy is close to 100%. This is again because of the dataset idiosyncrasy as in the case of the Gaussian Kernel above. As the number of instances increases, the true accuracy of the classifier is revealed.

Again, to keep execution time reasonable, the number of instances are increased in steps of 30 which explains the blockiness of the graph.

8.3 aNN classifier

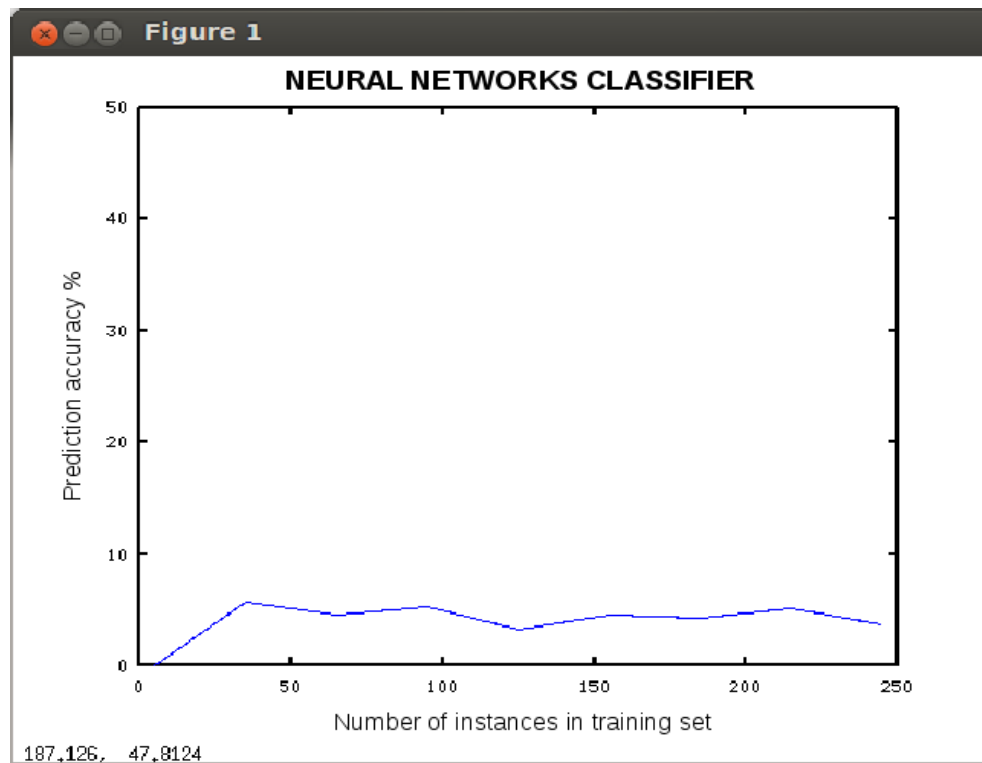


Figure 8.3: Accuracy vs. number of instances for aNN

Figure 8.3 shows the prediction accuracy for the aNN prototype implementation. Clearly, the accuracy is a whole lot inferior to those reported by any of the SVM classifiers above. The reason is that, the structure of the aNN is not fixed. It is decided by the programmer. We have gone with a NN structure with 3 stages, with the middle stage containing 50 neurons. The simple matter is, If you get the structure of the NN right, you get good accuracy, you get the structure wrong, you can only expect below average accuracies. The upshot is that, the NN

structure is decided by the programmer by intuition, which is not what we want in professional software programs. It is very difficult to come up with the right NN structure for the particular problem at hand. On the other hand, SVM implementation is free from such parameters that are decided by the programmer's intuition. The parameters of the SVM classifier are determined by the dataset and not by the programmer and thus easily adapts to different problems, unlike a NN implementation. This is not to say that, NN are always inferior. Several advanced NN strategies exist to overcome all the problems listed above, but with increased complexity.

Chapter 9

Applications

Applications of Machine Learning and Data Mining in healthcare are only starting to appear in the real world. Machine Learning is still in its embryonic stages. Classifiers like the one we have implemented can be used by professional physicians

- To make an informed decision regarding the diagnosis of a patient, rather than just relying on intuition.
- To discover hidden information and patterns in the vast quantities of medical data available throughout the world, which is not possible by any human per se, because of the inability to make sense of such vast quantities of numerical data.
- As an aid to discover new relationships in the data that can provide the physician with previously unknown information about the patient's characteristics and thus enable him/her to provide the best possible care to the patient.

As an example, consider that a cardiologist has details about a patient like race, sex, age, previous medical history etc. He/she can make use of the vast knowledge database provided by medical institutions around the world to make a diagnosis. Through software programs like ours the cardiologist can be provided with information such as races, age groups, patient sex and combinations of the previous that are more likely to have a heart disease. Previous medical history of

the patient can be correlated with those of patients around the world and the accuracy of diagnosis on those patients can be considered while making a diagnosis for the current patient. Drugs prescribed and the outcome of those drugs are available in medical databases which can be used by software programs to provide sensible information regarding drugs that the patient is more likely to respond to in a positive manner.

Chapter 10

Conclusions and Future Work

Our classifier only performs two class classification i.e, whether the results indicate a possible heart disease, or does not indicate any heart disease. Also the data used to make this classification only involves numerical data obtained directly by processing the SPECT heart images. Considering the above, we have the following areas in which the classifier can be improved in future versions :

- Classification accuracy can further be improved by including the persons age, racial background, sex, pathophysiology, diet, air pollution in place of dwelling and previous medical history as features that aid the classification. Such data though hard to obtain due to privacy issues can be a major accuracy booster. Such characteristics about patients can only be fully understood and made use of through a computer software that can crunch through vast quantities of data and provide the physician with easy-to-use, sensible information.
- Another feature that has been debated for a long time in medical research is the demography in which the person has been brought up. It is obvious why this is so important. Take for example the fact that a malaria outbreak in America is more likely to reach pandemic proportions when compared to Africa or the fact that Africans are more suited to marathon running compared to Indians. To put it simply, it is just in their genes !!!.

- Another simple improvement would be to provide a more fine-grained classification involving several classes of cardiac diseases instead of just a yes/no solution. Of course the training data should support such a classification by providing training instances for each of those classes. Using this allows us to abstain from making a very generalized prediction. We will be able to tell whether the patient suffers from Cardiomyopathy or Cardiac dysrhythmias or Endocarditis or Inflammatory cardiomegaly, and many more.

Bibliography

- [1] Kurgan L. A., Cios K. J., Tadeusiewicz R., Ogiela M., and Goodenday L. S. Knowledge discovery approach to automated cardiac spect diagnosis. *Artificial Intelligence in Medicine*, 23(2):149–169, 2001.
- [2] Radwan E. Abdel-Aal, Mona R. E. Abdel-Halim, and Safa Abdel-Aal. Improving the classification of multiple disorders with problem decomposition.
- [3] J. A. K. Blokland, J. H. C. Reiber, and E. K. J. Pauwels. Quantitative analysis in single photon emission tomography (spect). *Eur J Nuclear Medicine*, 19:47–61, 1992.
- [4] K. J. Cios, L. S. Goodenday, K. K. Shah, and G. Serpen. A novel algorithm for classification of spect images of a human heart. *Proc. of the CBMS96*, pages 1–5, June 1996.
- [5] J. R. Corbett, DePuey E. D., Berman D. S., and Garcia E. V. *Cardiac SPECT Imaging*. Raven Press, New York, 1995.
- [6] M. Kukar, I. Kononenko, C. Groselj, K. Kralj, and J. Fettich. Analysing and improving the diagnosis of ischaemic heart disease with machine learning. *Artificial Intelligence in Medicine*, 16(1):25–50, 1999.
- [7] Lucila Ohno-Machado and Mark A. Musen. Hierarchical neural networks for partial diagnosis in medicine. *Section on Medical Informatics, Stanford School of Medicine*.
- [8] J. P. Sacha, K. J. Cios, and L. S. Goodenday. Issues in automating cardiac spect diagnosis. *IEEE Engineering in Medicine and Biology Magazine, special issue on Medical Data Mining and Knowledge Discovery*, 19(4):78–88, 2000.