

1-1-2010

Pattern classification of time-series signals using Fisher kernels and support vector machines

Yashodhan Rajiv Athavale
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>

 Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Athavale, Yashodhan Rajiv, "Pattern classification of time-series signals using Fisher kernels and support vector machines" (2010).
Theses and dissertations. Paper 819.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

PATTERN CLASSIFICATION OF TIME-SERIES SIGNALS USING FISHER KERNELS AND SUPPORT VECTOR MACHINES

Yashodhan Rajiv Athavale
B.Tech, Visvesvaraya National Institute of Technology
India, 2007

A thesis
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Masters of Applied Science
in the program of
Electrical and Computer Engineering

Ryerson University
Toronto, Ontario

2010

©Yashodhan Rajiv Athavale, 2010

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature

Instructions for Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and mention address and date.

Abstract

PATTERN CLASSIFICATION OF TIME-SERIES SIGNALS USING FISHER KERNELS AND SUPPORT VECTOR MACHINES

©Yashodhan Athavale, 2010

**Masters of Applied Science
Electrical and Computer Engineering
Ryerson University**

The objective of this study is to assess the performance and capability of a kernel-based machine learning method for time-series signal classification. Applying various stages of dimension transformation, training, testing and cross-validation, we attempt to perform a binary classification using the time-series signals from each category. This study has been applied to two domains: Financial and Biomedical.

The financial domain study involves identifying the possibility of collapse or survival of a company trading in the stock market. For assessing the fate of each company, we collect its real stock market data, which is basically a financial time-series composed of weekly closing stock prices in a common time-series interval. This study has been applied to various economic sectors such as Pharmaceuticals and Biotechnology, Automobiles, Oil & Gas, Water Supply etc. The data has been collected using Thomson's Datastream software.

In the biomedical study we are dealing with knee signals collected using the Vibration arthrometry technique. This study involves using the severity of cartilage degeneration for assessing the possibility of a subject getting affected by Osteoarthritis or undergoing knee replacement surgery at a later stage. This non-invasive diagnostic method can also prove to be an alternative to various invasive procedures used for detecting osteoarthritis. For this analysis we have used the vibroarthro-signals for about 38 abnormal and 51 normal knee joint case studies.

In both the studies we apply Fisher Kernels incorporated with Gaussian Mixture Model (GMM) for dimension transformation into feature space created as a three-dimensional plot for visualization. The transformed data is then trained and tested using support vector machines for performing binary classification. From our experiments we observe that our method fits really well for both the studies with the classification error rate between 10% to 15%.

Acknowledgment

I would like to acknowledge my supervisors Dr. Aziz Guergachi and Dr. Sridhar Krishnan who have been a strong source of inspiration throughout my research. Their splendid guidance and motivation has helped me achieve my career goals and succeed in all my endeavors. Also, I thank NSERC, CFI, OIT and Ryerson University for funding this research.

I would also like to acknowledge Dr. Javad Alirezai, Dr. Kamraan Raahemifar, Dr. Kristina McConville, Dr. Matthew Kyan and Dr. Sergiy Rakhmayil for guiding me through my courses at Ryerson University.

I would also like to thank my fellow colleagues from the Signal Analysis Research (SAR) group - Peyman Shokrollahi, Dr. Karthi Umapathy, Nasim Shams, Elnaz Shokrollahi, Mehrnaz Shokrollahi, Andrea Niu, Farhat Kaleem and Lakshmi for their fun-filled and enlightening company. Also sincere thanks to my colleagues from Research Lab for Advanced System Modelling - Pouyan Hosseinizadeh, Dr. Vikraman Bhaskaran, Dr. Muhammad Shahbaz and Lyubomir Halachev.

I would like to thank my friends Swapnil, Ambarish, Ashis, Krishna, Hartej, Sukrit, and Yogesh for their unconditional support during my studies at Ryerson University. And last but not least I would like to thank everyone at Ryerson University for giving me a superb educational experience in world-class facilities.

Dedication

To my parents, Rajiv Athavale and Chhaya Athavale, my sister Tanvi Athavale, and my grandparents for their continual support and encouragement, which has given me the ability to pursue my goals with confidence and full determination.

Contents

1	Introduction	1
1.1	What is a Complex System ?	1
1.2	About the research - a brief overview	3
1.3	Objective and Organization of the Thesis	6
2	Literature survey, Motivation and Problem definition	9
2.1	Literature survey	9
2.2	Motivation and Problem definition	12
3	Background theory and Methodology	14
3.1	Introduction to Machine Learning	14
3.2	Kernel-based Machine learning	16
3.2.1	Fisher Kernels	19
3.2.2	Support vector machines (SVMs)	21
3.3	Proposed Work	24
3.3.1	Applying Fisher Kernels to a time-series model	25
4	Application to Financial domain - Stock Markets	30
4.1	About Stock Markets	30
4.2	Application of Fisher kernels and SVMs to financial time-series	32
4.2.1	Data collection and Processing	33
4.2.2	Plot generation, SVM application and LDA validation	38
4.3	Observations and Discussions	45
5	Application to Biomedical domain - Osteoarthritis	49
5.1	About Osteoarthritis and other joint-related disorders	49
5.2	Studying Osteoarthritis using intelligent methods - Literature survey	52
5.3	Application of Fisher kernels and SVMs to knee-signals for classification and predictions	54
5.3.1	Signal collection and Processing	55
5.3.2	Plot generation and SVM application	57
5.3.3	LDA analysis and cross-validation	60
5.4	Observations and Discussions	61

6	Discussions, Conclusions and Future Work	64
7	Appendix A	71
7.1	Machine Learning Design	71
7.2	Designing the optimum classifier	73
7.3	Types of Machine Learning techniques	75
8	Appendix B	79
8.1	Expectation Maximization Algorithm	79
8.2	Sequential Minimal Optimization	80
8.3	Linear Discriminant Analysis (LDA)	80
8.4	Cross-validation techniques	82
8.4.1	Leave-one-out technique (LOO)	83
8.4.2	Hold-out technique	83
	Vita	86

List of Figures

1.1	Organization of Thesis	7
3.1	Classification between Salmon and Sea-bass [1]	16
3.2	Dimension Transformation	17
3.3	Transformation in to Feature Space [2]	18
3.4	Application of Kernel Methods	20
3.5	Linearly Separable case	22
3.6	Non-linearly separable case	22
4.1	Methodology for classifying companies	33
4.2	Sample Stock price data: Pharmaceuticals and Biotechnology	36
4.3	Stock price distribution of Active companies	37
4.4	Stock price distribution of Dead companies	37
4.5	Log-Normal stock returns distribution of Active Companies	38
4.6	Log-Normal stock returns distribution of Dead Companies	39
4.7	Fisher score plot for visualization - Separate Gaussian estimates	40
4.8	Fisher score plot for visualization - Common Gaussian estimates	43
5.1	OA development [3]	51
5.2	X-ray image showing OA affected knee joint [4]	53
5.3	Methodology for classifying knee joints	55
5.4	Sample knee signal data	56
5.5	Normal Knee Signals	57
5.6	Abnormal Knee Signals	57
5.7	Sample normal knee signal	58
5.8	Sample abnormal knee signals	58
5.9	Fisher score plot for visualization	59
6.1	Fisher Score plot: Financial Time Series	66
6.2	Fisher Score plot: Knee Signals	68
6.3	Histogram of a sample stock returns distribution	69
7.1	Typical Stages: Pattern Recognition	71
7.2	Classifier design cycle	74

7.3	Supervised Learning [5]	76
7.4	Unsupervised Learning	77
7.5	Semi-Supervised Learning	77
7.6	Reinforcement Learning [5]	78

List of Tables

3.1	List of variables used for Fisher scores' computation	26
4.1	Typical search criteria in Thomson Datastream	35
4.2	Search Criteria for Pharmaceuticals and Biotechnology sector	35
4.3	Results from SVM performance - Financial time series	41
4.4	LDA results for 3-dimensional Fisher scores - Financial time series	41
4.5	LDA results for 6-dimensional Fisher scores - Financial time series	42
4.6	Results from SVM performance - Financial time series	43
4.7	LDA results for 3-dimensional Fisher scores - Financial time series	44
4.8	LDA results for 6-dimensional Fisher scores - Financial time series	45
5.1	Results from SVM performance - Knee Signals	60
5.2	LDA results for 3-dimensional Fisher scores - Knee Signals	61
5.3	LDA results for 6-dimensional Fisher scores - Knee Signals	62

Chapter 1

Introduction

1.1 What is a Complex System ?

A COMPLEX system is basically a network of various independent components which interact with one another in a non-linear fashion, thus creating an emergent behavior [6]. This definition of a complex system is applicable to various domains such as animal colonies, human civilizations, climate, human body, as well as modern infrastructures and computing systems. Modelling many of these systems to monitor their performance is of human interest. Many multi-disciplinary fields such as systems theory, cybernetics, ecology and complexity theory include the study of complex systems as an essential component. Complex systems, which always pose a challenge for scientific modelling, usually display a trend of random behavior in their working. Assessing the fate and performance of such systems with a more holistic approach is not an easy task considering the various internal and external factors affecting these systems.

Complex systems are studied using the following interrelated approaches [6]:

- Behavioral patterns obtained through interactions
- Studying the ways of analyzing complex systems
- Building of complex systems from pattern formation and evolution

The main properties of complex systems are [7]:

- A complex system is fundamentally non-deterministic. It is difficult to predict accurately the behavior of such systems even if the underlying properties are completely known.
- The relationships that exist within the elements of a complex system are non-linear, and may contain extensive feedback loops in the working structure.
- It is difficult to study the properties of a complex system by decomposing it into functionally stable parts.

The study of complex system behavior has essentially led to the development and advancement of electronic computing techniques - Direct Programming and Machine Learning [8, 9, 7]. While the former method relies more on human inputs and coding, the latter technique is more dependent on large amounts of examples, which help the machine learn the complex system behavior from experience, thus creating a better learning model. Direct programming basically involves a human expert who creates a set of computer instructions in a high-level or low-level language, based on a small set of examples. These instructions generate an algorithm which exhibits a simple "give the input and take the output" pattern. It does not involve learning in any form and thus may lead to inaccurate modelling of complex systems.

Moreover if the complexity of the systems is such that there is no known method for finding the desired emerging real-time signal, then such issues cannot be resolved by a traditional programming methodology, since the human expert cannot define a specific method by which correct output can be computed. For example, if we consider a complex voice recognition system, for securing the entrance to a building, then applying just a simple software program to the system is not sufficient to ensure security. We need to create an algorithm which would learn from a substantial set of examples, and improvise on its output capabilities. For example, consider a group of children who learn to identify the sounds of two kinds of animals - dog and cat, and are later able to correctly identify and map the sounds to the respective species, simply by listening to the voice of the animal. Such an

approach of using examples to generate intelligent programming instructions, is known as *learning methodology* [8], and the examples themselves constitute the *training data* [8].

Assessing the fate and behavior of complex systems such as environmental systems, human body, stock markets or even countries' economies has always posed a major challenge. Continuous research work in these domains in order to monitor the complex systems' performance definitely aides in achieving certain standards and benchmarks. In this research various computer based machine learning techniques, such as kernel-based methods, discriminant analysis and cross-validation, are applied to model some interesting time-series systems, and predict their future states. Before we proceed further, let us ponder over the concept of time-series, and how they are applicable to our research work.

A *time series* is basically a sequence of data points, measured at successive uniform time intervals. Real-time signals are highly useful in monitoring the behavior of complex systems over a given period [10, 11]. They can be used for analysis and forecasting of complex systems. One such example would be predicting the future stock price of a company, based on its historical stock price distribution. In this case the historical distributions would be the real-time signals emerging from the company as a result of its ongoing business in the market. These financial time-series signals are affected by various external factors in the stock market, such as political events, rumors, etc. and need to be considered when we are modelling the signals.

The time-series signals used in our research work are non-deterministic in nature, i.e. a comprehensive knowledge of their properties needs to be collected for defining certain explicit mathematical relationships, within the limits of uncertainty. Modelling signals in this way basically constitutes the principles of a stochastic process [12, 10, 11].

1.2 About the research - a brief overview

In our experiments we have taken time-series signals emerging from two such complex systems. These time-series signals are processed using intelligent techniques for generating certain classification patterns and determining the fate of the complex systems.

- **Stock Markets - Financial time series signals**

One of the most important places for firms to make profits in businesses would be stock markets. Earnings can be achieved by trading publicly or by selling stocks of ownership in the public market. The stock market behavior exhibits a highly random behavior, with respect to the constant fluctuations in the stock prices. As a result, predicting or simply guessing these stock prices based on the company's profitability or any other factors is not sufficient enough. Extensive study of this random behavior and considering some concrete evidence about the company's profile (such as stock price distributions, income statements, etc.), can give us a better idea of modelling this chaotic behavior, and hence put it to good use for the investor. Trade analysts rely heavily on various fundamental and technical methods of deciphering stock market behavior, in order to reveal more insights into the influx-deflux of businesses.

Monitoring stock market behavior is not just restricted to predicting stock prices, or bankruptcy forecasts, or even profitability. Other avenues that open up include assessing the fate of businesses in stock markets, and making predictions about the future prospects of companies, in various economic sectors. In particular, our research does not involve predicting stock prices of a company, but rather focuses on generating statements about the potential survival or collapse of the firm. The input to the learning algorithm in this study is a financial time-series comprising of historical stock prices of companies in a given time frame (say about 20 years or so). As we proceed further, we will see how various financial time-series are transformed into a three dimensional space, thus giving us a classification pattern between active and dead companies.

- **Knee Joints - Biomedical signals**

Vibration arthrometry signals or in short, Vibroarthrographic signals (Vibroarthrogram or VAG) refer to the vibration signals captured at the human body joints (knees, hips, elbows) using an electronic accelerometer, during the joints' normal movements. These signals reveal great deal of information about the joint condition, and when applied to a learning algorithm as inputs, help in assessing the severity of cartilage

degeneration. This further helps us to predict the chances of a person getting affected by Osteoarthritis and other joint-related disorders, and thereby undergoing invasive treatments such as joint replacement surgery.

Early detection of such disorders can prove to be useful in addressing the degeneration severity and thereby promoting more non-invasive treatment options such as physiotherapy. Recent studies on similar topics focus on just addressing the severity of cartilage disorders. Our study on the other hand emphasizes predicting the possibility of a subject requiring joint replacement surgery, by applying the cartilage degeneration severity. In our experiments the input signal to the learning machine is a biomedical time-series signal comprising of knee signals collected using vibration arthrometry. As mentioned in previous application of stock markets, we will study a three dimensional classification plot in this study as well.

The above mentioned complex systems used in our research, have been modelled using extensive mathematical techniques which try to analyze and decode useful information from the signals coming out of these systems. Before the time-series is reduced to a complex mathematical problem, we first transform this time-series into a three dimensional vector space (feature space), using Fisher kernels, such that the separation of categories is easier.

The data points (Fisher scores) available in this feature space helps us in creating the required intelligent classification model using support vector machines (SVMs). The mathematical techniques are essentially a part of the machine learning process, as they assist in designing learning algorithms for classifying and predicting patterns in the complex systems. They are an important component in reducing the complex mathematical problem to a more solvable one.

In summary, one can infer that this research work is focussed on converting a complex time-series analysis problem into a pattern classification problem, which would reveal the dynamics of the complex system in question and thus help in generating statements about the future state of that system. For doing these assessments and conversions, computer-based tools have been developed. The core benefit of these tools is that they can deal with

variable-length time series data, and still assess the dynamics of the complex system. In the first part of our experiments, we are dealing with financial data as mentioned above.

Based on the promising results obtained with the financial data, we proceed with the second part of this thesis wherein we investigate if our proposed work could be extended to complex systems in other domains. For this, we have used the biomedical time-series signals for classifying abnormal and normal knee joints. We again got good classification results from this study as well. The following section describes the objective and organization of this thesis.

1.3 Objective and Organization of the Thesis

The main objective of this thesis is to discuss a kernel-based machine learning approach for classifying a time-series signal emerging from a complex system. We can assume the complex system to be a black-box, wherein the properties of its various components are unknown, but the signal remains non-deterministic. For example, if we assume the stock market to be a black-box system, then we know the properties of its components such as the different economic sectors present, the companies that trade, types of stock prices (daily, weekly, monthly, quarterly and yearly) and the factors (such as political events, rumors, business strategies, etc.) affecting the stock prices. The time-series signal emerging from this black-box system is in the form of historical stock prices of different companies in an economic sector, but the stock price behavior is non-deterministic in nature, i.e. we cannot identify any specific correlation between two consecutive stock prices, and moreover predicting future stock prices by merely observing a historical trend is not possible. In order to model this signal, we apply various concepts of machine learning such as dimension transformation, training and testing the machine. This study has been applied to two domains: Stock Markets (Financial time series), and Knee Joints (Biomedical time series).

This thesis has been organized into the following chapters as illustrated in Figure 1.1.

- **Chapter 2 - Literature survey, Motivation and Problem definition**

This chapter gives a brief description of the complex system problem to be addressed,

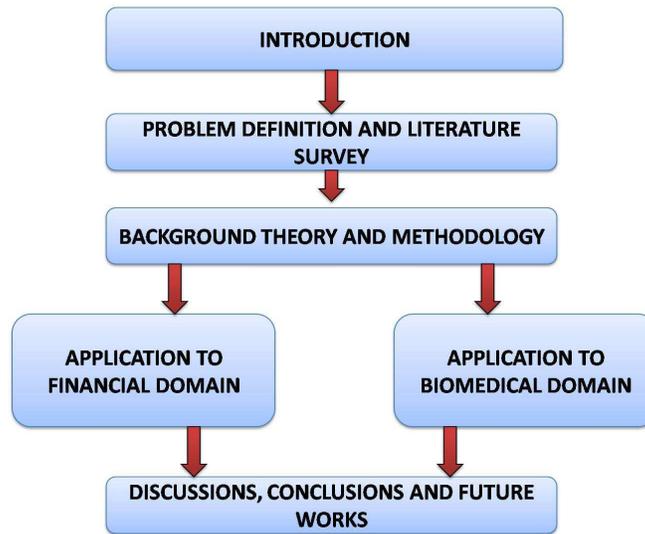


Figure 1.1: Organization of Thesis

related studies that have been done so far in analyzing the complex system behavior, the limitations imposed by these works and the motivation to apply our proposed model to study the complex system behavior.

- **Chapter 3 - Background theory and Methodology**

This chapter gives a detailed description of the field of Machine Learning. The purpose of including this chapter is to make the reader aware of the background theory and concepts applied to this research work, for a better understanding of the application domain. We discuss the various types of learning methods and their applications, with an emphasis on *kernel-based machine learning*.

- **Chapter 4 - Application to Financial domain**

As the name suggests, we deal with financial time-series data in this chapter. The financial time-series signals have been obtained from stock markets (NYSE [13], TSX [14] and NASDAQ [15]) in the form of historical stock price distributions. This chapter gives us a brief know-how of the stock market functioning and the signals used in our study. As discussed in Chapter 3, we apply the machine learning methods to stock

market data for classifying and predicting the potentially dead and active companies. Finally, we conclude this chapter with discussions of our experiments with financial data, their inferences and how this study can be helpful in real world scenarios.

- **Chapter 5 - Application to Biomedical domain**

This chapter gives a brief introduction about the human joint structure, with focus on joint related disorders such as cartilage degeneration and Osteoarthritis. We then review the various computer aided diagnostic methods (invasive and non-invasive) applied for detecting and treating joint related diseases. Some of the important terms and concepts, related to Osteoarthritis, have also been discussed in this chapter. Finally we discuss the application of machine learning methods discussed in Chapter 3, and results obtained from our experiments with knee signals classified and predicted as abnormal or normal joints.

- **Chapter 6 - Discussions, Conclusions and Future works**

This chapter contains all the discussions from our experimental work, the conclusions and the directions for future work. We discuss, how our research work can be applied to real world scenarios, thereby proving useful not only on an individual level but also on a provincial or national administrative level.

Chapter 2

Literature survey, Motivation and Problem definition

2.1 Literature survey

AS mentioned in Section 1.2, we are assessing the future state of a complex system by analyzing the real-time signals emerging from this system. Once such complex system that we have taken into account is the *stock market*.

The stock market, be it anywhere in this world always displays a random nature with regards to the continuous fluctuations in the stock prices. Such a random behavior is not just affected by the influx-deflux of trading in the market, but also by certain important external factors such as political events, change in government policies and decisions, market emotions such as rumors on the trading floor, and by the usual factor of gambling. As a consequence of such mannerisms, it becomes nearly impossible to predict the future stock price of a company in the following day, week, month or year.

Taleb [16] has discussed in his book "The Black Swan", the challenge of making accurate and useful predictions with a great deal of contempt to finance academics and economists. According to Taleb [16], the Black Swan refers to a large-impact unpredictable event, which is beyond the capacity of normal expectations. Taleb [16] contended that the amount of uncertainty that underlies complex systems, such as the stock market, is so high that one should not attempt to predict the future dynamics of these systems. Guergachi and Boskovic

[17] argued that traditional state space modelling techniques, which aim to capture the laws governing the behavior of such systems into a mathematical model, expressing the system state $s(t)$ as a function of time, do not work when system uncertainties involve heavy-tailed probability distributions. Alternatively, they suggested to replace the traditional ‘system model’ instrument with ‘learning machines’.

Financial prediction methods are usually categorized into two:

- Fundamental Analysis
- Technical Analysis

The *fundamental analysis* of a company involves the analysis of its financial statements, its managerial position, its competitors and its overall market strength in its own domain. Usually financial analysis is applied by trade experts in forecasting the company’s future prospects. It is usually done for estimating future stock prices, making business growth projections, evaluating managerial decisions, calculating the company’s credit risk, etc. When it comes to predicting stock prices, fundamental analysis follows the rule that - the stock exchange may mis-price a security for a short period of time, but the correct stock price will be eventually reached, thus generating profits for the investor. Fundamental analysis can also be used for managing investors’ portfolios, evaluating good and bad companies, determining stock growth rates, etc.

Technical analysis primarily deals with forecasting future stock prices and company’s growth, by using historical market data such as stock prices, income statements, financial ratios, etc. Technical analysts try to identify patterns and trends from the businesses in the stock market, and attempt to mathematically model these trends using certain hard and soft assumptions. It is different from fundamental analysis, with the fact that technical analysis doesn’t consider factors such as the type of company, the stock market it trades in, etc. and rather, focuses on the stock price charts and volume information. By contrast, fundamental analysis always considers the company’s nature, its market, etc. while forecasting its performance.

While applications of machine learning to anticipate mergers and acquisitions remain limited, bankruptcy predictions have been studied extensively through the use of various machine learning approaches, including neural networks, genetic algorithms and combination thereof [18, 19, 20], support vector machines [21, 22, 23, 24, 20, 25], self-organizing maps [26] and other machine learning models [27, 28, 29]. However, these approaches tend to make little or almost no use of stock prices. Instead they use other datasets such as financial ratios, income statements, etc.

These financial ratios are advantageous to use considering the fact that they provide a great deal of information about a particular company that may help with predicting a possible bankruptcy. But these financial ratios are not easily available, and hence assessing the dynamics of various economic sectors at large cannot be conveniently automated, which goes against the major purpose of this research. On the other hand, stock prices are easily obtained and can be retrieved in real-time fashion and automatically from various databases using a variety of Internet-based software services. Therefore, using stock prices as the major performance indicators for monitoring the overall state of companies, and making decisions about the corresponding sectors makes it possible to automate the task of assessing and anticipating the dynamics of economic sectors.

Using stock prices of a company for monitoring its performance has its significance in an interesting way. The change in stock price is correlated to the corresponding changes in earnings and dividends of the company. Higher stock prices lead to more profitability, and hence higher returns. An increase in profitability indicates higher returns from the market, thus increasing the earnings of the company and hence the dividends flowing out to the shareholders. Higher earnings would definitely increase the current assets of the company thereby allowing room for covering the short-term and long-term liabilities. The lower the liabilities, the lesser the chance of bankruptcy or total collapse of the company. In addition to being significant indicators of the company's performance, stock prices also help in assessing the behavior of the economic sector in which the company is listed. For more details on the functioning of the stock market refer to [30]. As mentioned, stock prices are affected by

many factors which become inherently a part of the weekly closing stock price.

2.2 Motivation and Problem definition

The main motivation behind this research work was that, although extensive studies have been done for studying the complex system behavior, these works are mainly limited to performance measurements or estimating future values of real-time signals emerging from these systems. Classifying and predicting the future states (such as survival or collapse of companies in the stock market) of such systems is helpful from a bigger perspective. For example, classifying and predicting potentially active and dead companies in the stock market helps in monitoring the dynamics of an economic sector, not only from an independent investor's perspective, but also from a provincial or national economy perspective.

This research work is not targeted at predicting the future value of a stock price S_t as a function of the time t . Instead, it looks at re-framing the prediction problem into another problem that is at *the same time* simple enough to be, theoretically at least, solvable (even under heavy-tailed circumstances) *and* useful for the users who are concerned by the problem at hand. More specifically, our study proposes to assess the fate (e.g. survival or collapse) of a company based on its historical financial time-series taken over a certain period of time. It thus focuses on addressing the binary classification question of whether a company will likely remain in the stock market ('survival' class), or not, i.e. the company will be either bankrupt, acquired or merged with another one ('collapse' class). For a discussion as to why classification problems can be considered as theoretically solvable even in the case of heavy-tailed distribution, the reader is referred to [31] and [17]. Also, explanations as to why the binary classification problem of *survival* versus *collapse* can be useful for the stock market analysis are available in [32].

In more specific terms, the research presented herein aims at developing computer-based tools to monitor and assess, in an *automated fashion*, the dynamics of the economic sectors in a certain country or internationally by performing a discrimination of the companies that are likely to survive and remain active from the companies that will likely disappear from the

stock market. This discrimination is carried out using Fisher kernels to reduce the financial time-series into a three-dimensional plot of the Fisher scores. These Fisher scores are used for training and testing support vector machines (SVM) for classification and prediction of the potential survivors. To develop the concepts and test them, our research has been focussed on the Pharmaceuticals and bio-technology sector, but once the methodology is proven to work, the research is to be expanded to other sectors to help develop industrial strategies and plan for restructuring of economies internationally.

Such kind of experiments can be conveniently applied to real world scenarios and do not restrict themselves to just scientific research. These applications can prove to be helpful in addressing some complex issues, which otherwise cannot be handled by using just direct programming methods. The extensive amount of datasets used in this study give more insights into the functioning of complex systems, which were otherwise assumed to be black-box systems. Continuous research work, and constant updating of data and algorithms are the key aspects that have helped in achieving desired results.

The proposed method in this research has been extended to the Biomedical domain, wherein we analyze the knee signals for assessing cartilage degeneration severity and perform further classification between abnormal and normal knee joints. The intelligent techniques have been used to monitor biomedical signals as well, and the related literature review can be found in Chapter 5.

The following chapter describes our proposed method for analyzing the future states of certain complex systems, and generate classification patterns by assessing the real-time signal behavior emerging from these systems.

Chapter 3

Background theory and Methodology

3.1 Introduction to Machine Learning

MACHINE Learning is a subset of Artificial Intelligence, wherein we use intelligent algorithms, which can be self-trained to recognize, classify and perform regression, on a set of input patterns, and map them to a given output or some category labels. Such types of learning machines are trained to learn from experience, and generate probabilistic outputs or mapping patterns. Continuous research in this field is being done in order to solve complex theories and problems such as speech recognition [33], DNA sequence identification [1], stock market forecasts [20] and many more. Applications of machine learning techniques can be generalized as follows [8, 9]:

- Complex problems where human expertise remains insufficient.
- Scenarios where the system behavior changes rapidly and randomly, such as stock market, climate.

Extensive use of computers for solving complex problems however, is not always easy, and requires the use of certain domain specific examples or *training data*, which would give some information about the mathematical models functioning behind the behavior of the complex systems. Learning from examples constitutes the theory of Machine Learning. The underlying function used for mapping the inputs to the correct outputs is known as the *target function*. The estimate of this target function or the output of the learning algorithm

is known as the *solution*. When we apply the *learning methodology* for pattern classification, as has been applied in this study, then the target function is referred to as the *decision function*. The most optimum decision function needs to be chosen from a set of candidate functions, also known as *hypotheses*. The algorithm which uses the set of examples as training data, and maps it to the appropriate outputs using one of the hypotheses functions, is known as the *Learning algorithm*. Two major application domains of machine learning are:

- Pattern Classification
- Regression Analysis

These two applications can be further divided or elaborated, depending on the specific applications. In order to illustrate the complexity of certain systems, consider an example of a fish packaging factory, where packaging of fish takes place in an assembly line [1]. Assume that we have to pack two kinds of fish: Salmon and Sea-bass. The question here arises is that: How do we separate the salmon from sea-bass fish while packaging? We employ some sort of scanner, fed with an algorithm on how to classify the two products. Some particular features of the fish such as length, width or weight are identified before the training of learning machine begins. These features are taken as input parameters for classifying the fish. Based on these features, the scanner is first trained using a training dataset; its performance is then tested and validated. While training the machine, one might encounter certain external disturbances such as white noise, disturbances on the assembly line, improper placement of fish on the conveyor belt or even noise coming from the manufacturing unit.

These external disturbances need to be handled carefully while training the machine, in order to achieve desired results. Only after vigorous training and testing, we can say that the scanner's learning algorithm is strong enough to learn from the actual data. So the main aim here is to classify the input patterns, given a category label, and map each of the inputs to the correct label. The classification is mainly achieved by separating the classes using a boundary. This boundary may be of the form of a straight line, a definable curve or some arbitrary curve, depending on the complexity of the dimensions of the space

in which classification takes place. This is evident from the illustrations shown in Figure 3.1, classifying two kinds of fish : sea-bass & salmon [1]. In this example, for classifying the fish, two features, the width & length have been considered by the classifying algorithm.

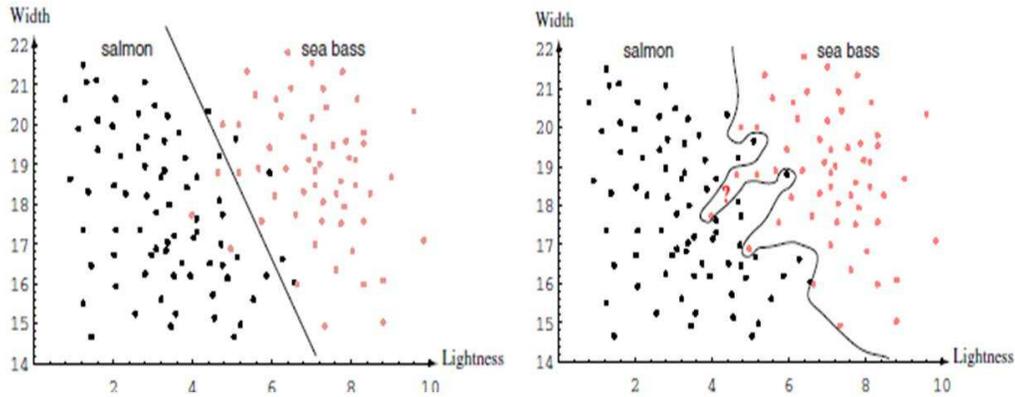


Figure 3.1: Classification between Salmon and Sea-bass [1]

Following sections describe the kernel-based machine learning methods employed for modelling complex systems. More specifically, we emphasize on their application to the Financial and Biomedical domains. Our research work is based on the application of kernel-based machine learning methods, and the following sections would provide a base for our proposed study and then proceed further with the description of our application model.

3.2 Kernel-based Machine learning

Detecting linear relationships in the input data is a desired approach in machine learning techniques, so that appropriate and suitable features can be extracted from the data for the learning algorithm. This however is not possible for complex datasets involving random variations in feature values. Applying machine learning methods in the input space, for pattern classification sometimes produces non-linear separation of classes. One solution to address this problem would be the application of kernels to input data, which would essentially transform the input to a higher dimensional space, wherein the probability of

linearly separating the classes is higher. In kernel methods, there is a strong possibility that the higher dimensional space may be non-linear in nature as opposed to the linear input space; and the separation of classes would be linear in feature space and non-linear in input space. As we proceed further, we will observe how kernel methods effectively transform input data for better classification. Any kernel-based machine learning method employed for classification, includes two major steps [8, 9]:

- Mapping the non-linearly separable input data from its current lower dimensional space (*input space*), to a linearly separable data in the (*feature space*), and,
- Application of a learning algorithm which is written for finding patterns and classifying in the feature space.

The use of kernels would become clear from the illustration shown in Figure 3.2.

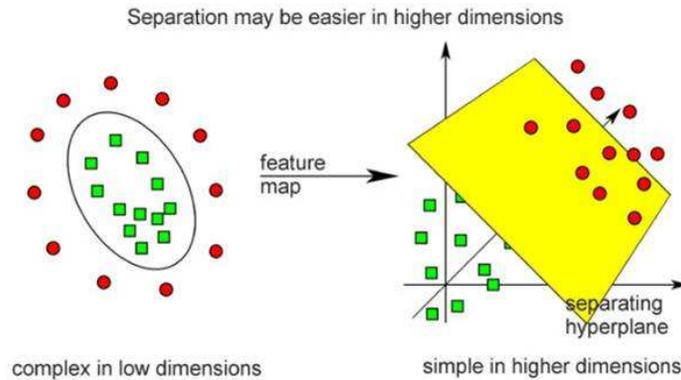


Figure 3.2: Dimension Transformation

In order to train a classifier, we assume the initial input data set as the training set. Using this set, and the learning algorithm given, the system learns to classify the data accordingly. A kernel function can be represented as a function that computes the inner dot product of the training sample. Assume that the input data is of the form,

$$X = x_1, x_2, x_3, \dots, x_n \quad (3.1)$$

Then, using a function ϕ , we can map the input data set X into a higher dimensional space as,

$$X = x_1, x_2, x_3, \dots, x_n \rightarrow \phi(X) = \phi_1(X), \dots, \phi_n(X) \quad (3.2)$$

This is essentially same as mapping the input space X into a new space F . The space F is known as the Feature Space. It also takes into consideration the input features introduced in order to classify the data to the correct category label. Additionally, some attributes of the input quantities can also be considered in the feature space for classification. Figure 3.3 shows the transformation of the data, from input space to the feature space.

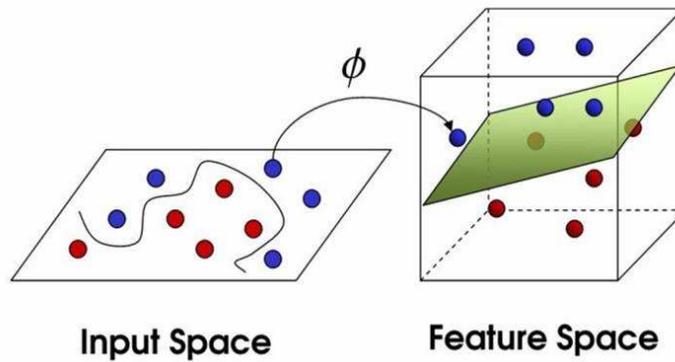


Figure 3.3: Transformation in to Feature Space [2]

Applying kernels to pattern classification can also lead to dimensionality reduction of the input data set, in the feature space. This is done by selecting the smallest number of features out of a given number of attributes, large enough to classify the input data and give the essential information contained in the attributes. Increasing the number of features can lead to poor performance & generalization of the classifier. That is, if we have a large set of redundant features, then there are more chances that the function to be learned, cannot be represented using a standardized learning machine. So basically, a good feature selection involves - dimensionality reduction, and, removal of irrelevant features. Now, let us assume that the classifier, in the feature space is governed by the following hypotheses for pattern

classification.

$$f(X) = \sum_{i=1}^N w_i \phi_i(X) + b \quad (3.3)$$

where, w_i is the weight associated with each of the input data points; and b is the bias of the system. $\phi : X \rightarrow F$ is the non-linear mapping from the input space to the feature space. Thus we can say that Linear Learning machines give us a dual representation for the inputs. So in the feature space, the hypotheses can be given by the relation,

$$f(X) = \sum_{i=1}^d \alpha_i y_i < \phi(x) \cdot \phi(z) > + b \quad (3.4)$$

where, α_i is the transformed weight vector in the feature space; y_i is the output corresponding to each input point and d is the number of training points. The kernel function K can be defined as $K(x, z) = < \phi(x) \cdot \phi(z) >$ for all $x, z \in X$. Thus we can use the kernel function to implicitly map the data into the feature space by finding the inner dot products between two consecutive transformed input samples, thereby ignoring the computational difficulties in finding the feature map. The kernel matrix thus formed, may be given as,

$$\mathbf{K} = (< x_i, x_j >)_{i,j=1}^d \quad (3.5)$$

Figure 3.4 summarizes the various stages involved in the implementation of a kernel-based machine learning method.

Kernel functions may be of different forms, such as Linear kernels, Polynomial kernels etc. One of the most popular forms of kernels is the Fisher kernels which were introduced in 1999 [34]. Section 3.2.1 highlights the functionality of Fisher kernels and how they were applied in our study.

3.2.1 Fisher Kernels

Fisher kernels are known for their wide usage in generative and discriminative pattern classification, and recognition approaches [35, 36, 34, 37]. Generative models are those which are used for randomly generating observable data; whereas Discriminative models are used in machine learning for assessing the dependency of an unobserved random variable on an

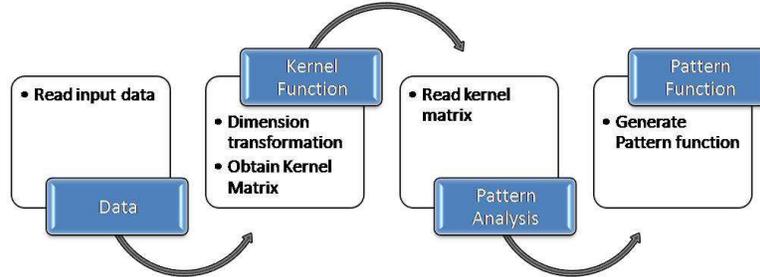


Figure 3.4: Application of Kernel Methods

observed variable using a conditional probability distribution function. They extract more information from a single generative model and not just its output probability [9]. The features obtained after applying Fisher kernels are known as *Fisher scores*. We analyze how these scores depend on the probabilistic model, and how they give us the information about the internal representation of the data items within the model. Fisher kernels have been applied in various domains such as speech recognition [33, 38, 39], protein homology detection [36], web audio classification [40], object recognition [41], image decoding [42, 43], etc.. For detailed information on Fisher kernels, refer to [9].

Fisher kernels combine the properties of generative models such as Hidden Markov model, and discriminative methods such as support vector machines [34]. The use of Fisher kernels for dimension transformation has two advantages [34, 9]:

1. Fisher kernels can deal with variable-length time series data.
2. Discriminative methods such as SVMs, when used with Fisher kernels can yield better results.

3.2.2 Support vector machines (SVMs)

SVMs constitute a unique machine learning method, developed by Vapnik *et.al.* [31]. The conventional learning machines were generally linear in nature, and did not cover the aspects of non-linearity. SVMs negate this factor and take non-linear input spaces into consideration, for classification and regression, at the same time keeping intact the basic theories of kernel induced feature spaces and the corresponding optimization problems. The optimization problem faced in case of simplifying the input space and mapping it to an appropriate output space is usually Lagrangian in nature. SVMs have gained tremendous interest in recent years, due to their powerful feature retaining capability of the complex systems and hence their good generalization performance. Applications include handwriting recognition, face detection, pedestrian detection, text categorization, speech recognition, stock market forecasts, Bioinformatics, etc.

In this study, we are dealing with the binary classification of time series signals, sorting them into two classes. Usually, whenever we do binary classification, we tend to separate the two categories of data using a hyperplane, such that the distance of this hyperplane from either sides is maximum. Hence, it is also known as Maximum Margin Hyperplane. In case of linearly separable categories, we observe that the two classes are clearly separated into two regions by the hyperplane. Whereas, in the non-linear case, it's seen that, some of the data from either sides, gets mixed up.

These data points are also known as slack variables, and the basic aim is to re-create the hyperplane such that maximum classification is achieved easily, along with the least number of slack components. The slack variables are otherwise known as Misclassified points. The above explanation becomes clear from the illustrations shown in Figures 3.5 and 3.6.

Let us assume that we have N data points representing two classes - $C1$ and $C2$. Now, assuming that we get a linear separation case i.e. the classification is easily done using a simple hyperplane, we get a scattered point distribution of the two classes. The mathematics used in this study with respect to classifying $C1$ and $C2$ has been adopted from [20, 8].

Let the data points be denoted by the set $\{x_i, y_i\}$ for $i : 1 \dots N$ and $x_i \in R^k$ (Real

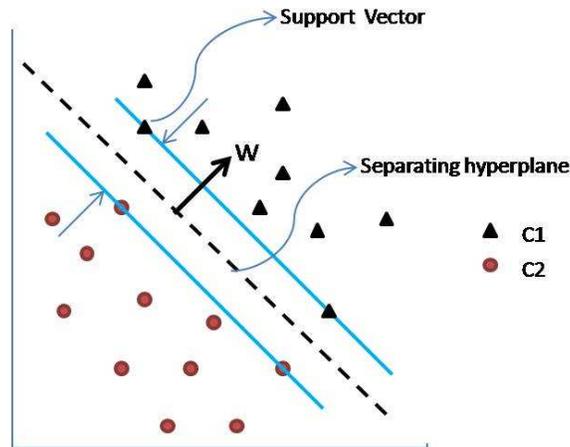


Figure 3.5: Linearly Separable case

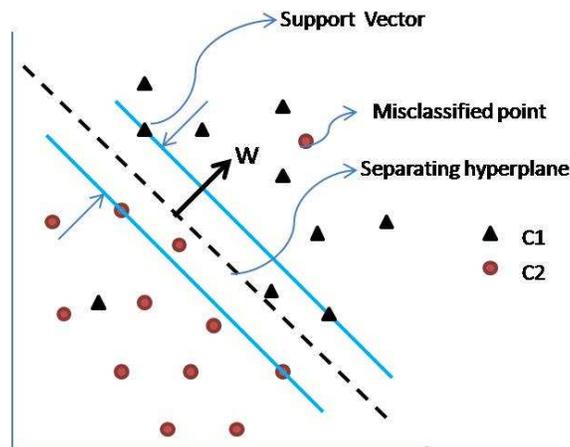


Figure 3.6: Non-linearly separable case

Number). The separating hyperplane is given by the equation,

$$f(x) = w \cdot x - b \quad (3.6)$$

where, w is the weight vector perpendicular to the separating hyperplane. In order to classify the categories as C1 and C2, we draw two hyperplanes (Figures 3.5, 3.6) on either side of the separating hyperplane, such that some of the data points lie on these hyperplanes. Then

the equations of the two hyperplanes would be:

$$HP_{C1} = y = w.x - b = +1 \quad (3.7)$$

$$HP_{C2} = y = w.x - b = -1 \quad (3.8)$$

Here we also assume that the distance between HP_{C1} and HP_{C2} is maximized. The coefficients weight vector w has been normalized in this case with respect to the separating hyperplane, HP_{C1} and HP_{C2} .

We now aim at maximizing the distance between HP_{C1} and HP_{C2} . During this process, we take into consideration the data points lying on HP_{C1} and HP_{C2} . These data points are known as Support Vectors, since they help in classifying of data. The other data points may be neglected for now, as long as they don't fall between, HP_{C1} and HP_{C2} .

Assume that we have a point on the plane HP_{C1} . Then the distance of this point from the separating hyperplane, will be given by the relation $\frac{|w.x-b|}{\|w\|} = \frac{1}{\|w\|}$

This implies that the distance between, HP_{C1} and HP_{C2} would be $\frac{2}{\|w\|}$. It becomes clear from $\frac{2}{\|w\|}$ that the distance can be maximized, if we minimize $\|w\| = w^T.w$, along with the condition that no data points exist between HP_{C1} and HP_{C2} , and hence $y_i = +1$ for C1 category, and $y_i = -1$ for C2 category.

We can now summarize the optimization problem as:

$$\min_{w,b} \frac{1}{2} w^T w \quad \text{subject to } y_i (w.x - b) \geq 1 \quad (3.9)$$

In order to simplify, we can introduce Lagrange multipliers such that,

$$D(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^N \alpha_i y_i (w.x_i - b) + \sum_{i=1}^N \alpha_i \quad (3.10)$$

Here α is the transformation of vector w into the higher dimensions. The above concept has been explained from a theoretical point of view. In order to consider the real data, we have to use the non-linear case, wherein we might get some data points between HP_{C1} and HP_{C2} hyperplanes. In this situation, we introduce the concept of Kernels. If we want to linearly

separate the data points, we need to transform the data points to a different dimension, using a function $\mu(x)$. Let us define a kernel function $k(x_i, x_j) = \mu(x_i) \cdot \mu(x_j)$ which represents the high dimensional input space. This way, we need to consider only the form for $k(x_i, x_j)$, and not bother about $\mu(x)$. Thus we can conclude from the works of Vapnik *et.al.* [31] that, while training a SVM for pattern classification problem, we get a quadratic optimization function using Lagrange multiplier [8],

$$\text{Minimize : } E(\alpha) = - \sum_{i=1}^N \alpha_i + \frac{1}{2} \sum_{i,j=1}^N y_i y_j \alpha_i \alpha_j k(x_i, x_j) \quad (3.11)$$

$$\text{subject to : } \sum_{i=1}^N y_i \alpha_i, \forall i : 0 \leq \alpha_i \leq C, \text{ where } C \text{ is a constant.}$$

The theory described so far, was just a basic touch to the know-how of support vector machines. For a detailed information on SVMs and Fisher kernels refer to [44, 8, 9, 1]. Detailed information on Lagrangian optimization can be found in [8]. For the time-series signal study, we have taken the Gaussian radial basis function as our kernel functions, and applied the Sequential Minimal Optimization method (SMO) [45, 44] for finding the separating hyperplane. When we use a Gaussian rbf as our kernel function, then the maximum margin classifiers in the feature space are well regularized, i.e. we do not face the issues with overfitting.

3.3 Proposed Work

In this study, we propose to use the Fisher kernels for analyzing the gradient (Fisher score derivatives) of a high dimensional feature spaced input signal coming from a complex system (financial or biomedical time series), using a recursive probabilistic model (applied to all input vectors in the input set), for assessing the fate of the system. Fisher kernels combined with the learning ability of SVMs make good classifiers in a binary context [35] since they retain the essential features of the input dataset for a good classification. In our analysis we have taken into consideration a Gaussian probabilistic model, for creating the Fisher kernel.

3.3.1 Applying Fisher Kernels to a time-series model

In this section, we study the application of Fisher kernels for identifying patterns emerging from a time-series model (section 1.2). We use a generative Gaussian mixture model [1, 9, 8] for our complex system. In our study, since we are performing binary classification, we have two Gaussian components for each time series signal. These two Gaussian components for each signal help us in assessing the tendency of the signal to be categorized in either of the two classes. Let us make the following assumptions before we mathematically generate the Fisher score model. The methodology in this section has been adopted from [46].

- The complex system to be modelled is a black box out of which a time series signal is emerging.
- In order to make the data distribution in the signal to be *i.i.d* (independent and identical distribution), we apply some mathematical relations such as log-normal relations or normalizing the dataset. By doing so, we generate another set of data which is nearly an *i.i.d* distribution, thus giving us a transformed time series signal.
- Upon applying the previous assumptions, we then generate the Fisher score model for the complex system as explained further.

Before we proceed with deriving the model and its equations, we assume the following variables and their meanings as shown in Table 3.1.

1. We use the assumptions mentioned in the beginning of this section for deriving the Fisher scores.
2. Our study is on binary pattern classification, and in order to achieve this we use the Fisher scores generated as described further in this section, for plotting and visualizing between the two categories (e.g. active and dead companies, or abnormal and normal knee joints).

Table 3.1: List of variables used for Fisher scores' computation

Symbol	Indication
$X=x_1, \dots, x_N$	The input dataset, which consists of a number of vectors or time-series signals
x	An input vector or a time-series signal
N	Number of input vectors in the given input dataset
N_d	Dimensionality of the input vector
N_c	Number of samples available in an input vector x_i . This value can be same for all input vectors within X or can be of variable length.
N_g	Number of Gaussian components for clustering or pattern visualization = 2
S_c	i^{th} sample value in an input vector ($i = 1 \dots N_c$)
r_c	Normalized value for the i^{th} sample
$\theta=[a_j, \mu_j, \sigma_j]$	Gaussian estimates for the N_g components, with a_j being the weight vector, μ_j being the mean vector and σ_j being the variance vector.
$R(i, j)$	Gaussian mixture model for the i^{th} week's returns, built using $j=2$ Gaussian components
$P(r_i \theta)$	Probability density function for the i^{th} normalized sample value
$P(C \theta)$	Probability density function for the entire input vector or time-series signal

3. The length of each input vector (or the number of samples available) can be variable, or all the input vectors can be of same length. We assume each input vector to be a unique time series signal.
4. Our study based on binary pattern classification has been applied to two domains : the financial domain wherein we classify between the potentially dead and active companies; and the biomedical domain wherein we classify between abnormal and normal knee joints for assessing the risk of cartilage degeneration. These two domains have been explained thoroughly in Chapters 3 and 4. In the financial study, we intend to find the log-normal stock returns using the weekly stock prices; whereas in the biomedical domain we normalize the knee angle signals between the interval $0 \dots 1$. These stock returns or normalized knee angles are taken as our r_c values. We do this so as to make the distributions *i.i.d* in nature, as per the assumptions mentioned in the beginning of this section.
5. We first find the initial values of the Gaussian estimates, $\theta = [a_j, \mu_j, \sigma_j]$ using the Expectation Maximization algorithm [47, 48] for $j = 1 \dots N_g$. The expectation max-

imization algorithm is used for estimating the likelihood parameters of certain probabilistic models.

6. Using these estimates we create the Gaussian mixture model M so that $R(i, j)$ is an $N_c \times 2$ matrix.

$$R(i, j) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{r_i - \mu_j}{\sigma_j}\right)^2\right) \text{ for } \begin{cases} i = 1 \dots N_c \\ j = 1 \dots N_g \end{cases} \quad (3.12)$$

7. The diagonal covariance GMM likelihood is then given by the Probability density function for the i^{th} normalized sample value. Thus, $P(r_i | M, \theta)$ is a $N_c \times 1$ matrix.

$$P(r_i | M, \theta) = \sum_{j=1}^{N_g} a_j R(i, j) \quad (3.13)$$

8. The global log-likelihood of an input vector's normalized values $C = \{r_1, r_2, \dots, r_{N_c}\}$ is given using the probability density function as follows. Therefore $\log P(C|M, \theta)$ is a single value for each input time series signal.

$$\log P(C | M, \theta) = \sum_{i=1}^{N_c} \log P(r_i | M, \theta) \quad (3.14)$$

9. The Fisher score vector is composed of derivatives with respect to each parameter in $\theta = [a_j, \mu_j, \sigma_j]$. The derivatives with respect to the j^{th} prior, mean and variance are given as follows.

$$\frac{d}{da_{j^*}} \log P(C | M, \theta) = \sum_{i=1}^{N_c} \frac{R(i, j^*)}{\sum_{j=1}^{N_g} a_j R(i, j)} \quad (3.15)$$

$$\frac{d}{d\mu_{j^*}} \log P(C | M, \theta) = \sum_{i=1}^{N_c} \frac{a_j R(i, j^*)}{\sum_{j=1}^{N_g} a_j R(i, j)} \cdot \frac{1}{\sigma_{j^*}} \left(\frac{r_i - \mu_{j^*}}{\sigma_{j^*}}\right) \quad (3.16)$$

$$\frac{d}{d\sigma_{j^*}} \log P(C | M, \theta) = \sum_{i=1}^{N_c} \frac{a_j R(i, j^*)}{\sum_{j=1}^{N_g} a_j R(i, j)} \cdot \left(\frac{(r_i - \mu_{j^*})^2}{(\sigma_{j^*})^3} - \frac{1}{\sigma_{j^*}}\right) \quad (3.17)$$

for $j^* = 1 \dots \dots N_g$

10. Each of the derivatives comprise of two components, thus giving us a 1 x 2 matrix for each derivative.

$$\frac{d}{da_j} = \left(\frac{d}{da_1}, \frac{d}{da_2} \right) \quad (3.18)$$

$$\frac{d}{d\mu_j} = \left(\frac{d}{d\mu_1}, \frac{d}{d\mu_2} \right) \quad (3.19)$$

$$\frac{d}{d\sigma_j} = \left(\frac{d}{d\sigma_1}, \frac{d}{d\sigma_2} \right) \quad (3.20)$$

11. The likelihood Fisher score vector for each signal is thus given as follows.

$$\Psi_{fisher}(C) = \left[\frac{d}{da_j}, \dots, \frac{d}{d\mu_j}, \dots, \frac{d}{d\sigma_j} \right]^T \log P(C | M, \theta) \quad (3.21)$$

for $j^* = 1 \dots \dots N_g$

12. Thus, for each input vector we get a 6 x 1 Fisher score matrix. In order to plot the scores, we then add up each pair of Fisher Scores (with respect to weights, mean and variance), to get a three-dimensional scatter plot.
13. The Fisher scores obtained for each input vector are then further used as input data for training and testing our SVM model. The SVM model basically performs binary classification and predictions in our scenario, using which we can infer statements about the future state of the complex system taken into consideration.

14. The correctness of the classification performed by SVM, is further verified when we apply the same set of Fisher score data for Linear Discriminant Analysis (LDA). As a note, our study is not intended to analyze the performance of LDA approach, or compare it with SVM. The reader may refer to Section 8.3 for detailed information.

The following chapters will basically give us a domain specific knowledge of the complex systems selected for our study, and the experiments conducted on the real-time signals emerging from these systems. More information on Machine Learning can be found in Appendix 7. Details on Expectation-Maximization algorithm, SMO method, LDA and Cross-Validation can be found in Appendix 8.

Chapter 4

Application to Financial domain - Stock Markets

4.1 About Stock Markets

A *STOCK EXCHANGE* is an organization which hosts a market where stocks, bonds, options and futures, and commodities can be traded. The buyers and sellers come together to trade during specific hours on business days. In financial terms, *stock* simply means a supply of money, the company raises from individuals or other organizations. That is, when you buy stocks of a company, you become the owner of a part of that company. This ownership is called *share* and is usually measured in percentage with respect to the number of shares owned by an individual, and the total number of shares made available by the company to be sold in the market. Individuals or firms who own shares are referred to as *shareholders* or *stockholders*.

When a company wants to sell its shares, it lists its stock on an exchange. The stock exchanges are set up to make it faster and easier for people to buy and sell company stock. When a company is traded on an exchange, it is referred to as *Listed* or *Active*. In order to trade, firms and brokers in the stock exchange need to follow certain rules and regulations such as the minimum number of shareholders, a minimum price per share, and a minimum market cap. If at any time a company fails to meet these regulations, it is *Delisted* or declared *Dead* by the stock exchange, and is not allowed to trade its shares in the market.

When a company gets delisted, it indicates a serious financial or managerial trouble, and may cause the stock prices to fall.

When individuals want to buy or sell shares, they can simply call a stock broker, which is a firm authorized to trade at stock exchanges. The stock broker relays the trade message to the floor of the correct exchange, and a representative of the company then completes the trade request. A stock broker receives a commission for providing this trading service. However, it is becoming increasingly popular for people to use online trading sites instead of stock brokers.

Stockholders always hope that the companies they invest in are profitable businesses, because they will then receive a share of the profits as well. However if the company ends up in a loss, the shareholders would lose a significant portion of their invested money as well. Shareholders usually have voting rights, typically one vote for every share they own. Many companies have yearly meetings where the shareholders can vote on company issues. Stockholders also receive annual or quarterly reports that let them know how the company is doing financially. Such reports are mostly based on the company's internal income statements, etc. When an investor thinks that the stock market is going to go down, i.e. the stock prices will fall, he is then known as a *Bearish investor*. Such investors buy stock very cautiously. Similarly, individuals are called *bullish* when they believe that the stock market will go up, thus making them invest more money into the stock market. Likewise, if the prices of stocks as a group tend to rise, the stock market is called a *bull market*. If stock prices as a group tend to fall, however, the stock market is referred to as a *bear market*.

The stock market is one of the most important sources for companies to raise money by allowing businesses to be traded publicly, thus generating liquid funds for the companies. These liquid assets can be helpful in clearing the company's liabilities and hence secure its businesses. The liquidity feature is also important from the view that, it makes the trading of company stocks more attractive than other investments such as real estate or bank deposits, since the investor can generate significant profits.

It has generally been observed that the stock prices and the stock market in general, prove

to be good indicators of the the dynamics of an economy, and can influence the overall social mood of the public. The better the stock market performance, the better the economy. In fact, the stock market is often considered to be a primary indicator of an economy's growth. Rising share prices generally indicate that business investment is increasing, and vice versa.

4.2 Application of Fisher kernels and SVMs to financial time-series

Till date many machine learning approaches have been applied to process stock market data for various purposes (see e.g., [49, 50, 51, 52]). In particular, the SVM approach, which is one of the main tools used in this study, has been applied quite extensively for analyzing financial data, and some of these applications have reported high accuracy rates such as in [51, 52]. In this research, the application of SVM is combined with the implementation of a kernel whose use has not been widespread for processing time series: Fisher kernel. More specifically, the approach used in this study to perform binary classification of companies is implemented in two major steps:

- Dimension transformation of the financial time-series using Fisher kernels, into the feature space; and,
- Application of SVM to the feature space data, for predicting the potentially active and dead companies. The implementation of SVM in this space uses the radial basis function as a kernel.

In this study, we are dealing with a binary classification of business enterprises, sorting them into 'collapse' and 'survival' categories. Usually, whenever we do a binary classification, we tend to separate the two categories of data using a hyperplane. In this study, the classification involves visual inspection of data points in the Fisher scores space, which is obtained by applying the Fisher kernel method; and further learning of the classification pattern is done using support vector machines. Figure 4.1 illustrates the step-wise procedure for classification.

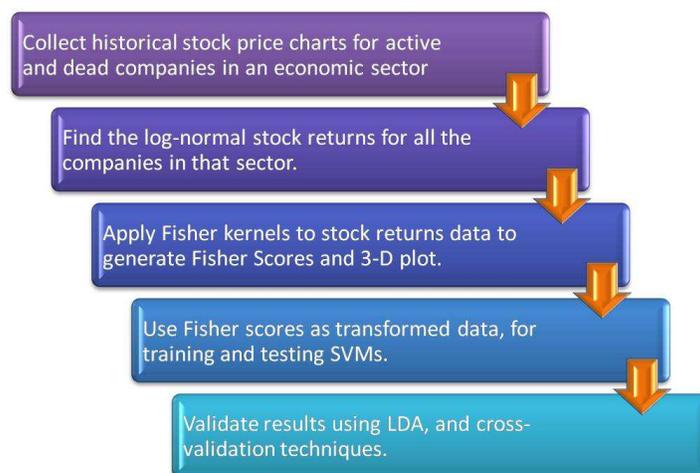


Figure 4.1: Methodology for classifying companies

Using this methodology we divide our implementation into various stages of data collection, processing and outcomes, in the following sections.

4.2.1 Data collection and Processing

Before we initiate the task of developing a comprehensive software system that monitors, assesses and anticipates the dynamics of a nation's economy and its sectors, there is a need to design and test the core methodology and algorithms that will underlie the functioning of the software system's engine - addressing such a need is what this thesis is all about. We have thus analyzed the random behavior of changes in weekly stock prices for different companies in one economic sector: the Pharmaceuticals and bio-technology sector. The intention of this analysis is to evaluate the performance of Fisher kernels and support vector machines in discriminating the companies that will likely survive from the other ones.

The stock price behavior exhibited by the companies is also dependent on the sector taken into consideration. So our experiments can also be extrapolated to monitoring the sector dynamics along with the assessment of the potentially surviving or collapsing firms in a particular sector. When we consider a common time frame, not all companies have the same amount of data. In a previous study we had taken the data such that we get

maximum number of firms along with maximum amount of weekly price data [53]. This was basically a selection of a fixed length financial time-series. In this research we deal with variable length stock data for each company in a common time frame and process it likewise in three dimensions. The stock price distribution length for each company is variable in this case. The financial time-series data is first processed and transformed using Fisher kernels into three dimensions. The three-dimensional scatter plot provides visualization between the active and dead companies, which would further aid in inferring the future states of active companies. The set of all the Fisher scores for each sector is then applied for classification and prediction using support vector machines.

In this study we have considered the companies falling under the Pharmaceuticals and Biotechnology sectors listed in the TSX [14], NYSE [13], TSX-Ventures [14] and NASDAQ [15] stock exchanges. We collected the weekly stock price data for various companies from a common time frame of January 1950 to December 2008. The data is collected using the following procedure on Thomson Datastream software. This software is one of the world's largest and most trusted sources for financial statistical data. It provides historical information on key datasets which include equities, investment trusts, warrants, bonds, exchange rates, interest rates, commodities and other macroeconomic data taken from worldwide financial institutions and agencies. Datastream also provides historical stock price data in the form of a time-series, and may cover more than 50 years of data. Available stock price frequencies include daily, weekly, monthly, quarterly and yearly. Although a multi-resolution analysis has not been done in our experiments, but such study can be used as a direction for future research. Thomson Datastream defines an *active* company as one which is currently listed in the exchange and is trading, whereas, a *dead* company as one which has been delisted from the exchange since it does not perform any trading. With regards to these definitions, we are classifying between active and dead companies, in a particular economic sector. Table 4.1 mentions the typical list of criteria to be set while retrieving a search in the Datastream.

In order to collect our data under the Pharmaceuticals and Biotechnology sector, we gave

Table 4.1: Typical search criteria in Thomson Datastream

SEARCH FIELD	INDICATION
Name	The name of a particular company (if known)
Market	Country in which the company's business is concentrated
Base Date	The date on which the started trading in the market (can be an approximation)
Currency	Currency in which the user wants the stock prices to be shown (can be adjusted)
Data Category	About 18 available on Datastream
Status	Active, Dead or Suspended
Instrument type	Type of financial instrument required e.g. equity, bonds, warrants, etc.
Sector	Specifies the particular industry sector, e.g. Pharmaceuticals and Biotechnology
Exchange	Name of the stock exchange
Data type	Opening, closing, highest and lowest stock prices
Data type base date	Approximate time period from which the data needs to be collected
Data frequency	Daily, weekly, monthly, quarterly, and yearly

the following inputs to the search engine in Datastream, as shown in Table 4.2.

Table 4.2: Search Criteria for Pharmaceuticals and Biotechnology sector

SEARCH FIELD	INPUTS
Market	Canada, United States
Base Data	After 01/01/50
Currency	United States Dollar
Status	Active, Dead
Instrument type	Equity
Sector	Pharmaceuticals and Biotechnology
Exchange	NASDAQ, NYSE, Toronto, TSX-Ventures
Data type	Price (Adjusted - Default)
Data type base date	After 01/01/50
Data frequency	Weekly

The data collected from Thomson Datastream, is downloaded into an integrated Microsoft

Excel workspace, as shown in Figure 4.2. The rows represent the time-line or week, and the columns represent various companies. As we can see from Figure 4.2, the names of the companies can be accessed from the top row. Also, we observe that not all companies started trading in the stock market on or before January 1950; which is the reason why we are able to see some null values in the data. In order to process this null data, we replace all the null values by zeros (0's) and then proceed with the calculation of log-normal stock returns.

Name	AASTROM BIOSCIENCES	ABBOTT LABORATORIES	ABBVIES BIOSCIENCE	ACADIA PHARMACEUTICALS	ACHILLION PHARMS.	ACORDA THERAPEUTICS	ACURA PHARMACEUTICALS	ADOLOR	AETERNA ZENTARIS (NAS)	AFFYMAX	AFFYMETRIX	AKORN
Code	@ASTM(P)	@ABI(P)	@ABV(P)	@ACAD(P)	@ACHN(P)	@ACOR(P)	@ACUR(P)	@ADLR(P)	@AEZS(P)	@AFFY(P)	@AFFX(P)	@AKRX(P)
1/2/1950	2.75	39.631	#NA	#NA	#NA	#NA	23.75	#NA	#NA	#NA	12.5312	5.5
1/9/1950	2.875	37.8194	#NA	#NA	#NA	#NA	21.875	#NA	#NA	#NA	13.9375	5.25
1/16/1950	3.25	39.514	#NA	#NA	#NA	#NA	19.6875	#NA	#NA	#NA	12.125	5.5
1/23/1950	2.875	40.4498	#NA	#NA	#NA	#NA	20.625	#NA	#NA	#NA	11.7812	5.5
1/30/1950	2.6875	36.0074	#NA	#NA	#NA	#NA	20	#NA	#NA	#NA	8.25	4.1875
2/6/1950	2.0313	37.0595	#NA	#NA	#NA	#NA	20	#NA	#NA	#NA	9.5625	4.0625
2/13/1950	2	40.040	#NA	#NA	#NA	#NA	18.75	#NA	#NA	#NA	11	4.25
2/20/1950	2.0625	41.1513	#NA	#NA	#NA	#NA	17.5	#NA	#NA	#NA	11.3125	5
2/27/1950	2.125	41.735	#NA	#NA	#NA	#NA	17.5	#NA	#NA	#NA	11.4375	4.5625
3/6/1950	2	40.3914	#NA	#NA	#NA	#NA	16.25	#NA	#NA	#NA	12.2812	4
3/13/1950	2.125	40.157	#NA	#NA	#NA	#NA	12.5	#NA	#NA	#NA	10.875	3.25
3/20/1950	2.125	41.9112	#NA	#NA	#NA	#NA	13.75	#NA	#NA	#NA	11	4.6875
3/27/1950	2.5625	42.612	#NA	#NA	#NA	#NA	17.5	#NA	#NA	#NA	12.5	4.5625
4/3/1950	2.4375	44.015	#NA	#NA	#NA	#NA	16.25	#NA	#NA	#NA	12.8125	5.125
4/10/1950	3.5625	43.547	#NA	#NA	#NA	#NA	13.125	#NA	#NA	#NA	12.125	6.3125
4/17/1950	2.625	45.243	#NA	#NA	#NA	#NA	13.75	#NA	#NA	#NA	12.7187	6
4/24/1950	2.375	46.470	#NA	#NA	#NA	#NA	13.75	#NA	#NA	#NA	13.125	5.875
5/1/1950	2.625	44.8923	#NA	#NA	#NA	#NA	11.875	#NA	#NA	#NA	12.5	5.875
5/8/1950	2.3125	46.529	#NA	#NA	#NA	#NA	11.875	#NA	#NA	#NA	13.6875	5.625
5/15/1950	3.25	44.6585	#NA	#NA	#NA	#NA	10.625	#NA	#NA	#NA	12.2187	4.75
5/22/1950	3.0625	43.314	#NA	#NA	#NA	#NA	11.25	#NA	#NA	#NA	11.8437	4.3125
5/29/1950	2.875	43.898	#NA	#NA	#NA	#NA	10	#NA	#NA	#NA	11.9375	4.375
6/5/1950	2.75	45.593	#NA	#NA	#NA	#NA	15	#NA	#NA	#NA	12.4375	5.0625

Figure 4.2: Sample Stock price data: Pharmaceuticals and Biotechnology

Figures 4.3 and 4.4 illustrate the stock price distribution for the active and dead companies falling under the Pharmaceuticals and Biotechnology sector. From observing the stock price distribution charts, it becomes clear that it is difficult or almost impossible to predict the next stock price or even the future state of the stock price, i.e. whether the price will be high or low. In our study, by classifying between the active and dead companies, we have tried to infer statements about the performance of each company and whether it would be

a potential survivor or not.

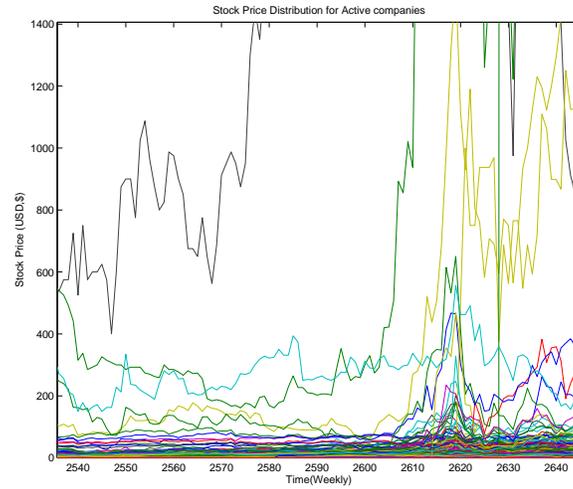


Figure 4.3: Stock price distribution of Active companies

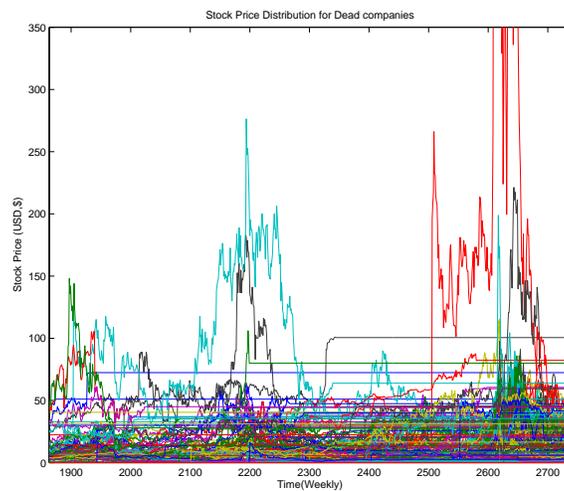


Figure 4.4: Stock price distribution of Dead companies

As observed quantitatively in Figures 4.3 and 4.4, the stock price distribution is not an i.i.d (independent and identical distribution). So in order to normalize the distribution, we

apply the concept of Black and Scholes [54, 55, 56] theory on stock price option trading as, *Average Weekly returns*, $r_w = \ln\left(\frac{S_w}{S_{w-1}}\right)$, where S_w is the closing stock price for the current week, and $S_w - 1$ is the closing stock price of the previous week. The normalized distribution would prove to be a better fit with the Gaussian Mixture Model used in the Fisher kernel. ..Figures 4.5 and 4.6 illustrate the log-normal stock price returns' distribution.

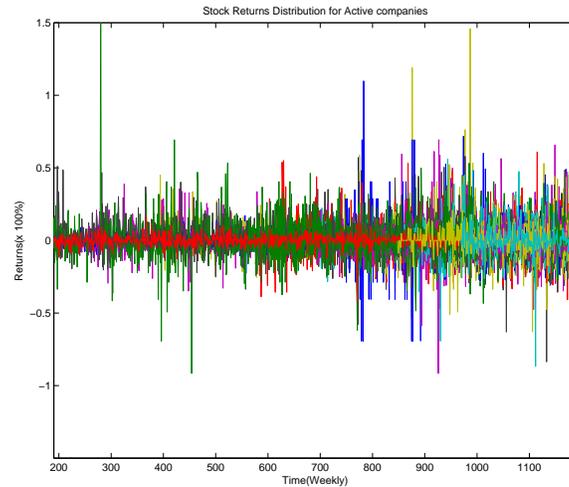


Figure 4.5: Log-Normal stock returns distribution of Active Companies

An *active* company in this context indicates that it is currently trading and is listed in a particular stock exchange. Whereas, a *dead* company indicates that the firm has been delisted from the stock exchange, and that it no longer performs stock trading. A company can be listed as 'dead' for many reasons such as bankruptcy, mergers or acquisitions. Thomson Datastream uses a flat plot or a constant value as an indication that the company has stopped trading in the exchange. This becomes clear from Figure 4.4.

4.2.2 Plot generation, SVM application and LDA validation

The log-normal stock returns data, obtained using the Black and Scholes theory are first used for finding the initial estimates of mean, variance and weight vectors using the Expectation Maximization algorithm [47, 48]. In this particular study, we have taken two scenarios for

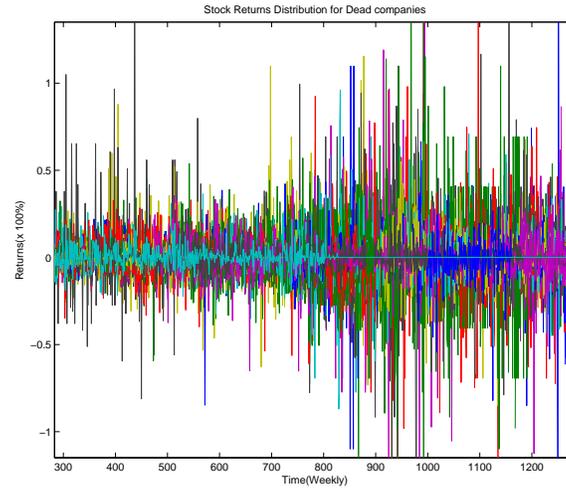


Figure 4.6: Log-Normal stock returns distribution of Dead Companies

finding the Gaussian estimates of the log-normal returns dataset. Using these two scenarios we generate two sets of results with respect to the Fisher score plots, SVMs and LDA.

1. Scenario 1 - Separate Estimates for Active and Dead Companies

The initial Gaussian estimates are calculated separately for Active and Dead companies' stock returns datasets, which are further used for calculating the Fisher scores for all the companies as per the methodology described in section 3.2.1. The Fisher scores for each company are obtained with respect to three parameters. These parameters are basically the derivatives of the global log-likelihood of each dataset with respect to each of mean, variance and weight vectors. These Fisher scores when plotted in three dimensions provide a scope for visually classifying between the active and the dead companies, as shown in Figure 4.6. At this stage, we have basically performed a transformation of a financial time-series into three dimensions. That is, for each company we have processed its stock market data into a set of six Fisher scores. In order to plot these Fisher scores, we summed up the Fisher score pairs for all the parameters, as indicated in Equations 4.1, 4.2 and 4.3. So the coordinates of the 3-D plot in Figure

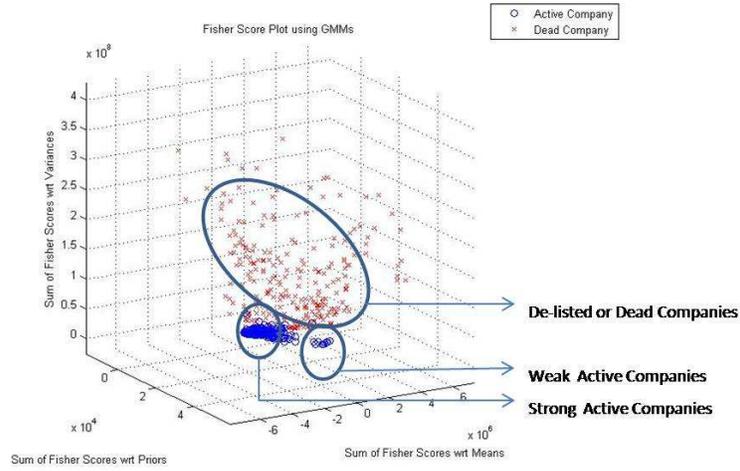


Figure 4.7: Fisher score plot for visualization - Separate Gaussian estimates

4.7 are basically the sum of Fisher scores - SFS_1 , SFS_2 , SFS_3 .

$$SFS_1 = \left(\frac{d}{da_1} + \frac{d}{da_2} \right) \log P(C | M, \theta) \quad (4.1)$$

$$SFS_2 = \left(\frac{d}{d\mu_1} + \frac{d}{d\mu_2} \right) \log P(C | M, \theta) \quad (4.2)$$

$$SFS_3 = \left(\frac{d}{d\sigma_1} + \frac{d}{d\sigma_2} \right) \log P(C | M, \theta) \quad (4.3)$$

Using the illustrated Figure 4.7 data, we used support vector machines for training and thereby classifying the potentially surviving and collapsing companies. Firstly, we randomly split the Fisher score dataset into training and testing groups, using the Hold-out method. We then train the SVM classifier using the training dataset, thus generating the support vectors for the learning model. These support vectors are further re-evaluated using the testing dataset. Table 4.3 shows the SVM results for 3-D and 6-D Fisher scores.

Table 4.3: Results from SVM performance - Financial time series

Dataset	Sensitivity (%)	Error rate (%)	Classification accuracy (%)
3D Fisher Scores	92.96	3.52	96.48
6D Fisher Scores	96.09	2.35	97.66

In order to validate our results, we further used the method of Linear Discriminant Analysis (LDA)[1] along with the Leave-one-out cross validation technique. Tables 4.4 and 4.5 show these results in details for training, testing and cross-validating 3D and 6D Fisher scores.

Table 4.4: LDA results for 3-dimensional Fisher scores - Financial time series

Category		Predicted Membership	
		Dead	Active
Original Data	Count of Data classified		
	Dead	235	21
	Active	0	256
% of Data Classified			
	Dead	91.8	8.2
	Active	0.0	100.0
Cross-validated data	Count of Data classified		
	Dead	234	22
	Active	0	256
% of Data Classified			
	Dead	91.4	8.6
	Active	0.0	100.0

Also,

- Cross validation was done only for those cases in the analysis. In cross validation,

each case was classified by the functions derived from all cases other than that case.

- 95.9% of original grouped cases correctly classified.
- 95.7% of cross-validated grouped cases correctly classified.

Similarly we got the results for the six-dimensional Fisher scores as shown in Table 4.5.

Table 4.5: LDA results for 6-dimensional Fisher scores - Financial time series

Category		Predicted Membership	
		Dead	Active
Original Data	Count of Data classified		
	Dead	234	22
	Active	0	256
	% of Data Classified		
	Dead	91.4	8.6
	Active	0.0	100.0
Cross-validated data	Count of Data classified		
	Dead	234	22
	Active	0	256
	% of Data Classified		
	Dead	91.4	8.6
	Active	0.0	100.0

Similarly as compared to previous case,

- 95.7% of original grouped cases correctly classified.
- 95.7% of cross-validated grouped cases correctly classified.

2. Scenario 2 - Common set of Estimates for Active and Dead Companies

As opposed to the first scenario, in this situation, we find common Gaussian estimates

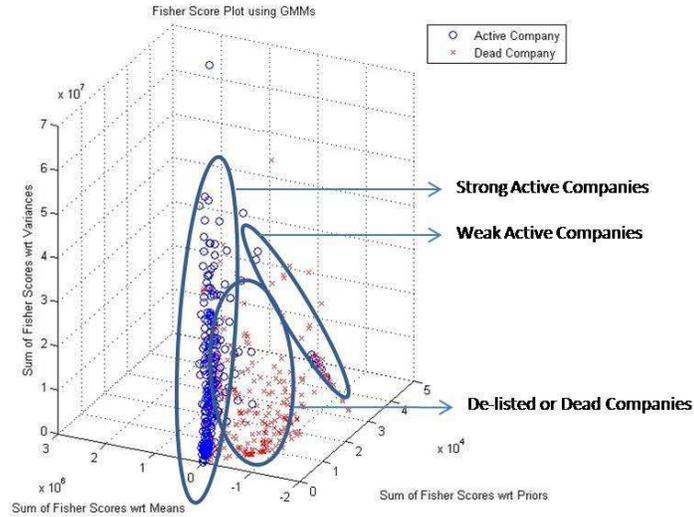


Figure 4.8: Fisher score plot for visualization - Common Gaussian estimates

for both Active and Dead companies' stock returns datasets. In order to calculate the estimates, we combine the Active and Dead stock returns data into one dataset. We then process this dataset using the Expectation Maximization algorithm [47, 48]. Subsequently we generate the Fisher scores, and thus obtain the three dimensional Fisher score plot, as shown in Figure 4.7.

As is in previous scenario, the SVM training and testing yields the following results in Table 4.6 , with respect to 3-D and 6-D Fisher scores.

Table 4.6: Results from SVM performance - Financial time series

Dataset	Sensitivity (%)	Error rate (%)	Classification accuracy (%)
3D Fisher Scores	70.31	16.01	83.98
6D Fisher Scores	78.90	14.06	85.93

These results are further validated using LDA approach as follows. Following Tables

4.7 and 4.8, show these results with respect to training, testing and cross-validation.

Table 4.7: LDA results for 3-dimensional Fisher scores - Financial time series

Category		Predicted Membership	
		Dead	Active
Original Data	Count of Data classified		
	Dead Active	167 13	89 243
	% of Data Classified		
	Dead Active	65.2 5.1	34.8 94.9
Cross-validated data	Count of Data classified		
	Dead Active	167 13	89 243
	% of Data Classified		
	Dead Active	65.2 5.1	34.8 94.9

Also,

- Cross validation was done only for those cases in the analysis. In cross validation, each case was classified by the functions derived from all cases other than that case.
- 80.1% of original grouped cases correctly classified.
- 80.1% of cross-validated grouped cases correctly classified.

Similarly we got the results for the six-dimensional Fisher scores as shown in Table 4.8.

Similarly as compared to previous case,

- 83.2% of original grouped cases correctly classified.

Table 4.8: LDA results for 6-dimensional Fisher scores - Financial time series

Category		Predicted Membership	
		Dead	Active
Original Data	Count of Data classified		
	Dead Active	188 18	68 238
	% of Data Classified		
	Dead Active	73.4 7.0	26.6 93.0
Cross-validated data	Count of Data classified		
	Dead Active	187 18	69 238
	% of Data Classified		
	Dead Active	73.0 7.0	27.0 93.0

- 83.0% of cross-validated grouped cases correctly classified.

The following section discusses the various observations and inferences we made from our experiments with Financial time-series.

4.3 Observations and Discussions

From the Fisher score plots shown in Figures 4.7 and 4.8 we can make the following observations:

- We observe two clusters in the plots 4.7 and 4.8 - Active companies (indicated in blue), and Dead companies (indicated in red). These clusters provide a scope for visually classifying between the two categories, and also indicate the tendency of each company - whether it is likely to remain active or become dead in the near future.
- The Fisher score points (corresponding to different companies) that lie very close to

each other in the 'active' region, indicate that these firms are likely to remain healthy (money-wise) in the stock market, and may not be delisted in the near future. Active companies' Fisher score points, which are away from the denser active region, and tend to move towards the dead cluster, indicate that these companies may likely be dead in the near future. They might go either bankrupt, or may undergo a merger or an acquisition.

- We also observe that the stock price fluctuation of dead companies is quite high as compared to active companies (Figures 4.3 and 4.4). This indicates that the active companies lying close to one another, thus creating a dense active region, are more suitable for long term investments. If an investor wants to make considerably large profits in a short amount of time, then investing in an active company (away from the dense cluster), which is likely to go dead, would be an appropriate option. The only constraint for the investor, that lies in this case would be - At exactly what time should the shares be sold in order to make maximum profits.

Further training and testing with support vector machines yielded us about 96-97% classification accuracy in **Scenario 1**, and about 83-85% in **Scenario 2**. These results were further validated by using the Fisher scores (3-D and 6-D) as input data in the LDA approach, and the classification rate turned out to be about 95% **for Scenario 1**, and 83% **for Scenario 2**. Note that the 3-D Fisher scores are actually the sum of the 6-D Fisher scores, as indicated in Equations 4.1, 4.2 and 4.3. We also observe that there is significant difference in the classification accuracies between **scenarios 1** and **2**.

In scenario 1 we find separate Gaussian estimates for active companies and dead companies, thus considering each category as a separate input set to the learning machine. Whereas in scenario 2, we find common Gaussian estimates by considering the active and dead companies as a single input dataset. This indicates that Gaussian estimates are essential indicators (in the form of mean, variance and weight vectors) for each category, and define the behavior of that category (active or dead).

However, if we find the common estimates, then we are considering both classes to belong to the same input set, and hence the values obtained for the Gaussian estimates define the behavior of the entire sector taken into consideration. This might be useful from a holistic perspective: i.e. if we are assessing the fate of an entire economic sector by analyzing and predicting the potential survival or collapse of active businesses in that sector. But if we are simply classifying and predicting the potential survivors and collapsing firms, without considering the economic sector as a whole, then using separate Gaussian estimates for active and dead companies' datasets might be more suitable.

The LDA results indicate that our method of applying Fisher kernels for dimension transformation, followed by the use of support vector machines was highly suitable for performing pattern classification and predictions on financial time-series data. One of the reasons behind such a visually accurate separation may be that, we have exploited the relationship between two consecutive stock prices by applying the Black and Scholes theory [54, 55, 56], thus obtaining the weekly stock price returns for each company. These stock price returns give us a distribution, which is easier to analyze and model using GMM, as compared to a non-i.i.d distribution. Other reasons behind such accuracy rates (96% using SVMs) could be the use of Gaussian radial basis function as our kernel function - which creates a good classifier for non-overlapping classes, and application of SMO method for finding the hyperplane, which splits our large quadratic optimization problem into smaller portions for solving.

This experiment was done for the financial data from Pharmaceuticals and Biotechnology sector (with 256 active and 256 dead companies), and can be extrapolated to other sectors as well. The classification accuracy of our learning model is highly dependent on the stock price fluctuations and hence the corresponding behavior of the stock returns, and so it may vary with other sectors. But the accuracy rate would tend to remain approximately the same in most sectors' study.

The above observations about the high performance of Fisher kernels and SVMs will become more convincing as we proceed to the next Chapter 4, wherein we apply these concepts to a biomedical time-series, for predicting the knee joint abnormality with respect

to cartilage degeneration.

Chapter 5

Application to Biomedical domain - Osteoarthritis

5.1 About Osteoarthritis and other joint-related disorders

OSTEARTHRTIS (OA) is a clinical syndrome of joint pain followed by certain functional limitations and hence a reduced quality of life. It is one of the most common forms of arthritis and a leading cause of pain and disability worldwide. The most common body parts affected by OA include knees, hips and small hand joints. Although OA is accompanied by acute joint pain and reduced functionality, sometimes structural changes occur without any other common symptoms. Such frequent discordance in OA pathology indicates that there is a need for separate consideration in epidemiological studies and clinical trials of osteoarthritis treatments.

The lower human body is considered to be the most significant support for the human structure's physical functioning. The major parts of the lower human body which are used in daily activities are the knee joints. They carry out the basic human movements of walking, running and sitting, and support almost three-fourths of the human body. The joints are lubricated and connected through the cartilage by a fluid comprising of about 70% liquid state protein, and about 30% of strong connective tissue [57].

Extensive use of knee joints during one's lifespan, may reduce the liquid protein in the

cartilage, thus leading to increased friction between the fibrous tissue and the bone, and can lead to the degeneration of articular cartilage. Some of the symptoms associated with this degenerative disorder are joint pain, tenderness, inflammation and inability to walk properly. Although the human body naturally tries to sustain the damage, the internal regrowth processes rather accelerate this damage, thus leading to more complications. Such disorders can eventually lead to conditions such as osteoarthritis.

OA is a metabolically active process which involves all the joint tissues (cartilage, bone, synovium/capsule, ligaments and muscle). The key physical changes in the joints include loss of articular cartilage and restructuring of adjacent bone with new bone formation. A combination of tissue loss and its regeneration defines osteoarthritis as the repair process of joints. This activity of tissue regeneration may be triggered by extensive trauma to the joint, but once initiated all the joint tissues take part, showing increased cellular activity and tissue production. Osteoarthritis is a slow but efficient repair process that often compensates for the pain caused in the regeneration process, resulting in an altered but symptom-free joint. However, in some cases, because of excessive injury or use, or low repair potential, the osteoarthritis process cannot compensate, thus leading to continuous tissue damage and joint failure.

OA is not defined as a disease but is more of a complex joint disorder, characterized by many risk factors. These risk factors can be broadly classified as follows:

- *Genetic* risk factors, which include hereditary characteristics of knees, hands and hips.
- *Bio-mechanical* risk factors such as joint injuries, overwhelming recreational or occupational usage, etc.
- *Constitutional* risk factors, which include bone density, ageing, obesity, etc.

OA is also characterized by other risk factors such as environmental factors, lifestyle, which are important, then an individual is taken into consideration for diagnosis.

Osteoarthritis is not just a disorder confined to the cartilage wear and tear. It is also associated with the growing age of a person. This disease is rarely found in case of younger

people. As a person ages, the liquid content of the cartilage decreases gradually, thus putting more pressure on the cartilage fibrous tissue. This eventually leads to wearing of the tissue due to rubbing against the joints, and thus causing acute pain as seen in osteoarthritis. Osteoarthritis is not just associated with knee joints but also with hands, feet, spine and hips. As this disorder progresses, certain common symptoms can be observed in a patient such as stiffness and acute pain, which exponentially become worse, when the joints are used more throughout the daily life. Osteoarthritis is categorized into two: Primary and Secondary. Figure 5.1 [3] depicts the degeneration of articular cartilage in knee joints, hence leading to OA.

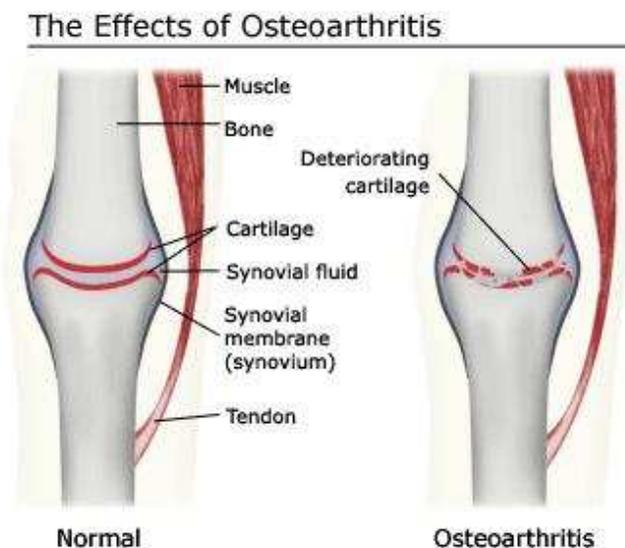


Figure 5.1: OA development [3]

Primary osteoarthritis is a chronic degenerative process of the articular cartilage, which progresses gradually with person's age. It is usually not associated with any other side-symptoms, and occurs with progressive wearing of the cartilage tissue, due to the slow depletion of the liquid content in the cartilage. The human body reacts to this disorder by exhibiting new bone outgrowths near the joints, thus causing acute pain in the joints and inability to walk or perform certain movements. Secondary osteoarthritis is however

associated with not just the degeneration of cartilage tissue, but also with other disorders such as congenital deformity of joints, injuries, hormonal imbalances, obesity, pregnancy and diabetes.

Early detection of such disorders can motivate the use of various treatment options for slowing down the wear and tear of cartilage. The methods usually employed for early treatment of osteoarthritis or other joint related problems, include physiotherapy, medication using anti-inflammatory drugs, topical ointments, and invasive methods such as joint replacement surgeries, acupuncture and fluid injections in the joint areas. For more detailed information on joint-related disorders and particularly osteoarthritis, refer to [57, 58].

5.2 Studying Osteoarthritis using intelligent methods - Literature survey

Joint related disorders such as osteoarthritis cannot be easily detected using simple laboratory tests. Blood tests are performed to exclude diseases that can cause secondary osteoarthritis, as well as to exclude other arthritis conditions that can mimic osteoarthritis. OA can be detected using certain invasive procedures such as Arthrocentesis and Arthroscopy. Arthrocentesis is often performed in the doctor's office. During arthrocentesis, a sterile needle is used to remove joint fluid for analysis. Joint fluid analysis is useful in excluding gout, infection, and other causes of arthritis. Removal of joint fluid and injection of corticosteroids into the joints during arthrocentesis can help relieve pain, swelling, and inflammation. Arthroscopy is a surgical technique whereby a doctor inserts a viewing tube into the joint space. Abnormalities of the cartilage and ligaments can be detected and sometimes repaired through the arthroscope. If successful, patients can recover from the arthroscopic surgery much more quickly than from open joint surgery. Finally, a careful analysis of the location, duration, and character of the joint symptoms and the appearance of the joints helps the doctor in diagnosing osteoarthritis. Bony enlargement of the joints from spur formations is characteristic of osteoarthritis.

Although certain symptoms such as acute joint pain or inability to walk may be help-

ful in diagnosing a patient with OA, such problems, require very accurate and preferably non-invasive procedures for detection. This has been made possible by the application of computers to medicine. OA can be easily detected using X-ray technology. If a subject has been affected by osteoarthritis, then in the X-ray, one might be able to observe the loss of cartilage fluid, making the knee-cap and the adjacent bones around it, more visible. The common X-ray findings of osteoarthritis include loss of joint cartilage, narrowing of the joint space between adjacent bones, and bone spur formation. Simple X-ray testing can be very helpful to exclude other causes of pain in a particular joint as well as assist in decision making as to when surgical intervention should be considered. Other methods of diagnosing include CT Scan, Ultrasound, or a radiograph. Figure 5.2 [4] shows an X-ray image of a typical knee joint affected by Osteoarthritis.



Figure 5.2: X-ray image showing OA affected knee joint [4]

Apart from using computers directly to find OA symptoms, some intelligent computing methods such as neural networks [59, 60, 61, 62, 63], support vector machines [64, 65], genetic algorithms, fuzzy logic, principal component analysis [66], Auto-regressive modelling [67, 68, 69, 70, 71], etc. can also be helpful in detecting OA and its extent of severity

in the subject. These intelligent or machine learning techniques have proven to be quite useful in detecting and predicting various joint related disorders. Some of the applications include pattern classification of hip images into OA severities, analyzing trabecular bone structure, assessing gait patterns, identifying joint locations with OA symptoms, detecting Patellofemoral pain syndrome and detecting recovery from knee replacement surgery.

The intelligent methods applied in our study may require extensive amounts of input data (since it is acquired from patients or test subjects, data collection is not simple), but the outcomes so far have been quite promising and can prove to be helpful in reducing the number of invasive procedures taking place each year. It would also aid in early detection and hence prompt solutions for treating osteoarthritis. This study will help us in determining the tendency of a subject getting affected by osteoarthritis, and even to predict the possibility that a subject may need to undergo joint replacement surgery.

5.3 Application of Fisher kernels and SVMs to knee-signals for classification and predictions

In this research, we are dealing with a binary classification of knee joint signals, sorting them into 'abnormal' and 'normal' categories. Usually, whenever we do a binary classification, we tend to separate the two categories of data using a hyperplane. In this study, the classification involves clustering of data points in the Fisher scores space, which is obtained by applying the Fisher kernel method. However, the two classes involved in the classification have been defined from the start, while collecting the data. Thus, the learning method that is used in this study is not completely unsupervised, as the data is already labeled. Figure 5.3 shows an illustration for the methodology adopted.

Using this methodology we divide our implementation into various stages of data collection, processing and outcomes, which will be explained as we read further.

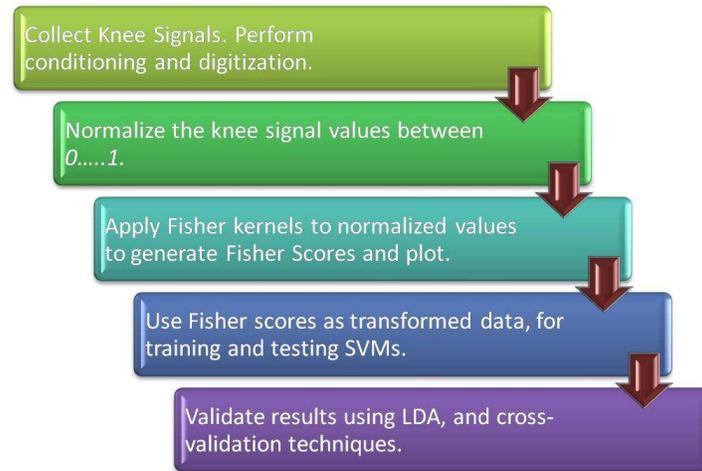


Figure 5.3: Methodology for classifying knee joints

5.3.1 Signal collection and Processing

The vibration signals collected from knee joints during normal movement are known as *vibroarthrographic* signals. These signals can be related to the properties of the joints, such as density, roughness, lubrication of cartilage, and can prove to be useful for early detection of cartilage degeneration related disorders such as OA. The signals for this experiment were collected by Krishnan *et.al* [71, 70], for 38 abnormal and 51 normal knee-joint cases. Following points summarize the data collection procedure described in [71].

- The subjects were made to sit on a table, with their legs suspended.
- In order to detect the signals and record them, a piezoelectric accelerometer was placed on the surface of the knee.
- the signals were recorded using an instrumentation recorder during the swing motion of the knee joints from 135° to 0° and back to 135° within a time period of 4 seconds.
- The angles were measured using an electronic goniometer [71], placed on the lateral side of the knee.

- The signals were then amplified and conditioned by a 10 Hz to 1000 Hz bandpass filter, using isolation preamplifiers and universal amplifiers. The VAG signals obtained were then digitized with a sampling rate of 2.5 kHz and 12 bits/sample.

The digitized signals were then stored in an Microsoft Excel [72] worksheet, as illustrated in Figure 5.4. The rows indicates the count of the data available and the columns represent the subjects.

	Subjects						
Timeline	27	71	166	61	131	74	99
	20	71	167	61	133	71	102
	20	70	168	61	133	69	102
	18	70	168	59	134	72	98
	16	70	168	59	132	73	99
	16	66	169	57	136	68	97
	17	70	169	59	136	71	102
	16	70	168	57	137	73	104
	17	68	167	60	136	74	99
	18	70	163	60	138	77	101
	19	74	164	61	141	72	106
	17	74	165	61	142	73	109
	16	71	165	57	140	77	108
	18	72	168	54	139	80	110
	17	68	163	54	137	76	108
	16	68	161	55	134	74	111

Figure 5.4: Sample knee signal data

The plots of these signals can be observed in Figures 5.5 and 5.6.

In order to simplify our calculations, we normalize the dataset values for each case study between the interval $0 \dots 1$, in order to generate the Fisher scores for all the subjects present. Note that, although the Fisher score plot in the next section, would be a cluster plot showing two classes, the labels of these classes (abnormal and normal) are known to us from start. So this learning method is supervised in nature. The normalized VAG signals are as shown in Figures 5.7 and 5.8.

Following section 5.3.2 describes the method of generating the Fisher scores and the results obtained after training and testing the subsequent support vector machine.

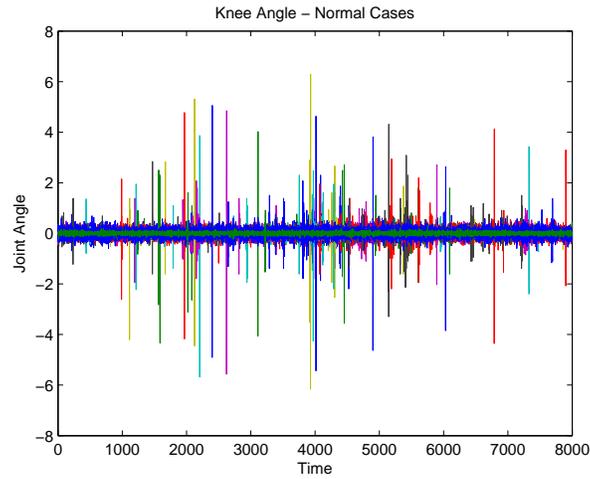


Figure 5.5: Normal Knee Signals

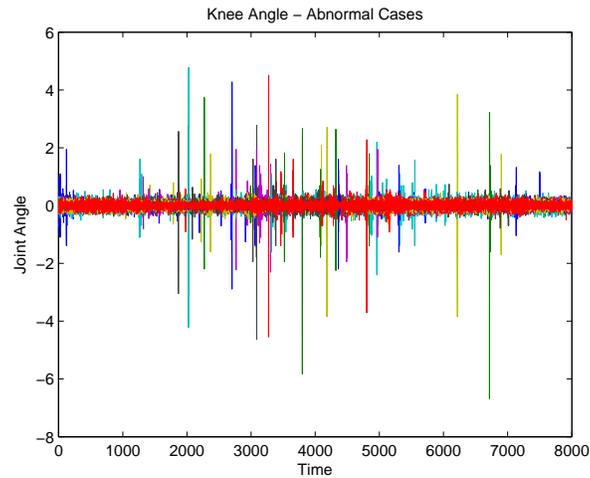


Figure 5.6: Abnormal Knee Signals

5.3.2 Plot generation and SVM application

In our study, we are trying to transform each signal into a three dimensional point, such that we get a classification pattern between the abnormal and normal cases. Using this classification pattern, we try to generate statements about the future states of knee joints,

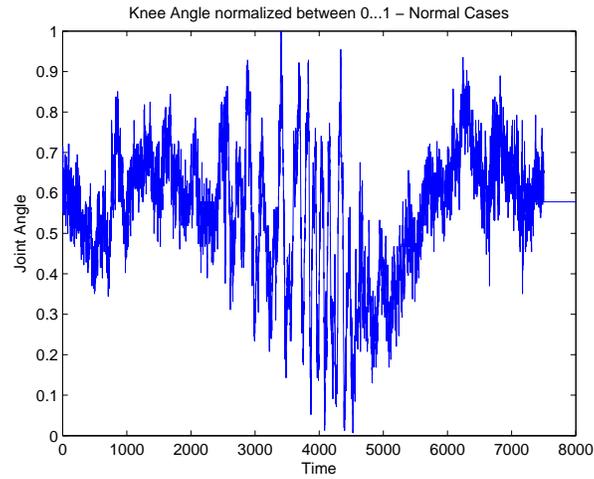


Figure 5.7: Sample normal knee signal

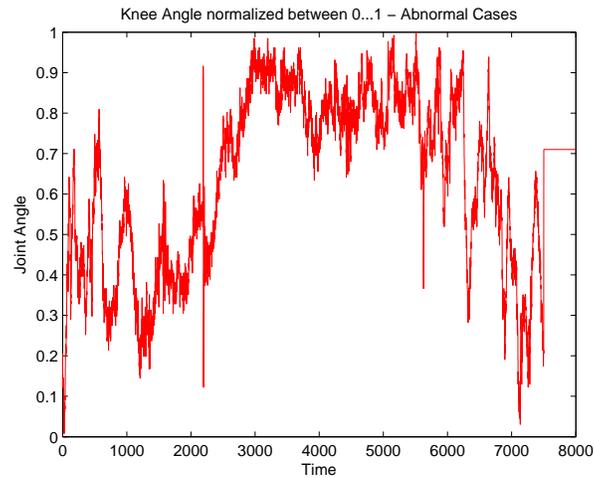


Figure 5.8: Sample abnormal knee signals

and their potential of getting affected by osteoarthritis.

An 'abnormal' case study in this context indicates that the knee joint is currently experiencing degeneration of cartilage and has the potential of getting affected by osteoarthritis. Whereas, a 'normal' case study indicates that the knee joint is functioning properly and has less chances of experiencing cartilage degeneration.

The Fisher scores were generated as per the methodology explained in section 3.2.1. The signals used for generating the Fisher score matrices for each subject were the normalized VAG signals. The VAG data was first used for finding the initial estimates of mean, variance and weight vectors using the Expectation Maximization algorithm [48, 47]. The dataset was then processed using Fisher kernels implemented with a Gaussian Mixture Model in order to obtain the Fisher Scores with respect to three parameters.

These parameters are basically the derivatives of the global log-likelihood of each dataset with respect to each of mean, variance and weight vectors. These Fisher scores when plotted in three dimensions provide a scope for visually classifying the abnormal and normal case studies, as shown in Figure 5.9. At this stage, we basically performed a transformation of the knee signal time-series into three dimensional sum of Fisher scores. That is, for each case study we processed its knee joint data into a set of six Fisher scores. In order to plot these Fisher scores, we summed up the Fisher score pairs for all the parameters, as indicated in Figure 5.9 and Equations 4.1, 4.2 and 4.3.

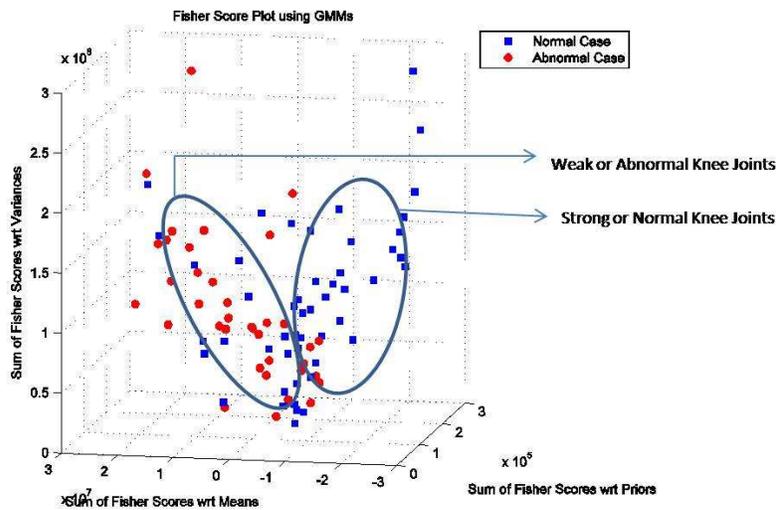


Figure 5.9: Fisher score plot for visualization

Using the illustrated (Figure 5.9) data, we used support vector machines for training and thereby classifying the potentially abnormal and normal case studies. Firstly, we randomly

split the Fisher score dataset into training and testing groups, using the Hold-out method. We then applied SVMs for training and testing of the system, using the Gaussian radial basis function (RBF), as our kernel function. For finding the hyperplane separating the two classes, we used the method of Sequential Minimal Optimization (SMO)[45]. SVMs were applied to both three-dimensional and six-dimensional Fisher scores for classification and prediction. The results for this have been shown in Table 5.1.

Table 5.1: Results from SVM performance - Knee Signals

Dataset	Sensitivity (%)	Error rate (%)	Classification accuracy (%)
3D Fisher Scores	84.21	15.91	84.09
6D Fisher Scores	89.47	11.37	88.63

5.3.3 LDA analysis and cross-validation

In order to validate our results, we further used the method of Linear Discriminant Analysis (LDA)[1] along with the Leave-one-out cross validation technique. Following Tables 5.2 and 5.3 show these results in details for training, testing and cross-validating 3D and 6D Fisher scores.

Also,

- Cross validation was done only for those cases in the analysis. In cross validation, each case was classified by the functions derived from all cases other than that case.
- 82.0% of original grouped cases correctly classified.
- 75.3% of cross-validated grouped cases correctly classified.

Similarly we got the results for the six-dimensional Fisher scores as shown in Table 5.3.

- 91.0% of original grouped cases correctly classified.

Table 5.2: LDA results for 3-dimensional Fisher scores - Knee Signals

Category		Predicted Membership	
		Abnormal	Normal
Original Data	Count of Data classified		
	Abnormal	35	3
	Normal	13	38
	% of Data Classified		
	Abnormal	92.1	7.9
	Normal	25.5	74.5
Cross-validated data	Count of Data classified		
	Abnormal	31	7
	Normal	15	36
	% of Data Classified		
	Abnormal	81.6	18.4
	Normal	29.4	70.6

- 88.8% of cross-validated grouped cases correctly classified.

As we proceed further to the next section, we discuss our observations and inferences from above results.

5.4 Observations and Discussions

From the Fisher score plot shown in Figure 5.9 we can make the following observations:

- Two patterns can be observed from the plot: Abnormal knee joints (indicated in red), and normal knee joints (indicated in blue). These patterns prove to be useful for visually classifying between the two classes respectively.
- The Fisher score points (which correspond themselves to different subjects) that lie very close to each other in the 'normal knee joint' region, indicate that those respective

Table 5.3: LDA results for 6-dimensional Fisher scores - Knee Signals

Category		Predicted Membership	
		Abnormal	Normal
Original Data	Count of Data classified		
	Abnormal	34	4
	Normal	4	47
	% of Data Classified		
	Abnormal	89.5	7.9
	Normal	7.8	92.2
Cross-validated data	Count of Data classified		
	Abnormal	34	4
	Normal	6	45
	% of Data Classified		
	Abnormal	89.5	10.5
	Normal	11.8	88.2

subjects have a greater probability of not getting affected by joint related disorders, and are likely to remain healthier, as compared to those points which are comparatively more nearer to the 'abnormal knee joint' region or quite at a distance away from the dense 'normal knee joint region'.

- The 'abnormal knee joint' region corresponds to those subjects, which have shown to have OA symptoms, as well as those which have been affected by OA or other joint related disorders. We observe that the points that are closely located to one another in the 'abnormal knee joint' region, their corresponding subjects have a higher probability of undergoing invasive treatments for OA, such as joint replacement surgery. We say this because these subjects tend to show a higher severity of cartilage degeneration, as compared to the abnormal subjects which are distant from the dense abnormal region. The Fisher points away from the denser abnormal region, indicate that the corresponding subjects have a lower probability of undergoing joint replacement surgery, and can

be treated by simple non-invasive procedures such as physiotherapy.

Upon further training and testing with support vector machines, we found out the accuracy to be about 84-88 % (Table 5.1). This is a fairly high classification accuracy. The SVM was trained using a Gaussian radial basis kernel, and the SMO method was employed for finding the hyperplane. SVMs were not just used for classification, but were also applied for predicting the potential of a subject getting affected by OA, or a subject needing to undergo a joint replacement surgery.

The SVM results have been further validated using LDA approach along with the Leave-one-out cross validation approach as indicated in Tables 5.2 and 5.3. Similar classification accuracy was obtained during validation, with 75-88 % of cross-validated groups correctly classified, and about 82-91 % for original groups. Note that the 3-D Fisher scores are actually the sum of the 6-D Fisher scores, as indicated in Equations 4.1, 4.2 and 4.3.

In conclusion, our experiments with the knee joint signals for classifying the potentially abnormal and normal joints, along with predicting the probability of a subject needing to undergo a joint replacement surgery, have yielded fairly high results. These good classification and prediction accuracies can be attributed to the fact that Fisher kernels along with SVMs and LDA approaches, have performed really well. One of the reasons behind such high accuracies may be the characteristic properties of Fisher kernels - retaining the essential features during dimension transformation. Also we consider the fact that SVMs have shown quite promising results in other applications such as face recognition, speech recognition and verification, pollen grain classification, document classification, etc.

As we proceed to the next Chapter 6, we will see how these experiments with time-series signals from different domains can help not only in modeling complex systems, but can also prove to be of good commercial use (profit or non-profit).

Chapter 6

Discussions, Conclusions and Future Work

THIS thesis work was aimed at modelling two complex systems, whose outputs were in the form time series signals. As per our experiments and the subsequent discussions in sections 4.3 and 5.4, we can conclude that our application of kernel based machine learning techniques for modelling time series signals was highly successful and promising. The study identifies not only the benefits to an individual but also brings out the advantages from a holistic approach. In our previous study [53] Fisher kernels were not able to perform binary classification in two dimensions. But in this study, by introducing three dimensional Fisher scores, we have been able to separate and visualize the two classes, more accurately. A normal distribution is easier to analyze and model using GMM (Gaussian Mixture Model), as compared to a non-i.i.d distribution. In other words, we can say that Gaussian mixture models give the best fit for normally distributed datasets. In our experiments with kernel-based machine learning methods, we applied the learning methodology to time-series signal classification and predictions. We had taken two different domain-specific time-series signals into consideration for our research:

- **Stock price distributions**

In this study we used the historical stock price distributions of various companies in a particular economic sector. Using Thomson's Datastream software, we downloaded the stock price data of all the active and dead companies in the Pharmaceuticals and

Biotechnology sector. The stock exchanges taken into consideration were NASDAQ [15], NYSE [13], TSX [14] and TSX-Ventures [14]. All the stock prices were adjusted to USD\$ (United States Dollar) currency. Using these stock price distributions we computed the stock returns which are assumed to show an *i.i.d* nature. The stock returns dataset for all the active and dead companies was used for dimension transformation using Fisher kernels. This dimension transformation gave a three dimensional scatter plot for visually classifying between the active and dead companies. The Fisher scores were then applied to SVMs for classifying and predicting the potentially surviving and collapsing firms in the Pharmaceuticals sector. The objective in this study was to predict the chances of a currently active company to be delisted from the stock exchanges due to bankruptcy, or merger, or acquisition, or simply because it failed to follow the rules and regulations for trading in the market. We did this study considering two scenarios, as mentioned in section 4.3. **Scenario 1** gave us a classification rate of 96-97%, while applying **scenario 2** resulted in 83-85% accuracy. These results were validated using the LDA approach as well, which gave us similar classification rates. Studying the stock market dynamics using intelligent kernel-based methods for predicting and classifying the active and dead companies has a lot of **benefits**.

- To begin with, at the core level, it can prove to be quite useful to investors. Some people invest in the stock market to make long-term gains and find it as a way of growing their monetary assets, while some consider the investments for short term gains in order to make quick profits. A qualitative observation from Figures 4.5, 4.6, and 6.1 would reveal that strong active companies which have a very low chance of collapsing, yield lower returns as compared to those firms which have a high probability of being delisted from the exchange. A lower returns yielding company would be beneficial for an investor aiming to get high returns in the long run, whereas a high returns giving firm would be helpful for investor who wants to make quick profits and withdraw his investment from the market.
- If we consider a strategic management approach to this study, it proves to be useful

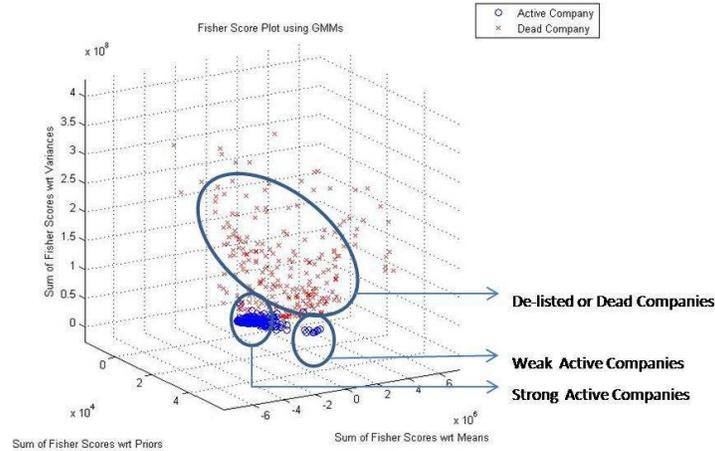


Figure 6.1: Fisher Score plot: Financial Time Series

from the observation that, we can figure out which company is likely to collapse or survive in the near future. Advising firms about their current situation in the potentially surviving region, would help them change their business strategies and management policies in order to remain strong competitors in the market.

- Looking from a holistic and sustainability perspective, this study can also prove to be beneficial in assessing the health of an economic sector, which might be of primary concern from a provincial or a national economic point of view. For example, if a particular economic sector has fewer number of strong active companies, as compared to those which are highly likely to collapse, then the administration can make certain decisions about managing that sector. These actions may include cuts in the provincial funds pumped into that sector, or completely transfer all the businesses in that sector to a different province where it would be more profitable to do business in the long run. On the contrary if a sector is observed to be very strong in generating revenues, i.e there are more potentially surviving firms than the collapsing ones, then one of the courses of action to be taken by the government might be promoting that sector by increasing public and private

investments into it.

- **Knee signal distributions**

The signals used from this domain, were the knee joint signals collected using the vibration arthrometry technique [70, 71]. These signals were digitized and conditioned before undergoing dimension transformation using Fisher kernels. The transformed data in the form of Fisher scores for each case study, was used as input to support vector machines, for performing classification and predictions. The main aim in this study was to classify between the potentially normal and abnormal knee joints, and predict the possibility of a subject to undergo a joint replacement surgery, or even predict the chances of a normal subject to get affected by osteoarthritis in the near future. About 84-88% of the subjects (abnormal and normal) were correctly classified using SVMs. This accuracy was further validated when the LDA approach gave about 75-88% correct classification rate. This study can prove to be helpful from an individual's perspective as well as from the health care system's **perspective**.

- This study is not restricted to knee joints, but can also be applied to other locations as well, such as hip joints or elbow joints. The normal knee joint scatter distribution in the Fisher score plot (Figure 6.2) reveals two kinds of subjects: the ones which have very healthy knee joints (normal region), and the ones with comparatively weaker joints (abnormal region). The subjects with strong and healthier knee joints have a very low probability of getting affected by Osteoarthritis and thus undergo a joint replacement operation in the near future. Whereas the subjects with weaker joints, have a higher probability of getting affected by joint related disorders.
- Considering an individual's perspective, this study can identify the subject's chances of getting affected by OA, and requiring further invasive treatments. At the same time, such research would also be helpful in reducing the number of invasive diagnostic and treatment procedures for OA, and promote the application

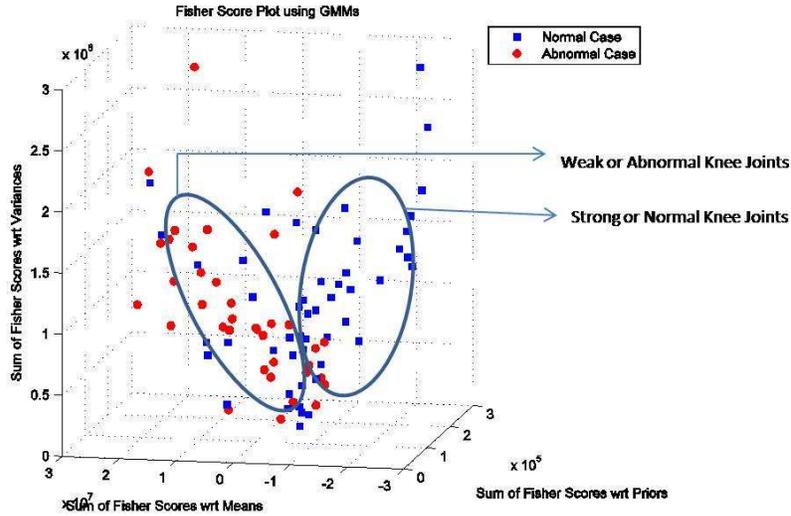


Figure 6.2: Fisher Score plot: Knee Signals

of non-invasive treatments such as physiotherapy, acupuncture, electrotherapy, external walking aids or even using nutraceuticals [57].

- Taking the health care system’s approach, this study can reveal certain demographics in a given region, such as the number of registered OA cases and their management, the treatment options available, etc..

A qualitative observation of the financial time-series signals (stock returns) reveals that the distribution is Laplacian in nature. That is, although the histogram of a sample vector appears to be bell-shaped as is in a normal distribution, we actually observe that the curve is peaked around the mean value and has fat-tails on either sides. In our study we had assumed the stock returns to have an *i.i.d* nature but the empirical distribution of the weekly stock returns has more observations around the mean value, and in the extreme tails of the distribution. Despite these facts, the results from our experiments look promising.

The high classification rates obtained in our study can also be justified by the fact that Fisher kernels and SVMs have performed really well for other applications such as object recognition (performances above 90%) [41], speech recognition [38, 33, 39] and speaker ver-

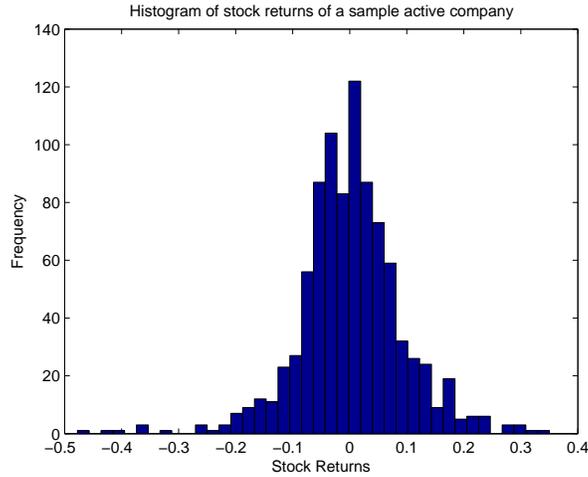


Figure 6.3: Histogram of a sample stock returns distribution

ification [46], recognizing hand gestures [73], detecting remote protein homologies [36], web audio classification [40], video classification [74], face recognition [75], pollen grain detection and classification [76], document classification [77], face detection [78], fault isolation in auto-braking systems [79], human skin color segmentation [80], document clustering [81], estimating length of stay in appendectomy patients [82], identity verification [83, 84], image fusion [85], spam categorization [86], image classification [87], digit recognition [88], human origin classification [89], etc.

Automation of our research work in order to yield dynamic outputs in the form of predictive statements and visualization plots can be pursued as a future study. Experimenting with variable-length time-series in this study has definitely opened doors for more research, such as,

- Assessing the time-frame within which companies can experience bankruptcy, mergers or acquisitions. Also there exists a scope for monitoring sustainability issues for various economic sectors on a provincial as well as national scale.
- Assessing the time-frame of cartilage degeneration, and a scope for monitoring Osteoarthritis.

Analyzing these issues in near future, will be quite interesting and challenging.

Chapter 7

Appendix A

7.1 Machine Learning Design

Many stages of machine learning theory are needed for pattern classification, which will be helpful in our study. Figure 7.1 illustrates the typical stages involved in a pattern recognition system [1].

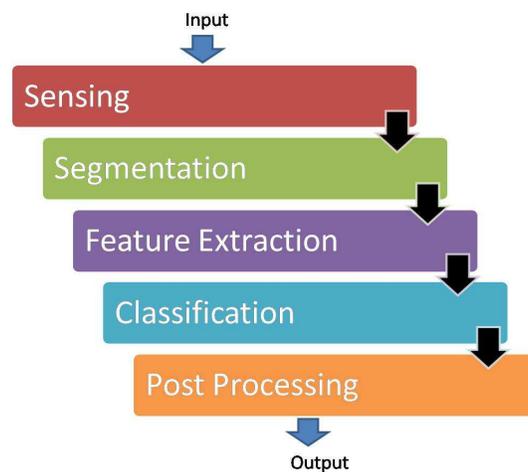


Figure 7.1: Typical Stages: Pattern Recognition

1. Sensing:

The inputs or set examples given to the learning machine, are sensed or identified using certain transducers such as an optical device, or a signal processing unit. The

properties of the sensor may be problem dependent such as bandwidth, SNR(signal to noise ratio), latency, etc.

2. **Segmentation:**

This involves separating the individual examples or patterns for mapping to the respective output or category labels. Segmentation is one of the most crucial stages, as there might involve problems with regards to overlapping of patterns; for example, while segmenting between salmon and sea-bass, one might observe some samples of fish overlapping each other or too close to one another, such that it becomes difficult for the sensor to recognize each fish sample. In our research, we have performed time-series segmentation. For example, the financial time-series signals in our study (composed of stock prices) has been segmented such that the companies with stock prices remaining constant over a certain time period, and are delisted from the exchange, are assigned to the Dead company label. Similarly companies with continuous variations in stock prices in a given time period and currently listed in the stock exchange are assigned to the Active company label.

3. **Feature Extraction:**

When a particular sample has been segmented by the sensor, its properties can then be easily identified and used for pattern classification for other samples. For example, when a fish has been identified as either a salmon or a sea-bass, then its various properties such as width, length, weight, color, etc., can be recorded, and later used for statistical comparisons within the entire dataset, for further classification. These specific or characteristic properties of the input samples constitute what are known as *features*, and the method hence employed to extract these features from each sample, is called *feature extraction*. This is the most important stage in pattern recognition systems. Only the most accurately *selected features* can lead to the designing of an optimum classification machine.

4. **Classification:**

The objective of the classifying stage is to fully use the feature vector extracted in the former stage, and hence map the input sample to the correct output or category label. Since perfect classification is not achievable, considering external and internal disturbances, we often determine the probabilistic value of an input sample being assigned to a particular output. The degree of difficulty in classification, again depends on the feature values available for each input sample. The feature values' variability in the same category, may be due to complexity of the pattern recognition problem and may be due to *noise*.

5. Post Processing:

In this final stage we evaluate the performance of the classifier. Various measures such as error rate can be used to monitor the classification. In order to improve system performance, the post-processor may exploit the input dependent information. We can also apply different feature combinations and create similar classifiers so as to assess their working, and select the most accurate classifier of the set.

7.2 Designing the optimum classifier

The design and selection of an optimum classifier from a given pool of features, can be achieved by employing the following design cycle [1] as shown in Figure 7.2.

Following briefly described are the various steps of the classifier design cycle:

- **Data Collection:**

Data collection can account for a major part of the cost of implementing a pattern classification system. In order to study the complex system, an initial small set of examples may be sufficient. But in order to create a learning model, for analyzing the complex system, we may need a larger set of examples (or input data). There may exist some limit on the number of examples to be taken into consideration while building an optimum classifier. This limit usually depends on the human expert's observations from the learning model's performance, as well as on the complex system taken into

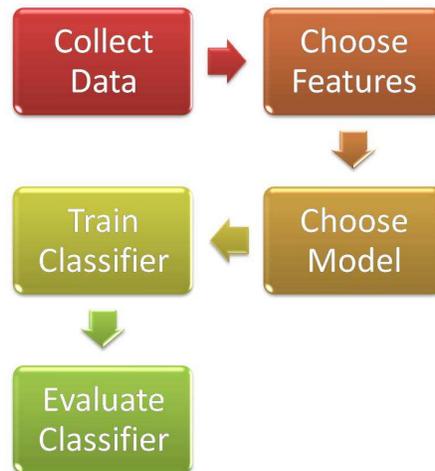


Figure 7.2: Classifier design cycle

consideration for modeling.

- **Feature Selection:**

When it comes to pattern classification, the method of selecting the most important and characteristic features from the input data, is a critical step. In order to classify two or more categories from a given input set, we need to select some distinguished features which are common properties of the input data. Using these features we try to parameterize the input data, so that classification becomes simpler. Also, prior knowledge about the input data plays an important role in feature selection. The desired features usually have the following properties:

- Simple to extract,
- Invariant to random transformations,
- Insensitive to noise, and,
- Useful for discriminating patterns in different categories

- **Model Choice:**

Creating a learning model using the most appropriate features is not an easy task. The

hypothesized learning model's performance may not be accurate enough. However by continuous analysis of various models, we can find the appropriate learning model to be trained as our classifier.

- **Training:**

The process of using the input data in order to find the best classifier, using a set of features, is known as *training* of the classifier. Different methods can be applied for training the learning model. The more trained the model is, the better classifier it becomes.

- **Evaluation:**

In this final step of the classifier design cycle, we evaluate the classification performance of the selected model. In case the performance or classification accuracy is not desirable, then the entire process of the design cycle is repeated, starting with feature selection. Evaluation of a model becomes simpler, if prior knowledge about the input data is available and aides in selecting the correct set of features and hence the optimum learning model.

The input patterns, sometimes, cannot be separated easily using a definable boundary in the linear input space. It may happen that the boundary is too complex, thus resulting in overfitting of data, thereby leading to poor generalization of the classifier. The aim here is to build a classifier such that, it is able to segregate between classes, given any label and any input pattern. This might be possible on an abstract level, but may not be on a practical one. However we can move closer to building a good generalized classifier model using various concepts such as Kernels, Support vector machines, Neural Networks, Fuzzy logic, Genetic algorithms, etc. As evident from Figure 7.2, any intelligent method which involves using information from training samples, in the design of a classifier, employs *learning*.

7.3 Types of Machine Learning techniques

Machine learning can be briefly categorized into four types [8, 1]:

- **Supervised Learning**

In this type of learning method, there exists a human user or teacher, who assigns category labels for classification, along with the input patterns as the training set. The system thus learns from the given labels and the corresponding inputs, to classify the given data effectively.

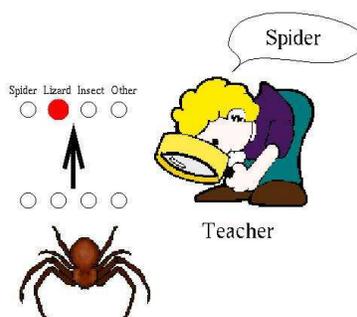


Figure 7.3: Supervised Learning [5]

- **Unsupervised Learning**

Unlike Supervised learning method, this technique of machine learning does not involve an external user or teacher, from whom the category labels can be taken, and we have only the input data. The aim here is to find the regularities in the input. The learning system takes the entire input set, and separates it into various clusters or natural groupings based on algorithms such as SOMs (Self Organizing Maps). This learning methodology essentially involves *density estimation*.

- **Semi-Supervised Learning**

In the semi-supervised learning method, the features of the input sets or the output labels are only partially known. This learning method involves training large amounts of unlabeled data, together with labeled data, in order to build better classifiers. An example of this learning technique would be k-means clustering.

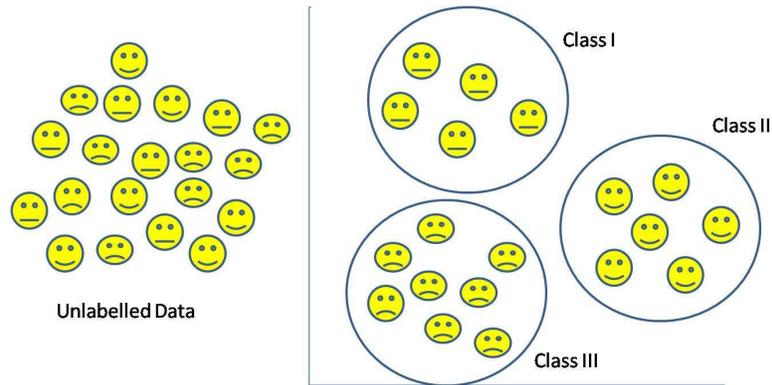


Figure 7.4: Unsupervised Learning

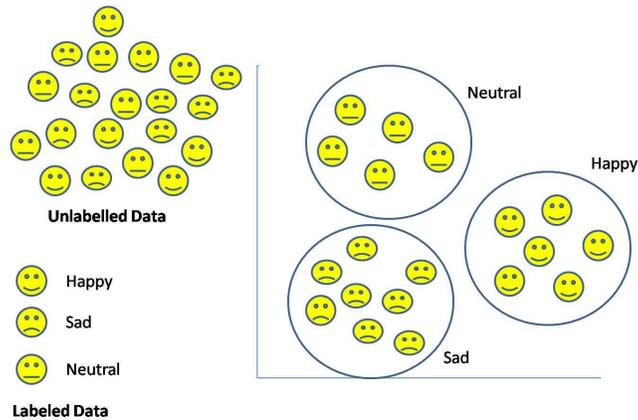


Figure 7.5: Semi-Supervised Learning

- **Reinforcement Learning**

In this type of machine learning technique, the external user does not provide any category label to the classifier, but instead makes a comment on the resulting output which the system assigns to the input data. The comment is usually like a right or wrong statement, i.e. the external teacher will simply say if the system has mapped the input to the correct label or not. This is similar to semi-supervised learning, where the intelligent system forms natural groupings or clusters, using some characteristic parameters for the input data, and the external teacher simply provides a hint on

whether the classification done was right or wrong.

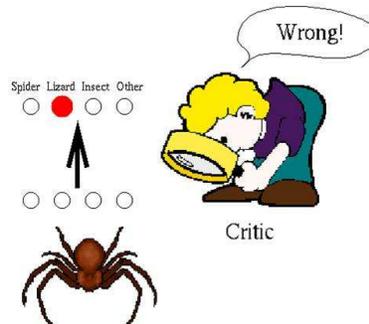


Figure 7.6: Reinforcement Learning [5]

Modelling learning algorithms for pattern classification and regression, sometimes gives us the desired results as opposed to direct programming techniques, wherein the computing system doesn't learn from experience. Also the learning methodology gives us better insights about the behavior of the complex system in focus.

Chapter 8

Appendix B

8.1 Expectation Maximization Algorithm

The Expectation-Maximization algorithm is used for estimating the likelihood parameters of certain probabilistic models. The algorithm was first generalized and compiled by Dempster *et.al* [47]. It is basically an iterative procedure which is followed in two steps. Assume that the likelihood function of a sample in an input vector is $C(\theta | X, Z)$ where θ is the parameter vector, X is the observed sample value and Z is an empty array for missing or unobserved values. We then find the maximum likelihood estimate using the marginal likelihood of the observed data X .

- Calculation of an Expectation E, of the log-likelihood using the initial estimates of the probabilistic model defining parameters as shown.

$$T(\theta | \theta^{(t)}) = E_{Z|x, \theta^{(t)}}[\ln C(\theta | X, Z)] \quad (8.1)$$

- Calculation of the Maximization M, which finds the parameters that maximize the value of E from first step as shown in the next equation. The maximizing parameters are then used for assessing the distribution of variables in the following expectation step.

$$\theta^{(t+1)} = \arg \max_{\theta} T(\theta | \theta^{(t)}) \quad (8.2)$$

For more detailed information on the Expectation-Maximization algorithm, refer to [48, 47]

8.2 Sequential Minimal Optimization

The SMO method is highly useful for large optimization problems, because the scaling factor with training set size is better. SMO basically involves breaking down of a complex quadratic optimization problem into smaller problems for analysis. Instead of using the conventional numerical quadratic programming (QP) as an inner loop, this method uses an analytic QP step. The overall QP problem is decomposed into smaller sub-QP problems using Osuna's theorem, in order to ensure convergence.

The algorithm then chooses to solve the smallest QP optimization problem at every step. In the standard QP problem, the smallest possible QP problem involves two Lagrange multipliers, since the multipliers must obey a linear equality constraint. At every step, two multipliers are chosen to jointly optimize and update the SVM. This reduces the computational time and the required matrix storage memory, thus increasing the efficiency of the learning machine. For a better understanding of the SMO method, refer to [45, 44].

In our study Fisher kernels, support vector machines, the expectation-maximization algorithm and sequential minimal optimization have been implemented using various functions and toolboxes in MATLAB software [90].

8.3 Linear Discriminant Analysis (LDA)

The main objective of Linear Discriminant Functions is to separate the given classes clearly, and the learning methodology that does this is known as Linear Discriminant Analysis (LDA). The LDA approach creates a discriminant function which maps the inputs to correct classification labels [1, 91]. A discriminant function that is a linear combination of the components of x can be written as

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \tag{8.3}$$

where \mathbf{w} is the weight vector and w_0 is the bias. In case of binary classification the linear classifier has the following decision rule: Decide w_1 if $g(\mathbf{x}) > 0$ and w_2 if $g(\mathbf{x}) < 0$. Thus \mathbf{x} is assigned to w_1 if the inner product $\mathbf{w}^T \mathbf{x}$ exceeds the threshold w_0 , and decide w_2 otherwise. If $g(\mathbf{x}) = 0$, \mathbf{x} can ordinarily be assigned to either class. The relation $g(\mathbf{x}) = 0$ defines a decision surface that separates points assigned to w_1 from points assigned to w_2 . When $g(x)$ is linear, this decision surface is a hyperplane. If \mathbf{x}_1 and \mathbf{x}_2 are both on the decision surface, then

$$\mathbf{w}^T(\mathbf{x}_1 - \mathbf{x}_2) = 0 \quad (8.4)$$

and this proves that \mathbf{w} is normal to any vector lying in the hyperplane.

In general, linear discriminant analysis separates the feature space by a hyperplane decision surface. The orientation of the surface is determined by the normal vector, and the location of the surface is determined by the bias. The discriminant function $g(\mathbf{x})$ is proportional to the signed distance from \mathbf{x} to the hyperplane, with $g(\mathbf{x}) > 0$ when \mathbf{x} is on the positive side, and $g(\mathbf{x}) < 0$ when \mathbf{x} is on the negative side.

In a two class linearly separable scenario, there is a set of n input samples $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ with labels w_1 and w_2 . This is being done to determine the weights \mathbf{w} in a linear discriminant function $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$.

A sampled \mathbf{x}_i is classified correctly if $\mathbf{w}\mathbf{x}_i + w_0 > 0$ and \mathbf{x}_i is labeled w_1 , or if $\mathbf{w}\mathbf{x}_i + w_0 < 0$ and \mathbf{x}_i is labeled w_2 . From this it can be concluded that in the binary classification case all samples of w_2 can be replaced by their negatives, and therefore the class label can be eliminated, in which $\mathbf{w}\mathbf{y}_i > 0$ for all of the samples. \mathbf{y}_i is the normalized sample. This is often called the solution or normalization sample. In order to find the solution to the set of inequalities, a criterion function has to be minimized [1].

The main objective of the LDA approach is to create a transformation matrix W which maximizes the ratio of between-class scatter to within-class scatter. A within-class scatter matrix S_w is defined as:

$$S_w = \sum_{i=1}^c \sum_{x \in C_i} (x - m_i)(x - m_i)^t \quad (8.5)$$

where c is the number of classes, C_i is a set of data belonging to the i^{th} class, and m_i is the mean of the i^{th} class. The within-class scatter matrix represents the degree of scatter within classes as a summation of covariance matrices of all classes. A between-class scatter matrix S_B is defined as:

$$S_B = \sum_{i=1}^c n_i (m_i - m)(m_i - m)^t \quad (8.6)$$

The LDA function generated in order to maximize the separation would be given as,

$$J(W) = W^t S_B W \frac{1}{W^t S_W W} \quad (8.7)$$

In our study we have employed Statistical Package for Social Science (SPSSTM) Software, for performing Linear Discriminant Analysis on the Fisher score data [92]. We have used LDA in our experiments as way of validating the correctness of our results obtained with SVM-Fisher kernel analysis.

8.4 Cross-validation techniques

Machine learning methodology is mainly focussed on building good classifiers. The error incurred by these classifiers over an entire input dataset, is defined as *Generalization Error (GE)*. However, in practice, a complete input dataset is not available every time, and hence we cannot compute the GE. Also we can't say exactly how the learning machine will respond or give predictions using input data it has not seen. In order to overcome this problem we try to use only a part of the input as training set. Some of the input data is kept aside for evaluating the machine's performance. This technique of evaluation is known as *cross-validation* [1, 93].

Cross-validation techniques are usually applied for assessing the approximate GE value for a given input data, and hence monitor how the learning model would generalize to an independent dataset. These techniques are usually applied to predicting or forecasting systems, and help in estimating how good the prediction capability of the system is. Cross-validation involves:

1. Partitioning the input dataset into *Training set*, and *Testing set* or *Validation set*.

2. Teaching the learning machine for analysis using the *training dataset*.
3. Validation of the analysis from previous step, using the *validation set*.

Assume that we have a learning model M , characterized by one or more unknown parameters. The input dataset has been split into training and validation sets. During the first stage of the model learning, we apply a fitting process which tunes the model parameters to the training data, so that it is mapped correctly to the given outputs or category labels. In order to check the fitting or generalization of the learning model, we apply the validation set, and check how well the estimated model parameters fit to this set.

Usually, in case of validation sets, fitting would not be as accurate as the training set, thus giving us the generalization error. But a correct estimation of the model parameters, and hence better training, can result in low GE from validation, and create better learning models. For further information on cross validation techniques, refer to [1]. Following sections 8.4.1 and 8.4.2 discuss two methods of cross-validation applied in our study.

8.4.1 Leave-one-out technique (LOO)

In this method of cross-validation, we split the input dataset such that, we get the first input sample to be the validation set, and the remaining input samples to be the training set. Using this training set, we train the learning model, thus obtaining the parameters' estimates. Then the learnt model is tested using the validation sample for checking generalization. This process continues until each of the input sample has been considered as a validation set. Although the computational constraints are higher in this method considering the large number of times training is done, the end results prove to be effective and accurate. In our experiments we have used the Leave-one-out technique as part of the LDA analysis, for validating the accuracy of classifying the original grouped cases.

8.4.2 Hold-out technique

This method is one of the simplest kind of cross validations [94]. The input dataset is separated into two sets : training and validation. Both the sets are made of randomly

selected samples from the input set. The hypotheses or the parameter function for the learning model is chosen by using the training set only. Then the trained model is asked to predict the outputs for the validation data. The errors collected from this process, are used for calculating the mean absolute error, which is further applied for evaluating the model's performance. The basic advantage of using this method is that, it takes much less computational time as compared to the Leave-one-out method. However, we can encounter high variance in outputs. The performance evaluation of the model heavily depends on which data points are in training set and validation set. Therefore, the performance of the learning machine may vary, depending on the sample selection in training and validation sets. The hold-out method has been applied in our study as part of designing the SVM classifier. Using this method, we randomly select training and testing (or validation) data (in the form of Fisher scores) and use them to design the SVM classifier.

List of Acronyms

3-D - Three Dimensional

6-D - Six Dimensional

EM - Expectation Maximization algorithm

LDA - Linear Discriminant Analysis

LOO - Leave-One-Out

NASDAQ - National Association of Securities Dealers Automated Quotations

NYSE - New York Stock Exchange

OA - Osteoarthritis

RBF - Radial Basis Function

SMO - Sequential Minimal Optimization

SPSS - Statistical Package for the Social Science

SVM - Support Vector Machines

TSX - Toronto Stock Exchange

VAG - Vibroarthrogram

VITA

PUBLICATIONS & PRESENTATIONS:

1. **IEEE SMC Conference 2009, San Antonio, USA** - Identifying the Potential for failure of businesses in Technology, Banking and Pharmaceutical sectors using kernel-based machine learning methods.
2. **International Congress on Environmental Modelling and Software, Ottawa, 2010** - Analysis of the Economic Sustainability of Companies in the Water Sector
3. **IEEE Transactions on Neural Networks** - Predicting the fate of businesses in the stock market using Fisher kernels and support vector machines (SVMs) - **Submitted for review.**
4. **Journal of Medical Engineering Physics** - Pattern classification of Knee-joint signals for predicting cartilage degeneration using Fisher kernels and support vector machines - **Submitted for review.**

Bibliography

- [1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.
- [2] G. Raghava, “A svm-based server for rice blast prediction dedicated to the farming community.” [Online]. Available: <http://www.imtech.res.in/raghava/rbpred/home.html>
- [3] [Online]. Available: <http://santamonicaarthritis.com/index.php?page=osteoarthritis>
- [4] “Robotic arm enables precise resurfacing of osteoarthritis knee.” [Online]. Available: <http://www.elements4health.com/robotic-arm-enables-precise-resurfacing-of-osteoarthritis-knee.html>
- [5] S. Dennis, “Introduction to neural networks,” 1997. [Online]. Available: <http://www.itee.uq.edu.au/cogs2010/cmc/chapters/Introduction/>
- [6] D. Braha, A. A. Minai, and Y. Bar-Yam, *Complex Engineered Systems: Science Meets Technology (Understanding Complex Systems)*. Springer, July 2006.
- [7] B. Pavard and J. Dugdale, “An introduction to complexity in social science,” *GRIC-IRIT*, 2000.
- [8] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines : and other kernel-based learning methods*. Cambridge University Press, March 2000.
- [9] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, June 2004.
- [10] N. Gershenfeld, *The Nature of Mathematical Modeling*. Cambridge University Press, November 1998.
- [11] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.
- [12] “Electronic statistics textbook.” [Online]. Available: <http://www.statsoft.com/textbook/>
- [13] “New york stock exchange.” [Online]. Available: <http://www.nyse.com/>

- [14] “Toronto stock exchange.” [Online]. Available: <http://www.tmx.com/>
- [15] “National association of securities dealers automated quotations.” [Online]. Available: <http://www.nasdaq.com/>
- [16] N. N. Taleb, *The Black Swan: The Impact of the Highly Improbable*. Random House, 2007.
- [17] A. Guergachi and G. Boskovic, “System models or learning machines” *Applied Mathematics and Computation*, vol. 204, no. 2, pp. 553 – 567, 2008, special Issue on New Approaches in Dynamic Optimization to Assessment of Economic and Environmental Systems.
- [18] Y. Sai, C. Zhong, and P. Nie, “A hybrid rst and ga-bp model for chinese listed company bankruptcy prediction,” in *ICNC '07: Proceedings of the Third International Conference on Natural Computation*. IEEE Computer Society, 2007, pp. 513–517.
- [19] H. Fu-yuan, “A genetic fuzzy neural network for bankruptcy prediction in chinese corporations,” in *Risk Management Engineering Management, 2008. ICRMEM '08. International Conference on*, Nov. 2008.
- [20] A. Abraham, N. S. Philip, and P. Saratchandran, “Modeling chaotic behavior of stock indices using intelligent paradigms,” in *International Journal of Neural, Parallel Scientific Computations, USA, Volume 11, Issue*, 2003, pp. 143–160.
- [21] X.-F. Hui and J. Sun, “An application of support vector machine to companies’ financial distress prediction,” in *Modeling Decisions for Artificial Intelligence*, 2006, pp. 274–282.
- [22] Z. Gao, M. Cui, and L.-M. Po, “Enterprise bankruptcy prediction using noisy-tolerant support vector machine,” *Future Information Technology and Management Engineering, International Seminar on*, vol. 0, pp. 153–156, 2008.
- [23] F. Tay, “Application of support vector machines in financial time series forecasting,” *Omega*, vol. 29, no. 4, pp. 309–317, August 2001.
- [24] K. Shin, T. Lee, and H. Kim, “An application of support vector machines in bankruptcy prediction model,” *Expert Systems with Applications*, vol. 28, no. 1, pp. 127–135, January 2005.
- [25] W. Hrdle, R. A. Moro, and D. Schfer, “Predicting bankruptcy with support vector machines,” Sonderforschungsbereich 649, Humboldt University, Berlin, Germany, SFB 649 Discussion Papers SFB649DP2005-009, Mar. 2005.
- [26] I. Nakaoka, K. Tani, Y. Hoshino, and K. Kamei, “A bankruptcy prediction method based on cash flow using som,” in *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, vol. 4, Oct. 2006.

- [27] A. Chan, W. Ng, D. Yeung, E. Tsang, and M. Firth, “Bankruptcy prediction using multiple classifier system with mutual information feature grouping,” in *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, Oct. 2006.
- [28] D. Yeung, W. Ng, A. Chan, P. Chan, M. Firth, and E. Tsang, “Bankruptcy prediction using multiple intelligent agent system via a localized generalization error approach,” in *Service Systems and Service Management, 2007 International Conference on*, June 2007.
- [29] B. Tang, W. Qiu, and X. Sun, “Business failure prediction using exponential smoothing forecasting and pattern recognition,” *Risk Management Engineering Management, 2008 International Conference on*, pp. 576–581, 2008.
- [30] S. Ross, R. Westerfield, B. Jordan, and G. Roberts, *Fundamentals of Corporate Finance, Sixth Canadian Edition*. McGraw-Hill Ryerson Higher Education, February 2007.
- [31] V. N. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.
- [32] A. Guergachi, L. Halachev, Y. Athavale, and S. Krishnan, “Analysis of the economic sustainability of companies in the water sector,” *International Congress on Environmental Modelling and Software, Ottawa*, 2010.
- [33] N. Smith and M. Gales, “Using svms and discriminative models for speech recognition,” in *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, 2002, pp. I–80–I–77 vol.1.
- [34] T. Jaakkola and D. Haussler, “Exploiting generative models in discriminative classifiers,” in *In Advances in Neural Information Processing Systems 11*, vol. 11, 1998, pp. 487–493.
- [35] K. Tsuda, S. Akaho, M. Kawanabe, and K.-R. Mller, “Asymptotic properties of the fisher kernel,” *Neural Computation*, vol. 16, p. 2004, 2003.
- [36] T. Jaakkola, M. Diekhaus, and D. Haussler, “Using the fisher kernel method to detect remote protein homologies.” *7th Intell. Sys. Mol. Biol.*, pp. 149–158, 1999. [Online]. Available: citeseer.ist.psu.edu/jaakkola99using.html
- [37] C. Saunders, J. Shawe-taylor, and A. Vinokourov, “String kernels, fisher kernels and finite state automata,” in *in Advances in Neural Information Processing Systems 15*, vol. 15, 2003, pp. 633–640.
- [38] N. Smith and M. Niranjan, “Data-dependent kernels in svm classification of speech patterns,” 2001.
- [39] N. Smith and M. Gales, “Speech recognition using svms,” in *Advances in Neural Information Processing Systems 14*. MIT Press, 2002, pp. 1197–1204.

- [40] P. Moreno and R. Rifkin, "Using the fisher kernel method for web audio classification," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, vol. 6, 2000, pp. 2417–2420 vol.4.
- [41] A. Holub, M. Welling, and P. Perona, "Combining generative models and fisher kernels for object recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, Oct. 2005, pp. 136–143 Vol. 1.
- [42] J. Chen and Y. Wang, "Exploiting fisher kernels in decoding severely noisy document images," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 1, Sept. 2007, pp. 417–421.
- [43] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.
- [44] M. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, no. 4, pp. 18–28, Jul/Aug 1998.
- [45] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.
- [46] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," 2005.
- [47] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977. [Online]. Available: <http://dx.doi.org/10.2307/2984875>
- [48] F. Dellart, "The expectation maximization algorithm," College of Computing, Georgia Institute of Technology, Tech. Rep. GIT-GVU-02-20, Feb 2002.
- [49] D. Zhang and L. Zhou, "Discovering golden nuggets: data mining in financial application," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 4, pp. 513–522, nov. 2004.
- [50] J. Hansen and R. Nelson, "Neural networks and traditional time series methods: a synergistic combination in state economic forecasts," *Neural Networks, IEEE Transactions on*, vol. 8, no. 4, pp. 863–873, jul 1997.
- [51] L. Cao and F. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *Neural Networks, IEEE Transactions on*, vol. 14, no. 6, pp. 1506–1518, nov. 2003.

- [52] T. Van Gestel, J. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle, “Financial time series prediction using least squares support vector machines within the evidence framework,” *Neural Networks, IEEE Transactions on*, vol. 12, no. 4, pp. 809–821, jul 2001.
- [53] Y. Athavale, A. Guergachi, S. Krishnan, and P. Hosseinizadeh, “Identifying the potential for failure of businesses in the technology, pharmaceutical and banking sectors using kernel-based machine learning methods bibtex,” in *Systems, Man and Cybernetics, 2009. SMC '09. IEEE International Conference on*, Oct. 2009.
- [54] F. Black and M. Scholes, “The pricing of options and corporate liabilities,” *The Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973.
- [55] Brorsen, B. Wade, and S.-R. Yang, “Nonlinear dynamics and the distribution of daily stock index returns,” *Journal of Financial Research*, vol. 17, no. 2, pp. 187–203, 1994.
- [56] M. Staunton, “Deriving black-scholes from lognormal asset returns,” *Willmott*, Feb 2002.
- [57] E. Nic, “Osteoarthritis: national clinical guideline for the care and management of osteoarthritis in adults (draft for consultation),” London, July 2007.
- [58] R. Rangayyan, S. Krishnan, and W. Yungfeng, “Computer-aided diagnosis of knee-joint disorders via vibroarthrographic signal analysis: A review,” *Critical Reviews in Biomedical Engineering*, 2009.
- [59] I. Boniatis, L. Costaridou, D. Cavouras, I. Kalatzis, E. Panagiotopoulos, and G. Panayiotakis, “Assessing hip osteoarthritis severity utilizing a probabilistic neural network based classification scheme,” *Med. Eng. Phys.*, vol. 29, no. 2, pp. 227–237, 2007.
- [60] J. S. Gregory, R. M. Junold, P. E. Undrill, and R. M. Aspden, “Analysis of trabecular bone structure using fourier transforms and neural networks,” *IEEE Trans Inf Technol Biomed*, vol. 3, no. 4, pp. 289–294, 1999.
- [61] W. L. Wu and F. C. Su, “Potential of the back propagation neural network in the assessment of gait patterns in ankle arthrodesis,” *Clinical Biomechanics*, vol. 15, no. 2, pp. 143–145, February 2000.
- [62] F. Su and W. Wu, “Design and testing of a genetic algorithm neural network in the assessment of gait patterns,” *Medical Engineering and Physics*, vol. 22, January 2000.
- [63] J. Duryea, S. Zaim, and F. Wolfe, “Neural network based automated algorithm to identify joint locations on hand/wrist radiographs for arthritis assessment,” *Medical Physics*, vol. 29, no. 3, pp. 403–411, 2002.

- [64] P. Levinger, D. Lai, K. Webster, R. Begg, and J. Feller, "Support vector machines for detecting recovery from knee replacement surgery using quantitative gait measures," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, Aug. 2007, pp. 4875–4878.
- [65] D. T. H. Lai, P. Levinger, R. K. Begg, W. L. Gilleard, and M. Palaniswami, "Automatic recognition of gait patterns exhibiting patellofemoral pain syndrome using a support vector machine approach," *Trans. Info. Tech. Biomed.*, vol. 13, no. 5, pp. 810–817, 2009.
- [66] K. Deluzio and J. Astephen, "Biomechanical features of gait waveform data associated with knee osteoarthritis: An application of principal component analysis," *Gait Posture*, vol. 25, no. 1, pp. 86 – 93, 2007.
- [67] R. M. Rangayyan and Y. F. Wu, "Screening of knee-joint vibroarthrographic signals using statistical parameters and radial basis functions," *Medical & Biological Engineering & Computing*, vol. 46, no. 3, pp. 223–232, 2008.
- [68] Y. F. Wu and S. Krishnan, "An adaptive classifier fusion method for analysis of knee-joint vibroarthrographic signals," in *Proc. 2009 Int'l Conf. Computational Intelligence for Measurement Systems and Applications (CIMSA '09)*, Hong Kong, May 2009, pp. 190–193.
- [69] R. Rangayyan and Y. Wu, "Modeling and classification of knee-joint vibroarthrographic signals using probability density functions estimated with Parzen windows," in *Proc. 30th Annual Int'l Conf. IEEE EMBS (EMBC'08)*, Vancouver, BC, Canada, Aug. 2008, pp. 2099–2102.
- [70] S. Krishnan, R. Rangayyan, G. Bell, and C. Frank, "Adaptive time-frequency analysis of knee joint vibroarthrographic signals for noninvasive screening of articular cartilage pathology," *Biomedical Engineering, IEEE Transactions on*, vol. 47, no. 6, pp. 773–783, June 2000.
- [71] R. Rangayyan, S. Krishnan, G. Bell, C. Frank, and K. Ladly, "Parametric representation and screening of knee joint vibroarthrographic signals," *Biomedical Engineering, IEEE Transactions on*, vol. 44, no. 11, pp. 1068–1074, Nov. 1997.
- [72] "Microsoft office productivity suite." [Online]. Available: <http://office.microsoft.com/excel>
- [73] O. Aran and L. Akarun, "Recognizing two handed gestures with generative, discriminative and ensemble methods via fisher kernels," in *MRCS*, 2006, pp. 159–166.
- [74] D. Brezeale and D. Cook, "Automatic video classification: A survey of the literature," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 3, pp. 416 –430, May 2008.

- [75] J. Ruiz-del Solar and P. Navarrete, "Eigenspace-based face recognition: a comparative study of different approaches," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 35, no. 3, pp. 315–325, Aug. 2005.
- [76] M. Rodriguez-Damian, E. Cernadas, A. Formella, M. Fernandez-Delgado, and P. D. Sa-Otero, "Automatic detection and classification of grains of pollen based on shape and texture," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 36, no. 4, pp. 531–542, July 2006.
- [77] N. Oza, J. Castle, and J. Stutz, "Classification of aeronautics system health and safety documents," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 6, pp. 670–680, Nov 2009.
- [78] J. Chen, R. Wang, S. Yan, S. Shan, X. Chen, and W. Gao, "Enhancing human face detection by resampling examples through manifolds," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 37, no. 6, pp. 1017–1028, nov. 2007.
- [79] J. Luo, M. Namburu, K. R. Pattipati, L. Qiao, and S. Chigusa, "Integrated model-based and data-driven diagnosis of automotive antilock braking systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 2, pp. 321–336, march 2010.
- [80] C.-F. Juang, S.-H. Chiu, and S.-J. Shiu, "Fuzzy system learned through fuzzy clustering and support vector machine for human skin color segmentation," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 37, no. 6, pp. 1077–1087, nov. 2007.
- [81] T.-H. Cheng and C.-P. Wei, "A clustering-based approach for integrating document-category hierarchies," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 2, pp. 410–424, march 2008.
- [82] T.-H. Cheng and P.-H. Hu, "A data-driven approach to manage the length of stay for appendectomy patients," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 39, no. 6, pp. 1339–1347, nov. 2009.
- [83] S. Pang, D. Kim, and S. Y. Bang, "Face membership authentication using svm classification tree generated by membership-based lle data partition," *Neural Networks, IEEE Transactions on*, vol. 16, no. 2, pp. 436–446, march 2005.
- [84] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz, "Fusion of face and speech data for person identity verification," *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1065–1074, sep 1999.

- [85] S. Li, J.-Y. Kwok, I.-H. Tsang, and Y. Wang, “Fusing images with different focuses using support vector machines,” *Neural Networks, IEEE Transactions on*, vol. 15, no. 6, pp. 1555–1561, nov. 2004.
- [86] H. Drucker, D. Wu, and V. Vapnik, “Support vector machines for spam categorization,” *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1048–1054, sep 1999.
- [87] O. Chapelle, P. Haffner, and V. Vapnik, “Support vector machines for histogram-based image classification,” *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1055–1064, sep 1999.
- [88] K. Labusch, E. Barth, and T. Martinetz, “Simple method for high-performance digit recognition based on sparse coding,” *Neural Networks, IEEE Transactions on*, vol. 19, no. 11, pp. 1985–1989, nov. 2008.
- [89] S. Gutta, J. Huang, P. Jonathon, and H. Wechsler, “Mixture of experts for classification of gender, ethnic origin, and pose of human faces,” *Neural Networks, IEEE Transactions on*, vol. 11, no. 4, pp. 948–960, jul 2000.
- [90] “Matrix laboratory.” [Online]. Available: <http://www.mathworks.com>
- [91] G. Balakrishnama, “Linear discriminant analysis - a brief tutorial,” 1998. [Online]. Available: <http://citeseer.ist.psu.edu/contextsummary/1048862/0>
- [92] “Statistical package for the social science.” [Online]. Available: <http://www.spss.com/>
- [93] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *International Joint Conference On Artificial Intelligence archive, Proceedings of the 14th international joint conference on Artificial intelligence*, vol. 2. Morgan Kaufmann, 1995, pp. 1137–1143.
- [94] J. Schneider and A. Moore, “A locally weighted learning tutorial using vizier 1.0,” 1997. [Online]. Available: <http://www.cs.cmu.edu/~schneide/tut5/node42.html>