1-1-2009

Predicting system collapse : application of kernel-based machine learning and inclination analysis

Pouyan Hosseinizadeh
Ryerson University

PREDICTING SYSTEM COLLAPSE: APPLICATION OF KERNEL-BASED MACHINE LEARNING AND INCLINATION ANALYSIS

By

Pouyan Hosseinizadeh
B. Eng, Azad Islamic University of Shiraz,
Iran, 2004

A thesis
presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the Program of
Mechanical Engineering

Toronto, Ontario, Canada, 2009

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

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

* Signature

I further authorise Ryerson University to reproduce this thesis or dissertation 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

Instruction on Borrowers

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

Abstract

Pouyan Hosseinizadeh

Predicting System Collapse:
Applications of Kernel-based Machine Learning
and
Inclination Analysis

MASc, Mechanical and Industrial Engineering Department,
Ryerson University, Toronto, 2009

While many modelling methods have been developed and introduced to predict the actual state of a system at the next point of time, the purpose of this research is to present and discuss two approaches NOT to predict the exact future states, but to identify the potential for final collapse of a system. The first approach is based on kernel methods, a sub category of supervised learning, and attempts to provide a visualization method to classify the active and dead companies and predict the potential collapse of a system. The second method aims to analyze the inclination of a system by looking at the local changes that have been observed over a certain period of time in the past. Applications of these modelling approaches to predict collapse in different companies belonging to two industrial sectors by looking at behaviour of their closing stock prices are discussed in this research. Advantages and limitations of each approach are also discussed.

Acknowledgement

I would like to formally thank Dr. Aziz Guergachi, my supervisor, for his hard work and guidance throughout this entire thesis process and for believing in my abilities. I have learned so much, and without him, this would not have been possible. Thank you so much for a great experience. Also, I thank NSERC, CFI, OIT and Ryerson University for funding this research.

I was delighted to interact with Dr. Venassa Magness for editing and co-writing of one of my papers submitted to SMC 2009 conference.

I would also like to thank Dr.Vikraman Baskaran for his help and instruction to prepare the presentation slides for my oral examination.

Last but not least, I especially want to appreciate Dr. Robert Roseberry for helping me with English writing throughout this thesis.

Dedication

I would like to dedicate this work to my parents, for their never-ending unconditional love and support in all my efforts, and for giving me the foundation to be who I am.

I would also like to dedicate it to Sara, my soul mate and confidant, for always being there for me. Thank you for your continual love, support, and patience as I went through this journey. I could not have made it through without you by my side.

Table of Contents

List of Tables

List of Figures

Nomenclature

Kernel-based Machine Learning

$\{Y_t\}$	time series
$\boldsymbol{x} \in X$	input and input space
$y \in Y$	output and output space
ω	weight vector perpendicular to the separating hyperplane
$\|\omega\|$	norm of ω
b	bias
m	margin
$\{x_i, y_i\}$	training pair
h	VC dimension
α_i	Lagrangian multiplier
W	dual Lagrangian
$sgn\,(x)$	equals 1, if $x \geq 0$ else -1
F	feature space
$\emptyset$	mapping to feature space
$K(x,z)$	kernel $\langle \varphi(x), \varphi(z) \rangle$
$\langle x,z \rangle$	inner product of x and z
M	generative model
θ	generative model parameters
x', X'	transpose of vector, matrix
D	training set
l	training set size
$f(\boldsymbol{x})$	real-valued function
m	dimension of input space
$\varphi(x_i)$	training point in feature space
$\mathcal{L}_\theta$	Likelihood
ξ	slack variables
$g(.,.)$	gradient vector
I_M	identity matrix
S_{FS_i}	sum of i^{th} coordinate of Fisher scores

Inclination Analysis

$+ or -$	local transitions toward survival or collapse
k	Period of time
m	fixed states length of the system
m^-/m^+	Collapse/survival Critical Level
n^-/n^+	No. Of minuses/pluses in a state
r^-/r^+	Probability of having minus/plus for the next transition
q	Probability of being at each of the states at $k = 0$
Z	transition probability matrix
P^-/P^+	system collapse/ survival probability
P_c^-/P_c^+	conditional system collapse/ survival probability
W	comparison index
TR	total rise of closing stock price during a certain time
TF	total fall of closing stock price during a certain time
L	observation length

Chapter 1

Introduction

1.1 What is a Complex System?

A system with numerous components and interconnections, interactions or interdependencies that are difficult to describe, understand, predict, manage, design, and/or change is called a complex system [1]. This definition applies to systems from a wide array of scientific disciplines. For example, ecosystems or economic systems, such as closing stock markets, can be pointed out as complex systems. Simulation and analysis of complex systems, as well as development of applications to understand and control such systems have always been a challenge.

There are three interrelated approaches to the modern study of complex systems:

- How interactions give rise to patterns of behaviour
- Understanding the ways of describing complex systems
- The process of formation of complex systems through pattern formation and evolution.

The main properties of complex systems are [2]:

- A complex system is fundamentally non-deterministic. It is impossible to anticipate precisely the behaviour of such systems even if the function of its constituents is completely known.

- A complex system has a dynamic structure. It is, therefore, difficult, if not impossible, to study its properties by decomposing it into functionally stable parts. Its permanent interaction with its environment and its properties of self-organization allow it to functionally restructure itself.
- The relationships that exist within the elements of a complex system are short-ranged and non-linear, and contain feedback loops (both positive and negative).
- A complex system comprises emergent properties that are not directly accessible (identifiable or anticipatory) from an understanding of its components.

1.2 Modelling Complex Systems

Making precise predictions about the future behaviour of complex systems, such as environmental systems, countries' economies or even big enterprises, has always been a challenge. As stated, these systems contain a large number of components interacting with each other and with other systems; thus, there are many factors that can affect them either directly or indirectly and alter their future behaviour. Some of these factors can be understood and the rest remain unknown. Unknown characteristics are those which make such systems very uncertain in such a way that their future states cannot be addressed easily.

To analyze the behaviour of a system in future, one needs to have knowledge about its past and current states, which usually can be obtained through the collection of data. These data can be considered as outputs of the system, which reflect its behaviour. As systems change over time, the data should be collected at various points of time to be representative of the condition of a system at that specific point. This type of data is called time series; dealing with time series data is inevitable in analyzing the systems' behaviour over time. Depending on the number and power of the factors that affect systems, variations of these time series and the levels of their uncertainty are different.

1.3 Traditional Modelling Techniques and Their Limitations

Traditionally, various forms of statistical models are used to deal with time series [3]. However, classical econometrics has a large literature for the time series forecast problem. The basic idea is as follows:

As a simplification, one may assume the time series follow a linear pattern. A linear regression predicting the target values from time to time, $y_t = a + b \times y_{t-1} + noise$, is found, where a and b are regression weights. The weights are then determined, given some form of noise. This method is rather crude but may work for deterministic linear trends. For nonlinear trends, one needs to use a nonlinear regression such as $y_t = a \times y_{t-1} + b \times y_{t-1}^2$. However, sometimes this nonlinear formulation is very hard to find. The econometrics method usually proposes a statistical model based on a set of parameters. The process of the estimation of these parameters brings in a human analyst bias. Another inherent problem of the statistical methods is that sometimes there are price movements that cannot be modelled with one single probability density function.

In [4], an experiment is conducted on crude oil data. The example shows there are four distributions that determine the difference between the logarithm of tomorrow's price and the logarithm of today's price rather than just one function. These four probability distributions must work together with each of them providing a contribution at different time points. Moreover, there are times when the best approximating function cannot be found. This can be caused by the rapid change in the data patterns. For different time windows, there are different patterns that cannot be described with one stationary model.

On the other hand, uncertainty is one of the most important issues that a modeller needs to deal with while working with time series. As the uncertainty of the data increases, the data patterns change and consequently the ability of classical modelling tools in predicting the future states of these systems highly decreases. Modelling approaches such as space states modelling or regression analysis methods cannot have a good performance in modelling and making prediction about such systems because, having a good understanding of the factors that affect the system is necessary in order to design the model in such a way that by having inputs, appropriate outputs can be obtained

from the model. However, many of the factors are unknown at the time of data collection, and the system is still changing while it is being modelled. Therefore, there needs to be another approach to deal with the high level of uncertainty in the data. This approach must either be able to extract almost all the factors affecting the system, and recognise the interactions inside it or basically be free from all these factors and interactions.

1.4 Concept of Time Series

A time series $\{Y_t\}$ is a discrete time, continuous state process where t $(t = 1, \ldots, T)$ are certain discrete time points. Usually time is taken at equally spaced intervals, and the time increment may be anything from seconds to years.

Direct statistical analysis of financial prices is difficult, because consecutive prices are highly correlated, and the variances of prices often change with time. This makes it usually more convenient to analyze changes in prices. Results for changes can easily be used to give appropriate results for prices.

In practice, financial models are influenced by time, both by time resolution and time horizon. The concept of resolution signifies how densely data are recorded. In applications in the finance industry, this might vary from seconds to years. For intra-daily, daily or weekly data, failure to account for the heavy-tailed characteristics of the financial time series will undoubtedly lead to an underestimation of portfolio Value-at-Risk (VaR). Hence, market risk analysis should consider heavy-tailed distributions of market returns.

Financial prices are determined by many political, corporate, and individual decisions. A model for prices is a detailed description of how successive prices are determined. A good model is capable of providing simulated prices that behave like real prices. Thus, it should describe the most important of the known properties of recorded prices. In this thesis a modelling approach based on statistical learning theory and kernel methods will be discussed that attempts to predict either collapse or survival of a company. Another modelling strategy called Inclination Analysis based on stochastic

processes will also be discussed that aims to make prediction about a potential collapse of a company.

1.5 Financial Time Series

Financial time series are a sequence of prices of some financial assets over a specific period of time. Daily news reports in newspapers, on television and radio inform people, for instance, of the latest closing stock market index values, currency exchange rates, and interest rates. It is often desirable to monitor price behaviour frequently and try to understand the probable development of the prices in the future. Private and corporate investors, business people, anyone involved in international trade and the brokers and analysts who advise these people can all benefit from a deeper understanding of price behaviour. Many traders deal with the risks associated with changes in prices.

There are two main objectives of investigating financial time series. First, it is important to understand how prices behave. The variance of the time series is particularly relevant. Tomorrow's price is uncertain and it must, therefore, be described by a probability distribution. This means that statistical methods are the natural way to investigate prices. Usually one builds a model, which is a detailed description of how successive prices are determined. The second objective is to use knowledge of price behaviour to make better decisions.

1.6 Applications of Financial Time Series Analysis

Analysis of financial time series is usually done for the following main applications:

- Prediction:

A financial time series model is a useful tool to generate forecasts for both the future value and the volatility of the time series. Moreover, it is important to have knowledge of the uncertainty of such forecasts.

- Risk management:

Financial risks often relate to negative developments in financial markets. Movements in financial variables such as interest rates and exchange rates create risks for most corporations.

Generally, financial risks are classified into the broad categories of market risks, credit risks, liquidity risks, and operational risks.

1.7 Objective and Organization of Thesis

The purpose of this thesis is to discuss two approaches for dealing with complex systems modelling. The goal, however, is not, for instance, to predict the next day closing stock price for a company. In contrast, the aim is to make a prediction about a potential for collapse /survival of an enterprise in the long run by having access to its closing stock prices over certain periods of time. In the first method, applications of statistical learning theory and kernel-based methods are used to predict the final fate i.e. survival or collapse, of a given company that belongs to a specific sector. The second method is called Inclination Analysis. This method enables an attempt to predict either collapse or survival of a given company using stochastic processes methods. Applications of these approaches as well as their advantages and limitations are discussed. The organization of this thesis is as follows:

Chapter 2 introduces the Machine Learning methodology, its goals, advantages and limitations along with learning models and important definitions. In Chapter 3, detailed aspects of the kernel-based methods and the so-called Fisher kernel as the tool that is used in this research for classification are studied. Application of Fisher kernel for visual inspection and data collection procedure are studied. Prediction results along with stress-test analysis are also discussed in this chapter. Another modelling approach called

Inclination Analysis that is used to predict the potential collapse for a system is discussed in Chapter 4. Designing of the model parameters, including data conversion procedure and observation windows extraction and finally results are included. In conclusion, discussion and directions for future studies that indicate possible improvements of the modelling approaches and a comparison between the two approaches are considered in Chapter 5.

Chapter 2

Machine Learning and its Disadvantages

2.1 What is Machine Learning?

Machine learning is the ability of a machine to improve its performance based on previous results over time. It makes use of some computer algorithms that help the machine to learn. One might, for instance, be interested in learning to make accurate predictions. The learning that is being done is always based on some sort of observations or data, such as examples, direct experience, or instruction. The emphasis on "machine" means that the leaning is an automatic process. In other words, the goal is to devise learning algorithms that do the learning automatically without human intervention or assistance. Although computers are applied to a wide range of tasks, and for most of them it is relatively easy for programmers to design and implement the necessary software, there are, however, many tasks for which this is difficult or impossible. These can be divided into four general categories:

- Problems for which there exist no human experts
- Problems where human experts exist, but where they are unable to explain their expertise
- Problems where phenomena are changing rapidly (e.g. closing stock price)
- Applications that need to be customised for each computer user separately

One often has a specific task in mind, such as prediction of bankruptcy of a firm in following years. However, rather than program the computer to solve the task directly, in machine learning, methods by which the computer will come up with its own program, based on provided examples are sought.

Machine learning is a core sub-area of *Artificial Intelligence*. It is very unlikely that we will be able to build any kind of intelligent system capable of any of the facilities that we associate with intelligence, such as language or vision, without using learning to get there. These tasks are otherwise simply too difficult to solve. Furthermore, we would not consider a system to be truly intelligent if it were incapable of learning since learning is at the core of intelligence. Machine learning also intersects broadly with other fields, especially statistics and mathematics.

2.2 Examples of Machine Learning Problems

There are many examples of machine learning problems. Much attention in this thesis will focus on classification tasks in which the goal is to categorise objects into a fixed set of categories.

Following are some examples of machine learning:

- optical character recognition [5]: categorizing images of handwritten characters by the letters represented
- face detection [6]: finding faces in images (or indicating if a face is present)
- spam filtering [7]: identifying email messages as spam or non-spam
- topic spotting: categorizing news articles as to whether they are about politics, sports, entertainment, etc.
- spoken language understanding [8]: within the context of a limited domain, determining the meaning of something uttered by a speaker to the extent that it can be classified into one of a fixed set of categories

- medical diagnosis [9]: diagnosing a patient as a sufferer or non-sufferer of some disease
- customer segmentation: predicting, for instance, which customers will respond to a particular promotion
- fraud detection [10]: identifying credit card transactions (for instance) which may be fraudulent in nature
- weather prediction: predicting, for instance, whether or not it will rain tomorrow

Although much discussion will be about classification problems, there are other important learning problems. In classification, objects are categorised into fixed categories. Regression, on the other hand, tries to predict a real value. For instance, one may wish to predict how much it will rain tomorrow; or, one might want to predict how much a house will sell for.

A richer learning scenario is one in which the goal is actually to behave intelligently, or to make intelligent decisions. For instance, a robot needs to learn to navigate through its environment without colliding with anything. To use machine learning to make money on the stock market, one might treat investment as a classification problem (will the stock go up or down) or a regression problem (how much will the stock go up), or, dispensing with these intermediate goals, one might want the computer to learn directly how to decide to make investments so as to maximise wealth.

2.3 Goals of Machine Learning Research

The primary goal of machine learning research is to develop general purpose algorithms of practical value. Such algorithms should be efficient. Learning algorithms should also be as general purpose as possible and easily applicable to a broad class of learning problems. Of course, the result of learning should be a prediction rule that is as accurate as possible in the predictions that it makes.

Occasionally, one may also be interested in the interpretability of the prediction rules produced by learning. In other words, the computer should find prediction rules that are easily understandable by human experts.

As mentioned above, machine learning can be thought of as programming by example. In brief, the main goal of machine learning research is to generalise and automate the prediction algorithm.

2.4 The Advantage of Machine Learning over Direct Programming

First, the results of using machine learning are often more accurate than what can be created through direct programming [11]. The reason is that machine learning algorithms are data driven, and are able to examine large amounts of data. On the other hand, a human expert is likely to be guided by imprecise impressions or perhaps an examination of only a relatively small number of examples.

Also, humans often have trouble expressing what they know, but have no difficulty labelling items. For instance, it is easy to label images of letters by the character represented, but one would have a great deal of trouble explaining in precise terms how to do it. Another reason to study machine learning is the hope that it will provide insights into the general phenomenon of learning. Some of the questions that might be answered include:

- What are the intrinsic properties of a given learning problem that make it hard or easy to solve?
- How much do you need to know ahead of time about what is being learned in order to be able to learn it effectively?
- Why are "simpler" hypotheses better?

2.5 Learning Models

To study machine learning mathematically, it is necessary to formally define the learning problem. This precise definition is called a learning model. A learning model should be rich enough to capture important aspects of real learning problems, but simple enough to study the problem mathematically. As with any mathematical model, simplifying assumptions are unavoidable.

A learning model should answer several questions:

- What is being learned?
- How are the data being generated? In other words, where do they come from?
- How are the data presented to the learner? For instance, does the learner see the entire data at once or only one example at a time?
- What is the goal of learning in this model?

In a learning problem the examples are the objects that are being classified. For instance, in spam filtering, the email messages are the examples. Usually, an example is described by a set of attributes, also known as features or variables. For instance, in medical diagnosis, a patient might be described by attributes such as gender, age, weight, blood pressure, body temperature, etc.

The label is the category that needs to be predicted. For instance, in spam filtering, the possible labels are "spam" and "not spam." During training, the learning algorithm is supplied with labelled examples, while during testing, only unlabeled examples are provided.

In some cases, it will be assumed that only two labels are possible, 0 and 1. The simplifying assumption that there is a mapping from examples to labels will also be made. This mapping is called a concept. Thus, a concept is a function of the form $f : X \rightarrow \{0,1\}$ where X is the space of all possible examples called the domain or instance space (training space). A collection of concepts is called a concept class. It will often be assumed that the examples have been labelled by an unknown concept from a known concept class.

2.6 Types of Learning

Based on the types of problem a machine learning method can solve, it can be placed under one of the following three major groups:

- Supervised learning
- Un-supervised Learning
- Semi-supervised Learning

The supervised learning algorithm attempts to learn the input-output relationship (dependency or function) by using a set of example called "training data set" including n pairs of (x_1, y_1), (x_2, y_2), $(x_3, y_3) \ldots (x_n, y_n)$ where inputs x_i are m-dimensional vectors $x_i \in R^m$ and y_i are discrete values for classification problems and continuous values for regression tasks which are called "data label." *Support Vector Machines* is a popular and seemingly the most reliable technique in this group of learning methods. Supervised learning itself is divided into two types of learning problem namely, Classification and Regression.

Un-supervised learning refers to the algorithms for which there are no data labels available and using the input values of x_i only, aims to find how the data are organised. The most popular algorithms under this group are called *Clustering* techniques and *Component Analysis* routines.

Semi-supervised learning is something between the other two learning algorithms, where the data labels are available only for a small portion of the data while most of the data is unlabelled. The reason for not having labels for all the data points is the cost or other difficulties in the process of obtaining labelled data points. As a result, the goal of a semi-supervised learning algorithm is to predict the labels of the unlabelled data by taking the entire data set into account [12]. In this thesis an attempt is made to use the application of a supervised learning algorithm in classification (specifically, Support Vector Machines and kernel-based methods) to design and train a model to be able to classify the enterprises which are more likely going to die, from those that are more likely to remain active in the market. In this research, the focus is on supervised learning

algorithms. Yet another modelling approach called Inclination Analysis that is based on stochastic processes is also discussed. Throughout this thesis, these modelling approaches are applied for classification task rather than regression. Modelling principles are described in Chapters 3 and 4 respectively.

2.7 Machine Learning Limitations and Inclination Analysis Approach

Despite the power and strong theory behind machine learning, it is still a statistical modelling approach. It means that, in order to train the model, there needs to be enough examples available. In case of speaker verification or hand writing recognition it is easy to have lots of examples of different speakers or hand writings. However, if a prediction needs to be made about the potential death of a lake, it is not easy, if not impossible, to gather enough examples of the lakes that are dead or active to train the model. Sometimes the cost of collecting enough data is such high that it is preferred to use other modelling tools.

Quality of the provided examples is also very important. Training phase in machine learning approaches is highly correlated to the quality of examples. Therefore, the amount of noises in the data used for training can be very deterministic in this approach.

In this research, application of Fisher kernel, one of the by-products of machine learning, for prediction is discussed. However, many other kernels can be selected and applied for classification. Choosing the best kernels for a certain application is very hard. Although there are many kernels available and many others can be made with respect to certain conditions, there is no evaluation method available to compare the performance of kernels in different applications. Therefore, choosing the best kernel is another drawback in this area.

In this thesis, another modelling approach is discussed that does not need a certain number of examples for prediction. Instead, it makes use of brief windows of the past

local changes of the same system to make prediction about its terminal state. This model is called Inclination Analysis and is based on stochastic processes. Application of this modelling approach in cases where there are not enough examples of system's input/output available is an appropriate choice.

Chapter 3

Kernel-based Machine Learning

3.1 Methodology

There are three key features for a pattern analysis algorithm to be recognised as an effective algorithm.

- Computational efficiency
- Robustness
- Statistical stability

As stated before, the concern in this research is to deal with financial time series classification in order to make a prediction about the fate of a company.

The classification problem can be restricted to consideration of the two-class problem without loss of generality. Assuming a set of data points, each one belonging to one of two classes and given a new data point, the object is to determine which class the new point belongs to. To carry out the training, one first needs to classify the available data points, i.e. examples, into two classes and to do so the best classifier among all separating hyperplanes must be chosen. There are plenty of possible linear classifiers that can separate the data that are linearly separable (Figure 3.1).

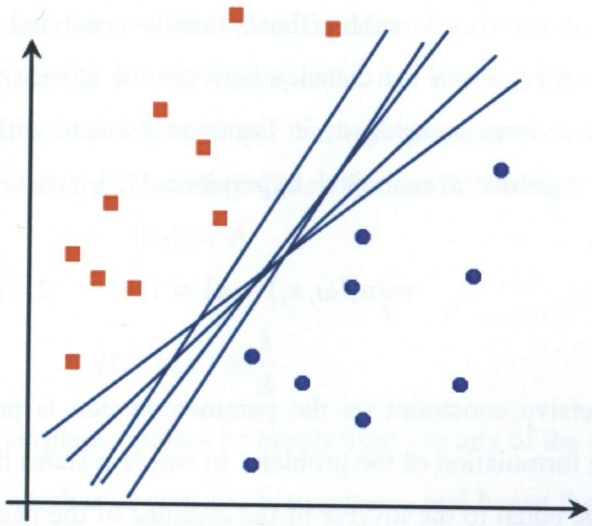


Figure 3.1 Possible decision boundaries to separate two classes of linearly separable data

However, only one classifier can separate the two classes the best. This line should be as far away from the data of both classes as possible. The distance between the linear classifier and the nearest data point of each class is called the margin. In other words, the margin needs to be maximised. In order to find the optimal separating hyperplane, an optimization problem needs to be solved with the objective function of margin maximization.

3.1.1 Optimal Linear Separating Hyperplane

Consider the problem of separating the set of training vectors belonging to two separate classes,

$$D = \{(x_1, y_1), \ldots, (x_l, y_l)\}, \quad x \in R^n, \qquad y \in \{-1, 1\} \tag{3.1}$$

with a hyperplane,

$$\langle \omega, x \rangle + b = 0 \tag{3.2}$$

The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vector to the hyperplane is maximal. There is some redundancy in Equation 3.2, and without loss of generality it is appropriate to consider a canonical hyperplane [13], where the parameters ω, b are constrained by,

$$\min_i |\langle \omega, x_i \rangle + b| = 1 \tag{3.3}$$

This incisive constraint on the parameterization is preferable to alternatives in simplifying the formulation of the problem. In words it states that: the norm of the weight vector should be equal to the inverse of the distance of the nearest point in the data set to the hyperplane. A separating hyperplane in canonical form must satisfy the following constraints,

$$y_i[\langle \omega, x_i \rangle + b] \geq 1, \; i = 1, \ldots, l \tag{3.4}$$

The distance $d(\omega, b; x)$ of a point x from the hyperplane (ω, b) is,

$$d(\omega, b; x) = \frac{|\langle \omega, x_i \rangle + b|}{\|\omega\|} \tag{3.5}$$

The optimal hyperplane is given by maximizing the margin, m, subject to the constraints of Equation 3.4. The margin is given by,

$$\begin{aligned} m(\omega, b) &= \min_{x_i : y_i = -1} d(\omega, b; x_i) + \min_{x_i : y_i = 1} d(\omega, b; x_i) \\ &= \min_{X_i : y_i = -1} \frac{|\langle \omega, x_i \rangle + b|}{\|\omega\|} + \min_{X_i : y_i = 1} \frac{|\langle \omega, x_i \rangle + b|}{\|\omega\|} \\ &= \frac{1}{\|\omega\|} \left(\min_{x_i : y_i = -1} |\langle \omega, x_i \rangle + b| + \min_{x_i : y_i = 1} |\langle \omega, x_i \rangle + b| \right) \\ &= \frac{2}{\|\omega\|} \end{aligned} \tag{3.6}$$

Hence the hyperplane that optimally separates the data is the one that minimises,

$$\emptyset(\omega) = \frac{1}{2} \|\omega\|^2 \tag{3.7}$$

It is independent of b because provided Equation 3.4 is satisfied (i.e. it is a separating hyperplane), changing b will move it in the normal direction to itself. Accordingly, the margin remains unchanged but the hyperplane is no longer optimal in that it will be nearer to one class than the other. Given that the following bound holds,

$$\|\omega\| < A \tag{3.8}$$

then from Equation 3.4 and 3.5,

$$d(\omega, b; x) \geq \frac{1}{A} \tag{3.9}$$

Accordingly, the hyperplanes cannot be nearer than $\frac{1}{A}$ to any of the data points and intuitively this reduces the number of possible hyperplanes, and hence the capacity. The VC dimension [10], h, of the set of canonical hyperplanes in n-dimensional space is bounded by,

$$h \leq min(R^2A^2, n) + 1 \tag{3.10}$$

where R is the radius of a hypersphere enclosing all the data points. Hence, minimising Equation 3.7 is equivalent to minimising an upper bound on the VC dimension. The solution to the optimization problem of Equation 3.7 under the constraints of Equation 3.4 is given by the saddle point of the Lagrange functional (Lagrangian) (Minoux, 1986),

$$\emptyset(\omega, b, \alpha) = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{l} \alpha_i (y_i[\langle \omega, x_i \rangle + b] - 1) \tag{3.11}$$

where α are the Lagrange multipliers. The Lagrangian has to be minimised with respect to ω, b and maximised with respect to $\alpha \geq 0$. Classical Lagrangian duality enables the primal problem, Equation 3.11, to be transformed to its dual problem, which is easier to solve. The dual problem is given by,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left(\min_{\omega, b} \emptyset(\omega, b, \alpha) \right) \tag{3.12}$$

The minimum with respect to ω and b of the Lagrangian, $\emptyset$, is given by,

$$\frac{\partial \emptyset}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{l} \alpha_i y_i = 0$$

$$\frac{\partial \emptyset}{\partial \omega} = 0 \quad \Rightarrow \quad \omega = \sum_{i=1}^{l} \alpha_i x_i y_i \tag{3.13}$$

Hence from Equations 3.11, 3.12 and 3.13, the dual problem is,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left(-\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^{l} \alpha_k \right) \tag{3.14}$$

and hence the solution to the problem is given by,

$$\alpha^* = \arg\min_{\alpha} \left(\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{k=1}^{l} \alpha_k \right) \tag{3.15}$$

with constraints,

$$\alpha_i \geq 0 \quad i = 1, \ldots, l$$

$$\sum_{j=1}^{l} \alpha_j y_j = 0. \tag{3.16}$$

Solving Equation 3.15 with constraints Equation 3.16 determines the Lagrange multipliers, and the optimal separating hyperplane is given by,

$$\omega^* = \sum_{i=1}^{l} \alpha_i y_i x_i$$

$$b^* = -\frac{1}{2} \langle \omega^*, x_r + x_s \rangle \tag{3.17}$$

where x_r and x_s are any support vectors from each class satisfying,

$$\alpha_r, \alpha_s > 0, \quad y_r = -1, y_s = 1 \tag{3.18}$$

The so-called maximal-margin decision boundary, i.e. hard classifier is then,

$$f(x) = sgn(\langle \omega^*, x \rangle + b) \tag{3.19}$$

The idea of maximal-margin decision boundary is depicted in Figure 3.2.

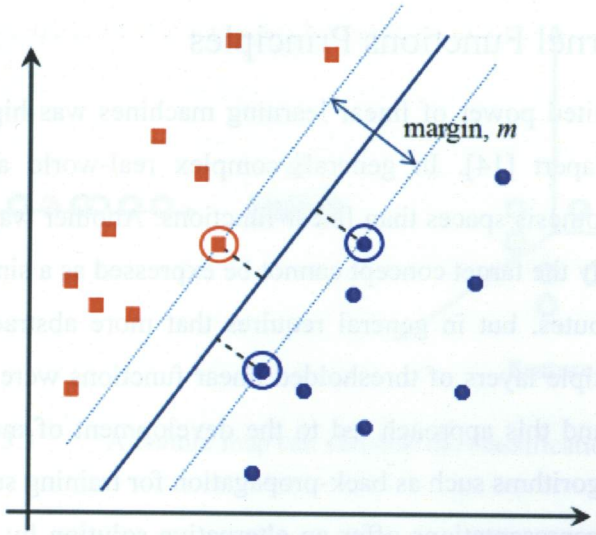


Figure 3.2 Maximal-margin Decision Boundary

Detecting linear relation has been the focus of much research in statistic and machine learning for a long time. However, when it comes to complex real-world systems, it is generally impossible to express the relations as simple linear combination of the given attributes. Kernel-based methods are of the effective approach for identifying patterns in a finite set of data because, this approach, first, transforms the data into a suitable so-called *feature space* and then uses algorithms based on linear algebra, geometry and statistics to discover patterns in the transformed data.

Support Vector Machines is recognised as one of the most powerful algorithms for pattern analysis and is out performed in different applications since its introduction in 1995. The kernel methods provide one of the main building blocks of Support Vector Machines (SVM). One of the remarkable features of SVM is that to a certain extent the approximation-theoretic issues are independent of the learning-theoretic ones. Therefore, one can study the properties of the kernel methods in a general and self-contained way, and use them with different learning theories. In this research, the main focus is on kernel based classification. For more details on SVM see Appendix A.

3.1.2 Kernel Functions Principles

The limited power of linear learning machines was highlighted in the 1960s by Minsky and Papert [14]. In general, complex real-world applications require more expressive hypothesis spaces than linear functions. Another way of viewing this problem is that frequently the target concept cannot be expressed as a simple linear combination of the given attributes, but in general requires that more abstract features of the data be exploited. Multiple layers of thresholded linear functions were proposed as a solution to this problem, and this approach led to the development of multi-layer neural networks and learning algorithms such as back-propagation for training such systems.

Kernel representations offer an alternative solution by projecting the data into a high dimensional feature space to increase the computational power of the linear learning machines. Another attraction of the kernel method is that the learning algorithms and theory can largely be decoupled from the specifics of the application area, which must simply be encoded into the design of an appropriate kernel function.

The fact that simply mapping the data into another space can greatly simplify the task has been known for a long time in machine learning, and has given rise to a number of techniques for selecting the best representation of data. The quantities introduced to describe the data are usually called features, while the original quantities are sometimes called attributes. The task of choosing the most suitable representation is known as feature selection. The space X is referred to as the input space, while $F = \{\emptyset(x): |x \in X\}$ is called the feature space.

Figure 3.3 shows an example of a feature mapping from a one-dimensional input space to a three-dimensional feature space, where the data cannot be separated by a linear function in the input space, but can be in the feature space. The aim of this chapter is to show how such mappings can be made into very high dimensional spaces where linear separation becomes much easier.

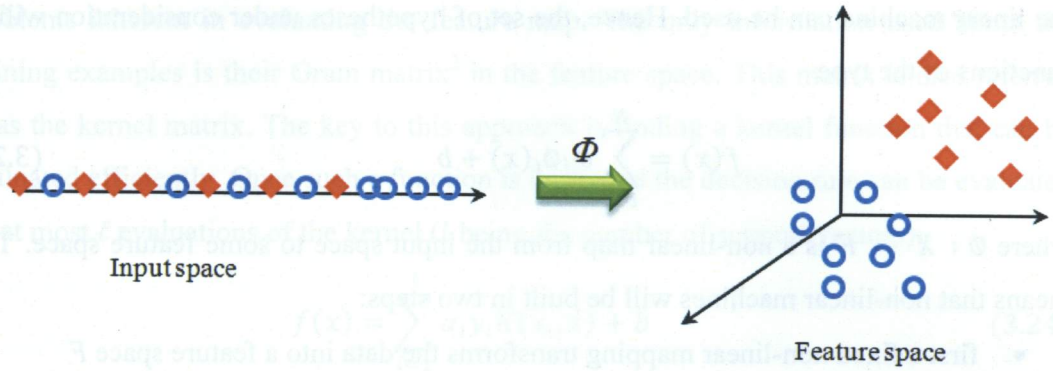


Figure 3.3 A feature map can simplify the classification task

Different approaches to feature selection exist. Frequently one seeks to identify the smallest set of features that still conveys the essential information contained in the original attributes. This is known as dimensionality reduction, i.e.

$$x = (x_1, \ldots, x_n) \rightarrow \emptyset(x) = \left(\emptyset_1(x), \ldots, \emptyset_d(x)\right), \qquad d < n \ , \tag{3.20}$$

and can be very beneficial as both computational and generalization performance can degrade as the number of features grows, a phenomenon sometimes referred to as the curse of dimensionality[1]. The difficulties with high dimensional feature spaces are unfortunate, since the larger the set of (possibly redundant) features, the more likely that the function to be learned can be represented using a standardised learning machine.

3.1.3 The Implicit Mapping into Feature Space

In order to learn non-linear relations with a linear machine, it is necessary to select a set of non-linear features and rewrite the data in the new representation. This is equivalent to applying a fixed non-linear mapping of the data to a feature space, in which

[1] The curse of dimensionality is the problem caused by the exponential increase in volume associated with adding extra dimensions to a (mathematical) space. The term was coined by Richard Bellman.

the linear machine can be used. Hence, the set of hypotheses under consideration will be functions of the type

$$f(x) = \sum_{i=1}^{N} \omega_i \emptyset_i(x) + b \tag{3.21}$$

where $\emptyset : X \rightarrow F$ is a non-linear map from the input space to some feature space. This means that non-linear machines will be built in two steps:

- first a fixed non-linear mapping transforms the data into a feature space F
- a linear machine is used to classify them in the feature space.

One important property of linear learning machines is that they can be expressed in a dual representation. This means that the hypothesis can be expressed as a linear combination of the training points, so that the decision rule can be evaluated using just inner products between the test point, x, and the training points, $\{x_i\}_{i=1}^{l}$,:

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i \langle \emptyset(x_i). \emptyset(x) \rangle + b \tag{3.22}$$

If a way of computing the inner product $\langle \emptyset(x_i). \emptyset(x) \rangle$ in feature space directly as a function of the original input points could be found, it becomes possible to merge the two steps needed to build a non-linear learning machine. This direct computation method is the so-called kernel function.

As mentioned earlier the kernel function K is defined such that for all $x_i, x_j \in X$

$$K(x_i, x_j) = \langle \emptyset(x_i). \emptyset(x_i) \rangle \tag{3.23}$$

where $\emptyset$ is a mapping from X to an (inner product) feature space F.

The name "kernel" is derived from integral operator theory, which underpins much of the theory of the relation between kernels and their corresponding feature spaces. An important consequence of the dual representation is that the dimension of the feature space need not affect the computation. As one does not represent the feature vectors explicitly, the number of operations required to compute the inner product by evaluating the kernel function is not necessarily proportional to the number of features. The use of kernels makes it possible to map the data implicitly into a feature space and to train a linear machine in such a space, potentially side-stepping the computational

problems inherent in evaluating the feature map. The only information used about the training examples is their Gram matrix[2] in the feature space. This matrix is also referred to as the kernel matrix. The key to this approach is finding a kernel function that can be evaluated efficiently. Once such a function is evaluated the decision rule can be evaluated by at most ℓ evaluations of the kernel (l being the number of support vectors):

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b \tag{3.24}$$

One of the facts about using a kernel is that it is not necessary to know the underlying feature map in order to be able to learn in the feature space. However, it must be a symmetric positive definite function, which satisfies Mercer's Conditions,

$$\iint K(x_i, x_j)\, g(x_i) g(x_j) dx_i dx_j > 0 \tag{3.25}$$

to represent a legitimate inner product in feature space.

In fact, the idea of a kernel generalises the standard inner product in the input space. It is clear that this inner product provides an example of a kernel by making the feature map the identity

$$K(x_i, x_j) = \langle x_i, x_j \rangle \tag{3.26}$$

3.1.4 Fisher Kernel

In this thesis the Fisher kernel [15] is explained since it has been shown to perform well in many applications and can process data that are not of the vector type [16]. Generative models such as Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM) [17] have been vastly used to model the observed data.

The Fisher kernel attempts to extract from a generative model more information than simply its output probability. The goal is to obtain internal representation of the data items within the probability model that describes the system's input to handle the variable

[2] In linear algebra, the Gram matrix of a set of vectors $v_1, \ldots, v_n$ in an inner product space is the symmetric matrix of inner products, whose entries are given by $G_{ij} = (v_i \mid v_j)$.

length data. The probability model needs to be parameterised so that derivatives of the model with respect to the parameters can be computed. Given a generative model which is a probability density function of input points x, say, $M = P(x, \theta)$, of data with parameters $\theta = \{\theta_i\}_{i=1}^{p}$ where p is the number of parameters, the probability of each data point x can be computed using the generative model M with the parameters θ. This probabilistic value with respect to the generative model M is defined to be the likelihood of that specific data point. So for an input space $X = \{x_j\}_{j=1}^{n}$ with n as the number of data points in a set of training data points, the model parameter θ can be learned by adapting all the points to maximise the likelihood of the training set. So basically one needs to maximise

$$\prod_{j=1}^{n} \mathcal{L}_\theta(x_j) \tag{3.27}$$

where $\mathcal{L}$ represents the likelihood of the training data. To get rid of the complicated calculations, in terms of dealing with several multiplications, in practice, the log-likelihood of the data points x_j with respect to the generative model is used in operation as follows:

$$log \prod_{j=1}^{n} \mathcal{L}_\theta(x_j) = \sum_{j=1}^{n} log\ \mathcal{L}_\theta(x_j) \tag{3.28}$$

Consider the vector gradient of the log-likelihood

$$g(\theta, x_j) = \left(\frac{\partial\, log\, \mathcal{L}_\theta(x_j)}{\partial \theta_1}, \frac{\partial\, log\, \mathcal{L}_\theta(x_j)}{\partial \theta_2}, \ldots, \frac{\partial\, log\, \mathcal{L}_\theta(x_j)}{\partial \theta_p} \right) \tag{3.29}$$

Hence for each data point x_j, $g(\theta, x_j)$ is defined as its *Fisher score* with respect to the generative model for the given set of parameters θ. The Fisher score gives an embedding into the feature space R^N and, therefore, immediately suggests a possible kernel, which is called the Fisher kernel and is represented as

$$K(x_i, x_j) = g(\theta, x_j)' \, I_M{}^{-1} g(\theta, x_i) \tag{3.30}$$

where I_M is called the *Fisher information matrix*, which is usually approximated by identity. So the practical Fisher kernel is defined as

$$K(x_i, x_j) = g(\theta, x_j)' \, g(\theta, x_i) \tag{3.31}$$

3.2 Application to System Collapse Prediction

In this thesis, an experiment is conducted for implicit mapping of the closing stock price time series of some companies belonging to two different industrial sectors to a feature space, where it is possible to make use of visualisation techniques to inspect the pattern behaviour of the associated companies. Then given a time series of a new company that belongs to the same sector, prediction of its potential collapse is made.

In brief, the experimental procedure includes the following steps:

1. Collection and preparation of financial time series, i.e. closing stock prices
2. Selection of a generative model and estimation of associated parameters
3. Calculation of Fisher scores for all data points
4. Plotting the summation over the Fisher scores pertaining to the mean parameters versus the summation over the Fisher scores pertaining to the variance parameters
5. Visual inspection and determination of possible clusters

3.2.1 Data Collection Process

Actual closing stock price data are collected using *Datastream* software. It is the world's largest and most respected financial statistical database. Datastream provides historical financial information with worldwide coverage. Key data sets include equities and company financials, stock market indices, unit and investment trusts, warrants, bonds, interest rates, exchange rates, commodities, and macroeconomic data sourced

from the IMF, OECD and government agencies. It is also possible to get time series data from Datastream. Time series coverage may vary for various countries and companies; however, Datastream covers up to 50 years of data. Available frequencies include daily, weekly, monthly, quarterly and annually. Table 3.1 indicates a list of criteria to be set for a typical search in Datastream.

Field Name	Description
Name	Name of a specific company(s)
Market	Country that the data are collected for
Base Date	Appearance date of a company in the Market
Currency	Currency of the prices
Data Category	Currently there are 18 data categories available
Status	Available categories are: Dead, Active, Suspended
Instrument Type	Basic types of financial instruments
Sector	Industry Sector
Exchange	Different Stock Exchanges
Data Type	Available types of prices are: Opening, Closing, Highest and Lowest Price
Data Type Base Date	The point of time from which one needs to collect the data
Data frequency	Available frequencies are: Daily, Weekly, Monthly, Quarterly and Yearly

Table 3.1 Criteria to be set for a typical search in Datastream

The closing stock price data are collected for the companies belonging to the "Oil & Gas Producers" and "Water & Gas Multiutilities" sectors.

In general, the data collection phase can be completed through the following steps:

1. Running Microsoft Excel and selecting "Datastream-AFO/ Time-Series Request" from the drop-down menu
2. Datastream uses codes to identify companies and other financial securities. If the code is known, it should be entered in the "Series/Lists" field. If not, the "Series Selection" button can be used to search.
3. Selection of the correct Data Category. To search for a company, "Equities" can be used.
4. Typing the name of the company in the "Name" field. It is better, however, to narrow down the search using "Market"; "Status"; "Exchange". To look for a list of companies, this field must be left blank.
5. Selecting the appropriate Market
6. Selecting the appropriate Base Date; It reflects the date starting from which the company is appeared in the Market
7. Selecting the appropriate Currency
8. Selecting the appropriate Status: Options are: Active, Dead, Suspended or All
9. Selecting the appropriate Instrument Type
10. Selecting the appropriate Exchange
11. Selecting the appropriate Sector
12. Selecting the appropriate Data Type
13. Selecting the appropriate Data Type Base Date: It reflects the date starting from which the data to be collected for a company
14. Clicking on the "Search" button: A list of companies will appear
15. Selecting the companies which the data are needed for and clicking on the Explorer link
16. At the "Time Series Request" window, the Data Type must be specified again
17. Adding a Start and End Date if the data for a specific interval are needed
18. Selecting the appropriate Frequency
19. Clicking on Submit: the data will export to an Excel sheet showing the time series values for each company

To get the list of companies which the data are collected for, after step 14, it is necessary to select the companies and click on the "Microsoft Excel" button at the top right of the window. Table 3.2 shows the values that are selected for this research.

Field Name	Value Set
Name	Blank (No specific company)
Market	USA and CANADA
Base Date	Before 01/01/95
Currency	USD
Data Category	Equities
Status	Both Dead and Active
Instrument Type	Equities
Sector	Oil & Gas Producers & Gas, Water & Gas Multiutilities
Exchange	NASDAQ, TSX, NYSE
Data Type	Closing Price
Data Type Base Date	After 01/01/95
Data frequency	Weekly

Table 3.2 Set values to collect the required closing stock price data

This procedure is repeated four times to obtain the required data for active and dead companies belonging to each of the stated sectors. It is realised that there are two major drawbacks in using the data as collected:

1. The length of time series is not the same for all of the companies. The amounts of available data are different for each company.
2. For the dead companies, the value of the closing stock price remains constant from some point of time, which means that the company is dead after that point. Figure 3.4 illustrates the weekly closing stock price for a dead company.

Figure 3.4 Weekly closing stock price fluctuation for a sample dead company

These drawbacks affect the quality of the results obtained from the experience. To address these problems the constant part of the time series for dead companies is ignored. Also the data need to be sorted out in a time frame in such a way that the maximum number of companies along with the maximum amount of weekly price data can be obtained. Selecting and sorting out the data between two points of time causes a part of the collected data for some companies to become useless. Those companies that do not have sufficient data available between the specified points have to be eliminated from the collected data. This is the main reason for high reduction of dead companies after the completion of the data preparation phase.

Finally, the prepared data to be entered into the system for further calculations are represented as

$$\text{Data} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1j} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2j} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & x_{i3} & \cdots & x_{ij} \end{bmatrix}_{C \times W} \tag{3.32}$$

where "x_{ij}" is the closing stock price of company "i" at week "j". There are "C" companies, i.e. number of rows, each, having "W" weeks, i.e. number of columns, of consecutive closing stock prices in each class of data.

The common interval in which the data are obtainable for all companies is between years 1995 and 1996 in this research. After preparing the data, a 104-length time series of weekly closing stock prices ($W = 104$) is available for each company. Table 3.3 indicates the number of active and dead companies, C, for each sector before and after the data preparation phase.

Sector	Companies Status	Number of Companies (C)	
		Before	After
Oil & Gas Producers	Active	74	72
	Dead	299	58
Gas, Water & Multiutilities	Active	49	44
	Dead	97	51

Table 3.3 No. of active and dead companies before and after data preparation

Figures 3.5 and 3.6 illustrate the weekly closing stock price time series for a sample of twenty active and twenty dead companies belonging to the "Oil and Gas Producers" sector, collected from various stock exchanges Toronto Stock Exchange (TSX), NASDAQ and New York Stock Exchange (NYSE) using Datastream software.

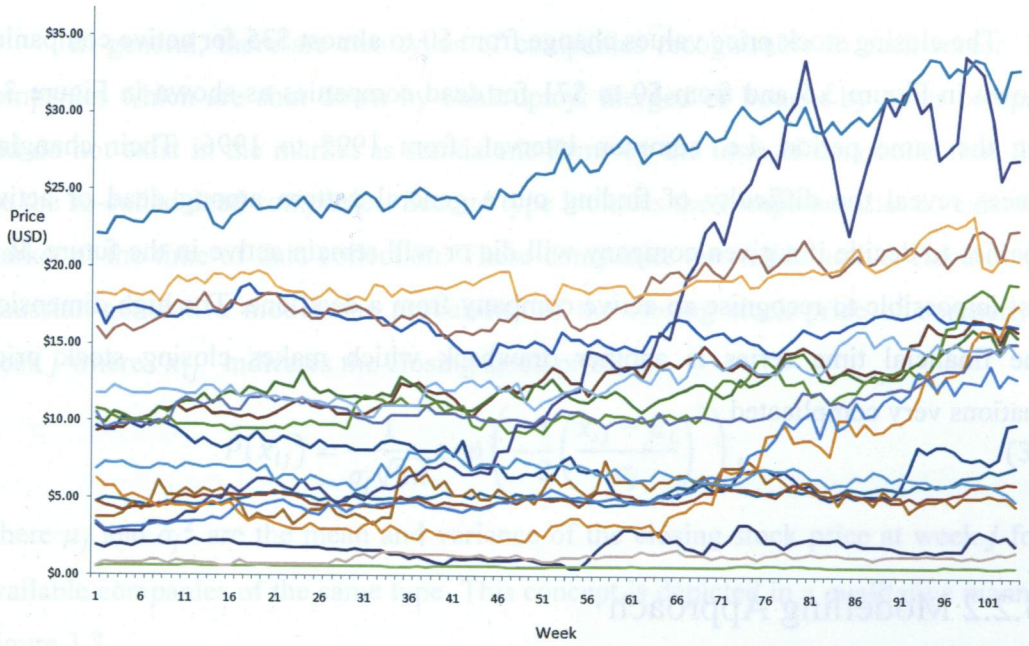


Figure 3.5 Weekly Closing stock prices for 20 active companies in "Oil & Gas Producers" sector between 1995 and 1996

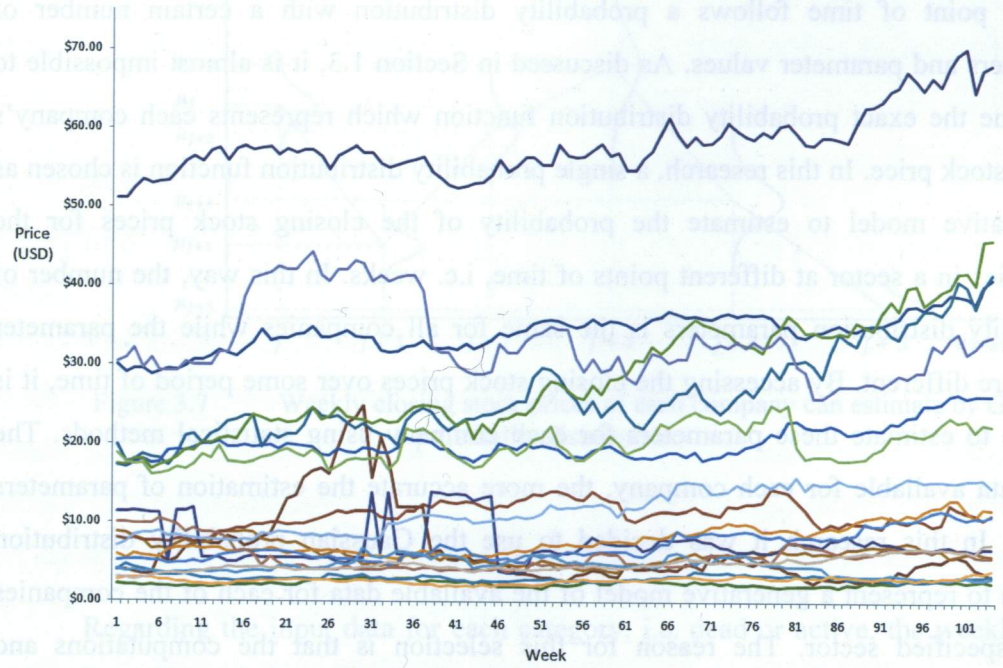


Figure 3.6 Weekly Closing stock prices for 20 dead companies in "Oil & Gas Producers" sector between 1995 and 1996

The closing stock price values change from $0 to almost $35 for active companies as shown in Figure 3.5 and from $0 to $71 for dead companies as shown in Figure 3.6 during the same period, i.e. common interval, from 1995 to 1996. Their changing variances reveal the difficulty of finding out a general pattern among dead or active companies to decide if a given company will die or will remain active in the future. It is almost impossible to recognise an active company from a dead one. The high dimension of the financial time series is another drawback which makes closing stock price evaluations very complicated.

3.2.2 Modelling Approach

The aim is to design an efficient model to provide an observer the ability to distinguish a more likely collapsing, i.e. dying, enterprise from one that is more likely to survive in a specified sector. It is assumed that the closing stock price of every company at each point of time follows a probability distribution with a certain number of parameters and parameter values. As discussed in Section 1.3, it is almost impossible to determine the exact probability distribution function which represents each company's closing stock price. In this research, a single probability distribution function is chosen as a generative model to estimate the probability of the closing stock prices for the companies in a sector at different points of time, i.e. weeks. In this way, the number of probability distribution parameters is the same for all companies while the parameter values are different. By accessing the closing stock prices over some period of time, it is possible to estimate these parameters for each company using statistical methods. The more data available for each company, the more accurate the estimation of parameters will be. In this research it was decided to use the Gaussian probability distribution function to represent a generative model of the available data for each of the companies in the specified sector. The reason for this selection is that the computations and modelling process are more tractable with Gaussian distribution. A stress-test analysis is also described in Section 3.2.6 which studies the robustness of the modelling approach.

In general, there are two types of companies recognisable in each sector. First, companies which are shut down by bankruptcy, merged or bought by other companies and do not exist in the market as standalone firms by the time of data collection. These are the so-called dead companies. Second type includes the companies that do exist in the market at the time of data collection. These companies are called active. Regarding the Gaussian generative model, the probability of the closing stock price of company i at week j where "x_{ij}" indicates the closing stock price, is

$$P(x_{ij}) = \frac{1}{\sigma_j\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{x_{ij}-\mu_j}{\sigma_j}\right)^2\right), \tag{3.33}$$

where μ_j and ${\sigma_j}^2$ are the mean and variance of the closing stock price at week j for all available companies of the same type. This concept is depicted in a qualitative manner in Figure 3.7.

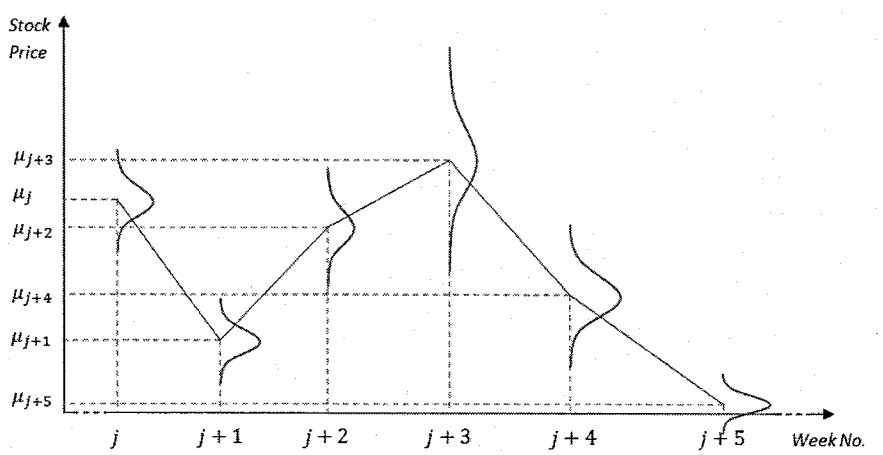


Figure 3.7 Weekly closing stock prices of each company can estimate by Gaussian probability distribution

Regarding the input data for each category, i.e. dead or active, the weekly mean and variance for each data category can be shown as coordinates of a vector as

$$\mu = [\mu_1 \quad \mu_2 \quad \mu_3 \quad \cdots \quad \mu_W]_{1\times W} \tag{3.34}$$

$$\sigma^2 = [\sigma_1^2 \quad \sigma_2^2 \quad \sigma_3^2 \quad \dots \quad \sigma_W^2]_{1\times W} \tag{3.35}$$

3.2.3 Application of Fisher Kernel in Analysis of Financial Time Series

The Fisher kernel is utilised in this research because it was shown to perform very well in many applications [15] [18] [19]. Generative models such as Gaussian Mixture Models and Hidden Markov Models [17] have been used to model the time series data [20]. However, the Fisher kernel is a different strategy that attempts to extract from a generative model more information than simply its output probability. The goal is to obtain internal representation of the data items within the model [21]. As it was decided to use a Gaussian probability distribution function as the generative model, $P_{\theta_0}(x_{ij})$ represents the probability of the closing stock price value of company i at week j, where $\theta_0 = \{\mu_j, {\sigma^2}_j\}$. Therefore, in order to calculate the probability of having a certain closing stock price for each data point $X_i = \{x_{ij}\}_{j=1}^{W}$, one needs to multiply W number of probabilities obtained from the Gaussian generative model, i.e. Equation 3.33, each having its specific estimated parameters μ_j, and ${\sigma_j}^2$ as follows,

$$P(X_i) = \prod_{j=1}^{W} P(x_{ij}) = \prod_{j=1}^{W} \frac{1}{\sigma_j\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{x_{ij}-\mu_j}{\sigma_j}\right)^2\right) \tag{3.36}$$

where W is the total number of weeks over which the closing stock price data are collected or randomly generated. $P(X_i)$ is defined with respect to the Gaussian model to be the likelihood of that specific data point X_i. So for an input space $X = \{X_i\}_{i=1}^{C}$ with C as the number of companies in a set of training data, the model parameters can be learned by adapting all the points to maximise the likelihood of the training set. In order to get rid of the complicated calculations, in terms of dealing with several multiplications, the log-likelihood of the data points, X_i, with respect to the generative model is represented as

$$log\, P(X_i) = log \prod_{j=1}^{W} P(x_{ij}) \tag{3.37}$$

and is used in all the computations.

Considering the vector gradient of the log-likelihood,

$$g(\theta, X_i) = \left(\frac{\partial \log P(X_i)}{\partial \theta_i}\right)_{i=1}^{p} \tag{3.38}$$

where p is the number of model parameters, in this case μ and σ^2.

Equation 3.38 can be expanded as

$$g(\theta, X_i) = \left(\frac{\partial \log P(X_i)}{\partial \mu}, \frac{\partial \log P(X_i)}{\partial \sigma}\right) \tag{3.39}$$

Each of the coordinates in Equation 3.39 is a vector because the gradient of the log-likelihood of the data points is calculated with respect to vectors μ and σ. In fact, the coordinates of Equation 3.39 can be shown as

$$\frac{\partial \log P(X_i)}{\partial \mu} = \left[\left(\frac{x_{i1} - \mu_1}{{\sigma_1}^2}\right), \left(\frac{x_{i2} - \mu_2}{{\sigma_2}^2}\right), \ldots, \left(\frac{x_{iW} - \mu_W}{{\sigma_W}^2}\right)\right] \tag{3.40}$$

$$\frac{\partial \log P(X_i)}{\partial \sigma} = \left[\left(-\frac{1}{\sigma_1} + \frac{(x_{i1}-\mu_1)^2}{{\sigma_1}^3}\right), \left(-\frac{1}{\sigma_2} + \frac{(x_{i1}-\mu_2)^2}{{\sigma_2}^3}\right), \ldots, \left(-\frac{1}{\sigma_W} + \frac{(x_{i1}-\mu_W)^2}{{\sigma_W}^3}\right)\right]$$

where each coordinate at the right hand side of the Equation 3.40 is defined as the Fisher score of the data point X_i at week j with respect to the log-likelihood of the generative model.

The Fisher score gives an embedding into the feature space R^N and, therefore, immediately suggests a possible kernel. The matrix I_M, which is called Fisher information matrix and is usually estimated by identity, can be used to define a non-standard inner product in that feature space. Finally, the Fisher kernel is defined as

$$K(X_i, X_j) = g(\theta, X_j)' \, {I_M}^{-1} g(\theta, X_i) \tag{3.41}$$

The practical Fisher kernel is defined as

$$K(X_i, X_j) = g(\theta, X_j)' \, g(\theta, X_i) \tag{3.42}$$

3.2.4 Training the Model

The data are collected as described in Section 3.2.1. A code is written and developed using MATLAB R2008b software to estimate all the parameters for each company. By completion of step two as stated in Section 3.2, there exists a mean, μ, and a variance, σ^2, for each week that are used for further computations. To calculate the Fisher scores, another code is developed using MATLAB R2008b.

As described earlier and from Equation 3.39, the number of coordinates of the gradient vector for each company is equal to the number of the generative model's parameters. Based on the dimension of the data, each coordinate of the gradient vector can be a vector itself. In this case, the dimension of the data is equal to the number of weeks, i.e. W. Therefore, each coordinate is a vector that contains W elements; each represents the Fisher score of the corresponding company at week j.

In order to perform a visual inspection of the data to compare the dead and active companies, the summation over all the elements of the first coordinate of the Fisher scores vector (Equation 3.39), versus the summation over all the elements of the second coordinate of the Fisher scores vector (Equation 3.39) are plotted. These values are calculated using Equations 3.43 and 3.44 respectively.

$$S_{FS_1} = \sum_{j=1}^{W} \left(\frac{x_{ij} - \mu_j}{{\sigma_j}^2} \right) \tag{3.43}$$

$$S_{FS_2} = \sum_{j=1}^{W} \left(-\frac{1}{\sigma_j} + \frac{(x_{ij} - \mu_j)^2}{{\sigma_j}^3} \right) \tag{3.44}$$

By finishing all the calculations of the Fisher scores and using Equations 3.43 and 3.44, a matrix of size $d \times 2$ is created, where d is equal to the total number of the active and dead companies in the specified sector. As a result, for every company in each class of data there are two unique values of S_{FS_1} and S_{FS_2} that can be easily plotted against each other.

3.2.5 How to Use the Model and Plots

After completion of the training phase, given a new time series of a company, the Fisher scores computes and S_{FS_1} versus S_{FS_2} plots for visual inspection. A plot is shown in Figure 3.8 that is pertaining to the "Oil and Gas Producers" sector. The new data point represent the corresponding company can be in one of the three possible areas. If it is inside or close to the red circled area, the company is considered as dead. Conversely, if the new data point is inside or close to the blue textured area the associated company is referred to as an active company. However, there is still a chance that the new point would be somewhere between the two areas. In this case it is depend on the observer's judgement to classify it as a dead or active company.

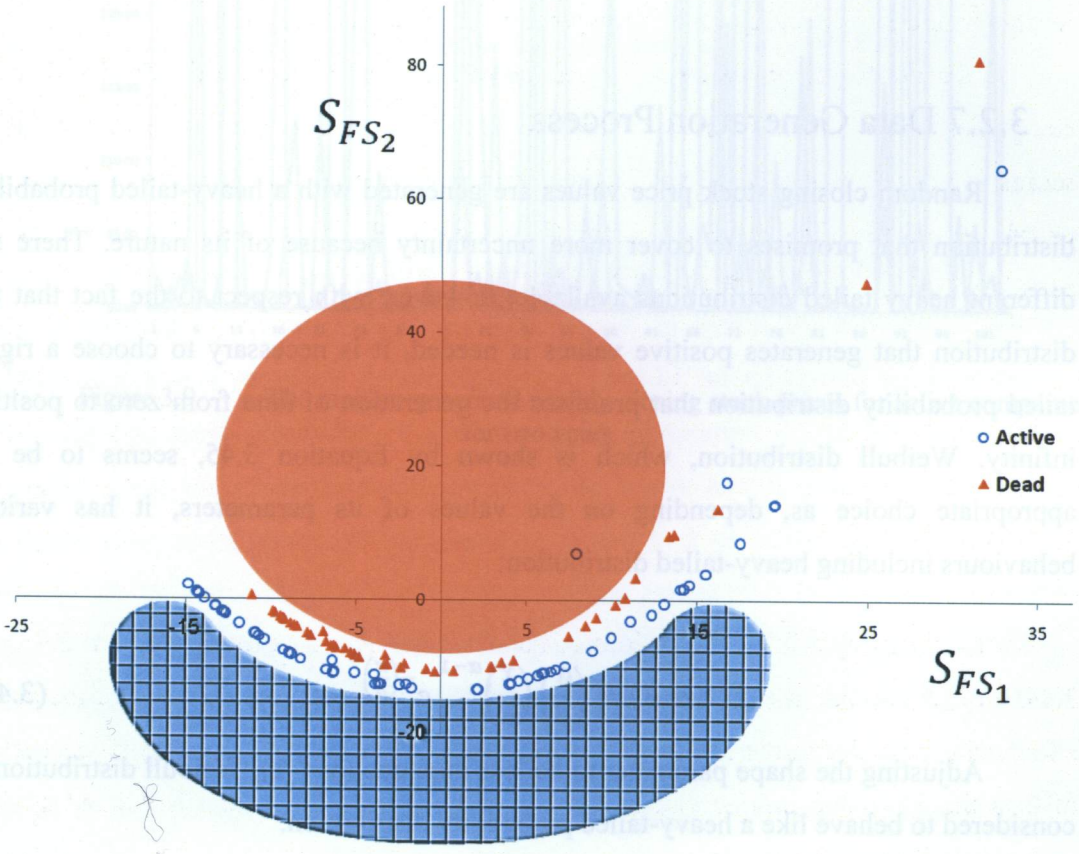


Figure 3.8 How to use the model for dead or active companies' recognition

3.2.6 Stress-test Analysis of the Modelling Approach

So far, the modelling approach has been described and the experiment was completed for data selection and visualisation. The next step is to perform a stress-test to the modelling approach by applying it to some highly randomly generated data. As explained in Section 3.2.1, the data need to be generated for two types of companies: companies that are active as standalone entities in the market and those that are dead.

One of the major drawbacks of dealing with closing stock prices is their unpredictable fluctuations from time to time. Therefore, to test the model, the level of uncertainty of the data, including the amplitude of the closing stock price variations, is increased, and the behaviour of the model under this condition is monitored.

3.2.7 Data Generation Process

Random closing stock price values are generated with a heavy-tailed probability distribution that promises to cover more uncertainty because of its nature. There are different heavy tailed distributions available; however, with respect to the fact that the distribution that generates positive values is needed, it is necessary to choose a right-tailed probability distribution that promises the generation of data from zero to positive infinity. Weibull distribution, which is shown by Equation 3.45, seems to be an appropriate choice as, depending on the values of its parameters, it has various behaviours including heavy-tailed distribution:

$$f(x,\alpha,\beta) = \left(\frac{\alpha}{\beta}\right)\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^{\alpha}} \tag{3.45}$$

Adjusting the shape parameter to be less than one ($\alpha < 1$), Weibull distribution is considered to behave like a heavy-tailed probability distribution.

Looking at Figures 3.9 and 3.10 once again proves how difficult it is to deal with the uncertainty of the data. The high dimension of the financial time series is another drawback that makes closing stock price evaluations very complicated. This is even more

complicated in this case where random closing stock prices are generated using a heavy-tailed probability distribution to increase the closing stock price variations. The randomly generated data by Equation 3.45 for active and dead companies are plotted in Figure 3.9 and Figure 3.10 respectively which indicate a higher variation uncertainty in closing stock prices compared to real values.

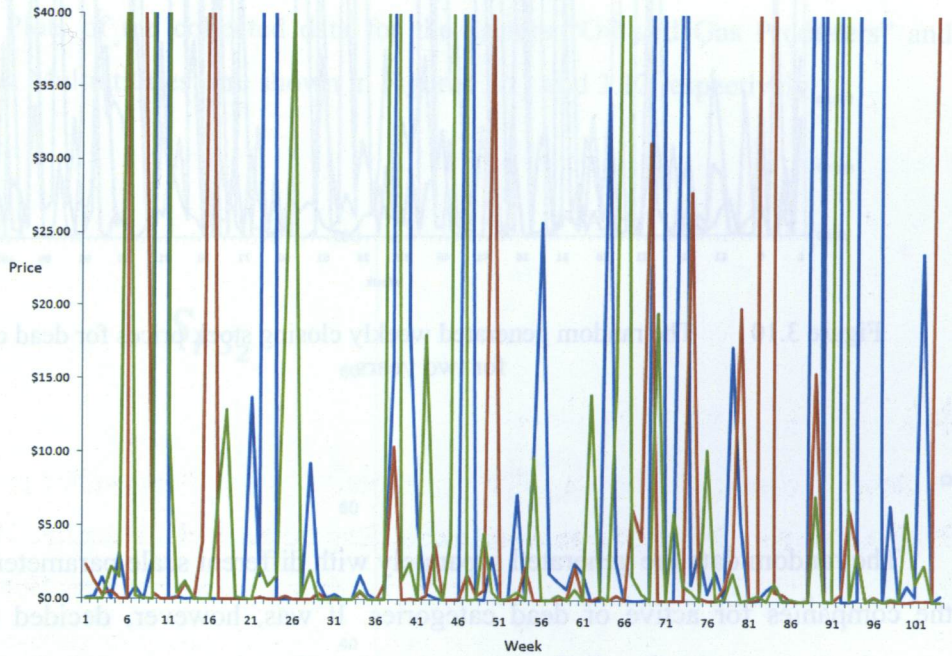


Figure 3.9 The random generated weekly closing stock prices for active companies for two years

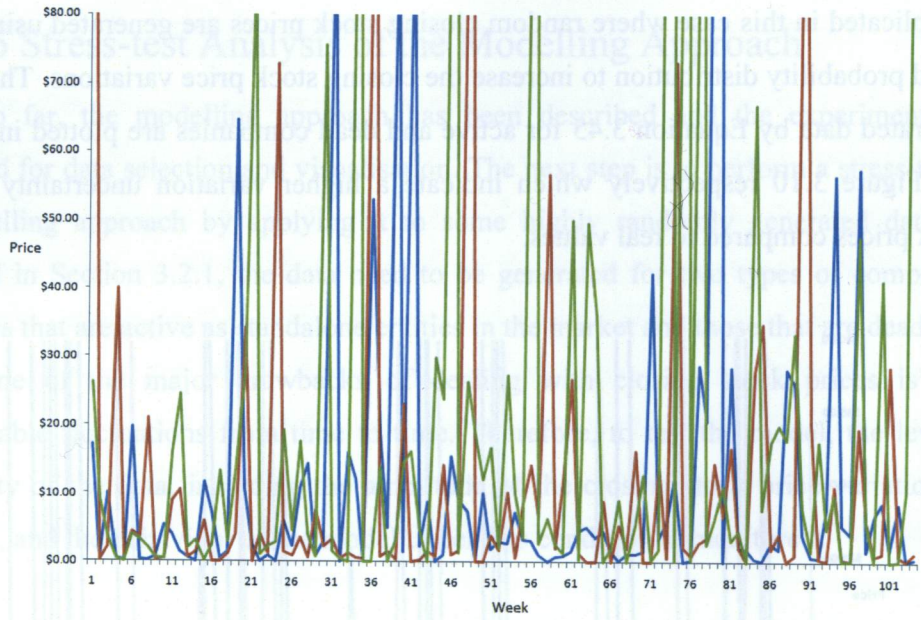


Figure 3.10 The random generated weekly closing stock prices for dead companies for two years

The random data are generated separately with different scale parameters for each of the companies for active or dead categories. It was, however, decided to have a constant shape parameter for all the companies in each category. It was also decided to train the model for 50 companies of each type, i.e. active and dead, and generate the random closing stock prices over a period of one hundred weeks. As stated before, it is not known what probability distribution each company's closing stock price follows. By generating a random time series for each company, the parameters of the generative model (see Section 3.2.2) can be estimated and used in further calculations.

In the stock market world, it is very likely to have a value out of the confidence interval. This affects the parameter estimation accuracy, the modelling process, and consequently, the results about the companies. For the model to be more robust, one needs to add as much confidence as possible to the estimated parameters to achieve the most accurate results. To address this problem it was decided to generate random data rather than make use of real values of closing stock prices. In this way it is possible to specify a wider range of closing stock price values and cover more scattered data to increase the

uncertainty and test the robustness of the model. It also provides the opportunity for sensitivity analysis of the model for different price variations.

3.3 Results and Discussion

Plots of the collected data for the sectors "Oil and Gas Producers" and "Gas, Water & Multiutilities" are shown in Figures 3.11 and 3.12 respectively.

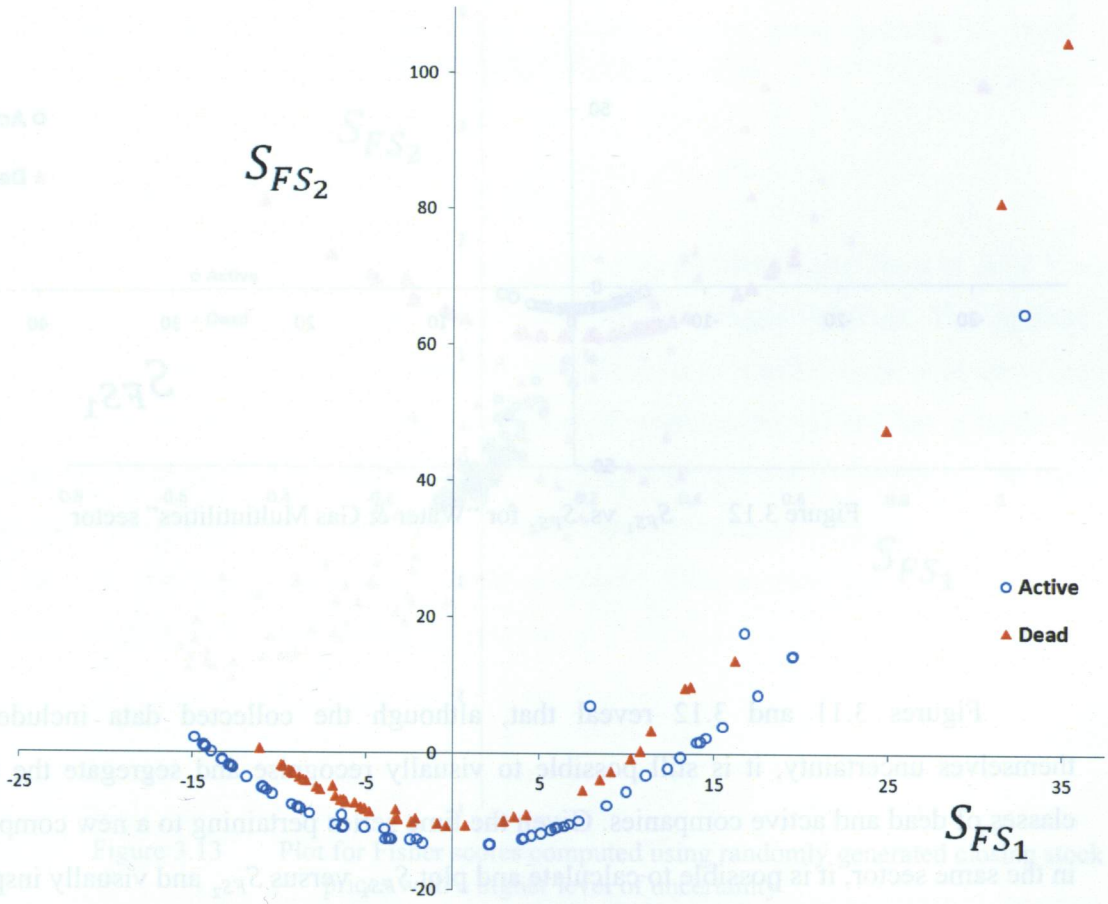


Figure 3.11 S_{FS_1} vs. S_{FS_2} plot for "Oil & Gas Producers" sector

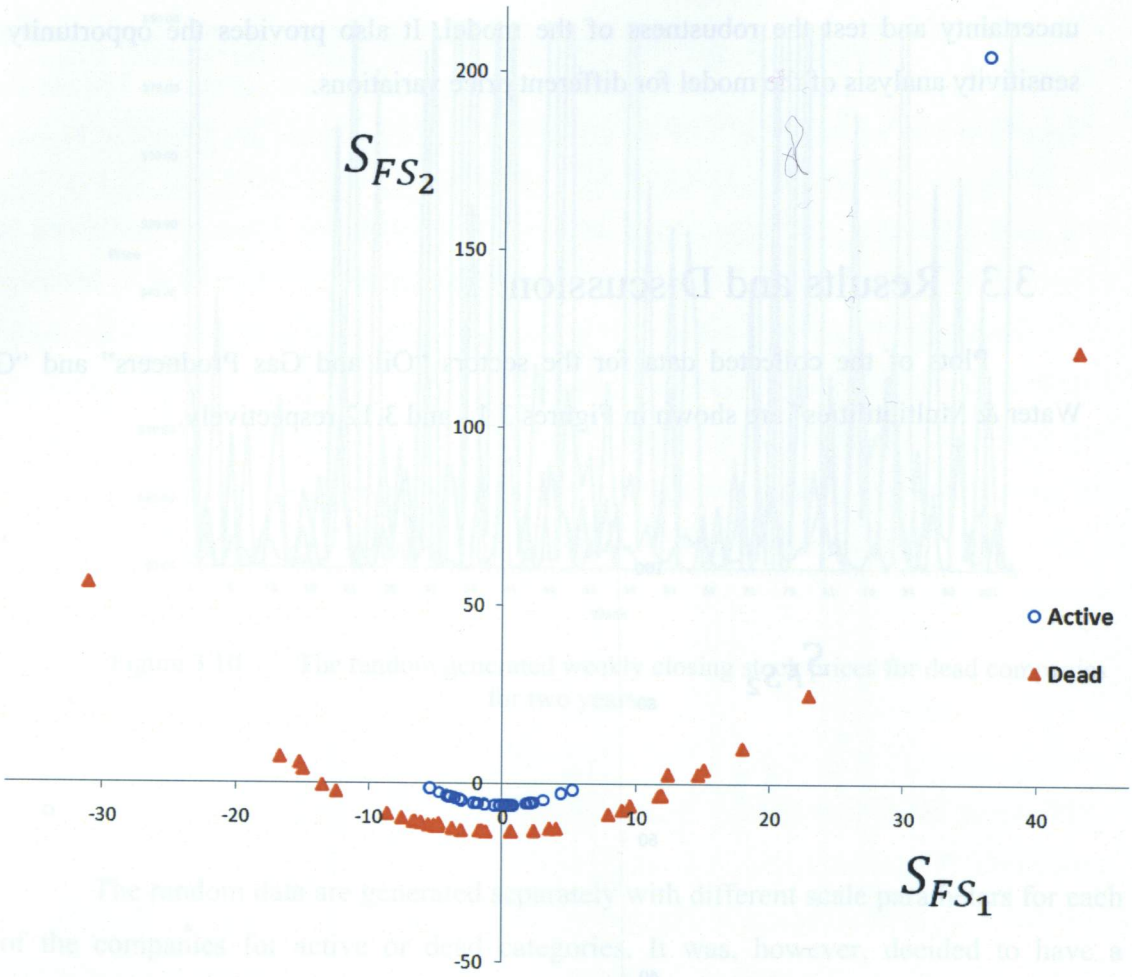


Figure 3.12 S_{FS_1} vs. S_{FS_2} for "Water & Gas Multiutilities" sector

Figures 3.11 and 3.12 reveal that, although the collected data include in themselves uncertainty, it is still possible to visually recognise and segregate the two classes of dead and active companies. Given the time series pertaining to a new company in the same sector, it is possible to calculate and plot S_{FS_1} versus S_{FS_2} and visually inspect the plot to determine if the obtained point is close to either the active or the dead companies' area.

As the amplitude of fluctuations in closing stock price decreases, the S_{FS_1} versus S_{FS_2} plot of each class of data tends to become very close to a nearly perfect parabola. In the study presented here, despite the variations in weekly closing stock prices, it is

possible to separate the two classes perfectly. Robustness of the modelling approach is examined using random generated data in Section 3.3.1.

3.3.1 Test of Robustness

S_{FS_1} versus S_{FS_2} plot of the randomly generated data is plotted in Figure 3.13.

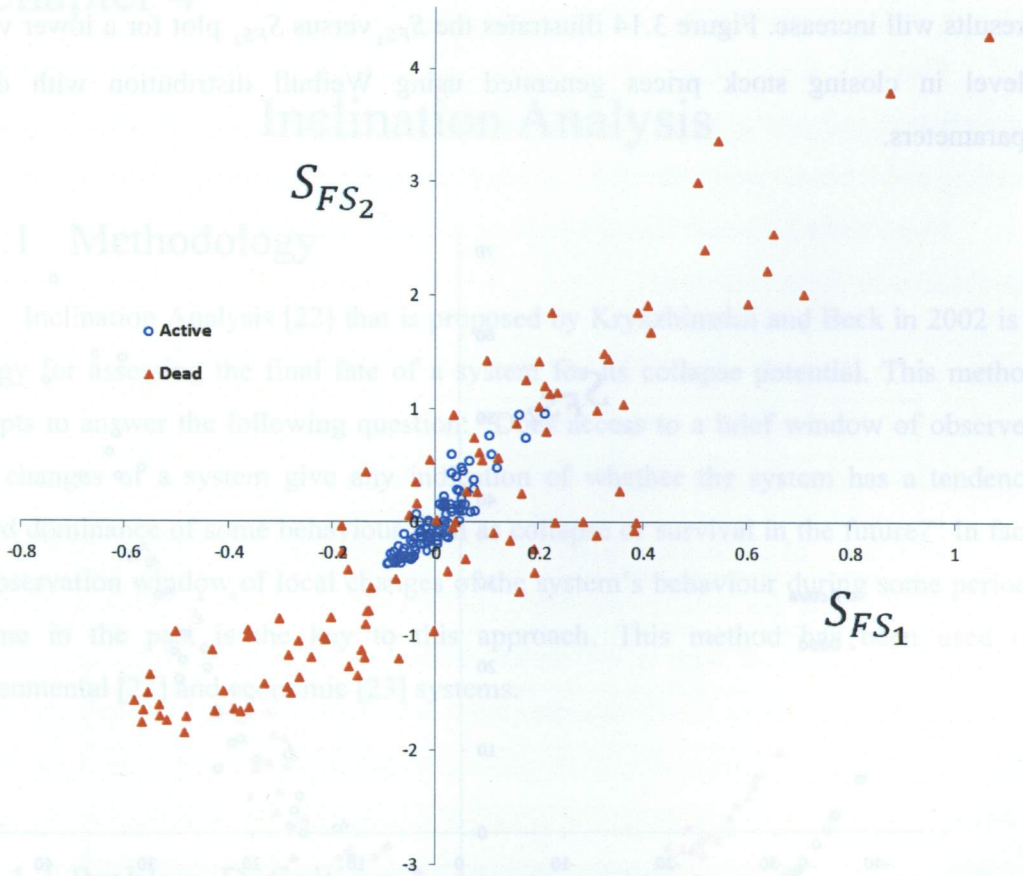


Figure 3.13 Plot for Fisher scores computed using randomly generated closing stock prices with a higher level of uncertainty

Figure 3.13 reveals that although the random generated data includes in itself more uncertainty, it is still possible to visually recognise the two classes of dead and

active companies. Hence, given a time series of a new company in the same sector, it is possible to simply plot S_{FS_1} versus S_{FS_2} and visually inspect if it is close to the areas of active or dead companies. Areas of active and dead companies in Figure 3.13 are not perfectly separable; however, compared to the level of calculations and the level of uncertainty, the judgment that one can make about a new data point is quite acceptable. As the amplitude of fluctuations in closing stock price decrease, the S_{FS_1} versus S_{FS_2} plots for each class of data tend to shape up as a parabola. From a certain level of variations it is possible to separate the two classes almost perfectly, and therefore, the accuracy of the results will increase. Figure 3.14 illustrates the S_{FS_1} versus S_{FS_2} plot for a lower variation level in closing stock prices generated using Weibull distribution with different parameters.

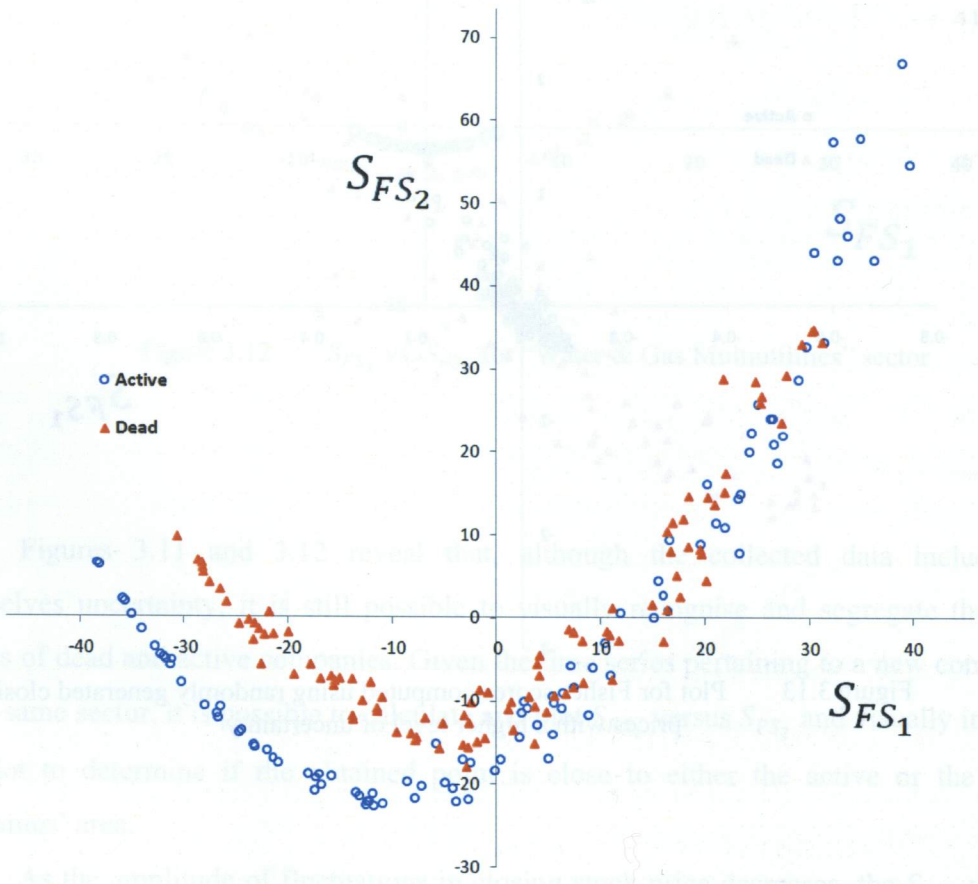


Figure 3.14 Plot of Fisher scores computed using randomly generated closing stock prices with a lower level of uncertainty

Chapter 4

Inclination Analysis

4.1 Methodology

Inclination Analysis [22] that is proposed by Kryazhimskii and Beck in 2002 is a strategy for assessing the final fate of a system for its collapse potential. This method attempts to answer the following question: "Does access to a brief window of observed local changes of a system give any indication of whether the system has a tendency toward dominance of some behaviour such as collapse or survival in the future?" In fact, the observation window of local changes of the system's behaviour during some periods of time in the past is the key to this approach. This method has been used on environmental [22] and economic [23] systems.

4.1.1 Problem Definition

In this method, the system's local changes during some periods of time in the past are classified as desired, (+), or undesired, (−), using a procedure that is specified based on the characteristics of the system under study. In fact, a set of binary models of the system defines and functions over indefinite periods of time. These periods can be considered as any real time period, such as day, week, or year depending on the system.

The system's final fate, then, is assessed based on the theory of path dependent stochastic processes. It is presumed that the system changes over discrete points of time. Systems can have different types of behaviour; however, in order to put things in a simple fashion, the model reflects a binary classification of the system's behaviour into either collapse or survival. Of course, one needs to have a clear definition of collapse and survival to express the indication and borders of each tendency. In general, there are two essential hypotheses about the system under analysis, to be made:

Hypothesis 1: The fate of the system in the far distance is governed by just two radically different terminal states: collapse or survival.

Survival means that the system continues to remain active in terms of interrelations and dealing with other systems. However, collapse mean the system stops working and it has no interact with its surrounded environment

Hypothesis 2: Local changes in time towards collapse or survival are positively correlated with those in the past.

4.1.2 Model Parameters

A number of parameters must be defined to construct the binary model. Time periods over which the data are available are denoted by k. The model's local change over a single period of time is of two categories: $+$ (toward survival) or $-$ (toward collapse). It is needed be clarify that the pluses and minuses do not indicate the system survival or collapse. They represent the local changes of the system toward survival or collapse. These changes are called transitions. It is assumed that a procedure is available to identify $+$ or $-$ transitions between each two periods. In each period k, a finite string s of length m (where m is fixed) is used to characterise those features of the system's past that have impact on the system's local change between periods k and $k+1$. The string s is any combination of $+$ and $-$ transitions realised sequentially between each two periods from $k-m$ to k. String s is defined as the state of the model in period k that gives rise to the model's transition between k and $k+1$. Therefore, the model's state is the m-long

moving window always adjoined to the latest – or + in the sequence. This concept is shown in Figure 4.1.

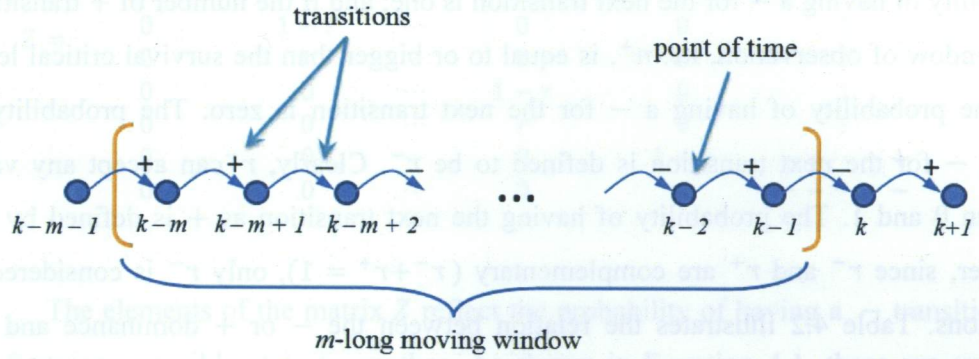


Figure 4.1 m-long moving window indicating the system's state between period $k-m$ and $k-1$

The binary model parameters are described in Table 4.1.

Parameter	Description
m	The states length of the model
m^-	Collapse Critical Level
m^+	Survival Critical Level
n^-	No. of minuses in a state
n^+	No. of pluses in a state
q	Probability of being at each of the states at $k=0$
r^-	Probability of having a minus for next transition
r^+	Probability of having a plus for next transition

Table 4.1 Binary model parameters

As a result of the second hypothesis, a strong dominance of the $-$ or $+$ in the state s causes a $-$ or $+$ at the next transition respectively. This dominance is recognised with respect to collapse critical level, i.e. m^-, for a $-$ transition and survival critical level,

i.e. m^+, for a + transition. In fact, if the number of − transitions in a window of observation, i.e. n^-, is equal to or bigger than the collapse critical level, then the probability of having a − for the next transition is one; and if the number of + transitions in a window of observation, i.e. n^+, is equal to or bigger than the survival critical level, then the probability of having a − for the next transition is zero. The probability of having − for the next transition is defined to be r^-. Clearly, r^-can accept any value between 0 and 1. The probability of having the next transition as + is defined by r^+; however, since r^- and r^+ are complementary ($r^-+r^+ = 1$), only r^- is considered in operations. Table 4.2 illustrates the relation between the − or + dominance and the probability of having − or + for the next transition.

− or + dominance	**Probability of having − (r^-)**	**Probability of having + (r^+)**
$n^- \geq m^-$	$r^- = 1$	$r^+ = 0$
$n^+ \geq m^+$	$r^- = 0$	$r^+ = 1$
$n^- < m^- \& n^+ < m^+$	$r^- = r$	$r^+ = 1 - r$

Table 4.2 Probability of next the transition being − or + respectively in different conditions

4.1.3 Construction of the Binary Model

Once the state's length, m, is fixed and with respect to the binary model, in each period of time, k, the system can be in one and only one of the 2^m different possible states. So the matrix of transition probabilities is defined as Z and is a square matrix of size $2^m \times 2^m$ as follows:

$$Z = \begin{array}{c} \begin{matrix} ++\cdots++ & ++\cdots+- & \cdots & --\cdots-+ & --\cdots-- \end{matrix} \\ \begin{bmatrix} 1-r & 0 & \cdots & 0 & 0 \\ r & 0 & \cdots & 0 & 0 \\ 0 & 1-r & \cdots & 0 & 0 \\ 0 & r & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 1-r & 0 \\ 0 & 0 & \cdots & r & 0 \\ 0 & 0 & \cdots & 0 & 1-r \\ 0 & 0 & \cdots & 0 & r \end{bmatrix} \end{array} \begin{matrix} \\ ++\cdots++ \\ ++\cdots+- \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ --\cdots-+ \\ --\cdots-- \end{matrix} \tag{4.1}$$

The elements of the matrix Z reflect the probability of having a $-$ transition, i.e. r^-, from one possible state to another. As shown in Equation 4.1, there are only two elements in each column that can have non-zero values because of the binary modelling of the system To assess the final fate of the system by having it operate over a very large number of time periods, calculation of Z_{inf} is required. This probability matrix can then be expressed as

$$Z_{inf} = \lim_{k\to\infty}(Z)^k \tag{4.2}$$

if the limit exists. In practice the value of Z_{inf} is estimated by Z^l, where l is a very large number. When the system is in period k, the probability of being in each of the possible states can be shown as a $2^m \times 1$ matrix as follows:

$$P(k) = Z^k q = \begin{bmatrix} P_{++\cdots++}(k) \\ P_{++\cdots+-}(k) \\ \vdots \\ P_{---\ldots-+}(k) \\ P_{---\ldots--}(k) \end{bmatrix} \tag{4.3}$$

where q is the initial probability of the state s at $k = 0$. q can be shown as a $2^m \times 1$ matrix:

$$q = \begin{bmatrix} P_{++\cdots++}(0) \\ P_{++\cdots+-}(0) \\ \vdots \\ P_{--\cdots-+}(0) \\ P_{--\cdots--}(0) \end{bmatrix} = \begin{bmatrix} q_{++\cdots++} \\ q_{++\cdots+-} \\ \vdots \\ q_{--\cdots-+} \\ q_{--\cdots--} \end{bmatrix} \tag{4.4}$$

It is assumed that at $k = 0$, for which there are no observational data available, the system is in a stable cycle. This helps one to specify the values of the matrix q elements. Finally, the binary model can be shown as

$$P = Z_{inf} q \tag{4.5}$$

where P is the probability vector with each element indicating the probability of being in one of the possible states after a large number of time periods. As the system operates over indefinite periods of time and with respect to the first hypothesis, one of the two states of $(- - \cdots - -)$ and $(+ + \cdots + +)$ dominates after some periods and the probability of being in any other possible states approaches zero. Therefore, the probabilities of collapse and survival are defined to be $P^- = P_{--\cdots--}$ and $P^+ = P_{++\cdots++}$ respectively. Due to the fact that the collapse and survival probabilities are complementary, the following relation holds:

$$P^- + P^+ = 1 \tag{4.6}$$

Up to this point the model-based analysis is completed without having access to past observations. Given an observation such as g, at some period, then, the Equation 4.5 will change to

$$P_c = Z_{inf} q_c \tag{4.7}$$

where c indicates the conditional probability vector with respect to observation g. Probabilities of collapse and survival will then change to

$$P_c^- = P_{--\cdots--_c}, \tag{4.8}$$

$$P_c^+ = P_{++\cdots++_c}. \tag{4.9}$$

By comparing the results from the model-based analysis with the results when there is an access to historical observations, the following can be investigated:

If $P_c^- > P^-$, one claims that the observed g sequence of local changes in the system increases the probability of collapse in the model. Hence, the model registers an inclination towards collapse.

If $P_c^- < P^-$, one claims that the observed g sequence of local changes in the system decreases the probability of collapse in the model. Hence, the model registers an inclination towards survival.

If $P_c^- = P^-$, the model registers no inclination, which is relatively rare.

4.2 Application to System Collapse Prediction

In this part the aim is to test the performance of the inclination analysis method on the same data sets of the active and dead companies belonging to the "Oil and Gas Producers" sector. In fact, having access to the closing stock prices of 50 dead companies and 50 active companies that were selected randomly from the above sector over different periods of time, an experiment was conducted to test if it is possible to predict the potential collapse of a company before it happens. This method has the advantage of not needing to cut-off the data to find the common interval of the same length of available data among all companies due to the separate consideration of each company. So it is possible to take advantage of all available data to make the prediction.

The following are the steps to establish the model:

1. Determination of the values for the parameters that shape the model dynamics:
    - m (the state's length), m^- (collapse critical level) and m^+ (survival critical level)
    - r^- (minus transition probability)
    - q (the initial probability of being at each of the possible states)
2. Calculation of the Z and Z_{inf}
3. Determination of P^- (model-based collapse probability)

4. Conversion of the closing stock prices to the binary sequences of + and – with respect to m, m^-,m^+ and r^-
5. Recognition of all the observed values and counting the number of each possible state observed
6. Computation of the q_c (conditional initial probability with respect to the observed record)
7. Determination of P_c^- (conditional collapse probability)
8. Comparison of the P^- and P_c^- to make the prediction

4.2.1 Determination of Model Parameters

Recalling from Section 4.1.2, it is necessary to determine how many previous local transitions, i.e. pluses and minuses, have an impact on determination of the next transition. Regarding the available data, which reflect the weekly closing stock price of each company, and with respect to the fact that the stock markets are very uncertain, it was decided to consider the four prior transitions as the state's length, i.e. $m = 4$, which means that the transitions over the last month have an influence on the transition to the next point of time. It should be mentioned that due to the simple structure of the model it is still possible to test it for different values of m and compare the results.

The next parameters that must be selected are m^- and m^+. It is assumed that if the number of minuses, n^-, in a state of length m is greater than m^-, the next transition will definitely be a minus; and if the number of pluses in the same state, n^+, is more than m^+, the next transition will be a plus. In this research, to avoid having an optimistic or pessimistic perspective for either side, it was decided that the critical levels for both collapse and survival would be set equal. In cases where the number of pluses or minuses in a state are not more that the specified critical level, the probability of having a minus for the next transition is equal to r^- which was set to be 0.5. Therefore, in an equal situation, equal chances were given to both pluses and minuses to arise for the next transitions. From Section 4.1.3, only r^- is used in operations.

For the initial probability of the model to be at each of the possible states, i.e. q, it was decided to give equal chances to all the possible states to appear at the very first state (uniform probability). The model parameter values are illustrated in Table 4.3

Parameter	Value
m	4
m^-	3
m^+	3
r	0.5
q	$[0.0625 \quad 0.0625 \quad \dots \quad 0.0625 \quad 0.0625]^T_{1\times 16}$

Table 4.3 Set parameter value for the experiment

4.2.2 Calculation of the Z and Z_{inf} Matrices

Having the values of state's length, collapse and survival critical levels and the probability of a minus transition for the next point of time, it is possible to calculate Z and Z_{inf} matrices. As described in Section 4.1.3, the number of possible states and consequently the size of Z and Z_{inf} matrices are exponentially correlated with the state's length. In this case, the number of possible states is 2^4 and hence, the size of matrices is $2^4 \times 2^4$. All the possible states are shown in Table 4.4.

$----$	$---+$	$--+-$	$--++$
$-+--$	$-+-+$	$-++-$	$-+++$
$+---$	$+--+$	$+-+-$	$+-++$
$++--$	$++-+$	$+++-$	$++++$

Table 4.4 Possible states for $m = 4$

4.2.3 Model-based Collapsed Probability

Regarding Equation 4.6, the model based collapse probability was calculated. In order to do so, a code was developed using Matlab R2008b. The inputs for this code are all the parameters that are determined in Section 4.2.1. The code calculates the Z_{inf} using the stated inputs and consequently provides the model-based final inclination of the system under study, which can be toward collapse or survival. By calculation of P^{-}, the first part of this method was completed. The next step was to calculate the model and observation-based final inclination of the system.

4.2.4 Conversion of Closing Stock Prices to Binary Sequence

In order to make the observed values (closing stock price time series) readable for the system, they need to be converted to a binary sequence consisting of pluses and minuses. This procedure is very important in Inclination Analysis. As stated in Section 4.2.1, the binary model in this case operates with a four-week interval ($m = 4$). Length of the observation records can be at most equal to m. In this case, to determine the plus or minus transition for each week, the closing stock price values during the last four weeks were considered. First, the total rise and total fall of the closing stock price over the last four weeks were considered and compared using the following survival condition:

$$TR + TF \geq \frac{(Average\ of\ all\ stock\ prices)}{N} \tag{4.10}$$

where TR and TF are the total amount of rise and fall in closing stock prices during the last four weeks respectively, and N represents the number of total observation records that can be calculated as

$$N = No.of\ stock\ price\ values\ in\ the\ time\ series - (L - 1) \tag{4.11}$$

where L represents the observation window length. This procedure was repeated N times until all the collected data converted to a sequence of pluses and minuses. The concept of TR and TF is shown in Figure 4.2.

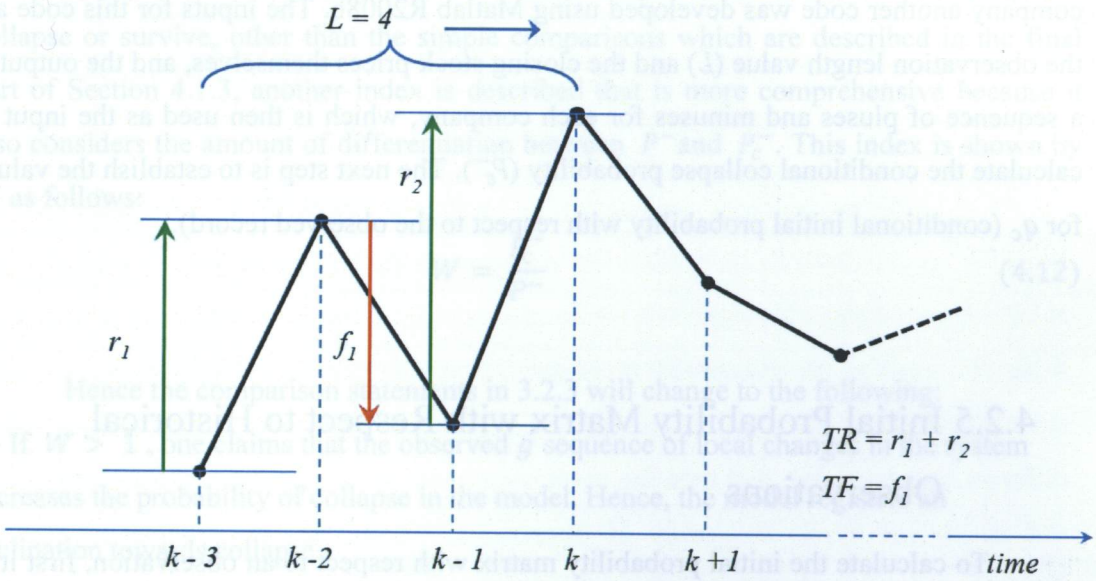


Figure 4.2 Transcribing closing stock prices to binary sequence of pluses and minuses with observation length equal to 4

For determination of the next transition, the moving window of length L, illustrated in Figure 4.2, moves one week to the right and considers the next four weeks (for example from $k-2$ to $k+1$).

The time series of observed closing stock price values were intentionally broken into a number of small windows of past local changes. Every time, a prediction is made using each observation and the results are compared to the model-based prediction to decide about the final inclination of the system.

The Equation 4.10 means that if the overall closing stock price fluctuations over the last four weeks satisfy a certain threshold, the company moves toward survival over the next week. Obviously, if it does not satisfy that condition, the company moves toward collapse over the next week. The settled threshold value reveals that, for a company to

survive during certain time periods there needs to be an overall rise at least equal to the average of its closing stock price over the same period of time.

To convert the collected data to a proper sequence of pluses and minuses for each company another code was developed using Matlab R2008b. The inputs for this code are the observation length value (L) and the closing stock prices themselves, and the output is a sequence of pluses and minuses for each company, which is then used as the input to calculate the conditional collapse probability (P_c^-). The next step is to establish the values for q_c (conditional initial probability with respect to the observed record).

4.2.5 Initial Probability Matrix with Respect to Historical Observations

To calculate the initial probability matrix with respect to an observation, first it is necessary to consider the length of the observation window. If the length of the observation window is less than m, one needs to consider the probability of the observed value occurring, using the probabilities of all possible states within which the observed value is contained. For instance, if the observed record is $(++-)$, the probability of its occurrence can be calculated as $P_{++-} = P_{-++-} + P_{+++-}$. As in this case the observation window length is equal to the state's length, i.e. $L = m$, the observed value will be one of the 16 possible states. Therefore, one needs to treat the corresponding element of the q_c matrix, that is equal to the observed record, as one and the rest of the elements as zero. For example, if the observation record is equal to $(+++-+)$ then $q_c = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0]^T$. It is now possible to calculate the P_c^-, i.e. the conditional probability of collapse for a company. All calculations of q_c and P_c^- are completed by another code that is written using Matlab R2008b.

4.2.6 Comparison of the Results of P^- and P_c^-

Although in this thesis the aim is to compare the results achieved from model-based analysis with model and observation-based analysis to see if a certain company will collapse or survive, other than the simple comparisons which are described in the final part of Section 4.1.3, another index is described that is more comprehensive because it also considers the amount of differentiation between P^- and P_c^-. This index is shown by W as follows:

$$W = \frac{P_c^-}{P^-} \tag{4.12}$$

Hence the comparison statements in 3.2.3 will change to the following:

If $W > 1$, one claims that the observed g sequence of local changes in the system increases the probability of collapse in the model. Hence, the model registers an inclination towards collapse.

If $W < 1$, one claims that the observed g sequence of local changes in the system decreases the probability of collapse in the model. Hence, he model registers an inclination towards survival.

If $W = 1$, the model registers no inclination, which is relatively rare.

Therefore, there will be a collection of models that can be separated into three classes: collapse-oriented models ($W > 1$), survival-oriented models ($W < 1$) and neutral models with respect to $W = 1$. Given the entire collection of models, using the index W, it is possible to calculate the intensity of collapse or survival for each model by subtracting its corresponded W from one. This concept of intensity of collapse or survival is depicted in Figure 4.3.

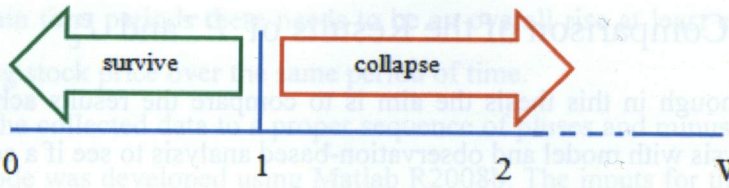


Figure 4.3 Intensity increases as the differences from 1 increase

P^- (the model-based probability of collapse), needs be to calculated one time; however, as the observation records are changing at each point of time (week), P_c^- needs to be computed every time there is a new observation. Therefore, with respect to Equation 4.11, P_c^- must to be calculated N times. To reduce the time and redundancy of the calculations, P_c^- values are calculated for all possible observation types (in our case 16 different observations), and then the associated index W is computed using Equation 4.12 and multiplied by its frequency of occurrence. Finally, the cumulative intensities of collapse and survival obtained from the entire collection of models are compared with each other to decide if a company will finally collapse or survive. Table 4.5 indicates the calculation of final collapse or survival for a sample company.

Row	Observed Record	Index W	(Intensity) Difference from 1	Frequency	Freq. × Difference
1	− − − −	2	1	34	34
2	− − − +	2	1	18	18
3	− − + −	2	1	4	4
4	− − + +	0.4	0.6	18	10.8
5	− + − −	2	1	6	6
6	− + − +	0.6667	0.3333	1	0.3333
7	− + + −	0.8	0.2	1	0.2
8	− + + +	0	1	27	27
9	+ − − −	2	1	18	18

10	+ − − +	1.2	0.2	5	1
11	+ − + −	1.3333	0.3333	3	0.9999
12	+ − + +	0	1	10	10
13	+ + − −	1.6	0.6	17	10.2
14	+ + − +	0	1	11	11
15	+ + + −	0	1	27	27
16	+ + + +	0	1	57	57
				Total collapse intensity	92.1999
				Total survival intensity	143.3333

Table 4.5 Calculation of collapse/survival intensity for a sample company

Table 4.5 reveals that the company will survive because the total intensity of survival is more than the total collapse intensity.

All described steps have been followed for 50 dead and 50 active companies from the Oil and Gas Producers sector. Each company is selected randomly among all available companies in its category to test the performance of the Inclination Analysis.

4.3 Results and Discussion

The results are shown in Table 4.6.

Category	No. of companies	Prediction Accuracy (%)
Active	50	64%
Dead	50	60%

Table 4.6 Inclination Analysis results for active and dead companies

The primary reason for conducting such an experiment is basically to create a warning system for those people who are trading the stocks of a company in the market or even for the business owners to watch the performance of their businesses and avoid a potential collapse before it is too late. In line with this approach the data for each company are collected up to 26 weeks before the current time for active companies and up to 26 weeks before the actual collapse happens for dead companies.

The above experiment could be conducted for dead companies only to test how accurately the Inclination Analysis can predict a potential collapse for a company because regarding active companies, there is no way to verify if the prediction is true or false. In fact, in the above experiment a collapse prediction for an active company would be considered as a wrong prediction with respect to the fact that is it active right now; however, the same company may become dead the next week, which means the prediction was right.

On the other hand, if the modelling performance for active companies is not tested, it is highly possible to end up with a collection of models that are mainly collapse oriented. Consequently, the Inclination Analysis will show a very high ability to predict the collapse for dead companies. The risk of predicting collapse for companies that are active will highly increase. This is, however, in contrast with the primary concern, which is to develop a warning method for active companies before they actually die. Hence, the best strategy would be tuning the modelling parameters with respect to the dead companies and make the prediction for active companies.

As for the parameters, it is possible to estimate more accurate values. In this experiment an equal chance of $\frac{1}{16}$ was given to all the possible states to happen at the initial period. However, if the dynamic of the local changes can be found, the chance of occurrence would be given to the states that are more likely to happen with respect to local changes dynamics. Regarding r^-, the model can be tested for different values say, between 0.1 and 0.9. Also different values of m, m^- and m^+ can be tested and the results can be compared to find the best parameters.

In brief, the advantages of this approach are minimum input information, simple modelling construction, and possibility to be easily tested for various parameter values.

Chapter 5

Conclusion and Future Studies

Regarding the application of kernel based machine learning, closing stock price behaviour of companies is modelled using a Gaussian generative probability model. The weekly values of the closing stock prices are mapped implicitly by application of Fisher kernel which has shown a high rate of success in dealing with classification problems in several research studies and a visualization method is proposed to provide an observer the ability to recognise the active companies from dead ones. The so-called dead companies can be one of the following types: First, companies that are bankrupted and do not exist in the market as standalone enterprises. Also there exist those that have merged or are bought by other companies. These companies cannot be referred to as bankrupted because they may have had a very successful performance in the market. As this modelling approach can be used for different systems, depending on the system size and the prediction horizon, the time scale can be different from daily to yearly or more. For instance, in dealing with ecosystems or economic systems of countries yearly time scale seems more reliable.

The objective of this prediction is to provide meaningful feedback to the business owners or stake holders to identify the problems and start to resolve them. Another prediction can be made at a later time based on the closing stock price behaviour to see if any changes have happened in terms of dealing with the problems in order to change the final fate of a company from dead to active. Also this prediction can be useful for those investors that would like to invest in stock markets.

As stated before, there are many factors that can affect systems. Another argument is that in order to make a more precise prediction for the future of a system, other features need to be considered. In this case, for example, sector indices and/or GDP could be considered as other features used along with closing stock prices for prediction. However, these features need to be recognised and quantified. This will add a cost to the modelling approach. Although the prediction made in this research is acceptable for the real values of closing stock prices, it has problems when the level of uncertainty increases. In fact, there is always a trade off between the model complexity and costs versus the accuracy of prediction.

For future studies, a Hidden Markov Model can be used as the generative model because it considers the correlation of the closing stock price variations from one point in time to the next. In this thesis the closing stock price data are sorted in such a way that a common interval for all companies can be used in the experiment. In fact, the collected data for each company are cut off to have the same length of time series data for all companies. Fisher kernel has the ability to work with different lengths of data and this is one of the reasons for choosing the Fisher kernel over other kernel functions. As another future work, Fisher kernel can be applied on the whole collected data, i.e. time series with variable lengths, and the results will be compared with what has been done so far. The principles and application of SVM have been discussed in this thesis. The calculated Fisher scores can be used as inputs for a support vector classification problem. Application of SVM for classification will help automate the classification of a company as likely to collapse or not.

As described, Inclination Analysis is a rather different method based on the stochastic processes. The structure of the model is very simple and it can apply in situations where there are not enough data available about the system under study such as complex systems. However, the performance of this method greatly depends on the parameter tuning phase. One of the most important parts of the modelling in this approach is to transcribe the collected data into a binary sequence of pluses and minuses. The more accurate this part is completed, the more accurate the result will be. Improvement in prediction accuracy from 50 percent to 60 percent at this early stage of development is acceptable due to complex nature of stock market behaviour. However,

there are some potential features, such as Sector Indices or GDP, which can be used to develop the appropriate threshold which is used for identifying the local changes towards collapse or survival.

In the Inclination Analysis method, the data that are used for prediction are cut off for the last twenty-six weeks. In fact, the model uses the data up to some point of time to make a prediction for the next twenty-six weeks. This time (around 6 months) may not be appropriate for all the companies. For the larger enterprises, one needs to provide the feedback well ahead of time compared to the smaller ones. However, in this research the aim was to test whether or not it is possible to make a prediction about the future of a company before it happens. Due to the fact that this modelling approach use the data belonging to each system separately, it is possible customise each model based on specific features of the corresponding system.

Considerations for future studies include the possibility of applying Support Vector Classification methods on the data in order to obtain the most accurate sequence of plus and minus sequence. Also, regarding the parameter tuning, finding a way to determine the optimum state length m with respect to specific applications should be considered.

Appendix A

Support Vector Machines

A.1. Overview of Support Vector Machines

Support Vector Machines (SVM) is a method that is used to train a machine in order to recognise possible patterns in a system. The training makes use of a set of examples consisting of only input and output data in cases where no human expert is able to extract and model the system dynamics or where the system under investigation is changing too fast, like a stock market. Despite some drawbacks of this learning methodology such as having access to a very limited number of examples of the system under study, SVM has been used in a wide variety of applications such as [24], [25], [26], [27] in only a few years since its introduction by Vapnik and co-workers [28], [13]

because of its ability to address many of the problems at hand. These applications can be divided into two main categories:

- Classification
- Regression

Because this research is involved with a classification experiment, the focus in what follows is only on the classification application of SVM.

A.2. SVM for classification

In Section 3.1.1, the case where the training data are linearly separable is discussed. However, there are times when the classes of data are not linearly separable, as shown in Figure A.1. This is the case in most real world problems.

A.1. Training data which are not linearly separable

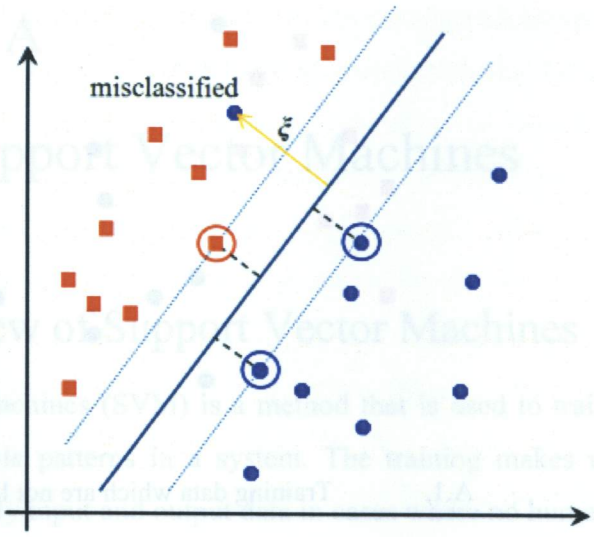
As mentioned before, SVM is one of the famous tools for classification problems. There are two approaches to generalising the problem to non-linearly separable case,

which are dependent upon prior knowledge of the problem and an estimate of the noise on the data:

- *Soft Margin* Linear hyperplane
- Non-Linear Separating hyperplane

A.2.1. The Generalised Optimal Linear Separating Hyperplane

Other than maximal-margin method, in the case where it is expected (or possibly even known) that a linear hyperplane can correctly separate the data, a method of introducing an additional cost function associated with misclassification is appropriate and is called Soft Margin (Figure A.2).



A.2. Soft Margin method where ξ is the slack variable

In fact, the soft margin method is the approach that helps to separate the data linearly by allowing non-negative variables, $\xi_i > 0$, and a penalty function,

$$F_\sigma(\xi) = \sum_i \xi_i^\sigma \qquad \sigma > 0 \tag{A.1}$$

where ξ_i is a measure of the misclassification errors. The optimisation problem is now posed so as to minimise the classification error as well as minimising the bound on the VC dimension of the classifier by actually minimising the norm $\|\omega\|$. The constraints of Equation 3.4 are modified for the non-separable case to

$$y_i[\langle \omega, X_i \rangle + b] \geq 1 - \xi_i, \qquad i = 1, \ldots, l \tag{A.2}$$

where $\xi_i \geq 0$. The generalised optimal separating hyperplane is determined by the vector ω, that minimises the functional,

$$\emptyset(\omega, \xi) = \frac{1}{2}\|\omega\|^2 + C\sum_i \xi_i \tag{A.3}$$

where C is a given value subject to the constraints of Equation A.2. The solution to the optimization problem of Equation A.3 under the constraints of Equation A.2 is given by the saddle point of the Lagrangian (Minoux, 1986),

$$\emptyset(\omega, b, \alpha, \xi, \beta) = \frac{1}{2}\|\omega\|^2 + C\sum_i \xi_i - \sum_{i=1}^{l} \alpha_i(y_i[\omega^T X_i + b] - 1 + \xi_i) - \sum_{j=1}^{l} \beta_i \xi_i \tag{A.4}$$

where α, β are the Lagrange multipliers. The Lagrangian has to be minimised with respect to ω,b, X and maximised with respect to α, β. As before, classical Lagrangian duality enables the primal problem, Equation A.4, to be transformed to its dual problem. The dual problem is given by

$$\max_{\alpha,\beta} W(\alpha, \beta) = \max_{\alpha,\beta}\left(\min_{\omega,b,\xi} \emptyset\,(\omega, b, \alpha, \xi, \beta)\right) \tag{A.5}$$

The minimum with respect to ω, b and ξ of the Lagrangian, $\emptyset$, is given by,

$$\frac{\partial \emptyset}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{l} \alpha_i y_i = 0$$

$$\frac{\partial \emptyset}{\partial \omega} = 0 \quad \Rightarrow \quad \omega = \sum_{i=1}^{l} \alpha_i y_i X_i$$

$$\frac{\partial \emptyset}{\partial \xi} = 0 \quad \Rightarrow \quad \alpha_i + \beta_i = C \qquad (A.6)$$

Hence, from Equations A.4, A.5 and A.6, the dual problem is,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left(-\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle X_i, X_j \rangle + \sum_{k=1}^{l} \alpha_k \right) \qquad (A.7)$$

and therefore the solution to the problem is given by,

$$\alpha^* = arg \min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle X_i, X_j \rangle - \sum_{k=1}^{l} \alpha_k \qquad (A.8)$$

with constraints,

$$0 \leq \alpha_i \leq C \quad i = 1, \ldots, l$$

$$\sum_{j=1}^{l} \alpha_j y_j = 0 \qquad (A.9)$$

The solution to this minimization problem is identical to the separable case except for a modification of the bounds of the Lagrange multipliers. C can be directly related to a regularization parameter (Girosi, 1997; Smola and Scholkopf, 1998). Blanz et al. (1996) use a value of $C = 5$, but ultimately C must be chosen to reflect the knowledge of the noise on the data.

A.2.2. The Generalised Optimal Non-Linear Separating Hyperplane

Subsequently, another way was suggested by Boser et al [29], in order to create non-linear classifiers. In the case where a linear boundary is inappropriate, the data from the input space can be transformed to a higher space called Feature Space that is generally of a higher dimension.

The reasons for this transformation are:

- Linear operation in the Feature space is equivalent to non-linear operation in the input space
- Classification can become easier with a proper transformation.

By choosing a non-linear mapping a priori, the SVM constructs an optimal separating hyperplane in this higher dimensional space. However, computation in the feature space can be costly because it is high dimensional (the feature space is typically infinite-dimensional).With respect to the optimization problem, the data points only appear as inner product. Therefore, a kernel function can be used so that there is no need to map data explicitly into the feature space.

Many kernel functions have been introduced, and it is possible to design kernels depending on the nature of the system nonlinearities and dynamics

Using the kernel function, the solution to the optimization problem of Equation A.7, i.e. Equation A.8, becomes,

$$\alpha^* = \arg\min_{\alpha} \left(\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(X_i, X_j) - \sum_{k=1}^{l} \alpha_k \right) \tag{A.10}$$

where $K(X_i, X_j)$ is the kernel function performing the non-linear mapping into feature space, and the constraints are unchanged,

$$0 \leq \alpha_j \leq C \quad i = 1, \ldots, l$$

$$\sum_{j=1}^{l} \alpha_j y_j = 0 \tag{A.11}$$

Solving Equation A.10 with constraints Equation A.11 determines the Lagrange multipliers, and a hard classifier implementing the optimal separating hyperplane in the feature space is given by,

$$f(X) = sgn\left(\sum_{i \in SV\ s} \alpha_i y_i K(X_i, X) + b\right) \tag{A.12}$$

where

$$\langle \omega^*, X \rangle = \sum_{i=1}^{l} \alpha_i y_i K(X_i, X)$$

$$b^* = -\frac{1}{2}\sum_{j=1}^{l} \alpha_i y_i \left[K(X_i, X_r) + K(X_i, X_r)\right] \tag{A.13}$$

The bias is computed here using two support vectors, but can be computed using all the SV on the margin for stability.

The obvious question that arises is that, with so many different mappings to choose from, which is the best for a particular problem? This is not a new question, but with the inclusion of many mappings within one framework it is easier to make a comparison.

References

[1] C. L. Magee, O. L. de Weck.(2004). Complex System Classification. Fourteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE).

[2] B. Pavard, J. Dugdale.(2000). An Introduction to Complexity in Social Science. GRIC-IRIT. Toulouse. France.

[3] C. W. J. Granger, P.Newbold.(1986). *Forecasting Economic Time Series*. San Diego: Academic Press.

[4] S. M. Ross.(2003). *An Introduction to Mathematical Finance*. Cambridge. New York: Cambridge University Press.

[5] I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, .(2004). Support Vector Machine Learning Interdependent and Structured Output Spaces. International Conference on Machine Learning (ICML)

[6] S. Zhou, R. Chellappa, B. Moghaddam. (2004). Intra-personal kernel space for face recognition. Proceeding of the 6th International Conference on Automatic Face and Gesture Recognition. FGR.

[7] H. Drucker, Wu. Donghui, V.N. Vapnik (1999). Support vector machines for spam categorization. IEEE Transactions on Neural Networks.

[8] G. Tur, R. E. Schapire, D. Hakkani-Tur.(2003). Active Learning for Spoken Language Understanding. IEEE Iternational Conference on Acoustics, Speech, and Signal Processing. Pages I-276- I-279.

[9] I. N. Flaounas, D. K. Iakovidis, D. E. Maroulis. (2006).Cascading SVMS as a Tool for Medical Diagnosis Using Multi-class Gene Expression Data. International Journal on Artificial Intelligence Tools. Vol 15. Pages 335-352.

[10] R.-Chang Chen , L. Shu-Ting and L. Shiue-Shiun. (2006). Detecting Credit Card Fraud by Using Support Vector Machines and Neural Networks. International Jurnal of Soft Computing. Pages: 30-35.

[11] N. Cristianini and J. SH. Taylor.(2000). *An introduction to support vector machines and other kernel-based learning methods*. United kingdom: Cambridge University Press.

[12] Huang, Te-Ming. (2006). *Kernel Based Algorithms for Mining Huge Data Sets : Supervised, Semi-supervised, and Unsupervised Learning*. Springer.

[13] V. Vapnik.(1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.

[14] M.L. Minsky and S.A. Papert. (1990). Perceptrons. MIT Press. Expanded Edition

[15] T.S. Jaakkola and Haussler (1998). Exploiting generative models in discriminative classifiers. Advances in neural information processing systems 11.

[16] L . Nicotra, A. Micheli and A. Starita.(2004). Fisher kernel for tree structured data. IEEE. International Joint Conference on Neural Networks.

[17] L. R. Rabiner.(1990). A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Doi: 10.1109/5.18626. Proceedings of the IEEE 77 (2): Pages 257-286.

[18] Moreno, Pedro J., and R. Rifkin.(2000). Using the Fisher Kernel Method for Web Audio Classification. International Conference on Acoustics, Speech, and Signal Processing: Proceedings, Volume IV. IEEE. Pages 2417–2420.

[19] Jaakkola, Tommi, M. Diekhans, and D. Haussler. (1999). Using the Fisher Kernel Method to Detect Remote Protein Homologies. Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology. Menlo Park, CA: AAAI Press. Pages 149–158.

[20] Y. Zhang.(2004). Prediction of Financial Time Series with Hidden Markov Models. School of Computer Science. vol. MASc: Simon Fraser University.

[21] J. SH. Taylor and N. Cristianini.(2004). *Kenel methods for pattern analysis*, United kingdom: Cambridge University Press.

[22] A.V. Kryazhimskii, M.B.Beck. (2002). Environmental Foresight and Models: a Manifesto. Identifying the inclination of a system towards a terminal state from current observation. Pages 425-451. Elsevier.

[23] S. Kryazhimskii.(2002). Economic Crisis in Russian, August 1998, Testing the Inclination Approach. Unpublished.

[24] D. A. Reynolds, T. F. Quatieri, and R. Dunn.(2000). Speaker verification using adapted Gaussian mixture models. Digital Signal Processing, vol. 10. no. 1-3. Pages 19-41.

[25] C. Bahlmann, B. Haasdonk and H. Burkhardt.(2002). On-line Handwriting Recognition using Support Vector Machines - A kernel approach. Proceeding of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02). pp 49.

[26] A. Ahmad, M. Khalid, C. Viard-Gaudin, E. Poisson.(2004). Online handwriting recognition using support vector machine. TENCON 2004 IEEE. Volume A. Pages 311-314.

[27] K. Kim.(2003). Financial time series forecasting using support vector machines. Neurocomputing. doi:10.1016/S0925-2312(03)00372-2. Pages 307-319. Elsevier.

[28] V. Vapnik.(1998). *Statistical Learning Theory*. Wiley-Interscience. New York.

[29] B. E. Boser, I. M. Guyon, V. N. Vapnik.(1992). A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory. Pittsburgh. Pennsylvania. United States. Pages 144 – 152.

[30] http://www.datastream.net/English/Default.aspx

[31] http://en.wikipedia.org