MINING OF TEXTUAL DATABASES WITHIN THE PRODUCT DEVELOPMENT PROCESS

MINING OF TEXTUAL DATABASES WITHIN THE PRODUCT DEVELOPMENT PROCESS

RAKESH MENON S/O GOVINDAN MENON
(M.Eng., M.Sc., National University of Singapore)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

MINING OF TEXTUAL DATABASES WITHIN THE PRODUCT DEVELOPMENT PROCESS

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr. R.A. van Santen, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op woensdag 15 december 2004 om 14.00 uur

door

Rakesh Menon s/o Govindan Menon

geboren te Johor, Maleisië

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. A.C. Brombacher
en
prof.dr. N. Viswandham

Copromotor:
dr. H.T. Loh

CIP-DATA BIBLIOTHEEK TECHNISCHE UNIVERSITEIT EINDHOVEN

Menon, Rakesh

Mining of textual databases within the product development process / door Rakesh Menon s/o Govindan Menon. – Eindhoven : Technische Universiteit Eindhoven, 2004. – Proefschrift. ISBN 90-386-2187-6
NUR 800
Keywords : Text classification / Quality / Reliability / Call centre / Data mining / Support vector machine / Feature selection / Product development process

ACKNOWLEDGEMENT

I never quite expected that doing a Phd would turn out to be such a daunting task. Had it not been for the guidance and support from many, this effort might not have seen fruition.

First and foremost, I thank A/Prof. Loh Han Tong, for his untiring support and guidance throughout my entire candidature. His valuable advice during the rough patches of this endeavor had proved to be very vital in shaping the course of it. Further, his critical comments and suggestions on various aspects of the thesis have definitely improved the quality of this work.

I learnt a lot about the intricacies of data mining from A/Prof. Sathiya Keerthi whose knowledge in this area is astounding. I would like to thank him for the much-valued technical advice he has rendered during our numerous discussions. His distinct ability to throw up valuable technical pointers in situations in which I thought I had exhausted all possibilities has always amazed me.

I also thank Prof. Brombacher who played a crucial role in not only convincing but also extensively supporting me in the pursuit of this Joint-Phd scheme. Despite the distance, his great enthusiasm and willingness to discuss any issue, any time at all, has made this endeavor much easier. Further, his contribution to the product development aspects of this thesis has been very valuable.

Special thanks to Prof. Viswanadham whose timely and most willing assistance as well as support was very crucial for the pursuit of this Joint-Phd scheme.

Further, I would like to thank the management of the Design Technology Institute (DTI) for supporting this work. Thanks to the members of the Knowledge Management group who willingly allowed use of their computers as additional resources. Special thanks to David for his kind assistance with Java programming and Lixiang for his miscellaneous help. Thanks also to Final Year Project students, Sachin, Ivan, Weng Seng, Ivy and Micheal as well as Tu/e Masters students Jaring, Karel and Roeland who have helped me in one way or the other.

I thank Dr. Jaya Shreeram for the time he spent having various discussions with me regarding my work and for his valuable suggestions rendered, upon painstaking reading of this thesis. Thanks very much to Dr. Lu Yuan, who so willingly and patiently took care of the numerous administrative details at the Tu/e side, from urging me to send my thesis across in time to helping me to coordinate the printing activities. I also extend my gratitude to Dr. Jan Rouvroye who kindly agreed to translate my summary into Dutch and also assisted in the administrative aspects. Thanks to Hanneke who saw to all the logistics during my visit to Tu/e. Thanks also to Dr. Shirish who had helped by providing the Markov Blanket source code.

I thank my good friend, Sivanand, whose constant encouragement, support as well as similar predicament, had provided solace. In fact, he owns the credit of initially implanting the idea of me embarking on a Phd program. Thanks very much for everything.

Family support had been very crucial for me in this effort. Thanks to my in-laws for their constant encouragement. Special thanks to my brother for helping me in his own way. This thesis is a small way of reciprocating the close to unconditional love, care, attention and support that my parents have been showering on me all these years. I am very grateful for that and am confident that this effort gives them much joy.

Lastly, special thanks to my dear wife who has been a constant pillar of support during this trying period. Her kind understanding definitely reduced additional stress that could have made this effort much more draining. The many late nights she had spent in helping me to prepare this thesis definitely deserve a special mention. A big THANK YOU to you.

TABLE OF CONTENTS

SUMMARY

As a result of the growing competition in recent years, new trends such as increased product complexity, changing customer requirements and shortening development time have emerged within the Product Development Process (PDP). These trends have given rise to an increase in the number of unexpected events within the PDP. Traditional tools/approaches are only partially adequate to cover these unexpected events. Therefore, new tools are being sought to complement traditional ones. This fact, coupled with the recent explosion of information technology that has enabled companies to collect and store increasing amounts of information, has given rise to the use of a collection of new techniques, popularly known as data mining (DM).

Although the advent of DM applications within the PDP has been quite recent, it has seen a tremendous increase lately. However, most of the applications have focused on the numerical databases found especially in the manufacturing and design phases of the PDP. There exist a large portion of textual databases within the PDP that go unanalyzed but contain a wealth of information. This thesis investigates the mining of such textual databases within the PDP.

As a first step towards the aforementioned focus, various textual databases within the PDP are identified and described. In particular, the purpose of these databases, the phase of the PDP in which these databases are used, the potential use of data mining tools on them and other relevant details are highlighted. As a particular application, the automatic classification of records in a call centre database was studied in detail. Call centre records, which are spontaneously created, exhibit unique characteristics such as

shorter document lengths, non-conformance to linguistic standards and others, which makes them different from the benchmark datasets widely studied in the literature. Hence conclusions from studies on benchmark datasets might not be directly applicable.

In view of designing an optimal classification system, an extensive study of five different factors that could potentially affect the accuracy of the classification system was undertaken. The contribution of the different factors to the classification accuracy was determined. Further, the optimal settings of these factors were identified which were then used for subsequent experiments.

Based on the previously determined settings, the representation of the documents was further investigated. Six schemes were studied in detail of which the binary representation scheme was found to give good results. In order to consider the semantics within the documents, a latent semantic representation using singular value decomposition techniques was also attempted. Such a representation resulted in a marginal improvement in the accuracy.

Textual documents usually contain a large number of features of which not many are useful for classification purposes. Hence, 3 feature reduction schemes: Information gain, Markov Blanket and a Corpus Based scheme were studied. The Markov Blanket scheme gave the best results for the investigated datasets with not more than 1% loss in accuracy after more than 50% reduction in the number of features, in the worst case setting.

A novel feature selection algorithm based on the Design of Experiment methodology was proposed. For the textual datasets studied, the success of the proposed scheme was found to be dependent on how well the training set represented the testing set. For other numerical benchmark datasets investigated, the proposed scheme was found to give good results, with an improvement in accuracy with a fewer number of features for 4 out of 5 datasets investigated.

In general, for the call centre datasets, the classification accuracies ranged from about 60% to 81%. Although the datasets were provided by a single MNC, they are quite representative of other call centre records as well since these records were generated by a third party help desk service provider who also handles calls for a number of other companies.

SAMENVATTING

Als een gevolg van de toegenomen competitie in de laatste jaren, worden moderne product ontwikkelprocessen (Engels: 'Product Development Process' of 'PDP') op dit moment gedomineerd door een aantal trends: toegenomen productcomplexiteit, veranderde klanten eisen en -wensen, en kortere beschikbare ontwikkeltijd. Deze trends hebben aanleiding gegeven tot een toename van het aantal onverwachte gebeurtenissen gedurende het product ontwikkelproces. Traditionele ontwerpgereedschappen en ontwerpmethodes kunnen deze onverwachte gebeurtenissen maar gedeeltelijk voorkomen. Daarom worden er nieuwe aanvullende gereedschappen en methodes gezocht. Dit feit, gecombineerd met de recente opkomst in informatie technologie, heeft het bedrijven mogelijk gemaakt om toenemende hoeveelheden informatie te verzamelen en op te slaan, en heeft ertoe geleid dat een aantal nieuwe technieken, algemeen bekend onder de naam 'data mining (DM)', steeds meer worden gebruikt.

Hoewel DM pas sinds kort binnen product ontwikkelingsprocessen wordt toegepast, groeit het gebruik ervan sterk. De meeste toepassingen zijn gericht op numerieke databases die gebruikt worden in de ontwerp- en productie fase van het product ontwikkelproces. Er bestaat een groot aantal tekst gebaseerde databases binnen het product ontwikkelproces, die een schat aan informatie bevatten, maar die niet geanalyseerd worden. Dit proefschrift onderzoekt de analyse van deze tekst gebaseerde databases binnen het product ontwikkelproces.

Als een eerste stap in de analyse worden diverse tekst gebaseerde databases binnen het product ontwikkelproces geïdentificeerd en beschreven. In het bijzonder wordt belicht het doel van de databases, de fase in het ontwikkelproces waarin ze worden gebruikt, het potentiële gebruik van data mining gereedschappen op deze databases en andere relevante details. Als specifieke toepassing, is de automatische classificatie van records van een call center database in detail bestudeerd. Call center records, die zonder vooraf gedefinieerde structuur (vaak spontaan), gemaakt worden, hebben karakteristieken als een korte document lengte, het niet voldoen aan linguïstische standaards, en nog een aantal andere aspecten die ze laat verschillen van datasets die in de literatuur als benchmark bestudeerd zijn. Dat is de reden waarom de resultaten van bestaande studies niet direct toepasbaar zijn.

Om een optimaal classificatie systeem te kunnen ontwerpen, is er een uitgebreide studie gemaakt van vijf verschillende factoren die potentieel invloed zouden kunnen hebben op de nauwkeurigheid van het classificatie systeem. De bijdrage van de verschillende factoren op de nauwkeurigheid van classificatie is bepaald. Verder zijn de optimale settings van deze factoren gebruikt voor volgende experimenten.

Gebaseerd op de voorafgaande settings is de representatie van de documenten verder onderzocht. Zes schema's zijn in detail onderzocht, waarvan de binaire representatie goede resultaten gaf. Om de taalkundige betekenis binnen de documenten te analyseren, is een latente taalkundige representatie met behulp van 'singular value decomposition" technieken getest. Deze representatie resulteerde in een marginale verbetering in nauwkeurigheid.

Tekstuele documenten bevatten gewoonlijk een groot aantal elementen waarvan alleen een klein deel geschikt is voor classificatie doeleinden. Daarom zijn er drie reductie algoritmes bestudeerd: 'Infomation gain', 'Markov blanket' en een 'Corpus based' algoritme. Het 'Markov blanket' algoritme gaf de beste resultaten voor de bestudeerde datasets met niet meer dan 1% verlies in nauwkeurigheid na meer dan 50% reductie in het aantal elementen, bij de slechtste setting.

Een nieuw element selectie algoritme gebaseerd op de 'Design of Experiment' methodologie is voorgesteld. Voor de bestudeerde tekstuele databases, was het succes van het algoritme afhankelijk van hoe goed de training dataset overeen kwam met de test dataset. Voor andere onderzochte numerieke benchmark datasets gaf het voorgestelde algoritme goede resultaten, met een verbetering in nauwkeurigheid met minder elementen voor 4 van de 5 onderzochte datasets.

In het algemeen zijn voor de call center data sets classificatie nauwkeurigheden bereikt tussen de ongeveer 60% tot 81%. Hoewel de datasets geleverd zijn door één bedrijf, zijn ze representatief voor andere call center records omdat deze records verkregen zijn via een 'third party' help desk die ook de telefoongesprekken voor een aantal andere bedrijven verwerkt.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

σ	Tuning parameter associated with Gaussian kernel for Support Vector Machine Algorithm
a_{ik}	Weight of the word *i* in document *k*
Area	Name of textual data set
KB	Knowledge Base
c	Tuning parameter for Support Vector Machine Algorithm
C4.5	Decision Tree algorithm
Call-Type	Name of textual data set
CB	Corpus-Based
CBR	Case based reasoning
CDP	Name of textual data set
df_i	Document frequency, the number of documents in which term *i* occurs
DM	Data Mining
Esc	Name of textual data set
f_{ij}	Term frequency, frequency of term *i* in document *j*
FWP	Name of textual data set containing three information fields, Area, Call_Type, Escalation
gf_i	Global frequency, the total number of times term *i* occurs in the whole collection
IG	Information Gain

IR	Information Retrieval
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
MB	Markov Blanket
NB	Naïve Bayes
NB (with K)	Naïve Bayes with density estimation
NN	Neural Networks
NPD	New Product Design
PCP	Product Creation Process
PD	Product Development
PDP	Product Development Process
PRP	Product Realization Process
PRS	Problem Response System
Solid	Name of textual data set
SVD	Singular Value Decomposition
SVM	Support Vector Machines
Tf	Term frequency
Tf-n	Term frequency normalized – term weighing scheme

Tfidf-ln	Term frequency inverse document frequency length normalized – term weighing scheme
Tfidf-ls	Term frequency inverse document frequency logistic scaled – term weighing scheme
Tfidf	Term frequency inverse document frequency – term weighing scheme

CHAPTER 1

INTRODUCTION

1.1 Introduction

As a result of the growing competition in recent years, new trends such as increased product complexity, changing customer requirements and shortening development time have emerged within the product development process (PDP). These trends have heightened the challenge to the already difficult task of product quality and reliability prediction and improvement. They have given rise to an increase in the number of unexpected events within the PDP. Traditional tools/approaches are only partially adequate to cover these unexpected events. Thus, new tools are being sought to complement traditional ones. This thesis investigates the use of one such tool, data mining (DM), within the PDP. Data mining, as defined by Fayyad et al. (1996a), is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data. The primary focus of this thesis is on the application of data mining techniques to databases with textual content. In particular, the classification of textual records from a Call Center database is investigated.

1.2 Product Development Process (PDP)

In the literature, there are different terminologies for a Product Development Process. Some of the common terminologies include, Product Creation Process (PCP), Product Realisation Process (PRP) and New Product Design (NPD).

Ulrich and Eppinger (2000) defined the PDP as the sequence of steps or activities which an enterprise employs to conceive, design and commercialise a product. Another term used in the literature is Product Realisation Process (PRP). Berden et al. (2000) described the PRP as a process that begins with the collection of customer requirements till the manufacture of an end product that is ready for use by the customers. Other authors such as Mill et al. (1994) used the term PRP to indicate only the last phase of the Product Development Process, the steps leading to the commercialisation of the end product. In the last few years, the term New Product Design process (NPD) has been used to describe the PDP. NPD is described as optimising a design within the constraints created by the conflicting parameters of development costs, production cost, product features, time-to-market and reliability (Goble, 1998).

As can be seen, many different definitions of product development exist. For our purposes, we adopt the following definition provided by de Graaf (Graaf, 1996):

"Product development is a sequence of design processes that converts generally specified market needs or ideas into detailed information for satisfactorily manufacturing products, through the application of scientific, technical and creative principles, acknowledging the requirements set by succeeding life-cycle processes"

The above definition well suits our needs as it recognizes the importance of information and its flow, obtained and enabled by application of technical methodologies such as data mining, in the production of high quality and reliable products.

The product development process consists of many phases, which would be outlined in the next section. It must be pointed out that given the various disciplines and expertise that constitute the PDP, the focus in this thesis would be limited to the technical aspects of developing a product in view of rapid product development with good quality and reliability. Marketing, scheduling, logistics and other like issues prevalent in the PDP would not be addressed.

1.2.1 Phases of PDP

Some organizations define and follow a precise and detailed development process, while others may not even be able to describe their processes. Although, every organization would follow a slightly different process, the basic ingredients are usually the same. In essence the major steps within the PDP, slightly modified from Ulrich and Eppinger (2000), are as follows:

- Market Need Identification
- Planning
- Design
- Testing and Refinement
- Production Ramp-up
- Service and Support

Depending on the PDP model (Function driven PDP, Sequential PDP, Concurrent PDP) these tasks are sometimes carried out in parallel or in sequence. A more elaborate discussion on each of these steps is provided in Appendix A.

1.3 Recent Challenges Within the PDP

Recently there have been numerous trends that have caused unexpected challenges within the PDP. These challenges could be outlined as follows (Brombacher, 2000):

- Increasing (technical) product complexity
- Increasing complexity of the business processes
- Changing customer expectations/requirements
- Shorter development times

Increasing complexity of products is one of the major complicating factors that influence product quality and reliability. Moore's law (Moore, 1965) has revealed that the complexity of microprocessors and other types of semiconductor integrated circuits, which are important building blocks in electronic components, has been doubling each year for the past few decades. Consequently, the complexity of professional and consumer products that consist of such building blocks have also increased. This has affected the ability of the designer to completely understand and hence effectively optimise quality and reliability of such products.

Further, with increasingly complex business processes, we have complex customer-supplier networks on a global scale that on the one hand are cost effective due to the use of resources at locations where they are optimally available while on the other hand, tremendously increase the complexity of information exchange.

With the constant stream of technological innovations, customer requirements have not only become more sophisticated, it has also become more diversified with each customer having a different set of specifications. Thus, it has become a lot more difficult to anticipate and more importantly clearly define customer requirements. Without a good appreciation for customer requirements, it becomes virtually impossible to accurately translate their needs into product specifications. This would inadvertently have adverse effects on product quality and reliability.

Due to advances in technology and the need to be the ‘first-in-the-market’, there has been an enormous pressure on the ‘time-to-market’ of a product. The company that is able, on a worldwide level, to maximally utilize the time-windows for its products will definitely have a considerable advantage. With the reduced development times, a corrective action approach to problem solving would, especially late in the process, be very expensive and inefficient. Furthermore, there is a high likelihood that the corrective action may not be applied in time. This can be seen in Figure 1.1 where, with reduced development times, the corrective action taken extends beyond the commercial release of the product. Hence, there is a need to resort to preventive actions through the use of quick and accurate quality and reliability predictive tools**.**

The challenges mentioned above can really affect the competitiveness of a company. As such, there is an urgent need to address them. This need serves as the motivation for the broad focus of the thesis.

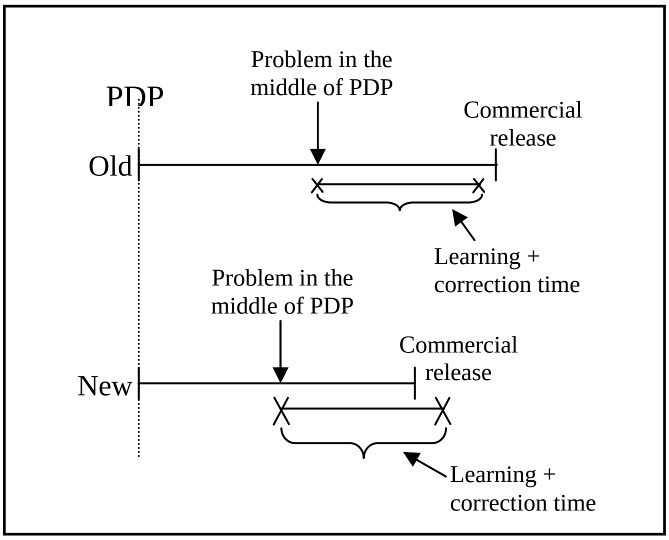

PDP
Problem in the middle of PDP
Commercial release
Old
Learning + correction time
Problem in the middle of PDP
Commercial release
New
Learning + correction time

Figure 1.1: Diagram showing inappropriateness of a corrective action approach

1.4 Broad Focus

It is clearly indicative that quality and reliability improvement is becoming increasingly difficult given the abovementioned trends. As such, companies are seeking the use of new technologies and methodologies to improve their product quality and reliability. This fact, coupled with the recent explosion of information technology that has enabled companies to collect and store increasing amounts of data, has given rise to the use of a collection of new techniques, popularly known as data mining (DM). In fact, the usefulness of such techniques is so well recognized that studies calling for re-engineering of business processes and incorporation of data warehousing and data mining to facilitate better service quality has emerged (Lee et al., 2002; Grigori et al., 2001; Miralles, 1999). These techniques perform best when massive amounts of data are available. Under these circumstances, manual processing of such data becomes inefficient and, in many cases, downright impossible. Hence the ***use of DM within the PDP*** can be a solution to this difficult problem.

Although the advent of DM applications within the PDP has been quite recent, it has seen a tremendous increase in the past two to three years. As will be shown in Chapter 2, which provides a comprehensive literature survey, most of the DM applications have focused on the manufacturing and design phases of the PDP. More importantly, a very large portion of these applications has focused on numerical databases. However, ***textual databases within the PDP go largely unanalysed***. This serves as motivation for the work in this thesis.

1.5 Motivation

The motivation for the research efforts undertaken in this thesis would be outlined below.

1.5.1 Lack of Attention Paid to Textual Data Within the PDP

As mentioned above, there has been a general lack of attention paid to the analysis of textual data within the PDP. The reasons for this lack of attention can be outlined as follows:

- Quantitative databases are relatively easy to handle and there are already various established techniques for this. In comparison, textual databases/fields are much more difficult to manipulate and there is a greater level of difficulty in handling such databases. Hence a lot of textual databases within the PDP end up simply as archives.

- Traditionally, the electronic storage of numerical inputs has been an integral part of various activities within the PDP such as in testing and process control where large amounts of numerical data is collected. However, for textual input, the usual procedure is to jot down failures, observations and etc. in a personal logbook or log sheet, usually not in electronic form.

- There has generally been a lack of know-how to handle textual databases. This emerges form the fact that tools and techniques to handle text processing are not part of the engineering curriculum. Such techniques are used and taught within the specialized areas of Information Retrieval and Natural Language Processing within the Computer Science Discipline. As a result, most engineers avoid textual databases or deal with them using simplistic treatment by working around the problem via coding of texts with keywords/phrases. More would be mentioned about coding schemes in the following subsections.

1.5.2 Wealth of Information Within Textual Data

Textual databases contain a wealth of information that would help processes within the PDP. This will become more apparent in Chapter 3, where some textual databases would be investigated in detail, including their importance. As an example, one database that is found within the design stage of the PDP is the Problem Response database. This database stores information of design problems and solutions in free text format. Such a database would provide extremely useful information to a design engineer in understanding and overcoming similar problems faced in his design work.

1.5.3 Need for Fully/Semi-Automated Text Analysis Methods

Most of the current methods of analysing textual data within the PDP include the use of spreadsheets and manual processing to decipher meaning and relationships from the textual input. Imagine having to go through 10,000 service centre records each month in order to identify new problems that have been reported on a particular record. Such tasks usually entail a lot of time and resources, which could be otherwise better utilised. Further, given the increased pressure on time-to-market, useful information needs to be extracted from these databases very quickly. Hence it would be extremely useful, if not, necessary to have automated or semi-automated text analysis schemes that would be able to infer important and pertinent information out of such huge and intimidating databases.

1.5.4 Text Encoding - Not a Good Enough Substitute

One might argue that the encoding of texts could be employed to deal with textual databases. However, many problems exist with respect to this. Firstly, such encoding could take a long time. Understanding the different possible problems that have occurred with the product, classifying such problems and finally encoding them are no easy tasks. Secondly, with rapid innovation in today's industries, products in the market change very rapidly. Every time a design change in the product occurs, the encoding system needs to be modified or, in some instances, even changed completely. It is actually possible that the product in question might have finished its market-life before such changes in the encoding system have been incorporated. Thirdly, even if an encoding list is available, it might be too long that the personnel using it might conveniently bypass it for some other quick alternatives (This disturbing trend has been observed for the call centre database investigated in this study). Finally, although it is possible for free-texts to contain a lot of unnecessary content and remarks, one could still obtain certain significant details from them that a structured and rigid encoding system would not facilitate. As such, encoding systems could never serve as a perfectly good substitute for free-texts.

Hence from the above issues, it can be seen that there is an important need (Menon, et al., 2004) to study textual data found within the PDP. Further, there is a necessity for the use of automated tools to extract useful information from these large databases in very quick time. These concerns give rise to the focus of this thesis which is the **Mining of textual databases within the Product Development Process.**

1.6 Research Efforts

Given the focus, the research efforts undertaken in this thesis could be outlined as follows:

- Mapping out DM applications within the PDP to identify missed opportunities
- Sourcing out for textual databases within the industry
- Categorization of Call Centre records as a particular application of automated schemes for text analysis. A varied number of issues are addressed in this regard. They include:
    - Suggesting an approach for the optimal design of a textual classification system
    - Conducting extensive experimental studies using a wide array of tools and techniques on the effect of:
        - Weighting schemes
        - Dimension reduction
        - Feature selection
- Proposing a novel design of experiment based methodology for feature subset selection

1.7 Thesis Organization

The thesis is organized as given below.

Chapter 2 presents the basic operations in DM. It carries out an extensive survey of DM applications within the PDP and classifies them according to the different phases of the PDP and consequently identifies the missing gaps.

Chapter 3 details textual databases that have been found in the PDP of some Multi-National Companies (MNCs). In particular, the purpose of these databases, the phase of the PDP in which these databases are used, their structure and content, the quality of information in them and the potential use of data mining tools is highlighted. Further, some of the difficulties in analysing these databases and the possible future efforts that could be taken with respect to these and similar databases found in the PDP are also presented.

Chapter 4 provides the definition and overview of concepts in text categorization. This would include document representation models, weighting schemes, feature selection methods, performance measure and machine learning techniques. It presents a brief summary of the state-of-the-art work in text categorization and argues the need for studying the text categorization problem of the call centre records.

Chapter 5 presents an approach for the optimal design of a classification system by studying the impact of various factors simultaneously. The factors studied would include the type of preprocessing, machine learning algorithm, data format, document-representation scheme as well dataset type. The popular notion of ‘designable and non-designable' factors has been adapted from the area of 'Robust Design' in the design of this system. Optimal factor settings are recommended for typical call centre datasets, which would be subsequently used in the following chapters.

Chapter 6 evaluates various document representation schemes. Six different schemes are studied on five different datasets. Recommendations are made.

Chapter 7 studies the usefulness of a dimension reduction scheme known as singular value decomposition on the call centre dataset. This provides for a 'latent semantic' document representation scheme which takes into account the meaning of words in a document.

Chapter 8 studies the effect of feature selection on the five different datasets. Three different filter-based algorithms are studied; a widely used Information Gain measure; another information theoretic based measure known as Markov Blanket and a class independent Corpus Based scheme. Modification to the corpus scheme to incorporate class information was proposed and tested.

Chapter 9 begins by presenting the design of experiment set up for identifying important features. This scheme is adapted to propose a novel feature selection scheme. The usefulness of this approach is attempted on three of the five textual datasets. To ensure the validity of this approach, 5 different numerical datasets are also studied.

Chapter 10 describes the text categorisation system implemented in an MNC. It outlines the benefits accrued due to the implementation of this system as opposed to use of conventional tools. It also presents the conclusion and suggests some directions for future work.

CHAPTER 2

DATA MINING WITHIN THE PRODUCT DEVELOPMENT PROCESS

Given its generic applicability, data mining has seen widespread application in many industries ranging from finance, bioinformatics, pharmaceuticals, telecommunications and others (Han et al., 2002 and Flower, 2003). Recently there has been a growing interest in the application of data mining to support activities within the product development process (PDP) (Braha, 2001). In this chapter, a comprehensive review of the varied applications of data mining (DM) within the PDP is presented. In order to have a better appreciation of the review, the basic ideas of the major operations of DM are explained.

2.1 Data Mining (DM)

Data mining, as defined by Fayyad et al. (1996a), is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in large databases. This is a revised definition from that of Frawley et al. (1992), to reflect developments and growth in data mining. Many people treat data mining as a synonym for another popularly used term, knowledge discovery in databases (KDD). Alternatively, others view data mining as simply an essential step in the KDD process.

Knowledge discovery, which is an iterative process, is depicted in Figure 2.1 (Fayyad et al., 1996b). According to this view, data mining is only one step in the entire process, albeit an essential one, since it uncovers hidden patterns for evaluation. However, in the industry as well as in the database research milieu, these two terms are often used interchangeably.

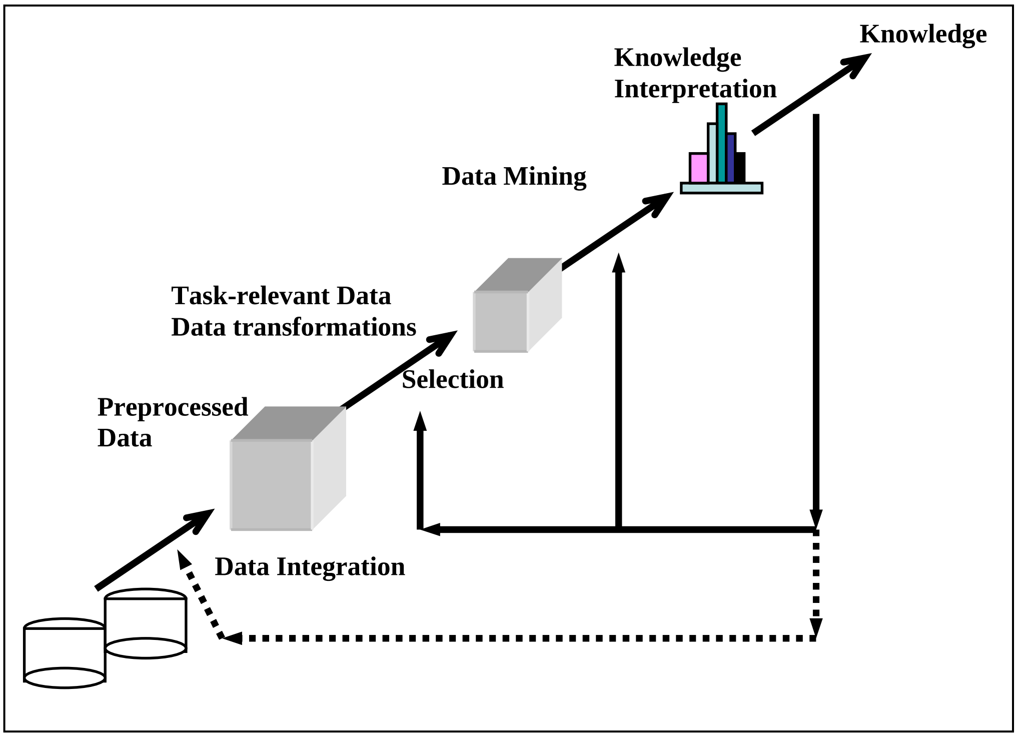


Figure 2.1: Knowledge Discovery Process

In the following subsections, the various operations within Data Mining are elaborated upon.

2.1.1 Data Mining Operations

Depending on the objective of the analysis, different types of data mining operations could be used. In general these data mining operations are used for characterizing the general properties of the database or performing inference on the current data in order to make predictions (Cabena et al., 1997; Berry and Linoff, 1997; Han and Kamber, 2000).

2.1.1.1 Predictive Modelling

Predictive modelling can be further split into two categories; Classification and Value prediction.

Classification

Classification is used to establish a specific class for each record in the database. It is basically the process of finding a set of functions that describe and distinguish data classes for the purpose of predicting the class of an object whose class label is unknown. The derived model is based on the analysis of a set of training data. The derived model may be represented in various forms, such as classification rules, decision-trees, mathematical formulae or neural networks.

Value Prediction

Value prediction is used to estimate some missing or unavailable data values rather than class labels. This is usually the case when the predicted values are continuous. Prediction also encompasses the identification of distribution trends based on the available data. Regression (Weisberg, 1985) and Neural Networks (Haykin, 1999) techniques are the more common techniques amongst a range of techniques that can be used for value prediction.

2.1.1.2 Clustering

The objective of clustering or database segmentation is to partition the database into segments of similar records that share a number of properties. Unlike classification and prediction, which analyse labelled data records, clustering analyses data objects without trained examples. The records are usually clustered or grouped based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. Each cluster that is formed can be viewed as a class of objects, from which rules can be derived. Clustering can also facilitate taxonomy, the organization of observations into a hierarchy of classes that group similar events together. Clustering can be used for both categorical and numerical data. Some classes of clustering methods would include hierarchical, partitioning, grid-based, model-based methods and so on.

2.1.1.3 Association Analysis

Association analysis seeks to establish links between items in individual transactions. The purpose of association analysis is to find items that imply the presence of other items. Rules are derived from the algorithms. It is important to make judgment on the validity of these rules. The support and confidence factors are important in determining the number of rules that are being generated. The support factor measures the relative occurrence of detected association rules within the overall data set of transactions while the confidence factor measures the degree to which the rule is true across individual records. This operation is widely used with transactional databases.

2.1.1.4 Deviation Detection

A database may contain records that do not comply with the general behaviour or model of the data. These data objects are referred to as outliers. Most data mining methods discard outliers as noise or exceptions. However in some applications such as fraud detection, the rare events can be more interesting than the more regularly occurring ones. The analysis of outlier data is referred to as outlier mining.

Outliers may be detected using statistical tests that assume a distribution or probability model for the data, or using distance measures where objects that are a substantial distance from any other cluster are considered outliers. Rather than using statistical or distance measures, deviation-based methods identify outliers by examining differences in the main characteristics of records in a group.

2.1.1.5 Evolution Analysis

Data evolution analysis describes and models regularities or trends for variables whose behaviour changes over time. Although this may include characterization, association, classification or clustering of time-related data, distinct features of such an analysis include time-series data analysis (Bowerman and O'Connell, 1993), sequence or periodicity pattern matching and similarity-based data analysis.

2.2 Data Mining Applications Within the PDP

The following subsections present a literature survey on DM applications within the PDP. An attempt is made to provide for a comprehensive and representative study of the type and amount of work carried out. Given the variety of applications, they would be broadly classified into the different phases of the PDP to which they belong. To the best of his knowledge, the author has not seen any attempt at such a classification.

2.2.1 Customer Need Identification

Yan et al. (2001) proposed an approach to study requirements of customers with varied socio-cultural background. They had made use of a laddering technique for effectively eliciting customer requirements in the early stages of product development. These requirements are expressed in the form of a hierarchy. Information from this hierarchy is made use of by a radial basis function network to study the similarities and differences between the various response groups.

Li and Yamanishi (2001) developed a system to extract information from answers to open ended questions in a consumer survey. In their system they initially identified some analysis targets. Using rules obtained from classification and association techniques, they extracted accurate characteristics for individual analysis targets. They then performed correspondence analysis using the extracted characteristics and their corresponding targets so as to position them on a 2-dimensaionl map for ease of visualization and understanding of relationship between these characteristics across various analysis targets.

Not many DM applications were found in this area since researchers seem to use more conventional techniques like, utility analysis (Pahl and Beitz, 1984), conjoint analysis (Tseng and Du, 1998) etc.

2.2.2 Planning

Chen (2003) proposed a cell-formation approach based on association rule induction to find the effective configurations for cellular manufacturing systems. Relationships among machines were found from the process database by inducting association rules. By applying association rules to cell-formation problems, certain sets of machines (machine groups) that frequently process some parts together were inducted.

Dagli and Lee (2001) mentioned about the advent of DM technologies in the product development process and highlighted the nature of changes and their implications on product development that DM would bring about. They remarked about changes in the creation of bills of materials, design of products and the generation of process plans.

Bracht and Holtz (1999) used data mining procedures for better component and part requirement forecasting against a background of widely varied series production in the automobile industry.

Here again not much work has been carried out.

2.2.3 Design and Testing

The section covers research efforts in two phases of the PDP; design and testing. This combination is due to the fact that most of the research is focused on design aspects and the distinction between design and testing is not very clear in the papers surveyed.

Rudolph and Hertkorn (2001) made use of a technique within the engineering domain, dimensional analysis, to represent the raw input data and coupled it with a technique from the artificial intelligence domain, case-based reasoning, to estimate many engineering properties of a technical object/process under consideration.

Schwabacher et al. (2001) used decision tree techniques such as C4.5 and CART to make choices for reliable optimization. In particular they used these techniques for selecting a starting prototype from a database of prototypes, synthesizing a new starting prototype and predicting which design goals are achievable. They mentioned that use of data mining techniques improved the speed and reliability of engineering design optimization.

Grabowski et al. (2001) investigated the classification of product data management (PDM) and geometry data. They made use of clustering techniques to group objects from which they derived rules for the automatic classification of future objects.

Ishino and Jin (2001) used data mining for knowledge acquisition in engineering design. They proposed an object oriented CAD system for gathering data, proposed a three layer design process model to represent the generic design process and developed the Extended Dynamic Programming method to extract know-how knowledge from the gathered design process data. Although there was mention about the use of clustering, it was not very explicitly shown how. The idea however was, to relate all the design events to an ultimately produced product model. Hence pattern-matching approaches were required. The limitation of the approach was that the knowledge to be captured must be pre-registered in advance.

Romanowski and Nagi (2001) discussed the difficulties and issues involved in developing a data mining-based engineering design support system to incorporate the heterogeneous and distributed information into the beginning stages of design. They claimed that most data mining research emphasizes learning from a knowledge base that is created “by hand” instead of being generated by machine learning algorithms.

Chi et al. (2001) have used a particular data mining tool, decision tree expert, to predict the effects of design changes, and determine the best design parameters by analysing the effect of drop test.

In their paper, Haffey and Duffy (2000) evaluated the level of support that may be achieved by using data mining tools to manipulate and utilize experiential knowledge to satisfy designers' ranging perspectives throughout a product's development. They acknowledged that data mining systems are gaining acceptance in several domains but to date remain largely unrecognised in terms of the potential to support design activities.

Most of the previous work cited within this section dealt with numerical databases. Some analysis carried out in the design phase on textual databases is presented next.

Hill et al. (2001) described a formal methodology for identifying a ‘shared understanding’ in design by analyzing design documentation. The premise of the paper was that topical similarity and voice similarity are identifiers of a shared frame of reference of the design. The voice of designers operating in a team environment was defined more as the ability of a designer to borrow the shared vision of a design team. Using the computational linguistic tool of latent semantic analysis, engineering courseware design of documents (www.needs.org) written by various authors were analyzed to reveal highly correlated groups of topics. This study also showed that there were characteristics within documents that allow the author of a document to be identified.

Yang et al. (1998) focused on making textual information more useful throughout the design process. Their main goal was to develop methods for search and retrieval that allow designers and engineers access past information and encourage design information reuse. They used informal information found in electronic notebooks since it is captured as it is generated, thereby, capturing the design process. They investigated schemes for improving access to such informal design information using hierarchical thesauri overlaid on generic information retrieval (IR) tools. They made use of Singular Value Decomposition techniques to aid in the automated thesauri generation.

Wood et al. (1998) described a method based on typical IR techniques for retrieval of design information. They created a hierarchical thesauri of life cycle design issues, design process terms and component and system functional decompositions, so as to provide a context based information retrieval. Within the corpus of case studies they investigated, it was found that the use of a design issue thesaurus can improve query performance compared to relevance feedback systems, though not significantly.

Dong and Agogino (1997) used data mining techniques to generate relationships among design concepts. In the first stage, the syntactic patterns within the design documents are analysed to determine content carrying phrases which serve as representation of the documents. In the second stage, these phrases are clustered to discover inter-term dependencies which are then used in the building of a Bayesian belief network which describes a conceptual hierarchy specific to the domain of the design.

2.2.4 Production Ramp-Up

Comparatively speaking, the production phase of the PDP has the most number of data mining applications. The primary reason for this is the fact that most of the data in the PDP is found here due to the numerous processes/activities involved in production.

Further, amongst the papers surveyed, it was found that the use of data mining is significantly greater in some industries as opposed to others. In particular, the semiconductor and aerospace industries boasts higher number of applications. These are industries in which the manufacturing processes are very complex and the data collected is immense, thereby naturally rendering DM a viable as well as a much needed tool.

Although the number of DM applications is very large, most of the papers address a common set of issues. As such, the issues tackled within this set were grouped together and are given as follows:

- Failure analysis/rapid defect detection
- Process understanding and optimization
- Yield/reliability improvement

2.2.4.1 Failure Analysis/Rapid Defect Detection

Lee and Park (2001) devised a sampling method that specifies the chip locations within the wafer to be measured, so as to detect defects quickly with a good sensitivity of 100% wafer coverage and defect detection. Self-organized maps (Kohonen, 1982) were used and results obtained were found to be very promising.

Fountain et al. (2000) studied the use of data mining in die-level functional testing for detecting die failures. In this study, a probabilistic model of patterns of die failure was constructed which was then combined with greedy value-of-information computations to decide in real time which die to test next and when to stop testing.

Other similar studies include those by: Kitler (1999) for the analysis of IC manufacturing data as a tool for failure analysis; Mieno et al. (1999) for failure analysis in the in-process monitoring stage of semiconductor plant and Kasravi (1998) for paint defect control using data mining.

2.2.4.2 Process Understanding and Optimization

In their paper, Braha and Shmilovici (2002) presented a comprehensive and successful application of data mining methodologies to the refinement of a new dry cleaning process that utilizes a laser beam for the removal of micro-contaminants in wafer manufacture.

Elsila and Roning (2002) used DM techniques in the rolling process in the steel industry to reveal novel and useful information from hundreds of features.

Giess et al.’s (2002) research was centred on discovering and quantifying relationships between the various balance and vibration tests performed throughout assembly of gas turbine rotors as well as to highlight critical parameters.

Maki and Teranishi (2001) developed an automated data-mining system for quality control in liquid crystal display fabrication. The system carried out periodical-analysis and extracted temporal-variances of the result. It was found to be useful for the rapid recovery from problems of the production process.

Gibbons and Scott (1999) described how data mining methods have been used in a high-tech manufacturing, magnetic recording facility to identify fundamental second-order process control parameter relationships which were responsible for causing process variance within manufacturing operations.

Other similar studies in the area include those by Mastrangelo and Porter (1999), Kim and Lee (1997) etc.

2.2.4.3 Yield Improvement

Gardner and Beiker (2000) carried out a case study in semiconductor wafer manufacturing where they used a combination of self-organizing neural networks and rule induction to identify critical poor-yield factors from normally collected wafer-manufacturing data. These factors were confirmed via subsequent controlled experiments.

Tsuda et al. (2000) have successfully used DM techniques to identify the correlation between yield and various wafer parametrical data to reduce the influence of the manufacturing fluctuation automatically.

Other similar studies in the area include those by Last and Kandel (2001) and McDonald (1999).

2.2.5 Service and Support

Hori et al. (2002) used data mining techniques to discover crucial repair cases from a field service database. These cases are those judged worth probing further to prevent an epidemic of quality problems. The authors employed a data mining technique known as the Apriori algorithm (Agrawal and Srikant, 1994) for this task.

Skormin et al. (2002) used data mining for field failure analysis of a avionics unit by utilizing information downloaded from dedicated monitoring systems of flight-critical hardware. They claimed that classical reliability addressed statistically-generic devices and was therefore less suitable for the situations when failures are not traced to manufacturing but rather to unique operational conditions of particular hardware units.

Fong and Hui (2001) developed a data mining technique to mine unstructured, textual data from the customer service database for online machine fault diagnosis. The data mining technique integrated neural networks (NNs) and rule-based reasoning (RBR) with case-based reasoning (CBR). In particular, NNs were used within the CBR framework for indexing and retrieval of the most appropriate service records based on a user's fault description.

Tan et al. (2000) investigated service centre call records comprising both textual and fixed-format columns, to extract information about the expected cost of different kinds of service requests. They found that the incorporation of information from free-text fields provided for a better categorization of these records, thus facilitating better predictions of the cost of the service calls.

Table 2.1 provides a summary of the work done within the PDP in relation to DM. As can be seen from the surveyed literature thus far, many of the studies carried out have focused on quantitative databases within the PDP. In particular, most of the applications seem to be centred on manufacturing and design processes. Comparatively less work has been done to explore the potential of textual databases within the PDP, although the potential for such data is large and the need for it is immediate (Menon et al., 2004). Hence as mentioned in Chapter 1, this serves as one of the motivating factors for the undertaken study.

2.3 Summary

In this chapter, a brief introduction to data mining and its relevant operations was provided. A comprehensive survey of the various applications of data mining within the PDP was provided. From the survey, it can be seen that DM applications have largely focused on numerical databases. There has been a general lack of studies on textual databases. As mentioned earlier this has been made the research focus of this study. As a first step towards this, an attempt is made to identify various textual databases within the PDP. The following chapter details this further.

Table 2.1: Summary of DM applications within the PDP

	Market Need Identification	Planning	Design & Testing	Production Ramp-up	Service & Support
Numeric Data		Component and part requirement forecasting Changes in generating process plan due to DM Effective configurations for cellular manufacturing systems	Evaluating level of support to design activities Classification of PDM and geometry data Estimating engineering properties of technical object/process Knowledge acquisition in engineering design Difficulties of developing a data mining-based engineering support system Determining effect of design parameters for drop test Selecting starting prototype	Specification of chip location for defect measurement Detecting failures for die-level functional testing Identifying second-order process control parameter in magnetic recording facility Quality control for liquid crystal display fabrication Quantifying relationship between balance and vibration test in assembly of turbine motors Refinement of cleaning process in wafer fabrication Identifying critical poor-yield factors in wafer fabrication Identifying the correlation between yield and various wafer parametrical data	Discovery of crucial repair cases from a field service database Field Failure analysis of avionics units Extracting information on expected cost of various service requests
Textual Data	Customer requirements with varied socio-cultural background Extracting information from open ended survey		Identifying shared understanding in design documentation Search and retrieval of design information Relationship among design concepts		Mining customer service database for online machine fault diagnosis

CHAPTER 3

TEXTUAL DATABASES WITHIN THE PRODUCT DEVELOPMENT PROCESS

The need for analyzing textual databases has already been highlighted in the previous chapters. In this chapter, different databases found within various MNC's are described. In particular, the purpose of these databases, the phase of the PDP in which these databases are used, their structure and content, the quality of information in them and the potential use of data mining tools on them is highlighted. Further, some of the difficulties in analysing these databases and the possible future efforts that could be taken with respect to these and similar databases found in the PDP are also presented. Most of these databases contain a mixture of both textual and numeric input. Our focus, however, would be largely directed towards the textual components of these databases.

3.1 Introduction

Textual databases are available at various stages of the PDP. However, not all databases found are amenable to analysis via data mining techniques. For example there must be an adequate amount of data in the first place before DM tools can be applied. Further, the data must be of a good quality. Although no objective criteria have been laid down to assess the quality of the data, a qualitative discussion is provided. The issues of the constraints in the data collection process, the use for the data as well as the owners and users of the data are discussed as well, since the data quality is very much dependent on these factors. This chapter is divided such that each section discusses all the above-mentioned issues with respect to a particular database investigated. The databases have been collected from various departments of two different Multi National Companies and hence reflect the status of a 'real-life' database. Interviews with the database users were conducted to understand their use of and concerns with the various databases.

3.2 Some Textual Databases Within the PDP

In this research, four databases are studied. They are: the Service Centre, the Call Centre, the Problem Response System and the Customer Survey databases. The details of these databases are provided in the sub-sections that follow. However, prior to that, the different phases in which these databases are found is presented. Figure 3.1 shows this information.

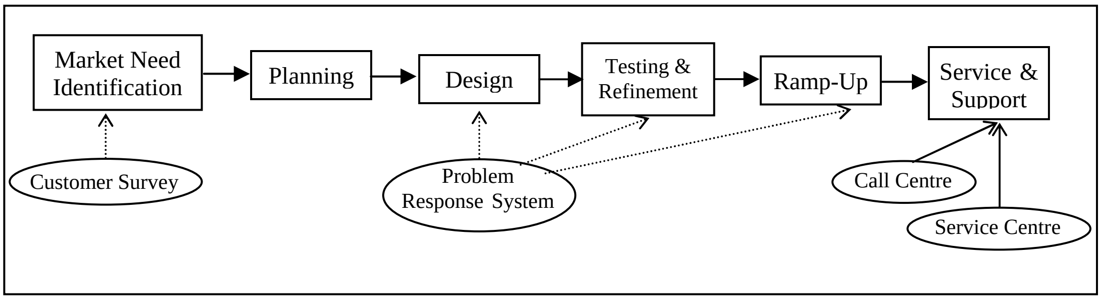


Figure 3.1: Databases studied within the PDP

Unlike most of the other databases that are confined to a single phase of the PDP, the problem response system database is used in different phases. The following sub-sections describe these databases in greater detail.

3.2.1 Service Centre Database

This database is a collection of warranty repair information from service centres located worldwide. It contains records of repair actions, customer complaints and individual product details of inkjet printers. This is a dynamic database, which grows at a rate of several thousand records per month. This database serves the following functions:

- Maintains transaction records for repair actions so as to carry out billing and accounting
- Controls the inventory of spare parts
- Obtains an estimate of component Field Call Rate

3.2.1.1 Database Collection Process

Generally, upon the malfunction of a product, a customer brings it to the service centre to have it repaired. At the front desk of the service centre, the complaints of the customers are recorded together with information of the serial number, model number and other similar details. These details together with the failed product are then forwarded to the repair personnel. Upon successful repair, the repair personnel fill in the repair details including the part numbers of the failed products. In this way, a service centre database is generated. This database is uploaded to a central repository from which authorised personnel worldwide may download the data.

3.2.1.2 Database Composition

This database is of a hybrid nature. It contains both fixed-format fields and free-form text fields. Fixed-format fields are those that have strict formatting criteria for the type, range and precision of its contents. The *"ID_Number"*, *"Sex"* and *"Date_of_Birth"* fields in a personal particulars database are examples of fixed-format fields. Free-text fields, on the other hand, are unstructured, with no formatting requirements. The *"Comments"* field in a typical questionnaire could serve as an example of a free-text field.

In the Service Centre Database that was provided by the company, there were 12 columns of data. The fields relevant to reliability and quality analysis are extracted from the database and presented in Table 3.1 below, together with their description.

Table 3.1: Information in the Service Centre database

Heading	Description	Field Format
Repair date	Contains year and month of repair	Fixed discrete
Product Serial Number	Serial Number that can be decoded to provide details of date and place of manufacture as well as unique product identification	Fixed discrete
Repair Office Code	Identifies place of repair	Fixed discrete
Replaced Part Number	Reveals the part numbers of the replaced parts	Fixed discrete
Fault Code	Failure code details	Fixed discrete
Customer Comments	Comments from customers or sometimes service centre agents about product failure	Free text
Repair Details	Details of repair action carried out by repair personnel	Free text

A typical extract of the textual component of the database is given in the Table 3.2 below.

Table 3.2: Extract of textual information in the Service Centre database

Customer Comments	Repair Details
Hardware rejection	x
Not Power	Fixed product with parts RD: Keypad Ass'y defect
EPAD/Paper Jamming at the back of the printer, Feeds Multiple Sheets	Exchange Unit CxxxA Called customer and explained we have exchanged unit and that a black ink cartridge will be sent as soon as we get them in from back order.
CSW xxx-Power Up Failure	RXE14x-Logic/Digital/Main PCA
Not printing clearly	Fixed product with parts

3.2.1.3 Quality of Database

Generally the fixed format fields are quite well filled, with less than 10% of missing values. However, for the free text fields, the quality of the input information is not as good. Although a large portion of the free-text portion is filled, not all the information is very useful. For example, examining the customer comments field will reveal frequently occurring comments such as "*Hardware Rejection*" and "*Printer does not print*". This type of information is very basic without much usable content. To a large extent such data are to be expected since the customer who returns the product only sees it to be non-functioning and hence his brief comment. The onus should therefore be on the agent at the service centre to request for further information, which would be more helpful to the company. However, this rarely happens because the primary objective of the service centre is to repair a failed product and not to accurately and meaningfully fill the database. Further, the service of repairing a product is subcontracted to third party agents who do not owe any allegiance to the company. Hence, there is a lack of interest in filling the database appropriately for useful information extraction in the future. The service centres are typically paid according to the amount of repair done and hence as long as they have records of the parts changed and some general repair details, their job is considered done. As such, these problems pose great challenges in our effort to compile a meaningful textual component of the database. Nevertheless, the present database can still be analysed to provide some information since it has a large number of records which provide enough diversity and usable content.

3.2.1.4 Potential Use of Data Mining

In this case, association rule mining would be an appropriate data mining tool. Such an approach could reveal relationship between the different columns of data. In fact, both free-text and fixed format fields can be analysed. For example, a comparison across service centres might reveal that, for a given customer complaint, a particular service centre seems to be consistently changing more parts as compared to another service centre. This kind of information could help us keep track of ‘fraudulent’ centres. This would be especially necessary, in instances where the company sub-contracts the product-servicing activities out. Association techniques could be used for such an analysis. However, prior to its application, clustering of the customer comments would be required so that each cluster could be represented as an item for the association analysis. Other free text fields could be handled in a similar fashion. This is one way of representing the free text data so that it could be used together with the fixed-format fields. Further, as presented in the table, the serial number field entry could be broken down to derive the production date information. This date, subtracted from the date of product return, would give an approximation to the Mean Time Between Failure (MTBF). Association between this derived MTBF values with the customer comments might provide some insight into failure patterns. Hence, it is possible to use association analysis to glean such useful information from the Service Centre database.

3.2.2 Call Centre Database

This database is a collection of the ‘voice of the customer’ from call centres in the US and UK. It contains records of various types of customer comments and queries, mostly product-related. Just like the Service Centre database, this is also a dynamic database. At the company from which we obtained this database, the update rate is about a few hundred records per month, depending on the product model under study. This database serves the following functions:

- Provides an avenue for customers to seek assistance and receive quick feedback
- Reduces the number of problems being escalated to the service centre
- Enables better understanding of the technical as well as the non-technical problems faced by the customer

3.2.2.1 Data Collection Process

In a typical setting, a customer dials into a call centre when he has a problem. An agent at the centre responds to the customer's query. Firstly, the incoming calls are registered and the call agent fills in the customer's details after which the exact complaint or question of the customer is typed in. This is referred to as the free text. The call agent then launches the so-called Knowledge Base (KB) to assist him in resolving the problem. The KB is a tool, which is created by knowledge engineers from the company who are familiar with the functioning of the product. Besides obtaining input from customers, via consumer tests in the initial product design stages, the knowledge engineer does his own testing to discover possible failure modes. Using this combined information, he constructs a knowledge base, which provides a 'question-answer' structure to problem solving. For example, based on the question of the customer, the agent guides the user through a series of questions and follow-up actions. At the end of this session, a solution could be reached in which case the problem is resolved. However, in the situation in which no solution is found, the problem is escalated to the service centre.

The KB however, does not cover all the possible problems since this would be impossible. It is also likely that the customer just had some enquiries, complaints or compliments instead of seeking for product-related technical assistance. In such instances, the agent types in the free texts comments and uses his own product knowledge to solve the problem of the customer or if not refers him to the service centre. This information gathered in this process constitute the Call Centre database.

3.2.2.2 Database Composition

The entire database is very huge, containing information pertaining to different models of various products. Only relevant fields pertaining to specific product models were provided by the company. Details of the fields and their description are shown in Table 3.3.

Table 3.3: Downloaded fields of the Call Centre database

Information Field	Description	Field Format
Call ID	Identification of the calls	Fixed
Call agents information	Characterization of the problem by the call agent	Fixed discrete
Text	Textual data containing customer's queries	Free text

An extract of the free text portion of the database is provided in Table 3.4 below.

Table 3.4: Extract of textual information in the Call Centre database

KB Text *UKTHIESSEN 2/4/2001 7:34:46 PM C/Cld to get help w/ unit saying picture not clear. referred to sc. going to exchange at store Advantage Knowledge Base Session:* *Description :* *Date : 2/4/01 7:33:16 PM* *Knowledge Base : FBXXX* *Notes :* *Status : Resolved* *Cause : System may be defective.* *Solution : The unit may be defective. Refer to service centre* *Symptoms :* *XXX XXX message was displayed in the Player System.* *Tests :* *Question: Is there any error message displayed?* *Response: No* *Question: Which type of xxxx is being used for xxxx?*
Free Text *UKBUCHANLS 1/13/2001 6:35:41 PM c/cld needing help with his connections to his video.*

3.2.2.3 Quality of Database

The characterization of the problem by the agent, for the fixed field, provides some summarized information with regards to the call. However, this information is often unreliable. It was found via manual analysis that a very high percentage (approximately 40%) of the calls had been wrongly characterized. With regards to the free text field, unlike benchmark databases, this database and typically Call Centre databases generated in a similar manner are quite unique with their own set of characteristics. These databases are generally not well structured. Since these databases are spontaneously created, they contain a fair amount of jargon, misspellings and grammatical inaccuracies. Generally, just as in the case of the functioning of the service centre, the call centre services are sub-contracted out. As long as the call agent can service the call and help the customer, there is no incentive for him to accurately fill in the data. In fact, for the database investigated, the performance indicator of the call agents contradicts the process for detailed data entry. The call agent is provided an incentive if he answers a call within a specified time frame. As such the agents would not be bothered to painstakingly fill in the customer comments. In fact some of the agents skip the use of the data entry system totally since, due to their vast experience, they have ready responses to some of the customers' problems. As such, it is sometimes quite difficult to accurately track the calls made to the call centre. In general however, the quality of the database is quite satisfactory with the gist of the problem being largely available in the data entries. However, more detailed data might be obtained if the call agents are judged according to the quality of the information they key in. This however is not an easy task since it might be difficult to objectively quantify the quality of information. More research would need to be done in this direction.

3.2.2.4 Potential Use of Data Mining

The classifcation of the incoming calls is an important problem pertaining to the Call Centre database. Once the calls are classified, they are channelled to appropriate personnels. For example, in an HIFI system, complaints on the different components, like the tuner, CD player and the tape deck would be channelled to the different design personnel reponsible for them. However, when analysed manually, the feedback time to breakdown the calls and alert the relevant personnel pertaining to each component becomes very long. Hence, information flow becomes delayed. In response to this, an automated classification system, which uses prediction techniques, would help save a lot of time. This would result in the early availability of information to the design personnel hence enabling them to address major problems sooner and consequently allowing them to take quicker preventive actions. Such time saving actions would definitely result in consequent cost savings.

3.2.3 Problem Response System Database (PRS)

The Problem Response system was established to track, mostly design-related problems in the different phases of the PDP and provide viable solutions to them. At the company at which it was implemented, it was available online, with several levels of access.

The objectives of setting up the PRS was threefold:

- To track problems in the different phases of the PDP
- To enhace communication between design and implementation teams situated in different geographical locations
- To enable learning from past experiece and facilitate effective preventive action with a knowledge base

Since its introduction, approximately 150 records are collected monthly. The database that was provided had about 1200 records.

3.2.3.1 Data Collection Process

Figure 3.2 shows the steps involved within the PRS. Basically, four groups of users log into the system. They are the designers, project leaders, problem solvers and the quality coaches. When a designer encounters a new problem, he logs into the system to look for a solution to his problem. If the solution to a similar problem exists, he would attempt it. Otherwise he creates a new problem record. As soon as a new problem is entered, the system automatically notifies the Project Leader, via e-mail, who would then assess the problem.

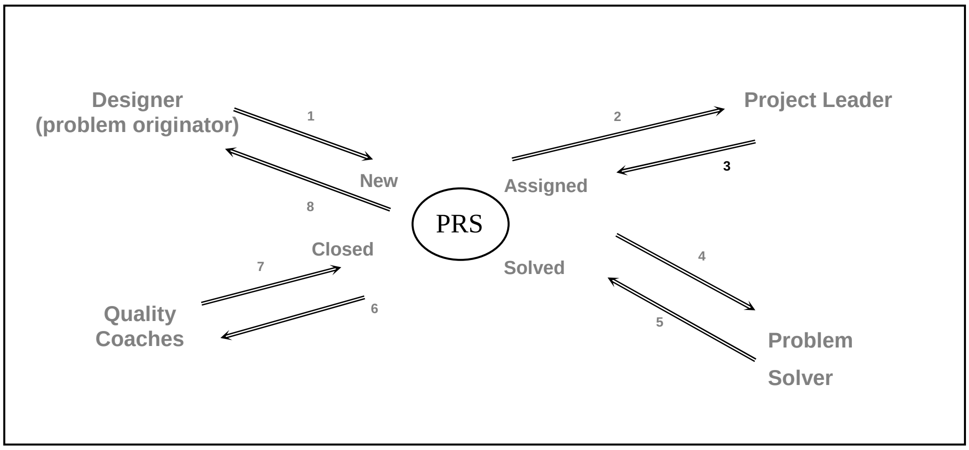


Figure 3.2: Process of Problem Response System

Depending on the problem type and content, the project leader appropriately assigns a solver to the problem who would need to work towards a solution in a given time which depends on the complexity of the problem. Once a solution is found, it will be assessed by the Quality Department. If there is a need to take preventive action, the Quality Department will do so after which they will close the problem. In this manner, a Problem Response System database is created.

3.2.3.2 Database Composition

This is a mixed format database with about 40 columns of data. A short description of some of the more important fields of the database is provided in the table below. Some of the entries into the data columns in the PRS database are modified with time, depending on the action taken. (e.g. the Defects Gravity). For example, initially when the problem is reported and no action has been taken, the defects gravity value would be high. However, after some action is taken, depending on the type of solution, this value would change appropriately. A description of the important fields of the database is provided in Table 3.5 below.

Table 3.5: Description of important fields of the PRS database

No.	Variable	Description	Format
1	Product details	Includes information like product family, values for important features (e.g screen size, speed)	Fixed discrete
2	Problem information	Includes information like o which site the problem originated from, o which field does the problem belong to (electrical, mechanical), o which part of the equipment is affected	Fixed discrete
3	Defects gravity	Reflects the severity the problem	Fixed discrete
4	Effect on FCR/FOR	A guess as to whether the problem could affect the Field Call Rate/Fall-off rate	Fixed discrete
5	Problem Description	A fairly detailed description of the problem at hand	Free text
6	Problem Solution	The solution to the problem	Free text
7	Problem Cause	Outlines the major reasons for the failure	Free text
8	Condition	The condition under which the problem is observed	Free text
9	Input Information	Who keyed the information and when was it done	Fixed discrete
10	Comments	Any additional remarks that might be useful	Free text
11	Category	Which category the problem falls into	Fixed discrete

Table 3.6 shows an extract of the textual components of the PRS database.

Table 3.6: Extract of textual information in the PRS database

Description	Symptons:TV cannot tune into PAL-xxchannels (in factory transmission & TV patten gen.), no picture or snowy picture. Description: xx byte in the Service Mode is wrong after TV power-on.Present wrong code is xx (which is UOC ROM default for xxs tuner if UOC is unable to read from NVM IC). Correct code which is stored in xxIC is y (designated for Tuner).
Comments	The chassis is an old chassis with the purpose to test the reliability of the xx. The objective is to check for any reliability failure in the xx. As there are no failure, the objective is met. The stress will have to be conducted on a full set(x y PCB) with the voice control panel.
Cause	Xx "Hi"IC having EExx was corrupted when power on/off.This pull the xC bus to low and data cannot be properly read from xx.
Solution	xx need a reset during power on/off as recommend by supplier after investigation. Reset circuit xx and yy are implement and re-run of stress test is positive.

3.2.3.3 Quality of Database

At the moment PRS is mainly used as a problem tracking system and to coordinate the communication in problem solving between design teams that are geographically located at various places. The data is not always complete as it is not compulsory to enter every record field. About 20% of the records are only partially filled or contain contents that are not very useful. This was found by manually assessing the text fields and filtering those descriptions with little/no meaningful content. It was also found that the information filled in by the engineers, especially for the description field, are of different levels of granularity. Some provided detailed information about the problem whilst others keyed in only a limited amount of information and they preferred to discuss the problem face-to-face or over the phone if a counterpart is located elsewhere. However, in general the information provided in the records was found to be quite valuable especially since the likelihood of a solution increases with better quality input.

3.2.3.4 Potential Use of Data Mining

Whenever the designer faces a problem, he would search for a similar record in the database. However this search is not an easy one. A keyword seach would end up bringing up a large set of records since the records can be quite lengthy and as such they usually contain words that repeat across documents. Sub-setting the search based on category would scale down the returned records to a certain extent. However, it is very common for the entries in the 'category' column to have the same input. In the database studied, about 70% of the problems belonged to the 'design' category. Hence this pre-defined category settings do not help much. Therefore there is a need to accurately return ranked records that might exhibit similar problems and causes. Under these circumstances, clustering might be a possible tool that could be employed. Records could be initially placed into groups using a clustering algorithm. Then, as a new problem arises, groups of records that exhibit similar problems could be returned.

3.2.4 Customer Survey Database

The customer survey database contains details of customer personal particulars together with their preferences and dislikes for different features/aspects of a particular product. The purpose of the customer survey is to:

- Understand the customer preferences
- Enable the designer/marketeer to identify key features to incorporate into the product

3.2.4.1 Data Collection Process

Generally there are various means by which this data is collected. The use of interviews, focus groups, observation of the product in use as well as questionnaire filling are some modes. Generally the customer need identification based on questionnaire is recommended for later in the process since such surveys do not provide enough information about the user environment of the product (Ulrich and Eppinger, 2000). For the database under study, the questionnaire approach was used. Recently, with the advent of web-enabled technologies, questionnaire-based surveys are actually carried out online with with real time analysis. Basically survey participants would need to go to a particular website where they fill up the questionnaire. These responses are downloaded into a database which is then analysed. For the purposes of this study, databases obtained from two surveys were analyzed. One was a simply structured paper-based survey whist the other was a longer web-based survey.

3.2.4.2 Database Composition

Both free format and fixed format fields are found in this database. Typically the fixed format fields are ordinal attributes whose values reflect the ranking of a particular feature over the other. The free text information would be responses to queries such as: why a particular feature/model is more appealing than the other; what are other functions/features that could be adapted into the product and so on. Typical data found in such a database are included in Table 3.7.

Table 3.7: Descriptions of fields within the Customer Survey database

No.	Variable	Description	Format
1	Personal details	Includes information like age, income, gender, address and others	Fixed discrete
2	Feature differentiating	Provides information as to which of these are the more important features	Fixed discrete(ordinal)
3	Directed questions	Directed towards understanding reasons for preference of a particular feature/model	Free text
4	General comments	Any additional information that the survey participant might want to input	Free text

An extract of some responses to open-ended questions on a camera and a MP3 cum CD player are given in the Table 3.8 below.

Table 3.8: Extract of textual information in the Customer Survey database

Comments	automatically select the setting for me when i take a pict. . set the mode when i take scenery picture,
	a) the camera muz be able to produce negatives which can later be even developed to any size b) the camera view slot (used for looking at objects before taking a picture should be bigger to enable old people to take pictures with ease) c) a holder for collecting and storing the pictures taken to facilitate the person take continuous shots without taking out the previous photo taken
Preferences	good design, surface finishing and sound quality
	can download the MP3 file from the internet
	lots of functions and can listen MP3 and CD at the same time
	Multi-functional, larger LCD display

3.2.4.3 Quality of Database

It was noticed that the quality of the survey is quite dependent on the feature of the survey. Generally for the shorter survey, all the questions were filled up and also to a greater detail (for the free text portion) as opposed to the longer web-based surveys. In general, the feedback from the customers was found to be constructive.

3.2.4.4 Potential Use of Data Mining

There is a need by the industrial designer to extract out the salient features/preferences from the free form text. The general understanding is that, the more often a word, phrase or a set of words occurs in the free-text (not necessarily in sequence), the more important it is. Currently this extraction of keywords/phrases is carried out manually. It could easily take two designers up to a week to analyse about 500 such records. Hence there is a need to speed up this task so that the survey could be extended to a wider audience and the results obtained faster. In this regards, sequential rule mining could be used as a tool that could facilitate this task. Using this approach, key phrases can be automatically derived from words that are not necessarily collocated side-by-side but occur with a frequency higher than a user-specified support. This would provide the user with a quick overview of the general ideas expressed in the text. This automation could result in enormous timesavings for the designers.

3.3 Improving Database Quality

A brief summary of the different databases is provided in Table 3.9.

Table 3.9: Brief summary of the various databases

Database	**Data-Entry Personnel**	**Owner of Information**	**User of Information**	**Phase of PDP**	**Data Minig Tool**
Service Centre	Third Party	Field Failure Engineer	Design Team/ Management	Service & Suport	Clustering/ Association
Call Centre	Third Party	Knowledge Engineer	Design Team/ Knowledge Engineer/ Management	Service & Suport	Classification
Problem Response	Designers	Designer	Design Team	Various Phases	Clustering
Customer Survey	Customers	Designer/ Marketeer	Design Team/ Management/ Marketing	Marketing	Sequential Rule Mining

As was mentioned in the previous subsections, the quality of the information in these databases, although is suitable for data mining purposes, could be further enhanced. Some efforts to improve the quality of the information in the databases could be outlined as follows:

a) Firstly, a general understanding must be created on the importance of the data being collected. Its potential uses and how its proper collection could facilitate and enhance some of the activities of the data collector’s themselves, must be explained. For example in the case of the call centre, the data collection process itself could be made easier if the automatic asssignment of categories for incoming calls is established. As a further example, for the PRS database, if the

Description and *Causes* fields are filled up well (bearing accurate information to the required level of granularity), then a similar problem could be more accurately retreived which would be beneficial to the designer who filled up the data records initially.

b) Secondly, it would be quite important to ensure that the data collectors are the owners of the data as well. For example, in the case of the PRS, the designers who input the data are the users of the data and therefore, there is a natural incentive for them to key in good data. However, this might be difficult to ensure in other instances. For example, in the case of the Call Centre and the Service Centre databases, it is difficult to ensure this since such decisions are guided by business motivations. Under these circumstances, proper incentives should be in place for the appropriate and accurate collection of the data. The performance indicator should be tied to the quality of the data to a certain extent. Although this is not going to be easy, some guidelines would need to be established in this regards to ensure higher quality data. Incentives that conflict with proper data entry, as in the case of the Call Centre database, should be avoided.

c) Further, some effort could be taken in the design of the database from the viewpoint of the type of information to be collected. Contrasting the web-enabled and the paper-based customer survey, the latter seemed to provide more reliable information. Careful planning should go into deciding what type of information is needed. There would be no point collecting a lot of information when most of it is not reliable.

The above are some of the efforts that could be undertaken to improve the quality of the information.

3.4 Difficulties in Analyzing Textual Databases

Textual databases pose a different set of problems that are not usually encountered with numerical databases. One such problem is that of information description. Users in different circumstances or with different needs, knowledge or linguistic habits will describe the same information using different terms. Indeed the degree of variability in descriptive term usage is much greater than is commonly suspected. For example, two people choose the same key word for a single well-known concept less than 20% of the time (Furnas et al., 1987). Also, sometimes the same word can have more than one distinct meaning. This will add to the difficulties of understanding the texts let alone analysing them.

Yet another problem involved in the analysis of textual databases is with their representation. Please refer to Section 4.4 for a detailed description of this.

Further, textual databases tend to be ‘noisy’ in nature, in that they could contain spelling and grammatical errors, which make their analysis even more complicated. All these reasons add up to the difficulties experienced in processing textual databases as opposed to dealing with numerical databases.

3.5 Summary

The contents in this chapter have been motivated by the need for a company to use the right information at the right time to gain a competitive edge. Many databases exist within the PDP that could serve as a good information source. However, we focused our attention on the much-ignored textual databases within the PDP. Some of these databases, obtained from various MNC's have been examined in great detail. In particular, the use of such databases, their quality, composition and the information potential in them had been described. Further, data mining tools that could be used to analyse such databases had also been presented.

Although at the current moment these databases do not have the 'best possible' input, they definitely have important content within them that would enhance decision-making within the PDP. However, concerted efforts need to be undertaken to improve the quality of data within these databases. Further, some of the difficulties involved in analysing textual databases as opposed to numerical databases have also been outlined.

In the following chapters, due to the availability of data for various models, the Call Centre databases would be studied in greater deal. In particular, various aspects of the classification of such databases would be examined.

CHAPTER 4

TEXT CATEGORIZATION: BACKGROUND

As the volume of information within corporate intranets continues to increase, there is a growing need for tools to help people better find, filter and manage these resources. Text categorization, the assignment of free text documents to one or more predefined categories based on their content, is an important component in many information management tasks such as real-time sorting of emails or files. In many contexts trained professionals are employed to categorise new items. This process is very time consuming and costly, thus limiting its applicability. Consequently there is an increasing interest in developing technologies for automatic text categorization. This chapter details the process of text categorization. In essence, it describes the steps needed to transform raw text into a representation suitable for the classification task, the text categorization methods used in this thesis, feature selection schemes as well as the performance measures used for categorization evaluation.

4.1 Introduction

Text categorization dates back to the early 60's, but only in the early 90's has it become a major sub-field of the information system discipline due to the increased applicative interest and availability of more powerful hardware. Until late 80's, knowledge engineering, which consists of manually defining a set of rules encoding expert knowledge on the classification of documents to given categories, was the most popular approach to text categorization. In the 90's, this approach had lost popularity in favour of the machine learning paradigm. The advantage of machine learning is the considerable savings in terms of manpower with an accuracy comparable to that achieved by human experts.

4.1.1 Need for the Text Categorization Study on 'Real Life' Datasets

Out of the various databases that we identified within the PDP in chapter 3, we focused our efforts on a call centre database. In particular we studied various issues regarding the classification of the call centre database provided by a multi-national company. At present the engineers at the company would manually read through every record and appropriately classify them. A typical product could easily have many different models and each model could have close to a few hundred to thousand records per month. As such, the amount of time and resources spent in manually classifying these records is enormous. At least two thirds of the total time taken could be saved if a semi-automated classification approach was employed. Hence, the use of classification analysis was attempted.

Most of the recent studies on text categorization, generally focus on the following issues; the choice of feature sets for text representation (Tan et al., 2002; Peters and Koster, 2003), use of transductive schemes to ease training label requirements (Taira and Haruno, 2002), use of external knowledge to enhance training (Benkhalifa et al., 2001), handling of multilingual documents (Chau and Yeh, 2002), issues of scalability (Damerau et al., 2002), improved/hybrid algorithms (Xu et al., 2002) as well as experimental studies comparing various algorithms (Lee and Kay, 2002 and Yang, 1999). Most of these studies use benchmark datasets such as the Reuters-21578, 20-Newsgroups or Medline, which are morphologically and syntactically well formed. However, as has been stressed by Frank Smadja in the report of Lewis and Sebastiani (2002) there is a need for substantial experiments with real data. In our study we investigate such a dataset. Results from many studies (Bekkerman et al., 2003) reveal that the performance of classification schemes widely depends on the dataset under study. In our application domain, our documents differ very much from these benchmark datasets. They have some unique characteristics which are not usually found in benchmark datasets. They can be outlined as follows:

- The average number of words in a typical document is less than 30 whereas for documents of Reuters-21578, for example, it is 129 words.
- The benchmark datasets are usually morphologically and syntactically well formed. As records in a call centre are more spontaneously created and are informal documents, they require us to cope with some amount of jargon, misspellings and grammatical errors.
- Non-conformance to linguistic standards is yet another characteristic pronounced in such call centre databases as opposed to benchmark datasets.
- Generally benchmark data sets have a very large number of documents and categories. Typical call centre databases, for example, pertaining to a particular model of a consumer electronic product, has much fewer records, in the order of a few thousands.
- The call centre databases exhibit multiple viewpoints. The information contained in the customer’s call is so rich that different kinds of knowledge can be extracted from analyses with different viewpoints. For example;

- The product designer might be interested in determining the complaints obtained for the component that he is interested in designing.
- The project manager might choose to find out whether the nature of the call was merely assistance seeking, technical or maybe enquiry related. This will provide him with a better understanding of the actual problem faced by the consumer.
- The service personnel, planning for logistics at a service-centre might be interested in knowing if a particular call has been escalated to the service-centre.

Given the aforementioned differences, the conclusions derived from studies using benchmark datasets cannot be directly applied to call centre datasets. Hence it would be necessary to ascertain the usefulness of some of the commonly used schemes for text categorization of the call centre records.

It must be pointed out that there have been some textual mining applications that have been carried out on a call/help centre database (Chu-Carroll and Carpenter, 1999 and Nasukawa and Nagano, 2001). In Nasukawa's and Nagano's work, the focus of the research was to develop a generic text mining system for information extraction. Linguistic processing was carried out together with manual annotation of relevant entities within the text. Analysis was carried out to determine, frequency and relationships between these entities. In their study they had used text from a Japanese help centre. Chu-Caroll and Carpenter (1999) focussed on the routing of calls coming to a call centre by analysing the initial utterances of the customer. Information retrieval techniques were used to assess similarity of originally transcribed documents with incoming customer calls to determine where these calls should be routed.

In this study the focus is on the application of classification techniques for categorising call centre records. More importantly, the extensive experimental studies conducted, covering various aspects of the textual classification of the call centre records, differentiate this work from the others.

4.2 Learning Task

Text categorization (Sebastiani, 2002) is the task of assigning a Boolean value to each pair $\left\langle d_j, c_i \right\rangle \in D \times C$ where *D* is a domain of document and $C = \{c_1, c_2 ....., c_m\}$ is a set of predefined categories. A value of *T* assigned to $\left\langle d_j, c_i \right\rangle$ indicates a decision to file d_j under c_i, while a value of *F* indicates a decision not to file d_j under c_i. Various settings could arise based on the different combinations of d_j under c_i.

4.2.1 Binary Setting

The binary setting is the simplest formulation of the learning task. In a binary setting situation there are exactly two classes. This implies that the class labels c_i has only two possible values. If we let these values, for notational convenience, be +1 and –1, then $C = \{+1, -1\}$. The binary setting is a very general case and can be used for multi-class and multi-label settings.

4.2.2 Multi-Class Setting

In some classification tasks, there are sometimes more than two classes. For example, an agent at the call centre might need to route a call to one of five customer representatives, depending on the problem under question. In this case, each document would be assigned to only one of the five classes, $C = \{1,2,3,4,5\}$. In this study, we deal with a **multi-class setting**. Some classification algorithms can handle multi-class settings directly whereas others basically split it into many binary class setting problems. Decision Trees and the multi-class formulation of Support Vector Machines (Weston and Watkins, 1998) are examples of the former whilst the one-against-the-rest (Scholkopf et al., 1995) or the pair-wise classification (KreBel, 1999) strategies are instances of the latter.

4.2.3 Multi-Label Setting

Unlike in the multi-class setting where there is a one-to-one correspondence between a document and its class, in the multi-label setting, for a fixed number of categories, each document can be in multiple, exactly one, or no category at all.

4.3 Classification Methods

There are many classification methods that have been used for the purpose of text categorization. Generally speaking classifiers fall into two categories: generative models which initially estimate class conditional densities P(document/class) and discriminant models which directly estimate the posterior probabilities P(class/document). The naïve Bayes Model (Lewis, 1998) is a popular generative categorization model whereas amongst discriminative techniques, Support Vector Machines (Burges, 1998) is an example. A comprehensive review of the different document categorization schemes is provided by Sebastiani (2002). In the following subsections, three of the classification methods studied in this thesis, namely; 1) Naïve Bayes Classifier, 2) Decision Tree C4.5 and 3) Support Vector Machines, are detailed.

4.3.1 Naïve Bayes Classifier (NB)

The basic idea in the Naïve Bayes approach (Witten and Frank, 2000) is the use of joint probabilities of words and classes given a set of records. This approach is based on the Naïve Bayes Theorem, which can be expressed as;

$$P(c_j \mid \boldsymbol{d}) = \frac{P(c_j)P(\boldsymbol{d} \mid c_j)}{P(\boldsymbol{d})} \tag{4.1}$$

where,

$P(c_j \mid \boldsymbol{d})$ is the posterior probability of observing class c_j given document vector, ***d***,

$P(\boldsymbol{d} \mid c_j)$ is the prior probability of observing document vector ***d*** given occurrence of class c_j,

$P(c_j)$ is the probability that a randomly picked document belongs to class c_j,

$P(\boldsymbol{d})$ is the probability that a randomly picked document has vector ***d*** as its representation.

The denominator in the above equation does not differ between categories and can be left out. The estimation of $P(\boldsymbol{d} \mid c_j)$ in equation (4.1) can be quite problematic. Hence in order to alleviate this problem, it is common to make the assumption that any two coordinates of the document vector, when viewed as random variables, are statistically independent of each other. This independence assumption allows us to compute the required probability as:

$$P(\boldsymbol{d} \mid c_j) = \prod_{k=1}^{m} P(w_k \mid c_j) \tag{4.2}$$

where *m* is the number of distinct words, *w*, in the document collection.

The Naïve Bayes approach can be used for binary as well as continuous variables. Continuous variables are usually handled by assuming that they have a 'normal' distribution. Dependencies between attributes inevitably reduce the power of Naïve Bayes to discern what is going on. Further, the normal-distribution assumption for numeric attributes is another restriction on Naïve Bayes. In this regards, the procedures of "kernel density estimation" that do not assume any particular distribution for the attribute values can be used.

4.3.2 C4.5

The C4.5 algorithm works by generating a classifier in the form of a decision tree, a structure that is either a leaf, indicating a class, or a decision node that specifies some test to be carried out on a singular attribute value, with one branch and subtree for each possible outcome of the test (Quinlan, 1993). The decision tree is generated by a systematic choice of which attribute to use as a splitting criterion. At every stage of the tree, the attribute that provides the most information after a split is chosen to be a decision node. This information measure is computed using an attribute-specific quantity called the Gain Ratio Criterion. The equations below show how it is derived.

Entropy, the expected information needed to classify a given document set, *D,* of interest, is given as,

$$Entropy(D) = -\sum_{j=1}^{C} p(D, j) \times \log_2\left(p(D, j)\right) \tag{4.3}$$

where *C* is the total number of classes, *j* the *j-th* class and *p(D,j)* the proportion of documents in *D* belonging to the *j-th* class. The concept of entropy is reused in

Sections 6.1.6 and 8.2.1 when measures for term weighting and attribute selection are presented, respectively.

The information gain of an attribute is the expected reduction in entropy caused by the partitioning on this attribute and is given as:

$$InformationGain(D,T) = Entropy(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times Entropy(D_i) \tag{4.4}$$

where *T* refers to an attribute of interest, *i* refers to the *i-th* value of the attribute *T*, *k* refers to the total possible values of attribute *T*, |*D*| is the number of documents in *D* and |D_i| is the number of documents in *D* that has the *i-th* value for the attribute *T*.

The entropy of the of the split for a particular attribute *T* is given as:

$$Split(D,T) = -\sum_{i=1}^{d} \frac{|D_i|}{|D|} \times \log_2 \frac{|D_i|}{|D|} \tag{4.5}$$

where *d* is the number of daughter nodes and |D_i| is the number of documents in the *i-th* daughter node. This takes into account the number and size of daughter nodes into which an attribute splits the dataset, disregarding any information about the class.

And the gain ratio is given as:

$$Gain\ Ratio = \frac{Gain(D,T)}{Split(D,T)} \tag{4.6}$$

At every stage of the decision tree, the gain ratio criterion is calculated for all the attributes. It measures the desirability of an attribute *T*, within the dataset (or sub-dataset) *D*. The attribute with the highest gain ratio criterion will be chosen to further split the node, and if necessary, the process will be repeated at each of the branches of this new decision node. The decision tree building process will stop when all possible tests on a sub-dataset have zero gain. When that is not possible, then the stopping criterion will be when the classification error within each leaf node is minimised.

The phenomenon of overfitting can take place, especially when there is noisy data. In such cases, the classifier tree that is generated, in an attempt to fit the eccentricities of the dataset, becomes overly complex. Generalisability is lost, causing poor accuracy when the model that was built is being tested on unseen data. To deal with this, C4.5 has the option of pruning, which aims to reduce the number of misclassified records by reducing the complexity of the generated tree.

There are two main ways to prune a decision tree. The first is subtree replacement, which investigates whether replacing a subtree with a leaf node will reduce the error rate. The second is subtree raising, which attempts to raise a decision node one level up the hierarchy of the tree, also with the aim of reducing the error rate.

4.3.3 Support Vector Machines (SVMs)

Kernel methods give a systematic and principled approach to train learning machines. Support Vector Machines (SVMs) (Burges, 1998 and Campbell, 2000) are the most well known learning systems based on kernel methods. In order to illustrate the learning approach of the SVMs, we shall first consider a binary classification problem. The motivation for considering binary classifier SVMs comes from theoretical bounds on the generalisation error (Campbell, 2000). From Statistical Learning theory (Vapnik, 1998) the following could be said:

a. The generalization error bound is minimized by maximizing the margin, γ, i.e. the minimal distance between the hyperplane separating the two classes and the closest datapoints to the hyperplane (Fig. 4.1).

b. The upper bound on the generalization error does not depend on the dimension of the space. This point explains why we use kernel representation of data to do a non-linear projection into a higher dimension space where it is easier to separate the two classes of data.

4.3.3.1 Binary Classifier (Separable Case)

Let us consider a set of separable datapoints $\mathbf{x_i}$ (i=1,…,m) having corresponding labels y_i = +1 or -1 and let the decision function be:

$$f(x) = \text{sign}\,(\mathbf{w.x} + b) \tag{4.7}$$

We can then define a hyperplane such that **w.x** + b = 1 for the closest points on one side and **w.x** + b = -1 for the closest on the other. The separating hyperplane is thus given by **w.x** + b = 0. The normal vector to the separating hyperplane is **w**/||**w**|| and the margin, γ = 1/||**w**||. Based on Vapnik's statistical theory, we try to maximize the margin. To maximise the margin the task is therefore:

$$\min\left[\frac{1}{2}\|\mathbf{w}\|^2\right] \tag{4.8}$$

subject to the constraints:

$$y_i(\mathbf{w}\cdot\mathbf{x_i} + b) \geq 1 \qquad \text{for all } i \tag{4.9}$$

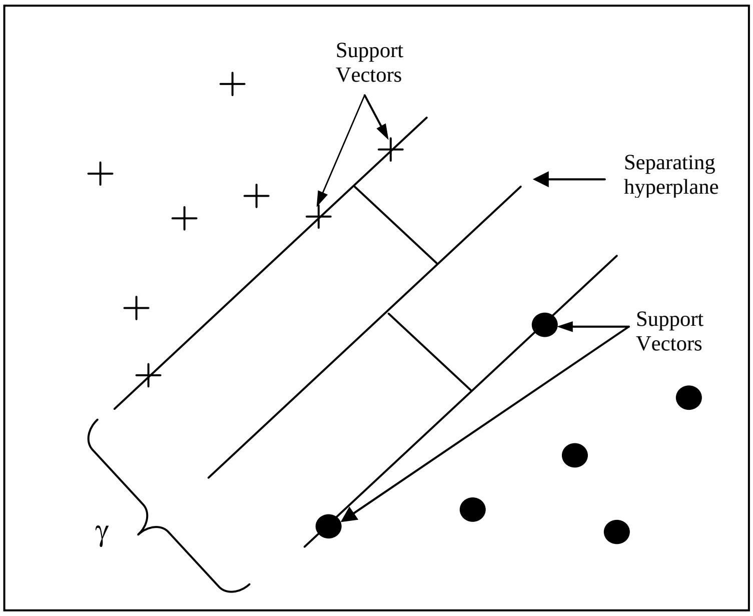


Figure 4.1. Binary Classification. '+' denotes a label of '+1' for the training example and the dark circle denotes a label of '-1'

This leads to a minimization problem where the primal objective function is given by:

$$L = \frac{1}{2}\,(\mathbf{w.w}) - \sum_{i=1}^{m} \alpha_i \,(y_i\,(\mathbf{w.x}_{\mathrm{i}} + b) - 1) \qquad (4.10)$$

where α_{i} are Lagrange multipliers and hence $\alpha_{\mathrm{i}} \geq 0$.

Taking derivatives with respect to *b* and **w** gives:

$$\sum_{i=1}^{m} \alpha_i\, y_i = 0 \qquad (4.11)$$

$$\mathrm{w} = \sum_{i=1}^{m} \alpha_i\, y_i\, \mathbf{x_i} \qquad (4.12)$$

and resubstituting these into (4.10) gives:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i\, \alpha_j\, y_i\, y_j\, (\mathbf{x_i} \,.\mathbf{x_j}) \qquad (4.13)$$

which must be maximized with respect to α_{i} subject to the constraints:

$$\alpha_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{m} \alpha_i\, y_i = 0 \qquad (4.14)$$

The above-constrained quadratic programming problem would provide an optimal separating hyperplane with a maximal margin if the data is separable, as mentioned in Section 4.3.3 (bullet (a)). Kernel substitution is then carried out to project the data into a higher dimensional feature space where the two classes of data are more readily separable. Since the datapoints in the Wolfe dual only appear inside an inner product, we can achieve the mapping by replacing the inner product:

$$\mathbf{x_i} . \mathbf{x_j} \rightarrow \phi(\mathbf{x_i}) . \phi(\mathbf{x_j}) \qquad (4.15)$$

Various choices of kernels are available which achieve this mapping. One possibility, is the Gaussian (RBF) kernel given by,

$$\mathrm{K}(\mathbf{x_i}, \mathbf{x_j}) = \exp(-0.5\|\mathbf{x_i} - \mathbf{x_j}\|^2 / \sigma^2) \qquad (4.16)$$

So for a given choice of kernel, the learning task therefore involves maximization of the objective function:

$$\mathrm{W}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \, \alpha_j \, y_i y_j \; K(\mathbf{x_i}, \mathbf{x_j}) \qquad (4.17)$$

The associated Karush-Kuhn-Tucker (KKT) conditions are:

$$y_i(\mathbf{w}.\mathbf{x_i} + b) - 1 \geq 0 \text{ for all i}$$

$$\alpha_i \geq 0 \text{ for all i}$$

$$\alpha_i \, (y_i(\mathbf{w}.\mathbf{x_i} + b) - 1) = 0 \text{ for all i} \qquad (4.18)$$

which are always satisfied when a solution is found.

The test examples are evaluated using a decision function given by the sign of:

$$f(\mathbf{z}) = \sum_{i=1}^{m} y_i \, \alpha_i \, K(\mathbf{x_i}, \mathbf{z}) + b \qquad (4.19)$$

Since the bias, *b*, does not feature in the above dual formulation it is found from the primal constraints :

$$\boldsymbol{b} = -\frac{1}{2}\left[\max_{\{i|y_i=-1\}}\left(\sum_{j\in\{SV\}}^{m} \boldsymbol{y}_j \alpha_j \boldsymbol{K}(\mathbf{x_i},\mathbf{x_j})\right)\right] + \min_{\{i|y_i=+1\}}\left(\sum_{j\in\{SV\}}^{m} \boldsymbol{y}_j \alpha_j \boldsymbol{K}(\mathbf{x_i},\mathbf{x_j})\right) \qquad (4.20)$$

using the optimal values of α_j. When the maximal margin hyperplane is found in the feature space, only those points which lie closest to the hyperplane have $\alpha_i > 0$ and these points are the support vectors, which means that the representation of the hypothesis is solely by these points and they are the most informative patterns in the data. Points that are not support vectors do not influence the position and orientation of the separating hyperplane and do not contribute to the hypothesis.

4.3.3.2 Soft Margin for Non-Separable Case

Most data sets contain noise and an SVM can fit to this noise leading to poor generalisations. The effect of outliers and noise can be reduced however, by introducing a soft margin. There are two schemes that are currently used : L_1 error norm and the L_2 error norm. The L_1 error norm will be mentioned as this is implemented in the LIBSVM software that is used in this study. For the L_1 error norm equation, the condition in equation (4.9) is relaxed by introducing a positive slack variable ξ_i giving rise to:

$$y_i(\mathbf{w}\cdot\mathbf{x_i} + b) \geq 1 - \xi_i \qquad (4.21)$$

and the task is now to

$$\min\left[\frac{1}{2}\mathbf{w}\cdot\mathbf{w} + \boldsymbol{C}\sum_{i=1}^{m}\xi_i\right] \qquad (4.22)$$

This is readily formulated as a primal obective function :

$$\boldsymbol{L}(\boldsymbol{w},\boldsymbol{b},\alpha,\xi) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + \boldsymbol{C}\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left[\boldsymbol{y}_i(\mathbf{w}\cdot\mathbf{x_i} + \boldsymbol{b}) - 1 + \xi_i\right] - \sum_{i=1}^{m}\boldsymbol{r}_i\xi_i \qquad (4.23)$$

with Lagrange multipliers $\alpha_i \geq 0$ and $r_i \geq 0$. Taking derivatives with respect to **w**, *b* and ξ give :

$$\begin{aligned} \frac{\partial \boldsymbol{L}}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \alpha_{\boldsymbol{i}} \boldsymbol{y}_{\boldsymbol{i}} \mathbf{x}_{\mathbf{i}} = 0 \\ \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{b}} &= \sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \alpha_{\boldsymbol{i}} \boldsymbol{y}_{\boldsymbol{i}} = 0 \\ \frac{\partial \boldsymbol{L}}{\partial \xi_{\boldsymbol{i}}} &= \boldsymbol{C} - \alpha_{\boldsymbol{i}} - \boldsymbol{r}_{\boldsymbol{i}} = 0 \end{aligned} \tag{4.24}$$

Resubstituting these back in the primal objective function, we obtain the same dual function as in equation (4.13), subject to:

$$\begin{aligned} &0 \leq \alpha_i \leq C, \\ &\sum_i \alpha_i y_i = 0 \end{aligned} \tag{4.25}$$

Once the α_i's are obtained the other primal variables **w**, *b*, ξ and *r* can be easily determined using the KKT conditions.

4.3.3.3 Multi-Class Classifier

The above formulation is valid for a binary classifier. However, the real-life datasets involve multi-class classification and a lot of different schemes have been proposed to do this. One set of schemes decompose the multi-class problem into many binary problems. Two popular approaches implementing this sort of scheme would be the "one-against-all" and "one-against-one". The other set of schemes is to formulate the problem using all classes at once (Weston & Watkins, 1998). However, the latter scheme involves a lot more effort and was not shown to yield better results and hence will not be discussed below.

One-against-all

In the "one-against-all" approach, *k* SVM models are built, where *k* is the number of classes. The *i-th* SVM is trained with all the examples in the *i-th* class as positive labels, and all other examples as negative labels.

One-against-one

This method constructs *k*(*k-1*)/2 classifiers where each one is trained on data from two classes. In this scheme, classification follows a voting strategy: each binary classification is considered to be a voting. Votes are cast for all data points. In the end, a point is designated to be in a class with maximum number of votes. This approach has been found to be effective compared to the "one-against-all" approach (Weston and Watkins, 1998; Platt et al., 2000) and as such it has been implemented in LIBSVM.

4.4 Document Representation

Unlike numerical databases, one has to grapple with the problem of representation of the documents when it comes to textual databases. The important question would be: *How do we convert the words into features that could be input into a conventional classification algorithm?* A popular approach that has been used is the vector space model. This approach lends itself well to text categorization. It represents the 'units of content' of a document as a vector. Each unit of content could be a single word, phrase, part of a word or even a 'concept'.

Every 'content unit' in the document collection is represented by one component of these vectors. Consider a document collection, where the units of content are single terms. Assume there are only three distinct terms, α, β and γ, in a document. Then we have a three dimensional vector. Say in the first document, $D_{1,}$ term 1 occurs only once and term 3 occurs only twice. This document could be represented by the vector (1,0,1), which is a vector showing the existence of a word. Another possible representation is (1,0,2) which accounts for the term frequency of the words. Such variations are generally referred to as various **weighting schemes**. The effect of such a representation would be explored in detail in the later chapters. In the following sub-sections however, a further description on the 'units of content' is provided.

4.4.1 Content Units

4.4.1.1 Single Terms

At this level, each content unit is a single word. Hence a word sequence in a document is projected into a bag of words. Such a scheme is called the bag-of-word, BOW, approach. In many cases, words are meaningful units of little ambiguity even without considering context. While synonyms exist, it is assumed that they have little impact on the document representation as a whole. This is especially true if documents belong to the same domain whose contents reflect a specific subject matter. Clearly this transformation leads to a loss of information. However, more sophisticated representations have not really shown consistent and substantial improvements. In text classification, it has been found that the word-based representation is very effective (Lewis, 1992a) and they serve as the basis for most work in text classification. David Lewis undertook a major study of the use of noun phrases for statistical classification as part of his Ph.D. thesis (Lewis, 1992b). He reviewed the existing literature on the use of syntactic phrases and found that most studies had not been able to demonstrate much improvement over word-based indexing. Some reasons that were cited for poor performance of phrasal representations include: a) an uneven distribution of feature values and b) many redundant features. A further study comparing single word representation to other types of representations yielded a similar conclusion (Scott and Matwin, 1999).

4.4.1.2 Sub-Word Level

Character n-grams are the most popular representation at the subword level. They use strings of *n* characters as the basic building blocks. For example, the bi-gram representation of the word ‘car’ is ‘ca’ and ‘ar’. The *n*-gram representation naturally models similarity between words. Example even though ‘cartridge’ and ‘cartridges’ are different words, they share most of their sub strings. Although this is a desirable similarity, there are instances, such as ‘computer’ and ‘commuter’ which again would show close similarity when the actual words are different. The n-gram representation provides some robustness against spelling errors. Neumann and Schmeier (1999) provide some experimental results on the use of *n*-grams.

4.4.1.3 Phrases

Phrases can be taken to be any combination of words that could be consecutively occurring or occur within a window of words. They are generated basically via statistical means or through the use of linguistic tools. Sequential rule mining is one of the many analysis techniques that has been used to generate phrases statistically. It can generate frequently occurring sets of words that are collocated within a specific window of words (Ahonen-Myka et al., 1999). With regards to use of syntactic structures, noun phrases are the most commonly used phrase structures (Lewis, 1992a; Neumann and Schmeier, 1999; Basili et al., 1999).

4.4.1.4 Concepts

Text classifiers can work optimally if they can capture the meanings within the documents. Various attempts have been made to include the context and capture semantic relationships in documents. The use of lexicons is a simple way of ensuring similar words are represented as a single unit (Fukumoto and Suzuki, 2002). Further, the use of taxonomy and a fixed vocabulary is another scheme that is employed to capture the semantics within the documents (Wang et al., 2003). This however, requires a lot of manual effort in generating the knowledge base in the first place. An automated scheme that is widely used in Information Retrieval and recently for text classification tasks is latent semantic indexing. By using the singular value decomposition (SVD) technique, it aims to generate semantic categories by transforming the original attributes into fewer new dimensions by combining attributes that express a similar meaning (Dumais, 1992). **This scheme would be further explored in Chapter 7**.

Despite considerable attempts to introduce more sophisticated techniques for document representation, like the ones that are based on higher order word statistics (Caropreso et al., 2001), NLP (Jacobs, 1992; Basili et al., 1999), “string kernels” (Lodhi et al., 2002) and even representations based on word clusters (Baker and McCallum, 1998), the simple minded independent word-based representation remains very popular. Indeed, to-date the best categorization results for the well-known Reuters-21578 and 20 newsgroups datasets are based on the bag-of-word representation (Dumais et al., 1998; Weiss et al., 1999; Joachims, 1997). Further, a substantial advantage of word-based representation is its simplicity. Given its simplicity, widespread usage as well as its effectiveness for text classification task, **in this study we adopt the use of single terms as the basic content units.**

4.5 Feature Selection

A unique characteristic of textual databases is the number of features (unique terms) that it contains. Even a small document collection of about 500 documents, within a specific domain, can easily have more than 1000 features. In some benchmark datasets like Reuters, the number of features can easily go up to tens of thousands. However, not all these features are relevant and in fact the inclusion of some of these actually causes noise and can result in a poorer performance. To address this problem, many feature selection schemes have been introduced, in the literature. These schemes can be broadly classified as embedded, filter based and wrapper based. The wrapper based methodology popularized by Kohavi and John (1997) treats feature selection as a wrapper around the induction process. In fact the learning machine is considered a perfect black box and the method lends itself to the use of off-the-shelf machine learning software packages. Embedded methods as the name implies incorporate the variable selection as part of the training process. Hence, these methods are more efficient than the wrapper approaches in that they make better use of the available data. Decision Trees is an example of one such method. Filter based approaches were termed as such (Rogati and Yang, 2002) because such methods filter out irrelevant attributes before induction occurs. Some filter based approaches will be presented in Chapter 8 whilst Chapter 9 proposes a novel DoE based feature selection method.

4.6 Performance Measures

There are many commonly used performance measures for evaluating text classifiers. In evaluating the performance of a text classifier, the purpose for which the output is to be used is crucial in choosing the appropriate evaluation methods (Lewis, 1991). The following subsections outline the performance measures in greater detail. However, prior to that, the contingency table as shown in Table 4.1 is introduced based on which estimators for these measures can be computed.

Table 4.1: Contingency table

	Yes is Correct	**No is Correct**
Decides Yes	a	b
Decides No	c	d

Each entry into the table specifies the number of decisions of the specified type. For instance c is the number of times the system decided ‘no’, when ‘yes’ was in fact the correct answer.

4.6.1 Classification Accuracy Rate

A very common measure in machine learning is the accuracy rate. From the contingency table, this accuracy value is given by:

$$Accuracy\ rate = \frac{a+d}{a+b+c+d} \tag{4.26}$$

In some applications (classification of web pages), the number of negative examples vastly outnumber the positive examples. This can have a misleadingly high accuracy value. This is partly due to the equal weightage assumed for false negatives and false positives in the above formula. Such situations can be handled using an asymmetric cost matrix.

4.6.2 Asymmetric Cost

In some applications, predicting a particular decision correctly provides the user with greater utility than predicting the other decision. Under these circumstances, this preference can be incorporated in the performance measure via a cost matrix as shown in Table 4.2.

Table 4.2: Cost Matrix

	Yes is Correct	**No is Correct**
Decides Yes	C_1	C_2
Decides No	C_3	C_4

The corresponding entries in the cost matrix and the contingency table are multiplied to provide a cost-weighted measure as such:

$$Cost\ Weighted\ Accuracy = \frac{C_1 a + C_4 d}{C_1 a + C_2 b + C_3 c + C_4 d} \qquad (4.27)$$

4.6.3 Recall and Precision

Recall and precision are widely used in information retrieval where they measure the proportion of relevant documents retrieved and the proportion of retrieved documents, which are relevant, respectively. For text categorization purposes, these measures can be defined as follows (Joachims, 2002):

Recall is the probability that a document with a '*yes*' label is actually classified as a '*yes*'. From the contingency table, it can be estimated using the following formula

$$Recall = \frac{a}{a+c} \qquad (4.28)$$

Precision is the probability that a document classified as '*yes*' is indeed classified correctly.

$$Precision = \frac{a}{a+b} \qquad (4.29)$$

In as much as Recall and Precision are good classification performance measures, the need to consider two scores makes comparison between alternative schemes difficult. This has caused researchers to seek single measures to characterize performance. The three-point average recall, which gives the average precision obtained at recall values of 20%, 50% and 80% and the eleven point average recall, which gives the average precision obtained at recall values of 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% are examples. The values can be plotted to provide a precision/recall curve.

4.6.4 F$_\beta$-measure

A single point measure, which includes recall and precision, is the F$_\beta$-measure. The F$_\beta$-measure can be written as follows:

$$F_\beta = \frac{(1+\beta^2)xPrecision\,x\,Recall}{\beta^2 Precision + Recall} \qquad (4.30)$$

The parameter β is usually set to 1 giving equal weight to precision and recall. Summarizing the whole precision recall curve using a single value leads to a loss of information.

4.6.5 Micro- and Macro-Averaging

Generally, in multi-label problems (eg. when a document is labelled with more than a single class), one is usually interested in how well all the labels can be predicted, not only a single one. Therefore, there is a necessity to determine how the results of *m* binary tasks can be averaged to obtain a single estimate. Micro-averaging and macro-averaging are two commonly used approaches.

Macro-averaging corresponds to computing an averaged performance measure from the performance measures obtained from *m* binary performance measures. For recall, this implies:

$$Recall^{Macro} = \frac{1}{m}\sum_{i=1}^{m} Recall(i) \tag{4.31}$$

Macro-averaged scores are more influenced by the performance on rare categories.

The micro-averaging method averages the entries in the contingency tables. For each cell of the table the arithmetic mean is computed leading to an averaged contingency table with elements a_{avg}, $b_{avg,}$ c_{avg} and d_{avg}. Hence a micro-averaged recall performance measure is computed as:

$$Recall^{Micro} = \frac{a_{avg}}{a_{avg} + b_{avg}} \tag{4.32}$$

Micro-averaged scores (recall, precision and F-measures) tend to be dominated by the classifier's performance on common categories. For the situation when each document is exactly assigned 1 of *m* binary categories, the micro-averaged recall measure becomes similar to the classification accuracy measure.

In this work, the accuracy rate (equation 4.26) is used as the performance measure. This stems from the fact that this measure serves our needs well. We are only interested in finding out how many documents are correctly classified and this measure determines exactly that.

Further, the documents obtained from the multinational company have been labelled with a single category only. This is especially true for the Call Centre datasets that are studied, since only a single problem is mentioned in almost all documents. The dynamics of the information gathering mechanism engenders this. For example, the structure of the question-answer approach, which is directed towards solving a single problem, the attitude of the customer to call when he sees the first problem occur, the attitude of the engineer to assign a label to the more important problem encountered, produces the situation of a multi-class learning task setting. Thus, under these circumstances, the accuracy rate has been used as the performance measure in this thesis.

4.7 Summary

This chapter provided the necessary background required for text categorization. The need for the categorization of a 'real life' dataset such as the Call Centre Dataset was explained. Three different classifiers that would be used were described: namely Naïve Bayes, Decision Trees and Support Vector Machines. The different schemes for representing the text, the essence of feature selection in dealing with textual problems as well as the measure for analyzing classification performance was also presented. For the studies carried out in the thesis, single words were used as the basic document representational unit. With regards to performance measure, classification accuracy would be used.

CHAPTER 5

DETERMINING OPTIMAL SETTINGS FOR TEXTUAL CLASSIFICATION

In the design of any text classification system, it is important to understand the factors that affect the performance of the system so as to optimize it. Many factors affect the performance of such a system. Amongst others, these factors would include, the type of classifier used, the weighting given to the words in the document, the type of preprocessing carried out as well as the dataset under consideration. Each of these factors would influence the final accuracy either solely by itself or sometimes due to interaction with other factors. In this chapter, the effect of five different factors and their interaction on the classification accuracy is studied. The different factors studied and their various levels considered are described in detail. An Analysis of Variance (ANOVA) is carried out which allows the contribution of each factor, to the classification accuracy, to be determined. Mean and interaction plots are provided to aid visualization of the importance of the various factors. Further, the notion of designable and non-designable factors is introduced to decide the factors over which the system would be optimized. Finally, the optimal settings of the various parameters are identified. These settings are subsequently used for other experiments carried out in the forthcoming chapters.

5.1 Introduction

As mentioned earlier, the main function of a call centre is to serve as an avenue for a company to be closer to its customers and hence provide better service to them. Although originally conceived as a help desk, due to increasing complexity of products and their rapidly changing nature, call centres being dynamic in nature, become an important source of firsthand information about customers' view of a company's products. This information is extremely important since it provides an avenue for understanding the customer opinions and hence allows the company to react accordingly to maximize customer satisfaction. Furthermore, fast feedback from the analysis of call centre databases can help to quickly identify potential problems so as to address them appropriately (Brombacher, 2000). This creates immense opportunities for timely product and process improvement. Usually knowledge engineers are employed at companies to manually study these textual records and extract relevant information. Such efforts can be extremely time-consuming and tedious. In the MNC which provided the datasets, it is estimated that every three out of five working days would need to be spent by the knowledge engineer to extract the required feedback/information from about 500 records. This hampers the engineer from working on his job of updating the knowledge base of the calls received. Hence to speed up the process, an automated textual classification system was sought to facilitate fast feedback and product/process improvement.

The development of such a system would entail the understanding of various factors affecting its performance, in terms of classification accuracy, so as to be able to tune the system to obtain higher accuracies. Five different factors affecting classification performance were studied. Details of the factors studied as well as the dataset statistics are provided in Section 5.2. The classifiers used in the study have been trained over ten thousand times, in total, to determine the accuracy values for the various factor combinations. Experiments have been carried out at all possible factor combinations. Section 5.3 discusses the results in which the independent as well as the interaction effects amongst factors are determined by carrying out an Analysis of Variance (ANOVA). In the final section, the determination of the optimal settings is explored.

5.2 Factor Settings

Five different factors that were believed to be important in influencing the classification accuracy were investigated. Each factor was varied from 2 to 4 settings. Table 5.1 provides the details of the factors studied.

Table 5.1: Factors investigated

Description	Short-Form	No. of Levels
Preprocessing	*Process*	4
Algorithm	*Method*	4
Information field	*Field*	3
Format of database	*Data_format*	3
Document Representation	*DocumRep*	2

5.2.1 Preprocessing

Studies have shown that preprocessing can affect classification accuracies (Yang, 1994). Having removed 'unwanted' text such as delimiters and alphanumeric texts from the call centre records, the following stages of preprocessing was carried out:

- Zero processing (original text in its raw form)
- Stopword removal
- Stemming

- Full Processing (Stopword removal + Stemming)

Removal of non-informative words (stop-words) is a common technique in text indexing and retrieval (Van Rijsbergen, 1979) to improve the accuracy of the results and to reduce the redundancy of the computation. Stop words are deemed relatively meaningless in terms of document relevance and are not stored in the index. These words are often defined by a “stop-list” which typically consists of about 200 to 400 words, including articles, prepositions, conjunctions, and some high-frequency words. Some examples of common stop words include words like *are*, *the*, *from* and *could*. These words have important semantic functions in English, but rarely contribute useful information. A stop list has the advantage that it reduces the size of the document representation matrix and provides a succinct description of the information within the document. In this study, the stop list suggested by Van Rijsbergen (1979) was used with modifications to exclude the negating words such as *not, can’t, couldn’t, neither, nor* and other similar words from the list. These negating words might be important in the Call Centre dataset which comprises of failure records with phrases such as *not working, can’t function* and so on.

Another common preprocessing step in dealing with textual databases is stemming. Stemming usually refers to a simplified form of morphological analysis by simply truncating a word. For example, laughing, laughs, laugh and laughed are all stemmed to “laugh”. Common stemmers are the Lovins and Porter stemmers, which differ in the actual algorithms used for determining where to truncate words (Lovins, 1968 and Porter, 1980). Two problems exist with truncation stemmers. Firstly, they conflate semantically different words (for example, gallery and gall may both be stemmed to gall-). Secondly, the truncated stems can become unintelligible to users (such as when “gallery” gets represented as “gall-“). They are also less appropriate or applicable for morphologically-rich languages. Nevertheless, the use of stemming can help to provide a more concise document space representation. Further, stemming is so widely regarded as useful that they are almost universally used in text classification experiments (Scott, 1999). In this study, Porter’s Algorithm (Porter, 1980) was used for word stemming.

5.2.2 Information Field Type

Generally, the extraction of useful information would require the reading of the textual input and the consequent extraction of key points that the user of the database might deem useful for his requirements. For example, from the record given below;

“*UKBUCHANLS 1/13/2001 6:35:41 PM c/cld needing help with his connections to his video*.”,

different types of information could be extracted. This might include items like the date and time of the call, the agent who attended to the call, the nature of the call (whether it is an assistance call or enquiry), the problem area of concern (connections problem), the possibility of escalation of the problem to the service centre (not likely) and so on. In this study, the following information fields were considered:

- Problem Area
- Escalation
- Call-type

Table 5.2 provides a description of the information fields. As mentioned above, the type of information extracted depends on the user of the database. For example, a project manager might choose to find out whether the nature of the call was merely assistance seeking, technical or maybe enquiry-based. A product engineer however, might want to determine if the problem occurring was related to the picture tube or some connections within the TV, whereas a service personnel planning for logistics at a service centre might be interested in knowing if a particular call has been escalated to the service centre. As such, different information fields were studied for classification purposes. Each information field is an output/response to be modelled. The distribution of the classes is provided in Appendix B.

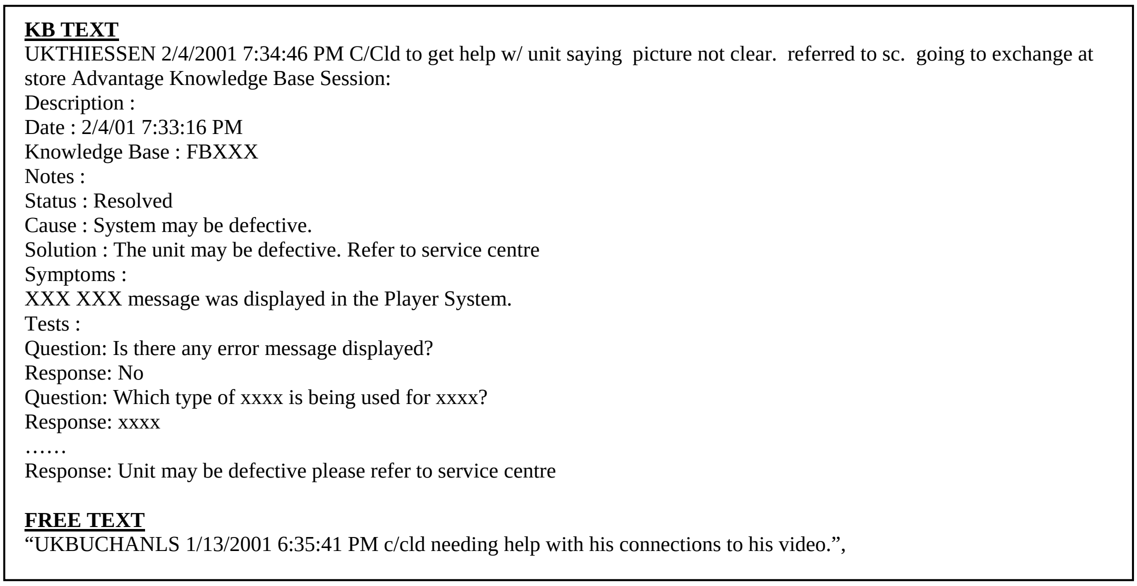
Table 5.2: Information field descriptions

Information Field	Description	Number of classes	Labels
Problem Area	Identifies what the major problem is	10	CD, accessories, connections, demo, operations, others, tape, tuner, speakers, unknown
Call Type	Identifies the nature of the call	7	Assistance, Enquiry, Complaints, Perception, Technical, Possible Technical, Unknown
Escalation	Determines whether a call has been escalated to the service centre	3	Yes, No, Not Sure

5.2.3 Format of Dataset

Call Centre datasets generally consist of 'free-text' and a 'knowledge base (KB) text', as shown in Figure 5.1. The format of the text has been preserved but the contents have been fabricated for the sake of confidentiality. The knowledge base is used when a known customer complaint is encountered while a call is recorded solely as free text if its content has never been previously encountered. Each of these texts has distinct characteristics. The average word length for the 'KB-text' is 83.6 while the average word length for the 'free-text' is 12.5. The 'free-text' is a lot more unstructured than the ‘KB-text’, which includes a small part of free-text and a large part of formatted text. Part of the formatted text also has a question and answer approach. As such, given these differences, the following combinations were investigated:

- Free-text
- KB-text
- Both (Free + KB-text)



```
KB TEXT
UKTHIESSEN 2/4/2001 7:34:46 PM C/Cld to get help w/ unit saying  picture not clear.  referred to sc.  going to exchange at
store Advantage Knowledge Base Session:
Description :
Date : 2/4/01 7:33:16 PM
Knowledge Base : FBXXX
Notes :
Status : Resolved
Cause : System may be defective.
Solution : The unit may be defective. Refer to service centre
Symptoms :
XXX XXX message was displayed in the Player System.
Tests :
Question: Is there any error message displayed?
Response: No
Question: Which type of xxxx is being used for xxxx?
Response: xxxx
……
Response: Unit may be defective please refer to service centre

FREE TEXT
“UKBUCHANLS 1/13/2001 6:35:41 PM c/cld needing help with his connections to his video.”,
```

Figure 5.1: Sample KB and free text

The Call Centre dataset studied in this chapter is the *FWP* dataset which comprises of records pertaining to a ‘Hi-Fi’ system. In total there were 1117 records out of which 234 of them had the ‘KB-text’ format whilst 883 of them had the ‘free-text’.

5.2.4 Document Representation

In order for the textual records to be used for classification, they need to be converted to a numeric format and represented in matrix form. This conversion process is known as document representation. It has been shown in many studies the type of document representation (Leopold and Kindermann, 2002) can affect classification accuracy. There are many different document representations available. However, we shall focus our attention on two such representations only, so as to keep the number of experiments within practical limits. The next chapter will examine other representations. The two representations that were studied are as follows:

- Binary representation
- Term frequency inverse document frequency length normalised (tfidf-ln) representation

Let a$_{ij}$ be the weight of term *i* in document *j*, f_{ij} be the frequency of term *i* in document *j*, df_i be the number of documents in which term *i* occurs, N be the number of documents and M be the number of distinct terms in the document collection.

The binary representation is one of the simplest but effective forms of representation. A word present in the document would be given a feature value of 1 whilst a zero is used if the word is absent. The formula below shows the weight for each word under the binary representation scheme.

$$a_{ik} = \begin{cases} 1 \ \ if \quad f_{ik} > 0 \\ 0 \ \ otherwise \end{cases} \tag{5.1}$$

The ‘tfidf-ln’ is widely used as a standard of comparison in the literature (Salton and Buckley, 1988). The ‘tfidf-ln’ is normalized with the length of the document. Such normalization will allow a better comparison between the ‘free’ and ‘KB texts’ which have different lengths. The formula for the ‘tfidf-ln’ representation is given below.

$$a_{ik} = \frac{f_{ik} * \log\left(\frac{N}{df_i}\right)}{\sqrt{\sum_{j=1}^{M}\left[f_{jk} * \log\left(\frac{N}{df_j}\right)\right]^2}} \tag{5.2}$$

5.2.5 Type of Algorithm

Studies (Yang, 1999 and Yang and Liu, 1999) comparing different algorithms reveal that there is no ‘best’ algorithm that has been found, that can work well in all situations. In this thesis, we study three algorithms plus a variant as given below:

- Support Vector Machines (SVM)
- Naïve-Bayes Classifier (NB)
- Naïve-Bayes Classifier with Kernel density estimation (NB-with K)
- C4.5 Decision trees

The ‘Naïve Bayes’ Classifiers do not require any parameter settings unlike the other two algorithms. As mentioned earlier, LIBSVM (Chang and Lin, 2003) was used to implement ‘SVM’ while WEKA (WEKA, 2003) was used to implement the other three algorithms. The ‘NB’ method assumes a normal distribution for continuous input variables while the ‘NB (With K)’ method (John and Langley, 1995) determines the input distribution through kernel density estimation. Given the enormous number of experimentation required, the settings for the algorithmic parameters were fixed based on preliminary trial experiments. An exhaustive tuning of the parameters is carried out later in the chapter once some settings of the other factors have been decided upon, thus making the number of experimental runs feasible. As of now, for ‘SVMs’, the parameters for cost and shape of the gaussian kernel were set at 500 and 0.1 respectively for the Escalation information field and at 500 and 0.03 for the other two. As for ‘C4.5’ (Ross, 1993), the pruning confidence parameter, -c was set at 0.1 whilst the minimum number of instances per branch was set to 2. For each combination of factors, the testing and training data was divided into a 30/70 split. To obtain a measure of variability, five stratified samples were used. Therefore, the total number of runs made were 5*4*4*3*3*2 = 1440.

5.2.6 Designable and Non-Designable Factors

Designable factors are those whose settings the designer is aware of and has full control over. Non-designable factors are those whose settings cannot be determined prior to its use since the designer is not informed about the levels of these factors. In this study, four of the factors - *Process*, *Method*, *Document Representation* and *Data_format* were considered to be designable since the designer is able to decide on and has control over the settings for these factors. The *Field* factor is considered as non-designable. The different informative fields that the *Field* factor represents can be viewed as representing different datasets. Hence, the type of dataset used on the system is not under the control of the designer. As such, it becomes a non-designable factor and the designer of the classification system is interested in choosing the settings of the other factors that provide the best performance when averaged over different datasets.

5.3 Results and Discussion

A small excerpt of the results obtained is shown in Table 5.3.

Table 5.3: Classification accuracies for a specific factor setting

Trial	Tfidf-ln	Binary
1	76.488	80.952
2	76.191	77.381
3	76.191	81.548
4	78.274	83.631
5	76.787	78.571
Mean	76.786	80.417
Std. Deviation	0.868	2.476

Here, the *Field*, *Method* and *Process* factors have been set to 'Escalate', 'SVM', and 'Full' respectively. Both the mean and the variance of the 5 stratified trials undertaken were computed for each of the different cases. For initial analysis purposes, box plots of the mean classification accuracies were made as shown in Figure 5.2.

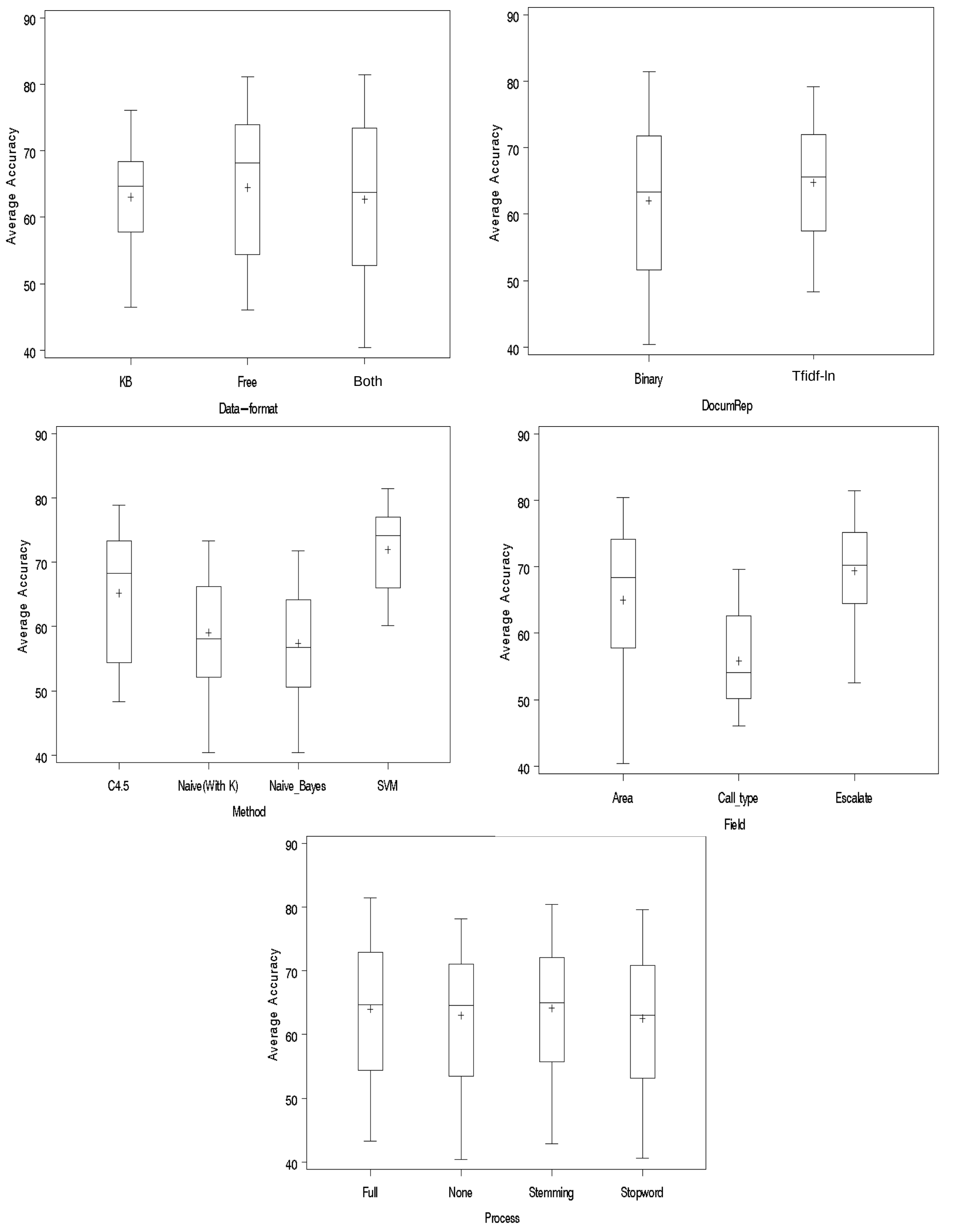


Figure 5.2: Box plots of the average accuracy of different factors with the '+' indicating the mean value

5.3.1 Box Plots

The box-and-whisker plots in the figure display data as follows: the horizontal line inside the box represents the median; the top and bottom of the box represent the 3rd quartile (75th percentile) and the 1st quartile (25th percentile), respectively. The distance between these two is the inter-quartile range (IQR). Whiskers are drawn from the upper edge of the box to the maximum value as well as from the lower edge of the box to the minimum value. The plus sign within the box represents the mean accuracy value.

Inspections of the plots reveal clearly observable differences in the accuracy values obtained for the two factors - *Field* and *Method*. For the *Field* factor, the accuracies obtained for the 'Escalate' and the 'Area' field is higher than the 'Call-type' field. For the type of method used, 'SVM' performs better than the 'C4.5' algorithm, which surpasses the 'Naïve-Bayes' methods. Part of the reason could be due to the fact that algorithms like 'Naïve-Bayes' methods are affected quite a lot by redundant features (Witten and Frank, 2000).

As for the *Process* factor, it is observed that this factor does not have much influence on the accuracies that were obtained. Part of the reason for this observation could be the fact that the texts within the database are short and domain specific. As such the words used as well as their morphological variations are not as extensive as one might find in news feeds or stories where these processing have been found to be more useful.

Although no significant difference in classification accuracies is observable for the data-format fields, the amount of variability exhibited by each of the different levels investigated for this factor seems to be very different. For example, the IQR for KB text is much smaller than that for the Free and Full texts. A possible explanation could be due to the fact that the KB text is a lot more structured due to the use of the knowledge base. As a result, there is lesser ambiguity in the manual class labelling as well as lesser variability in the contents of the KB text which could have resulted in the smaller variability observed for such text.

Further, comparison of box plots for the *Method* factor shows that the variability for SVM is much smaller than the other three methods. It seems to suggest that SVM is more robust to the various factors. This could possibly be due to the instability (Breiman, 1996) of the other methods.

In order to better visualize the differences within the *Method* and *Field* factors, Figure 5.3 is presented. This plot shows these results better, by charting the differences between the levels of these factors. Though the box-whisker plots were able to provide some insight into the data, more objective analysis was carried out using the Analysis of Variance approach.

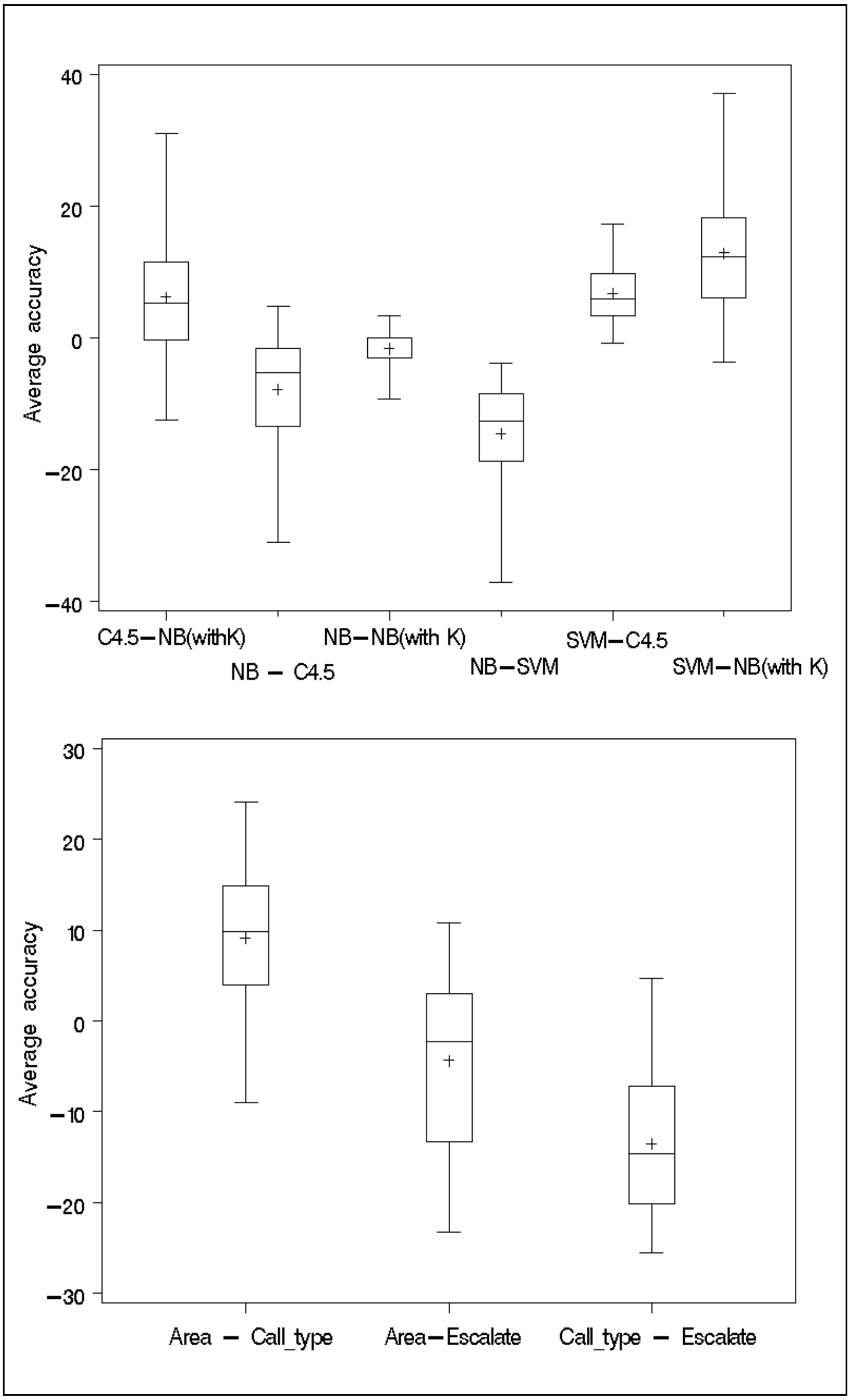


Figure 5.3: Difference in accuracies for different levels of method and field factor with a '+' indicating the mean value

5.3.2 Analysis of Variance (ANOVA)

To determine the significance of the various factors studied, an ANOVA was conducted using the SAS software, using the average accuracy of the 5 trials as the response. The independent effects together with all possible two-way interactions were studied. Third and higher order interactions have been assumed to be zero. After pooling the sum of squares of the insignificant factors at the 95% confidence interval with the error estimates, another ANOVA was carried out on the reduced number of factors. The results are shown in Table 5.4.

This reduced model explains about 92% of the variability observed in the data. The sum of squares for a particular effect measures the amount of variation in the response due to that effect. The last column displays the percentage contribution of each factor (Ross, 1996).

Table 5.4: Analysis of Variance of reduced model

Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr> F**	**% Contribution**
Data_format	2	169.728	84.864	9.24	0.0001	0.522
DocumRep	1	552.769	552.769	60.15	<.0001	1.906
Field	2	9165.415	4582.707	498.71	<.0001	31.599
Process	3	134.559	44.853	4.88	0.0026	0.464
Method	3	9411.634	3137.211	341.4	<.0001	32.448
DocumRep* Data_format	2	103.731	51.866	5.64	0.004	0.358
Field*Data_format	4	3175.147	793.787	86.38	<.0001	10.947
Method*Data_format	6	662.804	110.467	12.02	<.0001	2.285
Method*DocumRep	3	1517.886	505.962	55.06	<.0001	5.233
Method*Field	6	1768.300	294.717	32.07	<.0001	6.096
Error	255	2343.239	9.189			

It can be seen that both *Data_format* and *Method* factors interact with three other factors while the *Process* factor does not interact with any of the other factors at all. Examining the last column of the table it can be seen that the two most important factors that affect the classification accuracies are the *Field* and *Method* factors. Each factor accounts for more than 30% of the variability observed in the data. This corresponds with the observations made earlier using the box-whiskers plots.

It is also interesting to observe that the *Data_format* factor considered on its own explains less than 1% of the variability observed in the data while through its interaction with the other factors, especially the *Field* factor, it accounts for about 14% of the variability in the data. The analysis also suggests that the type of document processing carried out does not affect the classification accuracy much. This implies that for the given data set, any form of the document processing (including no processing) could have been carried out without compromising on the accuracies obtained.

Although two factors, *Method* and *Field*, affect the accuracy of the classification significantly, the designer of a classification system only has control over the *Method* factor. As mentioned, since the information to be extracted from a particular dataset would depend very much on the end user, the flexibility of controlling the variability due to the information field is not available to the designer of the classification system. As such, further investigations would be carried out on the designable factor, *Method*.

5.3.3 Method Factor

In order to better appreciate the influence of the different methods on classification accuracies, a separate analysis of variance was carried out for three methods. The 'Naïve Bayes' was not considered since it is quite similar to the 'Naïve Bayes' with density estimation. For the three methods; 'Naives Bayes (with K)', 'SVM' and 'C4.5', R^2-values of .95, .98 and .98 was obtained, respectively, from the analysis of variance. Table 5.5 shows the percentage contribution of each factor for the three methods.

The zero entries in the table denote that the factors are insignificant at the 95% confidence level and therefore their sum of squares contribution has been pooled to that of the error sum of squares (Ross, 1996). It is interesting to note that, the *Field* factor contributes to a large amount of variability present in the data. In fact for all three methods, at least 60% of this variation observed is explained by this factor or its interaction.

Table 5.5: Percentage contributions of different factors for various methods

Factors	SVM	NB(with K)	C4.5
Data_format	5.415	3.549	3.266
DocumRep	2.236	24.650	0.636
Field	71.410	31.102	81.070
Process	1.797	0.000	1.431
DocumRep* Data_format	0.000	2.778	0.000
Field*Data_format	14.720	29.334	10.297
Process*Data_format	0.000	0.000	0.000
DocumRep*Field	0.431	0.620	0.000
DocumRep*Process	0.000	0.000	0.000
Process*Field	0.899	0.000	0.000

This means that having chosen a particular method, the range of classification accuracies obtained depend to a very large extent on the information to be extracted from it. For 'SVM' and 'C4.5,' other factors like document representation, database format and type of processing does not contribute as significantly to this variation. However, the interaction between *Data_format* and *Field* factors is found to be significant. This implies that the accuracy obtained for a given information field is affected by the format in which the data is available. Figure 5.4 shows this for the 'Naïve-Bayes (with K)' method. The plot reveals that the 'Call_type' field and the 'Area' field can be better distinguished using the information within the 'KB-text'. This observation is interesting since it also provides insight into the data. For example, for the 'Area' field, the knowledge base system would have a series of related questions that has a strong association with the problem area under concern. This provides greater evidence/information for classification purposes as opposed to when a similar problem appears in the free text field.

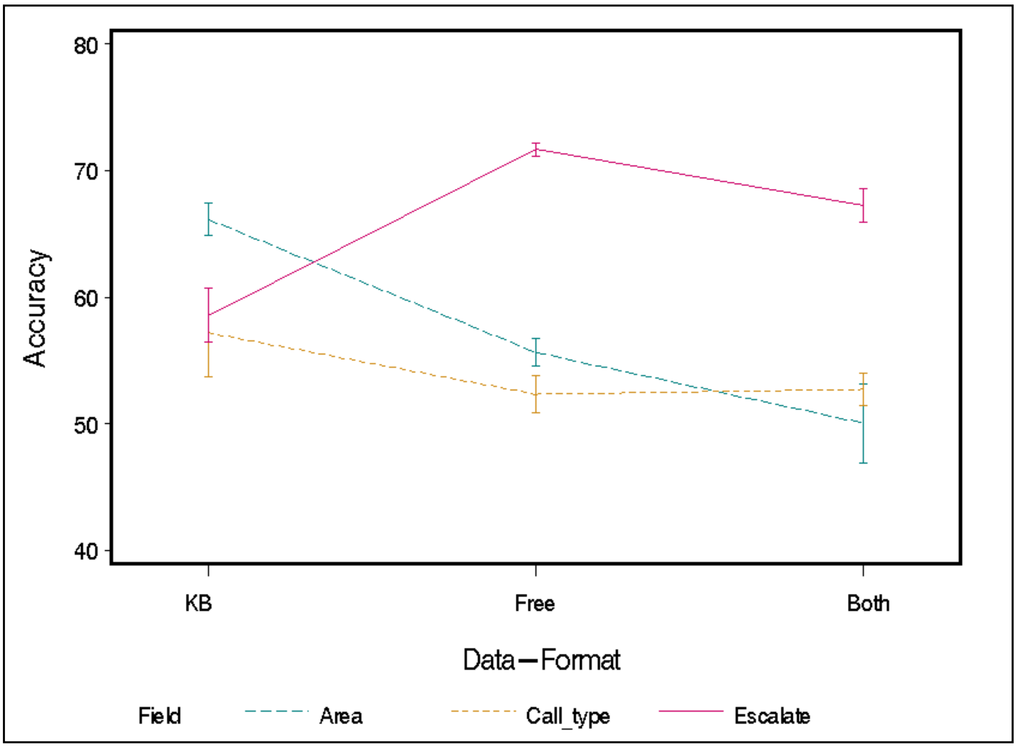


Figure 5.4: Interaction plot between Format and Field factors for 'Naïve-Bayes (with K)'

Another interesting point to observe is that the breakdown of the contribution of each factor is similar for ‘SVM’ and ‘C4.5’ but different for ‘Naïve Bayes (with K)’. For this method, the document representation was found to be a significant contributor to the variation. This is understandable since for ‘Naïve Bayes (with K)’, the type of document representation would affect the probabilities that are computed for each of the input variables and hence consequently affect the classification accuracy.

5.4 Mean and Interaction Plots

It is important to determine the levels of setting for each of the factors so as to maximize classification accuracy, in an ‘averaged’ sense. Since non-designable factor settings are not controllable, the selection of optimal settings was done with respect to the four designable factors, namely – *Method*, *Data_format*, *DocumRep* and *Process*. Therefore, the classification accuracy for a given combination of these four factors is an averaged value over the possible combinations of the non-designable factor. Figure 5.5 shows the mean plots for the four designable factors whilst Figure 5.6 shows their interactions. Generally, it can be observed that the ‘C4.5’ and the ‘SVM’ methods perform better than the ‘Naïve Bayes’ algorithm. Further, it is interesting to note that the ‘C4.5’ and the SVM methods vary in a similar fashion with respect to the other factors. When averaged over the *Field* Factor, interactions are found between the *Method* and *DocumRep* factors as well as the *Method* and *Data_format* factor.

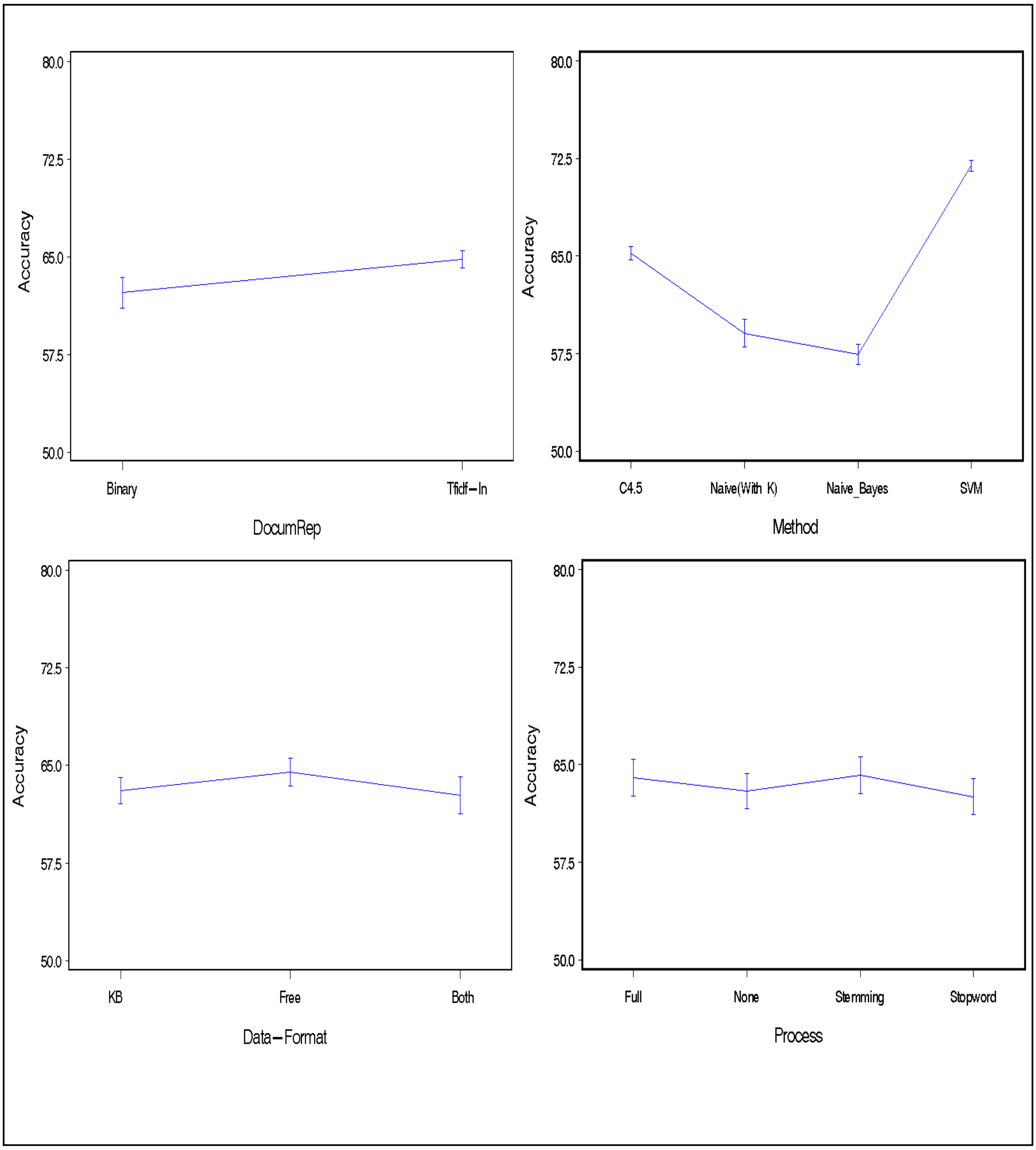


Figure 5.5: Mean plots of designable factors

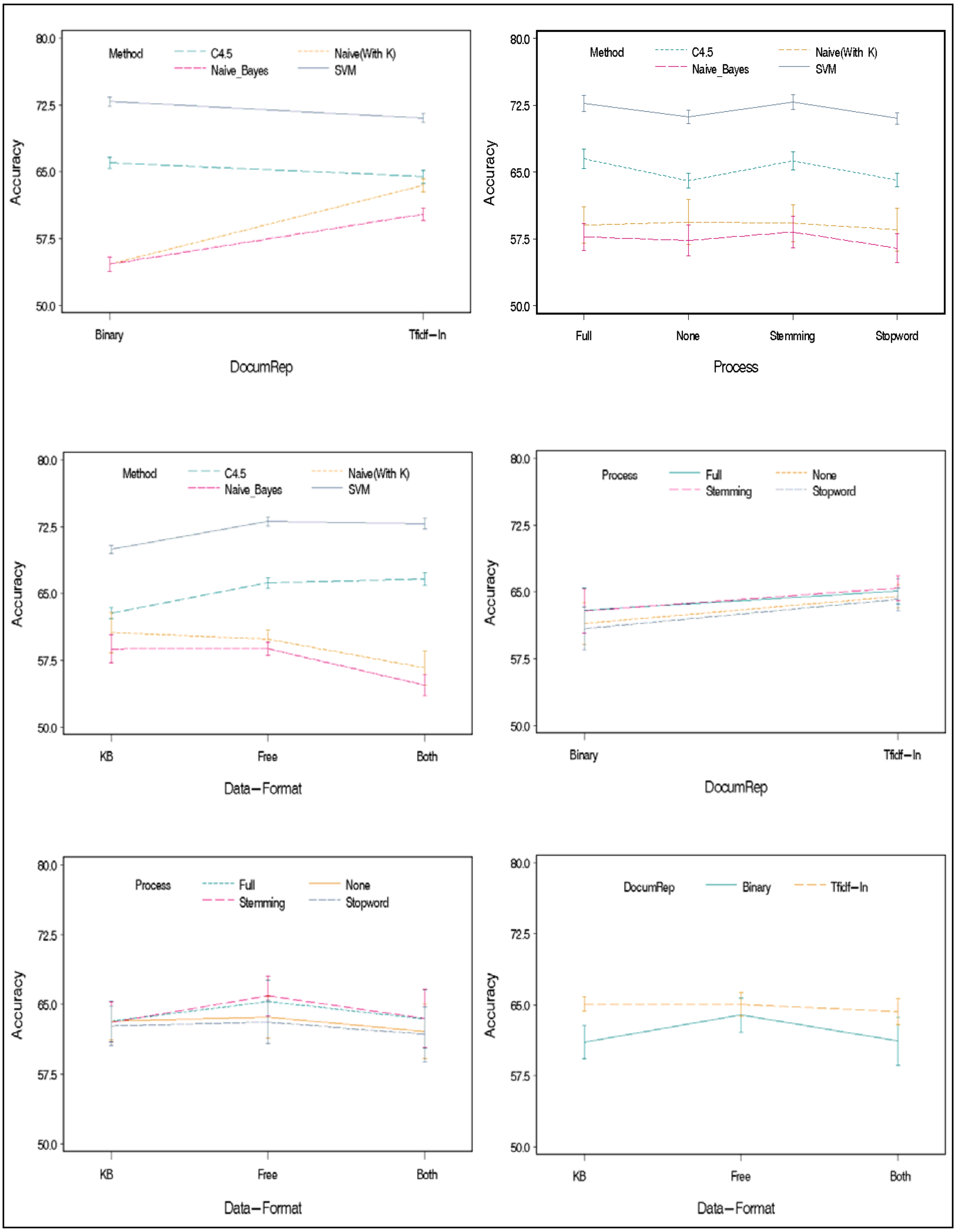


Figure 5.6: Interaction plots of designable factors

5.5 Sensitivity of Results

In the above study, fixed tuning parameters were used for the ‘C4.5’ and ‘SVM’ methods. It is known that the parameters c (pruning confidence) and m (number of examples in leaf node) affects the performance of the ‘C4.5’ whilst the tuning parameters c (cost of misclassifying instances) and sigma (shape of gaussian kernel) affects the performance of ‘SVMs’. Hence, to extend the generality of our study, further experiments were conducted to determine the influence of the algorithmic parameters on the results obtained. For the ‘C4.5’ the c values were varied from 0.05 to 0.3 in steps of 0.05 whilst the values of 2, 6 and 10 were attempted for the m parameter. For SVM, the parameters were varied in steps of 2^k; for the c parameter, k goes from 1 to 11 whilst for sigma, k goes from –11 to 1. For both schemes a 3-fold cross validation was carried out to determine the training set accuracy at the various parameter settings. The parameter that gave the best training accuracies was then used to determine the accuracy values for the test set.

The knowledge from the results of the previous experiments has been used to cut down the number of experiments required which could otherwise become overwhelming. From Figure 5.6, it can be observed that both the ‘C4.5’ and ‘SVM’ algorithms, exhibit almost identical variation with the type of processing and the format of the data. Hence, instead of varying all levels for these factors, only the ‘stemming’ and ‘full-processing’ levels were considered for the *Process* factor whilst the format of the data was fixed at ‘Both’. As for the *DocumRep* and *Field* factors all the previously considered levels were investigated.

5.6 Optimal Settings

The optimal settings can be decided from Figure 5.7, which shows the mean and interaction plots for the designable factors after the ‘C4.5’ and the ‘SVM’ methods were tuned.

Examining the plots, the optimal settings for the different factors could be decided. Table 5.6 shows these settings. As can be seen from the table, the optimal settings may not be distinct. As such, other considerations can be taken into account to break the ties. For example, for the *Process* factor, either ‘stemming’ or ‘full-processing’ is an appropriate choice. However, ‘full-processing’ was chosen since it results in a lesser number of features due to the removal of stopwords. Hence following this approach, the optimal factor settings were determined.

Table 5.6: Optimal settings of designable factors

Factors	**Settings**
Process	Stemming or Full
Method	SVM
Data_format	Both
DocumRep	Binary

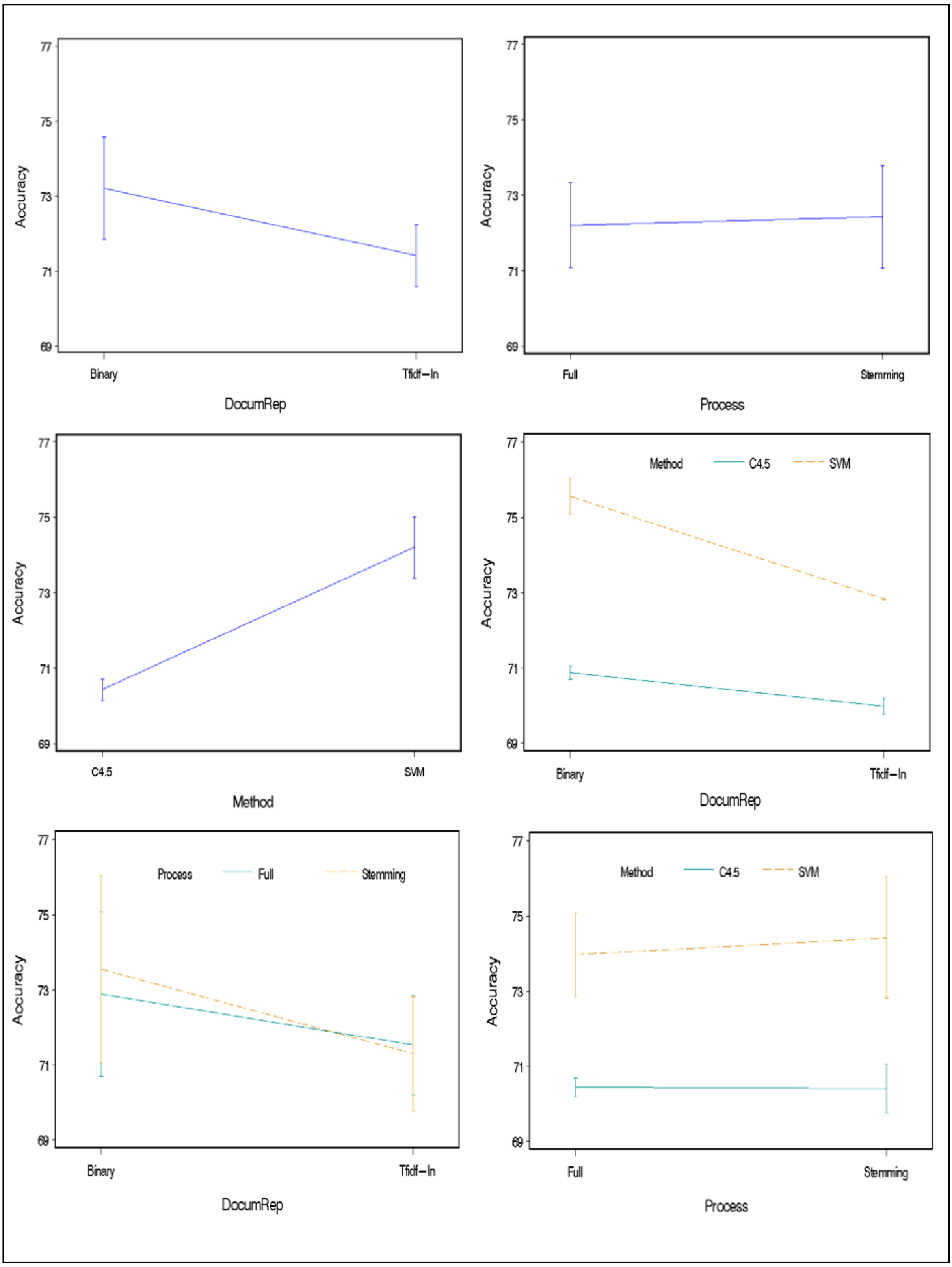


Figure 5.7: Mean and interaction plots of designable factors with parameter tuning for C4.5 and SVM

5.7 Summary

In an attempt to optimise the classification performance of the system, a detailed analysis of variance on the *FWP* Call Centre dataset was carried out. Some of the important findings of the study, pertaining to the investigated data set, would include the following:

- The type of algorithm and the information field in the dataset were found to be important factors affecting classification accuracy. They affect the accuracy to a similar extent.
- The type of preprocessing was not particularly important.
- Interaction between information field and database format cannot be ignored.
- The 'Naïve Bayes(with K)' algorithm is very sensitive to how the document is represented.
- Different algorithms favour different representation schemes; 'Naive Bayes' in general gave better averaged results with a 'tfidf-ln' representation whilst 'SVM' and 'C4.5' gave better results with a binary representation.
- The optimal design settings for the *Process*, *Method*, *Documrep* and *Data_format* factors were 'Full', 'SVM', 'binary' and 'Both' respectively.

In general, a structured approach to determine the optimal settings of a textual mining system was presented. Such an approach is generic and could be applied by designers of similar textual mining systems, with slight modifications to the type and levels of factors considered. The notion of non-designable factors was presented and it was shown how to handle them. In addition, it must be stressed that the optimal design settings obtained are specific to the investigated data sets. It could differ for other data sets. Further, obtaining the optimal design settings is not a final step but rather a key step to better understand the database being studied. With knowledge from such an analysis, more focused efforts can then be undertaken to improve classification accuracies further.

CHAPTER 6

TERM WEIGHTING SCHEMES

In the previous chapter, two document representation schemes were investigated. However, given the many other schemes of representation available and the comparative importance of an appropriate document representation (Leopold and Kindermann, 2002), further investigations are carried out. In total, 6 representations are studied, inclusive of the two in the previous chapter. One of the schemes has not been used in textual applications before. It combines the popular tfidf scheme with a 'softmax normalisation' in an attempt to better handle documents with exceptionally high/low frequency words. The *FWP* dataset as well as two other datasets are investigated. Both stopword removal and stemming was carried out on all the datasets. The SVM was used as the classifier.

6.1 Term Weighting Schemes

In the previous chapter, two weighting schemes have been investigated; namely binary and the **tfidf-ln** (**term frequency inverse document frequency - length normalized**). The other weighting schemes (Salton and Buckley, 1988, Leopold and Kindermann, 2002, and Dumais, 1992) studied include variants of the tfidf, a normalized term frequency scheme as well as a complicated entropy weighting scheme. Each of these schemes captures different features of a document; some more, some less. More would be mentioned in the subsections below. For completeness sake and to appreciate the role of each scheme in perspective, the two schemes presented previously are repeated here.

There are several ways of determining the weight a_{ik} of the word *i* in document *k*, but most of the approaches are based on two empirical observations regarding the text:

- The more times a word occurs in a document, the more relevant it is to the topic of document.
- The more times the word occurs throughout all documents in the collection, the more poorly it discriminates between documents.

Let

Term frequency, f_{ij} be the frequency of term *i* in document *j*

Document frequency, df_i the number of documents in which term *i* occurs

Global frequency, gf_i the total number of times term *i* occurs in the whole collection

N – number of documents and

M – number of terms

In a general case, the value of a term *i* in document *j* is $L(i,j)*G(i)$ where $L(i,j)$ is a local weighting for term *i* in document *j* and $G(i)$ is the term's global weighting.

6.1.1 Binary Weighting

This is the simplest approach in which the weight is 1 if the word occurs in the document and 0 otherwise.

$$a_{ik} = \begin{cases} 1 \ \ if \quad f_{ik} > 0 \\ 0 \ otherwise \end{cases} \tag{6.1}$$

6.1.2 Tf-n Weighting

This approach uses the frequency of the word in the document (Salton and Buckley, 1988). It normalizes the length of each term to 1. This has the effect of giving high weight to infrequent terms. Such a weighting only depends on the sum of square frequencies and not the distribution of those frequencies.

$$a_{ik} = \frac{f_{ik}}{\sqrt{\sum_{j=1}^{N} \left[f_{ij} \right]^2}} \tag{6.2}$$

6.1.3 Tfidf Weighting

The previous schemes do not take into account the frequency of occurrence of the word throughout all documents in the collection (Salton and Buckley, 1988). A well-known approach for computing word weights is the tfidf weighting, which assigns the weight to word *i*, in proportion to the number of occurrences of the word in the document, and in inverse proportion to the number of documents in the collection for which the word occurs at least once. This weighting scheme does not depend on the distribution of terms in documents but only on the number of different documents in which a term occurs.

$$a_{ik} = f_{ik} * \log\left(\frac{N}{df_i} \right) \tag{6.3}$$

6.1.4 Tfidf-ln Weighting

Same as the previous weighting except that the document feature vector is normalized to unit length. This takes into account the fact that documents might be of different lengths.

$$a_{ik} = \frac{f_{ik} * \log\left(\frac{N}{df_i}\right)}{\sqrt{\sum_{j=1}^{M}\left[f_{jk} * \log\left(\frac{N}{df_j}\right)\right]^2}} \tag{6.4}$$

6.1.5 Tfidf-ls Weighting

This weighting scheme scales the tfidf weight using a logistic function (Pyle, 1999). This type of scaling has two ranges, a linear range and a non-linear range towards either end of the linear range. One could think of such a scaling as the mapping from the original weight vector space into a [0,1] space with major portion of the values being linearly transformed while a few ‘outlying’ values being squashed to be between [0,1]. The following transformations is applied to each row vector c_i

$$b_{ij} = (c_{ij} - \bar{c}_{i.})/[\lambda(\sigma(c_{i.})/2\pi)] \tag{6.5}$$

where c_{ij} is the tfidf value for word *i* in document *j*, $\bar{c}_{i.}$ is the average value in row *i*, $\sigma(c_{i.})$ is the standard deviation value for row i, λ is the response indicator in standard deviations where a value of 3 indicates that the linear portion of the transformation covers about 99.7% of the range of data given a normal distribution.

The final normalized value is

$$a_{ik} = \frac{1}{1+e^{-b_{ik}}} \tag{6.6}$$

6.1.6 Entropy Weighting

Entropy weighting is based on information theoretic ideas. In the paper by Dumais (1992), it turned out to be the most effective scheme in comparison to 6 others. Averaged over five test collections, it was for instance 40% more effective, for the task of information retrieval, than word frequency weighting. The formulae for entropy weighting is given as:

$$a_{ik} = \log(f_{ik} + 1.0) * \left(1 + \frac{1}{\log(N)} \sum_{j=1}^{N} \left[\frac{f_{ij}}{gf_i} \log \frac{f_{ij}}{gf_i} \right] \right) \tag{6.7}$$

where

$$\frac{1}{\log(N)} \sum_{j=1}^{N} \left[\frac{f_{ij}}{gf_i} \right] \log \frac{f_{ij}}{gf_i} \tag{6.8}$$

is the average uncertainty or entropy of word *i*, which is –1 if the word is equally distributed over all documents and 0 if the word occurs in one document only. Thus entropy takes into account the distribution of the terms over the documents. This differs from the use of entropy in Section 4.3.2, which accounts for the distribution of the class information over the documents.

The schemes discussed in subsections 6.1.1-6.1.4 and 6.1.6 are generally used in the information retrieval and text processing disciplines. The scheme in subsection 6.1.5 is a ‘softmax normalisation’ generally used in the machine learning literature and has been attempted here.

6.2 Datasets Studied

In this chapter, inclusive of the *FWP* dataset, two other datasets were studied, namely; *CDP* and *Solid*. The *CDP* dataset had 1073 records with 5 classes whilst the *Solid* dataset had 415 examples with 7 classes. Appendix C shows the distribution of the classes within the two datasets. For the *FWP* dataset which was considered in the previous chapter, for easy reference, each of the information fields (response), *Call-Type*, *Escalate* and *Area* would be referred to as 3 different datasets from now onwards. For these 3 datasets, three data formats (Section 5.2.3) were studied. The short form of *Esc* would be used for the *Escalate* dataset from now onwards.

6.3 Experimental Study on Term Weighting Schemes

SVM was used as the classifier. Each data set was split into 70% and 30% for training and testing respectively. This was done 5 times using a stratified sampling scheme to obtain 5 different trials, for each dataset. The parameters of the SVM were tuned in steps of 2^k. For the c parameter, k goes from 1 to 11 whilst for sigma, k goes from –11 to 1. A 3-fold cross validation was carried out to determine the training set accuracy at the various parameter settings. The parameter that gave the best training accuracies was then used to determine the accuracy values for each of the corresponding test sets.

The Duncan’s Multiple Range Test was used to determine if the mean accuracies obtained for the different weighting schemes on the different data sets were different from one another. A p-value of 0.05 was used. This test is used to test for all possible means. The null hypothesis would be $H_0 : \mu_i = \mu_j, \text{for all} \, i \neq j$ where μ_i is the mean accuracy due to the *i-th* weighting scheme. If we test all possible means using t-tests, the probability of type I error for the entire set of comparisons can be greatly increased. To help avoid this problem the Duncan's Multiple Range test was used (Montgomery and Runger, 1994).

Table 6.1: Duncan's Groupings for the 'KB' format

Dataset	A	B	C
Area	Tfidf-ln Binary	Binary Entropy Tfidf-ls Tfidf	Tf-n
Call-Type	Tfidf-ln Tfidf-ls Binary Entropy	Binary Entropy Tfidf	Tfidf Tf-n
Esc	Binary Tfidf-ls Tfidf-ln Tfidf Entropy Tf-n	-	-

Table 6.2: Duncan's Groupings for the 'Free' format

Dataset	A	B	C
Area	Binary Tfidf-ls Tfidf-ln Entropy Tfidf	Tf-n	-
Call-Type	Tfidf-ls Binary Tfidf-ln Entropy Tfidf	Tf-n	-
Esc	Tfidf-ls Binary Tfidf-ln Tfidf	Tfidf-ln Tfidf Entropy	Tf-n

Table 6.3: Duncan's Groupings for the 'Both' format

Dataset	A
Area	Binary Entropy Tf-n Tfidf-ls Tfidf-ln Tfidf
Call-Type	Tfidf-ls Tf-n Binary Tfidf-ln Entropy Tfidf
Esc	Tfidf-ls Binary Tf-n Tfidf Entropy Tfidf-ln

Tables 6.1 to 6.3 show the results for the *Area*, *Call-Type* and *Esc* datasets for various data formats and weighting schemes. The weighting schemes found within each cell are not statistically significantly different from one another at a significance level of 0.05. The mean accuracies decrease as we traverse the tables from left to right. In some instances, a particular weighting scheme is found only within a single cell while in other instances it is found in two. In the case of the later, such an instance would imply that the weighting scheme under consideration is not statistically different from the other weighting schemes found within the two cells.

Investigation of the results reveal that the performance of the different weighting scheme is dependent on the dataset and as well as the data format. In general, for the 'KB' and 'free' format (Table 6.1 and Table 6.2), the tfidf-ls, tfidf-ln and the binary scheme outperform the rest of the schemes. In fact, the tfidf-ls scheme was found in the top two spots of the A-grouping 7 out of 9 times, as seen from Tables 6.1-6.3.

In the cases investigated, complicated weighting schemes such as entropy weighting did not perform as well as a simple weighting scheme such as a binary representation. A possible explanation for this could be the fact that for most of the cases, a particular record is usually classified into a category based on the existence of a word or a group of words. This is especially true, in the case of free format text, which has an average word length of about 12.5 words. In such instances, it is common for keywords not to be repeated in a given record and the existence of a particular word/or group of words is good enough to determine class information. Hence in the free format and the Both format binary representation scheme does especially well.

From Tables 6.1–6.3, it is interesting to note that for the *Area* dataset, the binary representation scheme was the best for the ‘Both’ and ‘Free’ formats but lost to the tfidf–ln scheme for the ‘KB’ format. In fact, there was about 4% difference between the binary and the tfidf-ln scheme for the ‘KB’ format. This difference could be attributed to the characteristics of the KB format. When a knowledge base is used to assist a customer, a number of questions are being asked. During this questioning, it is highly likely that some of the keywords associated with some of the classes within the problem area arise. The existence of a particular word or group of words within the questioning procedure might falsely indicate a particular class. Hence that may give rise to the poorer performance observed in the binary weighting scheme, in comparison with the tfidf-ln scheme, for the KB format (Table 6.1). This observation is not present for the *Call-Type* and the *Esc* datasets since the ‘keywords’ for these two datasets are generally not prevalent in the questions being asked during use of the knowledge base system.

Table 6.4: Duncan's Groupings for two other datasets

Datasets	A	B
CDP	Tfidf-ln Binary Entropy Tf-n Tfidf-ls	Entropy Tf-n Tfidf-ls Tfidf
Solid	Tfidf-ls Tf-n Binary	Entropy Tfidf-ln Tfidf

Similar analysis of means was carried out on the *CDP* and *Solid* datasets as shown in Table 6.4. For these two databases only the 'Free' text format was available and the response variable was similar to the *Area* dataset. From Table 6.4, it can be seen that the tfidf-ls, tf-n and the binary scheme are present in A-grouping for both datasets.

As before, it is interesting to note that the tfidf representation scheme in its original form produce very poor results. However, with some modification like length – normalization or logistics scaling, classification accuracies see a major improvement. A sample of the results of the analysis is provided in Appendix C.

6.4 Summary

In this chapter, the influence of different term weighting schemes was studied. From the datasets investigated, it was found that the type of representation can affect the results obtained. The tfidf-ls scheme and the binary representations exhibited good performance. The promising performance of the tfidf-ls scheme provides motivation for its use in text processing applications. Even though a very simple scheme, the good performance of the binary representation could be attributed to the fact that the records comprise of short texts, where the words generally do not repeat too much and the very existence of a word in itself is good enough to classify it. Given their better performance, the binary and tfidf-ls schemes will be used for the experiments with singular value decomposition (SVD) presented in the next Chapter. Although the entropy weighting scheme did not perform as well, it will be used in the following study with SVD, given its much better performance reported in the literature (Dumais, 1992). In studies described in subsequent chapters, only the ‘Full’ format of the *Area, Call-Type* and *Esc* datasets would be studied since this provides for a larger number of instances.

CHAPTER 7

LATENT SEMANTIC ANALYSIS

As seen from the previous chapter, the manner in which the document is represented (weighting scheme used) affects the classification accuracy that is finally obtained. Thus far, as explained in section 4.4.1.1, the bag-of-words approach has been used to capture the information within the document. However, sometimes a 'latent semantic' or 'conceptual' representation of the document might result in a better performance (Foltz, 1990 and Zelikovitz, 2001). A widely used technique that allows for such a representation is Singular Value Decomposition (SVD). SVD is basically a dimension reduction technique which shows the breakdown of the original relationships into linearly independent component or dimensions. In this chapter, the usefulness of this scheme when applied to the Call Centre datasets is evaluated by studying the classification accuracies as the number of dimensions is varied. Further, the existence of an optimal number of dimensions for the investigated datasets is studied empirically. Also the link between the accuracy values and the information loss, as the number of dimensions is reduced, is explored.

7.1 Introduction

One of the drawbacks of the bag-of-words representation of a document is that it totally neglects semantic similarities between two words. For example, it would be desirable if words that describe a similar concept are represented as a single unit. This is quite similar to the task of query expansion, which usually requires the use of prior knowledge, semantic networks, etc. Furthermore, semantic similarity between two terms can depend heavily on context.

A source of information about semantic similarity in the given context is co-occurrence analysis. If two terms co-occur in documents very frequently, in a given corpus, they can be considered as being semantically related. Incorporating co-occurrence information in a learning system for exploiting semantic similarity between words would seem to be a very expensive task. However, the SVD technique allows us to extract this information automatically. In the Information Retrieval literature, the semantic representation of a document is known as Latent Semantic Indexing.

7.1.1 Singular Value Decomposition

Latent Semantic Analysis (LSA) (Deerwester et al., 1990) refers to the incorporation of semantic information in the document representation. LSA is carried out using the Singular Value Decomposition (SVD) technique which projects all documents into a space where co-occurring terms are projected in similar directions, while non co-occurring ones are projected in very different directions. In such a space, closeness between documents is determined by the overall pattern of term usage. Therefore, documents can be considered to be similar to one another regardless of the precise words that are used to describe them and their description depends on a kind of consensus of their term meanings.

In SVD, a rectangular matrix is decomposed into a set of orthogonal factors from which the original matrix can be approximated by linear combination. More formally, any rectangular matrix, for example a t by d matrix, X, of terms and documents, can be decomposed into the product of three other matrices:

$$\underset{txd}{X} = \underset{txr}{T} \bullet \underset{rxr}{S} \bullet \underset{rxd}{D'} \tag{7.1}$$

such that T and D have orthonormal columns, S is diagonal (also known as matrix of singular values) and r is the rank of X, which is equal to the number of nonzero singular values in S. This is the so-called singular value decomposition of X. If only the k largest singular values of S are kept along with their corresponding columns in the T and D matrices, and the rest deleted (yielding matrices $\overline{T}$, $\overline{S}$ and $\overline{D}$) the resulting matrix, $\hat{X}$, is the unique matrix of rank k that is closest in the least squares sense to X:

$$\underset{txd}{X} = \underset{txd}{\hat{X}_k} = \underset{txk}{\overline{T}} \bullet \underset{kxk}{\overline{S}} \bullet \underset{kxd}{\overline{D}'} \tag{7.2}$$

The idea is that the $\hat{X}_k$ matrix, by containing only the first k independent linear components of X, captures the major associational structure in the matrix and throws out noise. It is this reduced model that is used to approximate the term-document association in the matrix X. Using the SVD to find the approximation $\hat{X}_k$ guarantees that the approximation is the best we can create for a given choice of k. The next subsection would present how the ‘error’ committed due to this approximation can be computed mathematically.

7.1.2 Relative Change Metric

As mentioned earlier, the rank r of the matrix X is equal to the number of nonzero singular values. It then follows directly from the orthogonal invariance of the Frobenius norm that $\|X\|_F$ is defined in terms of those values:

$$\|X\|_F = \|TSD'\|_F = \|SD'\|_F = \|S\|_F = \sqrt{\sum_{j=1}^{r} S_{jj}^2} \tag{7.3}$$

As outlined in Berry et al. (1995), it can be shown that the norm of the distance between *X* and its approximation, $\hat{X}_k$ are related. It reads as

$$\left\|X - \hat{X}_k\right\|_F = \sqrt{\sum_{j=k+1}^{r} S_{jj}^2} \tag{7.4}$$

Hence the relative change required to reduce from rank *r* to rank *k* is given as

$$\|X - X_k\|_F / \|X\|_F \tag{7.5}$$

A smaller number implies that there is relatively a smaller change that is needed to bring about the change in ranks. It also implies a closer approximation to the original matrix. Hence it can be seen that determining the relative change for the various approximations (k-values) is as good as determining the singular values of the *X* matrix.

7.1.3 SVD and SVM

The computation of the kernel (equation 4.15) within SVM involves the computation of the matrix

$$\begin{aligned} \underset{dxd}{A} &= \underset{dxt}{\hat{X}_k}{}' \bullet \underset{txd}{\hat{X}_k} \\ &= \underset{dxk}{\overline{D}} \bullet \underset{kxk}{\overline{S}}{}' \bullet \underset{kxt}{\overline{T}}{}' \bullet \underset{txk}{\overline{T}} \bullet \underset{kxk}{\overline{S}} \bullet \underset{kxd}{\overline{D}}{}' \\ &= \underset{dxk}{\overline{D}} \bullet \underset{kxk}{\overline{S}}{}' \bullet \underset{kxk}{\overline{S}} \bullet \underset{kxd}{\overline{D}} \\ &= \underset{dxk}{X_k}{}' \bullet \underset{kxd}{X_k} \end{aligned} \tag{7.6}$$

The original input matrix into the classifier was X' a *d* x *t* matrix. Using LSA the input is $X_k' = \overline{D} \cdot \overline{S}'$ (equation 7.6) which is a *d* x *k* matrix where k < t, hence resulting in a reduction in the input dimension.

7.1.4 Issues Studied

The effectiveness of the LSA approach for information retrieval depends amongst other things, on the datasets under study as well as the weighting schemes employed. It was found by Dumais (1992) that the weighting scheme could make as much as a 40% difference in the retrieval performance. A similar study investigating the effectiveness of the LSA approach, using various weighting schemes, but on the classification accuracy is undertaken here. Although there have been some recommendations on the number of dimensions (usually 100 to 300 (Letsche, 1997)), for homogenous and larger datasets, the choice of the number of dimensions that provides optimal performance of LSA for any given data set remains an open question and is normally decided via empirical testing (Berry, 1995). Further, to the best of the author's knowledge, the possible use of a relative change metric for suggesting optimal number of dimensions has not been explored before. If any distinct and suggestive relationship exists between the classification accuracy and the relative change metric, this could be used to guide the process of determining the optimal dimensions, which could save an enormous amount of time.

Given these considerations, the following issues are studied, for the Call Centre datasets;

1) What is the effect of various weighting schemes on the classification results?
2) Is there any link between the relative change metric and the classification accuracy results obtained?
3) How effective is latent semantic analysis for the datasets under investigation?
4) Is it possible to recommend an optimal number of dimensions k, for such datasets?

7.1.5 Related Work

Latent Semantic Analysis (LSA) was introduced in 1990 by Deerwester et al. (1990), as a scheme to enhance information retrieval. Given that it was used for the retrieval of documents it was popularly called Latent Semantic Indexing (LSI). Thereafter, various studies have been undertaken focusing on its use for information filtering and retrieval (Foltz, 1990 and Dumais, 1993, Schutze and Pederson, 1997).

Dumais (1992) also studied the effect on accuracy due to differential term weighting schemes applied with LSI. Tfidf, raw Tf, Tf-normalised and log(Tf)-Entropy were some of the schemes studied. In that study it was found that the entropy weighting scheme was very promising, improving performance (3-point averaged precision) by an average of 40%. The datasets that were investigated were ADI, MED, CISI, CRAN and TIME. Here again the emphasis was on information retrieval.

Recent applications of LSA have seen its use in text classification. Zelikovitz and Hirsh (2001) used LSA in conjunction with background knowledge. The SVD is performed on an expanded term by document matrix that includes the labeled training examples as well as the unlabelled examples. The data sets investigated included technical papers, web page titles from the NetVet Site, WebKB dataset and 20 newsgroups. The results show that such an approach can improve the performance of nearest-neighbor text classification.

Cristianini et al. (2001, 2000) demonstrated how LSA can be implemented implicitly to a kernel defined feature space and can be adapted for application to any kernel based learning algorithm and data. Experiments with the Reuters 21578 data set show that the technique can improve generalization performance (F1 measures) by focusing attention of a Support Vector Machine into informative subspaces of the feature space.

Most of the work described thus far address the algorithmic or theoretical basis of the LSI model. Few, if any, present implementation issues in practice. In their paper, Chen et al. (2001) and Bassu and Behrens (2003) address issues of scalability and production-level implementation of LSI, in view of information retrieval. They report that, using their implementation schemes, preliminary results showed substantial improvement in the query response time with minimal loss in relevance recall.

7.2 Experimental Study

From the previous chapter, it was seen that the binary and the tfidf-ls weighting schemes performed well. Hence, these two schemes are used here. Further, the entropy weighting scheme introduced previously was also studied as a third weighting scheme. From the results in the previous chapter, although this scheme did not perform as well, it has been included here since it was found to be very promising (Dumais, 1992).

7.2.1 Relative Change Metric Variation with Dimension Reduction

Figure 7.1 below shows how the relative change metric changes as the number of dimension is reduced. A hyperbolic profile is obtained as the metric varies with the number of dimensions. Depending on the dataset as well as the weighting scheme under study, the curvature of the profiles change slightly. For the widely recommended dimension settings of 100-300 dimensions (Letsche, 1997), the relative change metric varies between 20-60% for the *Area* and *CDP* datasets and between 10-40% for the

Solid datasets. Across the three weighting schemes, for a given dimension, the relative change metric was highest for the entropy scheme and lowest for the tfidf-ls scheme.

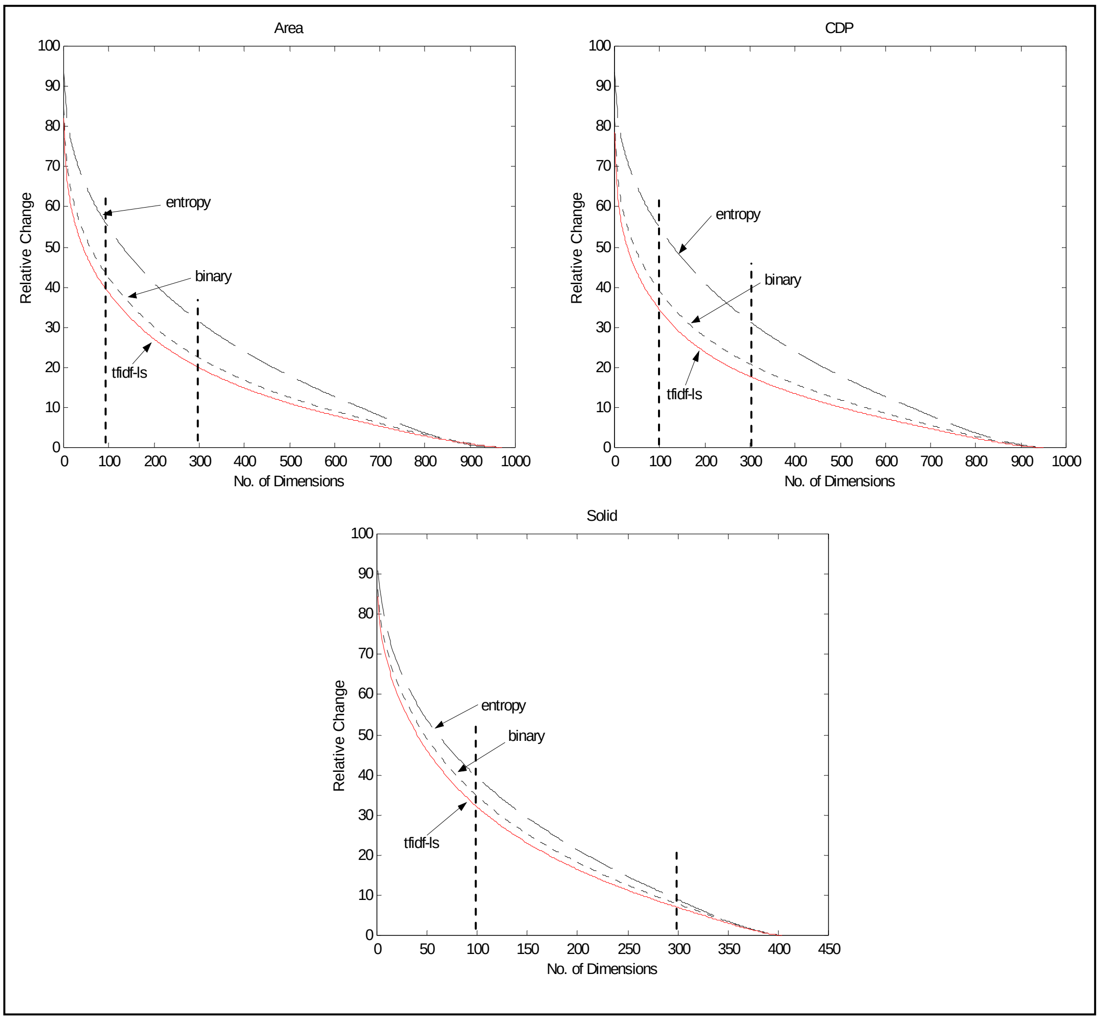


Figure 7.1: Variation of relative change metric with dimension reduction

7.2.2 Accuracy Variation with Dimension Reduction

In order to determine the effect of dimension reduction on classification accuracy, 15 different trials of randomly selected training and test samples were attempted for each dataset. For each trial, the dimensions were reduced from the original dimension value to a value of about 50. Parameter tuning was carried out for the c and sigma parameters just as in the previous chapter (Section 5.5). Table 7.1 shows an excerpt of the results for the *Area* dataset using the tfidf-ls weighting scheme.

Table 7.1: Classification accuracies of different trials for Area dataset using tfidf-ls weighting scheme

Dimension	Trial 2	Trial 4	…….	Trial 11	Trial 12	Trial 13	Trial 15
50	73.512	76.488		75.595	73.214	77.083	74.405
100	75.893	73.81		74.107	74.702	78.274	76.488
200	77.083	80.06		76.786	77.976	81.548	75.298
:							
600	75	83.036		77.083	80.655	79.762	76.488
800	76.488	83.036		78.571	79.167	80.06	76.786
968	76.786	82.738		79.464	78.869	80.06	76.786
1117	76.786	78.869		79.464	78.869	80.06	76.786

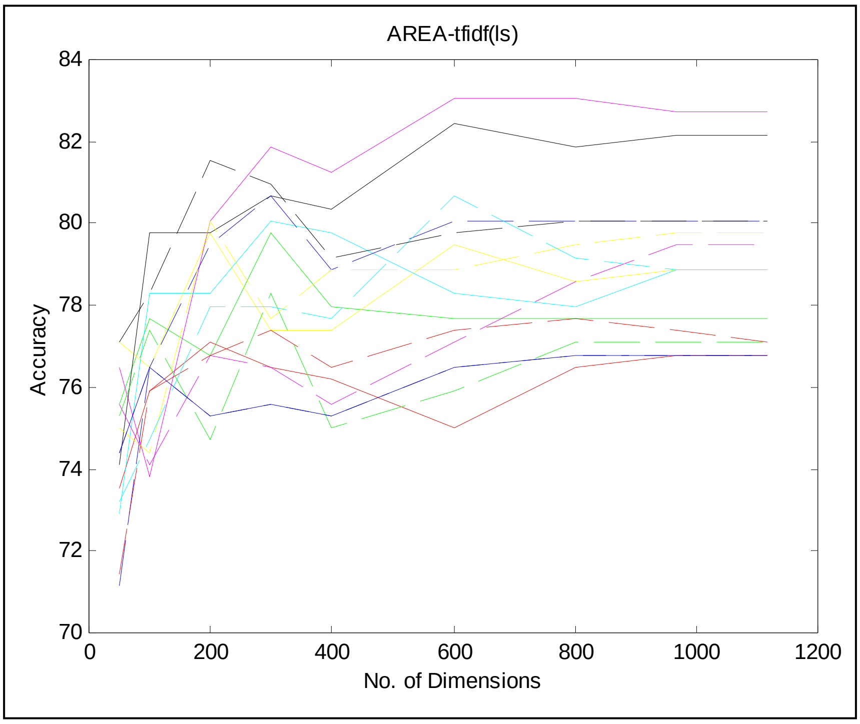


Figure 7.2: Accuracy plots for various trials for Area Dataset

Figure 7.2 shows the corresponding plots for the different trials. As can be observed from the graph, the accuracy-dimension relationship is quite dependent on the trial under study (the choice of training and testing examples). Visual inspection reveals that, in some instances, a sizeable improvement is obtained whereas in other instances it is quite marginal or negligible. It is noteworthy that the points at which the peak values occur exhibit a wide variation.

The undulating pattern as observed in the graph could be explained in the following manner. The criteria for dimension reduction is based on the fact that the particular dimension that is being reduced causes the least loss in information compared to the matrix from which it is being reduced. However, this need not necessarily correlate with the classification accuracy. For example, a particular dimension that is removed earlier, that which has caused a comparatively smaller loss in information, might have been important in distinguishing between the different classes as opposed to a particular dimension that was removed later which might not be as discriminating. In such instances, it is possible for the accuracy to exhibit such undulating behavior as observed in the graph.

In order to compare an 'averaged' behaviour, the individual accuracies from different trials for a given dimension, have been averaged and plotted in Figure 7.3. It must be pointed out that the number of dimensions at which we commence the removal is not the same as the original number of features. In fact, the removal starts from the smallest dimension at which the singular value is non-zero.

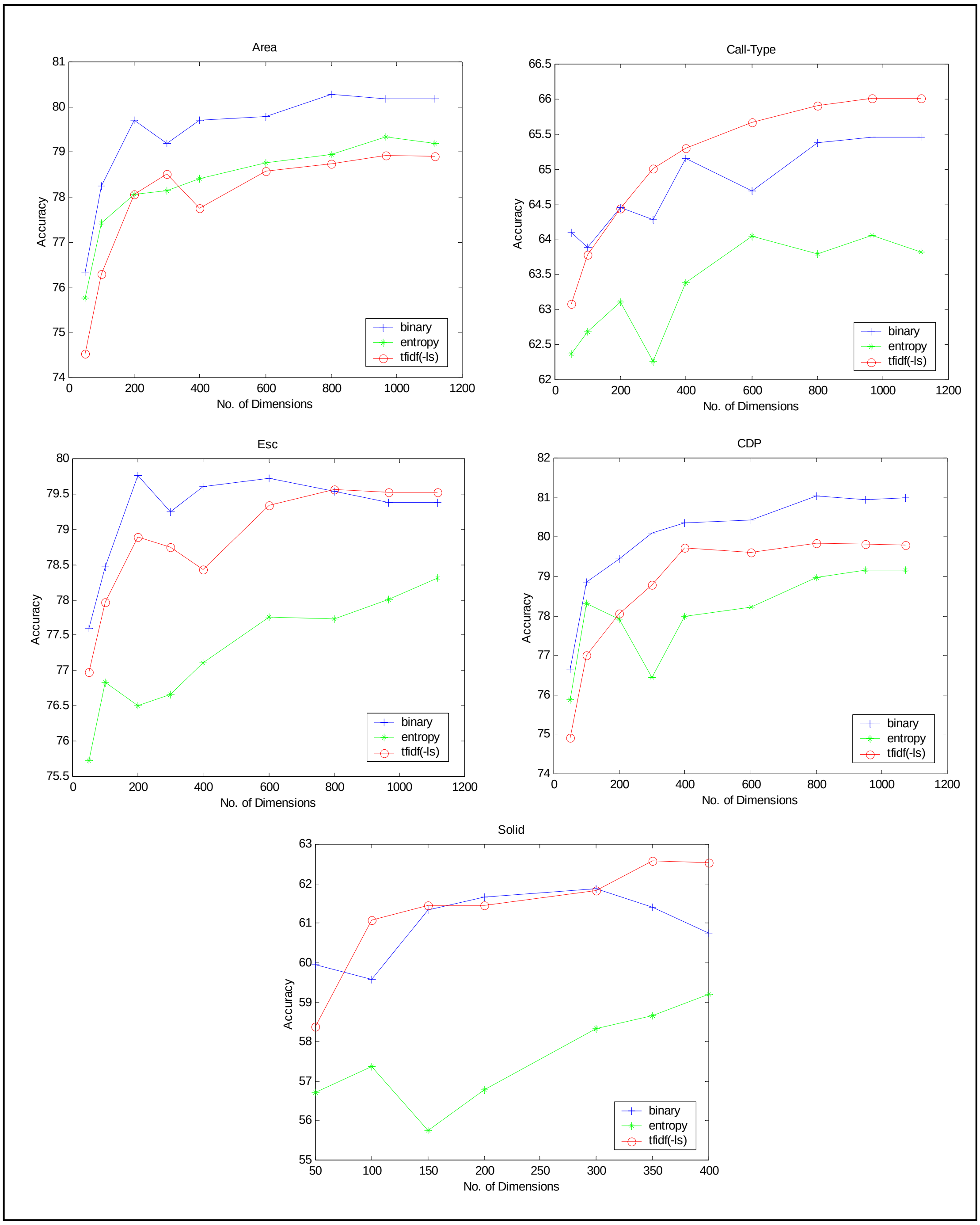


Figure 7.3: Variation of averaged accuracy with dimension reduction

As can be seen from the plots, the different weighting schemes exhibit slightly different profiles with dimension reduction. It is worth noting that the binary weighting scheme results in a reasonable improvement for the *Esc* and the *Solid* datasets. In fact, as much as 1% improvement in accuracy with about 50% reduction in dimension is observed for the *Solid* dataset. The other weighting schemes do not exhibit such improvements.

In general, the improvement in classification accuracy is marginal for the datasets investigated. Studies by Cristianini et al. (2000) and Yang (1995) on the use of LSA for text classification also reveals similar magnitudes of improvements, though different performance measures had been used.

Possible explanations for the marginal performance of LSA for the investigated datasets could be that:

- the size of the investigated datasets is not very large,
- the dataset records are short and as a result not to many related concepts are found in them,
- a reasonably consistent vocabulary usage is present, given that only a few personnel are keying-in the problems, (Hull et al., 1996)

all of which might deem that the co-occurrence of terms might not be as pronounced as one might expect for LSA to produce better results.

7.2.3 Hypothesis Testing

7.2.3.1 Performance Improvement with LSA

In order to objectively determine if the use of singular value decomposition causes any significant improvement, statistical tests are performed. For a given dataset and weighting scheme combination, for each trial, let the difference between the peak classification accuracy obtained with a reduced dimension and the accuracy obtained using the original feature set be D = $\text{Accuracy}_{\text{max}} - \text{Accuracy}_{\text{original}}$.

The null and alternative hypothesis tested are:

$$H_o\text{: } \mu_D = m$$

$$H_1\text{: } \mu_D > m \tag{7.6}$$

where m = 0, 0.5%, 1.0% and μ_D is the mean improvement in accuracy obtained due to dimension reduction. Table 7.2 shows the different p-values obtained when the values of *m* are changed.

A value of m = 0.5 would represent a null-hypothesis test as follows:

$$H_o\text{: } \mu_D = 0.5$$

$$H_1\text{: } \mu_D > 0.5 \tag{7.7}$$

The p-values given in the table would correspond to the significance value associated with the test. For a 95% confidence interval, rows that have a p-value entry of less than 0.05 would imply that the null hypothesis is rejected. Almost all except one weighting scheme-dataset combination was found to be significant when m = 0. **This shows that in general the LSA approach improves the classification performance**. As expected, from inspection of the table, it can be seen that as m increases, the number of dataset-weighting combination with p-values less than 0.05 decreases (such instances are shown with an asterix in the table). With m = 0.5%, there are only 6 dataset-weighting scheme combinations that are significantly different, whilst with m = 1%, there is only one, Solid_binary.

Table 7.2: p-values of T-test for testing improvement in accuracy due to SVD

Dataset-weighting	p-value (m=0.0)	p-value (m=0.5)	p-value (m=1.0)
CDP_binary	0.0219*	0.5027	0.9786
CDP_entropy	0.0003*	0.0117*	0.2527
CDP_tfidf-(ls)	0.0007*	0.1159	0.9226
Solid_binary	0.0017*	0.0065*	0.0242*
Solid_entropy	0.009*	0.0409*	0.151
Solid_tfidf-(ls)	0.0004*	0.0062*	0.0858
Area_binary	0.0005*	0.0939*	0.8994
Area_entropy	0.023*	0.3442	0.904
Area_tfidf-(ls)	0.005*	0.2683	0.9455
ESC_binary	0.0003*	0.028*	0.5798
ESC_entropy	0.1265	0.6469	0.965
ESC_tfidf-(ls)	0.0025*	0.3314	0.9856
Call-Type_binary	0.0042*	0.1275	0.7498
Call-Type_entropy	0.0099*	0.0612	0.2607
Call-Type_tfidf-(ls)	0.0191*	0.5047	0.9818

It is interesting to note that the *Solid* dataset gave the best performance even though it was the smallest dataset with only 483 examples. It could be said that the co-occurrence pattern is more pronounced in this dataset as opposed to others. Thus a smaller number of examples are sufficient.

7.2.3.2 Performance Difference due to Weighting Schemes

In order to determine if some weighting schemes were significantly better than the others, a Duncan's multiple range test was carried out. The null hypothesis would be H_o: $\mu_{Di} = \mu_{Dj}$ for all $i \neq j$ where μ_{Di} is the mean accuracy improvement due to the *i-th* weighting scheme. It was found that there was no statistically significant difference (at the 95 % confidence interval) in the mean improvements between the various weighting schemes for all datasets except *CDP*. For this dataset, the entropy weighting scheme was found to be better than the other two.

7.2.4 Link Between Accuracy and Relative Change Metric

Comparing Figures 7.1 and 7.3, it appears that it is inappropriate to correlate the relative change metric to the accuracy values obtained. The relative change metric values cannot be used to determine the number of dimensions to be used to achieve better accuracies. As mentioned earlier, this difficulty in correlating the two could be attributed to the fact that the criteria for dimension reduction and that for classification accuracy are different. The values for the relative change metric are determined by the inherent structure of the datasets whilst those for the classification accuracy would depend on the document-to-category mapping which involves human relevance judgement.

7.2.5 Determining the Optimal Dimension, k

From Figures 7.1 & 7.3, the optimal dimension *k* which maximizes classification accuracy, is seen to be very much dependent on the dataset under study, the weighting scheme used and even the trial under consideration. Hence it would be difficult to suggest an optimal *k* value. In order to determine the optimal *k* values shown in Table 7.3 the averaged results in Figure 7.3 are used.

Table 7.3: Dimension at which peak performance is observed

Dataset	Binary	Entropy	Tfidf-ls
Area	800	968	968
Call-Type	968	600	968
Esc	200	1117	800
CDP	800	950	400
Solid	300	400	350

As can be seen from the table, these peak accuracy values occur at a much higher dimension as compared to the widely proposed values between 100 and 300. Table 7.4

shows the range of the difference in accuracies between the maximum value and the accuracy values (highest and lowest) obtained at the dimensions of 100, 200 and 300.

Table 7.4: Difference in accuracy from maximum if k=100,200,300

Dataset	Binary	Entropy	Tfidf(-ls)
Area	(-0.575, –2.024)	(-1.29, –1.925)	(-0.417, –2.638)
Call-Type	(-0.995, -1.572)	(-0.955, -1.791)	(-0.995, -2.229)
Esc	(0, -1.293)	(-1.473, -1.81)	(-0.677, -1.592)
CDP	(-0.934, -2.181)	(-0.852, -2.720)	(-2.824, -1.060)
Solid	(0, -2.311)	(-0.860, -2.419)	(-0.753, -1.505)

As can be seen from Table 7.4, by operating at dimension values of 100 to 300 we would be about 0-3% points away from the optimal performance.

Hence from the above study, it can be said that;

1) The improvement in classification accuracy using LSA was not found to be dependent on the weighting scheme, for four out of five datasets investigated.
2) There is no link between the relative change metric and the classification accuracy results.
3) Latent semantic representation improves the accuracy only marginally, with only 6 of the 18 weighting-dataset combinations showing an improvement of more than 0.5%.
4) The optimal number of dimensions is dependent on the dataset as well as the weighting scheme under study. It is not quite possible to recommend an optimal number of dimensions *k*, for the investigated datasets.

7.3 Summary

In general, statistical tests carried out show that latent semantic representation provides some improvements to the classification results obtained, albeit marginally. The amount of improvement was dependent on the dataset under consideration. There was only one dataset-weighting scheme combination for which performance improved by more than 1%. The overall marginal performance obtained using the latent semantic representation may possibly be explained by the fact that the co-occurrence pattern may not be very pronounced in these investigated datasets. The relative change metric, which provides an indication of the loss of information due to dimension reduction, does not provide a good correlation with the accuracies obtained. Further, it has been found that the optimal value of reduced dimension, *k*, depends to a large extent on the dataset under study as well as the weighting scheme used. Unlike previous recommendations for large homogenous datasets, the optimal values of *k* are generally not between 100 and 300. They exhibit a much wider spread.

CHAPTER 8

FILTER BASED FEATURE SELECTION SCHEMES

For classification purposes in textual datasets, given the large number of features, some of them are usually found to be ‘unimportant’. In fact some of these ‘unimportant’ or ‘irrelevant’ features can actually deteriorate the classification accuracy (Yang, 1995). The process of determining the appropriate set of features for the classification task is known as feature selection. In this chapter three feature selection schemes, namely; Information gain, Markov Blanket and a Corpus based scheme, are studied. These approaches are filter based, implying that the ordering of the features is independent of the classifier used but rather based solely on the feature selection algorithm. This approach allows a large number of features to be studied. Extensive experiments are carried out to determine the appropriateness of these feature selection methods on the datasets being investigated. A comparison is also carried out to determine the best algorithm for the given task.

8.1 Introduction

Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and allows learning algorithms to operate faster and more effectively. In some cases, accuracy on future classification can be improved. In others, the result is a more compact, easily interpreted representation of the target concept. The objective of feature selection is three-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors and providing a better understanding of the underlying process that generated the data (Guyon and Elissee, 2003). Feature subset selection schemes can be grouped together in three classes; *embedded, wrapper and filter* (Blum and Langely, 1997, Kohavi and John, 1997).

The Wrapper methodology recently popularised by Kohavi and John (1997) treats feature selection as a wrapper around the classification algorithm. In fact, the classification algorithm is run each time a decision is made to either include or remove a feature. The general argument for wrapper approaches is that the induction method that will use the feature subset should provide a better estimate of the accuracy than a separate measure that may have an entirely different inductive bias. The major disadvantage of wrapper methods is its computational cost, which results from calling the induction algorithm for each feature set considered. This has prevented the use of such methods on very large datasets, such as textual databases (Forman, 2003).

Embedded methods, as the name implies, incorporate variable selection as part of the training process. As such, these methods are more efficient than the wrapper approaches in that they make better use of the available data. By not needing to split the training data into a training and validation set, they reach a solution faster by avoiding retraining of a predictor from scratch for every variable subset investigated. Cart, Quinlan’s ID3 and C4.5 are examples of embedded methods that carry out variable selection by recursive partitioning.

Filter based approaches were termed as such (John et al., 1994) because such methods filter out irrelevant attributes before the induction occurs. The filter model has several characteristics outlined as follows (Liu and Motoda, 1998):

a. It does not rely on a particular classifier's bias, but on the intrinsic properties of the data, so the selected features can be used to train different classifiers.
b. It is much cheaper since, to select the features the classifier need not be run unlike in the wrapper based approach.
c. It can be used to handle larger data sets due to its low demand on resources.

The filter based approach is the most popular approach used in feature selection in textual databases. Most of the other feature subset selection approaches used in machine learning are not designed for situations with a large number of features (Forman, 2003). The usual way of learning on text defines a feature for each word that occurred in the training documents. This can easily result in several tens of thousands of features. Most methods for feature subset selection that are used on text are very simple compared to the described methods developed in machine learning. Basically, some evaluation function that is applied to a single feature is used. All the features are independently evaluated, a score is assigned to each of them and the features are sorted according to the assigned score. Then, a predefined number of the best features are taken to form the solution feature subset.

In our experiments we would be studying filter based methods to determine their effectiveness on the datasets we have been studying all along. In particular, we are interested in addressing the following issues:

1) How effective is a filter based approach for the investigated datasets?

2) Which of the investigated algorithms performs better, with SVM used as the classifier?
3) Is it possible to recommend an adequate number of features for such datasets?

Prior to examining the filter based methods investigated in this study, a review of related literature is presented in the following section.

8.2 Review of Filter Based Approaches

Given the large number of features that are present in textual classification problems, many filter based feature selection approaches have been proposed in the literature. Most of the studies revolve around the development of a new feature selection technique or carry out a comparative study on various existing techniques or a combination of both. Some of these studies are described below.

In their study, Li et al. (2003) made use of summaries for their feature selection schemes. In their scheme, a model is first trained using the texts and their corresponding summaries. This model represents the conditional probabilities of finding a particular keyword given a text. These conditional probabilities are then used as features in the classification task.

Yang et al. (2002) stated that for some commonly used measures such as document frequency (DF), information gain (IG) and mutual information (MI) the drawback is that, for a single document, a recurrent term is treated the same way as a rare term, since only the existence of the word is taken into account. Hence they propose a possible approach to overcome this problem by adjusting the occurrences count according to the relative term frequency, thus stressing those recurrent words in each document.

Kolcz et al. (2001) also studied the effectiveness of summarization techniques for the task of document categorization. The words from the extracted summaries using various techniques were used as features for the documents. Their scheme produced comparable performance when evaluated against the Mutual Information method on the Reuters dataset, using SVM as the classifier.

Ko and Lee (2001) used association word mining as a feature selection scheme. Documents are represented as association word vectors that include a few words instead of single words. In their paper, they also discuss the selection of confidence, support and the number of words for composing association words using the Apriori algorithm, given that these factors affect classification accuracy.

Baker and McCallum (1998) studied the use of distributional clusters of words for feature reduction. Their approach clusters words into groups based on the distribution of class labels associated with each word. Given the use of class-label information, the authors argue that their method is able to carry out more aggressive feature selection than other schemes like latent semantic indexing. The results of their study on some real world datasets reveals that their scheme performs better than semantic indexing, class based clustering, feature selection by mutual information, or Markov-blanket-based feature selection.

For our study we focused on three techniques; namely Information Gain, Markov Blanket and a corpus based feature selection technique.

Information Gain (IG) is one of the most commonly used feature selection schemes in most comparative experimental studies. Although simple in concept and implementation it has been shown to be effective in quite a number of studies (Forman, 2003; Yang and Pederson, 1998) which explains its widespread use (Joachims, 1997 and Xing et al., 2001). Given its ease of use and promising results, IG was used as one of the feature selection schemes in this work.

Markov Blanket (MB), a feature selection scheme originally proposed by Koller and Sahami (1996) has been more widely used in conventional Machine Learning applications than to feature selection in textual documents. The main reason for that is the high demands for computer resources it places (Baker and Mccallum, 1998). Recently however, efforts have been taken to make the scheme more scalable (Aliferis et al., 2003). In some instances, it is also common to use Information Gain algorithm first followed by the Markov Blanket algorithm as a feature reduction strategy (Xing et al., 2001). The MB approach however is very promising in that it not only removes irrelevant features but also redundant features, which is quite a common problem in textual applications. Since we are not dealing with features in the order of tens or hundreds of thousands, the application of this algorithm is manageable for our application.

The corpus based approach was originally proposed by Wilbur and Sirotkin (1992) for generating domain specific stoplists and was subsequently used by Yang and Wilbur

(1996) for removing redundant words for text categorization. The drawing factor towards its application is the fact that its feature selection scheme is independent of the class labels.

These three approaches are discussed in further detail in the following subsections.

8.2.1 Information Gain Approach

Information gain is frequently employed as a term-goodness criterion in the field of machine learning. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a word in a document.

Let c_1,c_2,..c_k denote the set of possible categories. The information gain of a word *w* is defined to be:

$$IG(w) = -\sum_{j=1}^{k} P(c_j)\log_2 P(c_j) + P(w)\sum_{j=1}^{k} P(c_j \mid w)\log_2 P(c_j \mid w) + P(\overline{w})\sum_{j=1}^{k} P(c_j \mid \overline{w})\log_2 P(c_j \mid \overline{w}) \quad (8.1)$$

where $P(c_j)$ is the probability that any randomly selected document belongs to class $c_{j,}$ $P(w)$ the probability that any randomly selected document contains the word *w* and $P(c_j|w)$ is the conditional probability of finding a document belonging to class c_j, given that it contains word *w,* whilst $\overline{w}$ refers to the situation where the word *w* does not exist in a document.

Here $P(c_j)$ can be estimated from the fraction of documents belonging to class c_j and $P(w)$ from the fraction of documents in which the word *w* occurs. Moreover, $P(c_j|w)$ can be computed as the fraction of documents from class c_j that have at least one occurrence of word, *w*, and $P(c_j \mid \overline{w})$ as the fraction of documents from class c_j that does not contain word, *w*.

The information gain in Equation (8.1) is similar to that provided in Equation (4.4), Section 4.3.2. However, in that section, the gain ratio, defined as the information gain over the split measure, was used as measure of computing the worth of a term. This was the case because the information gain tends to prefer attributes with a large number of possible values and the split measure helps correct this. In the current context of feature selection however, each attribute (term) has only two values (1 or 0) since a binary term representation is considered. Hence information gain is a sufficient measure.

The lower the information gain the less important is the feature. For example, if a particular word is uniformly distributed over the various classes then it will have an information gain value of zero. However, if a particular word is only found in a single class then it becomes a good differentiating feature and would therefore have an information gain value of 1.

8.2.2 Markov Blanket Algorithm

The Markov Blanket Algorithm proposed by Koller and Sahami (1996) is an information theoretic filtering method. The MB algorithm uses the conditional distribution *Pr*(*C*|*F*=*f*) where,

F = set of features (F_1, … , F_n)

f = assignment of values (f_1, … , f_n) onto ***F***

C = class variable

as the basis to build its feature reduction scheme. Suppose we have a reduced feature set, ***G***, where one feature F_i from ***F*** is excluded, i.e. ***G*** = ***F*** – {F_i}. Let $\boldsymbol{f_G}$ be the projection of ***f*** onto the variables in ***G***. For example, for the feature vector ***f*** = (a,b), $f_{(A)}$ = (a). The probability distributions for each class, $Pr(C|\boldsymbol{G} = \boldsymbol{f_G})$ would possibly be different from $Pr(C|\boldsymbol{F} = \boldsymbol{f})$, and to measure the extent of "error" if the new distribution were used as a substitute for the original, a cross-entropy measure of μ to σ is used which is given as:

$$D(\mu,\sigma) = \delta_{\boldsymbol{G}}(\boldsymbol{f}) = \sum\nolimits_{x\in\Omega} \mu(x) \log \frac{\mu(x)}{\sigma(x)} \tag{8.2}$$

where $\mu = Pr(C|\boldsymbol{F} = \boldsymbol{f})$, $\sigma = Pr(C|\boldsymbol{G} = \boldsymbol{f_G})$, and Ω is the probability space which is the set of possible classification {c_1 , c_2 ,……. c_n }.

In order to have a metric which allows us to compare one feature set ***G*** to another, we must integrate values of $\delta_G(\boldsymbol{f})$ for different feature vectors ***f*** into a single quantity. This is done as follows:

$$\text{Expected cross-entropy, } \Delta_{\boldsymbol{G}} = \sum_{\boldsymbol{f}} Pr(\boldsymbol{f})\delta_{\boldsymbol{G}}(\boldsymbol{f}) \tag{8.3}$$

In wanting to choose a reduced dataset ***G*** to approximate ***F***, we must minimise $\Delta_{\boldsymbol{G}}$. In choosing which feature F_i to remove, we look to see if a **Markov Blanket, *M*** of feature F_i can be found. ***M*** is some subset of features in ***F***, not containing F_i, and will be considered a Markov Blanket if F_i is conditionally independent of ***F*** – ***M*** –{F_i} given ***M***. That is, F_i does not give any additional information about *C* other than what is already contained in ***M***. Thus, the Markov Blanket criterion requires that ***M*** subsumes the information {F_i} has about class *C*.

In the practical implementation of this feature selection method, it is noted that for a feature F_i, a full Markov Blanket is rarely found. What is more plausible is finding a partial Blanket that approximately subsumes the information content of the feature. Here, expected cross-entropy is given as:

$$\delta_{\mathbf{G}}(F_i \mid \boldsymbol{M_i}) = \sum_{\mathbf{f}_{\mathbf{M}_i}, f_i} Pr(\boldsymbol{M_i} = \boldsymbol{f}_{\boldsymbol{M}_i}, F_i = f_i) \cdot D(Pr(C \mid \boldsymbol{M_i} = \boldsymbol{f}_{\boldsymbol{M}_i}, F_i = f_i), Pr(C \mid \boldsymbol{M_i} = \boldsymbol{f}_{\boldsymbol{M}_i})) \tag{8.4}$$

where,

F_i is the feature that is considered for removal,

$\boldsymbol{M}_i$ is the candidate Markov Blanket for F_i,

$\boldsymbol{f}_{\boldsymbol{M}i}$ contains values assigned to $\boldsymbol{M}_i$ and

f_i is the value assigned to F_i.

The feature F_i which has the lowest $\delta_G(F_i|\boldsymbol{M}_i)$, will be removed.

The above procedure can be implemented in the following steps:

(1) Compute the correlation factor $\rho_{ij} = \dfrac{Cov(F_i, F_j)}{Stdev(F_i)Stdev(F_j)}$ of every pair of features $\boldsymbol{F_i}$ and $\boldsymbol{F_j}$

(2) Define $\boldsymbol{G} = \boldsymbol{F}$

(3) For each feature $\boldsymbol{F_i} \in \boldsymbol{G}$, let $\boldsymbol{M_i}$ be the set of K features $\boldsymbol{F_j}$ in $\boldsymbol{G} - \{\boldsymbol{F_i}\}$ for which ρ_{ij} has the largest magnitude

(4) Compute $\delta_G(F_i \mid M_i)$ for each i.

(5) Choose the i for which this quantity is minimal, and define $\boldsymbol{G} = \boldsymbol{G} - \{\boldsymbol{F_i}\}$

(6) Repeat steps 3 onwards until a certain pre-specified number of features have been eliminated

Theoretically, the MB algorithm requires $O(n^2(m+\log n))$ (Koller and Sahami, 1996) operations for computing the correlation matrix and sorting it, whilst the subsequent feature selection process requires $O(r.n.k.m.2^k.c)$ time, where *n* is the initial number of features, *m* the number of instances, *r* the number of features to be eliminated, *k*, the small, fixed number of conditioning features and *c* is the number of classes.

The Markov Blanket criterion aims to only remove attributes that are unnecessary. Such attributes are either totally irrelevant to the target concept, or are redundant given other attributes. It is one of the few feature selection techniques that is able to deal with both these types of unnecessary features.

8.2.3 Corpus Based Approach

The corpus based approach (Yang and Wilbur, 1996) measures the importance of a word by measuring the word strength, which shows how informative a word is in identifying related documents, and is computed based on word distribution over related documents. The strength, *S*, of a word, *t*, is defined as the probability of finding *t* in a document which is related to any document in which *t* occurs,

$$S(t) = \Pr(\text{word } t \text{ is in document } y|\text{word } t \text{ is in a related document } x) \qquad (8.5)$$

where *x* and *y* denote an arbitrary pair of distinct but related documents. For computing word strength, the availability of a training corpus with relevance judgement between documents would be ideal. Such relevant judgments are not available in real-world applications. Wilbur and Sirotkin (1992) have shown that one can relax the relevance criterion by assuming two documents are related to each other if they have many words in common. That is, one can use the conventional cosine-efficiency of two vectorized documents to measure the similarity and identify a pair of documents as related if their cosine-similarity value is above a threshold. Using a pair of related documents, one can approximately compute word strength as

$$s(t) = \frac{\text{number of documents pairs in which word } t \text{ co-occurs in both documents}}{\text{number of documents pairs in which word } t \text{ occurs in first document}} \qquad (8.6)$$

where the ‘first’ document can be any training document. That is, there are no constraints on the first document of a pair; if (x,y) is a pair of related documents, then (y,x) is also a pair of related documents.

Hence the procedure of stop word identification consists of the following steps:

(1) Compute similarity values of all pairs of training documents.

(2) Select the document pairs whose similarity values are above a document relevance threshold (chosen experimentally) as related document pairs.

(3) Compute the strength of each word using the related document pairs.

(4) Select the words with strength values equal to or higher than a threshold value.

As an addition to the above procedure, a minor modification was carried out in an attempt to use the information of the classes of the documents since they are available. The documents belonging to the same class can be treated to be related document pairs. This would be a good relevancy judge between related documents. This scheme would be referred to as *Corpus-Class* whilst the original scheme as *Corpus-Thr*. In general, however, this approach need not make use of class information unlike the other two approaches.

8.3 Feature Selection Experiments

The following sections outline the experiments carried out using the various feature selection methods. The documents have been represented using the binary weighting scheme. As found previously, this scheme yielded good results for the datasets investigated. Furthermore, this scheme is simple and hence makes it amenable for use with algorithms such as Markov Blanket and Information Gain in which probability density functions would need to be specified if feature values were continuous in nature.

The feature selection algorithms are initially employed on the training data, from which an order of feature removal is determined. This order is then used to remove features from the testing data and the corresponding accuracies are obtained. The features are removed in blocks of 25. Preliminary experiments show that the accuracy profile is sufficiently captured with this choice without making excessive demands on computational resources. At each value of the feature number, 15 stratified trials are carried out. As in the case of dimension reduction using SVD, all five datasets are studied. Given the extensive number of experiments required, all experiments were conducted using fixed parameter settings of c at 500 and σ at 0.03 using the gaussian kernel, for all datasets except *Esc* for which σ was set to 0.1 while maintaining c at 500.

8.3.1 Experiments with Information Gain (IG) Approach

Figure 8.1 shows the variation of the information gain values with the features for the *Call-Type* and *Solid* and *Esc* datasets. The other two datasets exhibit a similar variation. As can be seen from the figure, the information gain values are very similar for the various trials considered. They exhibit a smooth variation with the feature number. The information gain values are not very high, with a maximum of about 0.2, indicating that there are no subset of words present in the datasets that are highly differentiating.

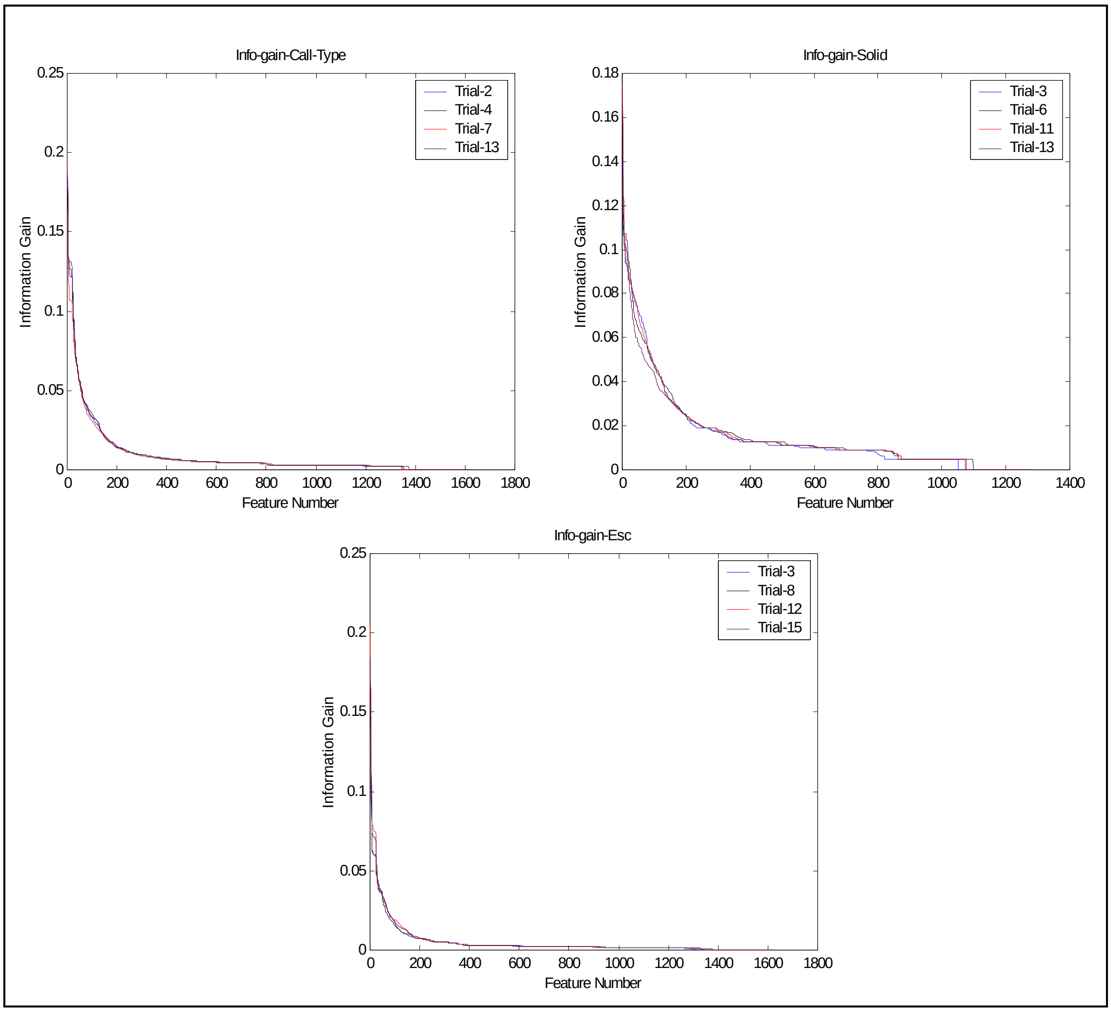


Figure 8.1: Information-gain values for Call-Type, Esc and Solid datasets

Figure 8.2 shows the accuracy values for the various datasets as the features are removed (from right to left) using the information gain criteria. Statistics such as the minimum, maximum, mean, 25th and 75th percentiles and the median are plotted for the features considered. This display was used as opposed to box plots for clarity reason. The box plots would be cluttered given the large number of features considered.

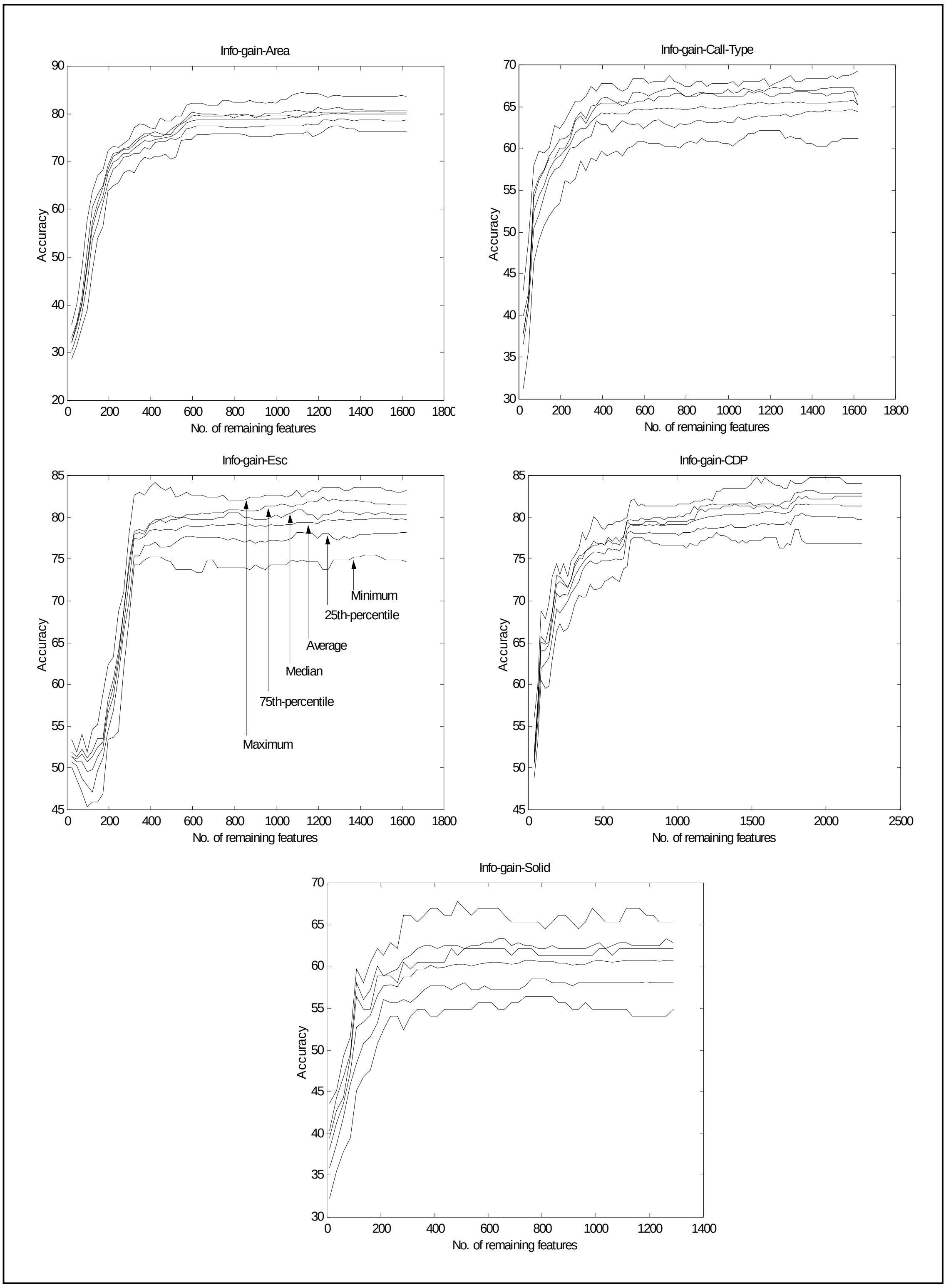


Figure 8.2: Accuracy values for IG based feature reduction for various datasets

From the plots, the general profiles exhibited by the various datasets are largely similar with some subtle differences between them. The spread of the points for a given set of features is about the same, with a 8 percentage point difference in accuracies between the maximum and the minimum and about 4 percentage point inter-quartile range (IQR), in most cases.

Generally as the features are removed (from right to left) according to the information gain values, the accuracy values are more or less stable up to a certain number of features, beyond which there is a drop in the accuracy values obtained. In most cases, the drop starts off gradually up to a certain point followed by a rather steep drop. This is seen for all the datasets except for *Esc*. For the *Esc* dataset, a steep drop follows the stable region, from about 350 to about 100 features, beyond which a marginal improvement is observed. This improvement is quite unexpected since removing these sets of features, which have a high information gain value, should ideally result in an accuracy reduction. In this instance, it seems that the individual contribution of features with high information gain values are negative with respect to the classification accuracy since their removal increases the accuracy.

The number of features at which the gradual drop takes place is different for the various datasets. Table 8.1 indicates the approximate beginning, from visual inspection, of the different drop-points. It also shows the amount of accuracy reduction from the drop-point to when the dataset contains less than 25 features. As can be seen from the table, there is a sizeable amount of reduction in accuracy when only a small amount of features are present. Further the approximate drop-points are different for the various datasets.

Table 8.1: Approximate drop-points and accuracy reduction amount (Information gain)

Datasets	**Approximate Drop-points**	**Approximate Reduction**
Area	600	50%
Call-Type	400	30%
Esc	350	30%
CDP	700	25%
Solid	350	25%

Using the information gain value approach, a sizeable number of features can be reduced without much loss in accuracy. However, a significant improvement in accuracy is not seen, with feature reduction.

8.3.2 Experiments with Markov Blanket (MB) Algorithm

Figure 8.3 shows the expected cross-entropy (Equation 8.4) from the MB algorithm for selected datasets when the features are arranged in an ascending order in terms of score. As mentioned earlier, the higher the error the more important is the feature. From the plots, it can be seen that the expected cross-entropy is almost negligible for the first few hundred features removed (from right to left) before registering a relatively significant positive value. As mentioned earlier, for a single feature that has almost a zero error-score, a MB formed by the other features can be found, that almost subsumes the information in the single feature. Unlike the information gain value that exhibits a smooth variation profile, the expected cross-entropy from the MB algorithm are similar to step functions. Furthermore, the different trials exhibit a slightly different error pattern, which is quite contrasting to the situation of information gain where the various trials produce almost identical patterns.

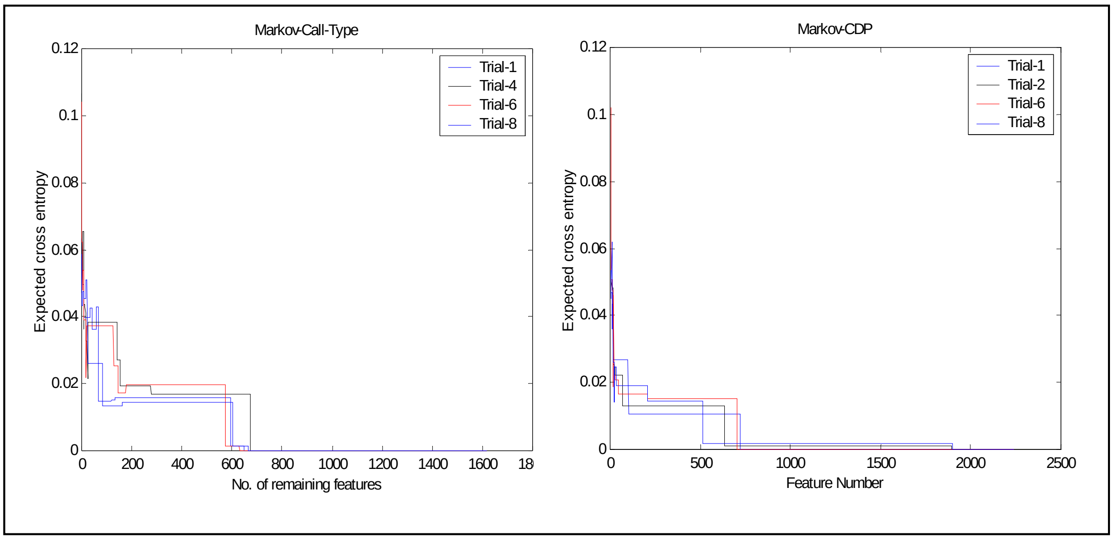


Figure 8.3: Expected cross-entropy for MB feature reduction for Call-Type and CDP datasets

Figure 8.4 shows the variation in accuracy values when features are removed in the order suggested by the MB expected cross-entropy values. The graphs look very much like those obtained with IG based feature reduction. The most distinct difference however would be the amount of drop in accuracy.

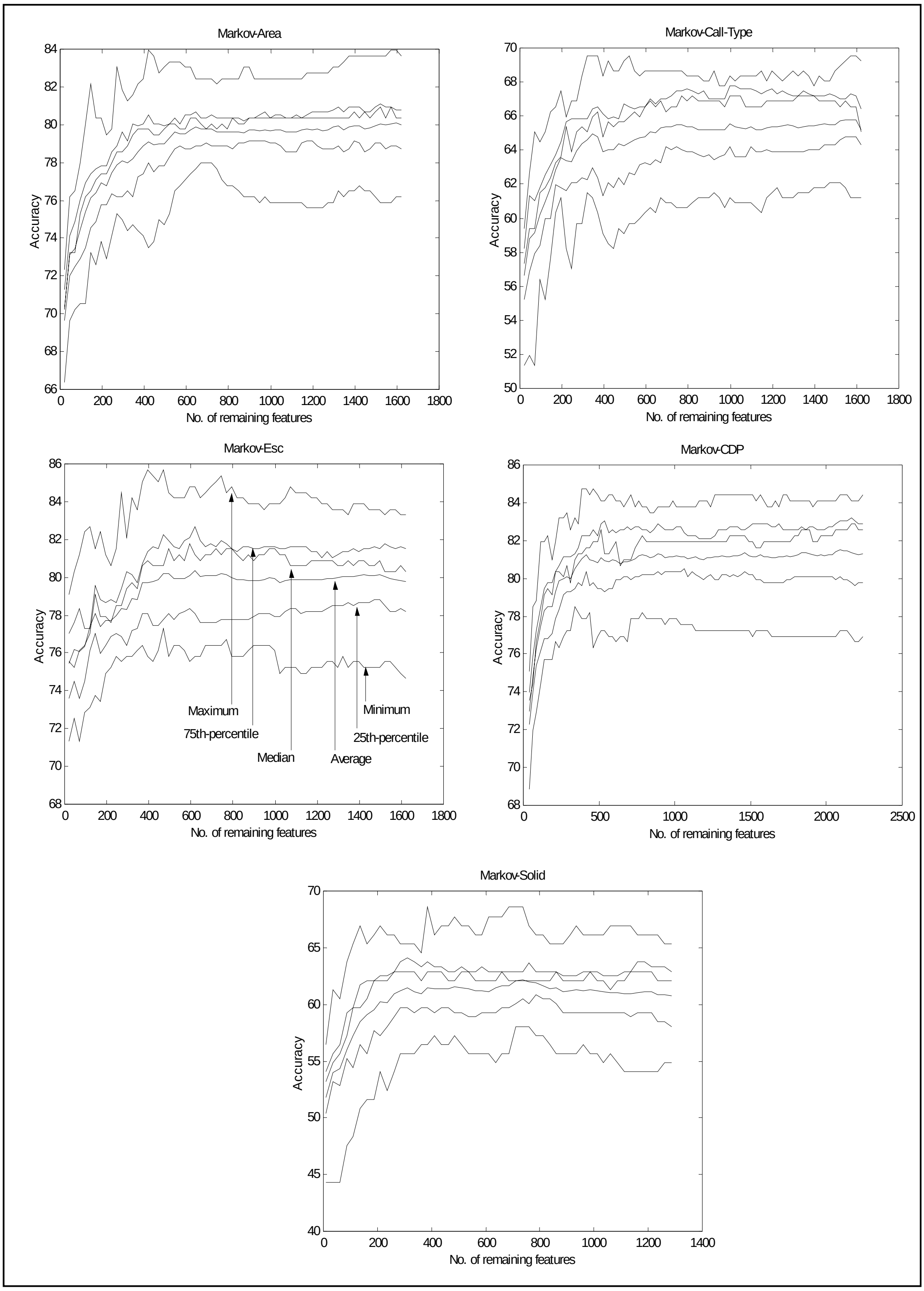


Figure 8.4: Accuracy values for MB based feature reduction for various datasets

As can be seen in Table 8.2, the amount of accuracy reduction is very much lower as compared to IG. The accuracy reduction using the MB algorithm is about 3 to 5 times smaller compared to that obtained by using the IG algorithm.

Table 8.2: Approximate drop-points and accuracy reduction amount (Markov Blanket)

Datasets	**Approximate Drop-points**	**Approximate Reduction**
Area	600	10%
Call-Type	350	8%
Esc	400	5%
CDP	400	8%
Solid	300	8%

As was in the case of the IG approach, a sizeable number of features can be reduced without much loss in accuracy using the MB approach. However, a significant improvement in accuracy is not seen, with feature reduction.

8.3.3 Experiments with Corpus based (CB) Scheme

As mentioned previously, for the corpus based scheme, the similarity between the documents is important in computing the strength of the words. In view of this, the similarity values between document pairs were computed and are displayed in the form of a histogram for the 3 different datasets, in Figure 8.5.

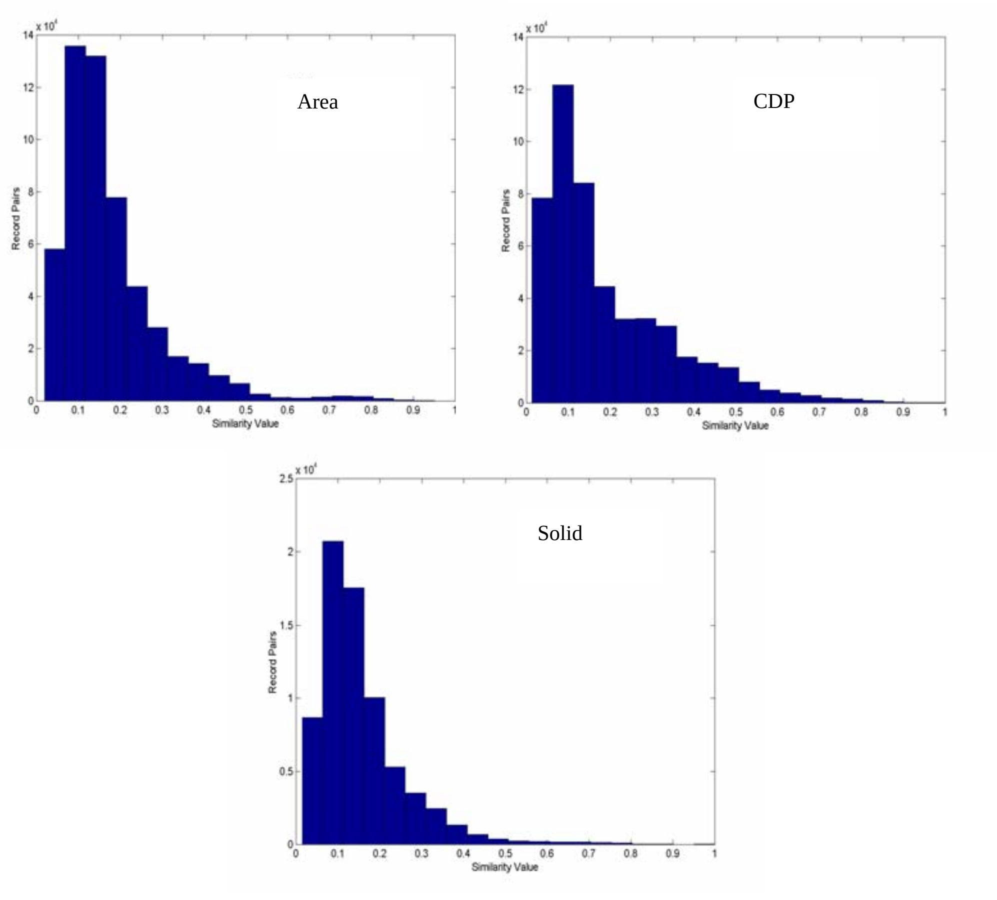


Figure 8.5: Distribution of similarity values between the record pairs for the different datasets

It can be seen that a large proportion of the documents' similarity ranges from 0.05 to 0.2 for all the three datasets investigated. In fact, there were very few document pairs that have similarity values above 0.5. Experiments carried out using the corpus based approach made use of different values of threshold to determine similar documents.

The plots in Figure 8.6 show the variation of the strength measure when the features are arranged in ascending order in terms of strength. As can be seen from the graphs, the strength measure exhibits a smooth variation. Depending on the similarity threshold that has been set there is a slight variation in the strength profiles. For a lower threshold, the strength measures are found to be lower. For the *Area* and the

Solid datasets, after the top 400 and 300 features the strength measure drops to almost zero, implying that the words beyond these are not very important. For the *CDP* dataset this value is at about 550 features.

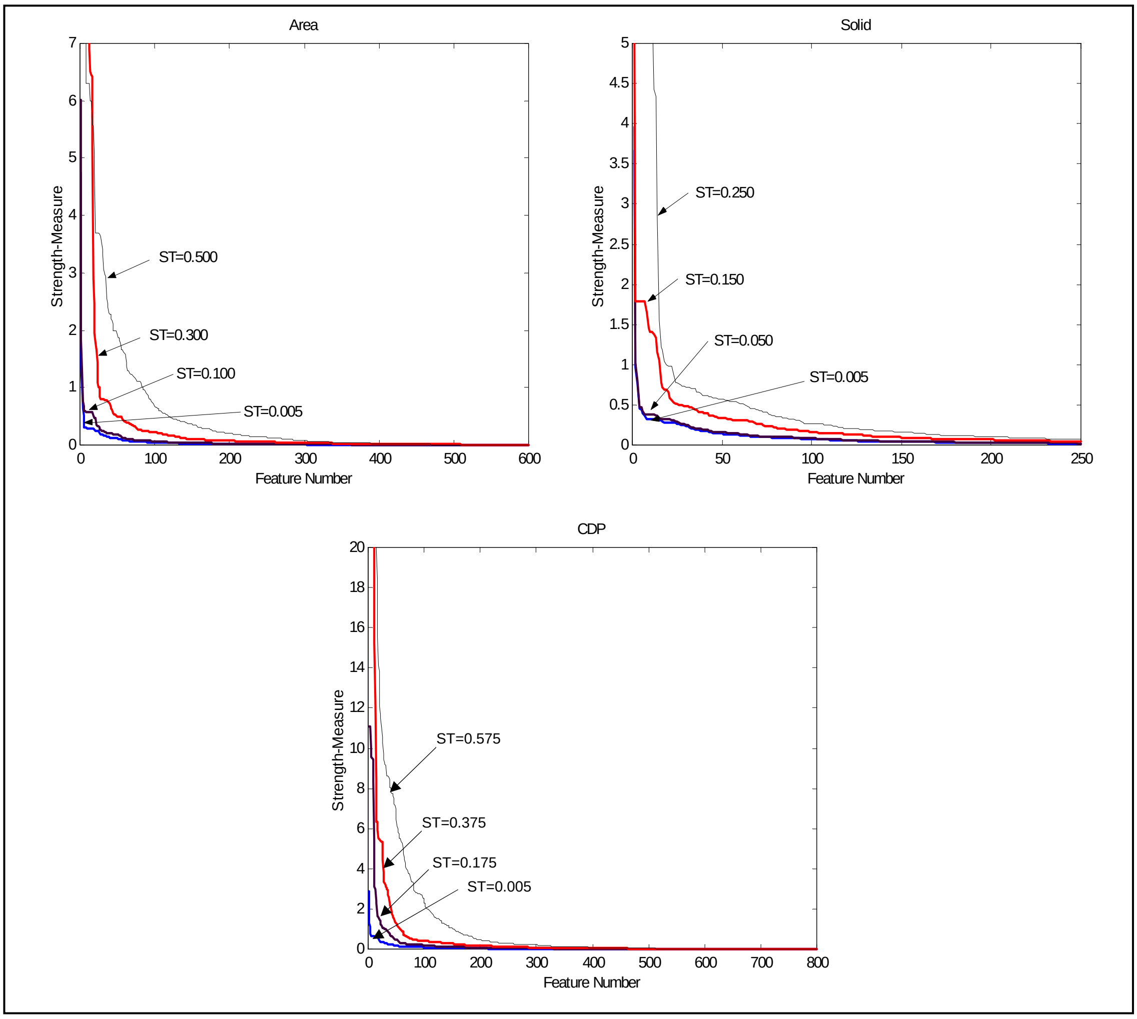


Figure 8.6: Strength Measures for Area, Solid and CDP datasets

Figure 8.7 shows the accuracy variation profiles for selected datasets and various thresholds. The last plot refers to the situation where the similarity between documents is based on the class labels. It can be seen that the accuracy variation profile is similar to those obtained from the other methods. Given the smaller number of examples, the variation from the *Solid* dataset is slightly higher than the others. It has a inter-quartile range (IQR) of about 5% whereas the IQR for the other databases is about 2-3%.

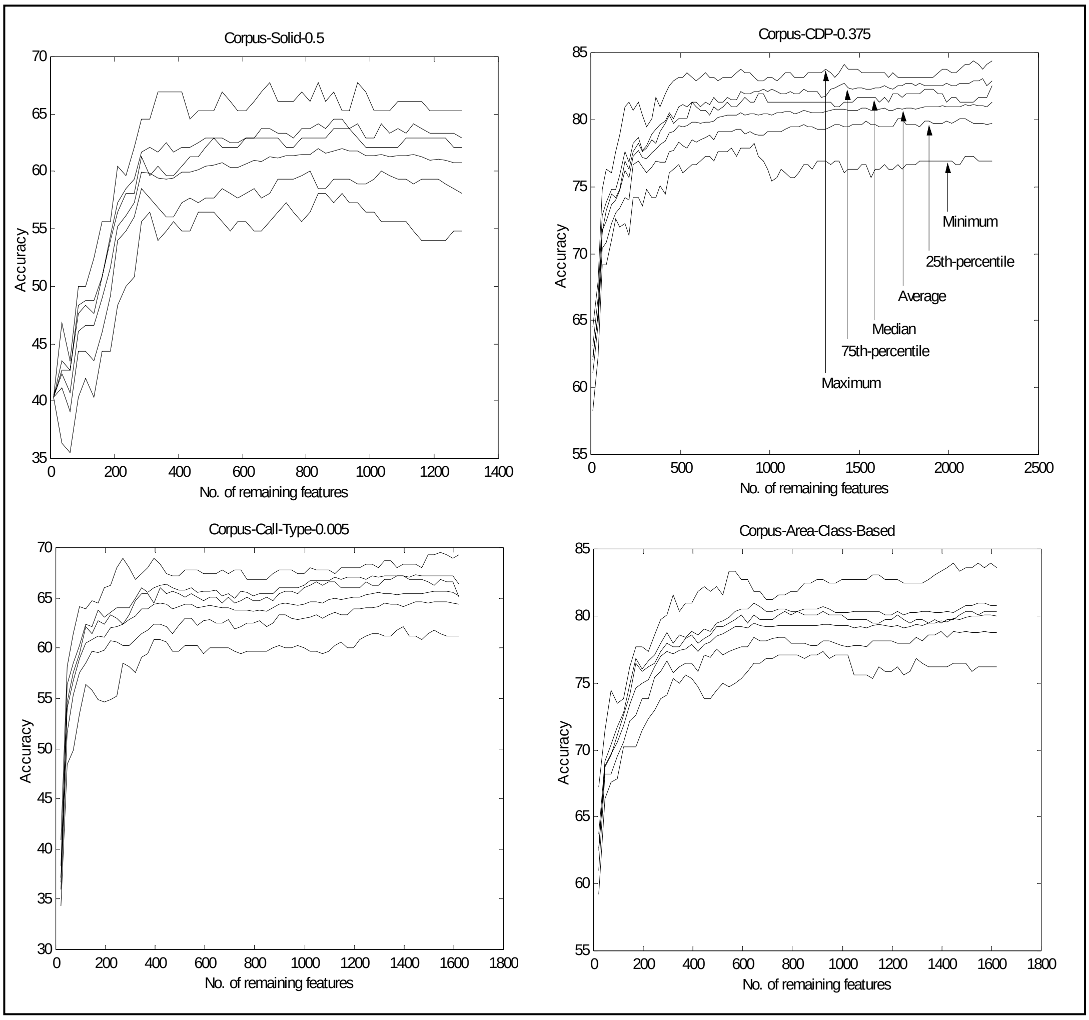


Figure 8.7: Accuracy values for Corpus based feature reduction for selected datasets and thresholds

Figure 8.8 shows the averaged (from 15 trials) accuracy values for the various thresholds selected (eg. ST= Similarity Threshold= 0.005, 0.1,…..). The accuracy obtained using the class based approach is also included. The profile follows very closely to those obtained by the previous algorithms. There is an initial stable region followed by a drop region thereafter. For most databases the drop is about 20-25%. However, in the case of the *Area* dataset, a drop of about 50% is observed.

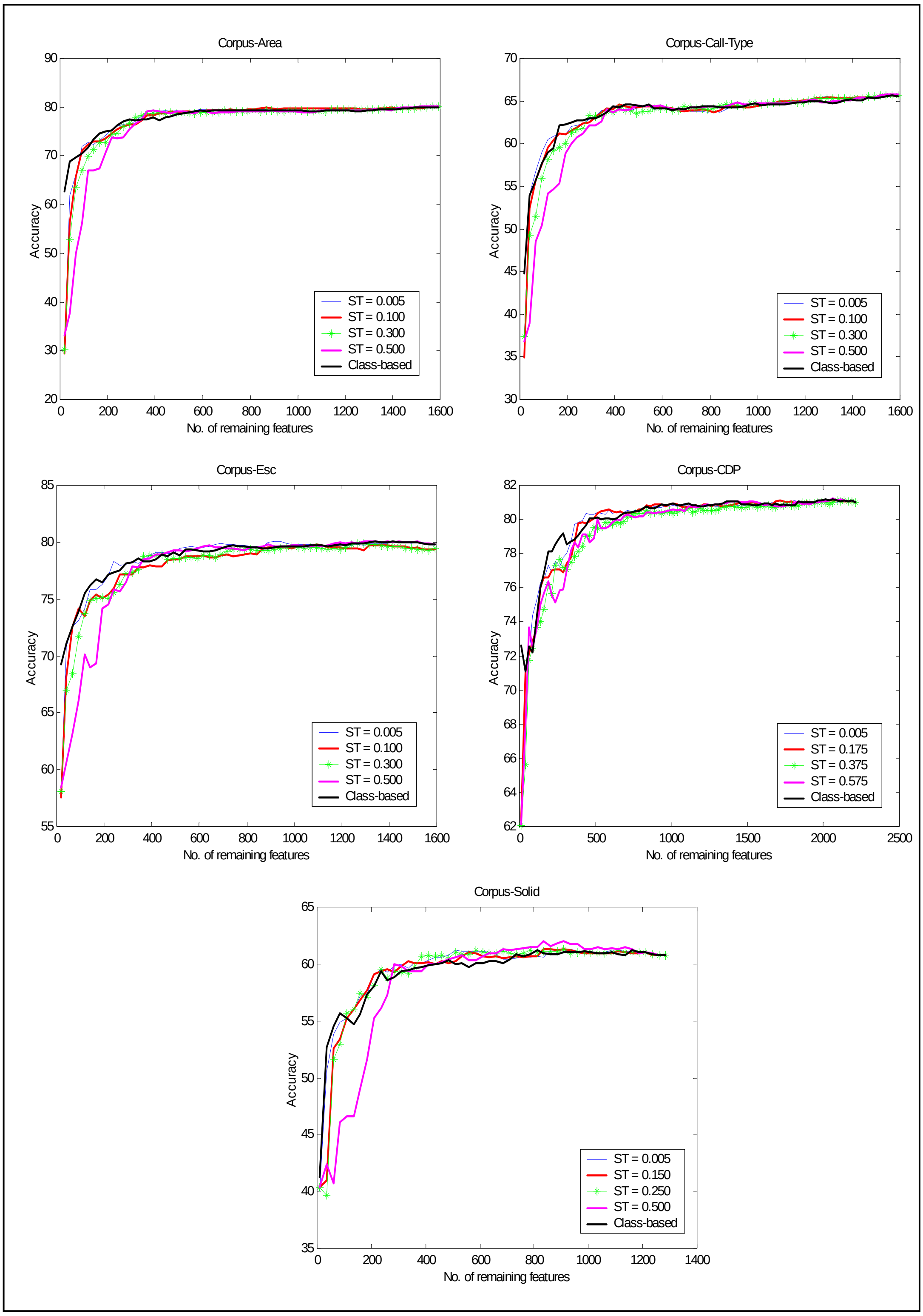


Figure 8.8: Averaged accuracy values for Corpus based feature reduction for various threshold values and datasets

It is interesting to observe that, for the datasets studied, the accuracy variation is not affected much by the similarity threshold value in the stable region. In the dropping region however, in most cases, the lower threshold seems to provide slightly better results. This could be roughly explained by the fact that in the stable region, immaterial of the threshold most of the features have a strength measure close to zero. As such removing them does not really impact the classification accuracy. As for the dropping region, as observed in Figure 8.6, it corresponds to the region in which the strength measures tend to be lower for lower thresholds. As such, removing lower strength words results in a lesser reduction in accuracy and therefore slightly better observed results for the lower thresholds.

The class based approach performs as well as the lower threshold settings. One distinct difference that can be seen for all the datasets with the exception of *Solid,* is that the drop in the accuracy is not as severe for the class based approach. For example for the case of the *Area* dataset, the accuracy from the class based approach is as high as 62% as opposed to a value of about 30% for the other threshold values. Table 8.3 below highlights these differences when the number of features is less than 25. However, it is very unlikely that one would operate at such reduced accuracies.

Table 8.3: Average accuracies for class based and threshold based schemes

Dataset	**Approximate Drop-points**	**Approximate Reduction (class based)**	**Approximate Reduction (threshold based)**
Area	400	18	50
Call-Type	400	20	25
Esc	400	10	20
CDP	750	8	20
Solid	300	20	20

Most often we are interested in operating in the stable region. Under these circumstances, the setting of the similarity threshold value is not very critical. In this event, the corpus based scheme becomes reasonably attractive since it does not require labelled examples for feature selection.

8.4 Discussion

8.4.1 Hypothesis Testing

Hypothesis testing was carried out in order to determine if there is any significant improvement in accuracy due to feature reduction. The paired t-test was used.

In our case, *d* is the difference between the mean accuracy values of the reduced feature set, ACC_{red}, and the original feature set, ACC_{org}. Hence the hypothesis tested was;

$$H_o : ACC_{red} - ACC_{org} = d$$

$$H_1 : ACC_{red} - ACC_{org} > d \qquad (8.7)$$

where *d* takes on the values of 0, -0.5 and -1. These negative values of *d* were chosen since our accuracy plots did not show any substantial improvements in accuracy. All tests were done at the 95% confidence level. As an example, rejection of the null hypothesis at *d* = -1 is equivalent to the strong conclusion that the accuracy due to the reduced feature set is no more than 1 percentage point less than the original feature set.

Figures 8.9 to 8.11 display the hypothesis test results. It shows the different feature values at which the null hypothesis is rejected for the various data sets. In the legend, *Corpus-Thr* refers to the original corpus based scheme with the threshold set to the best values obtained from Figure 8.8 whilst the *Corpus-Class* refers to the corpus based scheme which uses the class information to determine similarity between documents. For the *Area* and *Call-Type* datasets for a value of *d*= 0, the graphs are empty indicating that the null hypothesis is rejected at none of the feature values.

For a value of *d*=1.0, the null hypothesis is rejected for a large number of features. This indicates that a large number of features can be removed whilst keeping the loss in accuracy to be less than 1% from the original. For example, for the *Area* and the *CDP* dataset about a 66% and 82% of the original feature set can be removed respectively, without much loss in accuracy. Hence it could be said that **filter based approaches are effective for the investigated datasets.**

It is observed that the **MB scheme performs better than the other algorithms**. This result justifies the effort in determining the feature selection order using the MB scheme. This method as mentioned before not only penalizes features that are irrelevant to the target concept but also features that are redundant. Comparing the performance of MB to IG, which only removes features based on its relevancy to the target concept, it maybe be said that there maybe quite a number of redundant features in the investigated datasets.

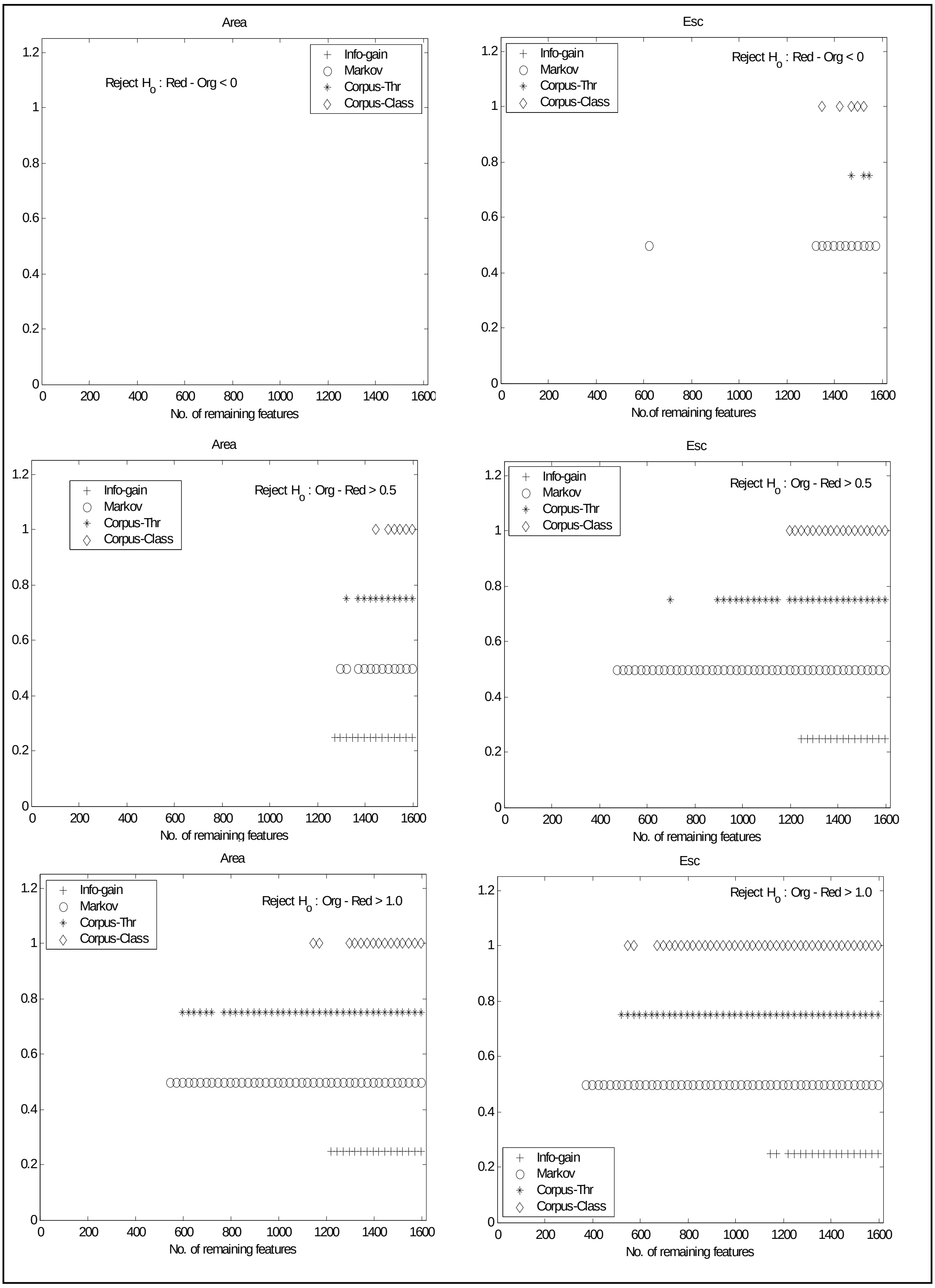


Figure 8.9: Hypothesis testing for Area and Esc datasets

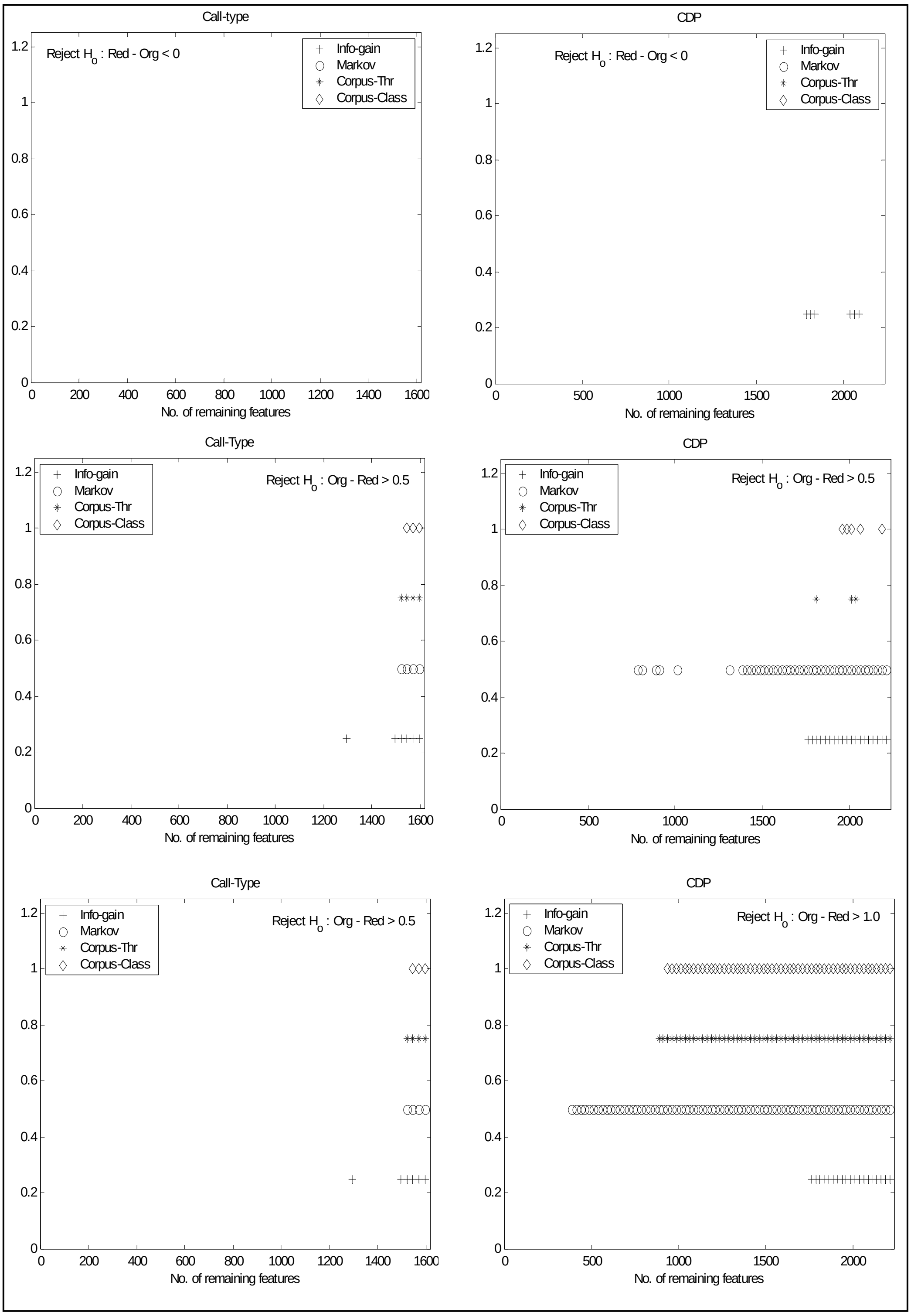


Figure 8.10: Hypothesis testing for Call-Type and CDP datasets

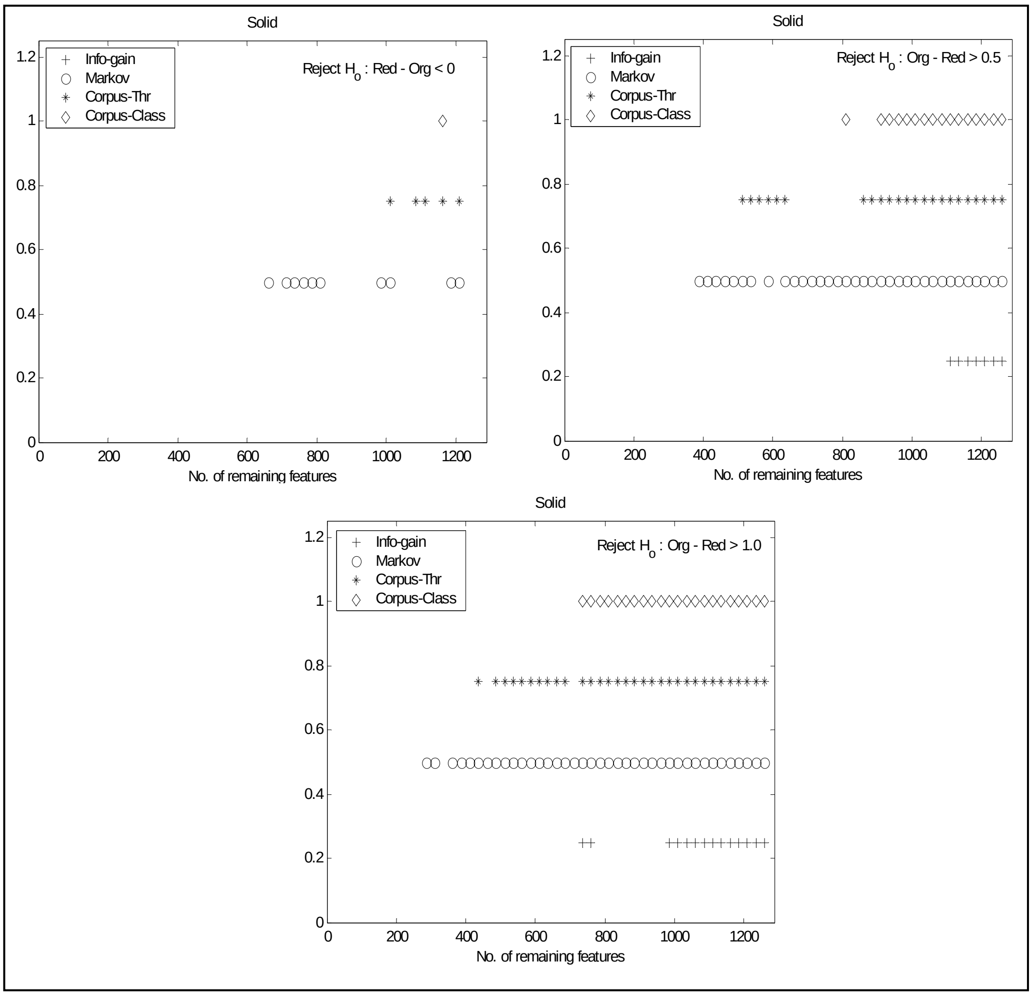


Figure 8.11: Hypothesis testing for Solid dataset

The corpus based scheme using a threshold to select similar documents also performed reasonably well. In most instances, it was either equal to or better than the corpus based scheme using class labels for similarity determination. The information gain scheme performed worst as compared to other schemes.

8.4.2 Adequate Number of Features

It would be very beneficial if an adequate number of features could be suggested for the representation of the various datasets investigated. In this case, adequate number would refer to the minimum number of features required to represent the dataset without much loss in accuracy. Table 8.4 shows this number for the value of d =-1.0 when the various algorithms are used.

Table 8.4: Minimum number of features resulting in accuracy loss of less than 1%

Dataset	MB	Corpus-thr	Corpus-Class	IG
Area	525	600	1300	1200
Call-Type	650	1050	1225	1425
Esc	375	500	675	1125
CDP	400	850	900	1750
Solid	350	450	725	950

As can be seen from the table, the minimum number of features required varies widely depending on the algorithm as well as the dataset being used. Given this variation, it would be rather difficult to recommend an adequate number of features for representing the documents. However, choosing the best algorithm, MB, it could be said that **less than 650 features is quite adequate**. In fact the *Call-Type* dataset required a higher number of features since determining the type of call is comparatively more difficult than determining the other class labels.

8.5 Summary

In this chapter, 3 different feature selection schemes, namely; Information Gain, Markov Blanket and Corpus based, were studied. Given the resource requirements, the MB algorithm is not usually used for feature selection of texts where the number of features can run into tens or hundreds of thousands of variables. However, for our application, the number of features is in the order of one to two thousands. Hence use of MB was manageable. Its use was beneficial since it was found that the MB algorithm provided the best performance, when studied on the five Call Centre datasets.

The Corpus based scheme, also produced reasonably good results. It was interesting to note that a threshold based Corpus scheme produced better results than a class based Corpus scheme. This spells well for the use of the Corpus based since no class labels are required.

Although no appreciable improvement was obtained as the features were removed, the loss in accuracy was very small even when a large number of features are removed. In fact, using the MB algorithm for the *CDP* dataset, as much as an 80% reduction in the number of features can be carried out with a loss in accuracy of less than 1% from the full feature set. Hence in general, using the above algorithms, a large percentage of features can be removed without greatly affecting the classification accuracy.

CHAPTER 9

FEATURE SELECTION WITH DESIGN OF EXPERIMENTS

In this chapter a novel feature selection approach based on Design of Experiments is suggested. It makes use of statistical ideas to select a minimal number of experimental runs to determine the importance of the considered features. The basic concepts of design of experiments are explained in the initial sections following which the formulation of the feature selection problem to suit the design of experiment framework is presented. Experiments have been carried out on the Call Centre datasets. To further test the suggested approach, commonly used numerical datasets have also been attempted.

9.1 Introduction

As seen in the previous chapter, filter based approaches assess the merits of the features solely based on the data, ignoring the induction algorithm. Such schemes are generally said to be less capable of producing better results than the wrapper based schemes. (Feldbusch, 2001 and Zhong et al., 2001). The wrapper based schemes, however although capable of producing better results, are rather expensive and difficult to implement. Although efficient search strategies such as forward selection backward elimination can be employed to gain some computational advantage, these methods can take their toll when the number of features is very large, especially in the domain of texts. In view of these shortcomings, a Design of Experiment (DoE) based approach is suggested for the problem of feature subset selection. The DoE based scheme could be viewed as a wrapper based approach in that it makes use of the output from the classifier to decide on a feature subset. However, unlike in the wrapper approach where many successive iterations are carried out in the search for an optimal feature subset, in the DoE based approach only a single set of carefully designed experiments are carried out from which a feature subset is decided. As such, it can be seen that the DoE approach is by no means an attempt to obtain a so-called 'optimal' feature subset; it is not the intention. However, its use is to serve as a comparatively inexpensive scheme for feature subset selection that incorporates classifier information in an attempt to provide for better classification accuracies. In this light, the feature subset chosen by the DoE approach could well be used as a starting point for wrapper based approaches.

A related piece of work to the DoE based feature selection approach, is that proposed by Taguchi and Jugulum(2002) in which the Mahalanobis-Taguchi System(MTS) is presented. MTS is a pattern information technology in which a measurement scale based on all input characteristics is constructed in order to make accurate predictions. An initial set of uniform observation(s) is identified to serve as a reference for the measurement scale, which is then validated. As a further step in the MTS method, which represents the screening stage, useful or significant variables are identified for future pattern analysis. In this context however, orthogonal arrays and signal-to-noise ratios are used to identify these useful variables.

In the next section, a basic appreciation of the DoE based approach is presented. Following that, the formulation of the feature subset selection problem to fit the DoE framework is presented. Experimental results on both numerical and textual datasets out to study the effectiveness of the suggested approach are finally presented together with some discussions.

9.2 Design of Experiments (DoE)

9.2.1 What is it?

Design of Experiments (DoE) is a statistically based structured approach to determine the influence of some input factor/variables on an output/response variable. In the engineering discipline, the value of statistically based experimental designs has been well established (Box et. al, 1978). The DoE approach strongly encourages the use of designed experiments where the input parameters are varied in a carefully structured pattern so as to maximize the information that can be extracted from the resulting experiments. The information that could be ascertained from such experimentation would include: a polynomial model of the input output relationship, the contribution of the factors to the output response as well as the indication of the important factors. In order to better appreciate the DoE approach, an example is provided in the next sub-section.

9.2.2 DoE Process Explained

9.2.2.1 Design Matrix and Models

In order to maximise the information from the experimental runs, various experimental design matrices have been proposed. Table 9.1 shows one such example where 3 factors, with two different discrete settings, are being studied. The entries into the table, 1 and -1 are for standardization purposes. A value of 1 would refer to the higher factor level setting while a value of -1 would refer to a lower factor level setting. For a factor with physical levels of say 10 and 14, they would be given values of –1 and 1 respectively.

Table 9.1: Factor settings of three factors A, B, C

Run No.	A	B	C
1	1	1	1
2	1	1	-1
3	1	-1	1
4	1	-1	-1
5	-1	1	1
6	-1	1	-1
7	-1	-1	1
8	-1	-1	-1

As pointed out earlier, the design settings in Table 9.1 is one of the many possible matrices. The matrix shown in that table is called a *two-level full factorial design*. All possible combinations of the 3 factors are investigated, resulting in a total of 2^3 runs. For model building purposes, such a design allows all possible first order effects and all interaction between these factors to be determined. We can obtain the coefficients of the following model from running the set of experiments as suggested by the design matrix:

$$OUTPUT = \beta_o + \beta_1 A + \beta_2 B + \beta_3 C + \beta_{12} AB + \beta_{13} AC + \beta_{23} BC + \beta_{123} ABC \qquad (9.1)$$

Here β_1 is the effect of factor A, whilst β_{12} is the effect of the interaction between factors A and B. If a different design matrix were used, then the model that could be built would be slightly different. If, for example, we had chosen a three-level full factorial design, there would be enough degrees of freedom for us to determine second order effects as well.

Although the full-factorial experiments can be very informative, it can be very expensive to run especially if the number of factors under consideration is large. For example, even for a 9 factor model, the number of experiments to be run is 2^9 or 512. Such a large number of experimentations might be not practically viable and is usually not required. Hence under such circumstances it is usual to carry out a subset of the full factorial experiments. Such a class of experiments is called the *fractional factorial* experiments. For example, a 2^{3-1} experimental set for the previously considered three factors, would be as shown in Table 9.2.

Table 9.2: A 2^{3-1} fractional factorial for 3 factors

Run No.	**A**	**B**	**C**
1	1	1	1
2	1	-1	-1
3	-1	1	-1
4	-1	-1	1

Since the number of experiments is reduced, the amount of information that could be derived from such experimentation is naturally less. As an example, from the 2^{3-1} experimental set up, the following model could be determined;

$$OUTPUT = \beta_o + \beta_1(A + BC) + \beta_2(B + AC) + \beta_3(C + AB) \qquad (9.2)$$

In the model above, the β_1 coefficient estimates the effect of (*A*+*BC*) and likewise for the other coefficients. The model is written as such because it is impossible to differentiate between the effects of *A* and *BC*, *B* and *AC* and *C* and *AB*. Two or more effects that have this property are called aliases. In this example *A* and *BC* are aliases, *B* and *AC* are aliases and *C* and *AB* are aliases. If however, we assume that factor interactions are unimportant then β_1, β_2 and β_3 estimate the effects of factors A, B and C respectively. As such, in instances where only factor main effects are sought and assuming that the higher order effects and interactions are unimportant the number of experiments to be conducted is only slightly higher than the number of factors investigated. For our textual datasets, we make use of this assumption as would be highlighted later. In the next sub section the determination of values for the coefficients is presented.

9.2.2.2 Determining the Coefficients

Using the example of a full-factorial design, with reference to Table 9.3, the estimate of the effect of factor A, can be computed as follows;

Effect of A $= \bar{y}_{A(1)} - \bar{y}_{A(-1)}$

$$= \frac{12.2 + 12.56 + 12.31 + 11.56}{4} - \frac{11.58 + 12.21 + 13.02 + 12.23}{4}$$

= 12.575 – 12.26 = 0.315

where $\bar{y}_{A(1)}$ and $\bar{y}_{A(-1)}$ is the average of the response values when A is set as 1 and -1 respectively.

Table 9.3: A 2^3 full factorial with response values

Run No.	A	B	C	Response(y)	
1	1	1	1	12.20	
2	1	1	-1	12.56	$y_{A(1)}$
3	1	-1	1	12.31	
4	1	-1	-1	11.56	
5	-1	1	1	11.58	
6	-1	1	-1	12.21	$y_{A(-1)}$
7	-1	-1	1	13.02	
8	-1	-1	-1	12.23	

The effect of factor A reflects the change in the response *y* due to a change in factor A, from a high to a low level setting. The fact that the design matrix columns are orthogonal to one another, allows us to compute the factor effects in this manner. The effects of the other factors can be found in a similar manner. This elegant way of computing factor effects is referred to as analysis of means. Further details of this can be found in Montgomery (1996).

9.3 DoE for Feature Selection

In the previous sections, the DoE approach used for determining factor effects was presented. In this study, this approach has been tailored for the purposes of feature selection. In feature selection, the objective is to determine a subset of features, the smallest subset that results in the classification error being the least. In view of this the problem was modelled under the following conditions:

1. Each feature is considered as a factor.
2. Each factor has 2 levels which correspond to the feature being present or otherwise.
3. The classification accuracy is the response variable.
4. Highly fractionated design matrices that drastically cut down the number of experimental runs are used.
5. The effects of each of the factors are computed and ranked.
6. The final feature subset is the set of features that have a positive factor effect on the classification accuracy.

Most of the aforementioned points, except point 4, are straightforward and easy to comprehend. Hence elaboration is provided for point 4.

Fractionated design matrices as mentioned previously would refer to running only a fraction of the full factorial experiment. This, as mentioned previously would cut down the number of experiments but will limit the potential information that could be obtained. In general practice, however, a trade off is needed. Imagine a problem where there are 500 features to be studied. A full factorial design to be studied would involve the use of 2^{500} experiments, which is an impossible number of experiments to run. Even if we are interested only in the main factor effects and 2 way interaction interactions, the number of experiments required would be 500 + $^{500}C_2$ = 125250 runs. This is still a very large number of experiments. Hence for all practical purposes, it has been assumed that the interaction between factors is negligible. This brings down the number of experiments close to 500. Although this assumption might not be totally correct, running a comparatively lower set of 500 experiments to possibly obtain a relevant subset of features is very appealing. The total number of experiments required via this scheme is equivalent to the number required in the first iteration of wrapper based approaches like Best First Search (Kohavi and John, 1997). The DoE based approach provides an opportunity to handle larger problems. Further, the selected set of features via this approach could serve as initial starting point to some of the search based wrapper approaches. The advantages in using the DoE based approach for feature reduction is outlined below:

1. It allows us to handle a larger original feature set.
2. It provides a ranking of the features.
3. It provides a cut off to determine the feature set since we only choose the features that contribute positively to the classification accuracy.
4. The number of experimental runs is linear to the number of features present.
5. It can be used as an initial algorithm for determining starting point.
6. It can be used to provide an initial understanding of the feature selection problem.

In our study, we focus on the set of 2^k design matrices. It must be pointed out that by using such a set, the number of experiments to be run would be a power of 2. Hence if the problem has 700 features, then 2^{10} experiments would have to be run, in which case there are several extra runs that are made. However, it must be pointed out that there are other sets of design matrices, Hadamard matrices (Montgomery, 1996), that could be used to cut down the number of experimentations. However, the generation of such designs is more involved. Given that this is an exploratory effort, we therefore focus on the 2^k set of matrices.

9.4 Experiments on Textual Datasets

Since the number of experimentations required are greater than those described in the previous chapters, only three of the five datasets were used. Instead of using the entire set of original features, the results from the filter based approaches has been used to determine a reduced initial set of features. From the Markov blanket algorithm (Chapter 8), a reduced set of 500 features results in an accuracy close to 1% less than the original data set. Hence this was used as the starting point.

9.4.1 Experimental Procedure

As before, each dataset was split into 15 randomly stratified test-train examples, with a 70-30 split between train and test sets, respectively. For each training set, a k-fold cross-validation was conducted and the accuracies on the training set were noted.

With 500 features, a highly fractionated, $2^9 = 512$ experimental design matrix was generated. With this set of runs, the factor effects of all of the factors could be determined. For each run of the design matrix, a certain set of features would be present whilst the rest of the features would be absent. The k-fold cross validation accuracy of the training set was used as the response for a particular design run. Hence after the completion of the 512 runs, a matrix similar to the one shown in Table 9.3 would be filled. Based on this matrix, the factors effects of the different factors would be determined and ranked. Only the factors having a positive effect on accuracy would be selected. This set of features would then be used to predict the accuracy of the test set. This is done for each of the 15 trials. The number of folds that were used for the *CDP*, *Call-Type* and *Solid* datasets are 3, 10 and 10, respectively. As in the previous chapter, all experiments were conducted using fixed parameter settings of *c* at 500 and σ at 0.03 using the gaussian kernel, for all datasets except *Esc* for which σ was set to 0.1 whilst *c* was maintained at 500.

9.4.2 Experimental Results

9.4.2.1 Training Set Based Feature Selection

Table 9.4 below shows the averaged accuracies from the 15 trials. The column information are given as follows; column 2 – accuracy of original set of features, column 3 – accuracy of top 500 features selected using Markov Blanket algorithm, column 4 – accuracy after DoE selected set of features, column 5 – DoE's improvement over start point, column 6 – DoE's improvement over full set, column 7 – number of features in original set, column 8 – number of features in final set.

Table 9.4: Accuracy results from DoE approach using training set for feature selection

DataSet	**Full Set**	**Start Point**	**After-DoE**	**Improve Over Start**	**Improve over Full**	**Original Features**	**Final Features**
Call-Type	65.20	64.44	62.67	-1.75	-2.51	1617	285
Solid	60.75	58.91	59.46	0.54	-1.29	1285	262
CDP	81.31	79.49	79.58	0.09	-1.72	2283	271

As can be seen from Table 9.4, for the *Solid* and the *CDP* datasets, there has been a marginal improvement in the average accuracies obtained as compared to the starting point with a reduction of about 50% in the number of features, from a starting value of 500. For the *Call-type* dataset however, there has been a decrease in the accuracy of about 1.75%. All three datasets however, when compared to the full feature showed a decrease in the accuracy.

In order to determine if the improvement for the *Solid* and the *CDP* dataset is statistically significant, hypothesis testing was carried out. The null hypothesis tested was that the difference between the mean DoE accuracy and the mean start point accuracy is less than or equal to 0. The paired t-test was used and p-values of 0.4221 and 0.2231 for the *CDP* and the *Solid* datasets were obtained, which implies that the difference is not statistically different from zero.

A number of reasons could possibly have resulted in the marginal performance of the DoE based approach. Two important possibilities are outlined as follows:

1. The interaction between the factors is rather significant. As such, the effects that have been obtained do not accurately reflect the factor effects.
2. The selection of the optimal feature set was based on the training set accuracy. This might not have reflected the test set accuracy well. Hence a set of features selected, based on the training set accuracy may not be appropriate for the test set.

In order to ascertain possible reasons, further experimentation was required. Determination of the first possibility is rather difficult, since a large number of experiments would need to be conducted to ascertain whether interaction effects are important. Investigating the second possibility was more practical. In this regard, instead of using the training set accuracies in the design matrix, the test set accuracies were used instead. The optimal feature set selection is then based on the test set accuracies and the optimal feature set is thereafter used to obtain the accuracies on the test set which represents the finally achieved accuracies based on the DoE approach. Though, such a scheme for selecting the optimal feature set is biased, the results of this experimentation will provide useful insights into our approach. Hence, a test set based DoE experimentation was conducted.

9.4.2.2 Biased Test Set Based Feature Selection

Table 9.5 shows the averaged results obtained by using the test set for identifying the features. As can be seen from the results all the datasets tested have shown a reasonable improvement in the averaged accuracies. It is worth mentioning that the *Solid* dataset has shown as much as a 6% improvement, with only about a fifth of the number of original features used. A test of hypothesis was also carried out. As before, the null hypothesis tested was that the difference between the mean DoE accuracy and the mean start point accuracy is less than or equal to 0. The paired t-test was used and the p-values are shown in the last column of the table. These results obtained are very encouraging in confirming that the DoE based methodology for feature selection is sound.

Table 9.5: Accuracy results from DoE approach using testing set for feature selection

DataSets	Full Set	Start Point	After-DoE	Improve Over Start	Improve over Full	No. Original Features	Final Features	P-value
Call-Type	65.20	64.44	66.97	2.53	1.77	1617	267	0.0026
Solid	60.75	58.91	67.04	8.13	6.29	1285	268	$<10^{-5}$
CDP	81.31	79.49	83.17	3.68	1.87	2283	269	$<10^{-5}$

Even though this analysis does not ascertain whether important interactions may have been ignored, the fact that such improvements have been achieved suggests that the method is able to correctly identify the important features, at least for the datasets investigated. Ideally if the train and test set data were 100% similar, then computing the important features from either would not have made any difference. It appears that the success of the DoE based approach is quite dependent on being able to find a representative set of training examples. To further test out the methodology, other commonly used numerical datasets were investigated.

9.5 Experiments on Numerical Datasets

9.5.1 Datasets Description

Five datasets were used for evaluating the DoE based feature selection scheme. The following is a brief description of the datasets.

Breast cancer : The task is to predict whether cancer will recur in patients. There are 9 nominal attributes describing characteristics such as tumour size and location.

Splice : The task in this dataset is to recognize two types of splice junctions in DNA sequences; exon/intron (EI) or intron/exon (IE) sites. A splice junction is a site in a DNA sequence at which 'superflous' DNA is removed during protein creation. Intron refers to the portion of the sequence spliced out while exon is the part of the sequence retained.

Image : The task is to classify the centre pixel of a 3x3 patch from an image as belonging to one of 7 categories. The inputs are typical image processing features of the patch.

Heart : The task for this dataset is to predict for the presence/ absence of heart disease. It contains attributes like age, sex, blood pressure and other similar variables.

German : This dataset classifies people described by a set of attributes as good or bad credit risks

Note that all the datasets except for the Australian credit screening (UCI) were downloaded from http://www.first.gmd.de/~raetsch/. [1](Ratsch et al., 1998). The response is binary for all the datasets. As in previous experiments each dataset was split into 15 randomly stratified test/train samples. The feature selection procedure is similar to that in Section *9.4.1*. The only difference here is that, tuning for the *c*, σ parameters of the SVM algorithm was carried out. The range of values for the tuning as well as the number of folds used for each data set, for the training accuracy determination, is shown in Table 9.6.

Table 9.6: Description of numerical datasets investigated

Datsets	**Instances (Train/Test)**	**Features**	**No. of folds**	**Range of c (2^x)**	**Range of Sigma (2^x)**
Breast Cancer (*Bc*)	200/77	9	10	0 to 6	-6 to 0
German (*Ger*)	700/300	20	10	0 to 10	-9 to -1
Image (*Im*)	1300/1010	18	3	0 to 10	-9 to -1
Splice (*Spl*)	1000/2175	60	3	0 to 10	-9 to -1
Heart (*He*)	170/100	13	10	0 to 5	-3 to -8

[1] Gunnar Ratsch has converted non-binary classification problems into a random partition of two classes. Although this step is not necessary for use in our algorithm, this set of data has been used due to its ready availability in the processed format

For all datasets except *Bc* and *He,* steps of 2 was used for the value of x (the last 2 columns of table). For *Bc* and *He* steps of 1 was used. The smaller range of settings for the *Bc* and *He* datasets is due to the use of the parameter values suggested in Rakatomomanjy (2003) as a guide. Generally a 10-fold cross-validation is carried out. For the *Im* and the *Spl* datasets however, due to the larger number of available examples only a 3-fold cross-validation was carried out.

9.5.2 Experimental Results

Table 9.7 shows the results obtained for the various datasets studied.

Table 9.7: Results from DoE approach on numerical datasets

Datasets	**Design Runs**	**Full Set**	**After-DoE**	**Improve**	**% reduction**	**P-value**
Breast Cancer (*Bc*)	16	74.010	72.375	-1.63	44.44	0.9809
German (*Ger*)	32	75.02	75.91	0.88	40.00	0.0284
Image (*Im*)	32	96.46	96.99	0.52	33.33	0.0069
Splice (*Spl*)	64	88.53	90.55	2.01	55.00	$<10^{-5}$
Heart (*He*)	16	83.40	84.13	0.73	30.77	0.1896

As can be seen from the table, the DoE based algorithm seems to perform well in 4 out of the 5 datasets tested. The performance was best for the *Spl* dataset and was worst for the *Bc* dataset. The last column of the table shows the p-value. As before, the null hypothesis tested was that the difference between the mean DoE accuracy and the mean start point accuracy is less than or equal to 0.

It is worth noting that for the *Im* and *Spl* datasets where the number of available examples was larger, the p-value obtained was smaller implying greater confidence that the DoE scheme is comparatively superior. Although the *He* data set produced a greater mean improvement than the *Im* data set, its p-value was smaller. This was because in the *He* data set there were a few trials, which had large improvements that resulted in a higher mean performance. There were a greater number of trials in which the DoE based approach was better than the original for the *Im* dataset as compared to the *He* dataset.

Thus far, the final feature set comprised of features that had a positive contribution to the classification accuracy. However, as mentioned earlier, the approach also provides a set of ranked features. As such it would be interesting to investigate if the removal of features based on this ranked output allows for better accuracies to be obtained.

9.5.3 Feature Removal Based on Rank

Figure 9.1 shows the variation in the averaged test set accuracy over the 15 trials as the remaining features are removed one by one. The starting number of features is the final feature set that has been suggested by the DoE based feature selection approach. The x-axis has been scaled to vary from 0% to100% since the number of starting feature is different for each dataset.

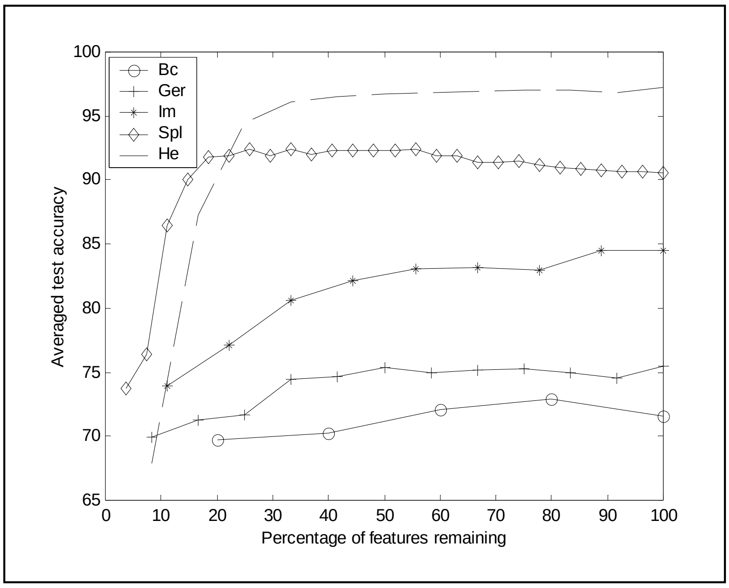


Figure 9.1: Variation of test set accuracies for various datasets

As can been seen from the figure above, on an average, reducing the features based on the ranked list can be useful to some datasets. Most of the datasets with the exception of the *Im* dataset either maintain the accuracy levels (up to a certain number of features) or show a slight improvement, due to the rank-based feature selection.

An important consideration that would need to be made whilst doing rank based feature selection is the optimal number of features to stop at. This is not an easy task to address. Some work that has been done to determine the stop point are outlined in Ambroise and McLachlan (2002). A simple stopping scheme is to select the set of features when the training accuracy reaches its maximum. Using this approach, the test set accuracies were determined. Table 9.8 shows the results obtained in this analysis.

Table 9.8: Results from selecting features based on ranking suggested by DoE approach

Datasets	**Full Set**	**After-DoE**	**After-ranked**	**Improve over DoE**	**% reduction (total)**	**P-value**
Breast Cancer (Bc)	74.010	72.375	72.21	-0.165	66.67	0.5436
German (Ger)	75.02	75.91	75.41	-0.510	65.00	0.8800
Image (Im)	96.46	96.99	96.99	0	50.00	0.5000
Splice (Spl)	88.53	90.55	93.78	3.230	88.33	$<10^{-5}$
Heart (He)	83.40	84.13	83.53	-0.600	53.85	0.7782

As can be seen from the table, the feature selection based on ranking does not seem to produce good results for most except one dataset. The *Spl* dataset shows a vast improvement of more than 3%. The primary reason for the poor performance is due to the inability to identify the stop point accurately. In most of the datasets, except for *Spl*, the peak accuracy point for the training set does not correspond to the peak accuracy point of the test set. As such, as mentioned before, other schemes to determine the stopping point for the ranked based feature selection approach should be explored.

9.6 Summary

In this chapter a novel Design of Experiment based feature selection approach was introduced. This approach is advantageous in that it requires only few runs and can be used on comparatively larger datasets than other wrapper based approaches. The approach was tested on both textual and numerical datasets. For the textual datasets investigated, improvements obtained were found not to be statistically significant. It was found that a possible reason for this could be the fact that the feature selection based on the training set did not suit the test set well. Additional experiments were run to confirm this hypothesis. It was found that when the experiments are carried out using test set accuracy values, significant improvements in the accuracy were obtained. Hence the suggested approach was confirmed to be sound but it was very sensitive to the set of training examples used. For further evaluation, this approach was used on five numerical datasets. In four out of the five datasets, the DoE approach was found to make statistically significant improvements in the classification accuracy with a reduction in the number of features by about 30-50%. The use of ranked features to further reduce the feature set and possibly increase the classification accuracy was also studied. The experiments reveal that greater accuracies can be obtained using the ranked feature list. However, the key question would be identifying the optimal point using the training set accuracies. Various schemes could be attempted to help identify the stopping criteria based on the training set accuracy. On the whole, the DoE approach that has been suggested seems to be promising for the investigated numerical datasets whilst not as encouraging for the call centre data sets. Further experimentation would be needed to fully qualify its use for feature reduction.

CHAPTER 10

CONCLUSION AND FUTURE WORK

This chapter details the text categorization system that has been implemented at the Multi National Company (MNC) that provided the datasets. The advantages of using this system are outlined. Further, the conclusions as well some possibilities for future work is also suggested.

10.1 Implementation of Text Categorization System in a MNC

A text categorisation system was implemented in the Multi-National Company that provided the data. The system was implemented by staff and students of the Design Technology Institute, under the guidance of the author, who also contributed java source codes for the preprocessing portion. In that company, it is estimated that every three out of five working days would need to be spent, by the knowledge engineer to extract the required feedback/information from about 500 records. This hampers the engineer from working on his job of updating the knowledge base using the information in the calls received. In a typical setting, there would be some weeks on which the engineer is not able to analyse these records. As a result there is a backlog. The usual practice then is to sample a portion of the records and present a set of representative statistics. Hence it can be seen that the manual feedback has the following disadvantages:

1) It is not exhaustive
2) Tendency for mistakes is high due to the laborious nature of manual labelling
3) It consumes a lot of resources
4) It results in much slower feedback
5) It is not quick to handle new problems

Sometimes the speed of the feedback in question can be very critical. In fact in the scope of the project there was a particular label within the *Area* field which kept on emerging very frequently. In this instance, it was a particular functional mode that could not be switched off. Since it was in the initial stages of product release and it was a simple enough problem to be solved, this problem was corrected in the next set of records that were released. The automated system that was implemented allowed for this quick feedback.

Further the system also facilitates the detection of new problems. The system automatically identifies new words that have not occurred previously and provides the user with the frequency of occurrence of these new words. The user interactively decides whether these new words could be used to form new classes.

With the automated system implemented the company realised the following advantages:

1) Quick and timely feedback
2) Early detection of new problems
3) Large savings in time and manual effort

4) The capacity to use more texts which were previously unused due to sheer volume

Some of these advantages have been summarized in the statements by a personnel using the implemented system:

"I have been using the data-mining SW for my receiver freetexts, cause that's where the software can really tell the effort and time saved. I could have spend 3 days to go through, all I need now is 1 hour to auto-classify, and another few hours to check through the results. With more than 3 thousands data to train the model, the accuracy level is satisfactory, ranging from 80% to 90% above."

For the various datasets studied, accuracies of about 60% to 80% were achieved. Three datasets had accuracy close to 80% whilst the other two had accuracy of 60% and 65%. Although the accuracies are not too high, with continuous retraining as more examples are added in, the accuracies can be improved. In fact, for the Area dataset, accuracies of up to 90% have been reported with extended use.

10.2 Conclusion

Recent trends in the PDP have brought about challenges such as shorter development times, increasing (technical) product complexity, increasing complexity of the business processes and changing customer expectations/requirements. These challenges have not only rendered the already difficult task of quality and reliability improvement even more difficult but also have made fast feedback within the PDP imperative. Traditional PD tools and methods, albeit being useful, are not completely capable of addressing the current challenges. As a result companies are looking for new tools and techniques to solve their problems.

In this thesis, one such tool, Data Mining, was introduced as a possible solution. An extensive review of Data Mining applications, classified according to the various stages of the PDP, was carried out. This review would serve as a good guide for practitioners interested in applying Data Mining within the PDP. It would also be useful for researchers to identify the missing gaps where more work could be carried out. It was found that there have been numerous DM related applications within the PDP. However, most of these applications were focused on the manufacturing and design phases. More importantly, a very large portion of these applications has focussed on numerical databases. There has been very little work done on textual databases. As these textual databases contain a vast amount of information that has been untapped to a large extent, it was the objective of this thesis to explore the application of Data Mining on such large textual databases to provide rapid feedback, which is extremely crucial to an organisation’s competitiveness.

In order to evaluate DM techniques on real life datasets, textual databases have been sought within the PDP of two MNC’s. These databases were found to contain a rich source of information. However, it was noticed that there is a lot of room for improvement in the quality of information within these databases. Generally these databases have not been filled with the intention of possible use for future data analysis. Further, for some of the investigated databases, the incentives provided for data entry worked against the accurate filling up of the database. Hence given these reasons, avenues for improvement are abundant. The current state of the investigated databases are amenable to the application of data mining. However some efforts could be taken to improve the data quality. Primarily, a general understanding must be created on the importance and the potential use of the data being collected. Futher, incentives for accurate data collection must be present. Finally, careful planning to collect only the the necessary information is also important.

As a particular application of data mining on a textual database, the automated classification of call centre records was studied in detail. Five textual datasets were used. The following conclusions were made:

- The type of pre-processing was not found to be very crucial to these datasets. Marginally better results were obtained from full processing (both stemming and stopword removal) and stemming alone as opposed to only using stopword removal alone or using no processing at all.

- As compared to Support Vector Machines and Decision Trees, the Naïve Bayes algorithm was found to be very sensitive to the type of document representation since the prior probabilities computed may be quite different depending on the document representation type. Further the general behaviour of the SVM and Decision Tree algorithms were very similar to each other and quite different from the Naïve Bayes algorithm. It was found that SVMs performed the best for the investigated datasets.

- The binary representation scheme was found to be the better representation for most of the datasets studied. This could possibly be due to the fact that the sentences in the records are rather short. In such instances, it is common for keywords not to be repeated in a given record and the existence of a particular word or group of words is good enough to determine class information. Also the relatively new tfidf logistic-scaled weighting scheme was found to have reasonably good performance.

- The use of singular value decomposition on the Call Centre datasets did not result in enormous improvements in the classification accuracy. Only a marginal improvement in accuracy was obtained. It was found that the results obtained are quite dependent on the dataset under study. Further, there was no link between the relative change metric (measure of loss of information due to dimension reduction) and the classification accuracy values obtained. As such, it is not possible to decide on the number of dimensions needed to represent the original matrix without having to run the classifier. For the investigated datasets, with the exception of *Solid*, it could be said that the co-occurrence pattern is not very pronounced.

- Amongst the 3 algorithms that were studied in relation to feature reduction (Information Gain, Markov Blanket and Corpus-Based algorithms), it was found that the Markov Blanket algorithm provided the best results. It was the most complicated algorithm of the three, as it considers both redundant as well as irrelevant features. The corpus-based scheme also performed reasonably well. The settings of various similarity thresholds were studied and it was found that these settings did not significantly influence the classification accuracy. The information gain scheme did not perform as well as the other two. In general, it was found that all the datasets could be represented with only about 50% of the original number of features with a reduction of less than 1% in accuracy.

- A novel approach, based on design of experiments, for feature subset selection was introduced. This scheme when used on the call centre datasets did not provide any statistically significant improvement. It was suspected that the training data, on which the feature subset selection was based, was not representing the testing data well enough. As such, feature subset selection based on the testing set accuracy itself was carried out. This approach gave very good results with as much as 9 % improvement in some cases. This reaffirmed the soundness of the suggested approach. To further validate the approach, numerical datasets were studied. 4 out of the 5 datasets showed improvement. In general, the DoE-based approach was found to be promising. However, further experimentation is required to fully qualify its use for feature reduction.

The datasets that have been studied can be said to be quite representative of other call centre records as well. The reason for this is that the call centre records were generated by a third party service provider who services many other companies as well. Each of the call agent goes through a similar training and therefore it is reasonable to assume similar output from them. In general for the five datasets investigated, three of them had accuracy values of 80% whilst the other two had accuracies of 60% and 65%. Although as an initial result these were acceptable, further improvements should be made.

10.3 Future Work

From the research undertaken, many questions and possible extensions have emerged. These are outlined as follows:

1. In chapter 9, it was found that carrying out feature subset selection using the training set accuracies did not produce good results. In order to validate the approach, test set accuracies were then used which produced very promising results. However, in practice, test set accuracies are not available. One way to overcome this problem is to carefully select the training set examples so that they better reflect the test set. A possible approach is outlined in the following steps:

    a. Cluster the test examples into an optimal set of clusters

    b. Determine the centroid or any other appropriate statistic to represent each cluster

    c. Determine the ‘similarity’ between each cluster and all the training examples

    d. Various similarity measures could be attempted

    e. Choose a subset of training examples based on the following criteria

        - The top *k* closest training examples within each class to any of the clusters determined earlier

    The above is only a general idea that might help in improving the earlier obtained results. Although simple in concept, the viability of the approach would depend very much on the detailed implementation. Issues like how to determine an optimal set of clusters, the similarity measure to be used, the value of *k* to be chosen and others would need to be addressed.

2. In any textual classification system, there are bound to be wrongly classified records. For the datasets investigated, about 20% to 40% of the records are

classified wrongly. However, there is no indication amongst the predicted labels as to which of these might have a higher probability of being wrongly predicted. If there was such an indication, then those records with a higher probability of being wrongly classified could be manually checked and labelled thereby increasing the classification accuracy.

3. The task of labelling the examples for training purposes imposes a great burden on the user. Since time is limited, an active learning scheme that reduces the number of examples to be labelled could be employed. This would relieve the user of some effort and in some cases could improve the accuracy obtained. Further, hybrid schemes that consider both feature and example reduction could be studied.

4. A separate study that focuses on the input data collection process and its quality is necessary. Guidelines for collecting useful and valid information would need to be proposed. This will ensure good input into the data mining algorithms and is imperative in obtaining a good output.

REFERENCES

Agrawal, R. and R. Srikant, Fast Algorithms for Mining Association Rules. In Proc. International Conference Very Large Data Bases, Sept. 1994, Santiago, Chile, pp. 487-499.

Ahonen-Myka, H., O. Heinonen, M. Klemettinen and A.I. Verkamo. Finding Co-Occurring Text Phrases by Combining Sequence and Frequent Set Discovery. In Proc. 16th International Joint Conference on Artificial Intelligence IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications, 1999, Stockholm, Sweden, pp. 1-9.

Aliferis, C.F., I. Tsamardinos and A. Statnikov. Large-Scale Feature Selection Using Markov Blanket Induction for the Prediction of Protein-Drug Binding. http://citeseer.nj.nec.com/aliferis02largescale.html. 2003.

Ambroise, C. and G.J. Mclachlan. Selection Bias in Gene Extraction on the Basis of Microarray Gene-Expression Data, Proceedings National Academic Science, *99*(10), pp. 6562-6566. 2002.

Baker, L.D. and A.K. Mccallum. Distributional Clustering of Words for Text Classification. In Proc. 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998, Melbourne, Australia, pp. 96-103.

Basili, R., A. Moschitti and M. Pazienza. A Text Classifier Based on Linguistic Processing. In Proc. IJCAI 99 Workshop on Machine Learning for Information Filtering, 1999.

Bassu, D. and C. Behrens. Distributed LSI: Scalable Concept-Based Information Retrieval with High Semantic Resolution. In Proc. 3rd SIAM International Conference on Data Mining, May 3, 2003, San Francisco, USA.

Bekkerman, R., R. El-Yaniv, N. Tishby and Y. Winter. Distributional Word Clusters Vs. Words for Text Categorization, Journal of Machine Learning Research, *3*, pp. 1183-1208. 2003.

Benkhalifa, M., A. Mouradi and H. Bouyakhf. Integrating External Knowledge to Supplement Training Data in Semi-Supervised Learning for Text Categorization, Information Retrieval, *4*(2), pp. 91-113. 2001.

Berden, T.P.J., A.C. Brombacher and P.C. Sander. The Building Bricks of Product Quality: An Overview of Some Basic Concepts and Principles, International Journal of Production Economics, *67*, pp. 3-15. 2000.

Berry, M.J. and G. Linoff. Data Mining Techniques; for Marketing, Sales and Customer Support. John Wiley and Sons. 1997.

Berry, M.W., S.T. Dumais and G.W. O'brien. Using Linear Algebra for Intelligent Information Retrieval, SIAM Review, *37(4)*, pp. 573-595. 1995.

Blum, A. and P. Langely. Selection of Relevant Features and Examples in Machine Learning, Artificial Intelligence, *97*(1-2), pp. 245-271. 1997.

Bowerman, B.L. and R.T. O'connell. Forecasting and Time Series: An Applied Approach. Wadsworth Incorporation. 1993.

Box, G. E. P., W. G. Hunter and S. J. Hunter. Statistics for Experimenters. John Wiley & Sons, Incorporation. 1978.

Bracht, U. and P. Holtze. Data Mining for Better Parts-Requirement Forecasting, ZWF Zeitschrift fur Wirtschaftlichen Fabrikbetrieb, *94*(3), pp. 119-122. 1999.

Braha, D. and A. Shmilovici. Data Mining for Improving a Cleaning Process in the Semiconductor Industry, IEEE Transactions on Semiconductor Manufacturing, *15*(1), pp. 91-101. 2002.

Braha, D. (ed). Data Mining for Design and Manufacturing: Methods and Applications. Kluwer Academic Publishers. 2001.

Breiman, L. Bagging Predictors, Machine Learning, *24*(2), pp. 123-140. 1996.

Brombacher, A.C. Part II of Betrouwbaarcheid Van Technische Systemen, Anticiperen Op Trends. Stichting Toekomstbeeld der Techniek, Postbus 30424, 2500 GK Den Haag, 200. 2000.

Burges, C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, *2*(2), pp. 121-167. 1998.

Cabena, P., P. Hadjinian, R. Stadler, J. Verhees and A. Zanasi. Data Mining from Concept to Implementation. Prentice Hall. 1997.

Campbell, C. Radial Basis Function Networks: Design and Applications. In An Introduction to Kernel Methods, ed by R. J. Howlett and L. C. Jain, pp. 155-192. Berlin: Physica Verlag. 2000.

Caropreso, M.F., S. Matwin and F. Sebastiani. A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization. In Text Databases and Document Management: Theory and Practice, ed by A. G. Chin, pp. 78–102. Hershey, US: Idea Group Publishing. 2001.

Chang, C.C. and C.J. Lin. LIBSVM 2.5. http://www/csie.ntu.edu.tw/~cjlin/libsvm/. 2003.

Chau, R. and C.-H. Yeh. Multilingual Text Categorization for Global Knowledge Discovery Using Fuzzy Techniques. In Proc. IEEE International Conference on Artificial Intelligence Systems, 2002, pp. 82-86.

Chen, M.-C. Configuration of Cellular Manufacturing Systems Using Association Rule Induction, International Journal of Production Research, *41*(2), pp. 381-395. 2003.

Chen, C.M., N. Stoffel, M. Post, C. Basu, D. Bassu and C. Behrens. Telcordia LSI Engine: Implementation and Scalability Issues. In Proc. 11th Int. Workshop on Research Issues in Data Engineering (RIDE 2001): Document Management for Data Intensive Business and Scientific Applications, April 2001, Heidelberg, Germany, pp. 51-58.

Chi, Z., P.C. Nelson, X. Weimin, T.M. Tirpak and S.A. Lane. An Intelligent Data Mining System for Drop Test Analysis of Electronic Products, IEEE Transactions on Electronics Packaging Manufacturing, *24*(3), pp. 222-231. 2001.

Chu-Carroll, J. and B. Carpenter. Vector-Based Natural Language Call Routing, Association for Computational Linguistics, *25*(3), pp. 361-388. 1999.

Cristianini, N., J. Shawe-Taylor and H. Lodhi. Latent Semantic Kernels. In Proc. 18th International Conference on Machine Learning, 2001, pp. 66-73.

Cristianini, N., H. Lodhi and J. Shawe-Taylor. Latent Semantic Kernels for Feature Selection. Department of Computer Science, Report No. NC-TR-2000-080, University of London. 2000.

Dagli, C.H. and H.-C. Lee. Engineering Smart Data Mining Systems for Internet Aided Design and Manufacturing, International Journal of Smart Engineering System Design, *3*(4), pp. 217-225. 2001.

Damerau, F.J., T. Zhang, S.M. Weiss, and N. Indurkhya. Experiments in High-Dimensional Text Categorization. In Proc. SIGIR Forum (ACM Special Interest Group on Information Retrieval), 2002, pp. 357-358.

Deerwester, S., S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman. Indexing by Latent Semantic Analysis, Journal of the American Society for Information Science, *41*(6), pp. 391-407. 1990.

Dong, A. and A.M. Agogino. Text Analysis for Constructing Design Representations, Artificial Intelligence in Engineering, *11*(2), pp. 65-75. 1997.

Dumais, S.T., J. Platt, D. Heckerman and M. Sahami. Inductive Learning Algorithms and Representations for Text Categorization. In Proc. 7th ACM International Conference on Information and Knowledge Management, 1998, Bethesda, US, pp. 148–155.

Dumais, S.T. LSI Meets Trec: A Status Report. In The First Text Retrieval Conference, ed by D. K. Harman, pp. 137-152. 1993.

Dumais, S.T. Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval. Report No. TM-ARH-017527, Bellcore. 1992.

Elsila, U. and J. Roning. Knowledge Discovery in Steel Industry Measurements, Proceedings Frontiers in Artificial Intelligence & Applications, *78*, pp. 197-206. 2002.

Fayyad, U.M., Piatetsky-Shapiro G. and P, Smyth. From Data Mining to Knowledge Discovery in Databases. AI Magazine, *17*, pp. 37-54. 1996a.

Fayyad, U.M. Data Mining and Knowledge Discovery: Making Sense out of Data, IEEE Expert, *1*(5), pp. 20-25. 1996b.

Feldbusch, F. A Heuristic for Feature Selection for the Classification with Neural Nets. In Proc. Joint 9th IFSA World Congress and 20th NAFIPS International Conference, 2001, pp. 173-178.

Flower, D.R. Databases and Data Mining for Computational Vaccinology, Current Opinion In Drug Discovery & Development, *6*(3), pp. 396-400. 2003.

Foltz, P.W. Using Latent Semantic Indexing for Information Filtering. In Proc. Conference on Office Information Systems, April 1990, pp. 40-47.

Fong, A.C.M. and S.C. Hui. An Intelligent Online Machine Fault Diagnosis System, Computing & Control Engineering Journal, *12*(5), pp. 217-223. 2001.

Forman, G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification, Journal of Machine Learning Research, *3*, pp. 1289-1305. 2003.

Fountain, T., T. Dietterich and B. Sudyka. Mining Ic Test Data to Optimise VLSI Testing. In Proc. Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, Boston, USA, pp. 18-25.

Frawley, W.J., G. Piatetsky-Shapiro and C.J. Matheus. Knowledge Discovery in Databases: An Overview. In Knowledge Discovery in Database, AI Magazine, *13*, pp. 57-70. 1992.

Fukumoto, F. and Y. Suzuki. Using Synonyms and Their Hypernymy Relations of Wordnet for Text Categorization, Transactions of the Information Processing Society of Japan, *43*(6), pp. 1852-1865. 2002.

Furnas, G.W., T.K. Landauer, L.M. Gomez and S.T. Dumais. The Vocabulary Problem in Human-System Communications. Communications of the ACM, *30*, pp. 964-971. 1987.

Gardner, M. and J. Bieker. Data Mining Solves Tough Semiconductor Manufacturing Problems. In Proc. Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, Boston, USA, pp. 376-383.

Gibbons, W.M. and T.M. Scott. Information Technology Trends in Wafer Engineering, Application of Data Mining Techniques to Assist in Process Control. In Proc. 4th UKAIS Conference, Information Systems - The Next Generation, 1999, pp. 307-316.

Giess, M.D., S.J. Culley and A. Shepherd. Informing Design Using Data Mining Methods, ASME Design Engineering Technical Conference, *2,* pp. 207-215. 2002.

Goble, W.M. The Use and Development of Quantitative Reliability and Safety Analysis in New Product Development. Ph.D Thesis, Technische Universitait Eindhoven. 1998.

Graaf, R.D. Assessing Product Development, Visualizing Process and Technology Performance with Race. PHD, Technische Universiteit Eindhoven. 1996.

Grabwoski, H., R.-S. Lossack and J. Weibkopf. Automatic Classification and Creation of Classification Systems Using Methodologies of Knowledge Discovery in Databases (Kdd). In Data Mining for Design and Manufacturing: Methods and Applications, ed by D. Braha, pp. 127-145. Netherlands: Kluwer Academic Publishers. 2001.

Grigori, D., F. Casati, U. Dayal and M.-C. Shan. Improving Business Process Quality through Exception Understanding, Prediction, and Prevention. In Proc. 27th VLDB Conference, 2001, Roma, Italy, pp. 159-168.

Guyon, I. and A. Elissee. An Introduction to Variable and Feature Selection, Journal of Machine Learning Research, *3*, pp. 1157-1182. 2003.

Haffey, M.K.D. and A.H.B. Duffy. Knowledge Discovery and Data Mining within a Design Environment. In Proc. Fourth Workshop on Knowledge Intensive CAD. From Knowledge Intensive CAD to Knowledge Intensive Engineering, 2000, pp. 59-74.

Han, J., R.B. Altman, V. Kumar, H. Mannila and D. Pregibon. Emerging Scientific Applications in Data Mining, Communications of the ACM, *45*(8), pp. 54-58. 2002.

Han, J. and M. Kamber. Data Mining; Concepts and Techniques. Morgan Kaufman. 2000.

Haykin, S. Neural Networks, a Comprehensive Foundation. Prentice Hall. 1999.

Hill, A., S. Song, A. Dong and A. Agogino. Identifying Shared Understanding in Design Using Document Analysis, ASME Design Engineering Technical Conference, *4*, pp. 309-315. 2001.

Hori, S., H. Taki, T. Washio and H. Motoda. Applying Data Mining to a Field Quality Watchdog Task, Electrical Engineering in Japan, *140*(2), pp. 18-25. 2002.

Hsu, C.-W. and C.-J. Lin. A Simple Decomposition Method for Support Vector Machines, Machine Learning, *46*, pp. 291-314. 2002.

Hull, D., J. Pedersen and H. Schuetze. Document Routing as Statistical Classification. In Proc. AAAI Spring Symposium on Machine Learning in Information Access, 1996, Palo Alto, CA.

Ishino, Y. and Y. Jin. (ed). Data Mining for Knowledge Acquisitions in Engineering Design. pp. 145-161, Netherlands: Kluwer Academic Publishers. 2001.

Jacobs, P.S. Joining Statistics with NLP for Text Categorization. In Proc. Third Conference on Applied Natural Language Processing, 1992, pp. 178–185.

Joachims, T. Learning to Classify Text Using Support Vector Machines. pp. 205, Kluwer Academic Publishers. 2002.

Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. LS-8 Report No. 23, Universitait Dortmund. 1997.

John, G.H. and P. Langley. Estimating Continuous Distributions in Bayesian Classifiers. In Proc. Eleventh conference on uncertainty in Artificial Intelligence, 1995, SanMateo, USA, pp. 338-345.

John, G.H., R. Kohavi and K. Pfleger. Irrelevant Features and Subset Selection Problem. In Proc. Eleventh International Conference on Machine Learning, 1994, pp. 121-129.

Kasravi, K. Paint Defect Control Using Data Mining, Technical. In Proc. of the 1998 Manufacturing Conference; Sept 1998, Chicago, USA.

Kim, S.H. and C.M. Lee. Non-linear Prediction of Manufacturing Systems through Explicit and Implicit Data Mining, Computers & Industrial Engineering, *33*(3-4), pp. 461-464. 1997.

Kittler, R. Advanced Statistical Tools for Improving Yield and Reliability. In Proc. of 25th International Symposium for Testing and Failure Analysis, Nov. 1999; Santa Clara, USA, pp. 233-238.

Ko, S.-J. and J.-H. Lee. Feature Selection Using Association Word Mining for Classification. In Proc. of 12th International Conference, Database and Expert Systems Applications, Sept. 2001, Munich, Germany, pp. 211-220.

Kohavi, R. and G.H. John. Wrappers for Feature Subset Selection, Artificial Intelligence, *97*(1-2), pp. 273-324. 1997.

Kohonen, T. Self-organized Formation of Topologically Correct Feature Maps, Biological Cybernatics, *43*, pp.56-59, 1982.

Kolcz, A., V. Prabakarmurthi and J. Kalita. Summarization as Feature Selection for Text Categorization. In Proc. International Conference on Information and Knowledge Management, Nov. 2001, Atlanta, United States, pp. 365-370.

Koller, D. and M. Sahami. Toward Optimal Feature Selection. In Proc. of International Conference on Machine Learning, July 1996, Bari, Italy, pp. 284-292.

Krebel, U. Pairwise Classification and Support Vector Machines. In Advances in Kernel Methods - Support Vector Learning, ed by C. J. C. Burges and A. J. Smola, pp. 255-268. MIT Press. 1999.

Last, M. and A. Kandel, Data Mining for Process and Quality Control in the Semiconductor Industry. In Data Mining for Design and Manufacturing: Methods and Applications, ed by D. Braha, pp. 207-234. Netherlands: Kluwer Academic Publishers. 2001.

Lee, K.H., J. Kay, B.H. Kang and U. Rosebrock. A Comparative Study on Statistical Machine Learning Algorithms and Thresholding Strategies for Automatic Text Categorization. In Proc. 7th Pacific Rim International Conference on Artificial Intelligence, 2002, pp. 444-453.

Lee, T.E., R. Otondo and K. Bonn-Oh. Data Mining for Business Process Reengineering. In Proc. Information Resources Management Association International Conference. Issues and Trends of Information Technology Management in Contemporary Organizations, 2002, pp. 318-322.

Lee, J-H. and Park S-H, Data Mining for High Quality and Quick Response Manufacturing. In Data Mining for Design and Manufacturing: Methods and Applications, ed by D. Braha, pp. 179-206. Netherlands: Kluwer Academic Publishers. 2001.

Leopold, E. and J. Kindermann. Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?, Machine Learning, *46*(1-3), pp. 423-444. 2002.

Letsche, T. and M. Berry. Large-Scale Information Retrieval with Latent Semantic Indexing, Information Science, *100*, pp. 105-137. 1997.

Lewis, D.D. and F. Sebastiani. Report on the Workshop on Operational Text Classification Systems (OTC-01), SIGIR Forum, *35*, pp. 8-11. 2002.

Lewis, D.D. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In Proc. 10th European Conference on Machine Learning, 1998, Heidelberg, Germany, pp. 4-15.

Lewis, D. An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. In Proc. International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, USA, 1992a, pp. 37-50.

Lewis, D.D. Representation and Learning in Information Retrieval. PhD, University of Massachusetts at Amherst. 1992b.

Lewis, D.D. Evaluating Text Categorization. In Proc. Speech and Natural Language Workshop, 1991, San Mateo, USA, pp. 312-318.

Li ,C. J-R. Wen and H. Li. Text Classification Using Stochastic Keyword Generation. In Proc. 20th International Conference on Machine Learning, 2003, Washington DC, USA.

Li, H. and K. Yamanishi. Mining from Open Answers in Questionnaire Data. In Proc. Knowledge Discovery and Data Mining, 2001, San Francisco, CA, USA, pp. 443-449.

Liu, H. and H. Motoda. Feature Selection for Knowledge Discovery and Data Mining. pp. 33-35, Kluwer Academic Publishers. 1998.

Lodhi, H., C. Saunders, J. Shawe-Taylor, N. Cristianini and C. Watkins. Text Classification Using String Kernels, Journal of Machine Learning Research, *2*, pp. 419–444. 2002.

Lovins, J. Development of a Stemming Algorithm, Mechanical Translation and Computational Linguistics, *11*, pp. 22 -31. 1968.

Maki, H. and Y. Teranishi. Development of Automated Data Mining System for Quality Control in Manufacturing. In Proc. of third International Conference, Data Warehousing and Knowledge Discovery, Sept. 2001, Munich, Germany, pp. 93-100.

Mastrangelo, C.M. and J.M. Porter. Data Mining in a Chemical Process Application. In Proc. IEEE International Conference on Systems, Man and Cybernatics 3, 1999, pp. 2917-2922.

Mcdonald, C.J. New Tools for Yield Improvement in Integrated Circuit Manufacturing: Can They Be Applied to Reliability?, Microelectronics Reliability, *39*(6-7), pp. 731-739. 1999.

Menon, R., H.T. Loh, S. Sathiyakeerthi, A.C. Brombacher and C. Leong. The Needs and Benefits of Applying Textual Data Mining within the Product Development Process. Accepted for publication in Quality and Reliability Engineering International.

Mieno, F., T. Sato, Y. Shibuya, K. Odagiri, H. Tsuda and R. Take. Yield Improvement Using Data Mining System, In Proc. IEEE International Symposium on Semiconductor Manufacturing Conference, Oct 1999; Santa Clara, USA pp. 391-394.

Mill, W.C.M.V., A.A.M. Ranke, P.J.M. Verboven, H.L. Hissel and S. Minderhout. Concurrent Engineering Handbook. Phillips CFT Development Support. 1994.

Miralles, F. BPR Based on Data Mining Tools: Redesigning the Sales Promotion Process in Retailing. In Proc. Fifth Americas Conference on Information Systems, 1999, pp. 61-63.

Montgomery, D.C. and G.C. Runger. Applied Statistics and Probability for Engineers. New York: John Wiley and Sons, Inc. 1994.

Montgomery, D.C. Design and Analysis of Experiments. John Wiley and Sons. 1996.

Moore, G.E. Cramming More Components onto Integrated Circuits, Electronics, *38*(8), pp. 114-117. 1965.

Nasukawa, T. and T. Nagano. Text Analysis and Knowledge Mining System, IBM Systems Journal, *40*(4), pp. 967-984. 2001.

Neumann, G. and S. Schmeier. Combining Shallow Text Processing and Machine Learning in Real World Applications. In Proc. IJCAI 99 Workshop on Machine Learning for Information Filtering, 1999, Stockholm, Sweden.

Pahl, G. and W. Beitz. Engineering Design. London: The Design Council. 1984.

Peters, C. and C.H.A. Koster. Uncertainty and Term Selection in Text Categorization, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, *11*(1), pp. 115-137. 2003.

Platt, J.C., N. Cristianini and J. Shawe-Taylor. Large Margin DAGS for Multiclass Classification. In Advances in Neural Information Processing Systems, Vol. 12, ed by S.S. Solla, T.K. Leen and K.R. Mueller, pp. 547-553. MIT Press. 2000.

Porter, M. An Algorithm for Suffix Stripping, Program, *14*(3), pp. 130-137. 1980.

Pyle, D. Data Preparation for Data Mining. pp. 251-258, San Francisco: Morgan Kaufmann Publishers, Incorporated. 1999.

Quinlan, R. C4:5: Programs for Machine Learning. Morgan Kaufmann. 1993.

Ramonwski, C.J. and R. Nagi. A Data Mining-Base Engineering Design Support System: A Research Agenda. In Data Mining for Design and Manufacturing: Methods and Applications, ed by D. Braha, pp. 145-161. Netherlands: Kluwer Academic Publishers. 2001.

Rakotomamonjy, A. Variable Selection Using SVM-based Criteria, Journal of Machine Learning Research : Special Issue on Variable and Feature Selection, pp. 1357-1370. 2003.

Ratsch, G., T. Onoda and K.-R. Muller. Soft Margins for Adaboost. NeuroCOLT2 Technical Report No. NC-TR-1998-021, 1998.

Rogati, M. and Y. Yang. High-Performing Feature Selection for Text Classification. In Proc. International Conference on Information and Knowledge Management, 2002, pp. 659-661.

Ross, J. Taguchi Techniques for Quality Engineering. Singapore: McGraw-Hill. 1996.

Ross, Q. C4.5: Program for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers. 1993.

Rudolph, S. and P. Hertkorn. Data Mining in Scientific Data. In Data Mining for Design and Manufacturing: Methods and Applications, ed by D. Braha, pp. 61-87. Netherlands: Kluwer Academic Publishers. 2001.

Salton, G. and C. Buckley. Term Weighting Approaches in Automatic Text Retrieval, Information Processing and Management, *24*(5), pp. 513-523. 1988.

Scholkopf, B., C. Burges and V. Vapnik. Extracting Support Data for a Given Task. In Proc. First International Conference on Knowledge Discovery and Data Mining, 1995, Menlo Park, CA, pp. 252-257.

Schutze, H. and J.O. Pederson. A Concurrence-Based Thesaurus and Two Applications to Information Retrieval, Information Processing and Management, *33*(3), pp. 307-318. 1997.

Schwabacher, M., T. Ellman and H. Hirsh. Learning to Set up Numerical Optimizations of Engineering Designs. In Data Mining for Design and Manufacturing: Methods and Applications, ed by D. Braha, pp. 87-127. Netherlands: Kluwer Academic Publishers. 2001.

Scott, S. and S. Matwin. Feature Engineering for Text Classification. In Proc. 16th International Conference on Machine Learning, 1999, pp. 379-388.

Sebastiani, F. Machine Learning in Automated Text Categorization, ACM Computing Surveys, *34*(1), pp. 2002.

Skormin, V.A., V.I. Gorodetski and L.J. Popyack. Data Mining Technology for Failure Prognostic of Avionics, IEEE Transactions on Aerospace and Electronic Systems, *38*(2), pp. 388-403. 2002.

Taguchi, G. and R. Jugulum, The Mahalanobis Taguchi Strategy: A Pattern Technology System. New York: Wiley. 2002.

Taira, H. and M. Haruno. Text Categorization Using a Transductive Boosting Method, Transactions of the Information Processing Society of Japan, *43*(6), pp. 1843-1851. 2002.

Tan, C.-M., Y.-F. Wang and C.-D. Lee. The Use of Bigrams to Enhance Text Categorization, Information Processing & Management, *38*(4), pp. 529-546. 2002.

Tan, P.-N., H. Blau, S. Harp and R Goldman. Textual Data Mining of Service Centre Call Records. In Proc. The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, Boston, MA, USA, pp. 417-423.

Tseng, M.M. and X.H. Du. Design by Customers for Mass Customisation Products, Annals of the CIRP, *47*(1), pp. 103–106. 1998.

Tsuda, H., H. Shirai, O. Takagi and R. Take. Yield Analysis and Improvement by Reducing Manufacturing Fluctuation Noise. In Proc. Ninth International Symposium on Semiconductor Manufacturing, Sept. 2000; Tokyo, Japan, pp. 249-252.

Ulrich, K.T. and S.D. Eppinger. Product Design Development. McGraw-Hill. 2000.

Van Rijsbergen, C. Information Retrieval. Butter Worths. 1979.

Vapnik, V. The Nature of Statistical Learning Theory. New York: Springer. 1998.

Wang Guo, R., G. Yu, B. Bao Yu and J. Lu Hong. Managing Very Large Document Collections Using Semantics, Journal of Computer Science and Technology (English Language Edition), *18*(3), pp. 403-406. 2003.

Weisberg, S. Applied Linear Regression. John Wiley and Sons. 1985.

Weiss, S.M., C. Apt´E, F.J. Damerau, D.E. Johnson, F.J. Oles, T. Goetz and T. Hampp. Maximizing Text-Mining Performance, IEEE Intelligent Systems, *14*(4), pp. 63–69. 1999.

Weston, J. and C. Watkins. Multi-class support vector machines. Department of Computer Science, Report No. CSD-TR-98-04, University of London. 1998.

Weka. Software. http://www.cs.waikato.ac.nz/~ml/weka/. 2003.

Wilbur, W.J. and K. Sirotkin. The Automatic Identification of Stopwords, Journal of Information Science, *18*, pp. 45-55. 1992.

Witten, H.W. and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers. 2000.

Wood, W.H., Yang, M.C. and Cutkosky, M.R. Design Information Retrieval:Improving Access to the Informal Side of Design. In Proc. ASME DETC Design Theory and Methodology Conference, 1998.

Xing, E.P., M.I. Jordan and R.M. Karp. Feature Selection for High-Dimensional Genomic Microarray Data. In Proc. Eighteenth International Conference on Machine Learning, 2001, Massachusetts, US, pp. 601-608.

Xu, Y-Y., X-Z. Zhou and Z-W. Guo. Weak Learning Algorithm for Multi-Label Multiclass Text Categorization. In Proc. International Conference on Machine Learning and Cybernetics 2, 2002, pp. 890-894.

Yan, W., C.H. Chen and L.P. Khoo. A Radial Basis Function Neural Network Multicultural Factors Evaluation Engine for Product Concept Development, Expert Systems, *18*(5), pp. 219-232. 2001.

Yang, M.C., W.H. Wood and M.R. Cutkosky. Data Mining for Thesaurus Generation in Informal Design Information Retrieval. In Proc. Congress on Computing in Civil Engineering, 1998, Boston, MA, USA, pp. 189-200.

Yang, S.M., X.-B. Wu, Z.-H. Deng, M. Zhang and Yang, D.-Q. Relative Term-Frequency Based Feature Selection for Text Categorization. In Proc. International Conference on Machine Learning and Cybernetics, 2002, pp. 1432-1436.

Yang, Y. An Evaluation of Statistical Approaches to Text Categorization. Information Retrieval, *1*(1-2), pp. 69-90. 1999.

Yang, Y. and X. Liu. A Re-Examination of Text Categorization Methods. In Proc. 22nd International Conference on Research and Development in Information Retrieval, SIGIR, 1999, pp. 42-49.

Yang, Y. and J.O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In Proc. 14th International Conference on Machine Learning., 1998, San Francisco, US, pp. 412-420.

Yang, Y. and J. Wilbur. Using Corpus Statistics to Remove Redundant Words in Text Categorization, Journal of the American Society of Information Science, *45*(5), pp. 357-369. 1996.

Yang, Y. Noise Reduction in a Statistical Approach to Text Categorisation. In Proc. 18th ACM International Conference on Research and Development in Information Retrieval, 1995, pp. 256-263.

Yang, Y. and C.G. Chute. An Example-Based Mapping Method for Text Categorization and Retrieval, ACM Transactions on Information Systems, *12*(3), pp. 252-277. 1994.

Zelikovitz, S. and H. Hirsh. Using LSI for Text Classification in the Presence of Background Text. In Proc. 10th {ACM} International Conference on Information and Knowledge Management, 2001, pp. 113-118.

Zhong, N., J. Dong, and S. Ohsuga. Using Rough Sets with Heuristics for Feature Selection, Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies, *16*(3), pp. 199-214. 2001.

APPENDIX A

Market Need Identification: In this phase, the needs and preferences of the customers are identified. If wrongly identified this would have dire consequences on the success of the product. Careful planning would need to be carried out to capture the 'voice of the customer'.

Planning: This phase involves an iterative and systematic search for and the selection and development of promising product ideas out of the market requirements. The input for this phase is the Market, Company and Other Sources such as economic and political changes, new technologies and etc. It also involves the careful planning of the work, in terms of long-term and short-term goals, while defining the tasks as fully and clearly as possible. The output from this phase is a requirement list, which describes the wished product specifications, which are translated out of the market requirements.

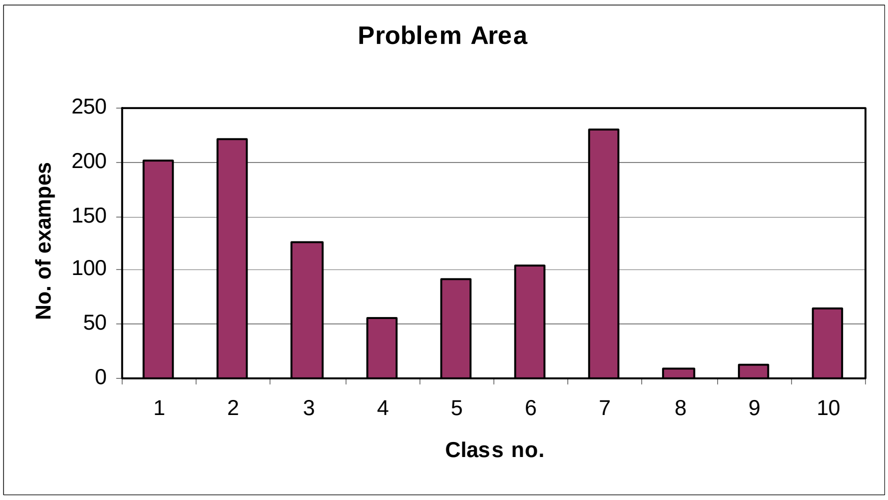
Design : The input to this phase is the requirements list of the former step. This phase initially involves the search for suitable working principles and then combination of those principles in a working structure. Based on this structure preliminary layouts to obtain more information about the advantages and disadvantages of the different variants are attempted. A definitive layout is then decided that provides a check of function, strength, spatial compatibility etc. In the final stage of this phase, the arrangements, forms, dimensions and surface properties of all the individual parts are finally laid down, the materials specified, production possibilities assessed, cost estimated and all the drawings and other production, assembly and transport documents produced. The specification of production, assembly and transportation then serve as output to this phase.

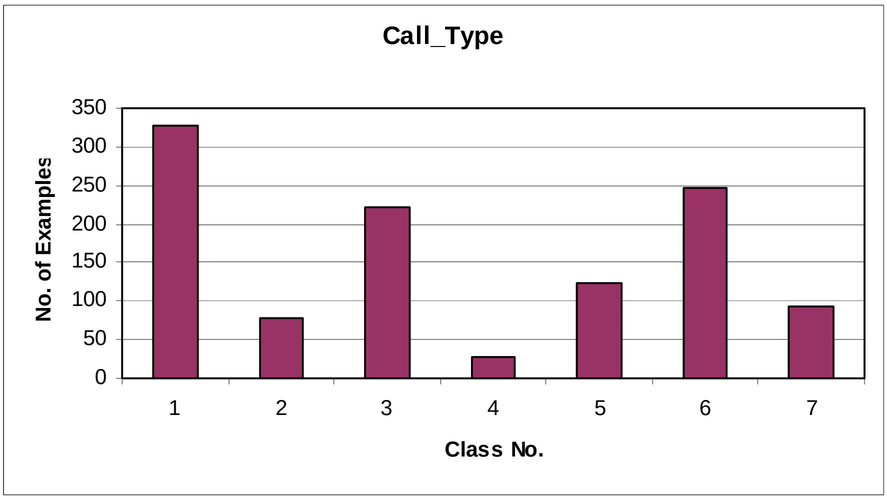
Testing and refinement: This step involves the construction and evaluation of multiple pre-production versions of the product. The pieces previously designed and implemented are put together. If applicable, software modules are tested with each other to make sure that outputs generated by one module match the inputs needed by another. When the designers are convinced that the product is complete and fully operational, a product is delivered to a formal test group that independently verifies product functionality and robustness.

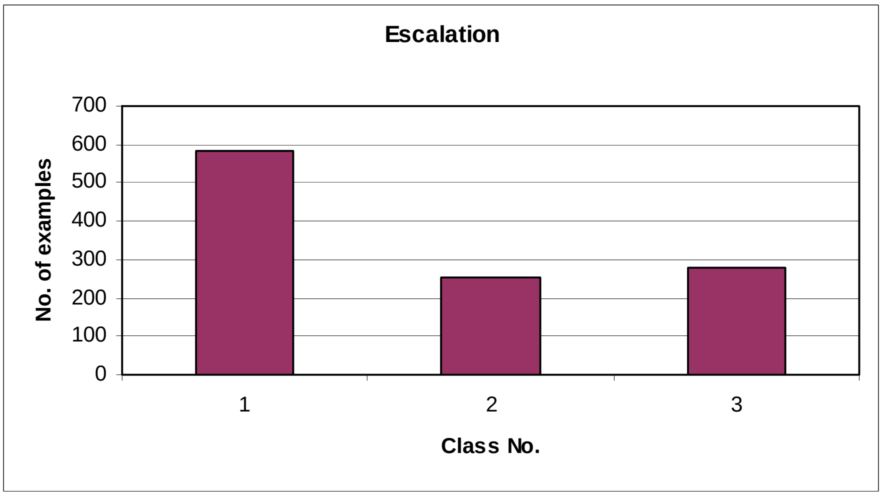
Production ramp-up: The product is made using the intended production system. The purpose is to train the work force and to work out any remaining problems in the production processes. The output is the transition to gradual ongoing production. At some point in this transition the product is launched.

Service and Support : This phase refers to the after-sales phase. In this phase, vital information of the filed failure of the product can be gathered. This information if quickly analyzed could be fed back into the PDP so that action could be taken on going production of the current product. Otherwise, improvements could be made on the future models.

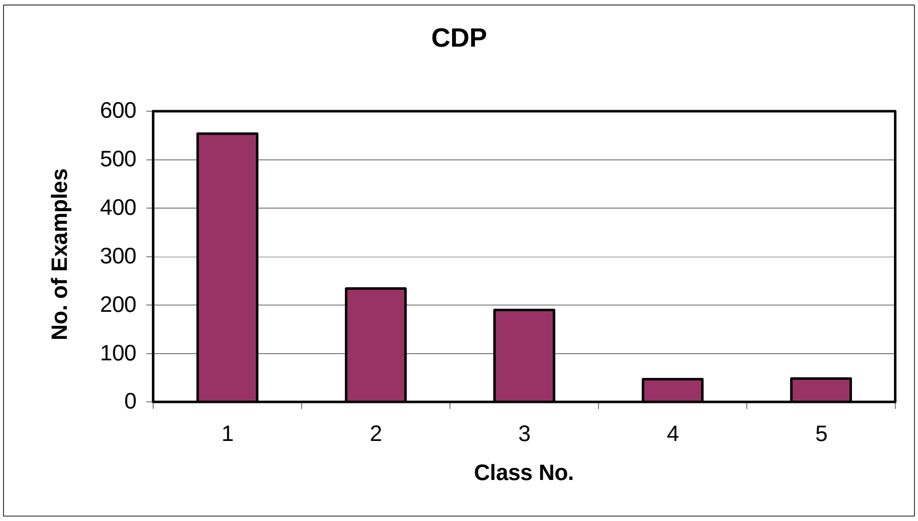
APPENDIX B

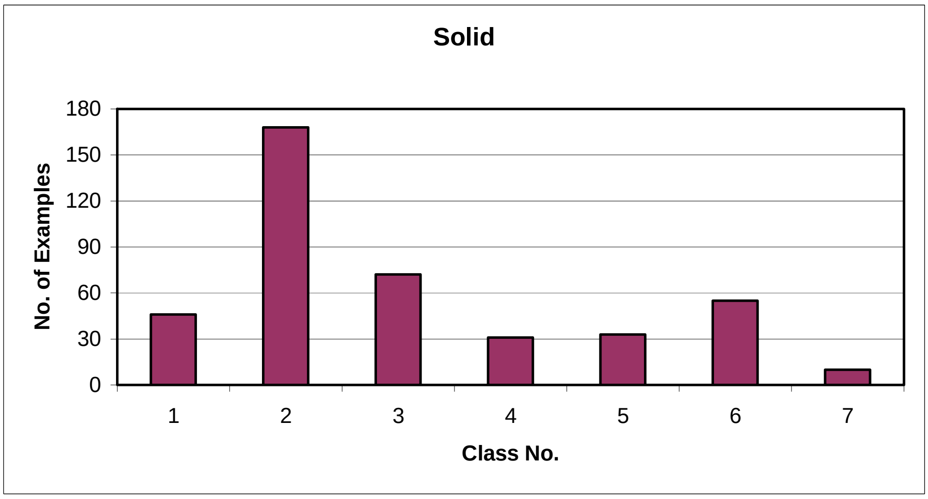

Problem Area
No. of exampes
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
10
Class no.


Call_Type
No. of Examples
350
300
250
200
150
100
50
0
1
2
3
4
5
6
7
Class No.


Escalation
No. of examples
700
600
500
400
300
200
100
0
1
2
3
Class No.

APPENDIX C


CDP
No. of Examples
0
100
200
300
400
500
600
1
2
3
4
5
Class No.


Solid
No. of Examples
0
30
60
90
120
150
180
1
2
3
4
5
6
7
Class No.

APPENDIX D

SOLID Dataset

Trial	Binary	Tf (n)	Tfidf	Tfidf-(ln)	Tfidf-(ls)	Entropy
1	65.32	64.52	61.29	60.48	65.32	62.10
2	62.90	65.32	58.87	54.84	62.10	59.68
3	64.52	66.13	56.45	60.48	66.13	60.48
4	66.94	63.71	62.10	57.26	65.32	61.29
5	59.68	62.10	51.61	60.48	64.52	54.03
Average	63.87	64.35	58.06	58.71	64.68	59.52
Std.	2.76	1.55	4.23	2.58	1.55	3.20

CDP Dataset

Trial	Binary	Tf (n)	Tfidf	Tfidf-(ln)	Tfidf-(ls)	Entropy
1	82.55	80.06	70.41	80.37	80.37	76.95
2	79.13	79.75	77.57	81.93	78.19	81.00
3	76.95	78.50	67.91	77.88	76.64	78.82
4	80.69	76.32	79.13	80.69	82.87	78.50
5	79.13	80.69	81.31	79.75	76.01	81.62
Average	79.69	79.07	75.26	80.12	78.82	79.38
Std.	2.08	1.73	5.80	1.48	2.82	1.91

AREA Dataset for Free Format

Trial	Binary	Tf (n)	Tfidf	Tfidf-(ln)	Tfidf-(ls)	Entropy
1	78.491	70.943	76.981	80.377	78.113	73.585
2	81.509	75.094	81.509	81.509	81.509	83.774
3	78.491	65.283	74.340	76.981	79.245	75.472
4	76.604	68.679	75.472	75.472	77.359	75.849
5	78.491	70.189	76.226	75.094	76.604	76.981
Average	78.717	70.038	76.906	77.887	78.566	77.132
Std.	1.762	3.566	2.752	2.906	1.913	3.909

CALL_TYPE Dataset for KB Format

Trial	Binary	Tf (n)	Tfidf	Tfidf-(ln)	Tfidf-(ls)	Entropy
1	66.197	57.747	63.380	71.831	66.197	69.014
2	64.789	59.155	60.563	71.831	66.197	60.563
3	63.380	60.563	66.197	70.423	69.014	64.789
4	70.423	66.197	63.380	69.014	70.423	69.014
5	67.606	56.338	57.747	61.972	66.197	63.380
Average	66.479	60.000	62.254	69.014	67.606	65.352
Std.	2.709	3.805	3.212	4.106	1.992	3.673

ESCALATE Dataset for Full Format

Trial	Binary	Tf (n)	Tfidf	Tfidf-(ln)	Tfidf-(ls)	Entropy
1	85.075	82.985	80.597	78.806	85.075	83.8806
2	78.508	80.000	79.702	77.612	79.105	78.5075
3	74.627	75.821	75.821	77.015	77.910	74.0299
4	83.582	79.403	80.597	76.716	83.582	80.8955
5	82.687	80.000	78.508	78.209	82.090	77.6119
Average	80.896	79.642	79.045	77.672	81.552	78.985
Std.	4.269	2.554	1.996	0.855	3.003	3.684

AREA Dataset for KB Format

Trial	Binary	Tf (n)	Tfidf	Tfidf-(ln)	Tfidf-(ls)	Entropy
1	77.778	66.667	73.611	79.167	73.611	75.000
2	68.056	62.500	66.667	76.389	68.056	68.056
3	75.000	66.667	73.611	77.778	76.389	73.611
4	72.222	69.444	70.833	80.556	75.000	73.611
5	77.778	62.500	72.222	76.389	70.833	75.000
Average	74.167	65.556	71.389	78.056	72.778	73.056
Std.	4.120	3.011	2.880	1.811	3.345	2.880

CURRICULUM VITAE

Rakesh Menon graduated with a degree in Civil Engineering in 1995 before pursuing a Master’s in the same discipline at the National University of Singapore. He started work as a Research engineer at the Centre for Robust Design where he embarked on his Doctoral degree, in the year 1998, on a part-time basis. He then continued his work at the Design Technological Institute. He has worked on projects in the areas of design of experiments, robust design, reliability, data as well as text mining. He was also part of a team that won the prestigious international Data Mining competition – Knowledge Discovery in Databases in the year 2002. He is currently employed with a Business Intelligence Software vendor, SAS, where he serves as the Principal for Data Mining. He also has a Masters Degree in Financial Engineering, from the National University of Singapore.