FEATURE SELECTION BASED ON SEQUENTIAL ORTHOGONAL SEARCH STRATEGY

By

Azlyna Senawi

A thesis submitted to the University of Sheffield

for the degree of

Doctor of Philosophy

Department of Automatic Control & Systems Engineering

The University of Sheffield

Mappin Street

Sheffield S1 3JD

United Kingdom

November 2018

To Nafrizuan, my amazing husband,

and the apples of my eyes; Ismael, Ielyas and Iedris.

To the memory of Abah,

With love

Abstract

This thesis introduces three new feature selection methods based on sequential orthogonal search strategy that addresses three different contexts of feature selection problem being considered. The first method is a supervised feature selection called the maximum relevance–minimum multicollinearity (MRmMC), which can overcome some shortcomings associated with existing methods that apply the same form of feature selection criterion, especially those that are based on mutual information. In the proposed method, relevant features are measured by correlation characteristics based on conditional variance while redundancy elimination is achieved according to multiple correlation assessment using an orthogonal projection scheme. The second method is an unsupervised feature selection based on Locality Preserving Projection (LPP), which is incorporated in a sequential orthogonal search (SOS) strategy. Locality preserving criterion has been proved a successful measure to evaluate feature importance in many feature selection methods but most of which ignore feature correlation and this means these methods ignore redundant features. This problem has motivated the introduction of the second method that evaluates feature importance jointly rather than individually. In the method, the first LPP component which contains the information of local largest structure (LLS) is utilized as a reference variable to guide the search for significant features. This method is referred to as sequential orthogonal search for local largest structure (SOS-LLS). The third method is also an unsupervised feature selection with essentially the same SOS strategy but it is specifically designed to be robust on noisy data. As limited work has been reported concerning feature selection in the presence of attribute noise, the third method is thus attempts to make an effort towards this scarcity by further exploring the second proposed method. The third method is designed to deal with attribute noise in the search for significant features, and kernel pre-images (KPI) based on kernel PCA are used in the third method to replace the role of the first LPP component as the reference variable used in the second method. This feature selection scheme is referred to as sequential orthogonal search for kernel pre-images (SOS-KPI) method. The performance of these three feature selection methods are demonstrated based on some comprehensive analysis on public real datasets of different characteristics and comparative studies with a number of state-of-the-art methods. Results show that each of the proposed methods has the capacity to select more efficient feature subsets than the other feature selection methods in the comparative studies.

Acknowledgement

In the name of God, the Most Gracious, the Most Merciful.

First and foremost, all praises and thanks are due to the Almighty Allah, the Lord of the Universe, who generously gave me the strength, knowledge and opportunity to complete this PhD journey. No word could adequately describe my upmost gratitude for His innumerable favours showered upon me along the way.

I am greatly indebted to my supervisor, Dr. Hua Liang Wei, whose professional guidance, thoughtful consideration and steady support over the years have been invaluable. Without his involvement, intellectual advice and critical comments, this thesis would not have been possible. Thanks to him for being receptive to my ideas and I consider the granted scientific freedom during the course of my study as a positive learning experience.

I am also indebted to Professor Billings as my second supervisor, who took time out of his busy schedule to give constructive feedbacks and helpful suggestions for the research work.

I would like to gratefully acknowledge Malaysian Government and Universiti Malaysia Pahang for the scholarship award under *Skim Latihan Akademik IPTA* (SLAI) program that kept me financially sound throughout a three-and-a-half year study period.

A heartfelt thanks to all my friends who made my Sheffield experience memorable and special, in particular, Vicktor, Ain, Nana, Ruzaini, Abang Zack, Kak Niza, Maniha, Hyreil and Zhang Yang. Personally, I would like to thank Kak Anoi and Abang Zam for their kind helps and countless delicious meals delivered to our doorstep at Basford Street.

A bunch of thanks is reserved to my friends, Fizah, Rozieana and Najihah, not only for helping me in many ways but also for lending an ear for my sad stories since I came back to Malaysia and need to finish my PhD from far.

My sincere thanks and deepest appreciation go to my late dad (Abah) and mum for their emotional support and unwavering faith in me although they don't understand what I researched on. Abah, I never realized how much I love you till you left me during the final stage of my thesis write-up. I really wish you were around to witness your dream come true. My sincere thanks are extended to my parents in law for their love, care and prayers through all these years.

My greatest gratitude to my dearest husband, Nafrizuan, for making things keep going on even at the hardest of times. During the last six months of writing this thesis I was particularly preoccupied but he always try to make the process as easy as possible for me. I cannot express how much thankful I am for his unremitting patience and support from the moment we began the PhD journey together at ACSE. Certainly, this thesis would not have been in its present form without him.

The last word goes for my three little sons: Ismael, Ielyas and Iedris, to whom I owe lots of fun hours. They have been the light of my life and always been my constant source of strength to get this roller coaster journey through to the end.

Thank you.

Table of Contents

List of Abbreviations

CART	Classification and regression trees
FOS-MOD	Forward orthogonal search by maximizing the overall dependency
k-NN	*k*-nearest neighbour
LDA	Linear discriminant analysis
LPP	Locality preserving projection
LS	Laplacian score
MCFS	Multi-cluster feature selection
MIFS	Mutual information based feature selection
MMLS	Minimum-maximum Laplacian score
MRmMC	Maximum relevance-minimum multicollinearity
mRMR	Minimal-redundancy-maximal-relevance
PCA	Principal component analysis
SOS-KPI	Sequential orthogonal search of kernel pre-images
SOS-LLS	Sequential orthogonal search for local largest structure
SVM	Support vector machine

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

This chapter presents an overview of the research conducted. It starts with a discussion of the motivation of the research to highlight the research problems. Then, objectives of this research are established. This chapter also allocates a section to preview the contribution of the research to the world of knowledge. The publications as outcomes of the research also have been listed. This chapter ends with a description of the overall thesis organisation.

1.2 Motivation

The birth of the Industrial Revolution has brought forward technological advances to the world and has thus motivated the industrial productivity to growth vividly (Gerbert, et al., 2015). As the world is currently moving towards the fourth wave of technological advancement, digital industrial technology has become the main essence and attracted considerable attention in recent years from numerous parties including policy-makers, practitioners, research communities as well as government organisations. This era is penned as Industrial Revolution 4.0 (Gerbert, et al., 2015) which is often simply noted as Industry 4.0. The route to the Industry 4.0 has been focused on nine foundational technology advances, more specifically referred as Industry 4.0 Technology Pillars as shown in Figure 1.1.

Notice that one of the pillars is “big data and analytics”. The two key components, big data and analytic, that become the basis for the pillar are really two different things but they are intertwined. As the two teamed up and worked together, they then brought a new discipline known as big data analytics that is increasingly becoming a trending practice by many organisations with a primary goal to gain useful information from big data (Sivarajah, et al., 2017). The potential of big data is evident from the fact that among the highest paid jobs in the world are related to big data (Bennett, 2017). According to a Glassdoor report, data scientist career was ranked as the number one best job in the United States for 2018, meanwhile it is the sixth best job in the UK in 2017 (Glassdoor Inc., 2018). Because there are still enormous sets of untapped big data in the industrial world, they thus offer valuable information with many new opportunities that are beneficial in aiding practitioners to have sound understanding about certain activities or processes. According to O'Donovan et al. (2015), big data analytics will provide significant help to the industrial community to optimize the quality of a production, perform better operations, acquire excellent services and most importantly support accurate and timely decision-making.

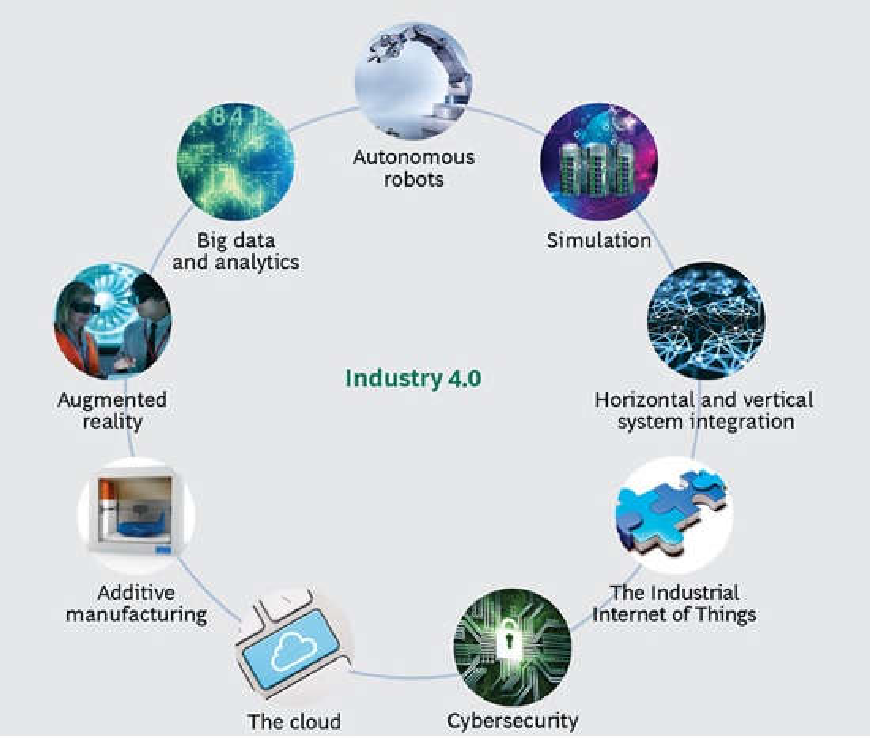

Figure 1.1: Nine Technological Pillars of Industrial Revolution 4.0 (Gerbert, et al., 2015).

Big data stored in any information system including but not limited to industrial related databases require special methods for processing and analysing before the data can be used to assist decision making. Under such a circumstance, there is a demand to automate the process intelligently and use each massive dataset as a source to extract useful information (Bhadani & Jothimani, 2016). Among the tools that can be utilized to meet the requirement is data mining.

Data mining can be defined as a process of discovering useful patterns from a large amount of data (Witten & Frank, 2005). It has been applied successfully in many different fields such as retail industry, marketing, banking, healthcare, science and engineering.

However, mining scientific data is often different from mining business or commercial data. According to Sivarajah et al. (2017), data analysis problems for science and engineering fields are more complex and therefore require more specific solutions. Hence, special attention must be given to the unique requirements of scientific datasets and related issues need to be addressed accordingly.

One of the most complex natures that receive considerable attention among researchers is the explosive growth in sizes of datasets with millions to billions of records. Remarkable innovation and advancement in data storage have made collecting and saving such tremendous amount of data more feasible. While massive datasets can be utilized as a source to mine interesting information, the analysis accuracy and efficiency could become intractable due to the high dimensionality. Although there are methods that can be used to construct predictive models from high dimensional data with high accuracy (Breiman, 2001), data analysis in lower dimensional space is still desirable in many applications since modelling high dimensional data is more likely too computationally expensive. In many applications, the analysis of big data can be performed in a reduced dimensional space and the resulting performance can be even better than that obtained from using the original datasets (Zhang, et al., 2009; Wang, et al., 2012; Likitjarernkul, et al., 2017) because the original feature space may contain a large number of irrelevant and redundant features. Hence, it is desirable and sometimes crucial to identify and remove these insignificant features so that learning from data become technically more effective. This can be done via dimensionality reduction which can be achieved by two different strategies, namely, feature extraction and feature selection.

In feature extraction approaches such as principal component analysis (Wold, et al., 1987) and linear discriminant analysis (Balakrishnama & Ganapathiraju, 1998), new features are constructed from the original features to form a new reduced dimensional space by combining or transforming the original features using some functional mapping. Although the new features in the new reduced dimensional space are related to the original features, the actual interpretation of the original features and hence the relation to the original system variables is completely lost in most cases. This drawback should be taken into account when considering dimensionality reduction since the actual interpretation may be important to understand the learning process that generates the new feature space (Somol, 2010). Feature extraction also often associated with computational inefficiency despite the fact that it may significantly reduce dimensional space since the new constructed features are based on transformation that involves all original features including irrelevant and redundant features. Nevertheless, its main advantage over feature selection is in the fact that no information from the original features is wasted or lost in the dimensionality reduction process (Yang, et al., 2010). This fact further offers another advantage of feature extraction approach in that the reduced dimensional space, in general, have more compact representation of the original features than the feature selection approach (Gao, et al., 2017).

Unlike feature extraction which attempts to create new features based on all original features, feature selection is an approach which requires a selection of the most significant subset of features to a targeted concept by removing redundant and irrelevant features (Wei & Billings, 2007). These redundant and irrelevant features can be ignored because they give very little or no unique information for data analysis and modelling (Hira & Gilles, 2015). Moreover, in many cases, the presence of irrelevant and redundant features can only make data analysis and modelling more complicated without increasing accuracy. Since feature selection does not alter the actual interpretation of any feature involve, it has the advantage of being able to facilitate the understanding of what really generates the new feature space and significantly benefit future analysis. Commonly used feature selection methods include Fisher score (Jaakkola & Haussler, 1999), Relief (Kira & Rendell, 1992), minimal-redundancy-maximal-relevance (mRMR) (Peng, et al., 2005) and Laplacian score (He, et al., 2006), to name a few. In contrast to feature extraction, the feature selection approachis perceived as having a lower flexibility in finding a reduced feature space, particularly when the best low-dimensional feature set for a certain data mining task should not only consists of original features (Zhang, et al., 2008).

Much of the early work on feature selection focused on choosing relevant features. But later, when the existence and effect of redundant features have been discovered, many have been directed to deal with both relevant and redundant features in the selection process. Feature redundancy was defined in some explicit or inexplicit manner, highlighting the need to remove redundant features (John, Kohavi, & Pfleger, 1994; Pudil, Novovicova, & Kittler, 1994; Koller & Sahami, 1996; Kohavi & John, 1997; Hall, 1999). For example, in Koller & Sahami (1996), the Markov Blanket filtering process was utilized to form the definition which highlights that a redundant feature removed earlier remains redundant when other features are removed. A more concrete definition of feature redundancy was given in Yu & Liu (2004), which considers an optimal feature subset is the one that essentially contains all strongly relevant features and also weakly relevant but non-redundant features.

The concept of mutual information has been widely employed as an evaluation criterion for choosing a set of relevant and non-redundant features. Some of the most prominent examples include the criteria proposed by Battiti (1994), Kwak & Choi (2002a) and Peng et al. (2005). Mutual information is preferable as an evaluation criterion over the correlation function for many proposed feature selection methods because of its ability to measure arbitrary dependence relationships between two features (Li, 1990; Battiti, 1994). The method is not only limited to numerical features, but also applies to symbolic features consisting of discrete categories (Li, 1990). These two advantages made the mutual information based criterion to be seen as a more universal and robust measure.

Despite the aforementioned advantages, the mutual information criterion also has a few notable drawbacks. Mutual information computation is straightforward for discrete (categorical) random variables where an exact solution can be obtained easily. However, for continuous random variables which are frequently encountered in mutual information computations, it is difficult to gain the exact solution since the computation of the exact probability density functions (pdfs) is impossible (Kwak & Choi, Input feature selection for classification problems, 2002a). Hence, an estimation of the mutual information is required and different methods can be employed. Among the possible methods are histogram-based (Moddemeijer, 1989; Haeri & Ebadzadeh, 2014; Jain & Murthy, 2016), kernel density estimation (Moon, Rajagopalan, & Lall, 1995), k-nearest neighbour (Kraskov, Stogbauer, & Grassberger, 2004; Gao, Oh, & Viswanath, 2017), Parzen window (Kwak & Choi, Input feature selection by mutual information based on Parzen window, 2002b; He, Zhang, Hao, & Zhang, 2015) B-spline (Daub, Steuer, Selbig, & Kloska, 2004), adaptive partitioning (Fraser & Swinney, 1986; Darbellay & Vajda, 1999); and fuzzy-based (Yu, An, & Hu, 2011; Hancer, Xue, Zhang, Karaboga, & Akay, 2015) approaches. These estimation methods typically involve some pre-set parameters whose optimal values heavily depend on problem characteristics. Parameter settings could possibly be the major source of large estimation errors but still the parameters are often assigned with arbitrary values because there is no clear-cut rule provided (Williams & Li, 2009). In addition, there are so many available options for the mutual estimation calculations. Therefore, the efficiency of a feature selection approach greatly relies on the method applied.

A frequently used criterion for dimensionality reduction is to identify features with the highest capability to preserve the manifold structure. Such a criterion has gained widespread attention since in many cases of interest, the recorded data are concentrated around a low dimensional manifold (submanifold) which is embedded in a high dimensional ambient space. The popular methods that use this criterion include principal component analysis (Wold, Esbensen, & Geladi, 1987), linear discriminant analysis (Izenman, 2013; Xanthopoulos, Pardalos, & Trafalis, 2013), Laplacian eigenmap (Belkin & Niyogi, 2003), locally linear embedding (Roweis & Saul, 2000), locality preserving projection (LPP) (He & Niyogi, Locality preserving projections, 2004) and Laplacian score (He, Cai, & Niyogi, Laplacian score in feature selection, 2006). The first two reduce the dimensionality based on global manifold structure preservation while the last four are based on local manifold structure preservation.

The term structure preservation in dimensionality reduction conceptually refers to the scheme to maintain major structural characteristics when mapping the data from high dimensional space to low dimensional space. Technically, the quality of structure preservation can be measured based on the preservation ability in terms of keeping connective similarity among sample points in high dimensional space to sample points in low dimensional representation.

Local structure preservation techniques, as its name implies, emphasize preserving the underlying local structure within the neighbourhood around each data point. Geometrically, such approaches try to retain the nature structure of the close-distance points in the original high dimensional space to a low dimensional representation. While global approaches may also involve preserving local structure, they are different from local techniques in that they attempt to preserve geometric data structure of faraway points in the high dimensional space to a low dimensional space. PCA serves as a good example of global techniques, which is solely based on global structure preservation, while LDA would be a simple example that preserves data structure at both orientations.

PCA is an unsupervised feature extraction approach that aims to find mutually independent projections in the directions where maximum variance of the data lies, which essentially reveals the global manifold structure of the data space. Though this approach may give optimal data representation, it may not be able to provide optimal solution in the classification context. LDA overcomes this problem in supervised mode with the main idea being to find projections that achieve optimum class discrimination in a setting where samples from different classes are well separated as far as possible whereas samples from the same class are scattered together as close as possible. Specifically, these projections are obtained based on an objective function that maximizes the ratio of between-class variance to the within-class variance.

Locality-based structure preservation techniques has gained considerable attention recently and demonstrated to be a successful strategy for dimensionality reduction in many learning tasks such as classification, clustering and visualization. The basic assumption of this technique is based on a simple geometric intuition that two data points tend to share the same characteristic (or class) if they are sufficiently close to each other. This assumption then leads to a key concept that any two close points in the original feature space should remain close in a reduced dimensional space. Owing to the fact that the technique relies on geodesic data structure, a nearest neighbour graph is constructed to model the proximity relation between data points and thereby discovers the intrinsic local manifold structure hidden in the high dimensional space. The technique has the advantage of relatively less affected by outliers since only local distances are considered which helps to prevent overfitting (Belkin & Niyogi, 2003). As data may reside on or close to a nonlinear submanifold structure, various nonlinear locality-based structure preservation methods were suggested in the literature, among which the most popular ones are locally linear embedding (Roweis & Saul, 2000), Isomap (Tenenbaum, De Silva, & Langford, 2000), and Laplacian eigenmap (Belkin & Niyogi, 2003). Though remarkable performance can be achieved by these nonlinear methods, their nonlinear property can only be achieved at the price of high computational cost.

Moreover, these nonlinear methods do not allow any new test point to be mapped into an existing reduced-dimensional space in a straightforward manner. An extension method is therefore required to evaluate the map of a new test point and in this case only estimation of the mapping can be performed (Maaten, Postma, & Herik, 2009). Since error in the estimation may occur, the embedding of new test points may not appropriately reflect the submanifold structure accordingly. Thus, how to map new points into an existing reduced-dimensional space still remains an issue.

Driven by the strength of locality-based geometrical approach as well as the aforementioned nonlinear method deficiencies, LPP emerged to provide a linear version of the Laplacian eigenmap. Although LPP is linear, it shares certain useful common properties of the nonlinear methods due to the fact that LPP adopts the same variational principle as for the

Laplacian eigenmap. This enables LPP to discover the nonlinear manifold structure of the data to some extent. Unlike the nonlinear methods that yield mappings which are defined only on training data points, LPP comes with a solution where its mapping is defined everywhere, thereby allows any new test point to be placed naturally into the reduced dimensional space.

Note that all the aforementioned local manifold structure preservation methods (except Laplacian Score) are designed for feature extraction. Yet, these methods have also been applied to feature selection context (Zhao, Lu, & He, 2008; Sun, Todorovic, & Goodison, 2010; Shang, Chang, Jiao, & Xue, 2017; Yao, Liu, Jiang, Han, & Han, 2017). As mentioned earlier, global techniques include either preserving the global structure of data alone or preserving both global and local structures simultaneously. Even so, there has been a growing interest in global manifold structure preservation methods which integrate both global and local information for feature selection. Recently reported studies in this field can be found in Zhang et al. (2011); Ren et al. (2012); Shu et al. (2012); Yu (2012) and Tong & Yan (2014). Interestingly, however, an important discovery made by Liu et al. (2014) revealed that preserving the local structure is more critical than preserving the global structure when feature selection is considered in unsupervised setting.

Real world data are rarely perfect because of numerous reasons such as faulty measuring device, error in data collection, inaccurate source or non-reporting information (e.g. missing data values). All these contributing factors to data imperfection creates a form of data known as noisy data.

Effectively handling noisy data is crucial for a classification task since the presence of noise may severely degrade the predictive accuracy and even slow down the construction of a classifier model (Zhu & Wu, 2004; Saez, Galar, Luengo, & Herrera, Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition, 2014; Wickramasinghe, 2017). Such negative impacts on performance usually happen because data corrupted by noise could bring new unnecessary and false-data patterns. For instance, when a high-level noise is present, an additional data cluster is formed or perhaps on the other way round, the extracted pattern will suffer loss of important data clusters (Saez, Galar, Luengo, & Herrera, Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition, 2014). Thus, managing noisy data is desired and one feasible solution to this problem is to perform a pre-processing step which specifically aims to enhance the data quality before a classifier is built.

In data mining research, there are two categories of noise, namely, class noise and attribute noise (Garcia, Luengo, & Herrera, 2016). Class noise refers to corruptions present in the class attribute which occur when instances are assigned with wrong class labels or when identical instances are recorded with different class labels. Meanwhile, attribute noise refers to errors or corruptions present in one or more values of the input attributes (or features) of the data instances. Generally, managing attribute noise is more complex than class noise. The rationale behind this should be easily understood as attribute noise may distort multiple values of an instance but class noise only corrupt one value, if any. Owing to the same rationale, it is not a good idea to handle noisy data by removing instances containing noise in only some of the attributes while there are still many remaining attributes carrying useful information. In this particular problem, feature selection is seen as an alternative solution to lead the data towards a finer quality.

Since data mining started to gain its popularity in 1990s, feature selection and noisy data have been well studied separately but little is known about the interaction between them. It is only recently that the combination of the two has been empirically investigated. However, among the efforts considering noisy data in feature selection, many have been directed to address the problems of class noise (Altidor, Khoshgoftaar, & Van Hulse, 2011; Shanab, Khoshgoftaar, Wald, & Napolitano, Impact of noise and data sampling on stability of feature ranking techniques for biological datasets, 2012; Shanab, Khoshgoftaar, & Wald, Evaluation of wrapper-based feature selection using hard, moderate, and easy bioinformatics data, 2014; Zhao Z. , 2017) because literature findings have shown that the effect of class noise is more detrimental than attribute noise in the classification context (Quinlan, 1994; Zhu & Wu, 2004; Nettleton, Orriols-Puig, & Fornells, 2010; Saez, Galar, Luengo, & Herrera, Tackling the problem of classification with noisy data using multiple classifier systems: analysis of the performance and robustness, 2013). Despite the fact that class noise is more harmful than the attribute noise, the empirical study conducted by Zhu & Wu (2004) revealed that class noise at some points could be more critical to learning classifiers. While many efforts have been made for dealing with class noise, research on handling attribute noise has not made considerable progress. The report by Zhu & Wu (2004) even highlighted that the class attribute of real-world data, in truth, is typically much cleaner than the input attributes. Accordingly, attribute noise deserves wider attention than it is currently receiving.

Over the past few decades, there has been a lot of interest on kernel methods in various learning systems for analysing nonlinear patterns. The basic idea of kernel methods is to map nonlinear data that is linearly inseparable in the original input space to a higher dimensional (possibly infinite) feature space where linear separations (or relations) can be achieved. Since the linear geometry of the data in the feature space is embedded in dot products between data instances, the mapping from the original data space to the feature space does not have to be performed explicitly but just needs some defining form of dot products in the original input space. This nonlinear mapping strategy is the so called 'kernel trick', which is the essence of the kernel methods. Taking into advantage of this kernel trick implies that the coordinates of the data in the feature space are not required. Kernel methods are preferable to other nonlinear methods because they do not involve any nonconvex nonlinear optimization procedure but merely require solution for the eigenvalue problem (Kwok & Tsang, 2004), thus the risk of being trapped in local minima can be avoided. This special feature, along with the brilliant idea of kernel approach, have led to many significant research advances such as kernel principal component analysis (kernel PCA) (Scholkopf & Smola, 1997), kernel discriminant analysis (Mika, Ratsch, Weston, Scholkopf, & Mullers, 1999a; Liu, Lu, & Ma, 2004; Zheng, Lin, & Wang, 2014), kernel-based clustering (Camastra & Verri, 2005; Yin, Chen, Hu, & Zhang, 2010; Tzortzis & Likas, 2012; Kang, Peng, & Cheng, 2017) and kernel regression (Blundell & Duncan, 1998; Yan, Zhou, Liu, Hasegawa-Johnson, & Huang, 2008; Brouard, Szafranski, & d'Alché-Buc, 2016).

It is not exaggerate to claim that kernel PCA is one of the most influential kernel-based methods for data dimensionality reduction reported in the literature. Kernel PCA was originally introduced by Scholkopf & Smola (1997) as a nonlinear feature extraction method to overcome the drawback of PCA which can only find linear structure in the data as mentioned earlier. Kernel PCA mimics the underlying concept of PCA but it applies the same linear scheme in the feature space instead of in the input space. Since its introduction, there has been a great deal of attention given to expand the approach for a variety of applications such as image processing (segmentation/face recognition) (Schmidt, Santelli, & Kozerke, 2016), process monitoring (Zhang, An, & Zhang, 2013; Reynders, Wursten, & De Roeck, 2014; Jaffel, Taouali, Harkat, & Messaoud, 2017), fault detection (Choi, Lee, Lee, Park, & Lee, 2005; Navi, Davoodi, & Meskin, 2015), and forecasting, just to name a few.

While the nonlinear mapping from the input space to the feature space in the kernel PCA has been a very useful concept for many applications, the reverse mapping from the feature space back to the input space is also of practical interest. The results of this reverse mapping are called pre-images. Knowing the fact that pre-images of kernel PCA are very useful for pattern denoising (Abrahamsen & Hansen, 2011; Mika, et al., 1999b; Zheng, et al., 2010; Li, et al., 2016), it is thus relevant to explore their potential for feature selection in the presence of noisy data.

1.3 Research Objectives

Based on the above detailed discussion, it is interesting to explore the followings opportunities that may enhance existing feature selection methods:

1. Application of non-mutual-information based criteria to measure feature relevancy and redundancy.
2. Utilisation of local data structure based on locality preserving projection to guide an unsupervised feature selection.
3. Exploitation of denoised patterns by kernel pre-images for feature selection from data with attribute noise.

These opportunities were explored in this research and new feature selection methods are proposed. These new methods can overcome some issues associated with existing methods and are more reliable in a way that they can find better or competitive feature subset for many real applications.

In Wei & Billings (2007), a *forward orthogonal search* (FOS) algorithm was introduced for feature selection and ranking. In the algorithm, features are selected by *maximizing the overall dependency* (MOD) between features where the primary objective is that the overall features in the original measurement space should be adequately represented by the selected feature subset. The hill-climbing search strategy with a straightforward measurement criterion makes the FOS-MOD algorithm conceptually simple and easy to implement. Although the algorithm may not always find optimal subset as the search is non-exhaustive, it is proven that the feature selection method is efficient enough to be employed for dimensionality reduction.

In the new methods to be proposed, the principal idea of the FOS-MOD approach is further developed and adapted to improve feature selection performance. Detailed discussions are given in the chapters to come.

1.4 Research Contributions and Publications

This research has made clear contributions to knowledge by exploring the three research opportunities mentioned earlier where each of which leads to a new feature selection method. Specifically, the contributions of the thesis are detailed as follows:

1. **The maximum relevance-minimum multicollinearity (MRmMC) method for feature selection**

   The MRmMC method addresses the issues concerning the existing maximum relevance-minimum redundancy methods, especially those which are based on mutual-information theory. This method can be seen as an alternative relevancy-redundancy criterion for feature selection that avoid mutual-information based approach. Unlike mutual information based approach, this feature selection method has the advantage of not involving any pre-defined parameters, thereby eliminating any uncertainty and allowing consistency in the feature selection results.

2. **The sequential orthogonal search for local largest structure (SOS-LLS) method for feature selection**

   The SOS-LLS method is meant to utilised the information of the local data structure as a measurement criterion in which the special characteristics offered by locality preserving projection will be employed. The approach is different from the other state-of-the-art feature selection methods that also utilised local data structure information as it is not just utilised purely local data structure information for the selection criterion but it also evaluates feature importance jointly to take into account feature redundancy rather than individually.

3. **The sequential orthogonal search of kernel pre-images (SOS-KPI) feature selection for noisy data**

   The idea of SOS-KPI method is to consider a research gap concerning data with noise where in particular, very limited research works have been emphasized on selecting features from data contaminated with attribute noise compared to the class noise. Since this feature selection is mainly intended to look at the effectiveness of considering the attribute noise and class noise is assumed as not available, the approach is therefore developed in unsupervised manner.

Several publications have been produced through the course of the research:

1. Azlyna Senawi, Hua-Liang Wei and Stephen A. Billings, 2017. *A new maximum relevance-minimum multicollinearity (MRmMC) method for feature selection and ranking.* Pattern Recognition, Vol 67, pages 47-61. (https://doi.org/10.1016/j.patcog.2017.01.026). [Impact Factor (2016): 4.582; Number of citations (Google Scholar): 11]

2. Azlyna Senawi, Hua-Liang Wei and Stephen A. Billings. *Unsupervised feature selection based on local largest structure preservation.* To be submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.

1.5 Organization of the Thesis

The thesis contains six chapters. The remaining five chapters are briefly summarized below.

In Chapter 2, the basic notions of feature selection is discussed in detail; these are important to fully understand associated specific topics. A theoretical review of the orthogonal transformation, as the pillar of the research, is also presented.

Chapter 3 is particularly focused a new relevancy-redundancy feature selection method, called the *maximum relevance-minimum multicollinearity* (MRmMC) feature selection method. Prior to the introduction of MRmMC, the deficiencies of the existing maximum relevance-minimum redundancy methods are analyzed to help the understanding of what are the forces that motivate the new method.

In Chapter 4, another new method which is referred to as *sequential orthogonal search for local largest structure* (SOS-LLS) is proposed; it is meant to utilise the underlying local geometrical structure in data. This chapter is preceded with a brief but concise discussion on the power of local structure that inspired the proposed method, followed by a review on related works which include a comprehensive discussion of locality preserving projection (LPP) to be utilised to detect significant features in SOS-LLS. The proposed SOS-LLS method is then presented theoretically and evaluated experimentally.

Chapter 5 presents the third feature selection method, which is referred to as the *sequential orthogonal search of kernel pre-images* (SOS-KPI) method. As this method deals with noisy data, a brief discussion on two categories of noise in data mining is given at the beginning of the chapter to highlight the motivation for the SOS-KPI method.

Chapter 6 gives the overall research summary and conclusion, followed by some future research directions.

1.6 Summary

This chapter has discussed the research background and specified the research objectives to be achieved. The contribution of the research to knowledge has also been highlighted.

In the next chapter, the basic notion of feature selection will be discussed.

Chapter 2

Feature Selection and Forward Orthogonal Search

2.1 Introduction

This chapter is mainly reserved for a comprehensive discussion on feature selection necessity, concepts, procedures and approaches. The discussion also includes reviews on past and recent feature selection strategies. A theoretical review of the orthogonal transformation which is a part of the key strategy for each of the new feature selection methods to be proposed is also provided.

2.2 Feature Selection Objectives

Basically, the objectives of feature selection are (a) to improve data mining performance, (b) to speed up data mining algorithms, (c) to facilitate learning for domain experts about the data generated, and (d) to provide more cost-effective future data collection (Guyon & Elisseeff, 2003).

Usually, not all of these goals can be successfully achieved in a proposed feature selection method. Some methods only cater for one or two of them and some even tried to reach all the three goals. When a method tries to meet an objective, it is often that the others are likely need to be compromised. This will be explained further later on.

2.3 Basic Concepts

Assume that there are a total of M original features in a dataset. Feature selection refers to a process of searching an optimal or suboptimal subset of m features from the M features (Abandah & Malas, 2010). The resulting feature subset from the process should essentially leads to performance improvement or at least with minimal performance degradation as much as possible for the task under consideration.

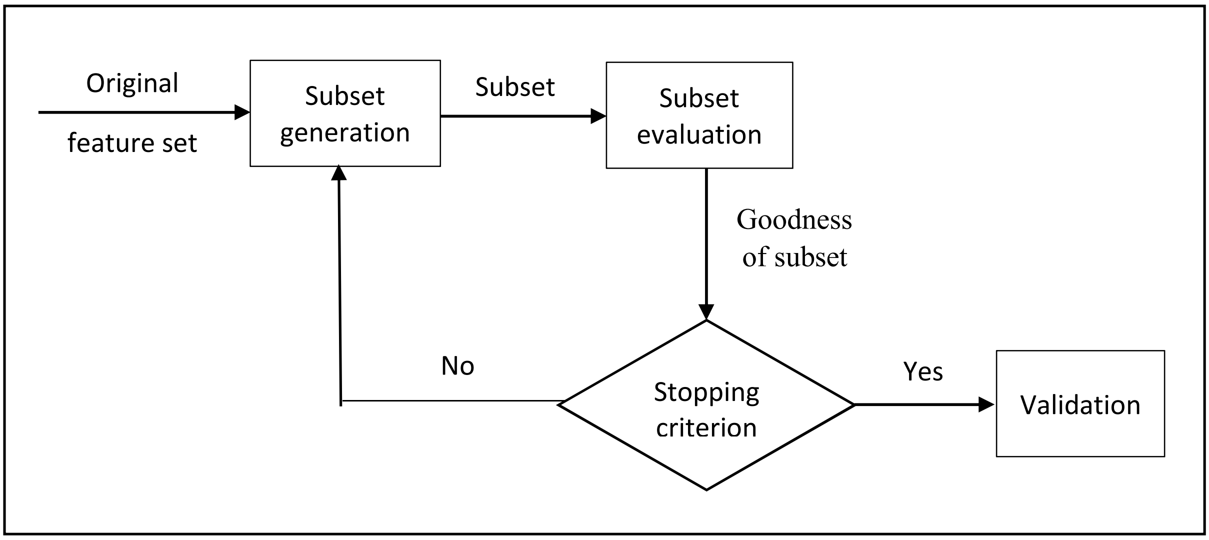

Figure 2.1: Four basic steps of a feature selection method (Dash & Liu, 2003).

Referring to Figure 2.1, a feature selection method is a composition of four basic steps (Dash & Liu, 2003): (1) feature subset generation, (2) feature subset evaluation, (3) stopping search decision and (4) results validation. Feature subset generation is a searching procedure that generates possible optimal/suboptimal subsets of features for evaluation by employing certain search strategy. The potential of every generated subset to be chosen is then evaluated either by using an independent or dependent criterion. Feature subset generation and evaluation processes are repeated until a subset that satisfies the imposed selection stopping criterion is met. After the best feature subset is obtained, a validation step is made using a test dataset by comparing the feature selection method constructed with other well established or competing methods.

2.4 Feature Subset Generation

There are two key concepts for feature subset generation: the search starting point(s) and the search strategy.

2.4.1 Search Starting Point

The search for the most significant feature subset may start with an empty set of features, a full set of features or a random subset of features. The search starting point(s) will determine the search direction (Liu & Yu, 2005). If the search starts with an empty set and the most significant features are progressively added to the set, it means that a *forward selection* approach is applied. Instead, if the search starts with a full set of features and the least significant features are progressively removed, a *backward selection* approach is adopted. An option to forward and backward selections is the *bidirectional selection* which is a simultaneous search approach of forward and backward selections. Meanwhile, if the search begins with a random subset of features, it can either proceed using any search direction discussed previously or continues with random features addition (or removal).

Assuming that there is no prior knowledge about which features contribute to optimal feature subset, there is no difference in searching capability between forward selection and backward selection for most problems (Caruana & Freitag, 1994; Aha & Bankert, 1996; Liu & Motoda, 2012). In other words, applying forward direction will find optimal/suboptimal feature subset as fast as using backward direction. However, employing bidirectional selection by holding the same assumption renders a faster result than using single directional search. This appears to be true since bidirectional selection starts searching from both end directions and the search will stop in one side of the directions before the other direction does.

2.4.2 Search Strategy

After the search starting point has been determined, the next step is to decide a search strategy to be used. An exhaustive search for the best subset when there exist 2^M candidate subsets is impractical for large M and even with a moderate M since it is too time consuming. Hence, different search strategies are used in feature selection algorithms and mostly render suboptimal solutions. The search strategies can be categorized into three main groups, namely complete search, sequential search and random search.

a) ***Complete search.*** A complete search warrants the acquisition of an optimal feature subset. An exhaustive search obviously falls into this category and it is best used when number of original features M of a dataset is small. Nevertheless, a search does not necessarily to be exhaustive in order for it to be complete. The non-exhaustive complete search strategy offers a more intelligent approach which just requires a smaller number of competing candidate subsets for evaluation. The optimality condition is assured as the approach is developed to have an ability to retrace evaluation of prior subsets (Dash & Liu, 1997). The most prominent example is the branch and bound (B&B) method (Narendra & Fukunaga, 1977). Generally, other complete search methods proposed after that such as best first search (Xu, et al., 1988) are an adaptation of B&B.

b) ***Sequential search.*** A sequential search is applied when one feature is added or removed progressively using a certain search direction. Also known as hill-climbing or greedy search, this type of search strategy is considered as having simple search structure although it may not be able to find optimal subset due to its incomplete search condition. Two simplest forms yet still popular sequential search are sequential forward selection (SFS) and sequential backward selection (SBS). SFS begins the search with an empty set and one feature is added iteratively whereas SBS begins with a full set of features and one feature is removed for each step of iteration. Instead of adding or removing one feature at a time, an alternative way of applying a sequential search is by using (p,q) sequential search (PQSS) that iteratively add (or remove) p features and then remove (or add) q features with $p > q$ (Dash & Liu, 1997). PQSS is an attempt to accommodate SFS and SBS deficiencies which fail to re-evaluate the goodness of a feature after being added/removed by having some backtracking abilities. The idea of PQSS was then extended with floating-based search concept and led to the introduction of two more popular sequential search methods: sequential forward floating search (SFFS) and sequential backward floating search (SBFS) (Pudil, et al., 1994). Both methods try to identify significant features by allowing dynamic number of features added or removed in the searching process. Among all methods in sequential search family, sequential floating-based search was found to be the best option (Pudil, et al., 1994; Somol, et al., 1999); although it is just limited to small and medium size of search space (Kudo & Sklansky, 2000).

c) ***Random search.*** Several feature subsets can be obtained as solutions to a feature selection problem using this search strategy. Also called as nondeterministic search strategy, the search begins from a subset selected at random. The search will then continue with subsets generated based on sequential search strategy as proposed in random-start hill-climbing and random mutation hill climbing (RMHC-PF1) (Skalak, 1994) methods. The sequential search procedure alone is irreversible to rectify poor features being added or good features being removed in the early phase of the search procedure. Therefore, random search enables sequential search to begin the search with a more significant starting point. A random search may also continue with subsets obtained in a totally random style using for example the Las Vegas Algorithm (Liu & Setiono, 1996a; Liu & Setiono, 1996b; Liu & Setiono, 1998). Another random strategy that can be used for feature selection is the evolutionary-based approaches. Inspired by the biological evolution and/or collective behaviour of species in nature, it has recently started gaining attention in the feature selection research due to its capability to give comparable performance with lower computational time. Two notable approaches are genetic algorithms (Siedlecki & Sklansky, 1989; Yang & Honavar, 1998) and particle swarm optimization (Lin, et al., 2008; Unler & Murat, 2010; Moradi & Gholampour, 2016; Mafarja & Mirjalili, 2017). The randomized search design of all approaches preventing the search being trapped by local optima (Liu & Motoda, 2007) and also identify interdependencies between features (Liu & Setiono, 1996a; Pradhananga, 2007). However, this search strategy requires values for some control parameters involved to be decided appropriately in advance. Poor values assigned to these parameters could lead to suboptimal results as the optimality of the final feature subset depends on the choice of values assigned to different parameters involved.

Basically, the choice of a search strategy is a trade-off between optimality and computational efficiency. Table 2.1 shows a comparison of search strategies in terms of optimality and computational efficiency which serve as a brief guideline for choosing an ideal search strategy.

Table 2.1: A comparison of different search strategies.

Search strategy	Optimality	Computational efficiency
Complete	The attainment of an optimal subset is guaranteed	Slow
Sequential	May not be able to find an optimal subset since it does not visit all possibilities from the search space	Generally faster than complete search
Random	The optimality subject to the determination of appropriate values for the parameters involved	Generally faster than complete search

All optimal methods can be expected considerably slow for high dimensional problems (Somol, et al., 2010). Therefore, it is often preferable for many high dimensional problems to employ the suboptimal methods that compromise subset optimality for better computational efficiency. In cases where time is not a constraint to gain optimal solution, complete search strategy should be employed.

Other than the search strategy factor, there are many other factors must be considered in choosing or designing a feature selection method. A comprehensive discussion on this can be found in Liu & Yu (2005). In the next section, another dominating factor is discussed.

2.5 Feature Subset Evaluation Criteria

Feature subset evaluation is a process to decide whether a feature should be included in or excluded from a feature subset for final selection. The process is performed by evaluating the quality of every possible feature subset generated using an evaluation criterion. Different types of criteria can be used for the evaluation. However, one criterion may not necessarily give the same optimal subset as that generated by another criterion.

Choices of evaluation criteria can be categorized into two broad categories which are independent criteria and dependent criteria (Dash & Liu, 1997; Dash & Liu, 2003; Liu & Yu, 2005). Essentially, a criterion is categorized as either one of the two categories according to its evaluation dependency on mining algorithms. Independent criteria such as distance measures (Parthalain, et al., 2010; Banka & Dara, 2015), dependency measures (Mitra, et al., 2002; Das, et al., 2014; Jain, et al., 2018), information measures (Peng, et al., 2005; Hoque, et al., 2014; Che, et al., 2017) consistency measures (Dash & Liu, 2003; Shin & Miyazaki, 2016) and margin-based measures (Kira & Rendell, 1992; Gilad-Bachrach, et al., 2004; Chen, 2016) evaluate a feature subset by merely utilizing hidden characteristics lying on training data, without being tied to any mining algorithm. Whereas subset evaluation based on dependent criteria requires a mining algorithm specified in advance and relies entirely on the mining algorithm performance. In other words, the measurement used to evaluate the quality of a selected feature subset is the same indicator used to measure the mining performance.

Typically, independent criteria are used in filter models while dependent criteria are used in wrapper models. When the two types of criteria are used together then feature selection is integrated in a hybrid model. Therefore, different types of evaluation criteria distinguish different feature selection models.

2.6 Feature Selection Models

Existing feature selection methods can be broadly categorized into three classes: filter, wrapper and hybrid. These are briefly discussed below.

2.6.1 Filter Model

Feature subset selection with a filter model is independent of specific mining algorithms as the search is based on the subset relevance to the targeted evaluation criterion (i.e., independent criterion). Hence, filter model is not affected by any bias caused by the mining algorithm and is usually computationally fast. The independent property also implies feature selection has to be carried out just once because the result can be used for different mining algorithms. In addition, filter model is also considered as having simple search structure and thus relatively easy to understand in comparison with other feature selection models. With all these advantages, it is not surprising that filter model is often preferred in real applications.

Despite all the advantages, feature subset selected by the filter model may not lead to an optimal mining performance since feature selection is done without taking into account the mining algorithms properties. Basically, there are two different approaches of filter model. One is called the univariate filter approach where the relevance score of each individual feature is evaluated and features having low-scores are removed, therefore, ignoring feature dependencies which possibly render performance degradation. Most proposed filter techniques use this approach (Saeys, et al., 2007) because of its computational efficiency. Another approach called multivariate filter where feature dependencies are taken into consideration to cope with the problem of ignored feature dependencies in univariate filter.

2.6.2 Wrapper Model

In contrast to the filter model which selects feature subset relevant to the targeted evaluation criterion, the wrapper model selects a feature subset which is relevant to a predetermined mining algorithm. The mining algorithm is used as a black box to evaluate the quality of each candidate feature subset in order to find the best feature subset. This means that wrapper model performs feature selection based on mining performance level in which a feature subset is selected when mining algorithm shows an optimal performance while taking into account feature dependencies in the feature selection procedure. As a result, the feature subset selected using the wrapper model will give higher mining performance than the filter model since the wrapper model is designed to search feature subset that is particularly tailored to the employed mining algorithm. For the same reason, however, rendering the feature subset obtained by the mining algorithm is unlikely to be suitable for use with other mining algorithms. Besides, the wrapper model is computationally slower when compared to the filter model since the mining algorithm of the wrapper model has to perform its task repeatedly until the final feature subset that gives maximum mining performance is found. This explains why the filter model is preferable than the wrapper model in handling large feature space problems.

2.6.3 Hybrid Model

The hybrid model emerged with an aim to combine the advantages possessed by both the filter and wrapper models. The model applies both an independent measure and a mining algorithm to measure the quality of each feature subset in the search space. Since mining performance is used as a guideline to stop the search, feature selection results based on the hybrid model is therefore specific to the mining algorithm employed. Consequently, the selected feature subset may not fit well with other mining algorithms and hence the hybrid model suffers the same problem as in the wrapper model.

2.7 Supervised and Unsupervised Feature Selection

In feature selection problems, the class of the data can be labelled or unlabelled. Corresponding to this classification, there are two categories of feature selection research: supervised and unsupervised feature selection. Comprehensive discussions on these categories of feature selection can be found from Huang et al. (2006) and Liu & Motoda (2007).

In supervised feature selection, with all or sufficiently large of the class labels are available, the relevance of the features are measured based on the relationship between features and the class labels. The feature selection objective is clear where a subset of the original features that induces the most accurate classier in which the class labels are well separated will be selected.

Without the class labels in unsupervised feature selection, different approaches are used to evaluate the relation between features by analysing other possible aspects of the data such as discriminative power to find different clusters or groups in data. In contrast to supervised feature selection, the objective of unsupervised feature selection is less clear since the class labels are not exist to facilitate learning about the data being considered. This limits the learning ability of unsupervised feature selection methods in order to identify patterns lie in a dataset and consequently may also affects the choice of feature subset that is expected to represent the original features. The problem becomes more complicated if the actual number of clusters is unknown prior to training.

When the class labels are just available for only small part of the dataset then semi-supervised feature selection may be used as an option for dimensionality reduction. Semi-supervised feature selection can be considered as a special form of unsupervised learning. In this feature selection scheme, the small amount of data labelled is utilized because the availability of the labelled instances is considered as significant to guide unsupervised feature selection.

Generally, much of the feature selection research focused on supervised feature selection (Guyon & Elisseeff, 2003). Thus, unsupervised feature selection comparatively can be considered as new research areas. However, unsupervised feature selection research is increasingly gaining attention as more and more unlabelled and partially labelled datasets exist in real applications (Pedrycz, 1986).

2.8 Orthogonal Transformation

An orthogonal transformation T of any vector $\mathbf{x} \in R^d$ is a linear transformation that preserves the length of the vector. The transformation can be expressed as follows:

$$T : R^d \rightarrow R^d \quad \forall \mathbf{x} \in R^d \tag{2.1}$$

where the transformation space is also R^d. The transformation not only preserves the length of the vectors but also the angles between them.

Considering a matrix $\mathbf{P} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ representing n variable vectors, an orthogonal transformation (or also mention as orthogonalization) is performed on $\mathbf{P}$ by decomposing it into

$$\mathbf{P} = \mathbf{QR} \tag{2.2}$$

where $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2 \ldots, \mathbf{q}_n]$ is an orthonormal matrix with n orthogonal vectors such that $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}_n$ and $\mathbf{R}$ is an upper triangular matrix.

Several orthogonalization methods can be employed to perform the **QR** decomposition include classical Gram-Schmidt (CGS), modified Gram-Schmidt (MGS), Householder reflections and Givens rotation. The MGS orthogonalization is more popular than the CGS for practical application since it has better numerical stability, which means it is less affected by rounding errors (Bjorck, 1994; Yokozawa, et al., 2006). However, when compared to Householder reflection orthogonalization, the MGS orthogonalization is numerically less reliable. Unlike the Gram-Schmidt method that produces orthogonal vectors at each iteration step, the orthogonalization by Householder reflection only generates the orthogonal vectors at the end of the procedure. This causes only the Gram-Schmidt type method can be used when the orthogonalization needs iterative transformation.

The orthogonal transformation can be used for feature selection because of three notable advantages as described below:

(1) The transformed variables $\mathbf{q}_1, \mathbf{q}_2 \ldots, \mathbf{q}_n$ and the original variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are one to one mapping where every $\mathbf{q}_k$ in the new space retains the length of its corresponding $\mathbf{x}_k$. This gives an advantage for formulation of a feature selection

method as it provides the basis for preserving the physical interpretation of the original variables in the transformed variables.

(2) The fundamental concept lies behind the transformation scheme is simple and the orthogonal variables computation is even straightforward but can still produces robust results (the meaning of robust results here is related to the third advantage explained next).

(3) The most notable is its ability to minimize ill-conditioning effects, that is, the capacity to make the transformed variables less sensitive to noise or small errors contained in data. This particular trait allows the transformed matrix $\mathbf{Q}$ to inherit the main structure of matrix $\mathbf{P}$ as much as possible.

2.9 Summary

In summary, this chapter discusses the fundamental ideas of feature selection. A typical feature selection method involves four basic steps: feature subset generation, feature subset evaluation, stopping search decision and result validation. There are two key concepts for feature subset generation, namely, the search starting point(s) and the search strategy. In principle, search strategy and evaluation criterion are two critical factors in designing a feature selection method. Type of evaluation criterion being used in the search for the best feature subset also determines the class of feature selection model which can be filter, wrapper or hybrid.

Chapter 3

A New Relevancy-Redundancy Method for Feature Selection and Ranking

3.1 Introduction

This chapter presents the first feature selection method to be proposed which is of filter model with a new measurement criterion named as *maximum relevance-minimum multicollinearity* (MRmMC). The criterion being used is a new type of relevancy-redundancy criterion that objectively overcomes some issues associated with existing state-of-the-art criteria. In the proposed method, relevant features are measured by correlation coefficient based on conditional variance whereas redundant features are quantified based on multiple correlation assessment using an orthogonal transformation scheme.

The presentation of the proposed method is preceded with Section 3.2 where a brief introduction to the concepts of feature relevancy and redundancy is given. Next, Section 3.3 provides a comprehensive discussion of the existing relevancy-redundancy criteria in which the issues associated with them are also pointed out. Section 3.4 is mainly reserved for a comprehensive discussion on how feature relevancy can be assessed by means of conditional correlation. Section 3.5 presents the idea of feature redundancy assessment by utilising the concept of multicollinearity. The description also includes the interrelation of multicollinearity and squared multiple correlation coefficient, as well as how the coefficient can be used to quantify feature redundancy. A new feature selection criterion that tries to optimize both feature relevancy and feature redundancy is then introduced in Section 3.6. Section 3.7 gives details of the experimental setup and the procedure used in order to show the efficacy of the proposed method. The empirical results and extensive discussion are given in Section 3.8, followed by summary for the chapter in Section 3.9.

3.2 Relevancy and Redundancy

The concepts of feature relevancy and feature redundancy are translated and expressed by means of certain feature relationships in feature selection methods. The relevance of a feature is measured by evaluating its relationship with the target class label, while the redundancy of a feature is measured by its relationship with other features in the currently selected feature subset.

3.3 Related Work

Many feature selection methods in the literature use mutual information to measure feature relevancy and redundancy. In Battiti (1994), features are ranked according to their mutual information with respect to the class label and also with respect to the previously selected features. The mutual information based feature selection (MIFS) method proposed by Battiti (1994) follows hill climbing selection scheme and chooses the next best feature that maximizes

$$J(\mathbf{f}_i) = I(\mathbf{c},\mathbf{f}_i) - \beta \sum_{\mathbf{f}_j \in S} I(\mathbf{f}_j,\mathbf{f}_i) \tag{3.1}$$

where $I(\mathbf{c},\mathbf{f}_i)$ denotes mutual information between class label $\mathbf{c}$ and candidate feature vector $\mathbf{f}_i$ while $I(\mathbf{f}_j,\mathbf{f}_i)$ denotes mutual information between previously selected feature $\mathbf{f}_j$ which have been accumulated in subset S and candidate feature $\mathbf{f}_i$. The parameter β is a user predefined value that will control the importance of redundant features. The larger the value, the more the measurement criterion will remove redundant features.

A variant of the MIFS method called the MIFS-U (Kwak & Choi, 2002a) emerged later to overcome the MIFS limitation which does not reflect relationships between feature and class label properly in its redundancy term if β is set too large. The MIFS-U approach brought a slight change to the right-hand side term so that the MIFS criterion becomes

$$J(\mathbf{f}_i) = I(\mathbf{c},\mathbf{f}_i) - \beta \sum_{\mathbf{f}_j \in S} \frac{I(c,\mathbf{f}_j)}{H(\mathbf{f}_j)} I(\mathbf{f}_j,\mathbf{f}_i) \tag{3.2}$$

where $H(\mathbf{f}_j)$ is the entropy of $\mathbf{f}_j$. However, the MIFS-U approach is limited for uniformly distributed information.

As the number of features to be selected increases, the right-hand side term becomes incomparable with the left-hand side term for both MIFS and MIFS-U methods due to magnitude expansion of the right-hand side term (Estevez, et al., 2009). Because of this problem, the methods may be forced to select and prioritize irrelevant features rather than relevant and/or redundant features. Another problem with both methods is that their optimal solution depends on the value assigned to β with optimal β's being considered subject to data structure. Hence, no specific guided rule was given on how to choose parameter β. Apparently, a user may need to try different values before an optimal or acceptable suboptimal solution can be obtained.

The issue of incomparable terms in MIFS and MIFS-U methods mentioned earlier was overcome in the minimal-redundancy-maximum relevance (mRMR) feature selection criterion (Peng, et al., 2005) by substituting β with reciprocal of the subset S cardinality, $1/|S|$. This will prevent the cumulative sum of the second term from having an excessive value in the expansion at any number of feature subsets to be considered which then lead to two equivalent terms for comparison. The mRMR criterion maximizes

$$J(\mathbf{f}_i) = I(\mathbf{c},\mathbf{f}_i) - \frac{1}{|S|}\sum_{\mathbf{f}_j \in S} I(\mathbf{f}_j,\mathbf{f}_i)\,. \tag{3.3}$$

In Ding & Peng (2005), another form of relevancy-redundancy measurement criterion similar to the three criteria discussed above (i.e., MIFS, MIFS-U and mRMR) was introduced particularly for continuous variables. This criterion, referred to as the F-test correlation difference (FCD), does not involve the calculation of mutual information. It selects the next best feature that maximizes

$$J(\mathbf{f}_i) = F(\mathbf{c},\mathbf{f}_i) - \frac{1}{|S|}\sum_{\mathbf{f}_j \in S} \left| r(\mathbf{f}_j,\mathbf{f}_i) \right| \tag{3.4}$$

where $F(\mathbf{c},\mathbf{f}_i)$ is the F-test statistic (or t-test statistic if two-class classification task is considered) comparing feature $\mathbf{f}_i$ and the class label $\mathbf{c}$ whereas $r(\mathbf{f}_j,\mathbf{f}_i)$ can be chosen to be Pearson correlation coefficient, Euclidean distance or any other appropriate measure. One problem with the FCD criterion is that the first term (F-test statistic) is not comparable with the second cluster of terms (redundancy terms) as they have different range of magnitude. The F-test statistic can take any positive value, while the value of redundancy coefficient ranging from zero to one. As a consequence, the F-test value may dominate the optimization criterion and reduce the impact of the second cluster of terms.

This chapter presents a new alternative relevancy-redundancy criterion for feature selection, which is designed to take advantage of the idea of both the mRMR and FCD criteria, and meanwhile avoid the drawback of the two methods inherited from the original MIFS algorithm introduced in Battiti (1994). It is known that MIFS has a drawback in that its performance relies on the choice of the parameter β for controlling and penalising the redundancy; the optimal choice of the parameter β, however, strongly depends on the problem to be solved (Estevez, et al., 2009). The proposed criterion is different from the two criteria in that it does not require any pre-specification or determination of thresholds for parameter settings. In the proposed method, relevant features are measured using conditional variance (Wang, et al., 1994) while redundancy elimination is achieved through multiple correlation assessment using an orthogonal projection scheme (Whitley, et al., 2000). The combination of these methods was motivated by the requirement to form a robust criterion that allow a comparable evaluation of feature relevancy and redundancy, yet avoiding mutual information based approach. Unlike mutual information based feature selection, the proposed method has the advantage of not demanding any control parameters, thus preventing any uncertainty associated with the method and providing consistency in the results.

3.4 Feature Relevancy Assessment

While many powerful feature selection methods were proposed in the literature to tackle various issues, relatively less and limited work has been done to assess the correlation between discrete (nominal) and continuous (quantitative) features directly. The majority of the prominent correlation measures were specifically designed for use either between two features of the same data type or between continuous and ordinal features.

The point-biserial correlation coefficient (Tate, 1954) is the most popular measure suggested when one feature is discrete while the other one is continuous. Yet the measure can only be used when the discrete feature is dichotomous or possibly be made dichotomous which is not always the case for many applications. An effort was made in Wang et al. (1994) to fill this gap where a correlation measure between discrete and continuous features based on the underlying properties of marginal and conditional expectation and variance was introduced. The measure was adopted as part of the evaluation criterion for the feature selection approach that is specific to address some problem in mineral resources domain. In Jiang & Wang (2016), an efficient correlation measure based filter (ECMBF) algorithm was proposed for the assessment of both feature relevancy and feature redundancy for more general applications. The ECMBF algorithm requires two predefined parameters, to distinguish weak irrelevance/relevance and redundancy, respectively. The choice of the two parameters can significantly affect the quality of the selected feature subset. This is probably the main disadvantage of the algorithm. Another drawback of ECBMF is that the assessment of the redundancy of each candidate feature is independent of the current selected features. In this study, an alternative approach is desired to overcome these drawbacks. The proposed correlation based method uses two measures that simultaneously evaluate features' dependency and redundancy, based on which 'best' features are selected using a sequential forward algorithm. The proposed method in this chapter is different from other types of filter approaches for example the Fisher score based methods (Gu, et al., 2012).

In this study, the potential of the correlation measure proposed in Wang et al. (1994) is exploited; it will particularly be used to assess feature relevance. Towards better understanding the reliability of this correlation measure, its theoretical properties and conditions will be discussed first in detail.

Let X represent a quantitative random variable and Y represent a nominal random variable with some possible outcomes y_i. If every outcome y_i is described by a certain probability $P(Y = y_i)$ then the marginal expectation (Grimmett & Welsh, 2014) (also known as the expected value of X) symbolized by $E(X)$, is given by

$$E(X) = \sum_{y_i} P(Y = y_i)\, E(X \mid Y = y_i) \tag{3.5}$$

where $E(X \mid Y = y_i)$ denotes the conditional expectation of X given $Y = y_i$. It can be shown from this definition that the expected value of the conditional expectations, denoted by $E[E(X \mid Y)]$, is $E(X)$, that is

$$E(X) = E[E(X \mid Y)]. \tag{3.6}$$

Marginal variance of the random variable X is defined as

$$\mathrm{Var}(X) = E([X - E(X)]^2) = E(X^2) - [E(X)]^2. \tag{3.7}$$

Analogous to equation (3.7) the conditional variance of X given $Y = y_i$ is

$$\mathrm{Var}(X \mid Y) = E(X^2 \mid Y) - [E(X \mid Y)]^2). \tag{3.8}$$

Note that $\mathrm{Var}(X \mid Y)$ can be considered as a random variable, thereby theoretically permits the computation of its expected value as

$$E[\mathrm{Var}(X \mid Y)] = E\{E(X^2 \mid Y) - [E(X \mid Y)]^2\}. \tag{3.9}$$

Based on the additive law of expectation, the equation (3.9) can be rewritten as

$$E[\mathrm{Var}(X \mid Y)] = E[E(X^2 \mid Y)] - E([E(X \mid Y)]^2). \tag{3.10}$$

Applying the relationship given by (3.6) to the first term at the right-hand side of (3.10) yields

$$E[\mathrm{Var}(X \mid Y)] = E(X^2) - E([E(X \mid Y)]^2). \tag{3.11}$$

Next, it is of interest to consider the variance of the conditional expectation, marked by $\mathrm{Var}[E(X \mid Y)]$. Using the marginal variance definition given in (3.7), $\mathrm{Var}[E(X \mid Y)]$ can be expressed as

$$\mathrm{Var}[E(X \mid Y)] = E([E(X \mid Y)]^2) - [E(E(X \mid Y))]^2. \tag{3.12}$$

Applying (3.6) in (3.12) implies

$$\mathrm{Var}[E(X \mid Y)] = E([E(X \mid Y)]^2) - [E(X)]^2. \tag{3.13}$$

Then adding (3.11) to (3.13) gives

$$E[\mathrm{Var}(X \mid Y)] + \mathrm{Var}[\mathrm{E}(X \mid Y)] = E(X^2) - [E(X)]^2. \tag{3.14}$$

Notice that the right-hand side of equation (3.14) is equal to $\mathrm{Var}(X)$ as stated in (3.7). Hence, the following relationship is obtained:

$$\mathrm{Var}(X) = E[\mathrm{Var}(X \mid Y)] + \mathrm{Var}[E(X \mid Y)] \tag{3.15}$$

which is well known as the law of total variance. A special case of the law is $\mathrm{Var}(X) = E[\mathrm{Var}(X \mid Y)] \Leftrightarrow \mathrm{Var}[E(X \mid Y)] = 0$. This biconditional implication is true when every conditional expectation given $Y = y_i$ is equal to the marginal expected value. Since variances can never be negative, it is apparent that $\mathrm{Var}(X) \geq E[\mathrm{Var}(X \mid Y)]$ and $\mathrm{Var}(X) \geq \mathrm{Var}[E(X \mid Y)]$.

From equation (3.15) it can be observed that the overall variability of a random variable X consists of two components. One component is the expected value of the conditional variance, $E[\mathrm{Var}(X \mid Y)]$, that quantifies the average variability within outcomes. Another component is the variance of the conditional means, $\mathrm{Var}[E(X \mid Y)]$, that indicates how much the variability is between outcomes. The former is considered in the correlation measure which will be presented next.

The correlation coefficient that measure the relationship between a quantitative random variable X and a nominal random variable Y is defined by

$$r_{\mathrm{qn}}(X,Y) = \left(1 - \frac{E[\mathrm{Var}(X \mid Y]}{\mathrm{Var}(X)}\right)^{1/2} \tag{3.16}$$

which actually exploits the law of total variance. Based on previous discussions about $\mathrm{Var}(X)$ and $\mathrm{Var}[\mathrm{E}(X \mid Y)]$, it can be verified that $0 \leq r_{\mathrm{qn}}(X,Y) \leq 1$. A value of $r_{\mathrm{qn}}(X,Y)$ approaching '1' indicates that there is a strong correlation or dependency between X and Y. Meanwhile, the value of $r_{\mathrm{qn}}(X,Y)$ approaching '0' suggests that there is a weak relationship between X and Y. If X and Y are totally independent or uncorrelated, then $r_{\mathrm{qn}}(X,Y) = 0$, which is the special case of the law of total variance mentioned before. On contrary, the presence of perfect dependency or correlation between X and Y is indicate by $r_{\mathrm{qn}}(X,Y) = 1$.

The above correlation coefficient will be used to measure feature relevance. It will be integrated with multiple correlation assessment in order to define a new feature selection criterion that can measure both feature relevancy and feature redundancy simultaneously. The multiple correlation assessment can be used to identify features with multicollinearity and thus can be used to detect and remove redundant features.

3.5 Multicollinearity Redundancy and the Squared Multiple Correlation Coefficient

The idea of feature redundancy assessment for the method to be proposed is centred around the concept of multicollinearity. With this attention, the notion of multicollinearity redundancy is discussed exclusively in sub Section 3.5.1 and how it is related to the squared multiple correlation coefficient is also described herein after clearly via sub Section 3.5.2.

3.5.1 Multicollinearity Redundancy

A feature subset selected from a feature selection process should essentially lead to a performance improvement or at least with minimal performance degradation as much as possible for the task under consideration. This objective can be realized by selecting representative features that hold important information characterizing all original features. In particular, it can be done by not only selecting features that have high relevancy to the targeted class but also have low redundancy within selected features.

An ultimate feature redundancy occurs if a feature has exact linear dependency with the current selected features and thus provides no extra information. While exact linear dependency is rarely present in many real data, a significant type of redundancy is also taken into account in such a way that features with any potential multicollinearity will be removed. Multicollinearity is a term to describe the presence of strong correlation or high linear dependency among two or more independent variables. This means that a feature with multicollinearity can be linearly estimated by a set of other features at some high level of accuracy and therefore suggests such a feature has redundant information. In comparison to features having ultimate redundancy, features with multicollinearity redundancy still provide some unique information but not important enough to give notable impact for effective data analysis tasks for example classification.

Multicollinearity can be identified from high values of the multiple correlation coefficient. However, since the actual interest is to assess predictive power of the current selected features in estimating a considered feature, the squared multiple correlation coefficient is often used instead of the multiple correlation coefficient. The squared multiple correlation coefficient specifically indicates the proportion of the variation in the considered feature that is predictable from the selected features. The value ranges from 0 to 1 with higher values implying a better predictive power. When a maximum value of the squared multiple correlation coefficient is obtained it indicates a full predictive power which is the ultimate redundancy. Thus, the ultimate redundancy can be regarded as the best achievable multicollinearity. Note that the squared multiple correlation coefficient can be computed by utilizing pairwise orthogonal projection of features already selected (Wei & Billings, 2007; Billings, 2013). This will be further discussed in the next section.

3.5.2 The Squared Multiple Correlation Coefficient

Suppose that the set $F = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_M\}$ is a complete dataset of M features where each $\mathbf{f}_i = [f_1^{(i)}, f_2^{(i)}, \dots, f_N^{(i)}]$ is a feature vector composed by N observations. Also suppose that a subset S consisting $(k-1)$ features $\mathbf{f}_{i_1}, \mathbf{f}_{i_2}, \dots, \mathbf{f}_{i_{k-1}}$ has already been selected from the set of M original features. These $(k-1)$ features are then transformed into orthogonal variables $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}$ using certain type of transformation. If the next feature $\mathbf{f} = \mathbf{f}_{i_k}$ is selected and included into S later on, then the k th orthogonal variable, $\mathbf{q}_k$, associated to $\mathbf{f}$ is calculated by

$$\mathbf{q}_k = \mathbf{f} - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{q}_1}{\mathbf{q}_1^{\mathrm{T}}\mathbf{q}_1}\mathbf{q}_1 - \cdots - \frac{\mathbf{f}^{\mathrm{T}}\mathbf{q}_{k-1}}{\mathbf{q}_{k-1}^{\mathrm{T}}\mathbf{q}_{k-1}}\mathbf{q}_{k-1}. \tag{3.17}$$

The squared correlation coefficient between a feature $\mathbf{f} \in F - S$ and an orthogonal variable $\mathbf{q} \in \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k\}$ is defined as

$$\mathrm{sc}(\mathbf{f}, \mathbf{q}) = \frac{(\mathbf{f}^{\mathrm{T}}\mathbf{q})^2}{(\mathbf{f}^{\mathrm{T}}\mathbf{f})(\mathbf{q}^{T}\mathbf{q})} = \frac{\left(\sum_{i=1}^{N} f_i q_i\right)^2}{\sum_{i=1}^{N} f_i^2 \sum_{i=1}^{N} q_i^2} \tag{3.18}$$

Based on (3.18), the squared multiple correlation coefficient for each remaining feature $\mathbf{f} \in F - S$ with the selected features $\mathbf{f}_{i_1}, \mathbf{f}_{i_2}, \ldots, \mathbf{f}_{i_k}$ (or equivalently with $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k$) can be computed as

$$R^2(\mathbf{f}; \mathbf{q}_1, \ldots, \mathbf{q}_k) = \sum_{\mathrm{i}=1}^{k} \mathrm{sc}(\mathbf{f}, \mathbf{q}_i) \tag{3.19}$$

where the square root of R^2 geometrically represents the length of orthogonal projection of $\mathbf{f}$ in the directions of the orthogonal variables $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k$ divided by the norm (energy) of $\mathbf{f}$.

3.6 Monitoring Criterion

In order to choose features that are most relevant to the targeted class $\mathbf{c}$, the monitoring condition is to maximize the measure V as

$$V = r_{\mathrm{qn}}^2(\mathbf{f}_j, \mathbf{c}) \quad \text{such that} \quad \mathbf{f}_j \in F - S \tag{3.20}$$

which utilizes the squared value of the correlation coefficient given in (3.16). On the other hand, the squared multiple correlation coefficient defined in (3.19) is suggested to guide selection of features that are least mutually dissimilar or least redundant. Thus, the redundancy condition to be considered for measuring redundancy between feature $\mathbf{f}_j$ and the current selected feature subset S is to minimize the measure W:

$$W = R^2(\mathbf{f}_j; \mathbf{q}_1, \ldots, \mathbf{q}_k) = \sum_{\mathrm{i}=1}^{k} \mathrm{sc}(\mathbf{f}_j, \mathbf{q}_i) \quad \text{such that} \quad \mathbf{f}_j \in F - S \tag{3.21}$$

where $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k$ are orthogonal variables associated respectively with preceding selected features $\mathbf{f}_{i_1}, \mathbf{f}_{i_2}, \ldots, \mathbf{f}_{i_k}$ contained in S.

Because the aim of the feature selection is to select features that are highly relevant to the targeted class $\mathbf{c}$ and also has low redundancy with other selected features, both measures V and W are optimized simultaneously. A new feature to be added will be based on one possible single criterion combining both measures. The monitoring criterion used in this study is to maximize

$$J(\mathbf{f}_j) = r_{\text{qn}}^2(\mathbf{f}_j, \mathbf{c}) - R^2(\mathbf{f}_j; \mathbf{q}_1, \ldots, \mathbf{q}_k) \qquad \text{such that} \quad \mathbf{f}_j \in F - S \tag{3.22}$$

which can also be written as

$$J(\mathbf{f}_j) = \max_{\mathbf{f}_j \in F-S} \left[r_{\text{qn}}^2(\mathbf{f}_j, \mathbf{c}) - \sum_{\text{i}=1}^{k} \text{sc}(\mathbf{f}_j, \mathbf{q}_i) \right] \tag{3.23}$$

The correlation coefficient r_{qn} is squared in (3.22) so that a fair comparison can be made with the R^2 term. It is known from Section 3.4 that $0 \le r_{\text{qn}} \le 1$. This implies that the range of values given by r_{qn}^2 is the same as for the r_{qn}, that is, $0 \le r_{\text{qn}}^2 \le 1$. Note that the R^2 term also has the same range of values. Hence, the r_{qn}^2 term is completely comparable to the R^2 term and as such makes (3.22) a well-defined criterion. Owing to the same fact that both terms are in the similar scale which directly follows the logic of criterion (3.3), the proposed criterion (3.22) is thus requires no pre-set parameter to control feature redundancy. Clearly, there is no other adjusting parameters are required from user in the criterion. The feature selection method, based on the criterion (3.23) is referred to as the maximum relevance – minimum multicollinearity (MRmMC) method.

In the MRmMC method, the first feature is selected if it satisfies the optimization criteria stated in (3.20) and the rest are selected based on criterion as in (3.23) by using forward sequential search strategy. At every subsequent step, a new feature will be added to previously selected feature subset. This simple piecewise feature search strategy will avoid excessive computational burden to the MRmMC feature selection, and can therefore accelerate the feature search procedure. Note that although the search may lead to a suboptimal solution, it can meet the requirements for most real applications.

The proposed criterion in (3.22) can overcome the drawback of the MIFS approach, and it can effectively manage relevance and redundancy as follows. The first part, *V*, measures relevance using a correlation coefficient defined by (3.16) and (3.20), while the second part, *W*, measures the redundancy of a candidate feature with features in a selected feature set by evaluating the multicollinearity when the candidate feature is added to the existing feature subset.

The proposed criterion has the following advantages: i) The two parts of the criterion are comparable, and can result in a good balance between relevance and redundancy; ii) There is no need to pre-specify a control parameter as required in MIFS, and iii) the algorithm is relatively easier to implement. Some implementation details (pseudo-code) of MRmMC is shown in Figure 3.1.

```
Input:          $F = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\}$      // A complete dataset of $M$ features
Output:         $S$                                   // Subset of features
Initialize:     $L_1 = \{1, 2, \ldots, M\}$, $S = \{\}$
                $m$                                   // Number of features to be selected

for $j = 1$ to $M$
        Compute $V_j = r_{\mathrm{qn}}^2(\mathbf{f}_j, \mathbf{c})$ such that $\mathbf{f}_j \in F$;
end for

$\ell_1 = \arg\max_{j \in L_1}[V_j]$ such that $\ell_1 \in L_1$; $\mathbf{q}_1 = \mathbf{f}_{\ell_1}$; $\mathbf{z}_1 = \mathbf{f}_{\ell_1}$;
add $\mathbf{z}_1$ to $S$;

for $h = 2$ to $m$
        $L_h = L_{h-1} \setminus \{\ell_{h-1}\}$; $k = h - 1$;
        for $j \in L_h$
                Find $J(\mathbf{f}_j) = r_{qn}^2(\mathbf{f}_j, \mathbf{c}) - \sum_{i=1}^{k} \mathrm{sc}(\mathbf{f}_j, q_i)$;
        end for
        $\ell_h = \arg\max_{\mathbf{f}_j \in F - S}[J(\mathbf{f}_j)]$ such that $\ell_h \in L_h$;
        $\mathbf{q}_h = \mathbf{f}_{\ell_h} - \frac{\mathbf{f}_{\ell_h}^{\mathrm{T}}\mathbf{q}_1}{\mathbf{q}_1^{\mathrm{T}}\mathbf{q}_1}\mathbf{q}_1 - \cdots - \frac{\mathbf{f}_{\ell_h}^{\mathrm{T}}\mathbf{q}_{h-1}}{\mathbf{q}_{h-1}^{\mathrm{T}}\mathbf{q}_{h-1}}\mathbf{q}_{h-1}$;
        $\mathbf{z}_h = \mathbf{f}_{\ell_h}$;
        add $\mathbf{z}_h$ to $S$;
end for
```

Figure 3.1: The MRmMC algorithm.

The time complexity of the MRmMC method is determined by three main parts: the assessment of feature relevancy to the class label, the computation of the squared correlation coefficient, and the orthogonalization operations. Feature relevancy assessment has a linear time complexity of $O(MN)$ where M is the number of candidate features and N is the number of observations. The computation of the squared correlation coefficient has a worst-case time complexity of $O(M^2N)$ while the orthogonalisation procedure is of a complexity of $O((M-1)N)$. As a result, the overall time complexity takes the order of $O(M^2N)$.

3.7 Experimental Setup and Procedure

A series of experiments were conducted to test and analyse the efficacy of the proposed MRmMC method from several perspectives. Eight datasets were used as benchmarks, and relevant results were compared with those generated from mRMR and MIFS.

3.7.1 Benchmark Datasets

The eight public real datasets available from the UCI Machine Learning Repository, are depicted in Table 3.1. In order to provide comprehensive evaluation, the datasets were picked based on three different categories of dimensional size: low-dimension $(M \leq 10)$, medium-dimension $(10 < M \leq 100)$, and high-dimension $(M > 100)$. Important details of the chosen datasets are summarized in Table 3.1. Observe that the datasets are also varied in terms of number of instances and number of classes.

Table 3.1: A summary of the datasets characteristics.

Dataset	Number of features	Number of instances	Number of classes
Glass [N]	9	214	7
Magic Gamma [N]	10	19020	2
Vowel [N]	10	990	11
Statlog [N]	18	846	4
Mfeat Zernike [N]	47	2000	10
Sonar	60	208	2
Musk [N]	166	476	2
Mfeat Factors [N]	216	2000	10

[N]: The raw dataset was normalized for the proposed method in the experiment. This also means the dataset was normalized in classification accuracy computation for all classifiers.

3.7.2 Comparison with Similar Methods

The MIFS and mRMR methods are specifically employed for a comparison purpose as they possess similar forms of measurement criteria and use the same sequential feature search strategy. Feature subset solutions of the MIFS and mRMR methods were obtained by running the Feature Selection Toolbox (FEAST) (available at: http://www.cs.man.ac.uk/~gbrown/fstoolbox/) that was originally developed by Brown et al. (2012). In this work, the redundancy parameter was chosen to be $\beta = 1$ for the MIFS method. This choice of parameter value was in the appropriate range suggested by Battiti (1994).

3.7.3 Validation Classifiers

MRmMC is a filter method, and hence its efficiency might be different from one classifier to another classifier. Thus, four of the ten most influential algorithms in data mining (Wu, et al., 2008), namely, the *k*-Nearest Neighbour (*k*-NN), Naïve Bayes, support vector machine (SVM) and CART classifier algorithms, are used to verify the classification capability of the performance of the MRmMC method for feature subset selection. These classifiers were chosen not only because of their popularity but also because of their distinct learning mechanism. The aim is to test the overall performance of the newly proposed method in comparison to these popular classifiers.

Note that the number of nearest neighbours in the *k*NN classifier was chosen to be $k = 5$ in all experiments, and this is a fair choice for all the three methods: MRmMC, mRMR and MIFS.

3.7.4 Cross Validation Procedure

For each of the classifiers, the same holdout cross-validation scheme was used to test the performance. Particularly, 80% of the data were used for training whereas the remaining 20% were holdout (for testing) and once the training completed, these holdout data were then used to assess the spotted classification models in the testing stage.

In addition, to reduce variability in the assessment, 30 rounds of cross-validation were performed. The validation results are presented as the 95% confidence intervals for the classification accuracies based on the accuracies obtained from that 30 rounds.

3.8 Numerical Results and Discussion

Figure 3.2 through to Figure 3.9 show classification results over different number of selected features by the three feature selection methods, tested with the four classifiers. The *x*-axis in each figure represents the number of selected features while the *y*-axis represents the average classification accuracy based on 30 rounds of cross-validation. For clear visualization and due to space limitations, the plots only present the performance of the first 30 selected features even if more than 30 were selected. This doesn't affect the performance evaluation of the feature selection methods.

It can be observed that the overall pattern of the classification accuracies of the three methods based on the selected feature subset for Mfeat Zernike and Mfeat Factors datasets is comparable to each other for all the four classifiers as illustrated in Figure 3.6 and Figure 3.9, respectively. Interestingly, the classification accuracy by MRmMC outperforms the other two methods if only a few number of significant features need to be identified, and as more features were progressively added, MRmMC gains the same level of accuracy as the other two. This pattern is particularly distinct for Magic Gamma, Statlog, Sonar and Musk datasets as depicted in Figure 3.3, Figure 3.5, Figure 3.7 and Figure 3.8, respectively.

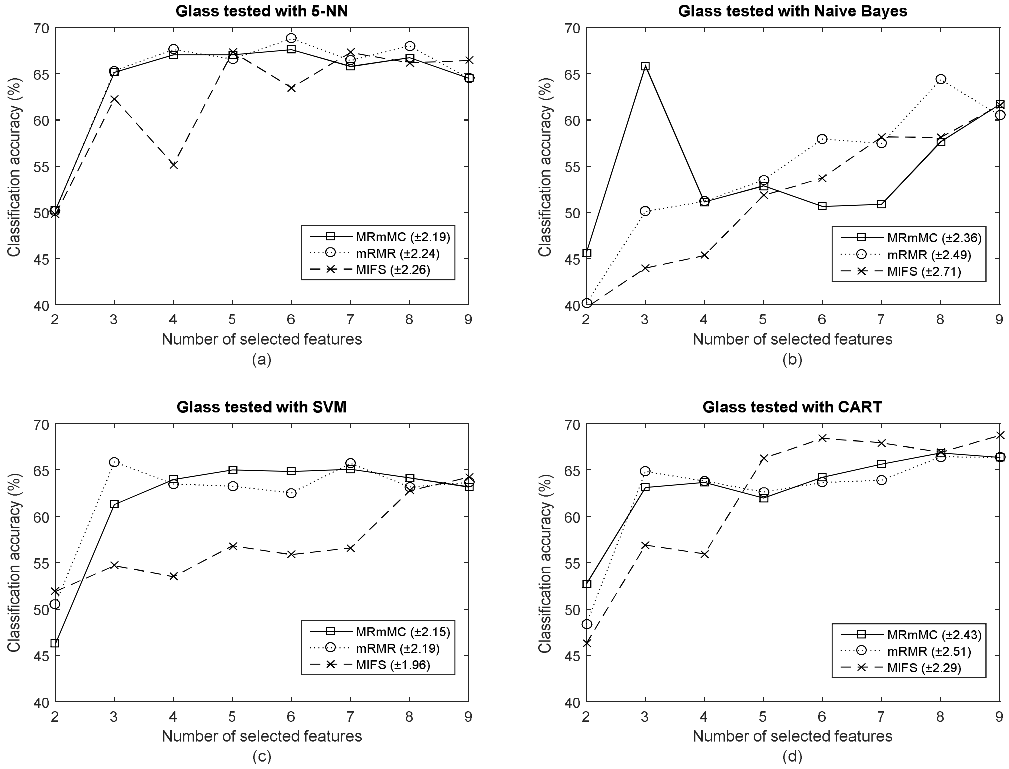

Figure 3.2: Classification results for Glass dataset over different number of selected features, tested with four classifiers: (a) 5-NN, (b) Naïve Bayes, (c) SVM and (d) CART. Each plot shows comparison among MRmMC, mRMR and MIFS methods.

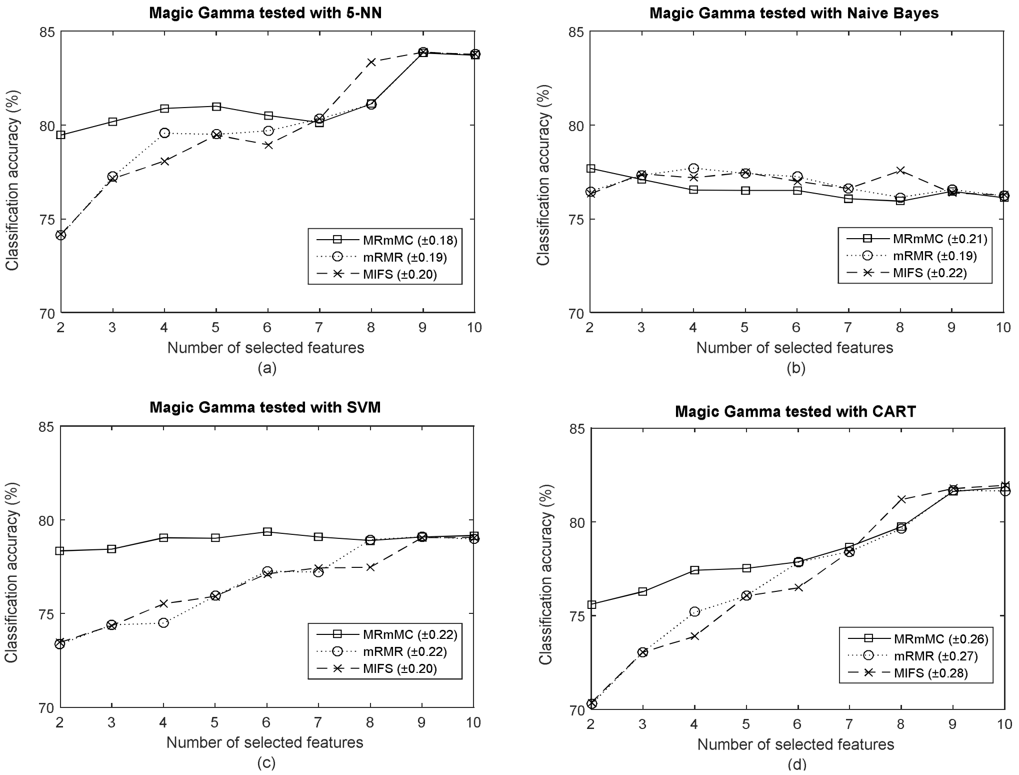

Figure 3.3: Classification results for Magic Gamma dataset over different number of selected features, tested with four classifiers: (a) 5-NN, (b) Naïve Bayes, (c) SVM and (d) CART. Each plot shows comparison among MRmMC, mRMR and MIFS methods.

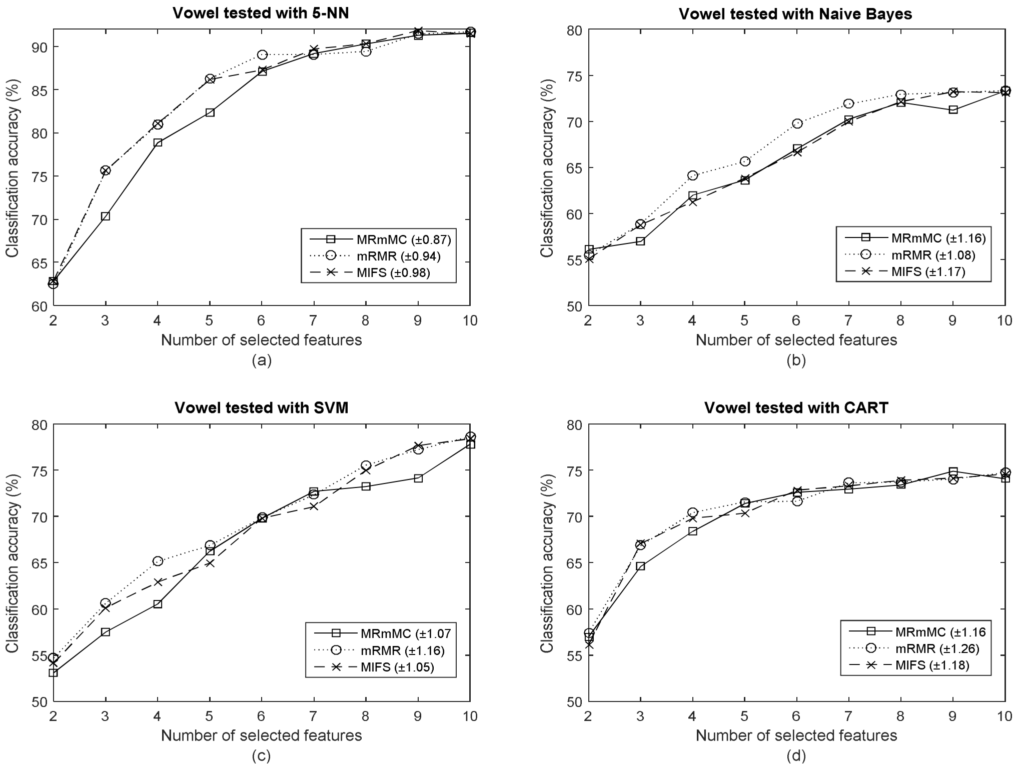

Figure 3.4: Classification results for Vowel dataset over different number of selected features, tested with four classifiers: (a) 5-NN, (b) Naïve Bayes, (c) SVM and (d) CART. Each plot shows comparison among MRmMC, mRMR and MIFS methods.

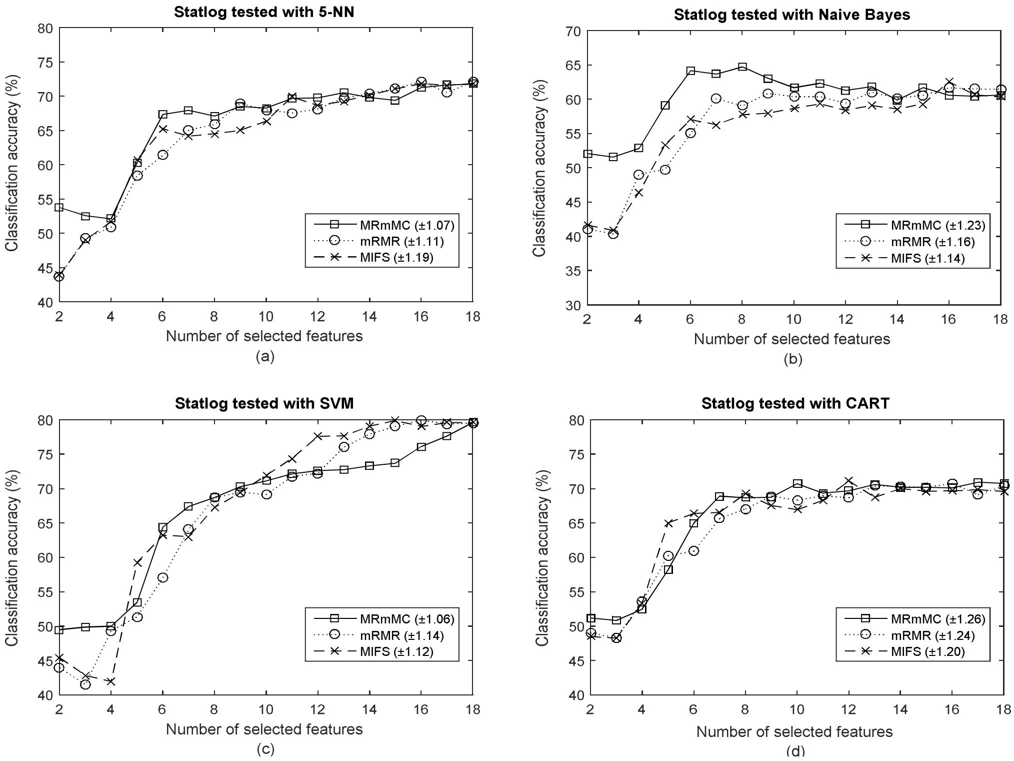

Figure 3.5: Classification results for Statlog dataset over different number of selected features, tested with four classifiers: (a) 5-NN, (b) Naïve Bayes, (c) SVM and (d) CART. Each plot shows comparison among MRmMC, mRMR and MIFS methods.

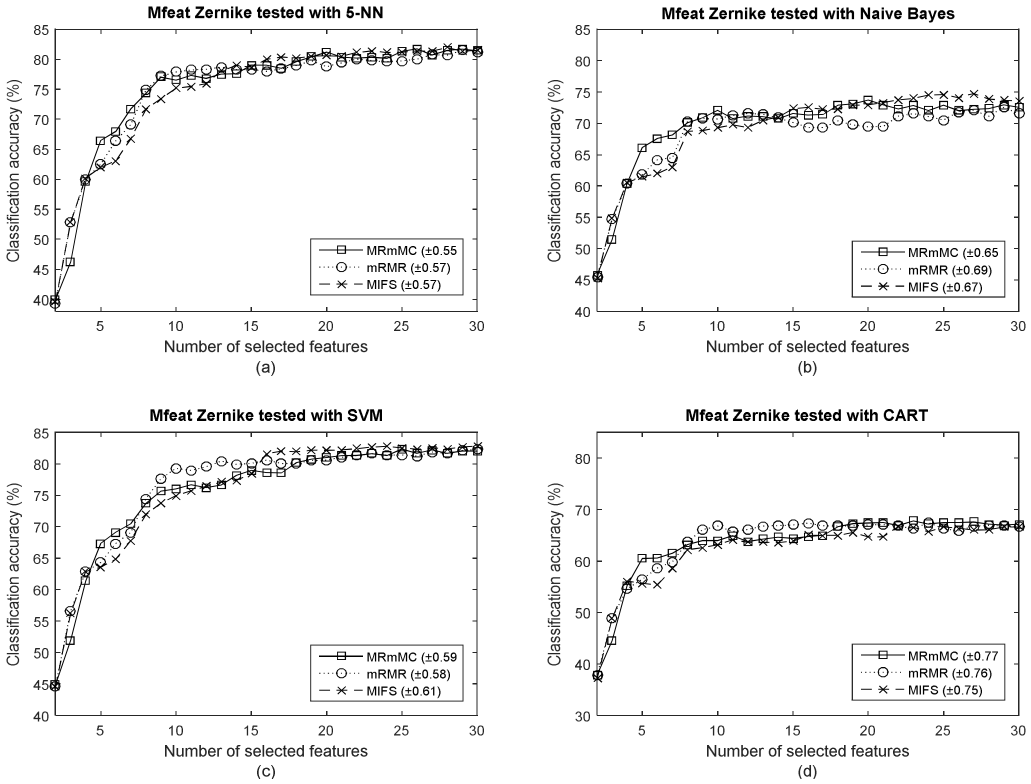

Figure 3.6: Classification results for Mfeat Zernike dataset over different number of selected features, tested with four classifiers: (a) 5-NN, (b) Naïve Bayes, (c) SVM and (d) CART. Each plot shows comparison among MRmMC, mRMR and MIFS methods.

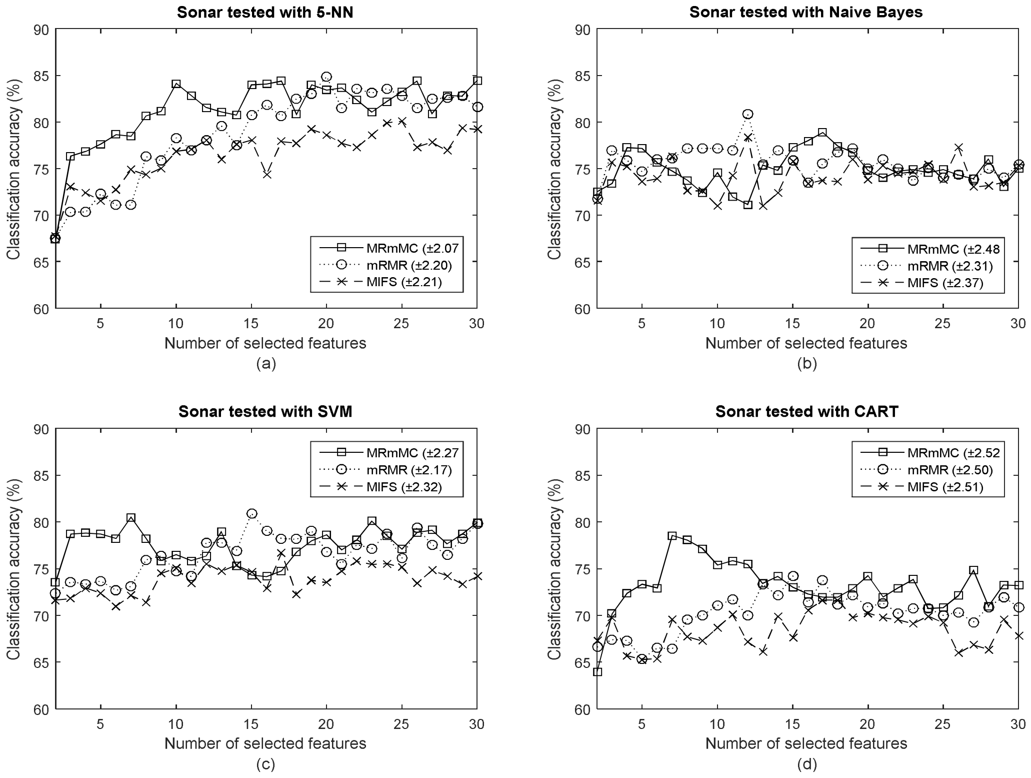

Figure 3.7: Classification results for Sonar dataset over different number of selected features, tested with four classifiers: (a) 5-NN, (b) Naïve Bayes, (c) SVM and (d) CART. Each plot shows comparison among MRmMC, mRMR and MIFS methods.

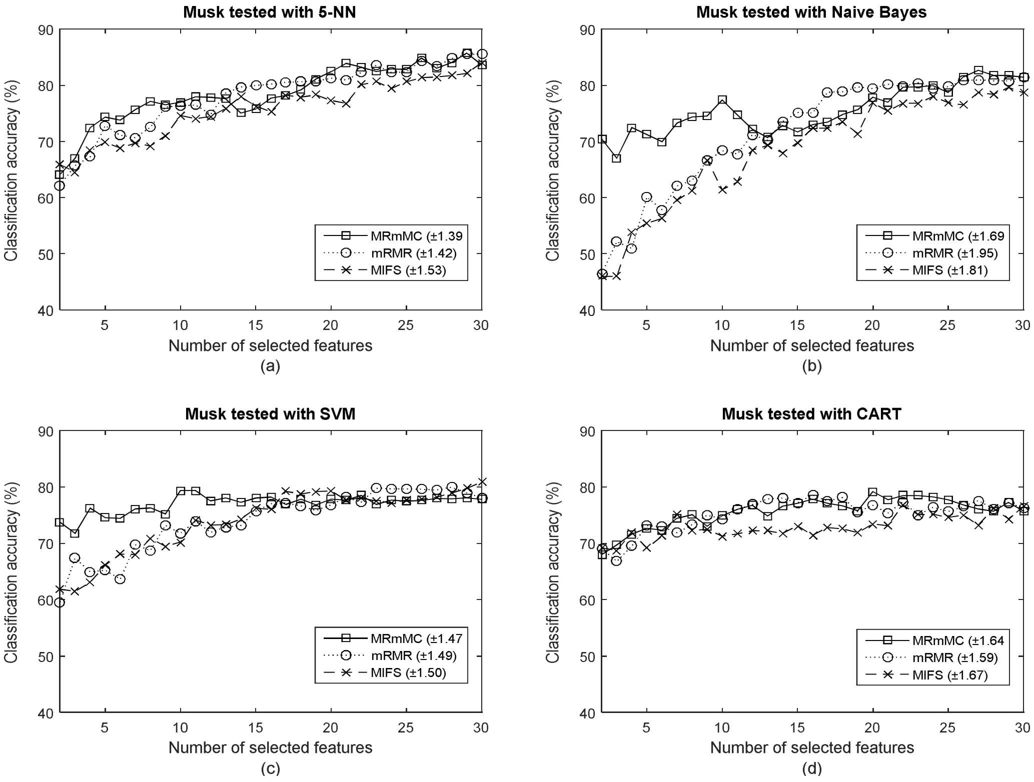

Figure 3.8: Classification results for Musk dataset over different number of selected features, tested with four classifiers: (a) 5-NN, (b) Naïve Bayes, (c) SVM and (d) CART. Each plot shows comparison among MRmMC, mRMR and MIFS methods.

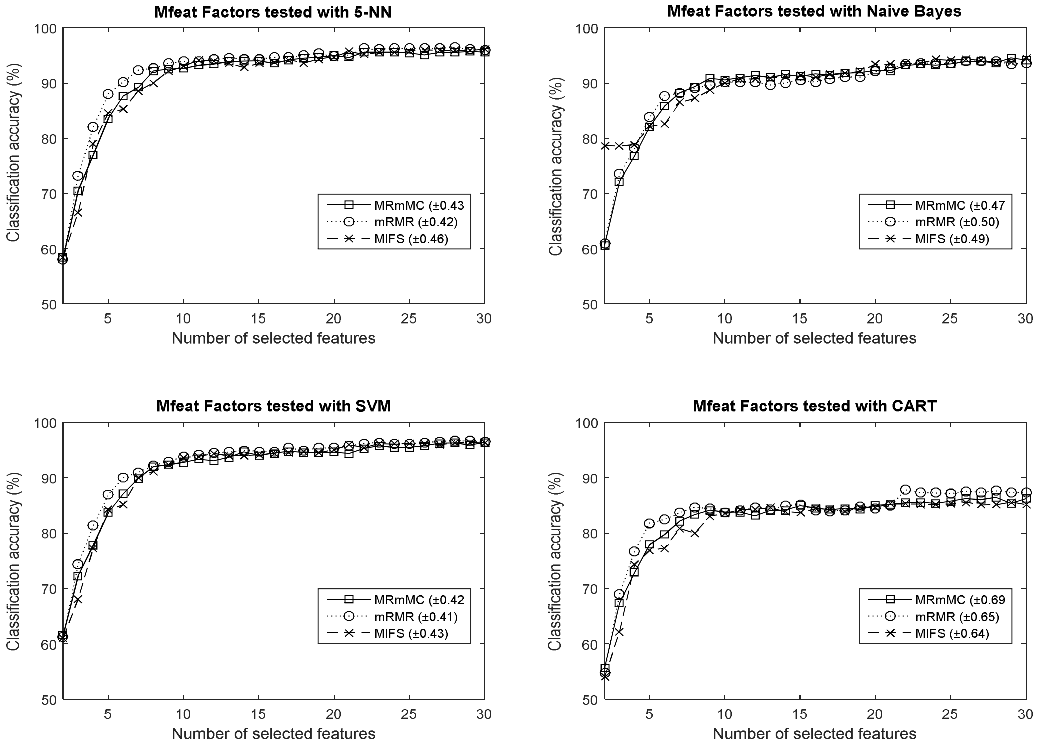

Figure 3.9: Classification results for Mfeat Factors dataset over different number of selected features, tested with four classifiers: (a) 5-NN, (b) Naïve Bayes, (c) SVM and (d) CART. Each plot shows comparison among MRmMC, mRMR and MIFS methods.

Table 3.2 and Table 3.3 summarize the mean of the average classification accuracies based on a number of first selected features. The results presented in rows with m = 5, 10, 15, and 30 provide the average classification accuracies of the selected features from 2 to $n_f = \min(m, M)$, respectively, where M is the number of original features. As suggested in Sotoca & Pla (2010), the four ranges of the number of selected features in our study here are representative as these choices cover the approximate transitory period where the classification accuracy becomes stable for most of the datasets (see Figure 3.2 until Figure 3.9). A one-tailed two-sample z-test was conducted for each case of the m values in order to evaluate the alternative hypothesis (H_1) that "the mean accuracy of the proposed method is greater than the mean accuracy of the compared method". The recorded p-value is the probability corresponding to the z-test. A significant difference is obtained to support the hypothesis if p is lower than 0.05 (5% significance level). Meanwhile, if p is greater than 0.95 then it can be concluded that the compared method outperforms the proposed method. For ease of viewing, results in the p-value columns are marked with the symbol "∗" and "■" to indicate that the MRmMC method is statically superior or inferior to the compared method, respectively. The p-value columns which are not highlighted by any symbol indicate that the two methods are comparable.

From Table 3.2 and Table 3.3, it can be observed that the MRmMC method generally provides either better or comparable classification accuracy in comparison with the other two methods for all classifiers when fewer features (e.g. 2 to 15 features) are used to represent all the candidate features, except in Vowel and Mfeat Factors. The performance of MRmMC is not as good as mRMR for the Vowel dataset with Nearest Neighbour, Naïve Bayes and SVM classifiers but is comparable to mRMR with CART classifier. Furthermore, MRmMC is only slightly inferior to the MIFS method for the Vowel dataset with Nearest Neighbour classifier.

Considering each classifier used, the MRmMC method is only inferior to either mRMR or MIFS for the Mfeat Factors dataset. Specifically, the MRmMC method shows slightly lower performance than the MIFS method with Naive Bayes classifier yet comparable/better performance with the other three classifiers, while conversely, MRmMC produces comparable performance with the mRMR with Naive Bayes classifier but slightly lower performance with the other three classifiers.

Table 3.2: A comparison of the average classification accuracy based on the first *m* selected features.

	Glass					**Magic Gamma**				
	MRmMC	mRMR		MIFS		MRmMC	mRMR		MIFS	
5-NN	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	62.38	62.42	0.51	58.65	0.01 *	80.38	77.61	0.00 *	77.22	0.00 *
m = 10	64.28	64.68	0.60	62.25	0.10	81.21	79.91	0.00 *	79.91	0.00 *
N Bayes	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	53.87	48.73	0.00 *	45.20	0.00 *	76.96	77.22	0.96 ▪	77.09	0.79
m = 10	54.53	54.40	0.47	51.55	0.05	76.55	76.85	0.98 ▪	76.91	0.99 ▪
SVM	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	59.13	60.79	0.87	54.22	0.00 *	78.71	74.55	0.00 *	74.82	0.00 *
m = 10	61.72	62.28	0.64	57.04	0.00 *	78.93	76.63	0.00 *	76.60	0.00 *
CART	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	60.36	59.92	0.40	56.35	0.01 *	76.70	73.64	0.00 *	73.34	0.00 *
m = 10	63.06	62.5	0.38	62.17	0.30	78.50	77.08	0.00 *	77.02	0.00 *

	Vowel					**Statlog**				
	MRmMC	mRMR		MIFS		MRmMC	mRMR		MIFS	
5-NN	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	73.6	76.32	1.00 ▪	76.45	1.00 ▪	54.69	50.57	0.00 *	51.34	0.00 *
m = 10	82.66	84.01	0.98 ▪	84.05	0.98 ▪	61.99	59.06	0.00 *	58.97	0.00 *
m = 15	-	-	-	-	-	64.79	62.75	0.01 *	62.84	0.01 *
m = 30	-	-	-	-	-	65.99	64.31	0.02 *	64.42	0.03 *
N Bayes	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	59.67	61.03	0.96 ▪	59.73	0.53	53.88	45.06	0.00 *	45.55	0.00 *
m = 10	65.83	67.24	0.96 ▪	66	0.58	59.20	52.84	0.00 *	52.21	0.00 *
m = 15	-	-	-	-		59.99	55.51	0.00 *	54.61	0.00 *
m = 30	-	-	-	-		60.08	56.57	0.00 *	55.77	0.00 *
SVM	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	59.34	61.83	1.00 ▪	60.53	0.94	50.7	46.54	0.00 *	47.37	0.00 *
m = 10	67.23	69.00	0.99 ▪	68.23	0.90	60.51	57.16	0.00 *	58.25	0.00 *
m = 15	-	-	-	-	-	64.93	63.67	0.06	65.20	0.63
m = 30	-	-	-	-	-	67.2	66.48	0.18	67.71	0.74
CART	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	65.35	66.56	0.92	65.84	0.72	53.16	52.78	0.34	53.77	0.75
m = 10	69.93	70.45	0.72	70.25	0.65	61.62	60.21	0.06	61.30	0.36
m = 15	-	-	-	-	-	64.61	63.60	0.13	64.25	0.34
m = 30	-	-	-	-	-	65.67	64.74	0.15	65.21	0.30

Table 3.3: A comparison of the average classification accuracy based on the first *m* selected features.

	Mfeat Zernike					**Sonar**				
	MRmMC	mRMR		MIFS		MRmMC	mRMR		MIFS	
5-NN	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	53.06	53.66	0.90	53.64	0.88	74.55	70.13	0.00 *	71.16	0.02 *
m = 10	64.43	64.46	0.53	62.74	0.00 *	77.92	72.56	0.00 *	73.15	0.00 *
m = 15	69.15	69.42	0.73	67.98	0.00 *	79.39	74.7	0.00 *	74.65	0.00 *
m = 30	75.05	74.78	0.25	74.70	0.19	81.24	78.76	0.05	76.45	0.00 *
N Bayes	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	55.96	55.58	0.24	55.54	0.20	75.08	74.81	0.43	74.00	0.27
m = 10	63.62	62.52	0.02 *	61.55	0.00 *	74.59	75.87	0.78	73.59	0.28
m = 15	66.28	65.57	0.08	64.77	0.00 *	74.41	76.35	0.88	73.86	0.37
m = 30	69.5	68.24	0.00 *	69.30	0.34	74.93	75.62	0.66	74.15	0.33
SVM	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	56.4	57.08	0.88	56.82	0.78	77.44	73.23	0.01 *	72.18	0.00 *
m = 10	65.63	66.24	0.88	64.51	0.01 *	77.67	73.97	0.01 *	72.52	0.00 *
m = 15	69.81	71.08	1.00 ■	68.97	0.04 *	77.12	75.23	0.12	73.31	0.01 *
m = 30	75.66	76.31	0.94	75.89	0.70	77.48	76.58	0.29	73.86	0.01 *
CART	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	49.54	49.47	0.45	49.45	0.44	69.96	66.67	0.04 *	67.01	0.07
m = 10	56.83	57.00	0.62	55.51	0.01 *	73.54	67.81	0.00 *	67.4	0.00 *
m = 15	59.53	60.40	0.94	58.46	0.03 *	73.84	69.4	0.01 *	67.68	0.00 *
m = 30	63.37	63.71	0.73	62.27	0.02 *	73.16	70.25	0.05	68.46	0.00 *

	Musk					**Mfeat Factors**				
	MRmMC	mRMR		MIFS		MRmMC	mRMR		MIFS	
5-NN	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	69.49	66.98	0.02 *	67.18	0.02 *	72.36	75.33	1.00 ■	72.13	0.32
m = 10	73.12	70.52	0.01 *	69.12	0.00 *	82.63	84.90	1.00 ■	81.95	0.05
m = 15	74.45	73.16	0.12	71.48	0.00 *	86.59	88.25	1.00 ■	86.11	0.10
m = 30	78.53	78.02	0.31	75.72	0.00 *	90.98	92.10	1.00 ■	90.82	0.31
N Bayes	Acc.	Acc.	*p*-value	Acc.	*p*-value	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	70.3	52.41	0.00 *	50.31	0.00 *	72.91	74.09	0.99 ■	79.56	1.00 ■
m = 10	72.3	58.61	0.00 *	56.26	0.00 *	81.83	82.35	0.90	83.69	1.00 ■
m = 15	72.35	63.24	0.00 *	60.33	0.00 *	85.18	85.11	0.43	86.31	1.00 ■
m = 30	75.58	71.78	0.00 *	68.51	0.00 *	89.22	89.05	0.31	89.92	0.98 ■
SVM	Acc.	Acc.	*p*-value	Acc.	Acc.	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	74.09	64.29	0.00 *	63.14	0.00 *	73.85	75.96	1.00 ■	72.69	0.01 *
m = 10	75.31	67.14	0.00 *	66.58	0.00 *	83.28	84.86	1.00 ■	82.57	0.04 *
m = 15	76.29	69.42	0.00 *	69.31	0.00 *	87.02	88.33	1.00 ■	86.68	0.18
m = 30	77.01	74.00	0.00 *	74.02	0.00 *	91.32	92.26	1.00 ■	91.29	0.46
CART	Acc.	Acc.	*p*-value	Acc.	Acc.	Acc.	Acc.	*p*-value	Acc.	*p*-value
m = 5	70.51	69.64	0.22	69.75	0.25	68.45	70.60	1.00 ■	66.84	0.00 *
m = 10	72.43	71.78	0.29	71.27	0.16	76.34	77.93	1.00 ■	74.68	0.00 *
m = 15	73.80	73.72	0.47	71.61	0.03 *	79.08	80.33	0.99 ■	78.05	0.02 *
m = 30	75.59	75.25	0.39	72.93	0.01 *	82.32	83.37	0.99 ■	81.64	0.08

Table 3.4 and Table 3.5 present the performance of MRmMC, mRMR and MIFS methods, generated by using the least number of selected features, m_{least}, with which a classification accuracy more than or close to that obtain by using the complete dataset (with no more than 5% difference). Results from Table 3.4 and Table 3.5 are further summarized in Table 3.6 with an intention to specifically demonstrate the capability of the MRmMC method in representing the full feature set. The win/tie/loss scores reported in Table 3.6 represent the number of benchmark datasets for which the MRmMC method gives lower/equal/higher number of selected features in comparison to other methods.

As can be seen from Table 3.6, the MRmMC method performs better than the MIFS for all four classifiers. It performs better for two out of four classifiers and shows comparable performance for the fourth classifier (CART) when compared to the mRMR method but does not perform well with SVM classifier. It can also be noticed that MRmMC gives outstanding performance with Nearest Neighbour and Naïve Bayes classifiers. Based on the average results given in the last row of Table 3.6, it can be concluded that the MRmMC method is the winner in overall when only a small number of features are required to represent the full feature set.

Table 3.4: The least number of selected features, m_{least}, by MRmMC, mRMR and MIFS methods that gives classification accuracy close to (at most 5% less than the full set accuracy) or better than the full feature set. The symbol "•" (or "□") denotes the proposed method has lower (or larger) value of m_{least} than the compared method. Results are based on Glass, Magic Gamma, Vowel and Statlog datasets.

	Glass			**Magic Gamma**		
5-NN	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	64.52 ± 2.61	3	65.16 ± 1.97	83.72 ± 0.16	2	79.46 ± 0.18
mRMR	64.52 ± 1.96	3	65.32 ± 1.86	83.76 ± 0.20	4 •	79.56 ± 0.18
MIFS	66.43 ± 2.27	3	62.30 ± 2.23	83.76 ± 0.19	5 •	79.46 ± 0.21
N Bayes	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	61.67 ± 2.49	3	65.87 ± 2.44	76.13 ± 0.28	2	77.69 ± 0.23
mRMR	60.48 ± 2.61	6 •	57.94 ± 2.66	76.22 ± 0.18	2	76.46 ± 0.15
MIFS	61.59 ± 2.31	7 •	58.17 ± 2.59	76.27 ± 0.21	2	76.32 ± 0.24
SVM	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	63.17 ± 1.98	3	61.27 ± 2.46	79.16 ± 0.22	2	78.34 ± 0.20
mRMR	63.65 ± 2.35	3	65.87 ± 1.59	78.98 ± 0.14	3 •	74.40 ± 0.24
MIFS	64.21 ± 2.03	8 •	62.78 ± 2.53	79.06 ± 0.22	3 •	74.36 ± 0.24
CART	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	66.35 ± 2.52	3	63.10 ± 2.36	81.84 ± 0.22	4	77.41 ± 0.29
mRMR	66.35 ± 2.30	3	64.84 ± 2.22	81.64 ± 0.21	6 •	77.84 ± 0.22
MIFS	68.73 ± 2.41	5 •	66.27 ± 2.45	81.95 ± 0.32	7 •	78.41 ± 0.29

	Vowel			**Statlog**		
5-NN	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	91.55 ± 0.64	6	87.12 ± 0.82	71.78 ± 0.95	6	67.34 ± 1.04
mRMR	91.73 ± 0.92	6	89.09 ± 0.75	72.13 ± 0.97	9 •	68.93 ± 1.19
MIFS	91.45 ± 0.89	6	87.29 ± 1.00	71.87 ± 1.23	11 •	69.90 ± 1.12
Naïve Bayes	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	73.30 ± 1.19	7	72.73 ± 1.01	60.61 ± 1.25	5	59.03 ± 1.35
mRMR	73.33 ± 1.03	6 □	69.87 ± 1.13	61.44 ± 1.24	7 •	60.06 ± 1.32
MIFS	73.13 ± 1.28	7	71.06 ± 1.28	60.34 ± 1.38	6 •	57.04 ± 1.23
SVM	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	77.81 ± 1.12	8	73.23 ± 1.21	79.59 ± 0.92	16	76.11 ± 0.77
mRMR	78.64 ± 1.18	8	75.57 ± 1.08	79.51 ± 0.89	13 □	76.00 ± 1.02
MIFS	78.42 ± 0.83	8	75.00 ± 1.01	79.57 ± 0.93	12 □	77.57 ± 0.97
CART	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	74.07 ± 1.23	5	71.41 ± 1.11	70.75 ± 0.97	7	68.90 ± 1.43
mRMR	74.75 ± 1.36	4 □	70.42 ± 1.11	70.37 ± 1.14	7	65.64 ± 1.31
MIFS	74.58 ± 1.19	4 □	70.37 ± 1.08	69.57 ± 1.08	5 □	65.03 ± 1.19

Table 3.5: The least number of selected features, m_{least}, by MRmMC, mRMR and MIFS methods that gives classification accuracy close to (at most 5% less than the full set accuracy) or better than the full feature set. The symbol "•" (or "□") denotes the proposed method has lower (or larger) value of m_{least} than the compared method. Results are based on Mfeat Zernike, Sonar, Musk and Mfeat Factors datasets.

	Mfeat Zernike			**Sonar**		
5-NN	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	80.61 ± 0.48	9	77.03 ± 0.52	78.13 ± 1.80	3	76.34 ± 2.17
mRMR	80.60 ± 0.54	9	77.20 ± 0.65	79.43 ± 1.92	8 •	76.26 ± 1.96
MIFS	80.58 ± 0.49	12 •	75.94 ± 0.60	77.89 ± 2.56	3	73.01 ± 1.74
Naïve Bayes	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	72.33 ± 0.70	6	67.58 ± 0.51	75.61 ± 2.59	2	72.52 ± 2.55
mRMR	72.43 ± 0.68	8 •	70.25 ± 0.72	75.12 ± 2.42	2	71.79 ± 2.25
MIFS	72.58 ± 0.70	8 •	68.69 ± 0.54	76.67 ± 1.41	3 •	75.69 ± 2.66
SVM	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	83.01 ± 0.57	14	78.17 ± 0.72	79.76 ± 2.25	3	78.70 ± 2.59
mRMR	82.53 ± 0.41	9 □	77.64 ± 0.52	76.18 ± 2.47	2 □	72.36 ± 2.36
MIFS	82.47 ± 0.45	15 •	78.38 ± 0.66	77.48 ± 1.86	4 •	72.93 ± 1.87
CART	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	66.58 ± 0.82	8	63.19 ± 0.67	73.01 ± 1.79	3	70.16 ± 2.82
mRMR	66.09 ± 0.64	8	63.74 ± 0.80	72.28 ± 2.30	3	67.40 ± 2.90
MIFS	66.68 ± 0.85	8	62.20 ± 0.81	73.66 ± 2.25	3	69.76 ± 3.06
	Musk			**Mfeat Factors**		
5-NN	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	88.49 ± 0.96	21	83.89 ± 0.91	96.47 ± 0.26	8	92.20 ± 0.50
mRMR	88.21 ± 1.21	23 •	83.54 ± 1.23	96.55 ± 0.24	7 □	92.34 ± 0.37
MIFS	87.37 ± 1.14	30 •	84.00 ± 1.41	96.63 ± 0.30	9 •	92.17 ± 0.51
Naïve Bayes	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	82.81 ± 1.63	20	77.88 ± 2.37	93.87 ± 0.39	8	89.34 ± 0.64
mRMR	82.14 ± 1.08	17 □	78.76 ± 2.19	94.08 ± 0.39	9 •	89.59 ± 0.38
MIFS	80.91 ± 1.50	20	76.86 ± 1.59	93.87 ± 0.32	10 •	90.03 ± 0.47
SVM	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	85.68 ± 0.99	40	81.47 ± 1.22	97.46 ± 0.25	10	92.79 ± 0.51
mRMR	85.05 ± 1.67	40	80.28 ± 1.61	97.62 ± 0.28	9 □	92.97 ± 0.48
MIFS	85.05 ± 1.27	30 □	80.88 ± 1.20	97.74 ± 0.27	10	93.68 ± 0.50
CART	Full set accuracy	m_{least}	Subset Accuracy	Full set accuracy	m_{least}	Subset Accuracy
MRmMC	77.09 ± 1.63	5	72.67 ± 1.17	88.38 ± 0.55	9	84.17 ± 0.73
mRMR	78.74 ± 1.76	9 •	75.02 ± 1.37	88.01 ± 0.57	7 □	83.67 ± 0.67
MIFS	77.30 ± 1.97	7 •	75.12 ± 1.69	87.88 ± 0.58	9	83.09 ± 0.59

Table 3.6: A comparison of win/tie/loss counts of the MRmMC method against the other methods. The counts are based on the results presented in Table 3.4 and Table 3.5.

Win/tie/lose	mRMR	MIFS
5-NN	4 / 3 / 1	5 / 3 / 0
Naïve Bayes	4 / 2 / 2	5 / 3 / 0
SVM	1 / 3 / 4	4 / 2 / 2
CART	2 / 4 / 2	3 / 3 / 2
Average	2.75 / 3 / 2.25	4.25 2.75 / 1

3.9 Summary

The MRmMC method uses a hill-climbing search structure with a straightforward measurement criterion that makes it simple and easy to implement. It is a filter feature selection method as it uses no specific classification scheme in the selection process, and therefore it works well with popular classifiers such as k-NN, naïve Bayes, SVM and CART.

Although the method may not always find the optimal subset as the search is non-exhaustive, it is shown from the experimental and numerical case studies that the method is competent for feature selection and dimensionality reduction.

As mentioned in Section 3.5, MRmMC possesses several attractive properties, one of which is that there is no need to pre-specify control parameters as required in MIFS methods, and another important one is that it is relatively easier to implement.

Chapter 4

Unsupervised feature selection based on local largest structure

4.1 Introduction

This chapter presents the second feature selection method to be proposed in which information of the local data structure is mainly utilised. Particularly, the special characteristics possesses by local largest structure (LLS) of locality preserving projection is employed in the new method for detecting significant features in an unsupervised setting. Being incline to a simple yet effective approach, a *sequential orthogonal search* (SOS) is used as the feature selection strategy. The method is thus referred to as *sequential orthogonal search for local largest structure* (SOS-LLS).

The remaining sections of this chapter is outlined as follows. In Section 4.2, a review of local structure preservation techniques is given. The proposed feature selection method is described in Section 4.3. Next, the experimental setup and the comparative results are given in Section 4.4, along with discussion about the performance of the proposed method. Finally, the chapter is ended with a summary in Section 4.5.

4.2 Related Work

4.2.1 Locality Preserving Projection

Locality preserving projection (LPP) emerged in response to the need for an alternative linear feature transformation approach that gives low dimensional space by optimally preserves local information of a dataset. Such transformations are obtained by constructing a nearest neighbour graph in which local geometric structure information is kept. The locality aspect is being considered in a sense that two data points are more likely connected to the same subject matter if they are close together.

4.2.1.1 LPP procedure

The main procedure to find LPP is briefly summarized in the following. Let a set of N data points in the original measurement space be $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ in $\mathbb{R}^M$. The LPP approach attempts to find a transformation matrix $\mathbf{A}$ that projects the N data points to a set of new points $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$, while preserving local neighbourhood structure of the data. Being regarded as representatives of the original data points, these new points are referred to as the locality preserving projections (LPP). They are obtained based on a mapping function $\mathbf{y}_i = \mathbf{A}^{\mathrm{T}}\mathbf{x}_i$ which lie in a reduced feature subspace $\mathbb{R}^m$ where normally $m << M$. In general, there are three main steps involved in finding the LPP:

Step 1: **Build an adjacency graph with N nodes**

Consider a graph G with M nodes in which the i-th node corresponds to a data point $\mathbf{x}_i$. An adjacency graph is built in a way where edges are drawn between nodes that are closest (adjacent) in distance based on nearest neighbours principle. One of the following nearest neighbour distance rules shall be used:

(1) k-nearest neighbours. For every node i in G, draw an edge to link the node with each of its k-nearest neighbour nodes. In this rule $k \in \mathbb{N}$ and k is typically a small value.

(2) ε-neigbours. For every node i in G, draw an edge to link the node with each of its nearest neighbour nodes j that satisfies $\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 < \varepsilon$. This radius-based neighbours distance rule is a good choice when a dataset is not uniformly distributed.

In both rules the Euclidean distance function can be used for simplicity.

Step 2: **Give weightage to the edges**

Based on the adjacency graph G, give weightage to the identified edges so that the neighbourhood relationship between data points can be expressed into a matrix $\mathbf{W}$. This step technically allows the local geometrical structure of the original measurement space being presented by the weight matrix $\mathbf{W}$. The elements w_{ij} of the matrix $\mathbf{W}$ particularly define the weights or the degree of closeness between nodes i and j. Two choices that are widely used for weighting the edges are as follows:

(1) Binary weighting. If there is an edge connecting nodes i and j, then $w_{ij} = 1$. Otherwise, if there is no edge between them then $w_{ij} = 0$. This is a simple weighting choice that does not involve any pre-set parameter.

(2) Heat kernel weighting. If there is an edge connecting nodes i and j, then

$$w_{ij} = \exp\left(-\frac{\left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2}{t} \right) \tag{4.1}$$

where $t > 0$. Otherwise, $w_{ij} = 0$. This type of weighting is more specific to the data structure compared to binary weighting as it gives preference to neighbouring nodes that are closer.

Step 3: **Find the projections**

Given the data matrix $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \cdots \mathbf{x}_N\ ]$ whose i-th column constitutes the point $\mathbf{x}_i$. Find the eigenvectors and their associated eigenvalues for the following generalized eigenvector problem:

$$\mathbf{XLX}^{\mathrm{T}}\mathbf{a} = \lambda \mathbf{XDX}^{\mathrm{T}}\mathbf{a} \tag{4.2}$$

where $\mathbf{D}$ is a diagonal matrix whose main diagonal elements D_{ii} are the column sums (or row sums since $\mathbf{W}$ is symmetric) of $\mathbf{W}$, that is, $D_{ii} = \sum_j W_{ij}$. The larger the D_{ii} value, the more the impact or local density of the node i. Meanwhile, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix whose role is to measure the extent to which every node differs from its neighbour nodes in the graph. Suppose that the solution for (4.2) is a series of significant eigenvectors denoted by $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$, correspond to the first m smallest eigenvalues. Then the following eigenmap can be obtained:

$$\mathbf{x}_i \rightarrow \mathbf{y}_i = \mathbf{A}^{\mathrm{T}}\mathbf{x}_i \tag{4.3}$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m]$ is an $M \times m$ matrix. The resulting map $\mathbf{y}_i$ is the so-called LPP projection which is a vector of *m*-dimensional.

4.2.1.2 LPP connection with Laplacian eigenmap

Generally, the aim of local manifold structure preservation approach for feature extraction is to map close points in high dimensional feature space in a way so that their mappings are also close to each other in the associated low dimensional representation. Let $\mathbf{y} = [y_1, y_2, \ldots, y_N]^{\mathrm{T}}$ denote the vector of such a map. According to Laplacian eigenmap (Belkin & Niyogi, 2002), an optimal map is obtained based on the following objective function:

$$\min \sum_{i,j=1}^{N} (y_i - y_j)^2 w_{ij} \tag{4.4}$$

where w_{ij} is the element of the weight matrix $\mathbf{W}$ as defined previously. A heavy penalty is imposed on the objective function via the weight w_{ij} when close points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the measurement space are mapped far apart in the transformed space. Hence, the objective function ensures that if two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close, then their mappings $\mathbf{y}_i$ and $\mathbf{y}_j$ will be set close too. Applying some simple algebraic operation reduces the objective function (4.4) to

$$\min \mathbf{y}^{\mathrm{T}} \mathbf{L} \mathbf{y} \tag{4.5}$$

which is subject to constraint $\mathbf{y}^{\mathrm{T}} \mathbf{D} \mathbf{y} = 1$. This constraint is important in order to avoid arbitrary scale in the mapping.

The vector $\mathbf{y}$ that meets this objective function can be obtained by solving the generalized eigenvector problem

$$\mathbf{L}\mathbf{y} = \lambda \mathbf{D}\mathbf{y} \tag{4.6}$$

where $\mathbf{y}$ is associated with the minimum eigenvalue.

Apparently, Laplacian eigenmap is a nonlinear approach. It was then adapted to provide a linear variant called LPP (He & Niyogi, 2004). In LPP, each $\mathbf{x}_i$ is intended to be linearly mapped by a transformation vector $\mathbf{a}$, such that

$$y_i = \mathbf{a}^{\mathrm{T}}\mathbf{x}_i . \tag{4.7}$$

This map is not only defined on original data points but also on any new test point.

Substituting (4.7) into (4.4) yields

$$\min \sum_{i,j=1}^{N} (\mathbf{a}^{\mathrm{T}}\mathbf{x}_i - \mathbf{a}^{\mathrm{T}}\mathbf{x}_j)^2 w_{ij} \tag{4.8}$$

which is the objective function of LPP. With this connection in place, LPP can be viewed as a linear approximation to the nonlinear Laplacian eigenmap. By some algebraic manipulation, the objective function (4.8) turns out to be

$$\min \mathbf{a}^{\mathrm{T}}\mathbf{XLX}^{\mathrm{T}}\mathbf{a} . \tag{4.9}$$

The minimizing problem of the objective function (4.9) can be formulated into a generalized eigenvector problem

$$\mathbf{XLX}^{\mathrm{T}}\mathbf{a} = \lambda \mathbf{XDX}^{\mathrm{T}}\mathbf{a} \tag{4.10}$$

under the constraint $\mathbf{a}^{\mathrm{T}}\mathbf{XDX}^{\mathrm{T}}\mathbf{a} = 1$, analogous to the constraint specified for Laplacian eigenmap objective function. In this formulation, the transformation vector $\mathbf{a}$ that satisfies the LPP objective function is provided by the minimum eigenvalue obtained from the generalized eigenvector problem.

4.2.2 Laplacian Score

Laplacian score is an unsupervised feature selection method, fundamentally based on the ideas of Laplacian eigenmap and LPP. It selects features with strong locality preserving power which contribute the most to the underlying local manifold structure of the data. This is done specifically through selection of features that respect geometrical structure of a pre-determined adjacency graph G for the data, represented by its resultant weight matrix $\mathbf{W}$ as defined for the LPP method.

Let $\mathbf{f}_r = [f_{r1}, f_{r2}, \ldots, f_{rN}]$ be the *r*-th feature vector formed by N observations. In order to reflect the targeted data structure, the criterion for choosing significant features of the Laplacian score is set to minimize the following objective function:

$$L_r = \frac{\sum_{i,j=1}^{N} (f_{ri} - f_{rj})^2 w_{ij}}{\mathrm{Var}(\mathbf{f}_r)} \tag{4.11}$$

where w_{ij} is the element of the weight matrix $\mathbf{W}$, while $\mathrm{Var}(\mathbf{f}_r)$ denotes the estimated variance of the *r*-th feature. Based on this objective function, feature selection is achieved by choosing the top ranked features with the lowest scores. As a mean to minimize the objective function (4.11), it obviously requires the numerator term $\sum_{i,j=1}^{N} (f_{ri} - f_{rj})^2 w_{ij}$ to be minimized meanwhile the denominator term $\mathrm{Var}(\mathbf{f}_r)$ should be maximized. Minimizing the term $\sum_{i,j=1}^{N} (f_{ri} - f_{rj})^2 w_{ij}$ will lead to selection of features that are consistent with the graph structure G or in other words features with strong locality preserving power. This is based on a key assumption that in order for a feature to be significant, any two data points defined specifically on this feature should be close to each other as they are in the original feature space. By maximizing the term $\mathrm{Var}(\mathbf{f}_r)$ Laplacian score does not only intend to prefer features with strong locality preserving power, but also more representative ability.

4.2.3 Multi-Cluster Feature Selection

Feature selection criterion of Laplacian Score method evaluates every candidate feature individually. This approach does not take into account the correlation between different features, thus ignores feature redundancy and makes it prone to suboptimal results. Even though a feature has high individual predictive power, it should not be selected if it concurrently has high correlation with preceding selected features since such a feature contributes no extra information. In the event that a feature has low individual predictive power, it may be of high predictive power when combined with the already selected features as together they form some relationship, which, if true, it should be considered for selection. Thus, it is crucial to evaluate feature importance jointly rather than individually.

Multi-Cluster Feature Selection (MCFS) considers this necessity by using spectral analysis integrated with L1-regularized regression model. In order to capture the multi-cluster structure of the data, MCFS exploits the top ranked eigenvectors of the generalized eigenvector problem for the Laplacian Eigenmap as defined in (4.6). Because this approach utilizes local discriminative information, it has thus considered the local manifold structure naturally.

Let $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K$ be the K eigenvectors obtained by solving the generalized eigenvector problem (4.6). A subset of significant features can be identified based on the following objective function as:

$$\begin{aligned} &\min \left\| \mathbf{y}_k - \mathbf{X}^{\mathrm{T}} \mathbf{a}_k \right\|^2 \\ &\text{subject to } \left| \mathbf{a}_k \right| \leq \gamma \end{aligned} \tag{4.12}$$

where $\mathbf{a}_k$ is an M-dimensional vector that contains the coefficients for the linear combination of different features in approximating the vector $\mathbf{y}_k$, $|\mathbf{a}_k|$ denotes the number of nonzero coefficients (entries) of $\mathbf{a}_k$ and γ is a pre-determined threshold. The objective function (4.12) is essentially the L1-regularized regression problem in which $\mathbf{a}_k$ is the optimal solution corresponds to $\mathbf{y}_k$. Provided that a dataset containing K clusters, then K sparse coefficient vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_K$ can be determined to represent the eigenvectors $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K$ that are most representative for the clusters. Under this formulation, each $\mathbf{y}_k$ is expected to reflect the data distribution among different features as well as on the associated cluster. Every feature $\mathbf{f}_r$ will be given a score based on the highest coefficient value of $\mathbf{a}_k$ that correspond to $\mathbf{f}_r$ and significant features can be finally selected according to the top high-scored features of the ranking list.

4.2.4 Minimum-Maximum Laplacian Score (MMLS)

A variant of the Laplacian Score method called Minimum-Maximum Laplacian Score (MMLS) (Hu, et al., 2013) was introduced to gain more discriminative power in unsupervised setting by considering two different perspectives of local structure information: the within-locality information and the between-locality information. Following the intuition that separation between points of the same class should be as small as possible, the within-locality information needs to be minimized to identify this particular manifold data structure.

On the other hand, the between-locality information needs to be maximized to ensure points from different classes are well separated, and therefore increase the discriminative power of the selected feature subset. Integrating both goals into one gives rise to the following objective function to be minimized:

$$\mathrm{MMLS}_r = \frac{(1-\alpha)\sum_{i,j=1}^{N}(f_{ri}-f_{rj})^2 w_{a,ij} - \alpha\sum_{i,j=1}^{N}(f_{ri}-f_{rj})^2 w_{b,ij}}{\mathrm{Var}(\mathbf{f}_r)} \tag{4.13}$$

where $w_{a,ij}$ and $w_{b,ij}$ denote the entries of the within-locality weight matrix $\mathbf{W}_a$ and the between-locality weight matrix $\mathbf{W}_b$ respectively, while α is a pre-defined parameter that controls the trade-off between the two types of local structure information being considered.

The within-locality weight matrix $\mathbf{W}_a$ has all the same properties of the weight matrix defined for LPP and Laplacian Score which assigns a nonzero weight entry for any two points with the nearest relationship whereas the between-locality weight matrix $\mathbf{W}_b$ is a matrix whose entries are set contrast with reference to $\mathbf{W}_a$ which gives a nonzero weight entry for any two points without the nearest relationship. Like in the Laplacian Score, the variance of the *r*-th feature, $\mathrm{Var}(\mathbf{f}_r)$, is also considered as a part of the MMLS criterion. By minimizing the objective function (4.13) as a whole forcing $\sum_{i,j=1}^{N}(f_{ri}-f_{rj})^2 w_{a,ij}$ to give a minimum value whereas $\sum_{i,j=1}^{N}(f_{ri}-f_{rj})^2 w_{b,ij}$ and $\mathrm{Var}(\mathbf{f}_r)$ are of maximum value. The term $\sum_{i,j=1}^{N}(f_{ri}-f_{rj})^2 w_{a,ij}$ is essentially the same term contained in the objective function of the Laplacian Score and minimizing it here is specifically intended to preserve the close relationship of each data point with its neighbouring points on the *r*-th feature. Conversely, maximizing the term $\sum_{i,j=1}^{N}(f_{ri}-f_{rj})^2 w_{b,ij}$ is expected to preserve any non-neighbouring relationship on the *r*-th feature.

Even though MMLS method claims that it selects features based on local manifold structure preservation, the approach can be seen as an attempt to choose features that preserve data structure in a global sense because maximizing the between-locality information will basically retain geometric data structure of faraway points in the high dimensional space to the low dimensional space. The MMLS method brought a slight change to the Laplacian Score criterion by incorporating the between-locality information and it was found to be an effective strategy for feature selection. It is however, does not take into account the correlation between different features and hence suffers the same problem highlighted earlier for Laplacian Score.

4.3 The Proposed Algorithm for Feature Ranking and Selection

This section is primarily aims to simultaneously exploit the potential of local geometric structure for unsupervised feature selection and the power of LPP approach for classification. It seeks to presents a new feature selection method based on local largest structure (LLS) of locality preserving projection. The new feature selection method can be represented as a multiple linear regression problem in which the most significant feature map defined by LPP will be treated as a response variable, while all the original features will be treated as predictor variables. The key idea of the method is to select a subset of predictor variables that best represents the response variable. In other words, the objective is to select a subset of features that has the highest capability to represent the most significant feature extracted by LPP which carries the major information about the local largest structure. Under this feature selection framework, the method can be seen as an attempt to take advantages of both feature extraction and feature selection approaches. In the new method, significant features are selected one by one using a sequential search strategy (SOS).

Let the set $F = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ denotes a collected full dataset of M features where each $\mathbf{x}_i = [x_i(1), x_i(2), \ldots, x_i(N)]^{\mathrm{T}}$ is a feature vector composed by N observations. The objective of feature selection is to find the best feature subset $S_d = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_d\}$ that gives compact representation of the full feature set F where $\mathbf{z}_j \in F$ and d should be the least possible integer with $d << M$ if the measurement space is of high dimensionality. In this regard, every feature vector $\mathbf{x}_i$ can be satisfactorily represented using the selected feature subset S_d via some functional relationship which generally can be expressed as

$$\mathbf{x}_i = f_i(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_d) + \mathbf{e}_i \tag{4.14}$$

where f_i is a function that supposed to well describe the relationship between the feature $\mathbf{x}_i$ and the selected features $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_d$, while the term $\mathbf{e}_i$ denotes the estimation error. In (Wei & Billings, 2007), the relationship between the feature $\mathbf{x}_i$ and the selected feature subset is assumed to be linear which leads to the commonly used multiple regression model

$$\mathbf{x}_i = \sum_{k=1}^{d} \theta_{i,k} \mathbf{z}_k + \mathbf{e}_i \tag{4.15}$$

where $\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,d}$ are the regression coefficients that need to be estimated based on the observed dataset.

Referring to the embedding map obtained by LPP as specified in (4.3), it is straightforward that the *k*-th component of LPP for any observation $\mathbf{x}_r$ is given by

$$y_k(r) = \mathbf{a}_k^{\mathrm{T}} \mathbf{x}_r = \sum_{j=1}^{M} a_j^{(k)} x_{r,j} \tag{4.16}$$

for $r = 1, 2, \ldots, N.$ Hence, the overall *k*-th component of LPP is

$$\mathbf{y}_k = \begin{bmatrix} y_k(1) \\ y_k(2) \\ \vdots \\ y_k(N) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{M} a_j^{(k)} x_{1,j} \\ \sum_{j=1}^{M} a_j^{(k)} x_{2,j} \\ \vdots \\ \sum_{j=1}^{M} a_j^{(k)} x_{N,j} \end{bmatrix} = \mathbf{X}\mathbf{a}_k = \sum_{j=1}^{M} a_j^{(k)} \mathbf{x}_j \tag{4.17}$$

where $\mathbf{x}_j$ is the *j*-th column vector of $\mathbf{X}$, representing the *j*-th feature vector made up of N observations . Note that each newly generated component of LPP is derived by using a linear model that involves all the original features. Since some of the original features may be linearly correlated with the others, it is reasonable to exclude these redundant features from the candidate set as they give little or no additional information to the component map. Hence, feature selection is accompanied with some basic feature elimination performance.

As the *k*-th component $\mathbf{y}_k$ of LPP defined in (4.17) is fundamentally a feature vector formed by a linear combination of all original features, it also should be well represented using the selected feature subset S_d through a simple adaptation of model as in (4.15). The approximation of $\mathbf{y}_k$ is therefore as follows

$$\mathbf{y}_k = \sum_{j=1}^{d} \beta_j^{(k)} \mathbf{z}_j + \mathbf{e}_k \tag{4.18}$$

where the response variable $\mathbf{x}_i$ in (4.15) is replaced by $\mathbf{y}_k$ in (4.18). This approximation model leads to the idea that feature vectors which are significant in representing the LPP component must also be significant for building a reduced dimension representation of the original full feature set F .

Motivated by the above observations, this work introduces a feature selection method by defining the reference response variable in the multiple linear regression model as the first LPP component whereas the candidate predictors are chosen to be the original feature variables. Basically, this means that the information contained in the first LPP component is used to guide the feature selection.

The rationale of using only the first LPP component as the reference response variable is to keep the data locality because the eigenvector that generates the component is the one that encodes perhaps the most important graph information (Mohar, et al., 1991). This eigenvector which corresponds to the smallest non-zero eigenvalue of the corresponding Laplacian matrix for a graph is well known as the Fiedler vector, named after the seminal work of Fiedler (Fiedler, 1973; Fiedler, 1989). The eigenvalue associated to the Fiedler vector is called the "algebraic connectivity" of a graph due to its special relation with the structural properties of the graph – the vertex connectivity and the edge connectivity. If a new edge is inserted in between two weakly connected nodes, the value of the algebraic connectivity will show the greatest increase among the spectrum of a graph (Maas, 1987; Wang & Mieghem, 2008). In this sense, the eigenvalue associated with the Fiedler vector can be viewed as an indicator of the degree of graph connectivity. As the Fiedler vector represents data structure with the highest graph connectivity, the projection of the Fiedler vector is referred as a projection that preserves the largest structure of the data. Since LPP particularly preserves local data structure, the first LPP component is therefore can be viewed as holding the local largest structure of the data. As such, the first LPP component is the one that reflects the local largest structure preservation. This explains the term LLS being used in the name of the proposed method.

Each entry of the Fiedler vector represents a value for a graph node while the vector as a whole represents an optimal segmentation for the graph (Bertrand & Moonen, 2013; Perazzi, et al., 2015). Intuitively, one can consider the projection of the Fiedler vector as a tool to evaluate features for selection.

By using only one LPP component, model (4.18) can be rewritten as

$$\mathbf{y} = \sum_{j=1}^{d} \beta_j \mathbf{z}_j + \mathbf{e} \tag{4.19}$$

where $\mathbf{y}$ is the first LPP component generated by a Fiedler vector. This means that the first LPP component $\mathbf{y}$ should be well represented by the selected features $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_d$. Because $\mathbf{y}$ is an LPP component, the selected features should have a good ability in preserving local structure of the manifold. In particular, note that these features are selected based on local largest structure of LPP so as to capture the optimal local separation as the LPP component being considered is induced by a Fiedler vector.

Note that the linear model (4.19) can also be expressed in a compact matrix form as

$$\mathbf{y} = \mathbf{P}\boldsymbol{\beta} + \mathbf{e} \tag{4.20}$$

where $\mathbf{P} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_d]$ is a full column rank matrix. The matrix $\mathbf{P}$ can be decomposed into

$$\mathbf{P} = \mathbf{QR} \tag{4.21}$$

where $\mathbf{Q}$ is an $N \times d$ matrix comprises of orthogonal vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_d$ as its columns whereas $\mathbf{R}$ is an upper triangular matrix with unity diagonal elements of size $d \times d$. Substituting (4.21) into (4.20) along with some simple algebraic manipulation gives the following equivalent representation for (4.20):

$$\mathbf{y} = (\mathbf{PR}^{-1})(\mathbf{R}\boldsymbol{\beta}) + \mathbf{e} = \mathbf{Qg} + \mathbf{e} \tag{4.22}$$

where $\mathbf{g} = \mathbf{R}\boldsymbol{\beta} = [g_1 \ g_2 \ \cdots \ g_d]^{\mathrm{T}}$ with its elements g_j are the orthogonal coefficients. Utilizing the orthogonal property of $\mathbf{Q}$, the coefficient g_j can be computed in terms of $\mathbf{y}$ and $\mathbf{q}_j$ by:

$$g_j = (\mathbf{y}^{\mathrm{T}}\mathbf{q}_j)/(\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j)\,. \tag{4.23}$$

Based on (4.22), the total sum of squares (or total variation) for the LPP projection $\mathbf{y}$ is expressed by:

$$\mathbf{y}^{\mathrm{T}}\mathbf{y} = \sum_{j=1}^{d} g_j^2 \mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j + \mathbf{e}^{\mathrm{T}}\mathbf{e}\,. \tag{4.24}$$

Observe that the total variation consists of two general components: the variation due to relationship of $\mathbf{y}$ with $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_d$ (or, equivalently, $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_d$) which is $\sum_{j=1}^{d} g_j^2 \mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j$ and the variation due to residual error which is given by $\mathbf{e}^{\mathrm{T}}\mathbf{e}$. Hence, $g_j^2 \mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j$ is interpreted as the amount of contribution by the variable $\mathbf{q}_j$ to the total energy of the response variable, i.e., $||\mathbf{y}||^2$.

Applying the concept of *error reduction ratio* (ERR) described in Billings et al. (1989); Chen et al. (1989) and Billings (2013), here, the ERR associated with $\mathbf{q}_j$ or equivalently with $\mathbf{z}_j$ is defined as

$$\mathrm{ERR}[j] = \frac{g_j^2(\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j)}{\mathbf{y}^{\mathrm{T}}\mathbf{y}} = \frac{(\mathbf{y}^{\mathrm{T}}\mathbf{q}_j)^2}{(\mathbf{y}^{\mathrm{T}}\mathbf{y})(\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j)}\,. \tag{4.25}$$

This ratio serves as a measure to quantify the significance of a feature with higher ratio indicating greater contribution in representing the original feature set.

Following Wei & Billings (2007), the feature selection procedure can be fulfilled in a stepwise manner. Let the set $F = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ denotes a full dataset of M features. At the first step, determine

$$\mathrm{ERR}[j;1] = \frac{(\mathbf{y}^{\mathrm{T}}\mathbf{x}_j)^2}{(\mathbf{y}^{\mathrm{T}}\mathbf{y})(\mathbf{x}_j^{\mathrm{T}}\mathbf{x}_j)}; \quad j = 1, 2, \ldots, M \tag{4.26}$$

$$\ell_1 = \arg\max_{1\le j\le M}\left\{\mathrm{ERR}[j;1]\right\} \tag{4.27}$$

where ERR[j;1] denotes the error reduction ratio obtained by choosing $\mathbf{x}_j$ as the first significant feature. The first selected feature is then given by $\mathbf{z}_1 = \mathbf{x}_{\ell_1}$ and the associated orthogonal variable is then set as $\mathbf{q}_1 = \mathbf{z}_1$. Notice that the variable vector $\mathbf{q}_j$ in (4.25) is substituted with a feature vector $\mathbf{x}_j$ in (4.26). This direct replacement is permitted here because $\mathbf{q}_1 = \mathbf{z}_1 = \mathbf{x}_{\ell_1}$. The selection of $\mathbf{x}_{\ell_1}$ as the first significant feature means it is the feature that explains the variation in the overall features with the highest percentage among all candidate features.

Suppose that a subset S containing $(r-1)$ significant features $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{r-1}$ has been identified at the $(r-1)$th search step. The selected features $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{r-1}$ are then transformed to a new set of orthogonal vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{r-1}$. Now, assume the task is to include the rth significant feature $\mathbf{z}_r$ into S , and let $\mathbf{f}_j$ be a possible candidate feature to be considered where $\mathbf{f}_j \in F - S$. The rth orthogonal variable, $\mathbf{q}_{j,r}$ associated to $\mathbf{f}_j$ is computed by

$$\mathbf{q}_{j,r} = \mathbf{f}_j - \sum_{k=1}^{r-1} \frac{\mathbf{f}_j^{\mathrm{T}} \mathbf{q}_k}{\mathbf{q}_k^{\mathrm{T}} \mathbf{q}_k} \mathbf{q}_k . \tag{4.28}$$

Similar to the first step and based on the criterion defined by (4.25), the followings are determined in the rth step so that the rth significant feature can be identified:

$$\mathrm{ERR}[j; r] = \frac{(\mathbf{y}^{\mathrm{T}} \mathbf{q}_{j,r})^2}{(\mathbf{y}^{\mathrm{T}} \mathbf{y})(\mathbf{q}_{j,r}^{\mathrm{T}} \mathbf{q}_{j,r})} \tag{4.29}$$

$$\ell_r = \arg \max_{1 \le j \le M} \{\mathrm{ERR}[j; r]\}. \tag{4.30}$$

The rth significant feature $\mathbf{z}_r$ will be selected as the ℓ_r-th feature vector from the original feature set, that is $\mathbf{z}_r = \mathbf{f}_{\ell_r}$ and the corresponding orthogonal variable is therefore $\mathbf{q}_r = \mathbf{q}_{\ell_r, r}$.

Subsequent significant features can be selected in the same way, employing a *sequential orthogonal search* (SOS) strategy where features are selected in a stepwise manner, one by one, through an orthogonalization scheme as described above. At each step, a feature that contributes the most to the total variation in the response variable $\mathbf{y}$ with the highest value of ERR is selected. As $\mathbf{y}$ represents the locality preserving projection resulting from a Fiedler vector, in which the *local largest structure* (LLS) of a dataset lies, the selected features are thus expected to preserve the main information of the local geometric structure hold by the original data set. For simplicity of the discussion onwards, the newly introduced method will be referred to as SOS-LLS (sequential orthogonal search for local largest structure) approach. The pseudocode of the SOS-LLS is given in Figure 4.1.

```
Input:        $F = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$     // A complete dataset of $M$ features
Output:       $S$                                    // Subset of features
Initialize:   $L_1 = \{1, 2, \ldots, M\}$, $S = \{\}$
              $m$                                    // Number of features to be selected

Find $\mathbf{y}$                                    // The first LPP component as defined by (4.17)

for $j = 1$ to $M$
        $\mathrm{ERR}[j; 1] = \dfrac{(\mathbf{y}^{\mathrm{T}}\mathbf{x}_j)^2}{(\mathbf{y}^{\mathrm{T}}\mathbf{y})(\mathbf{x}_j^{\mathrm{T}}\mathbf{x}_j)}$;
end for
$\ell_1 = \arg\max_{j \in L_1}\{\mathrm{ERR}[j;1]\}$ such that $\ell_1 \in L_1$; $\mathbf{q}_1 = \mathbf{x}_{\ell_1}$; $\mathbf{z}_1 = \mathbf{x}_{\ell_1}$;
add $\mathbf{z}_1$ to $S$;

for $r = 2$ to $m$
        $L_r = L_{r-1} \setminus \{\ell_{r-1}\}$;
        for $j \in L_r$
                $\mathbf{q}_{j,r} = \mathbf{f}_j - \sum_{k=1}^{r-1} \dfrac{\mathbf{f}_k^{\mathrm{T}}\mathbf{q}_k}{\mathbf{q}_k^{\mathrm{T}}\mathbf{q}_k}$;
                $\mathrm{ERR}[j; r] = \dfrac{(\mathbf{y}^{\mathrm{T}}\mathbf{q}_{j,r})^2}{(\mathbf{y}^{\mathrm{T}}\mathbf{y})(\mathbf{q}_{j,r}^{\mathrm{T}}\mathbf{q}_{j,r})}$;
        end for
        $\ell_r = \arg\max_{j \in L_r}\{\mathrm{ERR}[j;r]\}$ such that $\ell_r \in L_r$;
        $\mathbf{q}_r = \mathbf{q}_{\ell_r,r}$; $\mathbf{z}_r = \mathbf{x}_{\ell_r}$;
        add $\mathbf{z}_r$ to $S$;
end for
```

Figure 4.1: The SOS-LLS algorithm.

4.4 Experimental Setup and Evaluation

In order to test and analyse the overall performance of the proposed SOS-LLS method, we applied the method to two categories of datasets: one with well-known data properties whereas the other does not. All datasets are publicly available online from the UCI machine learning repository excluding the Alate Adelges data. A complete Alate Adelges data matrix is accessible from Krzanowski (1987).

4.4.1 First Category of Benchmark Datasets

The true data characteristics in this category are known in advance, so the performance of a data analysis method (e.g. feature selection method) can easily be revealed through such datasets. Two datasets are considered: Iris and Alate Adelges. As both original datasets contain features with different units and scales, they were aligned using normalization prior to execution with the SOS-LLS algorithm so that a fair comparison can be made between features.

4.4.1.1 Experiments on Alate Adelges Dataset

The effectiveness of the SOS-LLS method is first depicted using a popular Alate Adelges dataset. The Alate Adelges dataset was first used by Jeffers (1967) as a case study for PCA application. It is characterized by 19 features, measured on a sample of 40 winged aphids caught in a light trap. Two main conclusions were drawn from the case study (Jeffers, 1967). First, it was concluded that two principal components is sufficient for representing the complete data. Second, it was concluded that the 40 aphids can be clustered into four distinct groups corresponding to four different types of aphids.

In order to evaluate the subset capability to represent the full feature set, the first two principal components score plot for a full Alate Adelges dataset and the score plots for two potential subsets by SOS-LLS were examined. These PC1-PC2 score plots are as presented in Figure 4.2.

In Figure 4.2 (a), both of the principal components are functions involving all 19 features. The plot exhibits four distinct clusters as inferred by Jeffers (1967). Note that observation number 34 is a real outlier because it belongs to a special class of aphid but this is not the case for observation number 19 (Heberger & Andrade, 2004). Therefore, observation 34 should not be attached to any of the four groups whilst observation 19 should be merged with the cluster marked with the symbol "✳". However, this is not the case when all features are considered as shown in Figure 4.2 (a).

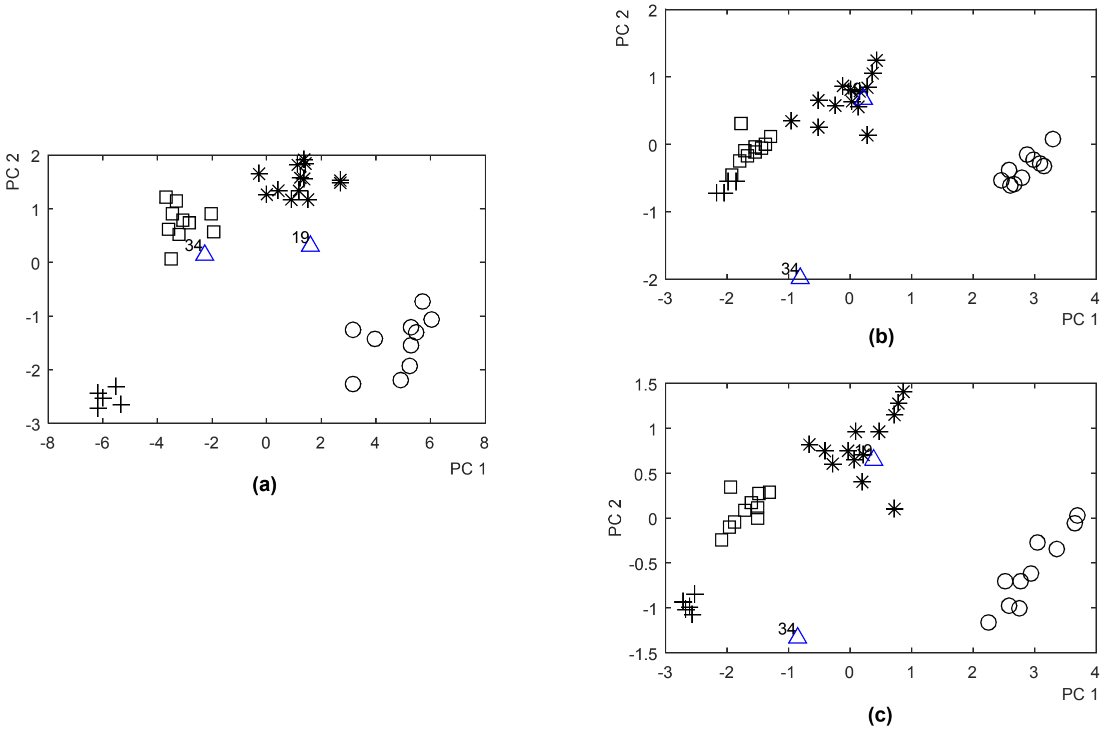

Figure 4.2: PC1-PC2 score plot for the Alate Adelges dataset based on (a) full feature set, (b) the first four selected features and (c) the first five selected features.

In Figure 4.2 (b) and (c), the two principal components only involve four and five selected features respectively. It can be seen from Figure 4.2 (b) that the four-feature subset obtained by using SOS-LLS method has started to form similar structural pattern as that formed by using the full feature set. A very similar pattern is captured by just including one more feature to the four-feature subset as depicted in Figure 4.2 (c). Thus, the five-feature subset can be considered as really good to substitute the full feature set if data structure is the main goal for feature selection. Notice that also the five-feature subset managed to reveal observation 34 as unique and anomalous while observation 19 was correctly grouped to its actual cluster. These results are of great importance as they signify that the SOS-LLS method is robust and can select the most representative features.

4.4.1.2 Experiments on Iris dataset

The Iris is one of the most popular benchmark datasets with well-known nature of data and is frequently used to test an algorithm's performance in pattern recognition studies. The dataset comprises of 150 observations where each observation is described by four continuous-valued variables (features) and belonging to one of the three distinct classes of iris flowers, namely, Setosa, Versicolour and Virginia. For each class, fifty observations were equally recorded. In this dataset, it is known that the Setosa class is linearly separable from the other two, while the other two are non-linearly separable from each other.

The four features of Iris dataset are sepal length $(\mathbf{f}_1)$, sepal width $(\mathbf{f}_2)$, petal length $(\mathbf{f}_3)$ and petal width $(\mathbf{f}_4)$. Among these four features, only the last two are relevant and sufficient to cluster the Iris dataset correctly into three groups corresponding to the three classes of Iris flowers.

Table 4.1 lists the feature ranking results of the Iris dataset given by different feature selection methods. The basic parameter settings for each of these feature selection methods are as follows. The number of nearest neighbours was set to be $k = 5$ for Laplacian Score, MCFS, and MMLS. This value was chosen for each of the methods based on the recommended range for the number of nearest neighbours that possibly lead to good feature subset solution. The same value $k = 5$ was also used in the proposed SOS-LLS method so that a fair comparison can be made with those obtained by the three competing methods. The heat kernel weighting scheme was specifically adopted to measure the closeness of neighbouring points for all these Laplacian graph-based methods including the proposed method. In MCFS, the required pre-defined number of clusters was set equal to the number of true classes of the dataset being considered. For ReliefF method, the number of nearest neighbours was restricted to $k = 10$ as suggested in Robnik-Sikonja & Kononenko (2003). In addition, the redundancy parameter of the MIFS approach was assigned a value $\beta = 1$, according to the appropriate range advised in Battiti (1994).

All methods listed in Table 4.1, except MCFS, rank $\mathbf{f}_3$ and $\mathbf{f}_4$ as the first two significant features, following the actual order that corresponds to the most relevant feature pair. Note that Laplacian Score, MCFS, MMLS and the proposed SOS-LLS are unsupervised methods while the others are supervised methods. This shows that even though SOS-LLS is unsupervised, it is capable to reach the same result as these most commonly used supervised methods.

Table 4.1: Feature ranking results of the Iris dataset given by different feature selection methods.

	Unsupervised feature selection			
	Laplacian score	MCFS	MMLS	SOS-LLS
Feature ranking	$\mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_1, \mathbf{f}_2$	$\mathbf{f}_3, \mathbf{f}_2, \mathbf{f}_1, \mathbf{f}_4$	$\mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_1, \mathbf{f}_2$	$\mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_2, \mathbf{f}_1$
	Supervised feature selection			
	Fisher score	ReliefF	mRMR	MIFS
Feature ranking	$\mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_1, \mathbf{f}_2$	$\mathbf{f}_4, \mathbf{f}_3, \mathbf{f}_2, \mathbf{f}_1$	$\mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_2, \mathbf{f}_1$	$\mathbf{f}_3, \mathbf{f}_4, \mathbf{f}_2, \mathbf{f}_1$

4.4.2 Second Category of Benchmark Datasets

In contrast to the first category, the second category considers datasets with unknown or unclear data properties. Eight datasets of this category are used to further demonstrate the efficacy of the proposed method from different perspective. Table 4.2 summarises some important details of the used datasets. Here, feature subset solutions obtained by the SOS-LLS method are evaluated by using classification performance for the listed datasets. The experimental results based on SOS-LLS are compared with that from a number of state-of-the-art methods; some details are described as below.

Table 4.2: Important details of the used benchmark datasets for 2nd category.

Dataset	Number of features	Number of observations	Number of classes
Pima Diabetes	8	768	2
Wbc	9	699	2
Glass [N]	9	214	7
Vowel [N]	10	990	11
Statlog [N]	18	846	4
Ionosphere	33	351	2
Waveform	40	5000	4
Mfeat Zernike [N]	47	2000	10
Sonar	60	208	2
Musk [N]	166	476	2
Mfeat Factors [N]	216	2000	10
Isolet	649	2000	26

[N]: The raw dataset was normalized before the experiment.

It is interesting to compare the results of SOS-LLS with a few similar methods. For this purpose, the Laplacian Score (LS), MCFS and MMLS are employed because these are also unsupervised methods, using filter approach and most importantly sharing the same type of evaluation criterion hinged on locality preserving information. In all the experiments, the parameter settings for all methods including the SOS-LLS method are the same as given in Section 4.4.1.1.

Since SOS-LLS is a filter method, its reliability might be different from one classifier to another. Thus, it is also interesting to validate its effectiveness by applying SOS-LLS across several classifiers with different learning architecture. Four widely used classifiers, listed among the ten most influential data mining algorithms (Wu, et al., 2008) , namely, *k*-nearest neighbour (*k*-NN), Naïve Bayes (NBayes), support vector machine (SVM), and classification and regression trees (CART), are considered here. To provide a fair comparison, the number of nearest neighbours of the *k*-NN classifier was set to $k = 5$ for all tests.

The same holdout cross-validation approach was adopted for each classifier in order to avoid overfitting and gain more accurate performance generalization. In particular, the considered dataset was randomly split into two sets where 80% were used as a training set while the remaining 20% were holdout and reserved as a validation set. The classification model was first built by using the training set and the validation set was later used to assess the model performance. In addition, 30 iterations of this cross-validation procedure were carried out, from which the average percentage of classification accuracies was calculated to reduce the effect of the random variation error in the result.

Table 4.3 and Table 4.4 present the average classification accuracy based on m selected features over the four classifiers (5-NN, Naive Bayes, SVM, and CART). Only a few cases with certain values of m are reported in the tables, as these cases should be sufficiently representative to demonstrate the overall performance of the four methods used (i.e., SOS-LLS, LS, MCFS, and MMLS). In order to determine whether the classification accuracy based upon the feature subset selected by SOS-LLS is significantly higher or lower than that induced by its competitor, a one-tailed two-sample *z*-test was performed for each case of the *m* values. As such, the test was conducted based on a hypothesis that "the average classification accuracy of the proposed method is greater than the average classification accuracy of the compared method". The value recorded within the bracket in Table 4.3 and Table 4.4 is the *p*-value corresponding to the *z*-test and it serves as an indicator to show how the results on the data are consistent with the aforementioned hypothesis. A *p*-value lower than or equal to 0.05 (5 % significance level) indicates that the z-test statistic provides enough evidence to support the original hypothesis. Meanwhile, a *p*-value of at least 0.95 suggests that the compared method wins over the proposed method. For ease of comparison, the results are marked with "✓" and

"✖" to indicate that the SOS-LLS method is significantly superior or inferior to the compared method, respectively. Otherwise, if no symbol is specified, it means that the two methods are comparable.

As can be seen from Table 4.3 and Table 4.4, SOS-LLS consistently or almost consistently shows better or comparable classification accuracy with all classifiers compared to other feature selection methods particularly on Pima Diabetes, Ionosphere, Waveform, Mfeat Zernike and Musk datasets. The performance of SOS-LLS, however, is not as good as that achieved by Laplacian Score in general for Glass, Vowel, Sonar and Isolet. Yet, SOS-LLS beats Laplacian Score on Vowel data with Naïve Bayes classifier for all the m cases. In the meantime, for Glass, Sonar and Isolet datasets, it can be observed that SOS-LLS still provides strong competition to Laplacian Score with SVM classifier. SOS-LLS basically gives competitive performance over the MCFS method in many cases of different combinations of classifiers and feature subset sizes except for Statlog, Sonar and Isolet data. For Wbc, though the proposed SOS-LLS obviously does not show as good as performance as MCFS for the same feature subset size, it just loses to MCFS for a few cases only. When compared to MMLS, SOS-LLS only fails to perform satisfactorily on Glass and Vowel datasets but it performs exceptionally well over the rest of the benchmark datasets.

Table 4.5 and Table 4.6 present the test performance results of SOS-LLS and the other three methods. It is noteworthy that the performance was calculated based on the least number of features for each of the methods, where the least number m_{least} was determined as follows: the classification accuracy of a method using only m_{least} features close to (with tolerance no more than 5% less) or higher than that obtained by using the full feature set.

Results from both Table 4.5 and Table 4.6 are abstracted and summarised in Table 4.7, to give more insightful inspection of the overall performance of SOS-LLS in representing the original full feature set. The recorded win/tie/loss scores in Table 4.7 refer to the number of benchmark datasets for which the SOS-LLS method uses lower/equal/higher feature subset size when compared with the other locality preserving methods.

Table 4.3: Performance comparison of the average classification accuracy based on *m* selected features with four classifiers. The value within the bracket is the *p*-value to test whether the accuracy of SOS-LLS is significantly larger than that obtained by its competitor.

		Pima Diabetes				**Wbc**			
		SOS-LLS	LS	MCFS	MMLS	SOS-LLS	LS	MCFS	MMLS
5-NN	*m* = 2	70.57	69.98 [0.21]	71.07 [0.72]	63.79 [0.00] ✓	93.60	95.90 [1.00] ✖	95.37 [1.00] ✖	93.91 [0.73]
	m = 4	72.11	70.50 [0.02] ✓	64.53 [0.00] ✓	71.87 [0.38]	95.69	96.28 [0.15]	96.52 [0.42]	96.43 [0.30]
	m = 6	71.39	71.66 [0.61]	65.86 [0.00] ✓	71.35 [0.48]	97.17	96.67 [0.13]	96.86 [0.24]	95.92 [0.00] ✓
NBayes	*m* = 2	69.56	68.80 [0.12]	68.61 [0.06]	66.14 [0.00] ✓	93.12	94.03 [0.97] ✖	93.76 [0.90]	92.01 [0.02] ✓
	m = 4	70.57	70.37 [0.39]	67.30 [0.00] ✓	70.33 [0.38]	96.52	94.80 [0.00] ✓	96.67 [0.68]	94.94 [0.00] ✓
	m = 6	73.36	73.31 [0.47]	68.06 [0.00] ✓	72.14 [0.05] ✓	95.54	96.28 [0.98] ✖	96.52 [1.00] ✖	95.95 [0.86]
SVM	*m* = 2	74.05	74.99 [0.90]	73.57 [0.26]	65.14 [0.00] ✓	93.96	95.83 [1.00] ✖	94.92 [0.99] ✖	94.84 [0.98] ✖
	m = 4	76.71	74.31 [0.00] ✓	65.73 [0.00] ✓	74.60 [0.00] ✓	97.17	96.19 [0.01] ✓	96.67 [0.09]	95.64 [0.00] ✓
	m = 6	75.86	76.14 [0.63]	67.60 [0.00] ✓	76.45 [0.79]	96.55	96.83 [0.76]	96.62 [0.57]	96.00 [0.08]
CART	*m* = 2	66.75	66.80 [0.52]	67.28 [0.72]	63.68 [0.00] ✓	94.53	95.16 [0.88]	94.96 [0.79]	93.48 [0.02] ✓
	m = 4	68.65	66.86 [0.03] ✓	62.11 [0.00] ✓	67.56 [0.16]	94.77	94.92 [0.65]	94.72 [0.45]	94.94 [0.65]
	m = 6	71.15	70.94 [0.39]	66.12 [0.00] ✓	69.17 [0.01] ✓	94.17	94.32 [0.62]	94.36 [0.67]	94.29 [0.60]

		Glass				Vowel			
		SOS-LLS	LS	MCFS	MMLS	SOS-LLS	LS	MCFS	MMLS
5-NN	*m* = 2	52.06	42.38 [0.00] ✓	41.19 [0.00] ✓	68.81 [1.00] ✖	45.99	62.54 [1.00] ✖	29.34 [0.00] ✓	62.64 [1.00] ✖
	m = 4	52.62	64.44 [1.00] ✖	60.63 [1.00] ✖	73.41 [1.00] ✖	81.40	82.26 [0.91] ✖	60.64 [0.00] ✓	82.86 [0.99] ✖
	m = 6	64.37	68.57 [0.99] ✖	61.51 [0.06]	71.03 [1.00] ✖	87.21	90.42 [1.00] ✖	82.73 [0.00] ✓	88.74 [0.99] ✖
NBayes	*m* = 2	39.37	41.59 [0.95] ✖	47.78 [0.02] ✓	55.48 [1.00] ✖	37.19	22.82 [0.00] ✓	24.70 [0.00] ✓	57.00 [1.00] ✖
	m = 4	42.78	49.84 [1.00] ✖	47.78 [0.00] ✓	65.87 [1.00] ✖	62.93	26.67 [0.00] ✓	28.77 [0.00] ✓	63.84 [0.86]
	m = 6	56.19	60.79 [1.00] ✖	48.33 [0.00] ✓	67.46 [0.85]	68.42	62.56 [0.00] ✓	50.47 [0.00] ✓	68.10 [0.35]
SVM	*m* = 2	50.95	45.95 [0.00] ✓	42.46 [0.00] ✓	50.16 [0.31]	31.48	53.91 [1.00] ✖	23.60 [0.00] ✓	52.95 [1.00] ✖
	m = 4	53.89	57.86 [1.00] ✖	52.14 [0.14]	62.38 [1.00] ✖	63.79	64.73 [0.87]	20.71 [0.00] ✓	64.85 [0.89]
	m = 6	65.71	63.17 [0.05] ✓	62.70 [0.07]	63.89 [0.17]	67.78	69.34 [0.96] ✖	46.52 [0.00] ✓	70.15 [1.00] ✖
CART	*m* = 2	47.06	45.56 [0.21]	47.54 [0.59]	61.43 [1.00] ✖	42.46	56.70 [1.00] ✖	26.13 [0.00] ✓	56.89 [1.00] ✖
	m = 4	56.83	60.32 [0.99] ✖	56.75 [0.48]	68.73 [1.00] ✖	69.70	70.88 [0.94]	44.24 [0.00] ✓	71.31 [0.97] ✖
	m = 6	65.00	64.60 [0.42]	63.81 [0.28]	68.97 [0.98] ✖	73.55	75.15 [0.98] ✖	60.82 [0.00] ✓	72.78 [0.19]

		Statlog				Ionosphere			
		SOS-LLS	LS	MCFS	MMLS	SOS-LLS	LS	MCFS	MMLS
5-NN	*m* = 5	60.91	55.74 [0.00] ✓	65.13 [1.00] ✖	55.54 [0.00] ✓	86.57	81.67 [0.00] ✓	83.86 [0.00] ✓	81.29 [0.00] ✓
	m = 10	69.53	67.91 [0.01] ✓	71.79 [1.00] ✖	69.37 [0.39]	85.76	83.29 [0.01] ✓	82.95 [0.00] ✓	84.71 [0.15]
	m = 15	72.45	70.95 [0.04] ✓	71.62 [0.15]	70.49 [0.01] ✓	85.67	84.00 [0.04] ✓	84.67 [0.13]	86.24 [0.73]
NBayes	*m* = 5	54.22	51.32 [0.00] ✓	61.34 [1.00] ✖	52.58 [0.02] ✓	75.71	73.43 [0.02] ✓	77.52 [0.95] ✖	72.48 [0.00] ✓
	m = 10	57.71	56.02 [0.03] ✓	61.66 [1.00] ✖	56.04 [0.05] ✓	80.38	74.38 [0.00] ✓	78.14 [0.04] ✓	75.90 [0.00] ✓
	m = 15	59.13	60.28 [0.88]	59.86 [0.82]	58.74 [0.33]	85.43	79.57 [0.00] ✓	81.86 [0.00] ✓	79.57 [0.00] ✓
SVM	*m* = 5	57.55	46.09 [0.00] ✓	59.63 [0.99] ✖	45.92 [0.00] ✓	81.71	70.33 [0.00] ✓	75.86 [0.00] ✓	70.57 [0.00] ✓
	m = 10	72.98	67.12 [0.00] ✓	72.84 [0.43]	70.26 [0.00] ✓	87.00	79.00 [0.00] ✓	79.43 [0.00] ✓	77.00 [0.00] ✓
	m = 15	75.78	77.20 [0.97] ✖	80.04 [1.00] ✖	77.34 [0.97] ✖	87.86	83.71 [0.00] ✓	83.95 [0.00] ✓	82.86 [0.00] ✓
CART	*m* = 5	60.20	55.25 [0.00] ✓	64.60 [1.00] ✖	56.04 [0.00] ✓	85.52	84.71 [0.20]	83.62 [0.02] ✓	84.10 [0.07]
	m = 10	68.26	66.33 [0.02] ✓	68.78 [0.71]	66.06 [0.01] ✓	88.52	84.57 [0.00] ✓	83.76 [0.00] ✓	80.76 [0.00] ✓
	m = 15	69.09	70.00 [0.83]	69.64 [0.73]	68.86 [0.41]	90.19	86.10 [0.00] ✓	85.90 [0.00] ✓	85.62 [0.00] ✓

Table 4.4: Performance comparison of the average classification accuracy based on *m* selected features with four classifiers. The value within the bracket is the *p*-value to test whether the accuracy of SOS-LLS is significantly larger than that obtained by its competitor.

		Waveform				Mfeat Zernike			
		SOS-LLS	LS	MCFS	MMLS	SOS-LLS	LS	MCFS	MMLS
5-NN	*m* = 5	76.22	68.10 [0.00] ✓	75.72 [0.05] ✓	65.22 [0.00] ✓	59.39	34.96 [0.00] ✓	53.95 [0.00] ✓	33.79 [0.00] ✓
	m = 10	81.68	78.19 [0.00] ✓	79.62 [0.00] ✓	77.78 [0.00] ✓	73.37	53.92 [0.00] ✓	74.91 [1.00] ✖	42.39 [0.00] ✓
	m = 15	83.42	83.58 [0.73]	82.80 [0.01] ✓	83.79 [0.93]	80.30	58.51 [0.00] ✓	77.31 [0.00] ✓	54.58 [0.00] ✓
	m = 30					80.24	74.98 [0.00] ✓	80.03 [0.29]	66.48 [0.00] ✓
NBayes	*m* = 5	75.93	67.28 [0.00] ✓	76.03 [0.63]	65.02 [0.00] ✓	53.68	34.53 [0.00] ✓	53.23 [0.18]	29.97 [0.00] ✓
	m = 10	77.42	73.69 [0.00] ✓	76.02 [0.00] ✓	73.60 [0.00] ✓	63.58	46.37 [0.00] ✓	68.08 [1.00] ✖	39.95 [0.00] ✓
	m = 15	79.90	80.22 [0.91]	79.35 [0.00] ✓	79.69 [0.17]	71.79	50.37 [0.00] ✓	71.85 [0.56]	45.91 [0.00] ✓
	m = 30	-	-	-	-	73.04	66.21 [0.00] ✓	73.84 [0.93]	58.47 [0.00] ✓
SVM	*m* = 5	79.00	72.81 [0.00] ✓	79.13 [0.63]	71.09 [0.00] ✓	53.91	38.29 [0.00] ✓	55.03 [0.97] ✖	35.61 [0.00] ✓
	m = 10	84.27	81.68 [0.00] ✓	83.70 [0.02] ✓	81.44 [0.00] ✓	72.13	57.13 [0.00] ✓	72.93 [0.94]	44.23 [0.00] ✓
	m = 15	86.71	86.56[17]	86.45 [0.18]	86.61 [0.37]	80.75	61.08 [0.00] ✓	78.99 [0.00] ✓	56.28 [0.00] ✓
	m = 30	-	-	-	-	82.06	78.23 [0.00] ✓	81.89 [0.29]	72.22 [0.00] ✓
CART	*m* = 5	71.00	63.00 [0.00] ✓	71.15 [0.64]	59.87 [0.00] ✓	50.49	31.42 [0.00] ✓	47.52 [0.00] ✓	30.35 [0.00] ✓
	m = 10	75.55	70.98 [0.00] ✓	73.73 [0.00] ✓	71.44 [0.00] ✓	58.92	46.62 [0.00] ✓	63.61 [1.00] ✖	38.52 [0.00] ✓
	m = 15	75.68	76.52 [0.99] ✖	75.75 [0.58]	76.08 [0.94]	64.25	50.05 [0.00] ✓	64.51 [0.68]	46.83 [0.00] ✓
	m = 30	-	-	-	-	66.78	63.68 [0.00] ✓	65.73 [0.02] ✓	56.12 [0.00] ✓

		Sonar				Musk			
		SOS-LLS	LS	MCFS	MMLS	SOS-LLS	LS	MCFS	MMLS
5-NN	*m* = 5	65.37	71.71 [1.00] ✖	68.94 [0.99] ✖	66.91 [0.84]	75.93	67.68 [0.00] ✓	70.95 [0.00] ✓	70.04 [0.00] ✓
	m = 10	66.67	70.57 [0.99] ✖	74.72 [1.00] ✖	69.59 [0.94]	77.86	71.44 [0.00] ✓	73.16 [0.00] ✓	69.86 [0.00] ✓
	m = 15	72.76	73.50 [0.66]	77.56 [1.00] ✖	73.25 [0.61]	76.35	76.00 [0.38]	75.33 [0.16]	72.49 [0.00] ✓
	m = 30	79.84	73.82 [0.00] ✓	79.11 [0.34]	71.87 [0.00] ✓	79.30	78.67 [0.31]	74.11 [0.00] ✓	71.82 [0.00] ✓
NBayes	*m* = 5	53.74	68.62 [1.00] ✖	68.62 [1.00] ✖	60.24 [1.00] ✖	64.18	62.35 [0.03] ✓	64.74 [0.68]	54.67 [0.00] ✓
	m = 10	66.50	65.04 [0.23]	69.92 [0.97] ✖	58.54 [0.00] ✓	67.16	65.23 [0.04] ✓	69.58 [0.98] ✖	57.96 [0.00] ✓
	m = 15	68.46	69.43 [0.69]	76.75 [1.00] ✖	60.89 [0.00] ✓	71.40	63.30 [0.00] ✓	66.25 [0.00] ✓	58.35 [0.00] ✓
	m = 30	74.39	70.73 [0.02] ✓	77.48 [0.95] ✖	64.80 [0.00] ✓	75.82	64.18 [0.00] ✓	71.47 [0.00] ✓	62.21 [0.00] ✓
SVM	*m* = 5	50.41	67.89 [1.00] ✖	60.65 [1.00] ✖	60.41 [1.00] ✖	58.77	57.68 [0.10]	63.51 [1.00] ✖	55.54 [0.00] ✓
	m = 10	59.43	63.17 [0.99] ✖	62.44 [0.96] ✖	61.14 [0.83]	67.51	60.63 [0.00] ✓	63.23 [0.00] ✓	55.89 [0.00] ✓
	m = 15	69.67	63.74 [0.00] ✓	72.20 [0.94]	60.49 [0.00] ✓	69.12	55.54 [0.00] ✓	67.19 [0.03] ✓	59.16 [0.00] ✓
	m = 30	75.61	69.27 [0.00] ✓	73.33 [0.07]	65.69 [0.00] ✓	75.02	60.35 [0.00] ✓	74.67 [0.38]	59.19 [0.00] ✓
CART	*m* = 5	56.59	67.32 [1.00] ✖	68.21 [1.00] ✖	59.11 [0.92]	72.67	69.79 [0.01] ✓	72.88 [0.57]	68.28 [0.00] ✓
	m = 10	61.06	69.43 [1.00] ✖	72.11 [1.00] ✖	58.78 [0.08]	71.02	72.00 [0.77]	71.61 [0.66]	67.05 [0.00] ✓
	m = 15	65.85	66.75 [0.70]	78.05 [1.00] ✖	61.06 [0.00] ✓	76.98	74.46 [0.01] ✓	74.35 [0.02] ✓	72.18 [0.00] ✓
	m = 30	72.76	68.70 [0.01] ✓	73.25 [0.62]	67.80 [0.00] ✓	77.72	77.44 [0.42]	76.00 [0.11]	75.19 [0.04] ✓

		Mfeat Factors				Isolet			
		SOS-LLS	LS	MCFS	MMLS	SOS-LLS	LS	MCFS	MMLS
5-NN	*m* = 5	71.09	69.63 [0.00] ✓	65.70 [0.00] ✓	63.57 [0.00] ✓	-	-	-	-
	m = 10	88.28	76.85 [0.00] ✓	89.59 [1.00] ✖	79.44 [0.00] ✓	32.17	36.86 [1.00] ✖	55.73 [1.00] ✖	29.55 [0.00] ✓
	m = 15	91.96	76.11 [0.00] ✓	92.18 [0.78]	83.72 [0.00] ✓	42.75	54.97 [1.00] ✖	61.53 [1.00] ✖	34.30 [0.00] ✓
	m = 30	94.99	89.52 [0.00] ✓	95.06 [0.61]	91.15 [0.00] ✓	58.35	68.83 [1.00] ✖	74.78 [1.00] ✖	47.13 [0.00] ✓
	m = 60	-	-	-	-	74.42	76.41 [1.00] ✖	78.29 [1.00] ✖	55.30 [0.00] ✓
	m = 120	-	-	-	-	84.05	80.79 [0.00] ✓	84.84 [1.00] ✖	71.57 [0.00] ✓
	m = 240	-	-	-	-	88.41	84.34 [0.00] ✓	88.15 [0.10]	81.67 [0.00] ✓
NBayes	*m* = 5	65.31	64.84 [0.20]	63.99 [0.01] ✓	60.08 [0.00] ✓	-	-	-	-
	m = 10	84.44	61.49 [0.00] ✓	85.97 [1.00] ✖	63.86 [0.00] ✓	21.35	21.45 [0.59]	37.05 [1.00] ✖	23.74 [1.00] ✖
	m = 15	84.48	64.06 [0.00] ✓	88.74 [1.00] ✖	67.16 [0.00] ✓	29.78	33.06 [1.00] ✖	42.08 [1.00] ✖	27.26 [0.00] ✓
	m = 30	91.83	77.97 [0.00] ✓	91.86 [0.53]	78.66 [0.00] ✓	40.98	43.97 [1.00] ✖	55.02 [1.00] ✖	33.94 [0.00] ✓
	m = 60	-	-	-	-	59.11	56.60 [0.00] ✓	66.20 [1.00] ✖	43.71 [0.00] ✓
	m = 120	-	-	-	-	62.66	66.64 [1.00] ✖	71.52 [1.00] ✖	63.09 [0.82]
	m = 240	-	-	-	-	77.09	75.09 [0.00] ✓	76.78 [0.23]	77.98 [0.99] ✖
SVM	*m* = 5	68.09	75.18 [1.00] ✖	62.97 [0.00] ✓	68.03 [0.45]	-	-	-	-
	m = 10	89.54	80.68 [0.00] ✓	88.99 [0.08]	83.54 [0.00] ✓	37.47	37.43 [0.45]	59.60 [1.00] ✖	31.10 [0.00] ✓
	m = 15	93.35	83.42 [0.00] ✓	93.01 [0.14]	88.81 [0.00] ✓	50.55	58.21 [1.00] ✖	65.96 [1.00] ✖	36.23 [0.00] ✓
	m = 30	96.33	93.71 [0.00] ✓	95.75 [0.01] ✓	93.72 [0.00] ✓	71.75	72.30 [0.96] ✖	81.31 [1.00] ✖	50.99 [0.00] ✓
	m = 60	-	-	-	-	87.20	82.92 [0.00] ✓	86.92 [0.10]	65.41 [0.00] ✓
	m = 120	-	-	-	-	93.05	89.06 [0.00] ✓	92.13 [0.00] ✓	81.45 [0.00] ✓
	m = 240	-	-	-	-	94.82	92.29 [0.00] ✓	94.73 [0.24]	91.41 [0.00] ✓
CART	*m* = 5	62.36	65.63 [1.00] ✖	60.52 [0.00] ✓	60.39 [0.00] ✓	-	-	-	-
	m = 10	76.38	70.15 [0.00] ✓	79.83 [1.00] ✖	72.41 [0.00] ✓	32.17	35.11 [1.00] ✖	53.59 [1.00] ✖	26.29 [0.00] ✓
	m = 15	81.02	70.04 [0.00] ✓	82.38 [1.00] ✖	77.49 [0.00] ✓	40.46	52.39 [1.00] ✖	58.47 [1.00] ✖	29.91 [0.00] ✓
	m = 30	84.57	80.88 [0.00] ✓	85.55 [0.99] ✖	80.76 [0.00] ✓	54.89	64.73 [1.00] ✖	70.10 [1.00] ✖	40.90 [0.00] ✓
	m = 60	-	-	-	-	70.45	72.44 [1.00] ✖	74.05 [1.00] ✖	50.21 [0.00] ✓
	m = 120	-	-	-	-	77.34	76.48 [0.00] ✓	79.46 [1.00] ✖	63.97 [0.00] ✓
	m = 240	-	-	-	-	79.57	78.52 [0.00] ✓	80.80 [1.00] ✖	77.01 [0.00] ✓

Table 4.5: The least feature subset size, m_{least}, given by different feature selection methods that reach classification accuracy close to (with tolerance no more than 5% less) or maybe more than that obtained by the full feature set of size M. The symbol "●" (or "□") marks that SOS-LLS gives smaller (or larger) value of m_{least} than the compared method. Results are based on eight benchmarks datasets.

	Pima Diabetes		Wbc		Glass		Vowel	
5-NN	M = 8	71.94 ± 1.36	M = 9	96.83 ± 0.58	M = 9	65.40 ± 2.32	M = 10	92.31 ± 0.70
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	2	70.57 ± 1.11	2	93.60 ± 0.62	5	63.49 ± 2.80	7	90.59 ± 0.91
LS	2	69.98 ± 0.94	2	95.90 ± 0.66	4 □	64.44 ± 2.16	6 □	90.42 ± 0.81
MCFS	2	71.07 ± 1.29	2	95.37 ± 0.63	5	64.84 ± 2.24	7	89.73 ± 0.72
MMLS	3 ●	71.94 ± 1.05	2	93.91 ± 0.76	2 □	68.81 ± 2.23	6 □	88.74 ± 0.74
NBayes	M	73.38 ± 1.05	M	96.26 ± 0.52	M	62.70 ± 2.80	M	73.03 ± 1.20
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	2	69.56 ± 0.80	2	93.12 ± 0.77	9	62.70 ± 2.80	8	71.77 ± 1.00
LS	3 ●	70.17 ± 1.13	2	94.03 ± 0.60	7 □	61.35 ± 2.75	7 □	69.71 ± 1.13
MCFS	7 ●	69.74 ± 1.17	2	93.76 ± 0.57	9	62.70 ± 2.80	9 ●	72.31 ± 0.94
MMLS	4 ●	70.33 ± 1.08	2	92.01 ± 0.67	3 □	65.00 ± 2.61	8	70.96 ± 0.99
SVM	M	76.69 ± 1.26	M	96.31 ± 0.43	M	62.06 ± 2.52	M	77.04 ± 1.01
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	2	74.05 ± 0.99	2	93.96 ± 0.61	6	65.71 ± 2.33	8	75.69 ± 1.09
LS	2	74.99 ± 1.08	2	95.83 ± 0.49	5 □	62.86 ± 1.77	8	75.34 ± 1.35
MCFS	2	73.57 ± 1.07	2	94.92 ± 0.47	6	62.70 ± 3.24	9 ●	73.75 ± 0.95
MMLS	3 ●	73.49 ± 0.87	2	94.84 ± 0.60	4 □	62.38 ± 2.22	8	75.12 ± 1.07
CART	M	70.61 ± 1.22	M	94.03 ± 0.80	M	66.75 ± 2.68	M	74.33 ± 1.43
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	4	68.56 ± 1.48	2	94.53 ± 0.80	9	66.75 ± 2.68	5	71.90 ± 1.43
LS	3 □	67.28 ± 1.49	2	95.16 ± 0.69	7 □	66.27 ± 3.02	4 □	70.88 ± 1.21
MCFS	2 □	67.28 ± 1.16	2	94.96 ± 0.68	9	66.75 ± 2.68	7 ●	71.94 ± 1.02
MMLS	3 □	68.24 ± 1.43	2	93.48 ± 0.59	3 □	65.71 ± 2.22	4 □	71.31 ± 1.41
	Statlog		Ionosphere		Waveform		Mfeat Zernike	
5-NN	M = 1 8	70.85 ± 1.02	M = 33	83.19 ± 1.48	M = 40	81.15 ± 0.40	M = 47	80.75 ± 0.49
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	7	68.34 ± 1.18	4	83.52 ± 1.54	6	77.13 ± 0.42	13	79.13 ± 0.56
LS	9 ●	69.33 ± 1.15	2 □	81.10 ± 1.45	10 ●	78.19 ± 0.36	32 ●	76.53 ± 0.59
MCFS	7	68.22 ± 1.04	3 □	82.57 ± 1.59	7 ●	76.95 ± 0.41	15 ●	77.31 ± 0.61
MMLS	9 ●	68.60 ± 1.07	2 □	82.00 ± 1.21	10 ●	77.78 ± 0.40	38 ●	77.16 ± 0.59
NBayes	M	61.52 ± 1.49	M	91.24 ± 1.05	M	79.75 ± 0.30	M	72.48 ± 0.58
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	7	58.88 ± 1.35	16	88.00 ± 1.51	4	75.37 ± 0.46	13	69.40 ± 0.87
LS	13 ●	59.94 ± 1.08	26 ●	90.38 ± 1.17	11 ●	76.85 ± 0.37	31 ●	68.32 ± 0.62
MCFS	2 □	61.14 ± 1.12	22 ●	88.86 ± 1.35	5 ●	76.03 ± 0.39	11 □	68.53 ± 0.58
MMLS	15 ●	58.74 ± 1.20	21 ●	87.81 ± 1.50	13 ●	76.48 ± 0.33	39 ●	68.78 ± 0.86
SVM	M	78.76 ± 0.83	M	86.90 ± 1.19	M	86.23 ± 0.33	M	82.17 ± 0.40
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	12	75.74 ± 1.22	6	84.00 ± 1.29	8	82.06 ± 0.39	13	79.23 ± 0.59
LS	15 ●	77.20 ± 0.92	15 ●	83.71 ± 1.40	10 ●	81.68 ± 0.42	28 ●	78.12 ± 0.43
MCFS	12	79.17 ± 0.96	15 ●	83.95 ± 1.30	8	82.24 ± 0.46	15 ●	78.99 ± 0.75
MMLS	14 ●	76.02 ± 1.17	16 ●	84.00 ± 1.20	11 ●	82.76 ± 0.34	36 ●	78.47 ± 0.67
CART	M	70.51 ± 1.05	M	87.24 ± 1.58	M	74.34 ± 0.49	M	67.51 ± 0.80
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{l5east}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	8	67.83 ± 1.15	5	85.52 ± 1.32	5	71.00 ± 0.48	13	63.64 ± 0.66
LS	11 ●	66.92 ± 1.12	5	84.71 ± 1.39	9 ●	70.40 ± 0.64	30 ●	63.68 ± 0.82
MCFS	7 □	68.17 ± 1.36	6 ●	87.19 ± 1.01	5	71.15 ± 0.69	10 □	63.61 ± 0.78
MMLS	13 ●	67.79 ± 1.38	3 □	85.67 ± 1.61	9 ●	69.91 ± 0.39	38 ●	64.88 ± 0.67

Table 4.6: The least feature subset size, m_{least}, given by different feature selection methods that reach classification accuracy close to (with tolerance no more than 5% less) or maybe more than that obtained by the full feature set of size M. The symbol "●" (or "□") marks that SOS-LLS gives smaller (or larger) value of m_{least} than the compared method. Results are based on four benchmarks datasets.

	Sonar		Musk		Mfeat Factors		Isolet	
5-NN	M = 60	80.65 ± 1.87	M = 166	87.26 ± 1.41	M = 216	96.74 ± 0.28	M = 617	88.60 ± 0.20
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	28	78.62 ± 2.35	65	84.60 ± 1.20	16	93.14 ± 0.43	120	84.05 ± 0.25
LS	41 ●	79.43 ± 2.00	120 ●	84.60 ± 1.12	60 ●	92.82 ± 0.52	160 ●	84.65 ± 0.27
MCFS	21 □	78.37 ± 2.08	100 ●	84.11 ± 1.25	15 □	92.18 ± 0.39	115 □	83.99 ± 0.28
MMLS	58 ●	79.35 ± 2.34	75 ●	83.82 ± 1.14	60 ●	93.22 ± 0.37	370 ●	83.96 ± 0.30
NBayes	M	75.69 ± 2.58	M	82.08 ± 1.46	M	94.04 ± 0.35	M	82.79 ± 0.42
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	24	75.85 ± 2.65	60	79.21 ± 1.61	20	90.25 ± 0.46	320	78.46 ± 0.40
LS	34 ●	75.45 ± 2.34	135 ●	78.77 ± 1.50	110 ●	89.69 ± 0.58	410 ●	78.39 ± 0.39
MCFS	13 □	75.69 ± 1.72	55 □	80.98 ± 1.88	16 □	90.26 ± 0.51	320	78.44 ± 0.39
MMLS	52 ●	75.85 ± 2.27	125 ●	78.95 ± 1.39	120 ●	89.99 ± 0.62	260 □	78.47 ± 0.38
SVM	M	77.89 ± 2.48	M	85.65 ± 1.13	M	97.61 ± 0.35	M	96.28 ± 0.17
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	29	75.77 ± 2.10	85	83.26 ± 1.36	15	93.35 ± 0.47	95	91.55 ± 0.26
LS	34 ●	75.94 ± 2.19	145 ●	82.77 ± 1.22	25 ●	93.42 ± 0.34	190 ●	91.77 ± 0.27
MCFS	31 ●	76.67 ± 2.33	60 □	82.81 ± 1.08	16 ●	94.22 ± 0.37	105 ●	91.54 ± 0.24
MMLS	59 ●	78.37 ± 1.65	125 ●	83.51 ± 1.29	27 ●	93.12 ± 0.51	250 ●	91.62 ± 0.25
CART	M	70.08 ± 2.27	M	79.37 ± 1.96	M	87.72 ± 0.49	M	81.05 ± 0.29
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-LLS	13	69.76 ± 2.16	15	76.98 ± 1.52	18	83.42 ± 0.55	110	77.27 ± 0.39
LS	6 □	68.94 ± 2.98	18 ●	78.21 ± 1.45	60 ●	84.76 ± 0.82	110	76.50 ± 0.32
MCFS	6 □	69.43 ± 2.76	26 ●	77.09 ± 1.66	17 □	83.60 ± 0.50	85 □	76.69 ± 0.40
MMLS	33 ●	68.86 ± 2.73	20 ●	76.46 ± 1.41	70 ●	84.73 ± 0.67	210 ●	76.87 ± 0.41

Table 4.7: Tabulations of the win/tie/loss counts of the SOS-LLS method versus other methods. The counts are based on the results presented in Table 4.5 and Table 4.6.

Win/tie/lose	LS	MCFS	MMLS
5-NN	7 / 2 / 3	3 / 5 / 4	8 / 1 / 3
Naïve Bayes	9 / 1 / 2	4 / 3 / 5	8 / 2 / 2
SVM	8 / 3 / 1	6 / 5 / 1	9 / 2 / 1
CART	5 / 3 / 4	3 / 3 / 6	7 / 1 / 4
Average	7.25 / 2.25 / 2.5	4 / 4 / 4	8 / 1.5 / 2.5

Based on the results presented in Table 4.7, it is clear that in comparison to MMLS, the proposed SOS-LLS method gives outstanding performance in terms of smaller subset size among all the four classifiers. SOS-LLS also shows remarkable performance for three out of four classifiers when compared to Laplacian Score but it narrowly wins with CART classifier. It should be stressed that the proposed method does not perform as good as MCFS except with SVM classifier. Nevertheless, it is worth to point out that the least number of features, m_{least}, attained by SOS-LLS, is not significantly different than that by the MCFS and relatively much less than the number of original features. This can be observed clearly from the results reported in Table 4.5 and Table 4.6, especially for cases where the original datasets have more than 40 features.

By referring to the average results listed in the last row of Table 4.7, it can be concluded that overall SOS-LLS is the winner against Laplacian Score and MMLS if the main interest is to find a smaller feature subset to represent the full feature set closely. In addition, SOS-LLS is generally comparable to MCFS, but it should be emphasized that MCFS can only achieve its best performance when the number of true classes of the dataset is known (Yan & Yang, 2015).

4.5 Summary

A new unsupervised data learning method, called sequential orthogonal search for local largest structure (SOS-LLS), has been introduced for feature selection and ranking. The method exploits the information lie in the first component of LPP and the structure of its mapping function and uses the component as a reference to select significant features that preserves the most important local structure information of the data. A simple yet effective sequential orthogonal feature search strategy has been employed to evaluate the significance of candidate features.

Experiments on two datasets with known data characteristics reveal that SOS-LLS is able to rank features appropriately according to their significance in representing the reference response variable. More experimental results based on twelve datasets clearly show the ability of SOS-LLS to yield small feature subsets that well represent the original full feature set in terms of classification performance. This performance achievement has been verified with four classifiers, each of which has distinct learning mechanism and thereby demonstrates the fitness of use for different problems. Owing to the fact that SOS-LLS largely outperforms the MMLS method, it does reaffirm that *focusing on preserving local structure* is more critical than *preserving the global structure* for unsupervised feature selection.

Chapter 5

Feature Selection based on Kernel Pre-Images

5.1 Introduction

This chapter presents the third feature selection method which utilises kernel pre-images (KPI) to guide the search for significant features in an unsupervised manner under the assumption that data are contaminated by noise. Again, here the same sequential orthogonal search (SOS) strategy as in Chapter 4 is employed but a different implementation procedure based on kernel pre-image approach is proposed to deal with attribute noise. Hence, the new feature selection scheme is referred to as the SOS-KPI method.

Theoretical background and brief overview of the pre-image problem based on kernel PCA are given in Section 5.2 since the idea of this subject forms the basis for the new feature selection method. The proposed method is then presented in detail in Sections 5.3 and 5.4, whereas the experimental setup and procedure employed to evaluate the overall performance of the SOS-KPI method is explained in Section 5.5. Next, the results of the experiments including comprehensive comparison with other state-of-the-art methods are reported and discussed in Section 5.6. The chapter is close with a summary in Section 5.7.

5.2 Kernel PCA and the Pre-Image Problem

Principal component analysis (PCA) is a powerful method that can be used to identify useful patterns in multidimensional datasets by projecting and compressing the data into lower dimensional space with the least possible amount of information loss. In particular, PCA attempts to identify lower dimensional hyperplane that sufficiently describes and represents the data in such a way where the sum of squares of orthogonal deviations (errors) of the data observations from the hyperplane is minimized, or equivalently, the variation of the projections is maximized. As these data projections create new features in lower dimensional space, the method is regarded as an example of feature extraction method. Whilst PCA has been widely used and works fairly well for various applications, it can only identify linear structure of the data and thus prone to loss useful nonlinear structure.

Over the past few decades, there has been a lot of interest on kernel methods in various learning systems for analysing nonlinear patterns. The basic idea of kernel methods is to map nonlinear data that is linearly inseparable in the original input space to a higher dimensional (possibly infinite) feature space where linear separations (or relations) can be achieved. Since the linear geometry of the data in the feature space is embedded in dot products between data instances, the mapping from the original data space to the feature space does not have to be performed explicitly but just needs some defining form of dot products in the original input space. This nonlinear mapping strategy is the so called 'kernel trick', which is the essence of the kernel methods. Taking into advantage of this kernel trick implies that the coordinates of the data in the feature space are not required. Kernel methods are preferable to other nonlinear methods because they do not involve any nonconvex nonlinear optimization procedure but merely require solution for the eigenvalue problem (Kwok & Tsang, 2004), thus the risk of being trapped in local minima can be avoided. This special feature, along with the brilliant idea of kernel approach, have led to many significant research advances such as kernel principal component analysis (kernel PCA) (Scholkopf & Smola, 1997), kernel discriminant analysis (Mika, et al., 1999a; Liu, et al., 2004; Zheng, et al., 2014), kernel-based clustering (Camastra & Verri, 2005; Yin, et al., 2010; Tzortzis & Likas, 2012; Kang, et al., 2017) and kernel regression (Blundell & Duncan, 1998; Yan, et al., 2008; Brouard, et al., 2016).

It is not exaggerate to claim that kernel PCA is one of the most influential kernel-based methods for data dimensionality reduction reported in the literature. Kernel PCA was originally introduced by Scholkopf & Smola (1997) as a nonlinear feature extraction method to overcome the drawback of PCA which can only find linear structure in the data as mentioned earlier. Kernel PCA mimics the underlying concept of PCA but it applies the same linear scheme in the feature space instead of in the input space. Since its introduction, there has been a great deal of attention given to expand the approach for a variety of applications such as image processing (segmentation/face recognition) (Schmidt, et al., 2016), process monitoring (Zhang, et al., 2013; Reynders, et al., 2014; Jaffel, et al., 2017), fault detection (Choi, et al., 2005; Navi, et al., 2015), and forecasting, just to name a few.

In recent years, finding pre-images based on kernel PCA has been proven to be very useful for pattern denoising. Given a noisy pattern $\mathbf{x}$, the first step of the denoising procedure (refer to Figure 5.1) is to map the noisy pattern from the input space into the feature space. The mapping ϕ which is normally nonlinear, utilizes the kernel trick in order to avoid explicit computation relating to mapped shaped vectors in the feature space, so that the entire operations in the feature space can be performed by merely using the dot products. PCA is then applied on the ϕ-mapped pattern, from which the principal directions in the feature space of the input data can be obtained. Next, the ϕ-mapped pattern is further projected onto the subspace spanned by the most significant principal directions which are characterised by the leading eigenvectors. The projection vector onto this subspace, denoted by $P_{\phi(\mathbf{x})}$, can be considered as the sought denoised pattern that retain the main structure of $\mathbf{x}$ while the projection on the complementary space can be regarded as the component that pick up the noise lies in $\mathbf{x}$. The projection $P_{\phi(\mathbf{x})}$, however, is still reside in the feature space and it has to be mapped back to the input space in order to observe its pre-image $\hat{\mathbf{x}}$, that is, the ultimate denoised pattern.

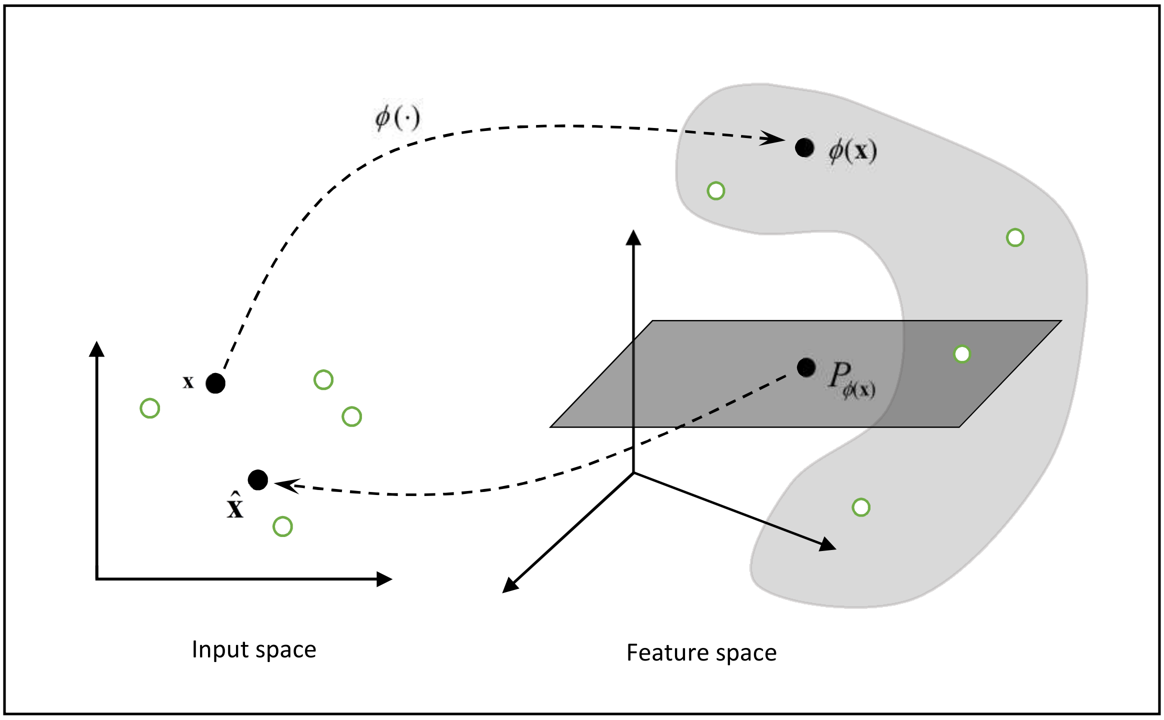

Figure 5.1: Pre-image problem in kernel PCA.

How to obtain the reverse mapping from the feature space back to the input space is often referred to as the “pre-image problem”. A pre-image in a kernel method is therefore can be defined as follows: If $P_{\phi(\mathbf{x})}$ is the projection of $\phi(\mathbf{x})$ onto the kernel principal component subspace in the feature space where $\mathbf{x}$ is a pattern in the input space while ϕ is some map function (usually nonlinear), a pre-image $\hat{\mathbf{x}}$ is a pattern in the input space that corresponds to $P_{\phi(\mathbf{x})}$ such that $\hat{\mathbf{x}} = \phi^{-1}(P_{\phi(\mathbf{x})})$. As the projection $P_{\phi(\mathbf{x})}$ captured the main structure of $\mathbf{x}$, the pre-image $\hat{\mathbf{x}}$ is then can be viewed as a denoised version of $\mathbf{x}$.

The most challenging part of the pre-image problem is that the mapping function from the input space to the feature space is not isomorphic in general (Abrahamsen & Hansen, 2009). Thus, one cannot expect a straightforward solution as the exact pre-image typically does not exist and even if it exists, it is not always unique. In order to alleviate this problem, many methods resort to approximate solution. A prominent pioneer effort in this direction was given by Mika et al. (1999b), who used a gradient decent approach to estimate the pre-image. Yet, the approach is numerically unstable, sensitive to the choice of initial starting point, and generally converge to a local optimum solution (Abrahamsen & Hansen, 2009). To address these problems, an approach using kernel ridge regression was introduced in Weston et al. (2004) but it requires that the training patterns should have a reasonably good distribution to represent the points that will be used to compute the pre-images. In Kwok & Tsang (2004), an approach based on the relationship between feature-space distance and input-space distance together with the idea of multi-dimensional scaling was taken to find the pre-image. By utilizing linear algebra manipulation, this method not only offers non-iterative procedure but it also tackled the problems inherent in the approach taken by Mika et al.(1999b). More recent techniques to estimate the pre-image can be traced from Zheng et al. (2010); Abrahamsen & Hansen (2011); Kallas et al. (2013); Shinde et al. (2014) and Li, et al., (2016).

5.3 Feature Selection Based on Pre-Images of Kernel PCA

This section is mainly devoted to present a new feature selection method dealing with data contaminated by attribute noise from which the search for a subset of relevant features will be performed. A new feature selection method based on pre-images of kernel PCA is introduced towards this goal. In this new method, the feature selection problem is formulated into a multiple linear regression model by considering the pre-images as the dependent (response) variables while all the original features as the independent variables. The key idea underlying the proposed method is to identify features that are significant in characterising the pre-images.

Pre-images are useful as they recover the denoised variation patterns of noisy input data and as such they have the potential to guide the search for significant features. The method is coupled with the sequential orthogonal search strategy so that identifying the significant features can be made in a stepwise manner, one after the other. At each step, the most representative feature to describe the overall variation patterns given by the pre-images is selected.

In principle, the proposed method should work well with any approach for estimating the kernel pre-images. However, the approach developed by Kwok & Tsang (2004) is adopted here as a base technique to perform the estimation due to its aforesaid advantages and widespread usage.

Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be a set of N pattern (observation) vectors in $\mathbb{R}^M$. The proposed feature selection procedure begins at computing the pre-image vector $\hat{\mathbf{x}}_i$ associated to the input pattern vector $\mathbf{x}_i$ for each i, provided that $\phi(\hat{\mathbf{x}}_i) = P_{\phi(\mathbf{x}_i)}$ when the exact pre-image exists or otherwise $\phi(\hat{\mathbf{x}}_i) \approx P_{\phi(\mathbf{x}_i)}$. Essentially, this first step serves as a tool to learn the intrinsic structures within the noisy input patterns and later on recover the denoised variation patterns.

Suppose that the set $F = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\}$ denotes an original dataset of M features in the input space where $\mathbf{f}_j = [f_j(1), f_j(2), \ldots, f_j(N)]^{\mathrm{T}}$ is the jth feature vector formed by N patterns and $[f_1(i), f_2(i), \ldots, f_M(i)]$ is the ith pattern vector. By retaining the same notations used in the preceding paragraph, the ith pattern vector is thus $\mathbf{x}_i = [f_1(i), f_2(i), \ldots, f_M(i)]$.

Let $\hat{\mathbf{x}}_{(k)} = [\hat{x}_k(1), \hat{x}_k(2), \ldots, \hat{x}_k(N)]^{\mathrm{T}}$ denotes a vector formed based on the vector entries of the pre-images $\hat{\mathbf{x}}_i = [\hat{x}_1(i), \hat{x}_2(i), \ldots, \hat{x}_M(i)]$. As such, this yields M unit vectors of $\hat{\mathbf{x}}_{(k)}$. Here, the feature selection approach is formulated as a multiple linear regression problem by setting the vector $\hat{\mathbf{x}}_{(k)}$ as the dependent variable while all the original feature vectors $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M$ as the independent variables. In this formulation, it is assumed that every vector $\hat{\mathbf{x}}_{(k)}$ can be approximated by a linear combination of the M features using the following regression model:

$$\hat{\mathbf{x}}_{(k)} = \sum_{j=1}^{M} \beta_{j,k} \mathbf{f}_j + \mathbf{e}_k \tag{5.1}$$

where $\beta_{j,1}, \beta_{j,2}, \ldots, \beta_{j,M}$ are the regression coefficients while the term $\mathbf{e}_k$ represents the unobservable error of the approximation.

Often, not all of the M features made a significant contribution to the variation in the dependent variable $\hat{\mathbf{x}}_{(k)}$ and some even perhaps redundant with other features. This observation brought up the idea that $\hat{\mathbf{x}}_{(k)}$ can be well approximated by merely relying on a subset of F and thereby feature selection is required to play its role. Let the subset be $S_m = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$ where $\mathbf{z}_j \in F$. As $\hat{\mathbf{x}}_{(k)}$ depends on the subset S_m, the regression model (5.1) can be rewritten as

$$\hat{\mathbf{x}}_{(k)} = \sum_{j=1}^{m} \theta_{j,k} \mathbf{z}_j + \mathbf{e}_k \tag{5.2}$$

This new reduced regression model became the primary reference model for the proposed method in this chapter.

5.4 Monitoring Criterion and Search Procedure

Based on the regression model (5.2), the objective of the proposed feature selection is thus stipulated to select the best feature subset $S_m = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m, \}$ that can represents any response variable vector $\hat{\mathbf{x}}_{(k)}$. In other words, the requirement is to select a feature subset S_m that adequately explains the overall variation in the dependent variables $\hat{\mathbf{x}}_{(k)}$. To fulfil this requirement, an adaptation to the assessment criteria presented in Billings & Wei (2005) and Wei & Billings (2007) is made for the present work.

The reduced regression model (5.2) can be presented in a compact matrix form then

$$\hat{\mathbf{x}}_{(k)} = \mathbf{P}\boldsymbol{\theta}_j + \mathbf{e}_k \tag{5.3}$$

Where $\mathbf{P} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m]$ is a full column rank matrix and $\boldsymbol{\theta}_j = \left[\theta_{1,j}, \theta_{2,j}, \ldots, \theta_{m,j}\right]^{\mathrm{T}}$ is a vector whose elements are regression coefficients. Note that the matrix $\mathbf{P}$ in equation (5.3) can be decomposed into the product of two matrices as

$$\mathbf{P} = \mathbf{QR} \tag{5.4}$$

Where $\mathbf{R}$ is an $m \times m$ upper triangular matrix with unity diagonal elements while $\mathbf{Q}$ is an $N{\times}m$ matrix whose columns correspond to orthogonal vectors $\mathbf{q}_1, \mathbf{q}_2 \ldots, \mathbf{q}_m$. How these orthogonal vectors can be obtained will be explained later. Substituting (5.4) into equation (5.3) and applying some simple algebra gives

$$\hat{\mathbf{x}}_{(k)} = (\mathbf{PR}^{-1})(\mathbf{R}\boldsymbol{\theta}_k) + \mathbf{e}_k = \mathbf{Qg}_k + \mathbf{e}_k \tag{5.5}$$

where $\mathbf{g}_k = \mathbf{R}\boldsymbol{\theta}_k = [g_{k,1}, g_{k,2}, \ldots, g_{k,m}]^{\mathrm{T}}$ is a vector of m orthogonal coefficients. By virtue of the orthogonal property of $\mathbf{Q}$, each coefficient $g_{k,j}$ can be readily computed based on $\hat{\mathbf{x}}_{(k)}$ and $\mathbf{Q}$ as follows:

$$g_{k,j} = (\hat{\mathbf{x}}_{(k)}^{\mathrm{T}}\mathbf{q}_j)/(\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j)\,. \tag{5.6}$$

Using relation (5.6) in equation (5.5), one can then express the total sum of squares (or total variation) of the overall response variable $\hat{\mathbf{x}}_{(k)}$ from the origin as

$$\hat{\mathbf{x}}_{(k)}^{\mathrm{T}}\hat{\mathbf{x}}_{(k)} = \sum_{j=1}^{m} g_{k,j}^2 \mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j + \mathbf{e}_k^{\mathrm{T}}\mathbf{e}_k \tag{5.7}$$

Notice that the total variation consists of two parts. One is the explained variation, given by $\sum_{j=1}^{m} g_{k,j}^2 \mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j$ which is obtained from the relationship of $\hat{\mathbf{x}}_{(k)}$ with $\mathbf{q}_1, \mathbf{q}_2 \ldots, \mathbf{q}_m$ (or equivalently $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m$). Another one is the unexplained variation which is due to chance or error, represented by the term $\mathbf{e}_k^{\mathrm{T}}\mathbf{e}_k$. The explained variation indicates the proportion to which the variation in the dependent variable $\hat{\mathbf{x}}_{(k)}$ is described by the independent variables $\mathbf{q}_1, \mathbf{q}_2 \ldots, \mathbf{q}_m$ Hence, $g_{k,j}^2 \mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j$ is referring to the amount of contribution made by $\mathbf{q}_j$ to the total variation. This idea has led to the concept of error reduction ratio (ERR) obtained by including $\mathbf{q}_j$ (or equivalently $\mathbf{z}_j$) to the model (5.3), which is defined by

$$\mathrm{ERR}(\hat{\mathbf{x}}_{(k)}, \mathbf{q}_j) = \frac{\mathbf{g}_{k,j}(\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j)}{\hat{\mathbf{x}}_{(k)}^{\mathrm{T}}\hat{\mathbf{x}}_{(k)}} = \frac{(\hat{\mathbf{x}}_{(k)}^{\mathrm{T}}\mathbf{q}_j)^2}{(\hat{\mathbf{x}}_{(k)}^{\mathrm{T}}\hat{\mathbf{x}}_{(k)})(\mathbf{q}_j^{\mathrm{T}}\mathbf{q}_j)}\,. \tag{5.8}$$

The above ratio is employed here as an evaluation criterion to measure the significance of a candidate feature in representing the full feature set.

Every $\mathbf{f}_j \in F$ is considered as a candidate feature to be chosen as the most significant feature, $\mathbf{z}_1$. Once $\mathbf{z}_1$ is identified, it is then directly taken as $\mathbf{q}_1$, that is, $\mathbf{q}_1 = \mathbf{z}_1$. As this is the case, the error reduction ratio to be computed in detecting the first significant feature is as below:

$$\mathrm{ERR}(\hat{\mathbf{x}}_{(k)}, \mathbf{f}_j) = \frac{(\hat{\mathbf{x}}_{(k)}^{\mathrm{T}} \mathbf{f}_j)^2}{(\hat{\mathbf{x}}_{(k)}^{\mathrm{T}} \hat{\mathbf{x}}_{(k)})(\mathbf{f}_j^{\mathrm{T}} \mathbf{f}_j)} \tag{5.9}$$

Before the first feature can be selected, the followings are determined:

$$\mathrm{ERR}[k, j; 1] = \mathrm{ERR}(\hat{\mathbf{x}}_{(k)}, \mathbf{f}_j); \ k, j = 1, 2, \ldots, M \tag{5.10}$$

$$\overline{\mathrm{ERR}}[j;1] = \frac{1}{M} \sum_{k=1}^{M} \mathrm{ERR}[k, j; 1] \tag{5.11}$$

$$\ell_1 = \arg \max_{1 \le j \le M} \{\overline{\mathrm{ERR}}\,[j;1]\} \tag{5.12}$$

The first significant feature is then chosen by taking $\mathbf{z}_1 = \mathbf{f}_{\ell_1}$ and the associated orthogonal variable is then set as $\mathbf{q}_1 = \mathbf{z}_1$. Note that $\overline{\mathrm{ERR}}$ is used to measure the percentage of variation in the overall response variables $\hat{\mathbf{x}}_{(k)}$ that can be explained by variable $\mathbf{q}_k$ (which also means the feature vector $\mathbf{z}_k$) individually.

Assume that a subset S of $(r-1)$ features, $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{r-1}$, has already been selected from the full feature set of size $M$and these features have been transformed into a new set of orthogonal variables $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{r-1}$ via some type of orthogonal transformation. In order to select the r th significant feature and add it to the subset S, consider each $\boldsymbol{\alpha}_j \in F - S$. The r th orthogonal variable, $\mathbf{q}_j^{(r)}$, associated to $\boldsymbol{\alpha}_j$ is calculated by

$$\mathbf{q}_j^{(r)} = \boldsymbol{\alpha}_j - \sum_{k=1}^{r-1} \frac{\boldsymbol{\alpha}_j^{\mathrm{T}} \mathbf{q}_k}{\mathbf{q}_k^{\mathrm{T}} \mathbf{q}_k} \mathbf{q}_k \,. \tag{5.13}$$

The followings are then obtained:

$$\mathrm{ERR}[k, j; r] = \mathrm{ERR}(\hat{\mathbf{x}}_{(k)}, \mathbf{q}_j^{(r)}) \tag{5.14}$$

$$\overline{\mathrm{ERR}}[j; r] = \frac{1}{M}\sum_{k=1}^{M} \mathrm{ERR}[k, j; r] \tag{5.15}$$

$$\ell_r = \arg\max_{1 \le j \le M} \{\overline{\mathrm{ERR}}\,[j; r]\}\,. \tag{5.16}$$

Thereby, the rth significant feature can be selected as $\mathbf{z}_r = \mathbf{f}_{\ell_r}$ and its corresponding orthogonal variable is therefore $\mathbf{q}_r = \mathbf{q}_{\ell_r}^{(r)}$.

Subsequent significant features can be found one by one iteratively (also known as sequential search strategy) via the same search procedure as listed from equations (5.13) through (5.16). At each search iteration, any new feature to be selected is the one that supposed to increase the percentage of contribution in explaining the variation in the overall response variables $\hat{\mathbf{x}}_{(k)}$ more than other remaining candidate features. To ease the discussion for the rest of this chapter, the newly proposed feature selection approach is referred to as sequential orthogonal search for kernel pre-images (SOS-KPI) method. The pseudo- code of the SOS-KPI is given in Figure 5.2.

```
Input:       $F = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\}$     // A complete dataset of $M$ features
Output:      $S$                                                            // Subset of features
Initialize:  $L_1 = \{1, 2, \ldots, M\}$, $S = \{\}$
             $m$                                                            // Number of features to be selected

Find $\hat{\mathbf{x}}_{(k)}$ where $k = 1, 2, \ldots, M$                  // As described in Section 5.4

for $j = 1$ to $M$
    for $k = 1$ to $M$
        $\mathrm{ERR}[k, j; 1] = \mathrm{ERR}(\hat{\mathbf{x}}_{(k)}, \mathbf{f}_j)$ ; // As defined by equation (5.9)
    end for
    $\overline{\mathrm{ERR}}[j;1] = \frac{1}{M}\sum_{k=1}^{M}\mathrm{ERR}[k, j; 1]$
end for
$\ell_1 = \arg\max_{j \in L_1}\{\overline{\mathrm{ERR}}[j;1]\}$ such that $\ell_1 \in L_1$; $\mathbf{q}_1 = \mathbf{f}_{\ell_1}$; $\mathbf{z}_1 = \mathbf{f}_{\ell_1}$;
add $\mathbf{z}_1$ to $S$;

for $r = 2$ to $m$
    $L_r = L_{r-1} \setminus \{\ell_{r-1}\}$;
    for $j \in L_r$
        $\mathbf{q}_j^{(r)} = \mathbf{f}_j - \sum_{k=1}^{r-1}\frac{\mathbf{f}_k^{\mathrm{T}}\mathbf{q}_k}{\mathbf{q}_k^{\mathrm{T}}\mathbf{q}_k}$;
        $\mathrm{ERR}[k, j; r] = \mathrm{ERR}(\hat{\mathbf{x}}_{(k)}, \mathbf{q}_j^{(r)})$ where $k = 1, 2, \ldots, M$;
    end for
    $\overline{\mathrm{ERR}}[j;r] = \frac{1}{M}\sum_{k=1}^{M}\mathrm{ERR}[k, j; r]$
    $\ell_r = \arg\max_{j \in L_r}\{\overline{\mathrm{ERR}}[j;r]\}$ such that $\ell_r \in L_r$;
    $\mathbf{q}_r = \mathbf{q}_{\ell_r}^{(r)}$; $\mathbf{z}_r = \mathbf{f}_{\ell_r}$;
    add $\mathbf{z}_r$ to $S$;
end for
```

Figure 5.2: The SOS-KPI algorithm.

5.5 Experimental Setup and Procedure

5.5.1 Modified Benchmark Datasets

In order to evaluate the overall performance of SOS-KPI method, we conducted our simulation experiments on 12 benchmark datasets which are frequently used in the literature. These datasets can be retrieved online from the UCI machine learning repository. We picked the datasets based on three different categories of dimensional size: low-dimension $(M \leq 20)$ medium-dimension $(20 \leq M \leq 100)$, and high-dimension $(M > 100)$. Table 5.1 summarises the important characteristics regarding the used datasets.

Table 5.1: Characteristics of the used benchmark datasets.

Dataset	Number of features	Number of observations	Number of classes
Pima Diabetes	8	768	2
Glass [N]	9	214	7
Vowel [N]	10	990	11
Statlog [N]	18	846	4
Wdbc [N]	30	569	2
Ionosphere	33	351	2
Waveform	40	5000	4
Mfeat Zernike [N]	47	2000	10
Sonar	60	208	2
Musk [N]	166	476	2
Mfeat Factors [N]	216	2000	10
Isolet	649	2000	26

[N]: The raw dataset was normalized before the experiment.

The main objective of SOS-KPI method is to gain a robust method that is less sensitive to attribute noise. However, all twelve datasets we used do not really contain noise. Hence, artificial noise were added into the attributes (or features) of our experimental datasets. It has been proven in Zhu & Wu (2004) and Xiao et al. (2010) that as the attribute noise level goes higher, the classification accuracy tend to be lower. As such, it is not important to make comparison with results when the datasets are clean.

There are two ways how attribute noise is usually distributed in a dataset, one is following Gaussian distribution and the other following uniform distribution. This has led to two common implementations of attribute noise injection, namely *Gaussian attribute noise scheme* and *uniform attribute noise scheme*. In this study, though, we only applied the latter because uniform attribute noise was found to be more disruptive than the Gaussian attribute noise (Saez, et al., 2013; Saez, et al., 2014). Particularly, we performed the same noise injection mechanism adopted by Teng (1999) and Zhu et al. (2004) to include the required uniform attribute noise to the datasets. Two levels of attribute noise are considered: 10% and 20%. According to the noise injection mechanism, approximately $r\%$ of the N observations from each feature vector will be given some random values. Since our datasets only consist of numerical features, the random values are generated between the maximum and minimum values of each feature vector being considered. As this perfectly follows the standard random sampling procedure, every observation is thus has equally likely chance to be injected by noise. Therefore, we only experimented with completely random attribute noise (Howell, 2007), which means noise introduced into an attribute has weak relationship with noise in other attributes.

5.5.2 Comparison with Other Methods

The results obtained based on SOS-KPI method are compared with other state-of-the-art feature selection methods which have been used in the previous chapter: Laplacian Score (LS), Multi-Cluster Feature Selection (MCFS) and Minimum-Maximum Laplacian Score (MMLS). These three methods are used again here for comparison not only because they are promising techniques but also because they involve same kind of feature selection scheme that evaluate features in the input space using unsupervised setting as for SOS-KPI.

5.5.3 Validation Classifiers

As the proposed feature selection method is of filter model, it is therefore imperative to test its versatility across different classifiers that belong to different learning paradigms. Looking at this perspective, four popular classifiers which have been acclaimed as among the ten most influential algorithms in data mining (Wu, et al., 2008) are employed to assess the predictive ability of the feature subsets induced by the proposed method. The classifiers are: *k*-nearest neighbour (*k*-NN), Naïve Bayes (NBayes), support vector machine (SVM), and classification and regression trees (CART).

The number of nearest neighbours of the *k*-NN classifier was set to $k = 5$ for all experiments (i.e., all different combinations of noise level and the tested feature selection method). This setting ensures a fair comparison between the four methods.

5.5.4 Cross-Validation Procedure

To prevent overfitting problem, the classification performance of each generated feature subset was evaluated over 30 rounds of holdout cross validation strategy. In each round, the strategy was set to split randomly 80% of the dataset for training while the remaining 20% were holdout for testing. The classification results are recorded based on the average classification accuracies computed from that 30 rounds of cross-validation.

5.6 Numerical Results and Discussion

Table 5.2 through Table 5.5 show the least number of selected features, m_{least}, achieved by different feature selection methods that gives classification accuracy more than or close to the one obtained by using the full feature set with at most 5% less than the figure recorded for full feature set. Each m_{least} was determined using a one-tailed two-sample z-test, comparing the average classification accuracy yielded by the full feature set to the average classification accuracy given by the targeted feature subset. Results in Table 5.2 through Table 5.5 are marked with '●' if the SOS-KPI method is statistically superior to the compared method whereas the symbol '□' is reserved to denote that the SOS-KPI method is statistically inferior to the compared method.

Results from Table 5.2 through Table 5.5 are then summarised into Table 5.6 through Table 5.9, so as to demonstrate the potential of the proposed SOS-KPI to produce optimal feature subset in representing the full feature set. Particularly, the information provided in Table 5.6 and Table 5.7 are aimed to gain an insight on how well the SOS-KPI method performs for the three specified categories of dimensional size. In the meantime, Table 5.8 and Table 5.9 are useful mainly for demonstrating the feasibility of SOS-KPI as a robust filter feature selection that capable to perform with different classifiers. The win/tie/loss scores recorded in Table 5.6 through Table 5.9 are referring to the number of test datasets for which the SOS-KPI method yields lower/equal/higher subset size compared against other feature selection methods.

As can been seen from Table 5.6, the SOS-KPI method shows higher performance than the all three methods in comparison for moderate and high dimensional sizes when 10% of attribute noise was added to the datasets. Considering the low dimensional size category, the proposed SOS-KPI method only loses to Laplacian Score.

It appears from Table 5.7 that the SOS-KPI method also performs better than the others for moderate and high dimensional sizes when test datasets were corrupted with 20% of attribute noise. However, the proposed method has been defeated by all of its rivals for low dimensional size category as 20% of attribute noise occurred.

From Table 5.8, it is clear that the SOS-KPI outperforms other methods for all four classifiers considered with 10% of attribute noise. When 20% of attribute noise was introduced into the benchmark datasets, the SOS-KPI method is just slightly inferior to Laplacian Score and MCFS with CART classifier yet it surpasses for other cases, as reported in Table 5.9. The average results provided in the final row of both Table 5.8 and Table 5.9 indicate that the SOS-KPI method is more robust against attribute noise than the other competing methods in overall if a small feature subset is desired to represent the original feature set.

Figure 5.3 shows comparison of the win/tie/loss cumulative counts of the SOS-KPI methods against LS, MCFS and MMLS when dealing with different categories of dimensional size. It can be observed that the SOS-KPI method performs very well in overall for moderate $(20 \leq M \leq 100)$ and high $(M > 100)$ dimensional sizes but it is slightly inferior when low $(M \leq 20)$ dimensional size is considered.

Table 5.2: The least number of selected features, m_{least}, induced by SOS-KPI, LS, MCFS and MMLS methods that gives classification accuracy close to (at most 5% less than the full set accuracy) or better than the full feature set. The symbol "●" (or "□") denotes the proposed method has lower (or larger) value of m_{least} than the compared method. Results are based on Pima Diabetes, Glass and Vowel datasets.

	Pima Diabetes				Glass				Vowel			
5-NN	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	2	69.96 ± 1.30	2	71.37 ± 1.29	3	55.63 ± 2.90	4	47.22 ± 3.29	6	63.67 ± 1.27	4	44.38 ± 1.18
LS	2	69.59 ± 1.15	2	71.94 ± 1.16	5 ●	56.43 ± 3.04	5 ●	47.54 ± 2.18	4 □	63.79 ± 1.23	3 □	46.50 ± 0.94
MCFS	4 ●	70.92 ± 1.20	2	66.97 ± 1.31	6 ●	55.56 ± 2.78	4	47.38 ± 1.99	7 ●	63.10 ± 1.52	6 ●	48.42 ± 0.98
MMLS	3 ●	68.06 ± 1.26	2	71.83 ± 1.19	4 ●	56.98 ± 2.26	3 □	47.13 ± 2.23	4 □	62.63 ± 1.10	3 □	45.71 ± 1.31
Full set		71.26 ± 1.31		70.54 ± 1.01		56.83 ± 2.25		47.62 ± 2.25		65.24 ± 1.38		47.44 ± 1.10
NBayes	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	2	71.29 ± 1.08	2	68.91 ± 1.00	6	55.71 ± 2.31	9	50.40 ± 2.79	6	58.27 ± 1.24	6	50.25 ± 1.00
LS	2	71.90 ± 0.91	2	68.39 ± 0.84	8 ●	59.60 ± 2.63	9	50.40 ± 2.79	9 ●	61.80 ± 1.34	6	49.18 ± 1.34
MCFS	4 ●	71.85 ± 1.05	2	71.20 ± 0.79	7 ●	56.90 ± 3.06	7 □	48.65 ± 2.81	8 ●	59.85 ± 1.04	7 ●	49.58 ± 1.11
MMLS	3 ●	73.20 ± 1.26	2	68.63 ± 1.13	5 □	62.22 ± 1.74	5 □	51.11 ± 2.40	8 ●	58.48 ± 1.45	7 ●	50.51 ± 1.54
Full set		73.57 ± 1.44		71.44 ± 1.07		57.70 ± 2.84		50.40 ± 2.79		61.41 ± 1.28		51.72 ± 1.14
SVM	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	2	72.64 ± 1.08	2	69.35 ± 1.03	8	57.30 ± 2.47	7	47.14 ± 2.45	6	52.39 ± 1.18	6	43.00 ± 1.13
LS	2	72.46 ± 0.88	2	69.67 ± 1.14	7 □	58.10 ± 2.21	4 □	47.06 ± 2.27	4 □	49.90 ± 1.12	2 □	40.99 ± 1.22
MCFS	4 ●	73.59 ± 0.86	2	69.15 ± 0.96	6 □	57.94 ± 2.05	5 □	47.70 ± 2.80	7 ●	52.14 ± 1.35	6	38.43 ± 1.08
MMLS	3 ●	72.77 ± 1.13	2	68.89 ± 1.19	6 □	56.98 ± 2.20	9 ●	48.89 ± 2.50	4 □	50.84 ± 1.09	2 □	40.34 ± 1.21
Full set		74.79 ± 1.15		69.69 ± 1.43		59.13 ± 2.19		48.89 ± 2.50		53.57 ± 1.11		41.75 ± 0.94
CART	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	3	65.84 ± 1.53	2	67.95 ± 1.17	5	57.62 ± 3.32	7	48.41 ± 2.26	6	55.86 ± 1.01	5	40.61 ± 1.01
LS	3	65.42 ± 1.22	2	67.49 ± 1.20	5	58.65 ± 2.29	5 □	52.94 ± 2.41	3 □	53.65 ± 1.20	2 □	40.02 ± 1.09
MCFS	4 ●	66.10 ± 1.29	2	65.40 ± 1.24	6 ●	60.87 ± 2.42	5 □	47.85 ± 1.15	7 ●	56.72 ± 1.14	6 ●	44.14 ± 1.36
MMLS	4 ●	65.08 ± 1.29	2	67.30 ± 1.22	4 □	58.89 ± 2.13	4 □	51.56 ± 1.23	3 □	53.30 ± 1.21	2 □	39.90 ± 0.92
Full set		68.04 ± 1.31		66.27 ± 1.29		58.17 ± 2.68		50.16 ± 2.64		55.57 ± 1.57		42.76 ± 1.07

Table 5.3: The least number of selected features, m_{least}, induced by SOS-KPI, LS, MCFS and MMLS methods that gives classification accuracy close to (at most 5% less than the full set accuracy) or better than the full feature set. The symbol "●" (or "□") denotes the proposed method has lower (or larger) value of m_{least} than the compared method. Results are based on Statlog, Wdbc and Ionosphere datasets.

	Statlog				Wdbc				Ionosphere			
5-NN	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	5	55.76 ± 1.11	4	43.67 ± 1.40	14	89.20 ± 0.99	2	84.45 ± 1.06	5	82.24 ± 1.15	10	78.95 ± 0.92
LS	4 □	56.67 ± 0.92	4	47.65 ± 1.37	9 □	88.82 ± 1.10	12 ●	85.63 ± 1.29	16 ●	83.33 ± 1.26	14 ●	79.86 ± 1.49
MCFS	4 □	57.46 ± 1.27	4	46.77 ± 1.09	8 □	88.70 ± 1.02	16 ●	84.22 ± 1.07	14 ●	82.43 ± 1.41	15 ●	78.29 ± 1.47
MMLS	5	56.51 ± 1.18	3 □	45.19 ± 1.19	3 □	90.38 ± 1.02	4 ●	84.78 ± 1.17	15 ●	81.29 ± 1.59	16 ●	80.52 ± 1.03
Full set		57.71 ± 1.30		47.02 ± 1.25		91.95 ± 0.92		87.55 ± 0.91		84.29 ± 1.13		81.52 ± 1.12
NBayes	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	6	58.97 ± 1.07	10	55.42 ± 1.15	7	89.38 ± 0.99	5	89.62 ± 1.19	19	88.67 ± 1.46	18	85.81 ± 1.48
LS	4 □	57.67 ± 1.10	6 □	54.44 ± 1.30	9 ●	87.99 ± 1.26	12 ●	90.41 ± 1.15	25 ●	86.76 ± 1.58	26 ●	85.71 ± 1.59
MCFS	5 □	55.72 ± 1.26	4 □	54.46 ± 1.46	8 ●	87.52 ± 1.08	16 ●	90.27 ± 0.97	27 ●	85.67 ± 1.74	29 ●	87.71 ± 1.35
MMLS	12 ●	56.31 ± 1.38	4 □	54.16 ± 1.14	3 □	90.65 ± 0.77	9 ●	90.06 ± 0.99	21 ●	85.48 ± 1.55	23 ●	86.05 ± 1.71
Full set		57.65 ± 1.10		57.65 ± 1.19		91.30 ± 0.86		93.33 ± 0.69		88.57 ± 1.07		88.33 ± 1.82
SVM	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	6	51.79 ± 1.53	9	43.81 ± 1.35	16	91.53 ± 0.96	6	85.78 ± 0.82	13	80.76 ± 1.82	11	76.29 ± 1.43
LS	13 ●	52.35 ± 1.31	10 ●	42.74 ± 1.07	14 □	91.24 ± 0.88	12 ●	85.37 ± 0.93	17 ●	80.33 ± 1.17	13 ●	75.62 ± 1.28
MCFS	11 ●	54.99 ± 1.24	8 □	42.78 ± 1.21	11 □	90.03 ± 0.96	18 ●	86.02 ± 1.15	27 ●	80.76 ± 1.43	18 ●	75.52 ± 1.38
MMLS	9 ●	56.39 ± 1.18	9	43.73 ± 0.95	5 □	90.35 ± 1.00	8 ●	86.70 ± 1.13	15 ●	81.57 ± 1.49	14 ●	75.86 ± 1.44
Full set		55.31 ± 1.04		46.37 ± 1.12		93.66 ± 0.89		89.29 ± 0.94		83.57 ± 1.50		78.29 ± 1.91
CART	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	5	57.67 ± 1.26	8	52.39 ± 0.83	7	86.78 ± 1.11	5	86.76 ± 1.31	4	82.10 ± 1.70	16	79.05 ± 1.51
LS	4 □	60.18 ± 1.31	6 □	53.87 ± 1.10	6 □	86.93 ± 1.03	9 ●	87.88 ± 0.89	16 ●	82.90 ± 1.81	17 ●	80.48 ± 1.57
MCFS	4 □	60.71 ± 1.47	5 □	54.81 ± 1.39	3 □	87.32 ± 1.15	12 ●	86.22 ± 1.17	16 ●	81.62 ± 1.21	18 ●	80.90 ± 1.92
MMLS	8 ●	57.51 ± 1.42	4 □	57.40 ± 1.21	3 □	86.99 ± 1.05	4 □	86.31 ± 1.20	16 ●	82.67 ± 1.97	17 ●	82.81 ± 1.95
Full set		60.65 ± 1.31		55.25 ± 1.35		89.14 ± 0.83		89.12 ± 1.20		84.76 ± 1.78		82.19 ± 1.52

Table 5.4: The least number of selected features, m_{least}, induced by SOS-KPI, LS, MCFS and MMLS methods that gives classification accuracy close to (at most 5% less than the full set accuracy) or better than the full feature set. The symbol "●" (or "□") denotes the proposed method has lower (or larger) value of m_{least} than the compared method. Results are based on Waveform, Mfeat Zernike and Sonar datasets.

	Waveform				Mfeat Zernike				Sonar			
5-NN	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	6	73.60 ± 0.49	8	66.52 ± 0.56	30	68.49 ± 0.74	37	55.88 ± 0.77	11	72.85 ± 2.36	10	67.80 ± 2.20
LS	10 ●	73.32 ± 0.43	8	65.78 ± 0.52	42 ●	67.65 ± 0.86	43 ●	56.05 ± 0.60	36 ●	72.36 ± 2.10	31 ●	66.83 ± 2.40
MCFS	10 ●	73.28 ± 0.36	5 □	66.93 ± 0.53	28 □	67.36 ± 0.94	34 □	55.77 ± 0.74	34 ●	72.68 ± 2.62	22 ●	66.59 ± 2.14
MMLS	10 ●	73.28 ± 0.36	9 ●	66.51 ± 0.48	44 ●	68.47 ± 0.59	43 ●	55.78 ± 0.85	50 ●	71.79 ± 2.00	26 ●	67.07 ± 2.15
Full set		76.40 ± 0.43		69.96 ± 0.54		71.20 ± 0.89		59.60 ± 0.85		73.74 ± 2.53		69.02 ± 2.18
NBayes	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	7	74.76 ± 0.41	14	73.19 ± 0.38	16	63.91 ± 0.60	33	63.24 ± 0.67	5	66.67 ± 2.08	18	71.14 ± 2.02
LS	11 ●	75.83 ± 0.44	10 □	72.75 ± 0.48	39 ●	64.15 ± 0.73	39 ●	62.76 ± 0.86	8 ●	67.97 ± 2.49	36 ●	71.63 ± 2.65
MCFS	7	75.97 ± 0.43	7 □	73.30 ± 0.49	19 ●	65.37 ± 0.77	30 □	63.17 ± 0.80	10 ●	68.54 ± 1.75	34 ●	71.14 ± 2.83
MMLS	13 ●	74.67 ± 0.37	14	74.43 ± 0.45	40 ●	65.99 ± 0.83	42 ●	63.57 ± 0.81	30 ●	68.86 ± 2.53	42 ●	71.79 ± 2.47
Full set		78.68 ± 0.38		77.08 ± 0.40		67.95 ± 0.78		66.70 ± 0.63		69.67 ± 2.12		73.17 ± 2.34
SVM	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	7	74.74 ± 0.32	15	70.07 ± 0.34	25	67.04 ± 0.72	36	57.53 ± 0.84	30	73.33 ± 2.11	19	70.08 ± 2.28
LS	11 ●	77.26 ± 0.34	10 □	70.00 ± 0.39	39 ●	67.43 ± 0.67	39 ●	57.90 ± 0.59	36 ●	71.71 ± 2.28	33 ●	69.92 ± 1.93
MCFS	7	75.06 ± 0.55	9 □	70.31 ± 0.47	26 ●	66.92 ± 0.64	26 □	57.32 ± 0.76	30	72.03 ± 1.98	34 ●	69.76 ± 2.34
MMLS	11 ●	74.79 ± 0.49	13 □	70.61 ± 0.42	39 ●	66.68 ± 0.75	40 ●	58.19 ± 0.83	59 ●	72.68 ± 2.66	53 ●	67.07 ± 2.14
Full set		79.12 ± 0.49		74.45 ± 0.45		70.62 ± 0.75		61.23 ± 0.92		73.66 ± 2.49		69.27 ± 2.46
CART	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	6	68.84 ± 0.47	9	63.25 ± 0.63	19	54.92 ± 0.93	17	43.76 ± 0.86	10	69.51 ± 2.47	18	67.48 ± 3.00
LS	9 ●	65.72 ± 0.45	8 □	60.50 ± 0.54	29 ●	53.51 ± 0.84	31 ●	43.33 ± 0.91	29 ●	67.24 ± 2.33	17 □	63.01 ± 2.70
MCFS	9 ●	65.69 ± 0.43	5 □	61.27 ± 0.46	17 □	53.69 ± 0.73	15 □	43.83 ± 0.76	7 □	71.63 ± 2.40	11 □	64.80 ± 2.59
MMLS	9 ●	65.69 ± 0.43	9	61.75 ± 0.47	38 ●	53.94 ± 0.97	36 ●	44.09 ± 0.77	60 ●	69.27 ± 2.60	51 ●	63.90 ± 2.05
Full set		69.98 ± 0.50		64.85 ± 0.47		57.27 ± 0.81		47.19 ± 1.00		69.27 ± 2.60		64.63 ± 2.96

Table 5.5: The least number of selected features, m_{least}, induced by SOS-KPI, LS, MCFS and MMLS methods that gives classification accuracy close to (at most 5% less than the full set accuracy) or better than the full feature set. The symbol "●" (or "□") denotes the proposed method has lower (or larger) value of m_{least} than the compared method. Results are based on Musk, Mfeat Factors and Isolet datasets.

	Musk				Mfeat Factors				Isolet			
5-NN	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	95	76.42 ± 1.25	6	64.98 ± 1.64	33	90.98 ± 0.49	80	89.23 ± 0.54	200	81.78 ± 0.28	310	77.87 ± 0.37
LS	140 ●	76.39 ± 1.65	120 ●	65.26 ± 2.04	70 ●	90.67 ± 0.48	100 ●	88.67 ± 0.39	250 ●	82.43 ± 0.33	300 □	77.76 ± 0.32
MCFS	115 ●	76.77 ± 1.21	70 ●	65.00 ± 1.38	36 ●	90.60 ± 0.38	70 □	89.30 ± 0.40	210 ●	82.01 ± 0.31	270 □	78.31 ± 0.30
MMLS	100 ●	76.53 ± 1.45	65 ●	64.84 ± 1.50	100 ●	91.45 ± 0.52	160 ●	89.57 ± 0.63	410 ●	82.05 ± 0.32	400 ●	78.00 ± 0.34
Full set		79.47 ± 1.57		68.11 ± 1.27		95.07 ± 0.39		93.12 ± 0.40		86.45 ± 0.29		82.32 ± 0.28
NBayes	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	14	67.05 ± 1.99	12	64.77 ± 1.44	23	88.32 ± 0.43	40	89.07 ± 0.61	85	73.41 ± 0.35	180	74.25 ± 0.70
LS	100 ●	66.42 ± 1.48	110 ●	64.95 ± 1.78	100 ●	88.97 ± 0.41	90 ●	88.13 ± 0.54	220 ●	73.48 ± 0.34	230 ●	73.92 ± 0.53
MCFS	35 ●	67.30 ± 1.81	50 ●	64.35 ± 1.93	21 □	88.21 ± 0.53	50 ●	88.92 ± 0.55	320 ●	73.73 ± 0.36	350 ●	74.20 ± 0.38
MMLS	95 ●	68.32 ± 1.60	105 ●	65.44 ± 1.66	120 ●	88.12 ± 0.56	130 ●	88.91 ± 0.59	280 ●	74.61 ± 0.36	250 ●	73.77 ± 0.40
Full set		69.35 ± 1.97		66.95 ± 1.51		92.35 ± 0.49		92.54 ± 0.48		77.99 ± 0.37		78.17 ± 0.49
SVM	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	18	65.37 ± 1.59	11	61.30 ± 1.52	70	92.64 ± 0.41	110	89.34 ± 0.41	200	88.58 ± 0.24	310	85.74 ± 0.28
LS	95 ●	65.65 ± 1.60	110 ●	61.65 ± 1.81	100 ●	91.36 ± 0.42	140 ●	89.93 ± 0.55	270 ●	88.37 ± 0.25	320 ●	86.41 ± 0.27
MCFS	45 ●	65.86 ± 1.73	29 ●	61.40 ± 1.98	60 □	91.71 ± 0.52	120 ●	89.76 ± 0.48	230 ●	88.57 ± 0.24	300 □	86.03 ± 0.31
MMLS	125 ●	65.89 ± 1.69	125 ●	61.51 ± 1.72	130 ●	92.20 ± 0.39	150 ●	89.05 ± 0.62	350 ●	88.48 ± 0.25	400 ●	86.06 ± 0.37
Full set		68.21 ± 1.64		64.32 ± 1.59		95.71 ± 0.30		93.37 ± 0.42		93.08 ± 0.24		90.30 ± 0.22
CART	10% Noise		20% Noise		10% Noise		20% Noise		10% Noise		20% Noise	
	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy	m_{least}	Subset Accuracy
SOS-KPI	75	71.47 ± 1.59	12	63.75 ± 1.56	11	70.17 ± 0.97	14	60.46 ± 1.01	75	60.27 ± 0.54	240	50.53 ± 0.48
LS	65 □	71.75 ± 2.07	50 ●	63.61 ± 1.46	40 ●	69.97 ± 0.77	40 ●	60.96 ± 1.05	110 ●	60.37 ± 0.51	115 □	50.41 ± 0.37
MCFS	70 □	71.30 ± 1.97	19 ●	63.37 ± 1.76	15 ●	69.73 ± 0.67	40 ●	61.36 ± 0.93	95 ●	60.36 ± 0.44	140 □	51.09 ± 0.37
MMLS	70 □	72.42 ± 2.15	55 ●	65.02 ± 2.11	70 ●	71.55 ± 0.84	50 ●	59.69 ± 0.75	210 ●	60.38 ± 0.47	250 ●	50.43 ± 0.48
Full set		74.21 ± 1.58		65.75 ± 2.16		73.59 ± 0.86		63.39 ± 0.80		64.56 ± 0.48		54.62 ± 0.56

Table 5.6: A comparison of the win/tie/loss counts of the SOS-KPI method against other methods for different categories of dimensional size. The counts are based on the results presented in Table 5.2 through Table 5.5 when the datasets are corrupted with 10% of attribute noise and considering all four classifiers.

Win/tie/lose	LS	MCFS	MMLS
Low dimension	4 / 5 / 7	12 / 0 / 4	9 / 1 / 6
Moderate dimension	17 / 0 / 3	11 / 3 / 6	16 / 0 / 4
High dimension	11 / 0 / 1	9 / 0 / 3	11 / 0 / 1

Table 5.7: A comparison of the win/tie/loss counts of the SOS-KPI method against other methods for different categories of dimensional size. The counts are based on the results presented in Table 5.2 through Table 5.5 when the datasets are corrupted with 20% of attribute noise and considering all four classifiers.

Win/tie/lose	LS	MCFS	MMLS
Low dimension	2 / 7 / 7	3 / 7 / 6	2 / 5 / 9
Moderate dimension	15 / 1 / 4	11 / 0 / 9	16 / 2 /2
High dimension	10 / 0 / 2	8 / 0 / 4	12 / 0 / 0

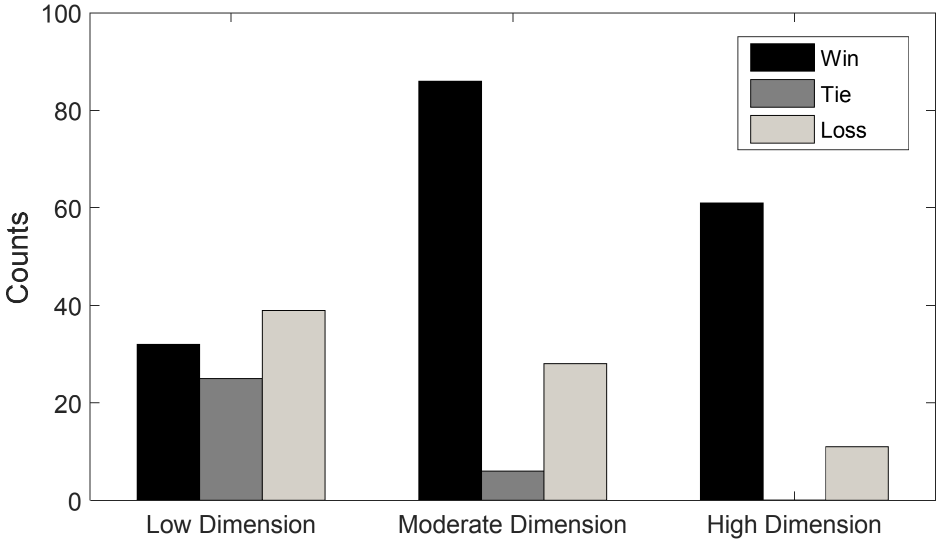

Figure 5.3: Comparison of the total win/tie/loss counts of the SOS-KPI method versus other methods according to different categories of dimensional size.

Table 5.8: A comparison of the win/tie/loss counts of the SOS-KPI method against other methods. The counts are based on the results presented in Table 5.2 through Table 5.5 when the datasets are corrupted with 10% attribute noise.

Win/tie/lose	LS	MCFS	MMLS
5-NN	8 / 1 / 3	9 / 0 / 3	9 / 1 / 2
NBayes	10 / 1 / 1	8 / 1 / 3	11 / 0 / 1
SVM	8 / 1 / 3	7 / 2 / 3	9 / 0 / 3
CART	6 / 2 / 4	7 / 0 / 5	8 / 0 / 4
Average	8 / 1 / 3	7.75 / 0.75 / 3.5	9 / 0.25 / 2.75

Table 5.9: A comparison of the win/tie/loss counts of the SOS-KPI method against other methods. The counts are based on the results presented in Table 5.2 through Table 5.5 when the datasets are corrupted with 20% attribute noise.

Win/tie/lose	LS	MCFS	MMLS
5-NN	7 / 3 / 2	5 / 3 / 4	8 / 1 / 3
NBayes	7 / 3 / 2	7 / 1 / 4	8 / 1 / 3
SVM	9 / 1 / 2	6 / 2 / 4	7 / 2 / 3
CART	5 / 1 / 6	5 / 1 / 6	6 / 2 / 4
Average	7 / 2 / 3	5.75 / 1.75 / 4.5	7.25 / 1.5 / 3.25

5.7 Summary

Numerous feature selection techniques found in the literature focus on the case when noise-free data are available. In fact, among the efforts considering noisy data in feature selection, many have been directed to address the problems of class noise. In practice, however, the data are often found not only containing irrelevant features but also corrupted with attribute noise. Since very limited works have been done considering attribute noise, a feature selection method called *sequential orthogonal search for kernel pre images* (SOS-KPI) is thus introduced.

Pre-images are interesting as they recover the denoised variation patterns of the noisy input data. The basic idea of the SOS-KPI method is therefore to identify features that are significant in characterising the pre-images. The same sequential orthogonal search strategy as in the SOS-LLS method is also applied for the SOS-KPI method to identify significant features but a somewhat different formulation is imposed according to the specific context being considered where noisy data are observed.

Experiments performed on 12 benchmark datasets that have been injected with attribute noise show that the proposed SOS-KPI method is competitive to the state-of-the-art methods. There are three important findings have emerged from the experiments. First, the SOS-KPI method is indeed less sensitive to attribute noise. Second, better performance achievement by the SOS-KPI method demonstrated through application with different classifiers suggests its adaptive flexibility as a filter feature selection approach. Finally, which is the third, the SOS-KPI particularly shows its best performance when moderate $(20 < M \leq 100)$ and high $(M > 100)$ dimensional sizes are considered.

Chapter 6

Conclusion

6.1 Research Summary and Conclusion

Technological advancement in data storage has led to the explosive growth in size of massive datasets which are usually of high dimensional with redundant and irrelevant features. Modelling high dimensional data is often computationally expensive and good predictive models are difficult to obtain because datasets may contain a large number redundant and irrelevant features. Thus, dimensionality reduction is seen as a crucial pre-processing step to overcome these problems, and one approach to achieve this is through feature selection.

Guided by extensive literature review, three research opportunities were explored to address some important issues in feature selection. Three research objectives were set.

The first research objective led to a feature selection method with a new evaluation criterion called maximum relevance–minimum multicollinearity (MRmMC) is being proposed. This newly proposed method was designed to overcome some problems associated with existing methods that apply the same form of feature selection criterion, especially those that are based on mutual information. Rather than using mutual information as the basis for the evaluation criterion, the MRmMC method adopts correlation coefficient from conditional variance to measure feature relevance, and an orthogonal projection scheme based on multiple correlation coefficient is employed to quantify feature redundancy. Unlike mutual information based feature selection, the new method has the advantage of not demanding any control parameters, thereby preventing any uncertainty associated with it.

The second research objective is achieved by introducing a new unsupervised feature selection method, namely, sequential orthogonal search for local largest structure (SOS-LLS). The method is designed to utilise the underlying information captured by LPP approach where the first component of LPP that preserves the most important local structure of the data is used as a reference to select significant features. As the SOS-LLS has largely outperforms the

MMLS method, it does reaffirm that focusing on preserving local structure is more critical than preserving the global structure for unsupervised feature selection.

The third research objective is accomplished by presenting another new unsupervised feature selection method named as sequential orthogonal search for kernel pre-images (SOS-KPI). This feature selection method attempts to offer a robust method that is less sensitive to attribute noise. Towards this goal, the kernel pre-images is exploited as the main reference to identify significant features because pre-images are seen offering the denoised variation patterns of the noisy input data. Even though the SOS-KPI method has been shown to work well with moderate $(20 < M \leq 100)$ and high $(M > 100)$ dimensional sizes, not with low- $(M \leq 20)$ dimensional size category, this should not be a serious practical limitation since feature selection main goal is apparently more critical to reduce higher dimensional sizes.

Note that the three proposed methods employed similar feature search strategy implemented by means of a sequential orthogonalization scheme. Each method, however, applied this feature search scheme differently according to specific mathematical formulation involved that suits the context of feature selection problem being considered. The MRmMC method is specifically devised to select a significant feature subset by utilising the information from both input features and class labels, whereas the SOS-LLS and SOS-KPI methods are forced to depend merely on information obtained from input features. The SOS-LLS and SOS-KPI are useful in cases where class labels are absence, probably due to the fact that class labels acquisition is costly and time-consuming. The SOS-KPI method, however, distinct from the SOS-LLS method as the SOS-KPI method is intended to provide a feature selection approach that has stronger noise resistance ability.

The sequential orthogonalization search scheme which selects significant feature in a stepwise wise iterative fashion, one feature at a time, coupled with a straightforward measurement criterion makes each proposed method easy to implement and suitable to be applied in many applications. All of the three methods are also based on the same feature selection model, which is filter approach, as they merely rely on characteristics of the data without involving any specific classification algorithms in the selection process. Therefore, they work well with different types of classification algorithms such as k-NN, Naïve Bayes, SVM and CART. Despite the advantages offered by such feature selection design, the proposed method however, may not always find the optimal feature subset as the search is non-exhaustive. Nevertheless, from experimental studies performed separately for each method show that all three proposed methods are functionally competent for feature selection based on their own unique goal and context.

6.2 Future Direction of the Research

In light of the present work, a number of new research directions will be explored. The list is as follows.

(i) **Expansion of the MRmMC method**: A limitation of MRmMC is that the proposed redundancy measure is reliable for quantitative features, but cannot effectively evaluate the redundancy between a quantitative and a nominal random variable. It is of interest to make use of other measures to assess feature redundancy and combine this idea with the feature relevancy measure applied in this research study. The combination is expected to form a new criterion that can be used to effectively deal with both nominal and quantitative features. It would be also interesting to explore the new criterion with other feature search strategies such as floating search selection and nature-inspired selection in order to find better feature subset solutions.

(ii) **Expansion of the SOS-LLS method**: It is of interest in future work to explore how smaller sample size affects the effectiveness of the proposed approach. While the results indicate that the sequential search strategy works well, it sometimes generates sub-optimal performance. Future research should therefore be focusing on enhancing the present approach by combining it with other search strategies (e.g. the bagging method based on distance correlation metric proposed in (Solares & Wei, 2015) so as to lead to more significant feature subset solutions. It would be also interesting to consider other projection schemes to replace or combine with LPP to define more powerful reference response variables.

(iii) **Expansion of the SOS-KPI method:** From many literature reports, it has been highlighted that the effect of class noise is more severe than attribute noise However, through SOS-KPI method, one can observe that handling attribute noise in a feature selection technique may lead to significant improvement in classification performance. Realizing the notable effect of attribute noise on feature selection solution, it would be interesting to conduct further research that will improve the SOS-KPI by taking into account both class noise and attribute noise. It is also desirable to consider class imbalance effect for future work. In addition, it is believe that a research on sensitivity of non-representative attribute noise also should be performed.

Exploring the above listed future works should address some open problems in feature selection research specifically and dimensionality reduction generally.

References

Abandah, G. A. & Malas, T. M., 2010. Feature selection for recognizing handwritten arabic letters. *Dirasat Engineering Sciences Journal,* 37(2), pp. 1-21.

Abrahamsen, T. J. & Hansen, L. K., 2009. *Input space regularization stabilizes pre-images for kernel PCA de-noising.* Grenoble, France, Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing, pp. 1-6.

Abrahamsen, T. J. & Hansen, L. K., 2011. Regularized pre-image estimation for kernel PCA de-noising. *Journal of Signal Processing Systems,* 65(3), pp. 403-412.

Aha, D. & Bankert, R. L., 1996. A comparative evaluation of sequential feature selection algorithms. In: D. Fisher & H. J. Lenz, eds. *Learning from Data. Lecture Notes in Statistics.* New York, USA: Springer-Verlag, pp. 199-206.

Altidor, W., Khoshgoftaar, T. M. & Van Hulse, J., 2011. *Robustness of filter-based feature ranking: a case study.* Florida, USA, Proceedings of the 24 International Florida Artificial Intelligence Research Society Conference, pp. 453-458.

Balakrishnama, S. & Ganapathiraju, A., 1998. Linear discriminant analysis- a brief tutorial. *Institute for Signal and Information Processing,* Volume 18, pp. 1-8.

Banka, H. & Dara, S., 2015. A hamming distance based particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation. *Pattern Recognition Letters,* Volume 52, pp. 94-100.

Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Network,* 5(4), pp. 537-550.

Belkin, M. & Niyogi, P., 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems,* pp. 585-591.

Belkin, M. & Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation,* 15(6), pp. 1373-1396.

Bennett, A., 2017. *Highest Paying Big Data Jobs in 2017.* [Online] Available at: https://www.cbronline.com/big-data/highest-paying-big-data-jobs-2017/ [Accessed 15 April 2018].

Bertrand, A. & Moonen, M., 2013. Distributed computation of the Fielder vector with application to topology inference in ad hoc networks. *Signal Processing,* 93(5), pp. 1106-1117.

Bhadani, A. & Jothimani, D., 2016. Big data: Challenges, opportunities and realities. In: M. K. Singh & D. G. Kumar, eds. *Effective Big Data Management and Opportunities for Implementation.* Pennsylvania, USA: IGI Global, pp. 1-24.

Billings, S. A., 2013. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains.* West Sussex, United Kingdom: John Wiley & Sons.

Billings, S. A. & Wei, H. L., 2005. *A multiple sequential orthogonal least squares algorithm for feature ranking and subset selection,* Sheffield, UK: University of Sheffield.

Billings, S., Chen, S. & Korenberg, M., 1989. Identification of MIMO non-linear systems using a forward-regression orthogonal estimator. *International Journal of Control,* 49(6), pp. 2157-2189.

Bjorck, A., 1994. Numerics of gram-schmidt orthogonalization. *Linear Algebra and Its Applications,* Volume 197, pp. 297-316.

Blundell, R. & Duncan, A., 1998. Kernel regression in empirical microeconomics. *Journal of Human Resources,* 33(1), pp. 62-87.

Breiman, L., 2001. Random forests. *Machine Learning,* 45(1), pp. 5-32.

Brouard, C., Szafranski, M. & d'Alché-Buc, F., 2016. Input output kernel regression supervised and semi-supervised structured output prediction with operator-valued kernels. *Journal of Machine Learning Research,* 17(176), pp. 1-48.

Brown, G., Pocock, A., Zhao, M. -J. & Lujan, M., 2012. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research,* 13(1), pp. 27-66.

Camastra, F. & Verri, A., 2005. A novel kernel method for clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 27(5), pp. 801-805.

Caruana, R. & Freitag, D., 1994. *Greedy attribute selection.* New Brunswick, USA, Proceedings of the 11th International Conference on Machine Learning, pp. 28-36.

Che, J. et al., 2017. Maximum relevance minimum common redundancy feature selection for nonlinear data. *Information Sciences,* Volume 409, pp. 68-86.

Chen, C. H., 2016. Unsupervised margin-based feature selection using linear transformations with neighbor preservation. *Neurocomputing,* Volume 171, pp. 1354-1366.

Chen, S., Billings, S. A. & Lo, W., 1989. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control,* 50(5), pp. 1873-1896.

Choi, S. W. et al., 2005. Fault detection and identification of nonlinear processes based on kernel PCA. *Chemometrics and Intelligent Laboratory Systems,* 75(1), pp. 55-67.

Darbellay, G. A. & Vajda, I., 1999. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory,* 45(4), pp. 1315-1321.

Dash, M. & Liu, H., 1997. Feature selection for classification. *Intelligent Data Analysis,* 1(1), pp. 131-156.

Dash, M. & Liu, H., 2003. Consistency-based search in feature selection. *Artificial Intelligence,* 151(1), pp. 155-176.

Das, S., Jyoti Choudhury, S., Das, A. K. & Sil, J., 2014. Selection of graph-based features for character recognition using similarity based feature dependency and rough set theory. In: G. P. Biswas & S. Mukhopadhyay, eds. *Recent Advances in Information Technology.* New Delhi: Springer New Delhi, pp. 57-64.

Daub, C. O., Steuer, R., Selbig, J. & Kloska, S., 2004. Estimating mutual information using B-spline functions- an improved similarity measure for analysing gene expression data. *BMC Bioinformatics,* 5(1), pp. 118-130.

Ding, C. & Peng, H., 2005. Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology,* 3(2), pp. 185-205.

Estevez, P., Tesmer, M., Perez, C. & Zurada, J. M., 2009. Normalized mutual information feature selection. *IEEE Transactions on Neural Network,* 20(2), pp. 189-201.

Fiedler, M., 1973. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal,* 23(2), pp. 298-305.

Fiedler, M., 1989. Laplacian of graphs and algebraic connectivity. *Banach Center Publications,* 25(1), pp. 57-70.

Fisher, R. A., 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics,* 7(2), pp. 179-188.

Fraser, A. M. & Swinney, H. L., 1986. Independent coordinates for strange attractors from mutual information. *Physical Review A,* 33(2), pp. 1134-1140.

Fukunaga, K., 2013. *Introduction to statistical pattern recognition.* Indiana, USA: Elsevier Inc..

Gao, W., Oh, S. & Viswanath, P., 2017. *Demystifying fixed k-nearest neighbor information estimators.* Aachen, Germany, Proceedings of IEEE International Symposium on Information Theory, pp. 1267-1271.

Gao, Z., Zhang, G., Nie, F. & Zhang, H., 2017. Local Shrunk Discriminant Analysis (LSDA). *arXiv:1705.01206 (cs).*

Garcia, S., Luengo, J. & Herrera, F., 2016. Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based System,* Volume 98, pp. 1-29.

Gerbert, P. et al., 2015. *Industry 4.0: The future of productivity and growth in manufacturing industries.* [Online] Available at:
https://www.bcg.com/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries.aspx
[Accessed 26 February 2018].

Gilad-Bachrach, R., Navot, A. & Tishby, N., 2004. *Margin based feature selection- theory and algorithms.* Alberta, Canada, Proceedings of the 21st ACM International Conference on Machine Learning, pp. 43-50.

Glassdoor Inc., 2018. *25 Best Jobs in the UK.* [Online]
Available at: https://www.glassdoor.co.uk/List/Best-Jobs-in-UK-LST_KQ0,15.htm
[Accessed 15 April 2018].

Grimmett, G. & Welsh, D., 2014. *Probability: An Introduction.* 2nd ed. Oxford : Oxford University Press.

Gu, Q., Li, Z. & Han, J., 2012. *Generalized fisher score for feature selection.* Barcelona, Spain, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, pp. 266-273.

Guyon, I. & Elisseeff, A., 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research,* Volume 3, pp. 1157-1182.

Haeri, M. A. & Ebadzadeh, M. M., 2014. Estimation of mutual information by the fuzzy histogram. *Fuzzy Optimization and Decision Making,* 13(3), pp. 287-318.

Hall, M. A., 1999. *Correlation-based feature selection for machine learning,* Waikato, New Zealand: The University of Waikato.

Hancer, E. et al., 2015. *A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information.* Sendai, Japan, Proceedings of IEEE Congress on Evolutionary Computation, pp. 2420-2427.

Heberger, K. & Andrade, J. M., 2004. Procrustes rotation and pair-wise correlation: A parametric and a non-parametric method for variable selection. *Croatica Chemica Acta,* 77(1-2), pp. 117-125.

He, D., Zhang, H., Hao, W. & Zhang, R., 2015. A robust parzen window mutual information estimator for feature selection with label noise. *Intelligent Data Analysis,* 19(6), pp. 1199-1212.

He, X., Cai, D. & Niyogi, P., 2006. Laplacian score in feature selection. *Advances in Neural Information Processing Systems,* pp. 507-514.

He, X. & Niyogi, P., 2004. Locality preserving projections. *Advances in Neural Information Processing Systems,* pp. 153-160.

Hira, Z. M. & Gilles, D. F., 2015. A review of feature selection and feature extraction methods applied on microarray data. *Advances in Bioinformatics,* pp. 1-13.

Holzinger, A. et al., 2014. On the generation of point cloud data sets: Step one in th knowledge discovery process. In: A. Holzinger & I. Jurisica, eds. *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics.* Berlin, Germany: Springer-Heidelberg, pp. 57-80.

Hoque, N., Bhattacharyya, D. K. & Kalita, J. K., 2014. MIFS-ND: a mutual information-based feature selection method. *Expert Systems with Applications,* 41(14), pp. 6371-6385.

Howell, D. C., 2007. The treatment of missing data. In: W. Outhwaite & S. Turner, eds. *The SAGE Handbook of Social Science Methodology.* California, USA: SAGE Publications, pp. 208-224.

Huang, T. M., Kecman, V. & Kopriva, I., 2006. *Kernel based algorithms for mining huge data sets: Supervised, semi-supervised, and unsupervised learning.* New York, USA: Springer-Verlag Inc.

Hu, W., Choi, K. -S., Gu, Y. & Wang, S., 2013. Minimum-maximum local structure information for feature selection. *Pattern Recognition Letters,* 34(5), pp. 527-535.

Izenman, A. J., 2013. Linear discriminant analysis. In: A. J. Izenman, ed. *Modern Multivariate Statistical Techniques.* New York, USA: Springer Science & Business Media LLC, pp. 237-280.

Jaakkola, T. S. & Haussler, D., 1999. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems,* pp. 487-493.

Jaffel, I., Taouali, O., Harkat, M. F. & Messaoud, H., 2017. Kernel principal component analysis with reduced complexity for nonlinear dynamic process monitoring. *International Journal of Advanced Manufacturing Technology,* 88(9-12), pp. 3265-3279.

Jain, I., Jain, V. K. & Jain, R., 2018. Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. *Applied Soft Computing,* Volume 62, pp. 203-215.

Jain, N. & Murthy, C. A., 2016. A new estimate of mutual informatio based measure of dependence between two variables: properties and fast implementation. *International Journal of Machine Learning and Cybernetics,* 7(5), pp. 857-875.

Janecek, A., Gansterer, W., Demel, M. & Ecker, G., 2008. On the relationship between feature selection and classification accuracy. In: *New Challenges for Feature Selection in Data Mining and Knowledge Discovery.* Antwerp, Belgium: PMLR, pp. 90-105.

Jeffers, J., 1967. Two case studies in the application of principal component analysis. *Applied Statistics,* pp. 225-236.

Jiang, S. -Y. & Wang, L. -X., 2016. Efficient feature selection based on correlation measure between continuous and discrete features. *Information Processing Letters,* 116(2), pp. 203-215.

John, G. H., Kohavi, R. & Pfleger, K., 1994. *Irrelevant features and the subset selection problem.* New Brunswick, USA, Proceedings of the 11th International Conference on Machine Learning, pp. 121-129.

Kallas, M. et al., 2013. Non-negativity constraints on the pre-image for pattern recognition with kernel machines. *Pattern Recognition,* 46(11), pp. 3066-3080.

Kang, Z., Peng, C. & Cheng, Q., 2017. *Twin learning for similarity and clustering: a unified kernel approach.* San Francisco, USA, Proceedings of the 31st AAAI Conference on Artificial Intelligence, pp. 2080-2086.

Khalid, S., Khalil, T. & Nasreen, S., 2014. *A survey of feature selection and feature extraction techniques in machine learning.* London, United Kingdom, Proceedings of IEEE Science and Information Conference, pp. 372-378.

Kira, K. & Rendell, L. A., 1992. *The feature selection problem: Traditional methods and a new algorithm.* San Jose, USA, Proceedings of the 10th AAAI National Conference on Artificial Intelligence, pp. 129-134.

Kohavi, R. & John, G. H., 1997. Wrappers for feature subset selection. *Artificial Intelligence,* 97(1-2), pp. 273-324.

Kohavi, R. & Sommerfield, D., 1995. *Feature subset selection using the wrapper method: Overfitting and dynamic search space topology.* Montreal, Canada, Proceedings of the First International Conference on Knowledge Discovery and Data Mining, pp. 192-197.

Koller, D. & Sahami, M., 1996. *Toward Optimal Feature Selection,* Stanford, USA: Stanford InfoLab.

Korenberg, M., Billings, S., Liu, Y. & Mcllroy, P., 1988. Orthogonal parameter estimation algorithm for non-linear stochastic systems. *International Journal of Control,* 48(1), pp. 193-210.

Kraskov, A., Stogbauer, H. & Grassberger, P., 2004. Estimating mutual information. *Physical Review E,* 29(6), pp. 1-16.

Krzanowski, W., 1987. Selection of variables to preserve multivariable data structure using principle components. *Applied Statistics,* pp. 22-33.

Kudo, M. & Sklansky, J., 2000. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition,* 33(1), pp. 25-41.

Kwak, N. & Choi, C. -H., 2002a. Input feature selection for classification problems. *IEEE Transactions on Neural Network,* 13(1), pp. 143-159.

Kwak, N. & Choi, C. -H., 2002b. Input feature selection by mutual information based on Parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 24(12), pp. 1667-1671.

Kwok, J. -Y. & Tsang, I. -H., 2004. The pre-image problem in kernel methods. *IEEE Transactions on Neural Networks,* 15(6), pp. 1517-1525.

Li, J., Tu, Q. & Yan, Z., 2016. *Refining pre-image via error compensation for KPCA-based pattern de-noising.* Cancun, Mexico, Proceedings of 23rd IEEE International Conference on Pattern Recognition, pp. 414-419.

Likitjarernkul, T. et al., 2017. PCA based feature extraction for classification of stator-winding faults in induction motors. *Pertanika Journal of Science & Technology,* Volume 25, pp. 197-204.

Lin, S. -W., Ying, K. -C., Chen, S. -C. & Lee, Z. -J., 2008. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications,* 35(4), pp. 1817-1824.

Liu, H. & Motoda, H., 2007. *Computational methods of feature selection.* Bota Racon, USA: Chapman and Hall/CRC.

Liu, H. & Motoda, H., 2012. *Feature selection for knowledge discovery and data mining.* New York, USA: Springer Science & Business Media LLC.

Liu, H. & Setiono, R., 1996a. *Feature selection and classification- A probability wrapper approach.* Fukuoka, Japan, Proceedings of the 9th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pp. 419-424.

Liu, H. & Setiono, R., 1996b. *A probabilistic approach to feature selection- A filter solution.* Bari, Italy, Proceedings of 13th International Conference on Machine Learning, pp. 319-327.

Liu, H. & Setiono, R., 1998. Incremental feature selection. *Applied Intelligence,* 9(3), pp. 217-230.

Liu, H. & Yu, L., 2005. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering,* 17(4), pp. 491-502.

Liu, Q., Lu, H. & Ma, S., 2004. Improving kernel fisher discriminant analysis for face recognition. *IEEE Transactions on Circuits and Systems for Video Technology,* 14(1), pp. 42-49.

Liu, X. et al., 2014. Global and local preservation for feature selection. *IEEE Transactions on Neural Networks and Learning Systems,* 25(6), pp. 1083-1095.

Li, W., 1990. Mutual information functions versus correlation functions. *Journal of Statistical Physics,* 60(5-6), pp. 823-837.

Lopes, F. M., Martins, D. C., Barrera, J. & Cesa, R. M., 2014. A feature selection technique for inference of graphs from their known topological properties: Revealing scale-free gene regulatory networks.. *Information Science,* Volume 272, pp. 1-15.

Luengo, J. et al., 2018. CNC-NOS: class noise cleaning by ensemble filtering and noise scoring. *Knowledge-Based Systems,* Volume 140, pp. 27-49.

Maas, C., 1987. Transportation in graphs and the admittance spectrum. *Discrete Applied Mathematics,* 16(1), pp. 31-49.

Maaten, L. V. D., Postma, E. & Herik, J. V. d., 2009. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research,* Volume 10, pp. 66-71.

Mafarja, M. M. & Mirjalili, S., 2017. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing,* Volume 260, pp. 302-312.

Mika, S. et al., 1999a. *Fisher discriminant analysis with kernels.* Madison, USA, Proceedings of the IEEE Signal Processing Society Workshop, pp. 41-48.

Mika, S. et al., 1999b. Kernel PCA and de-noising in feature spaces. *Advances in Neural Information Processing Systems,* Volume 2, pp. 536-542.

Mitra, P., Murthy, C. A. & Pal, S. K., 2002. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 24(3), pp. 301-312.

Moddemeijer, R., 1989. On estimation of entropy and mutual information of continuous distributions. *Signal Processing,* 16(3), pp. 233-248.

Mohar, B., Alavi, Y., Chartrand, G. & Oellermann, O., 1991. The Laplacian spectrum of graphs. *Graph Theory, Combinatorics, and Applications,* Volume 2, pp. 871-898.

Moon, Y. -I., Rajagopalan, B. & Lall, U., 1995. Estimation of mutual information using kernel density estimators. *Physical Review E,* 52(3), pp. 2318-2321.

Moradi, P. & Gholampour, M., 2016. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing,* Volume 43, pp. 117-130.

Narendra, P. M. & Fukunaga, K., 1977. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers,* 100(9), pp. 917-922.

Navi, M., Davoodi, M. R. & Meskin, N., 2015. Sensor fault detection and isolation of an industrial gas turbine using partial kernel PCA. *IFAC-Papers on Line,* 48(21), pp. 1389-1396.

Nettleton, D. F., Orriols-Puig, A. & Fornells, A., 2010. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review,* 33(4), pp. 275-306.

O'Donovan, P., Leahy, K., Bruton, K. & O'Sullivan, D. T. J., 2015. Big data in manufacturing: A systematic mapping study. *Journal of Big Data,* 2(20), pp. 1-22.

Parthalain, N., Shen, Q. & Jensen, R., 2010. A distance measure approach to exploring the rough set boundary region for attribute reduction. *IEEE Transactions on Knowledge and Data Engineering,* 22(3), pp. 306-317.

Pedrycz, W., 1986. Techniques of supervised and unsupervised pattern recognition with the aid of fuzzy set theory. In: L. N. Kanal & E. S. Gelsema, eds. *Pattern Recognition in Practice.* Amsterdam, Holland: Elsevier, pp. 439-448.

Peng, H., Long, F. & Ding, C., 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 27(8), pp. 1226-1238.

Peng, Y., Wu, Z. & Jiang, J., 2010. A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics,* 43(1), pp. 15-23.

Perazzi, F., Sorkine-Hornung, O. & Sorkine-Hornung, A., 2015. *Efficient salient foreground detection for images and video using fiedler vectors.* Zurich, Switzerland, Proceedings of Eurographics; Computer Graphic Forum, pp. 21-29.

Perrin, E. B., Durch, J. S. & Skillman, S. M., 1999. Data and information systems: Issues for performance measurement. In: E. B. Perrin, J. S. Durch & S. M. Skillman, eds. *Principle and Policies for Implementation an Information Network.* Washington, USA: National Academic Press, pp. 70-92.

Pradhananga, N., 2007. *Effective linear-time feature selection,* Waikato, New Zealand: Master of Science Thesis, University of Waikato.

Pudil, P., Novovicova, J. & Kittler, J., 1994. Floating search methods in feature selection. *Pattern Recognition Letters,* 15(11), pp. 1119-1125.

Quinlan, J. R., 1994. The minimum description length principle and categorical theories. *Machine Learning Proceedings*, pp. 233-241.

Ren, Y., Zhang, G., Yu, G. & Li, X., 2012. Local and global structure preserving base feature selection. *Neurocomputing,* Volume 89, pp. 147-157.

Reynders, E., Wursten, G. & De Roeck, G., 2014. Output-only structural health monitoring in changing environment conditions by means of nonlinear system identification. *Structural Health Monitoring,* 13(1), pp. 82-93.

Robnik-Sikonja, M. & Kononenko, I., 2003. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning,* 53(1-2), pp. 23-69.

Roweis, S. T. & Saul, L. K., 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science,* 290(5500), pp. 2323-2326.

Saeys, Y., Inza, I. & Larranaga, P., 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics,* 23(19), pp. 2507-2517.

Saez, J. A., Galar, M., Luengo, J. & Herrera, F., 2013. Tackling the problem of classification with noisy data using multiple classifier systems: analysis of the performance and robustness. *Information Sciences,* Volume 247, pp. 1-20.

Saez, J. A., Galar, M., Luengo, J. & Herrera, F., 2014. Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. *Knowledge and Information Systems,* 38(1), pp. 179-206.

Schmidt, J. F., Santelli, C. & Kozerke, S., 2016. MR image reconstruction using block matching and adaptive kernel methods. *PLOS One,* 11(4), pp. 1-10.

Scholkopf, B. & Smola, A. M. K. -R., 1997. *Kernel principal component analysis.* Lausanne, Switzerland, Proceedings of 7th International Conference on Artificial Neural Networks, pp. 583-588.

Shanab, A. A., Khoshgoftaar, T. M. & Wald, R., 2014. *Evaluation of wrapper-based feature selection using hard, moderate, and easy bioinformatics data.* Boca Raton, USA, Proceedings of IEEE International Conference on Bioinformatics and Bioengineering, pp. 149-155.

Shanab, A. A., Khoshgoftaar, T. M., Wald, R. & Napolitano, A., 2012. *Impact of noise and data sampling on stability of feature ranking techniques for biological datasets.* Las Vegas, USA, Proceedings of IEEE 13th International Conference on Information Reuse and Integration, pp. 415-422.

Shang, R., Chang, J., Jiao, L. & Xue, Y., 2017. Unsupervised feature selection based on self-representation sparse regression and local similarity preserving. *International Journal of Machine Learning Cybernetics,* 9(44), pp. 1-14.

Shinde, A., Sahu, A., Apley, D. & Runger, G., 2014. Preimages for variation patterns from kernel PCA and bagging. *IIE Transactions,* 46(5), pp. 429-456.

Shin, K. & Miyazaki, S., 2016. A fast and accurate feature selection algorithm based on binary consistency measure. *Computational Intelligence,* 32(4), pp. 646-667.

Shu, X., Gao, Y. & Lu, H., 2012. Efficient linear discriminant analysis with locality preserving for face recognition. *Pattern Recognition,* 45(5), pp. 1892-1898.

Siedlecki, W. & Sklansky, J., 1989. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters,* 10(5), pp. 335-347.

Sivarajah, U., Kamal, M. K., Irani, Z. & Weerakkody, V., 2017. Critical analysis of big data challenges and analytical methods. *Journal of Business Research,* Volume 70, pp. 263-286.

Skalak, D. B., 1994. *Prototype and feature selection by sampling and random mutation hill climbing algorithms.* New Brunswick, USA, Proceedings of the 11th International Conference on Machine Learning, pp. 293-301.

Solares, J. R. A. & Wei, H. L., 2015. Nonlinear model structure detection and parameter estimation using a novel bagging method based on distance correlation metric. *Nonlinear Dynamics,* 82(1-2), pp. 201-215.

Somol, P., Novovicova, J. & Pudil, P., 2010. Efficient feature subset selection and subset size optimization. *Pattern Recognition Recent Advances,* pp. 75-98.

Somol, P., Pudil, P., Novovicova, J. & Paclik, P., 1999. Adaptive floating search methods in feature selection. *Pattern Recognition Letters,* 20(11), pp. 1157-1163.

Sotoca, J. M. & Pla, F., 2010. Supervised feature selection by clustering using conditional mutual information-based distances. *Pattern Recognition,* 43(6), pp. 2068-2081.

Sun, Y., Todorovic, S. & Goodison, S., 2010. Local-learning-based feature selection for high-dimensional data analysis. *IEEE Transactions on Pattern Analysis and Machine,* 32(9), pp. 1610-1626.

Tabakhi, S. & Moradi, P., 2015. Relevancy-redundancy feature selection based on ant colony optimization. *Pattern Recognition,* 48(9), pp. 2798-2811.

Tang, J., Alelyani, S. & Liu, H., 2014. Feature selection for classification: A review. *Data Classification: Algorithms and Applications,* pp. 37-70.

Tate, R. F., 1954. Correlation between a discrete and a continuous variable. Point-biserial correlation. *The Annals of Mathematical Statistics,* pp. 603-607.

Tenenbaum, J. B., De Silva, V. & Langford, J. C., 2000. A global geometric framework for nonlinear dimensionality reduction. *Science,* 290(5500), pp. 2319-2323.

Teng, C. -M., 1999. *Correcting noisy data.* Bled, Slovenia, Proceedings of the 16th International Conference on Machine Learning, pp. 239-248.

Tong, C. & Yan, X., 2014. Statistical process monitoring based on a multi-manifold projection algorithm. *Chemometrics and Intelligent Laboratory Systems,* Volume 130, pp. 20-28.

Tzortzis, G. & Likas, A., 2012. *Kernel-based weighted multi-view clustering.* Washington, USA, Proceedings of the IEEE 12th International Conference on Data Mining, pp. 675-684.

Unler, A. & Murat, A., 2010. A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research,* 206(3), pp. 528-539.

Van Hulse, J. & Khoshgoftaar, T., 2009. Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering,* 68(12), pp. 1513-1542.

Vergara, J. R. & Estevez, P. A., 2014. A review of feature selection methods based on mutual information. *Neural Computing and Applications,* pp. 175-186.

Wang, H. & Mieghem, P. V., 2008. *Algebraic connectivity optimization via link addition.* Hyogo, Japan, Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Systems, pp. 22-30.

Wang, P., Jin, C. & Jin, S. W., 2012. *Software defect prediction scheme based on feature selection.* Shanghai, China, Proceedings of the 2012 IEEE International Symposium on Information Science and Engineering, pp. 477-480.

Wang, Y., Tan, B., Wang, Y. & Wu, J., 1994. Information structure analysis for quantitative assessment of mineral resources and the discovery of a silver deposit. *Nonrenewable Resources,* 3(4), pp. 284-294.

Wei, H. -L. & Billings, S., 2007. Feature subset selection and ranking for data dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 29(1), pp. 162-166.

Weston, J., Schölkopf, B. & Bakir, G. H., 2004. Learning to find pre-images. *Advances in Neural Information Processing Systems,* Volume 16, pp. 449-456.

Whitley, D. C., Ford, M. G. & Livingstone, D. J., 2000. Unsupervised forward selection: A method for eliminating redundant variables. *Journal of Chemical Information and Computer Sciences,* 40(5), pp. 1160-1168.

Wickramasinghe, R. I. P., 2017. Attribute noise, classification technique and classification accuracy. In: *Data Analytics and Decision Support for Cybersecurity.* Cham, Switzerland: Springer International Publishing, pp. 201-220.

Williams, J. W. & Li, Y., 2009. *Estimation of mutual information: A survey.* Gold Coast, Australia, Proceedings of the 4th International Conference on Rough Sets and Knowledge Technology, pp. 389-396.

Witten, I. H. & Frank, E., 2005. *Data mining: Practical machine learning tools and techniques.* 2nd ed. San Francisco, USA: Morgan Kaufmann.

Wold, S., Esbensen, K. & Geladi, P., 1987. Principle component analysis. *Chemometrics and Intelligent Laboratory Systems,* 2(1), pp. 37-52.

Wu, X. et al., 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems,* 14(1), pp. 1-37.

Xanthopoulos, P., Pardalos, P. M. & Trafalis, T. B., 2013. Linear discriminant analysis. In: P. Xanthopoulos, Panos M. Pardalos & Theodore B. Trafalis, eds. *Robust Data Mining.* New York, USA: Springer Science & Business Media LLC, pp. 27-33.

Xu, L., Yan, P. & Chang, T., 1988. *Best first strategy for feature selection.* Rome, Italy, Proceedings of 9th International Conference on Pattern Recognition, pp. 706-708.

Yang, C. et al., 2017. Big data and cloud computing innovation opportunities and challenges. *International Journal of Digital Earth,* 10(1), pp. 13-53.

Yang, J. & Honavar, V., 1998. Feature subset selection using a genetic algorithm. In: *Feature Extraction, Construction and Selection.* Boston, USA: Springer, pp. 117-136.

Yang, J. M., Yu, P. T. & Kuo, B. V., 2010. A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data. *IEEE Transactions on Geoscience and Remote Sensing,* 48(3), pp. 1279-1293.

Yan, H. & Yang, J., 2015. Locality preserving score for joint feature weights learning. *Neural Networks,* Volume 69, pp. 126-134.

Yan, S. et al., 2008. *Regression from patch-kernel.* Anchorage, USA, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8.

Yao, C. et al., 2017. LLE score: a new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition. *IEEE Transactions on Image Processing,* 26(11), pp. 5257-5269.

Yin, X., Chen, S., Hu, E. & Zhang, D., 2010. Semi-supervised clustering with metric learning: an adaptive kernel method. *Pattern Recognition,* 43(4), pp. 1320-1333.

Yokozawa, T., Takahashi, D., Boku, T. & Sato, M., 2006. *Efficient parallel implementation of classical gram-schmidt orthogonalization using matrix multiplication.* Rennes, France, Proceedings of 4th International Workshop on Parallel Matrix Algorithms and Applications, pp. 37-38.

Yu, D., An, S. & Hu, Q., 2011. Fuzzy mutual information based min-redundancy and max-relevance heterogeneous feature selection. *International Journal of Computational Intelligence Systems,* 4(4), pp. 619-633.

Yu, J., 2012. Local and global principal component analysis for process monitoring. *Journal of Process Control,* 22(7), pp. 1358-1373.

Yu, L. & Liu, H., 2004. Efficient feature selection via analysis of relevance and redundacy. *The Journal of Machine Learning Research,* Volume 5, pp. 1205-1224.

Yu, L., Ye, J. & Liu, H., 2007. *Dimensionality reduction for data mining- techniques, applications and trends.* Maryland, USA, Proceedings of 2006 SIAM International Conference of Data Mining, pp. 10-18.

Zhang, L., Meng, X., Wu, W. & Zhou, H., 2009. *Network fault feature selection based on adaptive immune clonal selection algorithm.* Hainan, China, Proceedings of the IEEE International Joint Conference on Computational Sciences and Optimization, pp. 969-973.

Zhang, L., Wang, X. & Qu, L., 2008. Feature reduction based on analysis of covariance matrix. *Computer Science and Computational Technology,* Volume 1, pp. 59-62.

Zhang, M., Ge, Z., Song, Z. & Fu, R., 2011. Global-local structure analysis model and its application for fault detection and identification. *Industry & Engineering Chemistry Research,* 50(11), pp. 6837-6848.

Zhang, Y., An, J. & Zhang, H., 2013. Monitoring of time-varying processes using kernel independent component analysis. *Chemical Engineering Science,* Volume 88, pp. 23-32.

Zhao, J., Lu, K. & He, X., 2008. Locality sensitive semi-supervised feature selection. *Neurocomputing,* 71(10-12), pp. 1842-1849.

Zhao, Z., 2017. *Classification in the presence of heavy label noise: a Markov chain sampling framework,* Burnaby, Canada: Master of Science Thesis, Simon Fraser University.

Zheng, W. -S., Lai, J. & Yuen, P. C., 2010. Penalized preimage learning in kernel principal component analysis. *IEEE Transactions on Neural Network,* 21(4), pp. 551-570.

Zheng, W., Lin, Z. & Wang, H., 2014. L1-norm kernel discriminant analysis via Bayes error bound optimization for robust feature extraction. *IEEE Transactions on Neural Networks and Learning Systems,* 25(4), pp. 793-805.

Zhu, X. & Wu, X., 2004. Class noise vs. attribute noise: a quantitative study. *Artificial Intelligence Review,* 22(3), pp. 177-210.

Zhu, X., Wu, X. & Yang, Y., 2004. *Error detection and impact-sensitive instance ranking in noisy datasets.* San Jose, USA, Proceedings of the 19st AAAI Conference on Artificial Intelligence, pp. 378-384.