DSV Report Series No. 24-003

Stockholm University

Nearest Neighbor Classification in High Dimensions

Sampath Deegalla

Doctoral Thesis in Computer and Systems Sciences at Stockholm University, Sweden 2024

Nearest Neighbor Classification in High Dimensions

Sampath Deegalla

Academic dissertation for the Degree of Doctor of Philosophy in Computer and Systems Sciences at Stockholm University to be publicly defended on Tuesday 5 March 2024 at 13.00 in lilla hörsalen, NOD-huset, Borgarfjordsgatan 12.

Abstract

The simple k nearest neighbor (kNN) method can be used to learn from high dimensional data such as images and microarrays without any modification to the original version of the algorithm. However, studies show that kNN's accuracy is often poor in high dimensions due to the curse of dimensionality; a large number of instances are required to maintain a given level of accuracy in high dimensions. Furthermore, distance measurements such as the Euclidean distance may be meaningless in high dimensions. As a result, dimensionality reduction could be used to assist nearest neighbor classifiers in overcoming the curse of dimensionality. Although there are success stories of employing dimensionality reduction methods, the choice of which methods to use remains an open problem. This includes understanding how they should be used to improve the effectiveness of the nearest neighbor algorithm.

The thesis examines the research question of how to learn effectively with the nearest neighbor method in high dimensions. The research question was broken into three smaller questions. These were addressed by developing effective and efficient nearest neighbor algorithms that leveraged dimensionality reduction. The algorithm design was based on feature reduction and classification algorithms constructed using the reduced features to improve the accuracy of the nearest neighbor algorithm. Finally, forming nearest neighbor ensembles was investigated using dimensionality reduction.

A series of empirical studies were conducted to determine which dimensionality reduction techniques could be used to enhance the performance of the nearest neighbor algorithm in high dimensions. Based on the results of the initial studies, further empirical studies were conducted and they demonstrated that feature fusion and classifier fusion could be used to improve the accuracy further. Two feature and classifier fusion techniques were proposed, and the circumstances in which these techniques should be applied were examined. Furthermore, the choice of the dimensionality reduction method for feature and classifier fusion was investigated. The results indicate that feature fusion is sensitive to the selection of the dimensionality reduction method. Finally, the use of dimensionality reduction in nearest neighbor ensembles was investigated. The results demonstrate that data complexity measures such as the attribute-to-instance ratio and Fisher's discriminant ratio can be used to select the nearest neighbor ensemble depending on the data type.

Keywords: *Nearest Neighbor, High-Dimensional Data, Curse of Dimensionality, Dimensionality Reduction.*

Stockholm 2024
http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-225627

ISBN 978-91-8014-645-6
ISBN 978-91-8014-646-3
ISSN 1101-8526

Department of Computer and Systems Sciences

Stockholm University, 164 07 Kista

NEAREST NEIGHBOR CLASSIFICATION IN HIGH DIMENSIONS

Sampath Deegalla

Stockholm
University

Nearest Neighbor Classification in High Dimensions

Sampath Deegalla

ISBN print 978-91-8014-645-6
ISBN PDF 978-91-8014-646-3
ISSN 1101-8526

Printed in Sweden by Universitetsservice US-AB, Stockholm 2024

To my loving wife, Susinika

Abstract

The simple k nearest neighbor (kNN) method can be used to learn from high-dimensional data such as images and microarrays without any modification to the original version of the algorithm. However, studies show that kNN's accuracy is often poor in high dimensions due to the curse of dimensionality; a large number of instances are required to maintain a given level of accuracy in high dimensions. Furthermore, distance measurements such as the Euclidean distance may be meaningless in high dimensions. As a result, dimensionality reduction could be used to assist nearest neighbor classifiers in overcoming the curse of dimensionality. Although there are success stories of employing dimensionality reduction methods, the choice of which methods to use remains an open problem. This includes understanding how they should be used to improve the effectiveness of the nearest neighbor algorithm.

The thesis examines the research question of how to learn effectively with the nearest neighbor method in high dimensions. The research question was broken into three smaller questions. These were addressed by developing effective and efficient nearest neighbor algorithms that leveraged dimensionality reduction. The algorithm design was based on feature reduction and classification algorithms constructed using the reduced features to improve the accuracy of the nearest neighbor algorithm. Finally, forming nearest neighbor ensembles was investigated using dimensionality reduction.

A series of empirical studies were conducted to determine which dimensionality reduction techniques could be used to enhance the performance of the nearest neighbor algorithm in high dimensions. Based on the results of the initial studies, further empirical studies were conducted and they demonstrated that feature fusion and classifier fusion could be used to improve the accuracy further. Two feature and classifier fusion techniques were proposed, and the circumstances in which these techniques should be applied were examined. Furthermore, the choice of the dimensionality

reduction method for feature and classifier fusion was investigated. The results indicate that feature fusion is sensitive to the selection of the dimensionality reduction method. Finally, the use of dimensionality reduction in nearest neighbor ensembles was investigated. The results demonstrate that data complexity measures such as the attribute-to-instance ratio and Fisher's discriminant ratio can be used to select the nearest neighbor ensemble depending on the data type.

Sammanfattning

Algoritmen för de k närmaste grannarna (kNN) kan användas för att klassificera högdimensionella data såsom bilder och microarrayer utan någon modifiering av den ursprungliga versionen av algoritmen. Studier visar dock att precisionen ofta är låg för högdimensionella data, vilket kräver ett stort antal instanser för att bibehålla en önskad korrekthet. Vidare kan avståndsmått som det euklidiska vara mindre framgångrsika i höga dimensioner. Dimensionalitetsreduktion skulle kunna användas för att förbättra algoritmen. Även om det finns några exempel där detta framgångsrikt har använts är det fortfarande en öppen frågeställning vilka metoder för dimensionsreducering som ska användas och när detta ska göras för att förbättra prestandan hos algoritmen.

Avhandlingen undersöker denna forskningsfråga genom att utveckla och utvärdera varianter av algoritmen som utnyttjar dimensionalitetsreduktion. Algoritmdesignen baserades på olika funktioner för att reducera dimensionaliteten och olika ansatser för att kombinera klassificerare som använder dessa, inklusive ensembler av sådana klassificerare.

En serie empiriska studier genomfördes för att bestämma vilka dimensionsreduktionstekniker som skulle kunna användas för att förbättra prestandan hos algoritmen för högdimensionella data. Baserat på resultaten från de inledande studierna genomfördes ytterligare empiriska studier, som visade att fusion av dimensionalitetsreduktion och klassificerare resulterade i att den prediktiva prestandan kunde förbättras ytterligare. Ett antal fusionsansatser föreslogs för kombination av klassificerare och attribut, och det undersöktes när dessa tekniker kunde rekommenderas. Resultaten indikerar att fusion av attribut är känslig för valet av dimensionsreduktion. Slutligen undersöktes användningen av dimensionsreduktion för ensembler av klassificerare. Resultaten visar att datakomplexitetsmått såsom förhållandet mellan antalet attribut och antalet instanser samt Fishers diskriminantkvot kan användas för att välja ensemble beroende på datatyp.

Acknowledgements

First and foremost, I would like to thank Prof. Henrik Boström for being my main supervisor and guiding me to reach this milestone. Without his continued guidance and patience, I would not have been able to complete the dissertation.

Then, I greatly appreciate the guidance of my local supervisor Prof. Keerthi Walgama. I value his contributions to the success of this work. I further acknowledge the assistance given to me by Prof. Panagiotis Papapetrou during the latter stages of my research. I would like to express my gratitude to Prof. Magnus Boman and Prof. Hercules Dalianis for constructive feedback during the predoc seminar.

My sincere thanks go to the Department of Computer and Systems Sciences (DSV) at Stockholm University, Sweden, for their support in facilitating the completion of this dissertation. They provided extensive support on numerous occasions, aiding my pursuit of higher education. Similarly, I would also like to thank the Department of Computer Engineering, University of Peradeniya, Sri Lanka, for facilitating me to continue my studies. I would like to extend special thanks to Dr. Devapriya, whose consistent encouragement has been a driving force behind this work.

I am eternally grateful to my late father, Dhanapala, and my late mother, Premalatha, for their unwavering love, support and guidance. Finally, I extend my heartfelt gratitude to my loving wife Susinika, our son Senhiru, and our daughter Sehansa, whose support and understanding of my work have motivated me greatly.

List of Included Papers

S. Deegalla and H. Boström. Reducing High-dimensional Data by Principal Component Analysis vs. Random Projection for Nearest Neighbor Classification. In *Proceedings of the 5th International Conference on Machine Learning and Applications*, Orlando, FL, USA, pages 245–250, IEEE Computer Society, 2006.

S. Deegalla and H. Boström. Classification of Microarrays with kNN: Comparison of Dimensionality Reduction Methods. In *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning*, Birmingham, UK, pages 800–809, Springer, 2007.

S. Deegalla and H. Boström. Fusion of Dimensionality Reduction Methods: A Case Study in Microarray Classification. In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, USA, pages 460–465, 2009.

S. Deegalla and H. Boström. Improving Fusion of Dimensionality Reduction Methods for Nearest Neighbor Classification. In *Proceedings of the 8th International Conference on Machine Learning and Applications*, Miami, FL, USA, pages 771–775, 2009.

S. Deegalla, H. Boström and K. Walgama. Choice of Dimensionality Reduction Methods for Feature and Classifier Fusion with Nearest Neighbor Classifiers. In *Proceedings of the 15th International Conference on Information Fusion*, pages 875–881, IEEE Computer Society, 2012.

S. Deegalla, K. Walgama, P. Papapetrou and H. Boström. Random Subspace and Random Projection Nearest Neighbor Ensembles for High Dimensional Data, *Expert Systems with Applications*, 191:116078, 2022.

Contents

List of Figures

Abbreviations

CF Classifier Fusion

FF Feature Fusion

ID3 Iterative Dichotomiser 3

IG Information Gain

kNN k Nearest Neighbor

MDL Minimum Description Length

PCA Principal Component Analysis

PLS Partial Least Squares

RP Random Projection

SVD Singular Value Decomposition

SVM Support Vector Machine

WEKA Waikato Environment for Knowledge Analysis

Chapter 1

Introduction

Machine learning is a sub-field of artificial intelligence that is related to both computer science and statistics [1]. It primarily focuses on the development of algorithms that improve based on experience. These algorithms are known as *learning algorithms* [2, 3]. For instance, a learning algorithm can be used to identify patients with a high probability of having a particular cancer (knowledge) based on the past medical records of similar patients (experience, data). Furthermore, it has been demonstrated that systems constructed using machine learning approaches outperform human experts in various commercially successful applications [4, 3]. Nevertheless, recent advances in science and engineering have complicated the learning process by generating complex data. For instance, massive datasets could be problematic for the present learning algorithms in terms of efficiency and effectiveness, where efficiency refers to the computational cost of building a learning algorithm. In contrast, effectiveness refers to the accuracy of the output produced by the learning algorithm. These complex datasets require further research in machine learning on new representations, algorithms, frameworks and techniques for effective and efficient learning.

Machine learning concerns learning from data. We refer to a row of a dataset as an *instance* (example, object) whereas a column is referred to as a *feature* (attribute, variable). For example, an instance could be a patient's medical record, whereas a feature could be the patient's age. All instances are assumed to have the same number of features. In some learning situations, instances are labelled using categoric values (*class* labels) or numeric values. A *model* is built using these labelled data in supervised learning. The output, or model, can later be used to predict the label of a new instance. Learning from *high-dimensional data*, where the data contains a large number of features, is an important yet challenging problem for current learning algorithms [4]. A popular example of high dimensional data is *microarrays* which utilises DNA microarray technology [5] to generate data. Microarrays enable the simultaneous monitoring of thousands of gene expression profiles [6] and can be used to quantify changes in gene expressions. Microarray analysis [5] has been used in a variety of applications, including classifying cancer patients into relevant disease classes [7, 8], identifying novel subtypes of cancers [9] and estimating the effectiveness of medical treatments [10, 11, 12]. Additionally, it has been used to reduce the waste of health care resources and unnecessary treatments compared to conventional clinical methods [11].

There are two major factors for the problem of building models in high dimensional domains, namely irrelevant features and redundant features. Irrelevant features are not important for discriminating between the classes, e.g., for separating cancer patients from others. Redundant features are features that may be relevant but are highly correlated with each other. Both irrelevant and redundant features often result in the poor performance of the learning algorithm, i.e., low accuracy when it classifies new instances. High dimensional datasets like microarrays often contain many features that are either irrelevant or redundant. Therefore, the use of all available features in the input data may often lead to poor performance. Moreover, how to handle these types of data appropriately is still an open question [4, 13]. One feasible approach to solving the problem of high dimensionality is to reduce the number of dimensions prior to the learning phase.

The nearest neighbor algorithm [14] is one of the machine learning algorithms that can be used to learn from high-dimensional data. It is a popular choice in academia and industry according to a poll by KDnuggets [15]. It has been used in many comparative studies (e.g. in [16]) with recently developed learning algorithms such as support vector machines (SVMs) [17] and random forests [18]. However, as with many other algorithms, its performance is poor in high dimensions. This poor performance is mainly due to the sensitivity to the input data due to irrelevant and redundant features, and improving the performance of the algorithm in high dimensions remains a challenge.

In this thesis, we investigate possible improvements of the nearest neighbor algorithm using dimensionality reduction [19], i.e., reducing the high number of dimensions to a much smaller number of dimensions, and also further enhancements using information fusion [20], i.e. combining the outputs of the dimensionality reduction methods.

1.1 Problem

The k nearest neighbor (kNN) classifier is one of the oldest, simplest and yet most powerful learning algorithms among the large number of machine learning algorithms used in practice. It does not assume any particular distribution for the input data and the only parameter to specify is k, the number of nearest neighbors to take into account in the final decision. Furthermore, it can be used to solve multiclass problems [14]. It does, however, assume that all features are numeric, which means that nominal features must be converted into numeric ones [21]. kNN may be one of the learning algorithms of choice for learning from high-dimensional data such as microarrays, where the number of features (expression levels of genes) is greater than the number of instances (patients or conditions/cell types/tissues) [10, 22, 23, 24]. However, the classification accuracy of kNN

on microarrays is generally inferior to that of recently developed learning algorithms such as SVMs, primarily due to the sensitivity [25] of the kNN algorithm to the input features. It has been demonstrated that kNN can perform nearly as well as or better than SVMs and aggregated trees when the feature set for kNN is carefully reduced [26, 27]. However, finding an appropriate method for reducing the number of features for kNN in high dimensions may not be straightforward. Langley and Sage have further shown that the number of instances needed to maintain a given classification accuracy of kNN exponentially increases with the number of irrelevant attributes [28]. In many cases, it may be impossible to improve the classification accuracy of kNN by adding more instances due to data scarcity and high production costs.

Recently, the focus on improving the performance of kNN has shifted to combining multiple classifiers, which is known as classifier fusion or ensemble learning [25]. However, it has been shown that traditional ensemble learning methods such as bagging [29] and boosting [30] are ineffective for the nearest neighbor classifier [25]. This is due to the fact that small changes in the instance space usually do not result in significant changes in the predictions made by the kNN classifier [27]. Several alternative methods have been suggested to address this problem. One suggestion would be to alter the feature space. For this approach, Bay [25] proposed using multiple random feature subsets with classifier fusion for kNN which introduces diversity into the subsets.

Furthermore, some researchers have applied several dimensionality reduction methods in sequence to reduce the limitations of using a single method. For example, a filter method is used prior to a wrapper method to reduce the computational complexity of the wrapper method in [23]. Similarly, feature selection is used prior to feature extraction to reduce the computational complexity of feature extraction [31].

The performance of the nearest neighbor method is often poor in high dimensions due to irrelevant and redundant features [32, 14, 25]. This inferior performance is due to the curse of dimensionality [19] which refers to the fact that more instances are required to reach a given level of accuracy [3]. Bagging and boosting techniques, which are used to improve the performance of classifiers by manipulating the instance space also failed to improve the nearest neighbor method [25]. One strategy for dealing with the high dimensionality of a dataset is to use dimensionality reduction to reduce the number of features, which improves the performance of the nearest neighbor classifier possibly by removing irrelevant and redundant attributes. However, which dimensionality reduction methods or combinations of dimensionality reduction methods should be chosen to improve the accuracy of the nearest neighbor classifier is an open issue.

This PhD thesis aims to conduct a comprehensive evaluation of the performance of the nearest neighbor classifier using a diverse range of dimensionality reduction methods. Furthermore, it investigates and proposes novel approaches for combining the outputs of dimensionality reduction methods to enhance nearest neighbor's accuracy for high-dimensional data.

1.2 Research Question

The objective of the study is to use dimensionality reduction to transform high-dimensional data into representative low dimensional data so that the nearest neighbor classification is effective in terms of the classification accuracy.

The main research question (Q) of the thesis is the following:

How can one effectively learn in high-dimensional space using the nearest neighbor classifier?

We addressed the main research question by breaking it into three less complex problems (Q1, Q2 and Q3). By answering the following questions,

we can better understand the underlying relationships between dimensionality reduction techniques and the kNN classifier performance, which leads to more effective solutions.

- Q1: To what extent does dimensionality reduction improve the nearest neighbor performance?
- Q2: How can the dimension reduction outputs be used to improve the nearest neighbor accuracy further?
- Q3: How can the nearest neighbor ensemble be used to improve the classification accuracy in high dimensions?

Based on the main research question, the research statement is:

Dimensionality reduction can be successfully used to reduce the original dimensions to much lower dimensions while maintaining the accuracy or even enhancing the accuracy. The outputs of dimensionality reduction can further improve the effectiveness of the nearest neighbor algorithm in high dimensions.

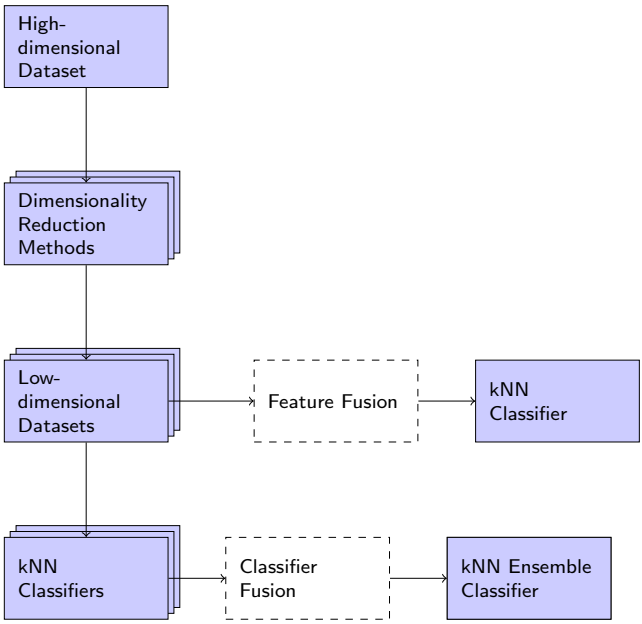

Figure 1.1: Overview of the methodology.

1.3 Methodology

This thesis investigates improving the effectiveness of the simple nearest neighbor classifier for high-dimensional datasets. To address the challenges posed by such datasets, we utilise dimensionality reduction. The research strategy employed in the thesis is quantitative and exploratory. Figure 1.1 represents an overview of the methodology. Our approach for this investigation is to perform a series of empirical studies to evaluate the nearest neighbor classifier with established dimensionality reduction methods.

These empirical studies run a series of experiments that involve applying the nearest neighbor classifier to a set of benchmark datasets. The datasets have been pre-processed using different dimensionality reduction methods. Note that the datasets do not necessarily concern realistic tasks such as cancer classification.

The dimensionality reduction methods chosen for the experiments were based on the existing literature. These methods are chosen because they frequently applied or recently evolved methods for dimensionality reduction. Although we consider established methods, these methods have not been extensively compared to one another for the variety of datasets used at the time at which the experiments were conducted.

This research started with research question Q1: 'To what extent does dimensionality reduction improve the nearest neighbor performance?'. In order to investigate this, dimensionality reduction methods capable of handling high dimensions have been explored. Initially, two dimensionality reduction methods were selected for this purpose: principal component analysis (PCA) and random projection (RP).

PCA is a well-known technique, and RP was an emerging method at that time [33, 34, 35, 36]. Both methods are unsupervised dimensionality reduction methods that do not use class information for the transformation. PCA is often a common choice for reducing dimensionality, and RP

was investigated as a dimensionality reduction method for high-dimensional data sets, which led us to investigate and compare the effectiveness of these methods on various high-dimensional datasets. PCA was useful for capturing the global structure of the data. At the same time, RP was useful for preserving the pairwise distances between instances and capturing the local structure of the data. The motivation for selecting RP is that its local structure preservation is particularly beneficial in high-dimensional contexts.

Since we have considered only unsupervised methods, which do not use class information for the transformation, we extend our inquiry to supervised dimensionality reduction methods. These methods, which consider class information, could provide a more customised approach to dimensionality reduction, especially when class separation is the main objective.

Therefore, we considered two additional dimensionality reduction methods in the subsequent work. Partial least squares (PLS) appeared as a dimensionality reduction method [37, 38], and we compared it with information gain (IG), which is often used in decision tree learning.

In both studies, we provided quantitative assessments of the predictive performance to investigate how well the existing dimensionality reduction methods can handle high dimensions.

To further improve the results of nearest neighbor in high dimensions, we raised the question Q2, 'How can the dimension reduction outputs be used to improve the nearest neighbor accuracy further?' It aims to boost the performance of kNN classifiers by using the outputs of multiple dimensionality reduction strategies. To tackle the research question, we consider feature and classifier fusion. Feature subsets from each reduction method are combined into one feature set for kNN for feature fusion. Classifier fusion, on the other hand, is developed from each individually reduced dataset. Then, the outputs of these classifiers are combined using simple majority voting.

Initially, our investigation involved feature and classifier fusion using three different dimensionality reduction methods. However, RP was excluded due to its inherent randomness; it requires multiple iterations to yield consistent results. Then, our focus shifted to examining the impact of the selection of dimensionality reduction methods, incorporating ReliefF [39, 40] into the set to introduce variation.

Based on the outcome of the previous study, our focus is on classifier fusion since the feature fusion strategy was sensitive to the selection of dimensionality reduction methods. To find the answer to research question Q3 'How can the nearest neighbor ensemble be used to improve the classification accuracy in high dimensions?', we have included random subspace and projection methods for forming ensembles, strengthening our methodological framework.

The empirical evaluations quantify the performance of a particular learning algorithm on an independent test set. They are used to assess the relative performance of different methods. There is no consensus on which measure to employ when a limited number of instances are available, but ten-fold cross-validation is a standard choice [21, 41, 42]; it is also adopted in this study. When ten-fold cross-validation is performed, the dataset is divided into ten equal folds, where one fold is taken as a test set and the rest are used as a training set. This process is repeated for each fold, i.e., ten times, and the mean accuracy, i.e., the average of the ten folds, is considered as the evaluation measure. The classification accuracy, which is defined as the ratio between the correctly classified instances and the total number of instances, is used as a measure of the predictive performance.

To assess the statistical significance of observed differences in the accuracy, we employed the Friedman test, which was followed by a Nemenyi post-hoc test when significant differences in accuracy were identified. In addition, all datasets considered in the study are publicly available real-world datasets so that the results of the study may be confirmed in future

research.

We focused on enhancing the effectiveness of simple nearest neighbor classifiers in high dimensions in this thesis. As an alternative method for improving efficiency, nearest neighbor algorithms based on kd-trees or ball-trees may be used; they can efficiently store and query high-dimensional data. However, the results may be identical to those of simple nearest neighbor algorithms. Other alternative methods that can be used instead of the nearest neighbor method include decision trees, SVMs, and random forests. We were interested in how simple nearest neighbor algorithms in high dimensions could be improved. Furthermore, for very high-dimensional data with a large number of features and instances, the computational complexity of other classifiers may be greater than that of simple kNN.

Classifiers such as SVMs and random forest classifiers could also be used with high-dimensional datasets. However, the nearest neighbor classifier is a simple method that can be directly used with high-dimensional data sets. Deep learning techniques could be a better choice for image data. However, in the initial stage, these methods were unavailable and they may need more instances, which was a limiting factor for datasets such as microarrays.

1.4 Contributions

The goal of this study is to improve the effectiveness of nearest neighbor classification in higher dimensions. The contributions include two comparative studies on dimensionality reduction for kNN, three studies on fusion approaches based on the output of dimensionality reduction to further improve the accuracy of kNN and one study on the nearest neighbor ensemble. Each study has been published in a separate paper.

Figure 1.2 provides an overview of the methodology, highlighting areas where the papers that have been published. Table 1.1 shows the relationship

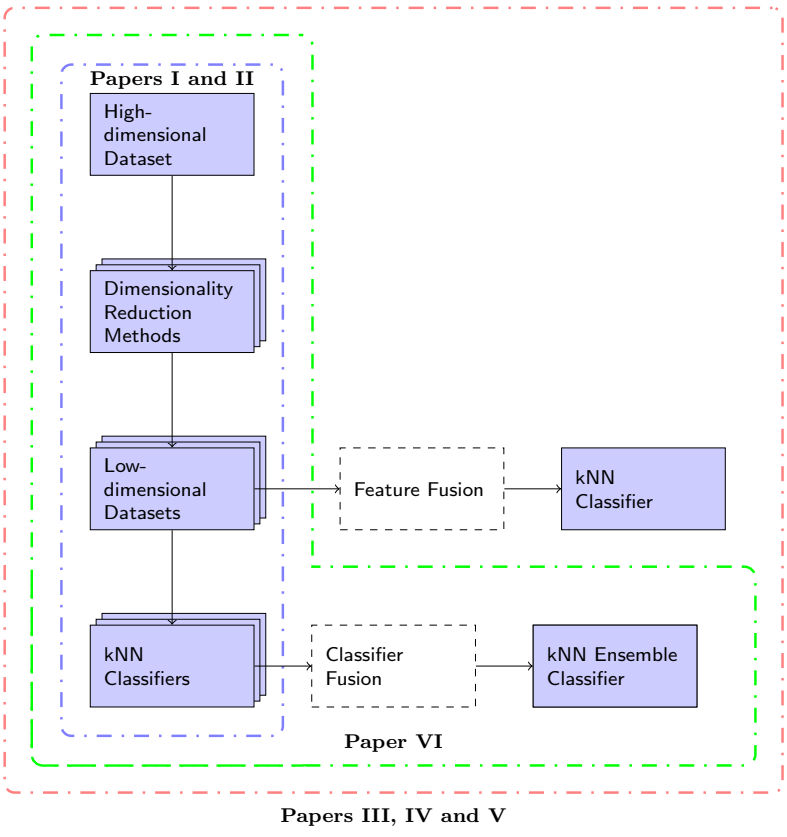

Figure 1.2: A visual representation of the methodology in terms of the published papers.

Table 1.1: The relationship between research questions and associated publications.

Publication	Q1	Q2	Q3
Paper I	✓	-	-
Paper II	✓	-	-
Paper III	-	✓	-
Paper IV	-	✓	-
Paper V	-	✓	-
Paper VI	-	-	✓

between our research questions (Q1, Q2 and Q3) and published papers (Papers I, II, III, IV, V and VI).

- **Paper I**: Sampath Deegalla and Henrik Boström. Reducing High-dimensional Data by Principal Component Analysis vs. Random Projection for Nearest Neighbor Classification. In *Proceedings of the 5th International Conference on Machine Learning and Applications*, Orlando, FL, USA, pages 245–250, IEEE Computer Society, 2006.

  Paper II: Sampath Deegalla and Henrik Boström. Classification of Microarrays with kNN: Comparison of Dimensionality Reduction Methods. In *Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning*, Birmingham, UK, pages 800–809, Springer, 2007.

  We address the following research question in these papers: 'To what extent does dimensionality reduction improve nearest neighbor performance?'

  In Paper I, we have experimentally investigated the effect of two dimensionality reduction methods, i.e., PCA and RP, when images and

microarrays are classified using kNN. In previous studies, Fradkin and Madigan [34] investigated both methods for three learning algorithms, including kNN. They argued that RP is best suited for kNN, as RP reaches the original or PCA accuracy quite rapidly compared to the other methods. They also showed that PCA is better than RP with respect to the predictive performance. However, their findings are based on binary classification problems, i.e., the classification problem is limited to two classes, and two highdimensional datasets only. In our experiment, we consider ten high-dimensional datasets, including images and microarrays, spanning binary and multiclass classification problems. Furthermore, we also consider the efficiency of kNN. In contrast to the findings in [34], our results show that the use of both dimensionality reduction methods outperforms the non-reduced feature set in a majority of the cases. This further confirms that PCA is better suited than RP even to multiclass classification problems when the computational complexity is not a problem. In conclusion, our study shows that PCA and RP are not only effective but also efficient for nearest neighbor classification in high dimensions.

Since the supervised dimensionality reduction methods use class information to select and transform the original features, they often give a better performance than unsupervised methods. Therefore, in Paper II, two additional supervised dimensionality reduction methods, PLS and IG, were considered for the empirical investigation.

Compared to the study reported in Paper I, four dimensionality reduction methods representing both unsupervised and supervised methods were considered for nearest neighbor learning in high dimensions. Furthermore, the study scope was narrowed to only consider microarray data. Although some studies [43, 44, 45] used these methods separately, none considered the four methods in the same experimental setting. Furthermore, PLS was introduced as a dimensionality reduction method for kNN. However, this method was not being investigated in combination with kNN at the time of the study.

The study shows that none of the four methods consistently outperforms the others when used with kNN. Furthermore, it is shown that PLS may be superior for binary classification problems, which confirms the results in [37]. The conclusion of the study is that the selection of the proper dimensionality reduction method may be of major importance for the predictive performance for learning in high dimensions with kNN.

- **Paper III**: Sampath Deegalla and Henrik Boström. Fusion of Dimensionality Reduction Methods: a Case Study in Microarray Classification. In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, USA, pages 460–465, 2009.

  Paper IV: Sampath Deegalla and Henrik Boström. Improving Fusion of Dimensionality Reduction Methods for Nearest Neighbor Classification. In *Proceedings of the 8th International Conference on Machine Learning and Applications*, Miami, FL, USA, 2009.

  Paper V: Sampath Deegalla, Henrik Boström and Keerthi Walgama. Choice of Dimensionality Reduction Methods for Feature and Classifier Fusion with Nearest Neighbor Classifiers. In *Proceedings of the 15th International Conference on Information Fusion*, pages 875–881, IEEE Computer Society, 2012.

  We address the following research question in these papers: 'How can the dimension reduction outputs be used to improve the nearest neighbor accuracy further?'

  Papers III, IV, and V address the problem of improving the performance of kNNs by utilising the outputs of multiple dimensionality reduction methods. Paper III presents a novel approach based on two fusion strategies, namely feature fusion and classifier fusion. In clas-

sifier fusion, a kNN classifier is obtained from each reduced dataset generated by each individual dimensionality reduction method and then the individual classifier outputs are combined using simple majority voting. In feature fusion, feature subsets generated by each dimensionality reduction method are combined into a single feature set and then kNN is applied. The experiment described in Paper III considers an equal number of features from each dimensionality reduction method for fusion. In previous studies, little attention has been given to the fusion of kNN classifiers. Bay [25] considered combining kNN classifier outputs for random subsets of features. His work is primarily based on datasets of moderate size. The fusion of dimensionality reduction methods in high dimensions has been investigated in a variety of ways. For example, in some studies, one dimensionality reduction method is applied after another, as in [31] and [23]. In contrast to the preceding studies, the experiment presented in Paper III focuses on the combination of features and classifiers from three-dimensionality reduction methods for high-dimensional datasets. The experimental results indicate that both the classifier fusion and feature fusion methods outperform the individual methods. Furthermore, the experiment demonstrates that the feature fusion method not only achieves the highest classification accuracy in the majority of cases, but also performs well when the number of dimensions is varied.

In Paper IV, whether further accuracy enhancement can be obtained by fusing dimensionality reduction methods using different numbers of dimensions is investigated. The different dimensionality reduction methods may achieve the highest accuracy with different numbers of dimensions. The optimal number of dimensions is defined here as the number of dimensions that results in the highest accuracy for each method. We consider combining the outputs of the optimal number of dimensions for the three dimensionality reduction methods in the experiment described in Paper IV. The results indicate that both classifier and feature fusion using the optimal number of features for each

dimensionality reduction method perform moderately better than the previous methods. Furthermore, the experiment demonstrates that classifier fusion is more effective than feature fusion when the optimal number of dimensions is considered. This may be due to the fact that the feature fusion method frequently generates a higher number of dimensions than the other methods, and hence the combined feature set may be suboptimal. As a result, classifier fusion is recommended when the outputs of multiple dimensionality reduction methods are combined using the optimal number of dimensions.

The effects of the selection of dimensionality reduction methods were explored in Paper V. The comparative study was based on feature and classifier fusion methods, which were used to choose the dimensionality reduction strategies that would be used in the fusion. We have considered the outputs of four dimensionality reduction methods, namely PCA, PLS, IG and ReliefF, on 18 medicinal chemistry datasets. In the results of Paper IV, it was observed that integrating an equal number of features from each reduced feature set yielded the highest classification accuracy for the nearest neighbor classifier. Therefore, we have considered an equal number of features from the reduced features using the PCA, PLS, IG and ReliefF techniques. We have examined the number of features for classifier fusion that results in the highest accuracy for ten-fold cross-validation. The classifier results were combined using majority voting. The results demonstrated that classifier fusion improves the accuracy irrespective of the selection of the dimensionality reduction methods whereas feature fusion was quite sensitive to the selection of the dimensionality reduction methods.

- **Paper VI**: Sampath Deegalla, Keerthi Walgama, Panagiotis Papapetrou and Henrik Boström. Random Subspace and Random Projection Nearest Neighbor Ensembles for High Dimensional Data, *Expert Systems with Applications*, 191:116078, 2022.

Following the results of Paper V, we considered forming nearest neighbor ensembles for different types of high-dimensional datasets. Two dimensionality reduction methods, namely random subsets and RP were investigated on 34 microarray, medicinal chemistry and image data sets.

Since common procedures such as bagging [29] and boosting [30] were shown to be ineffective for the nearest neighbor method [46], combining different feature subsets was considered in this study [47]. The results show that the best method for forming nearest neighbor ensemble depends on the type of data being considered. It is further observed that the nearest neighbor ensemble may even outperform state-of-the-art techniques such as random forests for certain types of data.

The author's individual contributions: The design and implementation of the experiments, initial draft preparation and revision of the papers up to the final version were done by the dissertation's author for all the mentioned papers.

1.5 Outline of the Thesis

The outline of the thesis is as follows. Chapter 2 provides an overview of the nearest neighbor classifier, fusion strategies, and the details of the nearest neighbor ensemble. Methods for dimensionality reduction are discussed in Chapter 3. The results of the empirical studies are discussed in greater detail in Chapter 4. Finally, Chapter 5 summarises the contributions of this thesis and future research directions. The appendix contains the six papers upon which this thesis is based.

Chapter 2

Nearest Neighbor Classifier

The nearest neighbor algorithm is one of the oldest and most successful algorithms in machine learning. The method is mainly used in classification and regression. However, it has also been extended and used in clustering [48] and semi-supervised learning [49]. This chapter briefly reviews the nearest neighbor algorithm and then discusses nearest neighbor ensembles. Finally, we suggest Aha et al. [14] for readers interested in nearest neighbor learning.

2.1 Introduction

The k nearest neighbor (kNN) algorithm belongs to the family of instance-based learning methods [14]. The general kNN algorithm does not create a model and therefore produces no output: it only stores instances during the training phase. Classification is delayed until a new instance is to be classified. This is in contrast to learning algorithms such as the decision tree algorithm, which creates a generalised tree-based model. Therefore, kNN is referred to as a lazy learning algorithm. Algorithms such as locally weighted regression methods [2] employ a similar concept.

The new instance is assigned a class based on the labels of the training examples that are a minimum distance from the new instance. When more than one nearest neighbor instance is considered for the final decision, the algorithm is known as kNN, where k denotes the number of nearest training instances. The underlying assumption, i.e. inductive bias, of the nearest neighbor classifier is that the class of the test instance could be determined by its neighbourhood.

The advantage of kNN is that it estimates the outcome locally and differently for each new instance [2]. This means that the decisions made by kNN can be easily tracked by examining the nearest neighbouring points in the dataset [50, 51]. Furthermore, distance metrics, such as the Euclidean distance, provide direct insight into which features are influencing the decision, leading to an interpretable model. However, the cost of calculating the distance is relatively high since all the calculations take place during classification. To handle the computational complexity of distance calculations, efficient indexing structures such as kD-trees and ball trees are used [46]. Another disadvantage is that kNN considers all the features equally important, while the concept description may depend only on a few available features. Dimensionality reduction and feature weighting could be used to overcome this problem.

In order to understand how kNN classifies a test instance, consider Figure 2.1 which concerns a binary classification problem that includes Class A and Class B. Let us assume that we have several instances and their respective features. We are interested in separating Class A from Class B based on two features, denoted by Feature 1 and Feature 2. The two axes of Figure 2.1 represent the two features, and the points represent instances from Class A and Class B. Class A instances are represented by circles (○) and Class B instances are represented by squares (□). The grey circle (●) is the test instance to be classified using the nearest neighbor algorithm, i.e. we are interested in finding out whether this instance belongs to Class A or not. Assuming $k = 1$, kNN finds the nearest training instance to the test instance, and thereby assigns the test instance as a square. Therefore,

the new instance in the example is classified as Class B. When one chooses $k > 1$, say $k = 5$, the algorithm seeks the nearest five training instances, which consist of three circles and two squares in the example. The test instance is assigned to the majority class of the neighbourhood and thereby is predicted to be in Class A. The parameter k in kNN is usually found by cross-validation.

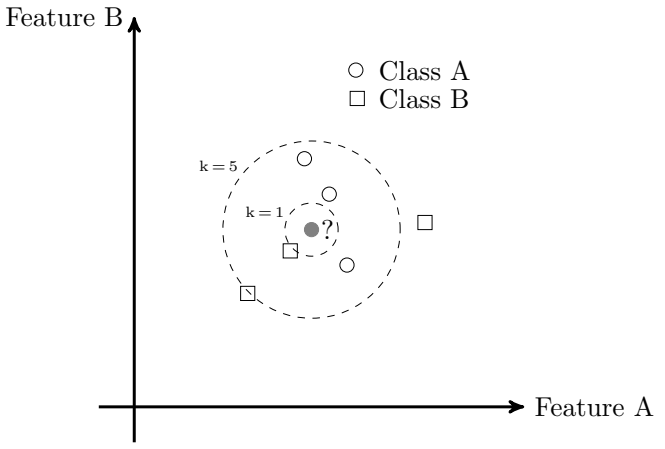

Figure 2.1: An example of nearest neighbor classification.

As we have seen in the example, kNN predicts the class of a test instance using the distance between instances, and there are many ways to calculate this distance. In this thesis, we consider the geometric distance between instances, i.e. the Euclidean distance metric [2]. The Euclidean distance is defined in Eq. 2.1 as follows:

$$d(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{\sum_{a=1}^{m} (\mathbf{x_{ia}} - \mathbf{x_{ja}})^2} \tag{2.1}$$

Here $\mathbf{x_i}$ and $\mathbf{x_j}$ refer to two instances whereas $d(\mathbf{x_i}, \mathbf{x_j})$ refers to the Euclidean distance between them. m refers to the total number of features and a refers to one of the m features. $\mathbf{x_{ia}}$ and $\mathbf{x_{ja}}$ denote the values for feature a of $\mathbf{x_i}$ and $\mathbf{x_j}$.

2.2 Nearest Neighbor Classifiers based on Fusion of the Outputs of Dimensionality Reduction

In this section, we consider forming nearest neighbor classifiers based on the outcomes of the dimensionality reduction methods, i.e., feature fusion and classifier fusion.

2.2.1 Feature Fusion

Feature fusion concerns how to generate and select a single set of features for a set of objects to which several sets of features are associated.

We have proposed the following two feature fusion strategies to generate the nearest neighbor classifier:

- An equal number of features from each dimensionality reduction method are considered for classification with kNN. Therefore, the total number of dimensions selected for classification is in the range of $n_d, 2 \times n_d, \ldots, d_r \times n_d$, where d_r is the reduced dimension. For each set of features, the first d_r/n_d reduced dimensions are chosen from the output of n_d dimensionality reduction methods.
- The minimum number of features required to yield the highest classification accuracy for each dimensionality reduction method is considered. Cross-validation is performed to find this number.

2.2.2 Classifier Fusion

The focus of classifier fusion is on either generating a structure representing a set of combined classifiers or combining classifier outputs. We have

considered the latter approach, i.e. combining nearest neighbor predictions when they are used in conjunction with dimensionality reduction methods.

The following classifier fusion strategies were proposed:

- For each dimension, nearest neighbor predictions from each dimensionality reduction method are combined using unweighted voting, i.e. giving equal weight to the output of each nearest neighbor classifier and selecting the majority output.
- The number of dimensions that results in the highest accuracy for each dimensionality reduction method, as estimated using cross-validation on the training set, is selected. The outputs are then fused using unweighted voting.

Both classifier fusion methods may lead to ties for multiclass problems which are resolved by randomly selecting one of the class labels that achieved the highest number of votes.

2.3 Nearest Neighbor Ensemble Algorithm

Nearest neighbor algorithms are known to be stable for variations in datasets according to Breiman [29]. This is mainly due to the high correlation of errors when the instance space is sampled. Therefore, manipulating the different feature spaces was considered to form nearest neighbor ensembles [47].

We have considered nearest neighbor ensembles based on randomness, i.e. using the random subset and RP projection methods.

The steps for forming nearest neighbor ensembles are as follows:

- Input:
  - the dataset,
  - the ensemble size s,
  - the number of features in each subset f,
  - the number of nearest neighbors k.
- Procedure:
  - For each base classifier from 1 to s,
    - Select f features randomly from the original dimension (or transform the original dimension into f new components using RP) in the training.
    - Select the same features (or transform the original dimension using the same random matrix) in the testing.
    - According to the Euclidean distance, find the class labels of the k closest instances and predict the final class using majority voting.
  - Using class labels among s base classifiers, find the final class of the ensemble using majority voting
- Output: the nearest neighbor ensemble.

Chapter 3

Dimensionality Reduction Methods

3.1 Introduction

Dimensionality reduction may be a critical step for the success of a learning algorithm in high dimensions. One benefit of dimensionality reduction is that it reduces the computational cost of the learning algorithm. For instance, it reduces the training time for eager learning methods such as decision trees (Iterative Dichotomiser 3 -ID3 [52]) and a reduction in testing time for lazy learning methods such as the kNN classifier. In decision tree learning, a suitable attribute is searched for at each (non-leaf) node of the tree. In ID3, the feature with the highest information content for the node is chosen, and the process is then recursively repeated. It is hence possible to reduce the time needed for the construction of decision trees by reducing the number of features. A similar procedure is applied to kNN learning during testing since reducing the number of features may lead to simpler distance calculations. In most cases, this procedure may help to improve both the efficiency and effectiveness of the kNN algorithm [23, 53].

Dimensionality reduction methods can be classified into feature extraction and feature selection methods. By using feature extraction methods, the original feature set is transformed into a new feature set that retains almost all of the information in the original features whereas in feature selection, a subset of original features is kept and the rest are discarded. Furthermore, feature extraction methods can be divided into linear methods through which new attributes are defined as linear combinations of the original attributes and non-linear methods through which new attributes capture the non-linearity in the data. On the other hand, feature selection can be split into filter [54] and wrapper [32] methods. In filter methods, a subset of the feature set is selected independently of the learning algorithm. In wrapper methods, the feature selection method employs a predetermined learning algorithm in order to find the optimal feature subset [55]. However, non-linear feature extraction methods and wrapper methods tend to be computationally expensive for high-dimensional datasets. Therefore, we consider five dimensionality reduction methods: three of them are linear feature extraction methods, i.e. principal component analysis (PCA), random projection (RP) and partial least squares (PLS), and two are filter methods, i.e. information gain (IG) and ReliefF. These methods will be described in more detail in the next section.

Some notions used in the subsequent section are described here. An input dataset is described as an $n \times m$ data matrix X, which contains n instances and m original dimensions. The reduced dataset is denoted by an $n \times d_r$ matrix Z, which contains n instances and d_r ($< m$) dimensions.

3.2 Principal Component Analysis

Principal component analysis [56, 57] transforms the original feature set into a new set of orthogonal features referred to as principal components. A principal component is a linear combination of original features with optimal weights. The first principal component contains most of the variation in the data and every subsequent component contains most of the remain-

ing variation. Therefore, PCA could be used to reduce a large number of dimensions to a much smaller number of dimensions.

There is recent research [58] that criticises the reliability of the outcomes derived from PCA and primarily focuses on its application in population genetics. PCA may work well for our specific needs since we work in a different domain. Given our work in high dimensions, PCA might be promising for the nearest neighbor method due to its ability to perform dimensionality reduction.

The first step is subtracting the mean from X along all the dimensions. Let C be the covariance matrix of X, as shown in Eq. 3.1:

$$C = \frac{1}{n-1} X^T X \tag{3.1}$$

where C is an $m \times m$ symmetric matrix that captures the correlations between all the features. The reduced dataset Z can be written as shown in Eq. 3.2:

$$Z = X E \tag{3.2}$$

where E is a $d_r \times m$ projection matrix that consists of d_r eigen vectors corresponding to the d_r highest eigen values of the covariance matrix C.

Recent implementations of PCA are based on singular value decomposition (SVD) which is known as a fast and numerically stable method [59].

In SVD, the original data matrix X can be decomposed as shown in Eq. 3.3:

$$X = U D V^T \tag{3.3}$$

where U is an $n \times m$ orthonormal matrix with left singular vectors, V is a $d_r \times m$ orthonormal matrix with right singular vectors and D is an $m \times m$ diagonal matrix that contains the singular values. Here m refers to

the rank of X which is min(n,m).

Considering the first d_r highest singular values, where $d_r < min(n, m)$, the transformation can be rewritten as shown in Eq. 3.4:

$$X_{n\times p} = U_{n\times r}\, D_{r\times r}\, V^T_{r\times p} \tag{3.4}$$

The principal component scores can be calculated as depicted in Eq. 3.5:

$$Z = X\, V \tag{3.5}$$

Here Z is the reduced dataset obtained after PCA is applied to the original dataset X. We have used the MATLAB implementation of PCA which is based on SVD in empirical studies.

3.3 Random Projection

Random projection is a simple and computationally efficient technique for reducing dimensionality. Furthermore, compared to PCA, which has a worst-case complexity of $O(m^2n + m^3)$, RP has a worst-case complexity of $O(d_rmn)$, meaning that it is much faster in high dimensions. With a small amount of error, RP can achieve faster processing times and smaller model sizes. It is based on the Johnson-Lindenstrauss lemma [60] (see Lemma 3.3.1), which states that a set of points in high dimensions could be mapped into low dimensions while preserving the Euclidean distance between the points by a small arbitrary factor.

Lemma 3.3.1 *(as in [61]) Given $\epsilon > 0$ and an integer n, let d_r be a positive integer such that $d_r \geq d_0 = O(\epsilon^{-2} \log n)$. For every set P of n points in $\mathbb{R}^m$, there exists $f : \mathbb{R}^m \to \mathbb{R}^{d_r}$ such that for all $u, v \in P$,*

$$(1 - \epsilon)||u - v||^2 \leq ||f(u) - f(v)||^2 \leq (1 + \epsilon)||u - v||^2. \tag{3.6}$$

Dasgupta and Gupta [62] identified that such a mapping is possible with normally distributed values in a random matrix. Later, researchers investigated probabilistic approaches to populate the random matrix, and the method proposed by Achlioptas [61] is widely recognised [33, 34] due to its fast and relatively simple implementation. The method has been investigated for several different types of high-dimensional datasets, text documents [36, 33], images [35, 33] and microarrays [34].

The transformation can be given as shown in Eq. 3.7:

$$Z = X\,R \tag{3.7}$$

where X is the $n \times m$ data matrix, R is an $m \times d_r$ random matrix and Z is the transformed data matrix.

Achlioptas [61] proposed the following two probability distributions to populate the random matrix, given as Eqs. 3.8 and 3.9:

$$r_{ij} = \begin{cases} +\sqrt{3} & \text{with} \quad P_r = \frac{1}{6}; \\ 0 & \text{with} \quad P_r = \frac{2}{3}; \\ -\sqrt{3} & \text{with} \quad P_r = \frac{1}{6}. \end{cases} \tag{3.8}$$

$$r_{ij} = \begin{cases} +1 & \text{with} \quad P_r = \frac{1}{2}; \\ -1 & \text{with} \quad P_r = \frac{1}{2}. \end{cases} \tag{3.9}$$

We have considered the default probability distribution implemented in the Waikato Environment for Knowledge Analysis (WEKA) [21] data mining tool for the RP method. However, the number of features to be retained in the transformed data, i.e. d, has to be chosen by the user. In the first two studies, we have recorded the classification accuracy for various values of d. In the last study, we set d to 10%,20%,...,100% of the original features in experiment #1 and 10% of the original features for experiment #2.

3.4 Partial Least Squares

The partial least squares method, which is also known as projection to latent structures [63, 59], was developed and is widely used in chemometrics, primarily as a regression technique. It seeks the linear combination of original features with the highest correlation with the output variable (class values). The resulting components are uncorrelated, i.e. there is no correlation between the new features. PLS is well-known for its applicability when the number of features is much higher than the number of instances. Later, it was used as a dimensionality reduction technique in the context of high-dimensional datasets [43, 44, 45, 37]. Since PLS considers the class labels for the transformation, it is considered one of the most effective supervised dimensionality reduction methods.

The basic PLS regression model can be represented as shown in Eq. 3.10:

$$Y = X\,B + E \tag{3.10}$$

where Y is an $n \times p$ response matrix (class matrix), X is an $n \times m$ data matrix, B is an $m \times p$ regression coefficient matrix, E is the $n \times p$ noise matrix and p is the number of dependent variables (class variables). Here, X and Y are centred by subtracting their respective means. The regression model could be developed as follows.

The factor score matrix Z can be found as shown in Eq. 3.11:

$$Z = X\,W \tag{3.11}$$

with the appropriate weight matrix W. The model is equivalent to the model shown in Eq. 3.12:

$$Y = Z\,Q + E \tag{3.12}$$

where Q is the regression coefficient for Z. Once the loadings Q are computed, the above regression model is equivalent to $Y = X\,B + E$, where

$B = W\,Q$. Here we are interested in dimensionality reduction, i.e., generating factor scores, Z, as shown in Eq. 3.13

$$Z = X\,W \tag{3.13}$$

The SIMPLS algorithm [38] has been used to perform PLS in our studies since it could be applied to both binary and multiclass classification problems. Although PLS was originally developed for continuous output (continuous response), we must use it for categorical output (categorical response). However, categorical outputs can be treated as continuous outputs in binary class problems since the PLS method does not employ any distributional assumptions [37]. Nevertheless, the class variables of multiclass problems are transformed using the following dummy coding, as described in [37], so that they can be used with the SIMPLS algorithm. Here, the class variable Y is transformed into a C-dimensional random vector, where C is the number of classes, as described in Eq. 3.14:

$$\begin{array}{rcll} y_{ij} & = & 1 & \text{if } Y_i = c \\ y_{ij} & = & 0 & \text{otherwise} \end{array} \tag{3.14}$$

where Y_i is the class label for the ith instance and y_{ij} is the ith row of the random vector for all i=1,. . . ,n and j=1,. . . ,C. Therefore, for binary problems Y can be used, while the transformed random vector is used for multiclass problems.

3.5 Information Gain

Information gain is a technique frequently used in decision tree induction, e.g. ID3, and it is used to determine which feature to choose from a set of features to use in a node of the tree [21]. This is a filter-based method for selecting features in which features are ranked according to their information gain and then the best d features are chosen. This method has been investigated as a dimensionality reduction method for high-dimensional datasets,

such as texts [64] and microarrays [23].

Information gain can be generally stated as shown in Eq. 3.15:

$$IG(class|feature) = H(class) - H(class|feature) \tag{3.15}$$

where $H(class) = -p \log_2 p$ denotes the entropy of the class, and $H(class|feature)$ denotes the conditional entropy of the feature with respect to the class.

Maximising the IG is equivalent to minimising the conditional entropy of the feature, i.e. $H(class|feature)$, as described in Eq. 3.16:

$$\sum_{i=1}^{V} \frac{n_i}{N} \sum_{j=1}^{C} -\frac{n_{ij}}{n_i} \log_2 \frac{n_{ij}}{n_i} \tag{3.16}$$

Here, C is the number of classes, V is the number of values of the feature, N is the total number of instances, n_i is the number of instances with the ith value of the feature and n_{ij} is the number of instances in the latter group belonging to the jth class.

All the attributes in the datasets used for experiments are numerical attributes, whereas IG assumes that all attributes are nominal. In order to calculate the information gain, one should discretise the numerical data prior to applying the IG algorithm. We have used WEKA for IG in the experiments presented in Papers II through V where Fayyad and Irani's minimum description length (MDL) [65] method is used as the default method of discretisation.

3.6 ReliefF

Kira and Rendell [39] developed the Relief algorithm which determines the quality of the attribute of two-class problems, and Kononenko [40] extended the original Relief to develop ReliefF to deal with noisy, incomplete

and multiclass problems. The weight of a feature is calculated by maximising the ratio between the distance of the closest instances from the same class and the distance of the closest instances from different classes. The method may be particularly suitable for the nearest neighbor method since it is inspired by instance-based learning [39]. This is also a filter based feature selection method in which the best d_r features are considered for the classification based on the relevance level and a threshold [39].

The algorithm chooses m instances at random. For each instance, the algorithm finds the closest instance from the same class (near-hit) and the closest instance from a different class (near-miss). In each iteration, a triplet, i.e. the chosen instance, its near-hit and its near-miss, is considered for calculating the weight vector W as shown in Eq. 3.17:

$$W_i - \frac{1}{m}\{(x_i - \text{near-hit}_i)^2 + (x_i - \text{near-miss}_i)^2\} \text{ for all } i = 1, \ldots, o \quad (3.17)$$

Here, W_i is the weight of the i th feature in the previous iteration, x is an instance from m instances, x_i is the value of the i th feature, near-hit$_i$ is the value of the i th feature of the nearest instance from the same class as x and near-miss$_i$ is the value of the i th feature of the nearest instance from a class that is not the class of x.

We have used the ReliefF algorithm in the WEKA data mining toolkit in the experiments presented in Paper V.

3.7 Random Subspace

The random subspace method [66] has been used as a feature selection technique, i.e. for selecting a subset of features d from the original feature set o, as well as an ensemble forming technique, i.e. for combining individual classifiers formed by using a random subset of features. Bay [25] proposed the use of the random subspace method to form nearest neighbor ensembles while Ho [66] proposed the method to form a decision forest. The random subspace method has been considered as a dimensionality reduction method

and an ensemble forming method for fMRI data and microarray data [67, 68].

3.8 Summary

In this chapter, we have explored the various dimensionality reduction methods considered in our research. Principal component analysis and random projection are discussed as unsupervised dimensionality reduction techniques. In contrast, partial least squares, information gain, and ReliefF are presented as supervised dimensionality reduction methods. Finally, random subspace and random projection are considered for feature selection and ensemble formation.

Chapter 4

Empirical Studies

A summary of the six papers included in the dissertation is presented in this chapter. A summary of each paper is included; the main contributions, details of the experiments and limitations of each study are discussed.

4.1 Paper I: Reducing High-Dimensional Data by Principal Component Analysis vs. Random Projection for Nearest Neighbor Classification

This paper examines how to improve the performance of the nearest neighbor classifier in high-dimensional datasets. Dimensionality reduction is used as a preprocessing step prior to the use of the nearest neighbor algorithm.

Two dimensionality reduction methods, i.e. PCA and RP, have been empirically investigated for two types of data: microarrays and images. The effectiveness and efficiency of both methods when they are used with nearest neighbor classification is investigated. Ten publicly available high-dimensional datasets, consisting of images and microarrays, are chosen for evaluation.

Table 4.1: Highest prediction accuracy obtained by the nearest neighbor classifier with dimensionality reduction methods (no. of dimensions in parentheses). Reproduced from Paper I.

Data set	RP		PCA		Original
IRMA	67.01	(250)	**75.30**	(40)	68.29
COIL100	98.79	(250)	98.90	(30)	**98.92**
ZuBuD	54.01	(250)	**69.46**	(20)	59.80
MIAS	44.05	(5)	**53.76**	(250)	43.17
Outex	21.04	(15)	**29.12**	(10)	19.85
Colon Tumor	80.22	(150,200)	**83.05**	(10)	77.42
Leukemia	91.32	(150)	**92.83**	(10)	89.47
Central Nervous	58.22	(150)	**66.33**	(50)	56.67
Srbct	93.23	(200)	**96.45**	(10)	87.30
Lymphoma	97.80	(250)	**99.86**	(20)	98.38

Kaski [36] and Fern and Brodley [35] compared both PCA and RP for clustering. Kaski investigated both methods for constructing self-organising maps. He concluded that the use of RP is as good as the use of PCA for forming self-organising maps. Fern and Brodley showed that the use of RP is better for forming clusters than PCA. Bingham and Mannila [33] compared both methods in terms of the amount of distortion compared to the original data and the computational complexity. Fradkin and Madigan [34] compared PCA and RP for supervised learning. They compared both methods for decision trees, SVM and kNN (k=1 and k=5). The results of their study revealed that in general, PCA outperforms RP. In contrast to the above studies, we consider kNN learning in high dimensions.

The contributions of the paper includes a comparative study of the two dimensionality reduction methods for nearest neighbor classification in high dimensions. Both methods improved upon using the original features in a majority of the cases. PCA outperformed RP in all the cases, but no method outperformed the classification accuracy of using the original

features in all the cases.

4.2 Paper II: Classification of Microarrays with kNN: Comparison of Dimensionality Reduction Methods

This paper extends the previous study by incorporating two additional dimensionality reduction methods that consider the class information during the construction of the reduced feature space. Furthermore, the investigation has been limited to considering only microarrays to focus on a small number of instances. In the previous study, we considered unsupervised dimensionality reduction, which means that neither of the methods takes class information into account during the transformation of the original features. It is reasonable to believe that incorporating class information into dimensionality reduction may further enhance the learning algorithm's performance. Therefore, PLS and IG are considered in addition to PCA and RP for nearest neighbor classification.

The four methods have been investigated separately for high-dimensional datasets (microarrays). Dai et al. [45] compared three feature extraction methods including PCA and PLS for supervised learning. Nguyen and Rocke [43] compared PLS to PCA for two statistical learning algorithms. Ghosh [69] showed that PCA is successful for classification and feature selection while removing possible correlations between features. Li et al. [26] compared IG with several other feature selection methods on multiclass classification problems. However, the performances of these methods for nearest neighbor classification may not be comparable due to differences in the experimental setup, e.g. the use of different evaluation strategies. Furthermore, none of the previous studies considered the four dimensionality reduction methods for the nearest neighbor classifier. On the other hand, at the time of our study, we found no earlier study that used PLS as a dimensionality reduction method for nearest neighbor classification.

The four dimensionality reduction methods have been investigated using eight publicly available microarray datasets using stratified ten-fold cross validation. Since both PLS and IG use class information in the transformation, the weight matrix generated by PLS during training is also used for the test set, whereas the same attributes selected in the training set by IG are chosen for the test set as well. To alleviate the variable performance of RP, the experiment was repeated 30 times with different samples. Furthermore, odd numbers of nearest neighbors from 1-9 were considered to address the noise effect on nearest neighbor classification in high dimensions.

Dimensionality reduction based on all methods is indeed effective for nearest neighbor classification in a majority of the cases; however, no method turned out to be a clear winner on the datasets considered in the study. Nevertheless, PLS was superior to all the other methods for binary classification problems. The single nearest neighbor method, i.e. $k = 1$, is better in most of the cases, whereas a higher k value is generally preferred for IG. Therefore, it is concluded that choosing the appropriate dimensionality reduction method for the nearest neighbor classifier is a major concern when data with a high number of dimensions.

4.3 Paper III: Fusion of Dimensionality Reduction Methods: A Case Study in Microarray Classification

Two strategies for combining the outputs of dimensionality reduction methods are considered in this paper, i.e. the fusion of an equal number of features in the reduced dimensions and the fusion of classifier outputs, where each classifier is generated using the same number of dimensions. The experiment shows that the fusion methods outperform the individual dimensionality reduction methods in a majority of the cases. In particular, the feature fusion method generally gives a higher accuracy.

Recently, classifier fusion has attracted more attention than feature fusion in supervised learning. However, common classifier fusion strategies such as bagging and boosting do not improve the performance of nearest neighbor learning [25]. The common classifier fusion strategy is to combine classifier outputs from single classifiers which are applied to separate subsets of features using a voting mechanism, e.g. [25] and [47]. Bay [25] investigated a classifier fusion method that is based on using an equal number of features from random sampling in the feature space and then combining classifiers using simple voting. In the same fashion, Domeniconi and Yan [47] compared a classifier fusion method based on random and weighted feature subsets using several voting schemes. Similarly, Skurichina and Duin [70] considered combining classifiers from different feature subsets in the reduced feature space using PCA. In all of the above studies, a large number of classifiers were considered for fusion. In contrast, we consider combining both the features and classifiers that are obtained from multiple dimensionality reduction methods.

This study is designed to investigate whether the classification accuracy of combining the outputs of different dimensionality reduction methods is better than the classification accuracy of the individual dimensionality reduction methods. The null hypothesis is that there is no difference between combining the outputs of dimensionality reduction methods and the individual dimensionality reduction methods for the nearest neighbor algorithm. To test the hypothesis, we consider the same datasets used in the previous study with PCA, PLS and IG. [1]

In the paper, we have reported the classification accuracies for the feature fusion method without normalising the number of features along the dimension and the classification accuracies of the other methods. Therefore, the highest classification accuracies of all the studied methods are shown

[1] The random projection method, which requires an additional computational burden with respect to the gain in the performance, has been abandoned for this study.

in Table 4.2. Two graphs from the study are reproduced in Figure 4.1.

Table 4.2: Results in terms of the best classification accuracies.

Dataset	Raw	PCA	PLS	IG	FF	CF
Central Nervous	56.67	70.00(31)	73.33(26)	68.33(18)	73.33(20)	71.67(11)
Colon Tumor	77.42	84.05(10)	88.81(4)	84.52(14)	87.14(7)	87.38(7)
Leukemia	89.47	95.00(10)	95.00(5)	96.67(32)	100.00(11)	97.50(4)
Prostate	85.29	86.27(25)	92.36(15)	93.27(11)	95.18(29)	95.09(50)
Brain	76.19	86.00(4)	79.00(23)	81.00(34)	88.00(32)	86.00(4)
Lymphoma	98.39	100.00(2)	100.00(5)	100.00(15)	100.00(2)	100.00(2)
NCI60	68.85	80.24(6)	78.57(11)	68.81(24)	80.24(17)	80.24(6)
SRBCT	87.30	96.67(10)	100.00(4)	100.00(10)	100.00(4)	100.00(4)

The results in Table 4.2 show that both fusion strategies, i.e. feature fusion and classifier fusion, outperform the other methods, i.e. classification using the original features (referred to as Raw) and the individual dimensionality reduction methods, in a majority of the cases. This means that the use of fusion may be better than the use of any single dimensionality reduction method alone. The feature fusion method outperforms all the other methods for all but the Colon Tumor dataset [6] and generally is robust with respect to changes in the number of dimensions. Therefore, one may recommend fusing the output of dimensionality reduction methods instead of using any of the individual methods alone.

4.4 Paper IV: Improving Fusion of Dimensionality Reduction Methods for Nearest Neighbor Classification

In this paper, an extension to the approach proposed in Paper III is considered, i.e. a different number of features may be selected from each dimensionality reduction method. In the previous study, we consider only the same number of features when combining the output of the dimen-

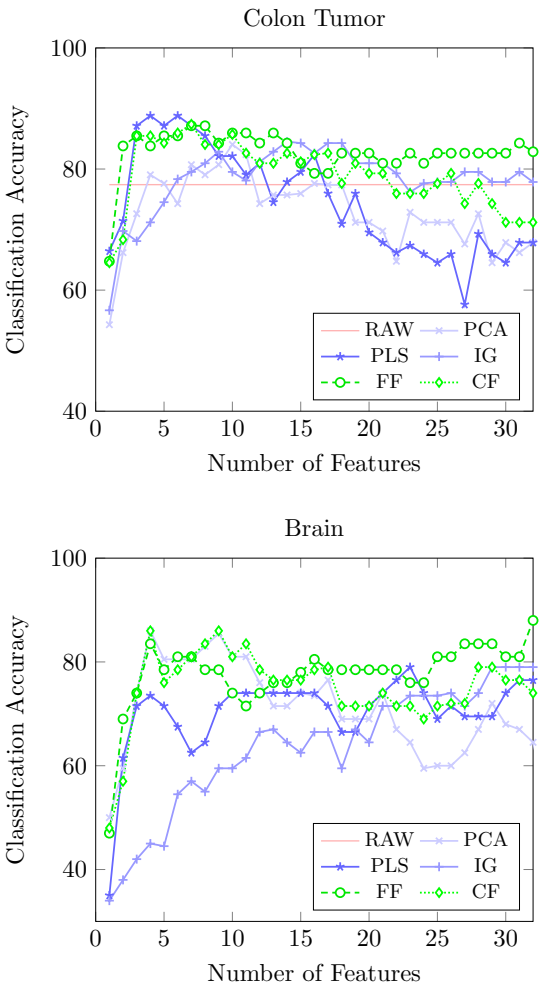

Figure 4.1: Two graphs from the third study obtained after normalising the accuracies of feature fusion along the dimension. Graphs are reproduced for two datasets: a binary classification problem (Colon Tumor) and a multi-class classification problem (Brain)

sionality reduction methods. However, different dimensionality reduction methods may yield the best performance using different numbers of dimensions. This suggests that the performance of the nearest neighbor classifier could be improved if one chooses the number of dimensions separately for each dimensionality reduction method. Therefore, we investigate fusing the outputs of multiple dimensionality reduction methods with different numbers of dimensions for each dimensionality reduction method.

The experimental setup of the empirical study has been slightly altered compared to the previous studies, in which the performance of kNN was reported for all numbers of dimensions that were considered. In this study, we consider only the performance of kNN for the optimal number of dimensions, i.e. the number of dimensions that gives the highest performance for kNN. The optimal number of dimensions for a particular method, i.e. for an individual dimensionality reduction method or fusion method, is selected using cross-validation. First, the training and testing sets are generated to evaluate the final accuracy of the nearest neighbor classifier using 10-fold cross-validation. Then, for each dimensionality reduction method, the number of features that results in the highest classification accuracy is found using 10-fold cross-validation on the training set. To transform the test set using PCA and PLS, the weight matrix generated from the training set is used. The same features selected in the training set using IG are chosen for the test set. Finally, the features and classifiers are fused. In the former case, the combination of the optimal number of features from each reduction method is used to create a single feature set, while in the latter case, the nearest neighbor classifier outputs from each reduction method with an optimal number of features are combined using majority voting. In all cases, a single nearest neighbor is considered.

Both fusion methods further improved the nearest neighbor classifier accuracy to some extent. The proposed methods outperform the raw accuracy in a majority of the cases. The proposed classifier fusion method gives the best accuracy for the NCI dataset [71] and outperforms the fusion methods studied in the previous study in half of the cases. However, the

performance of the proposed feature fusion method is quite low compared to the previous fusion methods. The reason for this could be the higher number of features that are selected by this method, which may lead to poor performance compared to other methods. Therefore, we suggest using classifier fusion instead of feature fusion when combining the outputs of dimensionality reduction methods that are based on choosing the optimal number of dimensions.

4.5 Paper V: Choice of Dimensionality Reduction Methods for Feature and Classifier Fusion with Nearest Neighbor Classifiers

An empirical study on how feature fusion and classifier fusion works in conjunction with the selection of dimensionality reduction methods was considered. In previous studies, we have investigated using feature and classifier fusion techniques based on the outputs of multiple dimensionality reduction methods. We have extended the previous studies by investigating the choice of the dimensionality reduction method when both features and classifiers are fused. None of the previous studies considered the selection of dimensionality reduction methods when the outputs of dimensionality reduction methods were combined for fusion.

We have considered four dimensionality reduction methods, PCA, PLS, IG and ReliefF, and two strategies for fusing their outputs for 18 medicinal chemistry datasets using two different representations. Using dimensionality reduction methods, the original number of dimensions was transformed into a lower number of dimensions, and then feature fusion and classifier fusion were performed. Equal numbers of features from the reduced dimensions were selected in feature fusion, whereas the optimal number of features to achieve the highest accuracy was selected in classifier fusion. The selected number of features for the different fusion strategies was based on the results of previous studies.

It is observed that the classifier fusion of all the combinations of dimensionality reduction outputs is better than individual reduction methods. However, the performance of feature fusion is quite sensitive to the selection of dimensionality reduction methods.

4.6 Paper VI: Random Subspace and Random Projection Nearest Neighbor Ensembles for High Dimensional Data

This study was designed to investigate generating nearest neighbor ensembles using dimensionality reduction; they were tested on 34 microarray, chemoinformatics and image datasets. The nearest neighbor ensembles were generated using random subspace and random projection methods. The results show that both ensemble methods improved the accuracy compared to the standard nearest neighbor classifier. However, it is observed that the best method depends on the type of data. Furthermore, the study also showed that the accuracy of the ensemble may even be competitive with state-of-the-art methods such as the random forest for microarray and chemoinformatics datasets.

Bagging [29] and boosting [30] were considered common procedures for generating ensembles. However, those techniques failed to improve the performance of nearest neighbor ensembles [25]. Therefore, manipulating the feature space was considered, in contrast to manipulating the instance space. There are several related works on the use of nearest neighbor ensembles [25, 47] with random subsets [66, 25, 47, 67] and random projections [72]. However, none of the studies considered investigating the two methods for high-dimensional data sets.

We have performed two experiments in this study. In the first experiment, the number of nearest neighbors (k) was set to 1, and the number of base classifiers (s) was set to 50. The number of features to be selected

for the ensemble varied by intervals of 10% of the total number of features. The optimal number of features required to have the best accuracy was investigated using internal cross-validation. Majority voting was employed to obtain the ensemble performance.

In the second experiment, we considered 10% of the total number of features as the number of features for the component classifiers in the ensemble; this is based on the outcome of the first experiment. The number of nearest neighbors (k) and the number of base classifiers (s) in the ensemble varied, and the ensemble performance was observed. To compare the ensemble performance to the performance of other classifiers, decision tree, random forest and support vector classifiers were included.

Table 4.3: Average ranks of the nearest neighbor algorithm (Raw) and the two nearest neighbor ensembles constructed using random subsets (RS) and random projections (RP) (Source: Paper VI, Table 5)

Data set	Raw	RS	RP
All	2.28	**1.84**	1.88
Microarray	2.38	2.44	**1.19**
Chemoinformatics	2.08	**1.83**	2.08
Image	2.63	**1.25**	2.13

The results show that the ensemble accuracy is better than that of the nearest neighbor classifier except for the ensemble using random subsets for microarray data. When considering all the data sets, it is observed that RS is better than all the other classifiers. Furthermore, the results show that the nearest neighbor ensemble using random subsets is the best for medicinal chemistry data and image data sets, while the nearest neighbor ensemble using random projection is the best for microarrays. The rank of each nearest neighbor ensemble is determined, with the algorithm with the

best performance being assigned a rank of one.

The codebase for the empirical investigations mentioned in this chapter can be found at `https://github.com/dsdeegalla/PhD`.

Chapter 5

Concluding Remarks

This thesis has studied the improvement of the nearest neighbor classifier performance in high dimensions using dimensionality reduction. First, we investigated the effect of different dimensionality reduction methods on nearest neighbor classification. Next, we investigated the outputs of dimensionality reduction to further improve the performance of the nearest neighbor classifier. Finally, we investigated how randomness in dimensionality reduction could be used to form nearest neighbor ensembles.

The main contributions of the study are two comparative studies on the use of dimensionality reduction methods in high dimensions and three studies on the fusion of the outputs of dimensionality reduction methods, which resulted in an improved classification accuracy. The final study is on nearest neighbor ensembles created using dimensionality reduction methods based on randomness. This chapter concludes the thesis by summarising our contributions and providing future research directions.

5.1 Conclusions

In this thesis, several dimensionality reduction methods are compared to find those that work best with nearest neighbor algorithms in high dimensions. The first research question investigated was the following: 'Q1: To what extent does dimensionality reduction improve the nearest neighbor performance?'. Initially, unsupervised dimensionality reduction methods (PCA and RP) were considered, followed by supervised dimensionality reduction methods (PLS and IG). Both images and microarrays were considered in this study. These studies build on previous research [36, 35, 33, 34], by focusing on high-dimensional datasets and taking efficiency and effectiveness into account. Furthermore, the studies show that the dimensionality reduction method used has a significant impact on the performance of the nearest neighbor classifier.

Feature fusion and classifier fusion strategies were considered to address the second research question: 'Q2: How can the dimension reduction outputs be used to improve the nearest neighbor accuracy further?'. Two techniques were considered: the fusion of an equal number of features from the different dimensionality reduction methods and the fusion of the optimal number of features required to achieve the best accuracy for the nearest neighbor classifier. The results indicate that fusion approaches yield better classification accuracies than individual reduction methods. They further reveal that the feature fusion method not only provides the best performance but is also robust to changes in the number of dimensions when equal numbers of dimensions for fusion are considered. When considering the optimal number of features for fusion, it is observed that classifier fusion yields a better performance than feature fusion.

The effects of selecting dimensionality reduction methods were explored in the fifth study. The comparative study was based on feature and classifier fusion methods that select the dimensionality reduction methods for fusion. We have considered the outputs of four dimensionality reduction methods, namely PCA, PLS, IG and ReliefF, for 18 chemoinformatics datasets. The

results show that classifier fusion improves the accuracy irrespective of the selection of the dimensionality reduction methods. In contrast, feature fusion was quite sensitive to the selection of dimensionality reduction methods.

To address the final research question 'Q3: How can the nearest neighbor ensemble be used to improve the classification accuracy in high dimensions?', a comparative study based on the use of randomness to create nearest neighbor ensembles for different types of high-dimensional datasets was considered in the final study. Two dimensionality reduction methods, namely random subsets and random projection, were investigated to form nearest neighbor ensembles on 34 microarrays, chemoinformatics and image data sets. The results show that the best method for forming nearest neighbor ensembles depends on the type of data. It is further observed that the nearest neighbor ensemble may even outperform state-of-the-art techniques such as random forests for certain types of data.

The field of machine learning, particularly learning from data sets such as microarrays, chemoinformatics, and images, raises several ethical considerations. While this thesis focuses on improving the efficiency and effectiveness of nearest neighbor classifiers, it is critical to carefully handle sensitive datasets, especially those related to medical information. It should be noted that the microarray gene expression and image data used in this research do not contain sensitive patient information. Furthermore, all datasets employed in this thesis are publicly available. Therefore, an extensive ethical evaluation may not be required. However, medical data always demand rigorous ethical attention, even though we handled the datasets carefully to protect privacy.

The improved nearest neighbor classifier accuracies presented in this research can potentially improve the predictive accuracy on high-dimensional data sets, such as microarray, chemoinformatics and image datasets. This could lead to more informed medical decisions, improving patient care, providing cost-effective treatments and addressing important societal con-

sequences.

Although the datasets employed in the thesis are from the natural sciences, the methodology and strategies explored in the thesis can also be utilised in the social sciences. They can help us to understand human behaviour and identify new societal trends.

Over the past decade, the field of machine learning research has changed considerably. This thesis focuses on a classical machine learning technique, the nearest neighbor algorithm, which is designed to handle high-dimensional data. However, deep learning algorithms such as convolutional neural network (CNN) are now considered for high-dimensional data such as images.

In the past, hyperparameter tuning, such as finding the optimal value for k in kNN, was done manually. However, automatic parameter tuning methods like grid search, random search and hyperband are now common. The selected parameters can lead to a better model performance as this may find optimal configurations.

We used data mining tools such as WEKA in our initial experiments, and they were run on personal or server computers at the university. With the growing complexity of models, software platforms such as the Google Colab, Microsoft Azure Machine Learning and Google Cloud AI platforms have emerged for experimentation. Due to popularity of the Python programming language in machine learning, we now have several core libraries such as scikit-learn, TensorFlow and PyTorch. In the last study, we used some of these Python libraries.

Although performing experiments was more manageable due to these technological changes, our results in this thesis are still valid. The results could be compared with the latest findings to understand the latest developments in machine learning.

5.2 Future Directions

Our work is limited to investigating the combination of five dimensionality reduction methods (PCA, RP, IG, PLS and ReliefF) with nearest neighbor classifiers in high dimensions. The nearest neighbor performance may be further improved by considering possible dimensionality reduction methods such as UMAP [73] individually or in combination with other reduction methods.

The empirical studies on combining the outputs of dimensionality reduction indicate that they can yield an improved performance. However, we have considered combining equal numbers of reduced features and the optimal number of reduced features to reach the best performance for each dimensionality reduction method. These results suggest that further exploration of other fusion strategies is needed to further enhance the performance of the nearest neighbor classifier.

The selection of the outputs of dimensionality reduction methods for combining features and classifiers built on reduced dimensions indicates that the selection of dimensionality reduction methods is quite sensitive to the performance of the nearest neighbor algorithm in high dimensions. However, we have explored the simple combination of the outputs of dimensionality reduction methods. Further exploration is needed to find other possible selection strategies.

We have treated all the features as equally important when handling the reduced features obtained from dimensionality reduction. However, feature weighting could be used to further improve the performance of the nearest neighbor classifer. More work in this direction could be beneficial for nearest neighbor classification in high dimensions.

When nearest neighbor ensembles were formed, the individual classifier accuracy after dimensionality reduction using random subsets and random projections was low compared to other dimensionality reduction methods,

such as principal component analysis. Since the ensemble performance is based on both the accuracy and diversity of component classifiers, other possible dimensionality reduction methods could be explored.

Furthermore, one can apply dimensionality reduction before applying a random subset or random projection method. This may further improve the accuracy of nearest neighbor ensembles. One could also investigate forming ensembles based on dimensionality reduction and other learning algorithms.

References

[1] T. M. Mitchell. The Discipline of Machine Learning. Technical Report CMU-ML-06-108, Carnegie Mellon University, July 2006.

[2] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[3] P. Langley and H. A. Simon. Applications of machine learning and rule induction. *Communications of the ACM*, 38(11):54–64, 1995.

[4] N. Lavrac, H. Motoda, T. Fawcett, R. Holte, P. Langley, and P. W. Adriaans. Introduction: Lessons learned from data mining applications and collaborative problem solving. *Machine Learning*, 57(1-2):13–34, 2004.

[5] J. Quackenbush. Microarray analysis and tumor classification. *The New England Journal of Medicine*, 354(23):2463–2472, 2006.

[6] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *Proc. Natl. Acad. Sci. USA*, volume 96, pages 6745–6750, 1999. URL:`http://www.pubmedcentral.nih.gov/articlerender.fcgi?tool=pubmed&pubmedid=10359783`.

[7] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class

discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999. URL:http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43.

[8] J. Kahn, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, and P.S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7:673–679, 2001.

[9] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson Jr, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[10] S. L. Pomeroy, P. Tamayo, M. Gassenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415:436–442, January 2002. URL:http://www-genome.wi.mit.edu/mpr/CNS/.

[11] Y. Pawitan, J. Bjöhle, L. Amler, A. L. Borg, S. Egyhazi, P. Hall, X. Han, L. Holmberg, F. Huang, S. Klaar, E. T. Liu, L. Miller, H. Nordgren, A. Ploner, K. Sandelin, P. M. Shaw, J. Smeds, L. Skoog, S. Wedrén, and J. Bergh. Gene expression profiling spares early breast cancer patients from adjuvant therapy: derived and validated in two population-based cohorts. *Breast Cancer Research*, 7, 2005.

[12] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, January 2002.

[13] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker. Analysis of dimensionality reduction techniques on big data. *IEEE Access*, 8:54776–54788, 2020.

[14] D. W. Aha, D. Kiblear, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[15] M. Mayo. Top data science and machine learning methods used in 2018, 2019. Available at `https://www.kdnuggets.com/2019/04/top-data-science-machine-learning-methods-2018-2019.html` (Accessed: 7 November 2021).

[16] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference of Machine Learning*, pages 161–168, 2006.

[17] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[18] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[19] M. À. Carreira-Perpiñàn. A review of dimension reduction techniques. Technical report, Dept. of Computer Science, University of Sheffield, 1997.

[20] H. Boström, S. F. Andler, M. Brohede, R. Johansson, A. Karlsson, J. van Laere, L. Niklasson, M. Nilsson, A. Persson, and T. Ziemke. On the definition of information fusion as a field of research. Technical report, Skövde : Institutionen för kommunikation och information, 2007.

[21] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2005.

[22] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.

[23] E. P. Xing, M. I. Jordan, and R. M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608. Morgan Kaufmann, 2001.

[24] R. Díaz-Uriarte and S. A. de Andrés. Gene selection and classification of microarray data using random forest. *Bioinformatics*, 7(3), 2006. URL: http://ligarto.org/rdiaz/Papers/rfVS/randomForestVarSel.html.

[25] S. D. Bay. Nearest neighbor classification from multiple feature subsets. *Intelligent Data Analysis*, 3:191–209, 1999.

[26] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20:2429–2437, 2004.

[27] S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.

[28] P. Langley and S. Sage. Scaling to Domains with Irrelevant Features. In R. Greiner, editor, *Computational Learning Theory and Natural Learning Systems: Volume IV: Making Learning Systems Practical*, pages 51–63. MIT Press, Cambridge, MA, USA, 1997.

[29] L. Breiman. Bagging predictors. *Machine Learning*, 26:123–140, 1996.

[30] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

[31] L. Wolf and S. Bileschi. Combining variable selection with dimensionality reduction. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 801–806, 2005.

[32] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2):273–324, 1997.

[33] E. Bingham and H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250, 2001.

[34] D. Fradkin and D. Madigan. Experiments with random projections for machine learning. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 517–522, 2003.

[35] X. Z. Fern and C. E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the Twentieth International Conference of Machine Learning*, pages 186–193, 2003.

[36] S. Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proceedings of International Joint Conference on Neural Networks*, volume 1, pages 413–418, Piscataway, NJ, 1998. IEEE Service Center.

[37] A. L. Boulesteix. PLS Dimension Reduction for Classification with Microarray Data. *Statistical Applications in Genetics and Molecular Biology*, 3, 2004.

[38] S. de Jong. SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263, 1993.

[39] K. Kira and L.A. Rendell. A practical approach to feature selection. In *Proceedings of International Conference on Machine Learning (ICML1992)*, pages 249–256, 1992.

[40] I. Kononenko. Estimating attributes: Analysis and extension of relief. In *Proceedings of European Conference on Machine Learning (ICML1994)*, pages 171–182, 1994.

[41] C. Ambroise and G. J. Mclachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences of the United States of America*, 99(10):6562–6566, May 2002.

[42] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

[43] D. V. Nguyen and D. M. Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18(1):39–50, 2002.

[44] D. V. Nguyen and D. M. Rocke. Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics*, 18(9):1216–1226, 2002.

[45] J. J. Dai, L. Lieu, and D. Rocke. Dimension Reduction for Classification with Gene Expression Microarray Data. *Statistical Applications in Genetics and Molecular Biology*, 5(1), 2006.

[46] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2011.

[47] C. Domeniconi and B. Yan. Nearest Neighbor Ensemble. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 228–231. IEEE Computer Society, 2004.

[48] S. Bubeck and U. von Luxburg. Nearest neighbor clustering: A baseline method for consistent clustering with arbitrary objective functions. *Journal of Machine Learning Research*, 10:657–698, 2009.

[49] Y. Wang, X. Xu, H. Zhao, and Z. Hua. Semi-supervised learning based on nearest neighbor rule and cut edges. *Knowledge-Based Systems*, 23:547–554, 2010.

[50] N. Rajani, B. Krause, W. Yin, T. Niu, R. Socher, and C. Xiong. Explaining and improving model behavior with k nearest neighbor representations. *ArXiv*, abs/2010.09030, 2020.

[51] C. J. Hazard, C. Fusting, M. Resnick, M. Auerbach, M. Meehan, and V. Korobov. Natively interpretable machine learning and artificial intelligence: Preliminary results and future directions. *Computing Research Repository*, abs/1901.00246, 2019.

[52] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[53] S. Deegalla and H. Boström. Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification. In *Proceedings of the 5th International Conference on Machine Learning and Applications*, pages 245–250, Washington, DC, USA, 2006. IEEE Computer Society.

[54] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 140–144, New Orleans, 1994. AAAI Press.

[55] L. Yu and H. Liu. Efficiently handling feature redundancy in high-dimensional data. In *Proceedings of the Ninth ACM SIGKDD Inter-*

national Conference on Knowledge Discovery and Data Mining, pages 685–690, 2003.

[56] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374, 2016.

[57] J. Shlens. A Tutorial on Principal Component Analysis. Available at `http://arxiv.org/abs/1404.1100` (Accessed: 7 November 2021), 2014.

[58] E. Elhaik. Principal Component Analyses (PCA)-based findings in population genetic studies are highly biased and must be reevaluated. *Scientific Reports*, 12:14683, 2022.

[59] W. Melssen and R. Wehrens. Chemometrics I Study Guide. Available at `http://www.webchem.science.ru.nl/ChemI/Pdf/ChemI.pdf` (Accessed: 7 November 2021).

[60] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 1984.

[61] D. Achlioptas. Database-friendly random projections. In *ACM Symposium on the Principles of Database Systems*, pages 274–281, 2001.

[62] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss lemma. Technical Report TR-99-006, International Computer Science Institute, Berkeley, California, USA, 1999.

[63] H. Abdi and L. J. Williams. Partial least squares methods: partial least squares correlation and partial least square regression. *Methods in Molecular Biology*, 930:549–579, 2013.

[64] G. Forman. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

[65] U. M. Fayyad and K. B. Irani. On the Handling of Continuous-Valued Attributes in Decision Tree Generation. *Machine Learning*, 8:87–102, 1992.

[66] T. K. Ho. Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, pages 278–282, 1995.

[67] L. I. Kuncheva, J. J. Rodriguez, C. O. Plumption, D. E. Linden, and S. J. Johnston. Random Subspace Ensembles for fMRI Classification. *IEEE Transactions on Medical Imagine*, 29:531–541, 2010.

[68] A. Bertoni, R. Folgieri, and G. Valentini. Feature selection combined with random subspace ensemble for gene expression based diagnosis of malignancies. In B.Apolloni, M.Marinaro, and R. Tagliaferri, editors, *Biological and Artificial Intelligence Environments*, pages 29–36. Springer, 2005.

[69] D. Ghosh. Singular value decomposition regression modeling for classification of tumors from microarray experiments. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 11462–11467, 2002.

[70] M. Skurichina and R. P. W. Duin. Combining Feature Subsets in Feature Selection. In N.C. Oza et al., editor, *Multiple Classifier Systems*, pages 165–175. Springer-Verlag, Berlin Heidelberg, 2005.

[71] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. Van de Rijn, M. Waltham, A. Pergamenschikov, J. C.F. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. N. Weinstein, D. Botstein, and P. O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, 24(3):227–235, 2000.

[72] S. Mylavarapu and A. Kaban. Random projections versus random feature selection for classification of high dimensional data. In *In Proceedings of the the UK Workshop on Computational Intelligence (UKCI 2013)*, pages 305–312, 2013.

[73] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction., 2018. URL: http://arxiv.org/abs/1802.03426.