The Role of Classifiers in Feature Selection: Number vs Nature

A Thesis submitted for the degree of Doctor of Philosophy by

Kyriacos Andrews Chrysostomou

School of Information Systems, Computing and Mathematics

Brunel University

October 2008

Abstract

Wrapper feature selection approaches are widely used to select a small subset of relevant features from a dataset. However, Wrappers suffer from the fact that they only use a single classifier when selecting the features. The problem of using a single classifier is that each classifier is of a different nature and will have its own biases. This means that each classifier will select different feature subsets. To address this problem, this thesis aims to investigate the effects of using different classifiers for Wrapper feature selection. More specifically, it aims to investigate the effects of using different number of classifiers and classifiers of different nature.

This aim is achieved by proposing a new data mining method called Wrapper-based Decision Trees (WDT). The WDT method has the ability to combine multiple classifiers from four different families, including Bayesian Network, Decision Tree, Nearest Neighbour and Support Vector Machine, to select relevant features and visualise the relationships among the selected features using decision trees. Specifically, the WDT method is applied to investigate three research questions of this thesis: (1) the effects of number of classifiers on feature selection results; (2) the effects of nature of classifiers on feature selection results; and (3) which of the two (i.e., number or nature of classifiers) has more of an effect on feature selection results. Two types of user preference datasets derived from Human-Computer Interaction (HCI) are used with WDT to assist in answering these three research questions.

The results from the investigation revealed that the number of classifiers and nature of classifiers greatly affect feature selection results. In terms of number of classifiers, the results showed that few classifiers selected many relevant features whereas many classifiers selected few relevant features. In addition, it was found that using three classifiers resulted in highly accurate feature subsets. In terms of nature of classifiers, it was showed that Decision Tree, Bayesian Network and Nearest Neighbour classifiers caused signficant differences in both the number of features selected and the accuracy levels of the features. A comparison of results regarding number of classifiers and nature of classifiers revealed that the former has more of an effect on feature selection than the latter.

The thesis makes contributions to three communities: data mining, feature selection, and HCI. For the data mining community, this thesis proposes a new method called WDT which integrates the use of multiple classifiers for feature selection and decision trees to effectively select and visualise the most relevant features within a dataset. For the feature selection community, the results of this thesis have showed that the number of classifiers and nature of classifiers can truly affect the feature selection process. The results and suggestions based on the results can provide useful insight about classifiers when performing feature selection. For the HCI community, this thesis has showed the usefulness of feature selection for identifying a small number of highly relevant features for determining the preferences of different users.

Publications

The following publications have resulted from various works conducted by the author, some of which are presented in this thesis.

Journal Papers

1. Chrysostomou, K.A., Chen, S.Y., and Liu, X., (2009). Combining Multiple Classifiers for Wrapper Feature Selection. *International Journal of Data Mining, Modelling and Management (IJDMMM),* 1, 1, 91-102.
2. Chrysostomou, K.A., Chen, S.Y., and Liu, X., (in press). Investigation of Users' Preferences in Interactive Multimedia Learning Systems: A Data Mining Approach. *Interactive Learning Environments.*
3. Lee, M., Chen, S.Y., Chrysostomou, K.A., and Liu, X., (2009). Mining Students' Behavior in Web-Based Learning Programs. *Expert Systems with Applications,* 36, 2, 3459-3464.

Conference Papers

4. Chrysostomou, K.A., Chen, S.Y., and Liu, X., (in press). The Effects of Multiple Classifiers on Feature Selection, *In Proceedings of the 14th International Conference on Automation & Computing, Brunel University, UK.*
5. Chrysostomou, K.A., Chen, S.Y., and Liu, X., (in press). The Influences of Number and Nature of Classifiers on Consensus Feature Selection, *In Proceedings of the 2008 International Conference on Data Mining, Las Vegas, USA.*
6. Chrysostomou, K.A., Frias-Martinez, E., Chen, S.Y., and Liu, X., (2006). Mining Users' Preferences of Multimedia Interfaces with K-modes, *In Proceedings of International Conference of IEEE Systems, Man, and Cybernetics 2006*, *Taiwan*, 4, pp 2849-2854.
7. Chrysostomou, K.A., Chen, S.Y., and Liu, X., (2006). Mining Users' Preferences in an Interactive Multimedia Learning System: A Human Factor Perspective, *In Proceedings of Conference of Human-Computer Interaction (HCI) 2006, London, UK,* pp 118-122.

Book Chapters

8. Chrysostomou, K.A., Lee, M., Chen, S.Y., and Liu, X., (2008). Wrapper Feature Selection. *Encyclopedia of Data Warehousing and Mining,* 4, 2103-2108.
9. Lee, M., Chrysostomou, K.A., Chen, S.Y., and Liu, X., (2008). Applications of Decision Trees for Data Modelling. *Encyclopedia of Artificial Intelligence,* 1, 437-442.

Acknowledgements

First, I would like to express my gratitude to my supervisors, Dr. Sherry Y. Chen and Professor Xiaohui Liu, who have provided me with endless support, guidance and advice throughout my PhD and thesis write up process. Their support and instruction has not only helped me become a better researcher but has also helped me become a better person. They have continuously inspired me as a person and my way of thinking. Thank you.

I would also like to thank the Department of Information Systems, Computing and Mathematics for providing me with a warm and comfortable environment for completing this thesis. Thanks also go to fellow members of the Centre of Intelligent Data Analysis group (my academic family!) for their continuous moral support and invaluable discussions.

Special thanks must go to my family including my father, my mother and both of my sisters, who have always been there for me and provided unconditional love and support throughout the highs and lows of my life. I truly love you all.

Finally, I would like to say that this experience will stay with me forever. Although this experience has been tough, it has been extremely worthwhile and rewarding. Once again, I would like to thank all the above persons for being there for me throughout this experience in my life.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

1.1 Thesis Context

This thesis presents interdisciplinary work which integrates several areas of research, namely data mining, feature selection and Human-Computer Interaction (HCI). This section briefly introduces each of these research areas.

Data mining encompasses techniques from a number of fields, including information technology, statistical analyses, and mathematical science (Bohen et al., 2003). Data mining techniques can help analyse, understand and visualise large amounts of data stored in databases, data warehouses or other data repositories (Li and Shue, 2004). This means that data mining techniques have the ability to handle large datasets, which consist of hundreds or thousands of features. The sheer number of features present in such datasets often causes problems for data miners because some of the features may be irrelevant to the data mining techniques used. Such irrelevant features can harm the quality of the results obtained from data mining techniques (Bhavani, Rani and Bapi, 2008) and in turn reduce the chances of identifying useful knowledge from the dataset. A way of dealing with irrelevant features is to use *feature selection*.

Feature selection is widely used for selecting the most relevant subset of features from datasets according to some predefined criterion (Sima and Dougherty, 2008). Feature selection thus focuses only on the relevant features in the dataset by removing any irrelevant features. There are many benefits associated with removing irrelevant features, some of which include reducing the amount of data (i.e., features) so that the data are easier to handle when performing data mining and being able to reveal the relevancies within the data (Czekaj, Wu and Walczak, 2008). These attractive benefits have led researchers to use feature selection for many different types of tasks, one of which is identifying relevant features from datasets belonging to HCI. This is mainly because of the inherent fuzziness of HCI datasets typically caused by users being unsure of their preferences (Frias-Martinez, 2007).

The area of HCI looks at the way in which different users interact with various Web-based systems (Frias-Martinez, et al., 2007). Popular examples of this involve

looking at the interactions and preferences of different users with regards to Web-based learning systems and search engines. By examining users' preferences of such systems, we can gain a better understanding of users' needs. In the context of HCI, such understanding can help us develop Web-based systems that can accommodate users' preferences and needs.

The rest of this chapter gives an overview of the areas under investigation in this thesis. First, Section 1.2 defines the problem to be investigated. Subsequently, Section 1.3 outlines the aim and research questions of the thesis. Section 1.4 then presents a description of the contributions of this thesis and finally Section 1.5 details the structure of the thesis.

1.2 Motivation of Thesis

As previously mentioned, feature selection is a useful data mining tool for selecting sets of relevant features from datasets. Currently, feature selection is typically performed by two types of feature selection methods, *Filters* and *Wrappers* (Kohavi, 1995b; Kohavi and John, 1997). A *Filter* method evaluates the relevance of features according to some discriminating criterion that looks at the general characteristics of the data (Bhavani, Rani and Bapi, 2008). The results from such a method are usually a ranked list of features, where the features at the top of the list are relevant and the features at the bottom of the list are not so relevant or totally irrelevant. A *Wrapper*, however, evaluates the relevance of features by using a classifier and selects only the most relevant subset of features (Ng, et al., 2008). Therefore, the results obtained from a Wrapper are different to that of a Filter because it actually selects a subset of the most relevant features rather than list all features in order of relevance (Huang, Yang, and Chuang, 2008).

Many researchers have used both Filters and Wrappers for the purpose of feature selection. Interestingly, it has been shown that Wrappers often give superior performance (in terms of classification accuracy) than Filters (e.g., Inza, et al, 2004; Ruiz, et al, 2006; Zheng and Zhang, 2008). Although Wrappers provide better performance, they tend to use only one classifier to select the relevant features. The problem with using a single classifier is that each classifier is of a different nature and will have its own biases. Classifiers that possess different nature and biases may

have a different effect on feature selection. For example, classifiers with one type of bias may be more (or less) suited to selecting relevant features from a dataset than classifiers with another type of bias. This may be due to the fact that the biases made by one of the classifiers match (or do not match) the underlying biases and characteristics of the dataset used. This may subsequently lead to different feature subsets being selected by the classifiers. In fact, each classifier may select a different feature subset which may contain diverse numbers of features and lead to varying levels of classification accuracy.

This problem associated with using a single classifier motivates the use of multiple classifiers for feature selection. However, little is known about the effects of using multiple classifiers for feature selection, especially the effects of using different numbers of classifiers and using classifiers with a different nature. On the one hand, different numbers of classifiers may affect feature selection results. The reason for this may lie within the level of agreement among the classifiers. If a low number of classifiers are used for feature selection, then it is likely that the level of agreement among them will be high. High agreement among classifiers may subsequently result in more relevant features being selected and differences in accuracy levels. If, however, a high number of classifiers are used for feature selection then the level of agreement will probably be low because more classifiers are required to agree on the relevance of a feature. This in turn may lead to fewer relevant features being selected and different levels of accuracy. This ultimately shows that varying the number of classifiers may influence the number of features selected and the accuracy levels of the features.

On the other hand, classifiers of a different nature may also lead to different feature selection results. This is because a classifier with a different nature will possess different biases. For example, consider the biases of classifiers belonging to two widely used classifier families, namely Bayesian Network and Nearest Neighbour. Classifiers of the Bayesian Network family typically aim to find features that have high conditional probability values when building a graphical network structure (Gammerman, 1997). However, classifiers of the Nearest Neighbour family aim to find features that are deemed the closest by a predefined distance metric. This example shows that classifiers which have different biases are highly likely to select

different features. In this way, they may select different number of features and features that lead to different levels of classification accuracy.

In summary, the number of classifiers and nature of classifiers are two issues that may affect the way in which features are selected. The thesis will therefore set out to investigate the effects of these two issues on feature selection.

1.3 Research Questions of Thesis

The aim of the thesis is to investigate the effects of using multiple classifiers on feature selection, namely the number of classifiers and nature of classifiers. More specifically, the following three research questions will be investigated in order to achieve the aim of this thesis. They are as follows:

1. To what extent does the *number* of classifiers used influence the number of features selected and the accuracy levels of the features (RQ1);

2. To what extent does the *nature* of classifiers used influence the number of features selected and the accuracy levels of the features (RQ2); and

3. Which of the two issues (i.e., number of classifiers or nature of classifiers) has a greater effect on feature selection (RQ3).

In order to help answer the three research questions mentioned above, the thesis proposes a novel data mining method called Wrapper-based Decision Trees (WDT). The WDT method combines multiple classifiers with Wrappers for feature selection and also visualises the feature selected by using decision tree classifiers. The WDT method thus provides two benefits. First, it can overcome the problem of using a single classifier with existing Wrappers since it uses several classifiers for feature selection. This means that it can reduce the biases associated with using individual classifiers so that sets of mutually agreed and unbiased features are selected. Second, WDT can be used with various numbers of classifiers and classifiers of a different nature to perform the feature selection. This means it can help us better understand how the number and nature of classifiers influence feature selection results.

To assist in uncovering the effects of number and nature of classifiers, the proposed WDT method is used in conjunction with: a) two different classifier arrangements approaches and b) two different types of user preference datasets. With regards to the former, classifiers are combined in two different approaches, namely same-type and mixed-type approaches. The same-type approach involves using different numbers of classifiers that have a similar nature whereas the mixed-type approach involves combining different numbers of classifiers that have a different nature. By using these two approaches, we will be able to provide a complete picture of how the number and nature of classifiers influence feature selection results.

With regards to the latter, the two chosen datasets, which are derived from the area of HCI, consist of users' preferences of 1) search engines and 2) Web-based learning systems. Such datasets were chosen because they are typically of a fuzzy nature (Tai and Chen, 2006; Castellano, et al., 2007) and may possess irrelevant features that need to be removed by feature selection. Removing irrelevant features from HCI datasets through feature selection offers great benefits to experts in the field. In fact, HCI experts will be equipped with a new type of tool which can not only assist them in reducing the fuzzy nature of these datasets but can subsequently help them better understand the preferences of different users. The use of such a tool therefore presents a new approach to analysing HCI datasets.

1.4 Contributions of Thesis

The investigation presented in this thesis makes contributions to three different communities, including the communities of Data Mining, Feature Selection, and HCI. These contributions are described below.

- With regards to the *Data Mining* community, this thesis proposes a novel method called Wrapper-based Decision Trees (WDT), which integrates the use of multiple classifiers to perform feature selection and decision trees to effectively visualise the most relevant features within a dataset. This method has been shown in the thesis to select accurate sets of relevant features and help gain insight into the relevancies present within the datasets used.

- With regards to the *Feature Selection* community, this thesis employs the WDT method to investigate the role of the number of classifiers and the nature of classifiers in feature selection. The results showed that different numbers of classifiers and classifiers of a certain nature play a significant role in influencing the feature selection results. Suggestions based on these results are also given, which may assist feature selection experts in choosing classifiers suitable to particular feature selection tasks, i.e., choosing classifiers that lead to compact feature subsets or classifiers that lead to high levels of classification accuracy.

- With regards to the *HCI* community, the WDT method was able to select highly relevant feature subsets from datasets consisting of users' preferences of search engines and Web-based learning systems. These highly relevant feature subsets can prove very useful in the context of HCI because they may contain features that help differentiate the search engine preferences and Web-based learning preferences of different users. Such features can be used by HCI experts to develop search engines and Web-based learning systems that better accommodate the preferences of different users. Considering the needs of users in such a novel way presents a new milestone in HCI research.

1.5 Thesis Structure

Following on from this chapter, Chapter 2 provides a detailed overview of the state-of-the-art of feature selection and reviews current works on the two main feature selection methods: Filters and Wrappers. The benefits and limitations of each method are also examined within this chapter.

Chapter 3 presents a detailed description of a novel method called Wrapper-based Decision Trees, or WDT, which combines multiple classifiers to select relevant features and visualises the relationships among selected features through use of decision trees. In this thesis, Bayesian Network, Decision Tree, Nearest Neighbour and Support Vector Machine are classifiers used with WDT to do the feature selection. Since the WDT method combines several different classifiers for feature selection, it is used in this thesis to investigate the role of the number and nature of classifiers which will help answer the research questions presented in the previous

section. More specifically, the WDT method makes use of two types of user preference datasets, where one includes user preferences of search engines (UP1) and the other includes user preferences of Web-based learning systems (UP2). In addition, WDT uses the same-type and mixed-type classifier arrangement approaches, which combine classifiers in a different manner. Details of both datasets and classifier arrangement approaches are also provided in this chapter.

Chapter 4 presents the results obtained using the WDT method with both datasets and the first of the two classifier arrangement approaches: the same-type approach. The results show that the number of classifiers plays a significant role in feature selection since few classifiers select higher number of relevant features and many classifiers select lower number of features. Furthermore, it is found that using Decision Tree classifiers leads to higher number of features and higher levels of accuracy than other types of classifiers. The chapter also examines the combinations of classifiers which build decision trees with the highest accuracy levels for both UP1 and UP2 datasets. Small subsets of the most relevant features are extracted from the decision trees which help determine the preferences of different users.

Chapter 5 presents the results using WDT with the mixed-type approach. Once again, the results show that reducing the number of classifiers increases the number of features selected and increasing the classifiers results in a decrease in relevant features. Interestingly, classifiers belonging to the Bayesian Network and Nearest Neighbour families influence the number of features selected and the accuracy levels generated for UP1 and UP2. Finally, a close examination of the decision trees with the highest accuracy levels in both datasets is carried out. Additional relevant features are obtained from these decision trees which in turn prove useful in distinguishing the preferences of the users from each dataset.

Chapter 6 compares the results obtained from the previous two chapters. This chapter therefore compares the results from the WDT method using the same-type and mixed-type approach. This is to provide a complete overview of the role of the number of classifiers and nature of classifiers in feature selection. In terms of number of classifiers, the comparison reveals two issues: (1) few classifiers select more relevant features and many classifiers select few features irrespective of nature of

classifiers used and (2) combinations comprising three classifiers select feature subsets that lead to the highest levels of accuracy irrespective of the nature of classifiers used. These two issues help answer the first research question of the thesis (RQ1). In terms of the nature of classifiers, the comparison reveals that the nature of classifiers is stronger in different contexts. On the one hand, Decision Tree classifiers influence feature selection results only when combined and used together. On the other hand, Bayesian Network and Nearest Neighbour classifiers influence feature selection only when combined with classifiers from other families. Such results provide answers to the second research question of the thesis (RQ2). Based on the answers to the first and second research questions, it is shown that the number of classifiers has more of an influence on feature selection results than the nature of classifiers used, which answers the third research question of the thesis (RQ3). Suggestions based on the comparison in this chapter are also presented, which show suitable (and not so suitable) numbers of classifiers and nature of classifiers for feature selection.

Finally, Chapter 7 presents the key findings of the thesis with regards to the role of number and nature of classifiers in feature selection. The chapter then outlines some limitations of the thesis and finally describes some ideas for future work based on the limitations identified.

Chapter 2 – Feature Selection: The State-of-the-Art

2.1 Introduction

Data mining is the process of extracting valuable information from large amounts of data (Hand, Mannila and Smyth, 2001). One of the most widely used data mining approaches is classification. Classification requires each instance in a dataset to be assigned a label. The purpose of classification is to predict the class label of a new instance using a set of already labelled data instances (Szpunar-Huk, 2006). However, datasets with irrelevant features (i.e., features which do not contribute to the prediction of class labels) may cause some problems for classification. The classification performance can be deteriorated by such irrelevant features so there is a need to remove irrelevant and redundant features from datasets (Blum and Langley, 1997). A widely used way of removing such features is known as feature selection. Feature selection is used to limit the effects of irrelevant features by seeking only the relevant subset from the original features (de Souza, Matwin and Kapkowicz, 2006). By reducing the number of irrelevant features in this manner, the time taken to perform classification can be greatly reduced and the reduced dataset is easier to handle, which often leads to more accurate classification results (Guyon and Elisseeff, 2003; Yang and Olafsson, 2006).

In general, there are two main methods for performing feature selection. The first is the Filter method. Filters usually rely on the underlying characteristics of the dataset and a statistical criterion to rank features according to their relevance. Those which are ranked top will be most relevant and those ranked bottom will be of least relevance (Huang and Chow, 2007). The second method is Wrappers. Unlike Filters, Wrappers use classifiers to select relevant features. They use the performance of classifiers to decide which features are relevant (shown by high classifier performance) and which are not so relevant (shown by low classifier performance) (Huang, et al, 2007).

In summary, Filters and Wrappers are two of the most commonly used feature selection methods. As such, this chapter will provide a review of the state-of-the-art of both of these methods. In order to provide a foundation to both methods, the

chapter will start by giving a brief introduction to feature selection in Section 2.2. Subsequently, Section 2.3 will explain how feature selection is performed using Filters and also provides examples of some of the most commonly used Filter methods. The main limitations of Filters will also be outlined. Following this, a detailed explanation of Wrapper feature selection methods will be given in Section 2.4. Since Wrappers use classifiers to perform the feature selection, the chapter will then give brief details on four of the most popularly used classifier families, namely, Bayesian Networks (Section 2.4.1), Decision Trees (Section 2.4.2), Nearest Neighbour (Section 2.4.3) and Support Vector Machines (Section 2.4.4), and how they have been used with the Wrapper. The limitations of Wrappers will also be detailed in Section 2.4.5. The limitations will include the amount of time Wrappers take to perform feature selection, the accuracy of features selected by Wrappers, and the use of a single classifier to do the feature selection.

2.2 The Basics of Feature Selection

Typically, feature selection can be formally defined in the following manner. Suppose F is the given set of original features with cardinality n (where n symbolises the number of features in set F), and $\overline{F}$ is the selected feature subset with cardinality $\overline{n}$ (where $\overline{n}$ symbolises the number of features in set $\overline{F}$), then $\overline{F} \subseteq F$. Also, let $J(\overline{F})$ be the selection criterion for selecting feature set $\overline{F}$. We assume that a higher value of J indicates a better feature subset. Thus, the goal is to maximise $J(\ )$. The problem of feature selection is to find a subset of features $\overline{F} \subseteq F$ such that,

$$J(\overline{F}) = \max_{Z \subseteq F, |Z| = n} J(Z)$$

Deriving a feature subset that maximises $J(\ )$ typically consists of four key steps namely, search direction, search strategy, feature subset evaluation and stopping criterion (Huan and Lei, 2005). In brief, search direction defines the point at which the search for the most relevant subset will begin. Complementary to the direction of the search is the search strategy. The search strategy outlines the way in which feature subsets are searched within the feature space. Each of the feature subsets found is then evaluated according to some evaluation criteria. Finally, a stopping

criterion is chosen and used for halting the search through feature subsets. Further details on each of these four key steps are given in the next sections.

1) Search Direction

Intuitively, the first issue that needs to be considered when performing feature selection is the point at which to start searching for the most relevant subset of features. Deciding on the point to start the search also requires one to decide the direction of the search. For example, a search may begin at the point where no features are involved and then successively add more. In this case, the direction of the search is said to proceed forward through the search space. Conversely, the search can begin with all features and successively remove the less relevant ones. In this case, the search proceeds backward through the search space. Another way is to begin at a point somewhere in the middle moving outwards from this point. This is sometimes referred to as a bi-directional search because it can search forwards and backwards through all the features in the dataset.

2) Search Strategy

When the direction of the search has been decided, the search strategy (also known as the organisation of the search) must also be considered. Typically, search strategies can be classed as either *exhaustive* or *heuristic*. An exhaustive search systematically examines all possible subsets of features and selects the 'best' (i.e., optimal) subset of features relevant to the classification task. Such a search strategy guarantees to find the optimal feature subset. Some classic examples of exhaustive search techniques are branch-and-bound, beam search (which is a variant of branch-and-bound), depth-first and breadth-first (Chen, 2003; Dash and Liu, 2003; Tso and Gu, 2004). Although such approaches often give optimal solutions, the numbers of possible subsets that need to be examined are exponential therefore making this form of search impractical for large datasets.

On the other hand, a more feasible and practical approach is a *heuristic* search. This type of search is often simpler as it does not consider all possible feature subsets. Instead, such a search only considers feature subsets that are very close to being the 'best' subset of features. The two most popularly used heuristic search strategies include the forward selection search and the backward elimination search (Aha and

Bankert, 1996; Kohavi and John, 1997). Both of these strategies consider local changes to the feature subsets during the search for the most relevant subset, where a local change is simply the addition or deletion of a single feature from the subset. When additions to the feature subset are considered the search strategy is known as forward selection. However, when deletions from the feature subset are considered the search strategy is referred to as backward elimination.

3) Feature Subset Evaluation

Irrespective of the search direction and strategy chosen, each selected feature subset needs to be evaluated according to some criteria. This is so that the feature subset with the highest accuracy can be identified. Typically, there are two main criteria for evaluating feature subsets (John, Kohavi and Pfleger, 1994). The first one functions independently of the classifier relying on statistical metrics and the general characteristics of the data to evaluate the relevance of feature subsets. In this case, the result is usually a ranked list of the features according to their relevance to the classification task. This criterion is widely known as the Filter. The second criterion, referred to as the Wrapper, is very different to the Filter. This is because the Wrapper requires a classifier to be used for feature selection. The Wrapper conducts the search for the most relevant feature subset using the classifier itself as part of the evaluation function. In other words, the Wrapper uses the performance of the classifier to determine how relevant the feature subsets are. Features which lead to high classifier performance will make up the final feature subset.

4) Stopping Criterion

Finally, some criteria must be chosen for stopping the search through feature subsets. When dealing with Filters, a commonly used stopping criterion is the ordering of the features according to some relevancy score (usually a statistical measure). Once ordered, the features with the highest relevance scores are chosen for use with a classifier. When using Wrappers, one might stop adding or removing features when there is no improvement to the accuracy of the current feature subset. Alternatively, the Wrapper might continue to adjust the feature subset (i.e., add or remove features from the subset) as long as the accuracy does not degrade past a certain value.

The aforementioned sections have outlined the four basic steps required to perform feature selection. Generally, there are two main methods that use these elements to perform feature selection: Filters and Wrappers. The difference between these two methods mainly lies within the way in which they evaluate the relevance of feature subsets. The former evaluates relevance of features without a classifier whereas the latter uses a classifier to evaluate the value of features (Hanczar, et al., 2003). A detailed explanation of both Filters and Wrappers is given in the next sections.

2.3 Filter Methods

Filter methods typically assess the relevance of features by looking at the intrinsic properties of the data and employing some statistical measure (Li, Xie and Goh, in press). Essentially, Filter methods begin by choosing a search strategy and deciding the direction of the search in order to start looking for the relevant features in the dataset. Then, each of the features in the dataset will be assigned a relevance score (either high or low), as calculated by a statistical measure (Liu and Yu, 2005). The features will then be ordered according to their relevance score. In some cases, however, features with high relevance scores will be selected and low scoring features will be discarded (Sayes, Inza, and Larrañaga, 2007). Finally, the selected features which have high relevance scores are presented as input to the classifier. This process which describes the way in which Filters perform feature selection is shown in Figure 2.1.

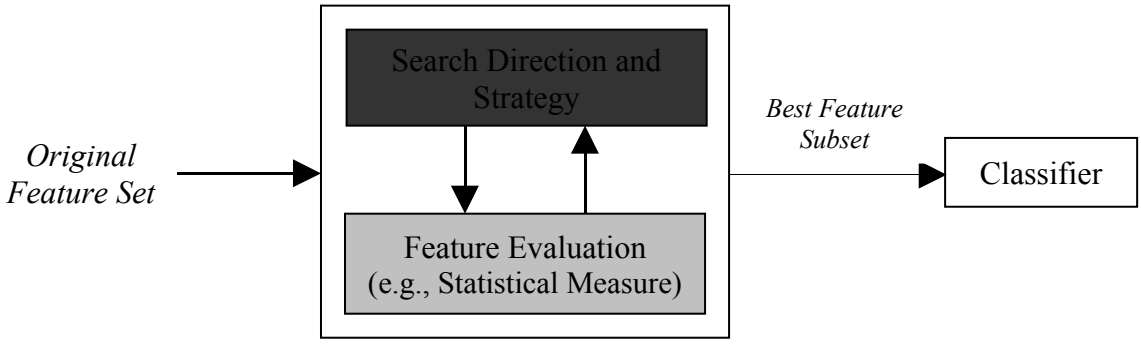

Figure 2.1. The Process of Filter Feature selection

In general, there are two types of Filter methods, namely univariate and multivariate (Zhu, Ong and Dash, 2007). The univariate filter methods consider each feature in the dataset separately when identifying relevant features whereas the multivariate

methods consider the interactions among different features in the dataset. Figure 2.2 shows some common examples of both univariate and multivariate filter methods. Detailed explanations of these examples and their applications for feature selection are given in the following sections.

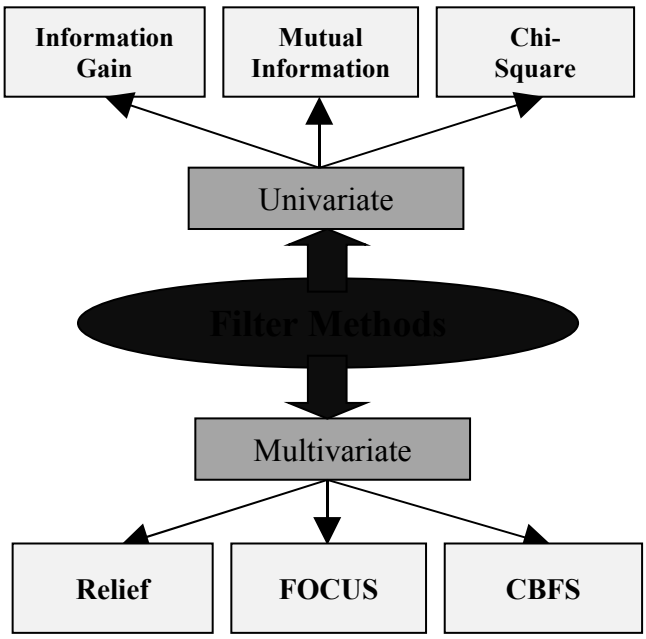

Figure 2.2. Some Examples of Filter Methods

2.3.1 Univariate Filter Methods

Univariate methods are probably the most popularly used Filter methods for performing feature selection (Sayes, Inza, and Larrañaga, 2007). Basically, univariate methods use statistical measures to determine the relevance of features in the dataset. More specifically, such Filter methods calculate the statistical significance of each individual feature (using the measure) with regards to the target variable. Features with high significance are considered to be highly relevant to the target variable whereas features with low significance are considered to be of low relevance. In addition, features which do not have any significance with the target variable are regarded as totally irrelevant.

Many examples of univariate methods carry out feature selection in the manner described above. Among them, methods that use information measures, such as Information Gain (Quinlan, 1993) and Mutual Information (Battiti, 1994), and other

statistical measures, such as χ^2 also known as Chi-Square (Chan and Wong, 1991) are most commonly used.

1) Methods with Information Measures

The information measures typically used with univariate Filters include Information Gain and Mutual Information. The Information Gain measure looks at the amount of information a feature (or more than one feature) contributes to predicting each class value of the target variable (Lee and Lee, 2006). To help determine the amount of information contributed by each feature, the entropy value is also calculated for each one with respect to the target variable (Yu and Liu, 2004). A feature with a high entropy value is considered to be relevant whereas a feature with a low entropy value is seen as less relevant. The Mutual Information measure, on the other hand, investigates the amount of uncertainty a feature (or more than one feature) possesses in relation to the target variable (Mladenic and Grobelnik, 2003). The level of uncertainty for each feature is determined by looking at dependencies between the features (Liu, et al., in press). The features which are highly dependent on other features signify a high level of uncertainty and are thus discarded. This leaves the features which are only dependent on the target variable. Typically, Filter methods, which adopt information measures such as these described, aim to select features which contribute the most information and discard those features which promote uncertainty (Li, Xie and Goh, in press). In this way, the features selected by such Filter methods will be highly relevant to the target variable. Examples of Filter methods that employ information measures namely Information Gain and Mutual Information are given in the next few pages.

In terms of Information Gain, Pazzani and Billsus (1997) incorporated this information measure into their Syskill & Webert system, which rates Web pages based on the preferences of users and makes recommendations to users in the form of what pages they may be interested in. The measure was used in their system to identify the most informative words (i.e., those that occur most often) in Web pages rated interesting by users. These words were then classified using Bayesian classifier. The result from this classifier is a recommendation to the user in the form a Web page. It was shown that the system was able to make accurate recommendations based on the words selected by Information Gain measure. In addition, Liu, Li and

Wong (2002) used the Information Gain measure for the purpose of identifying a set of the most important genes for diagnosing different forms of cancer. The measure was used with two types of cancer datasets, namely the acute leukaemia dataset and the ovarian cancer dataset, each of which consist of more than one thousand genes (i.e., features). The results showed that Information Gain was able to identify 13 important and relevant genes from the acute dataset and 20 genes from the ovarian cancer dataset. These features were then classified using an array of different classifiers, including the C4.5 decision tree classifier and Support Vector Machine. The classification accuracies generated by these classifiers using identified features were shown to be much higher than those generated using all of features in the datasets.

In terms of Mutual Information, Prabowo and Thelwall (2006) used this measure to detect significant topics in online news story documents and Web blogs. The significant topics within these two applications are those which consist of frequently occurring words. This measure was used to rank all words according to how many times they appear in the applications. Those which appear many times are assigned a high significance value while those which appear few times are assigned a low significance value. It was found that the Mutual Information measure helped identify a small number of the most significant words from Web blogs and news stories, which generated high classification accuracies. Furthermore, Blanco et al. (2005) used the Mutual Information measure for the purpose of bioinformatics. The authors used this measure to predict the survival rate of cirrhotic patients (i.e., patients with liver disease) after having the TIPS liver disease treatment. The measure was used to select a small number of relevant features from the patient dataset. The relevant features were then used with a Bayesian network classifier to make the predictions. The predictions made by the classifier using the identified features were highly accurate and proved useful to physicians for better understanding the effectiveness of TIPS treatment.

2) Methods with Statistical Measures

Statistical measures such as Chi-Square can also be used to do Filter feature selection. The Chi-Square method considers one feature at a time. The method then evaluates the relevance of the feature by computing the value of the chi-square

statistic with respect to the target variable. A feature which has a higher chi-square value is considered more relevant to the target variable, whereas a feature with a lower value is regarded as being less relevant. This is repeated for every feature in the dataset. In general, the Chi-Square method has been used to perform feature selection in many areas. Some areas in which this method has been widely used also include Web text mining and bioinformatics. In terms of the former, Fan, Gordon and Pathak (2005) looked at the automatic identification of users' interests and preferences based on a set of significant keywords from their past Web page history of news story documents. Initially, the Chi-Square measure was used to determine the significance of each keyword used by the users. All keywords were then ranked based on their significances, where the significant keywords were those with high statistical values and insignificant keywords had low values. The keywords that had high statistical values were used to determine the users' interests and preferences of news story documents. This approach used by the authors was shown to be more robust and accurate than the approach without the Chi-Square Filter method.

In terms of the latter, Sartore et al, (2008) recently employed the Chi-Square method to identify features that would help in predicting the likelihood of a tumour evolving into cancer. The Chi-Square was used with a dataset comprised of several features describing the characteristics of patients with cancerous and non cancerous tumours. From the large number of features present in the patient dataset, Chi-Square method was able to select three features that were responsible for predicting the chances of a tumour turning into cancer. These features were found to generate higher classification accuracy compared to the accuracy generated using all features in dataset.

2.3.2 Multivariate Filter Methods

The Filter methods described in the previous section consider each feature independently when searching for the set of most relevant features. This can be a problem if there are some features in the dataset which are more relevant when considered together. This problem can be reduced by using multivariate filter methods. These types of methods are able to determine the significance of features by considering the interactions between more than one feature. There are many examples of multivariate filter methods, the most common ones being Relief (Kira

and Rendell, 1992), FOCUS (Almuallim and Dietterich, 1991) and Correlation-based Feature Selection (Hall, 2000).

The aim of the Relief method is to assign a relevance weight (i.e., score) to each feature in the dataset. Each feature's weight reflects its ability to distinguish among the class values of the target variable. A feature will be assigned a high weight if it is able to differentiate between different class values but not differentiate among identical class values (Zheng and Zhang, 2008). Features are then ranked according to their weight and those that exceed a user-specified weight threshold (which is usually high) are selected to form the final feature subset. An example where the Relief method has been used to perform feature selection can be found in Ruan et al, (2006). In this example, the authors use the Relief method to analyse gene expression data. The Relief method was used to identify a set of the most informative genes that would help in diagnosing different types of cancerous tumours like lung and pancreas. The Relief method identified several sets of such genes for the different types of tumours, which were then presented as input to the Support Vector Machine classifier. The selected set of genes led to higher classification accuracies compared to that of all genes being used. More importantly, the selected genes were shown to provide useful insight into the identification of cancerous tumours.

Another multivariate filter method is FOCUS. The FOCUS method conducts an exhaustive search of all feature subsets until it finds the smallest set of features that consistently labels the instances within the entire dataset. This means that FOCUS looks for features that can successfully divide the instances in the dataset into the number of classes of the target variable. Due to the fact that an exhaustive search is carried out by FOCUS, it is more likely to find accurate feature subsets from data. Many researchers are aware of this fact and have used FOCUS to perform feature selection for different tasks. As an example, Wittmann and Ruhland (1999) use feature selection and the Neuro-fuzzy system classifier (which combines the benefits of neural networks and fuzzy set theory) to predict the likelihood of people replying to emails regarding financial promotions at banks. In terms of the feature selection, the FOCUS method was used to remove the irrelevant features and identify the relevant features in the dataset. This process managed to reduce the dataset from 28 features to 13 features, i.e., 13 relevant features were selected. In terms of the

classifier, the selected relevant features were fed as input to the Neuro-fuzzy classifier to generate rules that would help make the predictions. The rules were found to generate high levels of classification accuracy, which led the authors to conclude that feature selection techniques, such as FOCUS, are of high relevance to any type of data mining task since they can help improve classification results.

Correlation-based Feature Selection (CBFS) is the other commonly used multivariate filter method. Basically, the CBFS uses a slight variation of the Pearson's Correlation coefficient to measure the relevance of individual features with regards to the target variable and also the relevance of features in relation to other features in the dataset (Hall and Smith, 1997). Features that have high relevance with the target variable but low relevance with other features are chosen to form the final feature subset (Hall, 2000). In other words, CBFS is useful for identifying and discarding feature correlations which can often be redundant and irrelevant to the target variable. This usefulness of CBFS has led data miners to use it for a variety of tasks, such as gene expression data analysis. More specifically, Wang, et al (2005) adopted the CBFS method to identify relevant features from gene expression data, namely acute leukaemia data and B-cell lymphoma data, in order to find the genes that cause these diseases (i.e., leukaemia and B-cell lymphoma). From the many thousands of genes present in these two datasets, CBFS was able to select a very small subset of relevant features. As an example, consider the B-cell lymphoma data which had 4026 features. The CBFS method found 25 relevant features from this dataset. The CBFS was not only able to reduce the size of the dataset by selecting the relevant ones but was also able to produce high accuracy levels using the selected relevant features.

2.3.4 Advantages and Disadvantages

In general, Filter methods have been widely used for many different feature selection tasks. The main reason for their wide use lies within the amount of time they take to perform the feature selection. Filter methods have the advantage of identifying relevant features relatively fast (Torres, Saad and Moore, 2007). In fact, they are able to identify relevant features faster than other feature selection methods. This is because they determine the relevance of features independently of the classifier intended for use and as a result need to perform feature selection only once. This is beneficial especially if datasets consist of thousands of features, like gene data.

Although Filter methods can select relevant features faster than other feature selection methods, they also have some disadvantages.

First, features found to be relevant by Filters, especially univariate methods, may in fact be redundant features. As previously explained, univariate methods consider the significance of each feature independently from the rest of the features (Huang, Cai and Xu, 2007). In this way, there may be several features with the same significance level. Such features are classed as being redundant since they add no extra value to the feature selection task. This means that some of these features can be removed without affecting the accuracy of the feature subset. In addition, it means that Filter methods may select more features than what is really required, which may turn out to be useless to the feature selection (Koller and Sahami, 1996; Mak and Kung, 2008). The second disadvantage also relates to univariate filter methods. The univariate Filters do not take into account the effects of combinations of features. This can limit the number of relevant features found because some features, which may be of low relevance or irrelevant, may end up being highly relevant in the presence of other features. For example, a feature with low significance may become very useful when combined with another feature that has low significance. In contrast, features which have high significance on their own may actually turn out to be of low significance when considered with other features. Disregarding feature interactions such as these can affect the features selected and in turn affect the classification accuracies generated by the features. The third and final limitation applies to both univariate and multivariate filter methods. Both of these types of Filter methods suffer from the fact that they ignore the classifier when selecting the relevant features (Zhu, Ong and Dash, 2007). In this way, the features selected by Filters may not match the classifier intended for use. As a result, Filters may miss out features highly relevant to the classifier, which may subsequently lead to low classification accuracies.

A way of dealing with these three disadvantages is to use another type of feature selection method, namely the Wrapper method. The reason for using the Wrapper is two fold. First, the Wrapper method considers different combinations of features when searching for the most relevant set of features. This means that Wrappers consider the interactions among features in the dataset, rather than considering the features individually. This means that there is less chance of selecting redundant (or

irrelevant) features. This overcomes the first and second disadvantages of Filters previously mentioned. Second, the Wrapper uses classifiers to select the relevant features. In this way, features selected by Wrappers will match the classifier intended for use, which can help increase classification accuracies (Li and Guo, 2008). This can therefore help overcome the third and final disadvantage of Filters. The Wrapper method seems to have many advantages in comparison to Filter methods. A deeper explanation of the Wrapper methods and their benefits is given in the next section.

2.4 Wrapper Method

The Wrapper uses a classifier as the evaluation criterion for maximising *J*( ). Basically, the Wrapper uses the classifier as a black box. The classifier is repeatedly run on the dataset using various subsets of the original features. These feature subsets are found through the use of a search strategy. The classifier's performance and some accuracy estimation method, like cross validation, are then used to evaluate the accuracy of each subset (John, Kohavi and Pfleger, 1994). The feature subset with the highest accuracy is chosen as the final set on which to run the classifier.

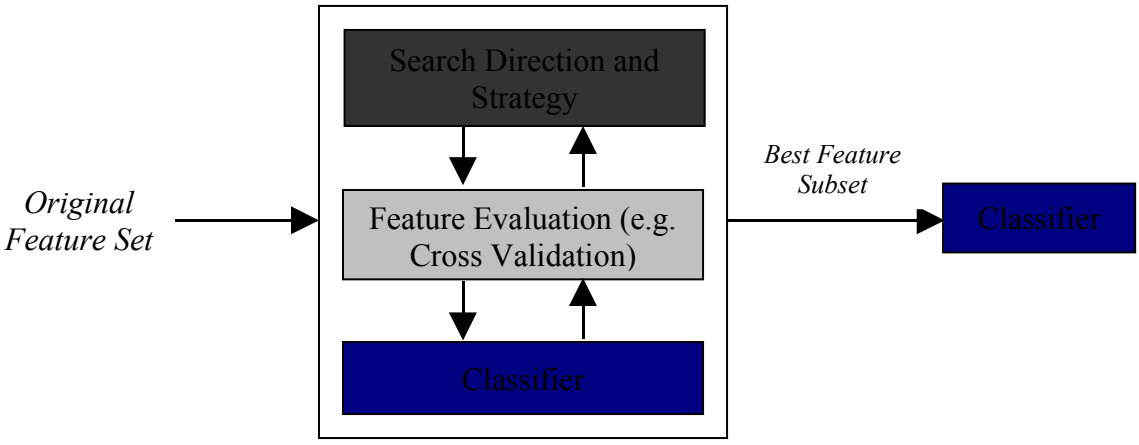

Figure 2.3. The Process of Wrapper Feature Selection

To have a better understanding of the Wrapper method, we use a Bayesian Network classifier as an example to explain the aforementioned process. The Bayesian Network classifier initially seeks and generates different feature subsets with some search strategy, one of which is forward selection. The forward selection strategy begins with no features and successively adds more features that are deemed relevant by the Bayesian Network classifier. In this way, interactions among the features can be considered. The feature subsets found using this strategy may then be evaluated

using the Bayesian Network's performance and 10-fold cross validation, where the subset with the highest accuracy is determined as the most relevant. This relevant subset is subsequently fed as input to the Bayesian Network classifier. The entire procedure for performing wrapper feature selection is illustrated in Figure 2.3.

In the above example, the Wrapper method was used with a Bayesian Network classifier. However, the Wrapper can be used with any type of classifier. Four types of classifiers most commonly used with the Wrapper to perform feature selection include: Bayesian Networks, Decision Trees, Nearest Neighbour and Support Vector Machines. These types of classifiers have been used to perform Wrapper feature selection in different research areas. Three areas which have received a lot of attention include Web mining, financial analysis and bioinformatics. Examples of how the aforementioned types of classifiers have been used with the Wrapper to perform feature selection in these three areas are presented in the following sections.

2.4.1 Bayesian Networks

Bayesian Networks (BN) are probabilistic classifiers used to model data (Pearl, 1988). BN are directed acyclic graphs consisting of links between nodes which represent features in a particular domain. Links are directed from a parent node (the target variable) to a child node (a feature in the dataset), and with each node there is an associated set of conditional probability distributions which describe how relevant it is with regards to the target variable and other nodes (Chen and Liginlal, 2007). In this way, BN classifiers have the ability to identify relationships between the nodes (features) in the graph structure. These attractive properties of BN classifiers have led several researchers to use them with the Wrapper to select relevant feature subsets. The BN classifiers have been used to select feature subsets mainly from bioinformatics data, notably gene expression data.

For example, Vinciotti et al. (2006) use a well known BN classifier called Naïve Bayes (Langley and Sage, 1994) with the Wrapper to select a subset of the most relevant genes from gene data that help diagnose cancer. The Naïve Bayes classifier was used with a simulated annealing search strategy, which reduces the chances of selecting less relevant gene subsets, on two gene datasets, including Prostate cancer dataset (N=1410 genes) and B-cell lymphoma cancer dataset (N=584 genes). This

feature selection approach was able to select a considerably small number of relevant genes from Prostate (approximately 25 genes) and from B-cell lymphoma (approximately 30 genes) datasets. These genes were found to generate very high accuracy levels and also strengthened biologists' understanding of the relationships between the relevant genes and the cancers investigated. Abraham, Simha and Iyengar (2007) also used the Naïve Bayes classifier with the Wrapper to select relevant genes from gene data. In fact, the authors used this classifier and the forward search strategy with the Wrapper to select relevant genes from 17 well known bioinformatics datasets. Experimental results using these datasets showed that the Wrapper with Naïve Bayes was able to select small number of relevant genes that led to classification accuracies higher than that of using the Naive Bayes classifier with all features.

2.4.2 Decision Trees

Decision trees (DT) are widely used to perform the task of classification and prediction (Mitchell, 1997; Polat and Güneş, 2006). The goal of DT is to uncover relationships by subdividing instances within data (Chien, et al., 2007). DT split instances in the data based on the values of one or more features. The splitting of instances typically relies on some criteria, which determine the relevance of instances and features with respect to the target variable (Chang, 2007). Once the splitting of the data reaches some predefined level, the classifier outputs a graphical representation of the splitting process in the form of a hierarchical tree structure (White and Sutcliffe, 2006). This tree structure built by DT helps visualise relevant features in the data as well as relationships among the relevant features. Due to this issue, many have used DT to identify a small number of the most relevant features within different datasets. More specifically, relevant features have been identified from datasets collected from the field of bioinformatics and the Web.

With regards to bioinformatics, Li, et al (2004) use a decision tree that employs the gini splitting criteria, typically referred to as Classification and Regression Trees (Breiman, et al, 1984), with the Wrapper method to select small number of most relevant genes for predicting cancer. Relevant features were selected from two well established cancer datasets, including the colon cancer (N=2000 genes) and leukaemia (N=7070 genes) dataset. The Wrapper method was able to identify 20

highly relevant genes from the colon dataset and 23 highly relevant genes from the leukaemia dataset. The relevant genes selected from both datasets were shown to generate higher classification accuracy levels than the accuracy generated using all genes. With regards to the Web, Stein et al. (2005) used the well known C4.5 decision tree (Quinlan, 1993) with the Wrapper method to search for a subset of relevant features that would help determine the likelihood of a hacker attack on the Web. The search for the subset of relevant features was facilitated using a Genetic Algorithm. Experimental results using the Wrapper with decision tree method and a dataset comprised of different types of Web attacks showed that the method was able to select a small number of relevant features from the dataset. These relevant features were then classified using the C4.5 classifier. Results clearly showed that features selected by decision tree led to higher accuracies than using all features with the decision tree. The selected features also helped identify the key characteristics associated with Web attacks.

2.4.3 Nearest Neighbour

Nearest Neighbour (NN) is probably the simplest instance-based classifier for carrying out classification tasks (Angiulli and Folino, 2007). The main premise behind NN is to classify a new instance x by finding the closest instance(s) to x according to some distance metric (usually the Euclidean metric) and assign it to the same class as the closest instance. The most common type of NN is the Nearest Neighbour Classifier (NNC) since it uses only the single closest instance for determining the class of x. An extension of NNC is the k-Nearest Neighbour (KNN) which uses more than one instance (k) for identifying the class value of x (Bell, Guan, and Bi, 2005; Huang, et al., 2007). NN family classifiers, like NNC and KNN, have been applied to many domains of research to do feature selection (Džeroski and Ženko, 2004). For example, Somol, et al. (2005) used the NNC classifier and the forward sequential floating search strategy with the Wrapper method in order to identify a set of relevant features that would assist in deciding whether or not to grant a loan to customers of a bank. The method was applied to two established financial datasets, including the German credit (N=20 features) and Australian credit (N=14 features) datasets. The relevant features selected from these datasets, which ranged from 5 to 10 features for both datasets, were found to yield higher classification

accuracies than using the full set of features and also facilitated the bank's decision making process for granting loans.

Another example can be found in Cornforth, et al (2004). In this example, the authors use the KNN classifier with the Wrapper and 10-fold cross validation for the purpose of determining the likelihood of individuals developing diabetes based on their heart rate. The KNN with the Wrapper was applied to a dataset comprising of 30 features which describe the characteristics of patients with and without diabetes. From this dataset, the Wrapper method was able to select a feature subset that consisted of seven relevant features. These features were classified using the KNN. The results showed that the selected relevant features: 1) generated higher classification accuracies than all 30 features and 2) accurately determined the chances of individuals developing diabetes.

2.4.4 Support Vector Machines

Support Vector Machines (SVM) are powerful classification techniques originally derived from Statistical Learning Theory (Cortes and Vapnik, 1995; Vapnik, 2000). The primary idea of SVM is to find an optimal partition (known as a hyperplane) that clearly separates the members and non-members of a given class in relation to the target variable (Stitson et al., 1996; Barakat and Bradley, 2007). This approach is typically used when the members of the target variable are linearly separable. In other words, the target variable has two class values. However, when there are more than two class values (i.e., non-linear target variable), it can be difficult to find a hyperplane to classify the data. SVM overcomes this difficulty by performing a non-linear mapping of the original feature space into a new feature space, usually with a higher dimension, so that the data are linearly separable (Gammerman, 1998; Saunders et al., 2000). This mapping is accomplished through the use of kernels, such as the polynomial and radial basis function kernels (Chen and Hsieh, 2006).

The SVM can therefore handle both linearly separable and non-linearly separable classification problems. This means that SVM can handle both simple (linear) and complex (non-linear) classification problems. Due to this issue, SVM have been successfully used to select relevant features from different types of datasets in order to facilitate the task of classification. One type of dataset that has been used with

SVM is gene dataset. Chiu, Chen, and Lin (2008) used the Wrapper with a greedy search strategy and the SVM classifier to select a small number of the most informative (i.e., relevant) genes for diagnosing breast cancer and the time it takes for the cancer to spread around the human body. This feature selection approach was applied to a breast cancer dataset comprised of 403 genes (features), and was subsequently able to select 44 genes that were most informative to the task of breast cancer classification. This subset of genes was able to produce better classification results compared to using all genes in the dataset. The genes were additionally found to help in the diagnosis of breast cancer.

The SVM has also been used to select features from datasets consisting of text content from Web pages. In fact, Abbasi, Chen, and Salem (2008) used the Wrapper method and the SVM with a Genetic Algorithm search strategy to analyse the text content of an English (N=12881 features) and Arabic (N=13811 features) Web forum. In fact, the method was used to identify a small number of features (in the form of words and sentences) from these forum Webpages that would prove relevant in determining the type of content (i.e., positive or negative information) present within the forums. The method was able to select a subset of 508 relevant features from English forum and a subset of 338 features from the Arabic forum Webpage. These relevant features were found to improve the classification accuracy compared to using the original set of features. The relevant features were also shown by the authors to be very useful for analysing the complex contents of Webpage forum documents.

2.4.5 Advantages and Disadvantages

As shown by the above examples, Wrapper methods have the ability to uncover small subsets of the most accurate features from different types of datasets. Many have also shown that the Wrappers are able to select more accurate feature subsets than the Filter methods which were previously outlined (Huang, Yang, and Chuang, 2008; Li and Guo, 2008). As a result, the Wrapper is typically regarded as being better than Filters for finding accurate feature subsets. The Wrapper method is able to find accurate feature subsets, but it has some disadvantages. The main disadvantages of the Wrapper are three fold: 1) the large amounts of time needed to perform feature selection, 2) the accuracy levels of selected feature subsets, and 3)

the use of a single classifier for selecting the relevant features. These three disadvantages may affect the way in which Wrappers select relevant features. Due to this issue, we will explain each disadvantage in depth and also present works that have attempted to overcome these disadvantages of the Wrapper.

1) Amount of Time

The main criticism of the Wrapper approach is the amount of time needed to perform feature selection (Chrysostomou, et al, 2008). Typically, the Wrapper will require considerably more time to examine feature subsets compared to Filters and any other feature selection approaches. This is because of two issues. The first issue concerns the use of a classifier. By using a classifier, more time is needed in examining each potential feature subset searched in the feature space. The second issue regards the use of cross validation. In general, cross validation is used in conjunction with the classifier to determine the level of accuracy of feature subsets. When both the classifier and cross validation are used together, the Wrapper runs prohibitively slowly. These drawbacks have led researchers to investigate ways of reducing the time of the Wrapper method.

As stated above, the classifier is one of the main reasons why the Wrapper performs slower than other feature selection approaches. To alleviate the effects of using a classifier, Caruana and Freitag (1994) developed a new method for speeding up the Wrapper approach when specifically used with decision tree classifiers. The method functions by reducing the number of decision trees grown during feature selection. This reduction is done by keeping a record of all the features that were used to construct the trees. By keeping such a record, less time is needed to analyse the features used in tree formation. In addition to two well known decision tree classifiers, ID3 and C4.5, five different search strategies were used to test the effectiveness of the method, including forward selection, backward elimination, forward stepwise selection, backward stepwise elimination, and backward stepwise elimination-SLASH, which is a bi-directional version of backward stepwise elimination. Empirical analysis revealed that, irrespective of the search strategy and decision tree classifier used, the time taken to perform feature selection decreased.

Furthermore, Kohavi and Sommerfield (1995) introduced the concept of 'compound' operators in an attempt to make the Wrapper perform in less time. The purpose of using compound operators is to direct the search strategy more quickly toward the most relevant features. In this way, the classifier will need to spend less time evaluating all the features. Experiments using the compound operators were carried out using two classifiers: ID3 and Naïve Bayes. Results showed a significant decrease in the amount of time needed to perform feature selection when either classifier was used. Improvements in classification accuracy for ID3 and Naïve Bayes were also found when compound operators were implemented.

Besides the classifiers, cross validation is another factor that can decrease the speed at which the Wrapper performs feature selection. A strategy for reducing the Wrapper's time complexity when used in conjunction with cross validation was presented by Moore and Lee (1994). The strategy reduces the number of instances used during the evaluation stage of feature selection so the cost of fully evaluating each feature subset is also decreased. They showed that the new strategy successfully reduced the number of feature subsets evaluated during feature selection. This led to a drop in the amount of time needed to perform Wrapper feature selection. It was also found that the reliability of the chosen feature subset was unaffected by the fall in the number of instances.

Hashemi (2005) also investigated the effects of reducing the number of instances used for feature selection. Hashemi presented a new Wrapper approach that performs feature selection roughly 75 times faster than traditional Wrapper approaches. The new Wrapper approach does this by using an algorithm called Atypical Sequential Removing (ASR). The ASR algorithm finds and removes those instances in the data, which do not influence classifier performance. By decreasing the number of instances, the process of feature selection can be sped up as there will be less data to deal with. Experiments were carried out using the proposed wrapper approach with different classifiers, including Support Vector Machine (SVM), *k*-Nearest Neighbour and C4.5. Overall, findings showed that although the accuracy of some classifiers did not improve when compared to the use of all instances, the new Wrapper method performed much faster.

2) Level of Accuracy

The previous section has shown that it is possible to reduce the time needed to perform Wrapper feature selection. Although time complexity is a major issue, especially when many features and instances are involved, the accuracy of the chosen feature subset is also very important. The idea of accuracy and time is interrelated because improving one may affect the other. Numerous studies have investigated this relationship by using evolutionary search strategies called Genetic Algorithms (GA) (e.g., Ni and Liu, 2004) because GA possess powerful search capabilities (Sikora and Piramuthu, 2007). Rhitoff, et al. (2002) is an example of works using GA. They incorporate GA with the Wrapper method to form a feature selection technique that avoids a suboptimal solution without sacrificing much in speed. Specifically, the GA Wrapper uses SVM as the classifier when performing feature selection. Results showed that accuracy significantly improved when compared to using no feature selection. Their approach was also tested against the well known sequential forward selection Wrapper with similar findings.

Another framework combining the uses of GA and feature selection approaches can be found in Sikora and Piramuthu (2007). This framework uses GA with the Hausdorff distance measure for Wrapper feature selection. Experimental results comparing this framework to a GA-based Wrapper approach without the Hausdorff distance measure showed that it provided superior performance. The GA and Hausdorff Wrapper feature selection approach was not only able to improve classification accuracy by about 10% but was also able to reduce the amount of time by 60%. This shows that the accuracy of the Wrapper can increase even when it performs feature selection at a faster rate.

Furthermore, Ruiz et al. (2006) developed a new gene selection method called Best Incremental Ranked Subset (BIRS) based on the Wrapper approach. The method aims to improve classification accuracy of cancer data without affecting the time taken to do the feature selection. BIRS does this by first ranking the genes. A small subset of genes with the highest rank is then fed as input to the Wrapper. The method was tested using three different classifiers, i.e., Naïve Bayes, Nearest Neighbour-IB1, and C4.5, on four DNA microarray datasets. Experimental results on these datasets showed that BIRS was a very fast feature selection approach when compared to a

Wrapper that uses all the genes. In addition, BIRS was found to produce good classification accuracy.

3) Single Classifier

The other limitation of Wrappers relates to the use of a single classifier. Wrappers make use of a single classifier when selecting relevant features. Using a single classifier can be a problem because each classifier is different, which means that they will possess different biases and assumptions. Table 2.1 shows the biases and assumptions of the four commonly used classifiers previously described. From this table, it can be seen that different classifiers have different biases and assumptions which make them focus on different features when doing feature selection.

Type of Classifier	Assumption of Classifier	Bias of Classifier
Bayesian Networks (BN)	Assume conditional independence in that each feature in the dataset is independent of all other features.	To focus on features that maximise/minimise some scoring metric when building the network structure
Decision Trees (DT)	Assume that features and instances in the dataset fulfil the splitting criterion used by the classifier	To focus on those features that satisfy the criterion used to build the decision tree
Nearest Neighbour (NN)	Assume distance among features and instances in dataset according to a predefined distance measure	To focus on features and instances that are deemed the 'closest' by the imposed distance measure
Support Vector Machines (SVM)	Assume that the dataset follows the Identical and Independent Distribution (I.I.D)	To focus on the features that lie on the classification boundary, which best separates the dataset

Table 2.1. Biases and Assumptions of Different Classifiers

As such, classifiers with different biases and assumptions may differ in the amount of time they take to perform feature selection and may also select features that generate different levels of accuracy. In terms of the amount of time, a classifier that is considered to be theoretically complex may take longer to select relevant features than a classifier which is regarded as theoretically simple. For example, using a Wrapper with a complex classifier, such as Support Vector Machine, may take more time to identify the relevant features than a Wrapper with a simpler classifier, like Nearest Neighbour. In terms of accuracy, classifiers with different biases and assumptions will most probably select different features (see Table 2.1). The fact that

they select different features may mean that they will generate different levels of accuracy. Collectively, this shows that using different classifiers with the Wrapper may significantly affect both the time taken to do the feature selection and the accuracy levels of selected features. This therefore suggests that the choice of a classifier plays an important role in feature selection.

A possible way of addressing this important limitation of using a single classifier is to use more than one classifier. It may be worthwhile using several different types of classifiers with the Wrapper to perform feature selection. By using several different classifiers for feature selection, the biases of individual classifiers can be reduced, which in turn can lead to the selection of mutually agreed and unbiased sets of relevant features. Such unbiased sets of features may subsequently lead to high classification accuracies. In addition, using multiple classifiers will result in the assumptions made by the classifiers to be considered during the feature selection. In other words, using multiple classifiers may help select features that lead to higher levels of accuracies than features selected by single classifiers. In general, few attempts have been made to address the problem of using a single classifier. In addition, there is little work to address this problem through the use of multiple classifiers. This therefore leaves a gap in Wrapper feature selection research that needs to be investigated.

2.5 Conclusions

This chapter gave a brief introduction to the notion of feature selection. Moreover, this chapter reviewed the state-of-the-art of two commonly used feature selection methods, namely Filters and Wrappers. The former methods do not use classifiers but instead use statistics and the general characteristics of the data to determine relevant features. The latter methods, however, rely on classifiers to select the most relevant sets of features. This means that Filters are classifier-independent and Wrappers are classifier-dependent. Many studies have shown that Wrappers perform better than Filters in the sense that they are able to select more accurate feature subsets. Wrapper methods are useful, but they suffer from some problems. Three problems were mentioned in this chapter, the most important one being the fact that Wrappers use a single classifier to select the relevant features. This is because each classifier is different and using different classifiers may result in different features

being selected. In fact, different features may be selected that lead to different accuracy levels.

As previously mentioned in this chapter, a possible way of overcoming this problem may be to use multiple classifiers with the Wrapper for feature selection. By considering several different classifiers when doing feature selection, mutually agreed and unbiased sets of relevant features that lead to high accuracies can be found. Since this approach may enhance the performance of the Wrapper and improve its chances of identifying highly relevant features, the next chapter describes and explains a novel data mining method called Wrapper-based Decision Trees (WDT). The WDT method is novel in that it combines multiple classifiers with the Wrapper to select relevant features and also has the added benefit of visualising the relationships between the selected features using decision trees. Consequently, the WDT method may be the first step in filling in the gap in Wrapper feature selection research.

Chapter 3 – Wrapper-based Decision Trees (WDT)

3.1 Introduction

The previous chapter gave a review of the two main methods for performing feature selection. The two main methods include: Filters and Wrappers. The former relies on the characteristics of the data and some statistical criterion to determine relevant features, whereas the latter uses a classifier to identify the subset of most relevant features. The former are therefore classifier-independent whereas the latter are classifier-dependent. Among these two feature selection methods, Wrappers usually select feature subsets that are of greater relevance to the target variable and in turn provide better classification performance (Talavera, 2005; Ruiz, Riquelme, and Aguilar-Ruiz, 2006).

However, as pointed out in the previous chapter, Wrapper methods possess a limitation in that they only use a single classifier when selecting a subset of the relevant features. The limitation with using a single classifier is that each classifier is of a different nature and will have its own biases. To better understand differences in classifiers, Table 3.1 presents the nature and biases of classifiers belonging to four of the most popular classifier families which were detailed in the previous chapter, including Bayesian Networks, Decision Trees, Nearest Neighbour, and Support Vector Machines. Table 2.1, which was previously presented in Chapter 2, additionally shows that these four classifier families make different assumptions on the data used. As shown in both Table 2.1 and Table 3.1, different families of classifiers use a different approach and focus on different aspects when handling the data. Due to these differences, each classifier will select a different feature subset which may contain different features and may also lead to different levels of classification accuracy. In other words, using a single classifier for feature selection may affect the feature selection outputs.

Family	Classifiers	Nature	Bias
Bayesian Networks (BN)	*Bayesian Network Classifier (BNC)*	To use conditional probability distributions to identify the relationship between a feature and a targeted variable.	To focus on features that maximise or minimise some scoring metric when building the network structure.
	Naïve Bayes (NB)		
	Average-One Dependence Estimators (AODE)		
Decision Trees (DT)	*C4.5*	To use a hierarchical structure to represent the most informative features.	To focus on those features that satisfy the criterion used to build the decision tree.
	Classification And Regression Tree (CART)		
	CN2		
Nearest Neighbour (NN)	*Nearest Neighbour (NNC)*	To use a distance metric to select the closest instance(s).	To focus on features and instances that are deemed the 'closest' by the imposed distance measure.
	k-Nearest Neighbour (KNN)		
	*K**		
Support Vector Machines (SVM)	*Support Vector Machine and Polynomial Kernel (SVMpoly)*	To follow the statistical learning theory to find the best division that separates the different categories of a dataset.	To focus on the features that lie on the classification boundary, which best separates the dataset.
	Support Vector Machine and Radial Basis Function Kernel (SVMrbf)		

Table 3.1. Nature and Biases of Different Classifiers

The fact that different classifiers may lead to different feature subsets suggests that there is a need to consider combining several different classifiers for feature selection. By combining multiple classifiers, the biases of each individual classifier can be reduced by deriving a consensus from the features selected by these different classifiers. In addition, the assumptions made by each classifier are also taken into account when selecting the consensus features. Based on this idea of combining multiple classifiers, a new data mining method called Wrapper-based Decision Trees (WDT) is proposed in this chapter. The WDT method (Chrysostomou, Chen, and Liu, in press a) combines two data mining techniques, namely feature selection and classification. In the case of feature selection, Consensus Feature Selection (Chrysostomou, Chen, and Liu, in press b), which uses multiple classifiers with the Wrapper approach to generate a mutually agreed and unbiased set of relevant features, is used. In the case of classification, decision trees are adopted to visualise the relationships among the selected relevant features.

This chapter is organised as follows. It will start in Section 3.2 by giving a detailed explanation of the WDT method and its two elements, i.e., consensus feature

selection and decision tree classification. The pseudocode for the WDT method will also be given in order to better understand how the method identifies relevant features using multiple classifiers (Section 3.3). Since the WDT method uses multiple classifiers for feature selection, choosing suitable classifiers becomes an important issue. More specifically, Section 3.4 explains the importance of the number and nature of classifiers when dealing with multiple classifiers, with an emphasis on how these two issues will be investigated in this thesis.

3.2 Wrapper-based Decision Trees (WDT)

As previously stated, WDT makes use of two elements: consensus feature selection and decision tree construction. A simple diagram illustrating these two elements is shown in Figure 3.1. Both of these elements will be explained in detail in the following sections. We begin by explaining the idea of consensus feature selection in Section 3.2.1 and then progress to explain the use of decision trees to visualise the interactions among the features selected by consensus feature selection (Section 3.2.2).

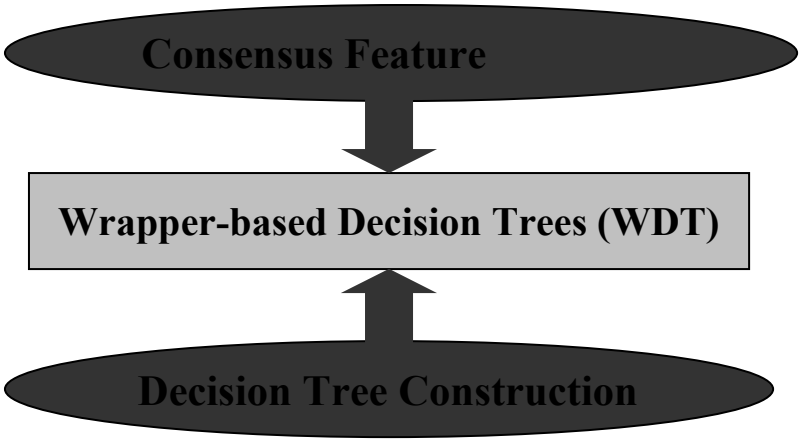

Figure 3.1. Elements of WDT

3.2.1 Consensus Feature Selection

The WDT method combines multiple classifiers to help reduce the effects of biases of individual classifiers. The combining of classifiers is done by consensus feature selection (CFS). CFS (Chrysostomou, Chen, and Liu, in press b) can be used with different types of classifiers. In this thesis, classifiers from four different classifier families are used, namely Bayesian Networks (BN) (Friedman, Geiger and

Goldszmidt, 1997), Decision Trees (DT) (Benbrahim and Bensaid, 2000), Nearest Neighbour (NN) (Cover and Hart, 1967) and Support Vector Machines (SVM) (Cristianini and Shawe-Taylor, 2000). These classifier families were shown previously in Table 3.1 and were also mentioned in Chapter 2. These four classifier families were chosen because of their different nature and biases (see Table 3.1). The advantage of combining such different classifiers for feature selection is that we will be able to generate a single feature set that contains features selected by each of the classifiers used in CFS.

To illustrate how the aforementioned classifiers perform feature selection in CFS, the SVM classifier will be used as an example. In this example, the SVM classifier is used to select relevant features from any given dataset comprised of a total of x features. Relevant features will be selected from these x features for a particular target variable. Initially, the SVM uses all data (i.e., all x features) as input. The classifier is then executed to identify the subset of relevant features from x in relation to the target variable (predictor) by using a search strategy, which outlines the way in which feature subsets are searched within the entire set of features. There are different search strategies that can be used, but they basically fall into two main categories: exhaustive strategies and heuristic strategies. An exhaustive search systematically examines all possible subsets and selects the 'best' subset of relevant features. Some classic examples of exhaustive search techniques are branch-and-bound (Chen, 2003), depth-first (Dash and Liu, 2003) and breadth-first (Tso and Gu, 2004). Exhaustive strategies often find the most relevant subsets of features but become very difficult to use when the number of features increases. Instead of exhaustive search, a more feasible approach to use may be a heuristic search because it does not consider every possible combination of features in the dataset. In this way, a heuristic strategy selects subsets of features that are as close as possible to being the 'best' subset of features. Common examples of heuristic strategies include forward strategy (Kittler, 1978) and backward strategy (Marill and Green, 1963).

Among these strategies, the forward search strategy, which belongs to the heuristic search family, was chosen for use in CFS. The forward search strategy guides the classifier in a forward manner so that few features are initially used and then more are continuously added to form the feature subsets. The forward search strategy has

two key benefits worth mentioning. The first benefit of the forward strategy is that it is able to select a relevant subset of features without taking large amounts of time to perform feature selection (Kohavi and John, 1997). This is particularly important because Wrappers usually perform feature selection very slowly. The second, and probably the most important benefit, is that a forward search strategy can help filter out redundant features (i.e., features that are dependent on the other features, and as such, provide no further information about the target variable). Since the forward search strategy starts with no features and gradually adds features, dependencies between selected features will be easily detected and therefore not be included in the final subset of chosen features (Langley and Sage, 1994). Excluding redundant features from the chosen feature subset in this manner can dramatically increase relevance of the feature subset, and consequently increase the classification accuracy.

Once the feature subsets have been identified by the forward search strategy, they are evaluated according to an accuracy estimation technique. In this work, *k*-fold cross validation was used as the accuracy estimation technique because it is most popularly used to evaluate the relevance of features selected by Wrappers (John, et al, 1994). Typically, *k*-fold cross validation involves splitting the training (i.e., input) data into *k* approximately equally sized partitions. The chosen classifier is then run *k* times using *k*-1 partitions as the training set and the remaining partition as the test set. The accuracy results from each of the *k* runs are then averaged to produce the estimated accuracy. In this thesis, *k* was chosen to be 10 because this value of *k* has been widely used when applying cross validation to different data mining tasks (Sboner, et al, 2003; Williams, Zander and Armitage, 2006; Pirooznia, et al, 2008). At the end of this cross validation procedure, each of the features will be assigned a value, which indicates the number of times they were chosen by 10-fold cross validation. This value is the output of the process of feature selection, which reveals the level of the relevance of this feature (outcome variable). For example, a feature that was assigned the value 10 by SVM implies that this feature was included in all 10 times/samples of the cross validation procedure. This also implies that this feature is highly relevant to the target variable.

The selected sets of relevant features will then form a matrix where each row will represent a feature contained in the dataset and each column will present the different

classifiers used. The contents of the matrix include the ranked values of each feature by each classifier, ranging from 0 (indicating that the feature was not selected) to 10 (indicating the feature was always selected through cross validation). In general, the main purpose of the matrix is to find agreement between features selected by the classifiers. In this case, the agreement can be calculated using different combination strategies. These can include calculating the maximum value of a feature's rank, the minimum value, the sum, the mean, and the median (Kittler, 1998; Kittler et al, 1998). Among these strategies, both the mean and median seemed suitable for CFS because the results of each classifier can be within the range of 0 to 10, which corresponds to the nature of 10-fold cross validation (i.e., the accuracy estimation technique used).

However, Kittler et al (1998) indicated that an outlier present in the data can seriously distort the result of the mean strategy. To better understand how the outlier affects the mean results, consider a feature which has three relevance levels associated with it (as determined by three different classifiers). In the event that two of the relevance values are 0 (i.e., the feature was regarded as irrelevant by two of the classifiers) and the third relevance value is 10, the resulting mean for the feature will be reasonably high; the mean will be 3.33. The mean of this feature shows that it is of reasonable relevance to the target variable, where in fact it is probably irrelevant to the target variable since two out of the three classifiers considered it to be irrelevant. In other words, the outlier, which is the highest relevance level of 10, may lead to incorrect results.

As a more robust approach to handling the presence of outliers, Kittler et al advised the use of the median combination strategy. In order to see how the median strategy can handle outliers, we consult the abovementioned example. In the example, one classifier assigned highest relevance value to the feature (i.e., 10) while the other two classifiers did not assign a relevance value (i.e., 0). The median of the feature will therefore be 0, indicating that the feature is irrelevant to the target variable. The median strategy can therefore give us a better idea of the relevance of features as determined by the classifiers. In their work, Kittler et al also showed that the results of different classifiers combined using the median strategy generally led to better classification performance than the results from the other classifier combination

strategies. Due to such evidence, the median strategy is employed to combine results from the multiple classifiers used.

To have a better understanding of the way in which the median strategy is used to combine classifiers in CFS, Table 3.2 shows a very simple example of what the final matrix would look like if three classifiers were used to select relevant features from a dataset comprised of five features. From this table, we can see that each feature has three values associated with it, which represents the results from the three classifiers. In addition, the table shows the median values for each feature, which corresponds to the final relevance value of each feature. In this example, Feature 2 and Feature 3 are irrelevant since the relevance values for these features are 0. On the other hand, Feature 1, Feature 4 and Feature 5 are relevant. However, they each have a different level of relevance. Feature 1 and Feature 5 are quite relevant whereas Feature 4 is the most relevant out of all features because it has the highest possible relevance level of 10.

	Classifier 1	**Classifier 2**	**Classifier 3**	**Relevance Level (Median)**
Feature 1	2	1	10	2
Feature 2	2	0	0	0
Feature 3	0	1	0	0
Feature 4	10	10	8	10
Feature 5	7	2	5	5

Table 3.2. Example of Classifier Combination Matrix

In order to obtain an entire set of relevant features, CFS not only takes into account features with high relevance, but also includes those with low relevance. More specifically, all features that have a median value of 1 or more are chosen and included in the final set of relevant features.

3.2.2 Differences Between CFS and Other Classifier Combination Strategies

As explained in the previous section, CFS, which is the first element of the WDT method, combines multiple classifiers for feature selection using the median combination strategy. However, the median combination strategy is one way of combining the results from classifiers. There are other combination strategies that have been widely used to combine classifier outputs, namely *bagging*, *boosting*, and *model averaging*. These well-known combination strategies are quite different to that

of the median strategy. In order to better understand how these three additional combination strategies differ from the one used in CFS, a brief description of each strategy is given in the next few pages.

Bagging (also known as Bootstrap Aggregating) involves selecting a set of instances randomly drawn (with the chance of replacement) from the original set of instances (Breiman, 1996). In this way, some instances may appear more than once while others may not appear at all. These newly formed sets of instances, also known as *bootstraps*, are then used as input to a classifier, which will generate some level of accuracy. This process can be repeated as many times as required in order to seek the highest accuracy level from the classifier. Boosting, on the other hand, deals with all instances present in the dataset. All instances are fed as input to a single classifier, which performs classification using these instances. The output of the classifier is then used as input to another classifier, which is identical to the first classifier originally used (Schapire, 1990). This input-output classifier process can be repeated many times, where each time different weights are assigned to different instances according to the number of times they were used by the classifiers. The outcome of this process is therefore a set of classifiers, each of which produces a different classification result (depending on the instances they used). The results from these classifiers are then combined through voting to create a composite classifier (Freund, 1995). Model averaging is another means of combining the results from a classifier. In this case, a classifier is executed several times on the same dataset using different input parameters (Yeung, Bumgarner, and Raftery, 2005; Gibbons, et al, 2008). As such, several classification models will be produced. Model averaging considers all these classification models in a weighted manner, where the weights are determined using a criterion, and selects the model which will provide better classification performance (Selen, et al., 2004).

Bagging, boosting and model averaging are useful classifier combination strategies. However, these strategies share two common limitations. The first common limitation of these strategies is that they combine the results of the classifiers by using the instances in the datasets rather than the features of the dataset. In fact, bagging uses different subsets of instances selected at random from the dataset, feeds these subsets to several classifiers and combines the results of the classifiers until the

desired level of accuracy is reached. Furthermore, boosting uses the subset of instances with the highest weights when performing classification with the classifiers. In model averaging, only the set of instances used in the model which provides best performance are used. The fact that all three strategies rely on the instances to combine classifier results means that they somewhat disregard the features in the datasets. In essence, the strategies place less emphasis on identifying the (relevant) features in the datasets. Disregarding relevant features in the dataset can in turn affect the classification accuracies generated. On the other hand, combinations strategies that combine classifier outputs based on the features in the dataset, like the median strategy, may be better suited to the feature selection process and may subsequently help increase classification accuracies. The second common limitation of these strategies is that they use a single (type of) classifier. More specifically, bagging and boosting run a single classifier on several different partitions of the dataset to produce different results while model averaging executes a single classifier several times, each time altering the classifier's input parameters. The disadvantage of using a single classifier is that each classifier has different biases, as explained in previous section. On the other hand, using several different classifiers can reduce the biases of each individual classifier. This is another reason why the median combination strategy, which can accommodate several different classifiers, was used to combine classifier outputs in CFS.

3.2.3 Decision Tree Construction

Decision tree construction is the second element of the WDT method. The relationships among the features selected by CFS (i.e., those features with a relevance value of 1 or more) are visualised with a decision tree. The reason for choosing the decision tree to classify the selected relevant features is two fold. First, we will be able to determine the classification accuracies of feature subsets, i.e., how relevant the features are to the target variable. Second, the decision tree presents the final classification result in the form of a hierarchical structure. This hierarchical structure can help understand the relationships among features in datasets (Polat and Güneş, 2009). In addition, the constructed tree structure enables the most relevant feature(s) to be located at the top of the tree and the least relevant feature(s) to be located at the bottom of the tree (Chien et al., 2007). This will improve our understanding of how features with different relevance interact with one another.

The decision tree employed in WDT to identify relationships among relevant features and determine accuracy levels of the features is C4.5 (Quinlan, 1993). The C4.5 classifier is probably the most widely known and used decision tree classifier in data mining literature (Lu and Chen, 2009; Polat and Güneş, 2009). In fact, C4.5 has been successfully used to visualise relationships in datasets belonging to different data mining applications, including 1) Web-based learning, 2) bioinformatics, and 3) finance. In terms of the first application, Cristian and Dan (2006) used the C4.5 decision tree, and classification rules extracted from the decision tree formed, to analyse student's activity and their behaviour on a Web-based learning system. In terms of the second application, Ture, Tokatli, and Kurt (2009) used decision tree, including the C4.5 classifier, on bioinformatics data to determine the survival of breast cancer patients based on their symptoms. Their symptoms were visualised in tree structures, each of which led to a set of classification rules to help identify factors affecting the survival of patients. Finally, Florez-Lopez (2007) used the C4.5 decision tree classifier to tackle the problem of calculating credit risk ratings for different financial insurance companies. The C4.5 classifier was able to select a relevant set of features from the financial data used and also show the relationships between the relevant features.

The aforementioned works demonstrated that the decision tree, especially the C4.5, is a very useful tool for identifying the relationships among relevant features in data. The C4.5 classifier is thus used in WDT to highlight relationships among the features selected by CFS.

3.3 Psuedocode of WDT

As detailed in the previous sections, the WDT method integrates the use of CFS and decision tree construction, with the aim of identifying a set of consensus features from a number of classifiers and visualising relationships between these consensus features. WDT selects relevant features with a consensus approach that looks for a consensus across several different classifiers. In this way, the consensus approach helps reduce the biases associated with each individual classifier used so that unbiased sets of relevant features can be identified. These features are further

analysed through decision tree classifiers, which is used to uncover hidden relationships among the selected features.

```
1.  Input: Dataset (data), Target Variable (tv), Classifiers c(j) where (j=1,2…r),
2.  threshold m_threshold
3.  Output: List of consensus features Flist(i) where (i=1,2…s), Decision Tree tree
4.
5.  Begin
6.  Flist(i) = {}
7.  m_threshold = 1
8.  k = 10
9.
10. //Consensus Feature Selection
11. For n = 1 to k
12.     For j = 1 to r
13.         Set target variable of c(j) to tv
14.         Perform wrapper feature selection using forward search on data
15.         Train c(j) using k-n data folds of data
16.         Test c(j) with current n data fold of data
17.         Assign each feature f (excluding tv) a rank 0~10,
18.             indicating level of relevance
19.     End_For
20. End_For
21. For i = 1 to s
22.     For each feature f in data (excluding tv) compute median
23.         If median(f) is ≥ m_threshold
24.             then add f to Flist(i)
25.         Else
26.             get the next feature
27.         End_If
28.     End_For
29.     //Decision Tree Construction
30.     Use Flist(i) as set of input features
31.     Set target variable of decision tree to tv
32.     Build and Display tree
33. End_For
34. End
```

Figure 3.2. Pseudo Algorithm of Wrapper-based Decision Trees (WDT)

To clearly explain the process of WDT, Figure 3.2 presents a description of its pseudo algorithm. The algorithm initially takes as input the entire dataset of instances and features, the target variable, any number of classifiers, and the median threshold value used to determine the consensus relevant features [Line 1-2]. After this, the list of consensus features is set to an empty list [Line 6]. The median threshold value is set to 1 [Line 7] and the value of k to be used for cross validation is initialised to 10 [Line 8]. Once all inputs and parameters are set, the algorithm performs two main steps: (1) Consensus Feature Selection and (2) Decision Tree Construction. In Consensus Feature Selection, different classifiers are used for Wrapper feature selection [Lines 11-20]. Firstly, the target variable is set for each classifier [Line 13].

Using different cross validation partitions (folds) of the dataset, each classifier (*c(j)*) generates a ranked subset of features in a forward manner through the Wrapper approach [Lines 14-16]. Each feature in the subset will have an associated level of relevance (between 0 and 10), which indicates how relevant the feature is to the target variable [Lines 17-18]. The ranked subsets from the (*c(j)*) classifiers are then used to form different classifier combinations using the median combination strategy. In this way, different lists of consensus features (*Flist(i)*) are identified [Lines 21-28]. Each consensus feature found with respect to the target variable will also have an associated level of relevance (i.e., the median value). In the event the associated level of relevance is greater than or equal to the set *m_threshold*, the feature is considered to be relevant. The relevant feature subsets with regards to the target variable are then used as input into the decision tree classifier [Lines 29-30]. The target variable of the decision tree classifier is also set (*tv*) [Line 31]. Finally, the decision tree classifier is executed to build and display a decision tree (*tree*) for each of the consensus feature subsets [Lines 32-34].

Both steps of the WDT algorithm explained and shown in Figure 3.2 were implemented using two software environments: (1) Waikato Environment for Knowledge Analysis (WEKA) v3.4.11 and (2) MATLAB v7.1. With regards to the step of Consensus Feature Selection, the WEKA environment was initially used to perform Wrapper feature selection with any number of the individual classifiers previously shown in Table 3.1. The features selected by the classifiers were then combined using the MATLAB environment. More precisely, MATLAB was used to form the matrix that included all features selected by the classifiers used. The median value was then calculated for each feature to determine the relevant consensus features. The consensus features were then presented as input to the C4.5 classifier in WEKA in order to build the decision trees (Decision Tree Construction step of WDT).

3.4 Effects of Using Multiple Classifiers

As detailed in the previous section, the novelty of the WDT lies within the combination of different types of classifiers for feature selection in order to generate unbiased relevant features and the use of decision trees to discover relationships

among several of the selected features. However, different classifiers for feature selection may result in different features being selected. In fact, the number of combined classifiers and the nature of classifiers combined may influence the relevant features selected. Regarding the former, more than one classifier can be used to perform the feature selection in WDT. Thus, varying the number of classifiers used may result in a decrease or an increase in the number of relevant features selected. Regarding the latter, there are many different types of classifiers that can be used, each with its own advantages and disadvantages. The fact that each classifier has its own unique characteristics means that it has the ability to select different features. As such, the nature of a classifier has the potential of influencing the features selected. In summary, the number and nature of classifiers are two issues that may influence the outputs of feature selection. To better understand the influences of the number and nature of classifiers, the WDT method will be used throughout this thesis in conjunction with two different classifier arrangement approaches and two types of datasets.

3.4.1 Classifier Arrangement Approaches

This thesis will use two different classifier arrangement approaches with WDT, namely the *same-type approach* and the *mixed-type approach*. On the one hand, the same-type approach combines classifiers from same family and uses them with WDT to select the relevant features. On the other hand, the mixed-type approach combines classifiers that are from different families to select features. In both approaches, different numbers of classifiers are combined. By using these two approaches, we will be able to: 1) investigate the influences of each of the classifier families on feature selection results, which will give us a better idea as to how each classifier family performs feature selection and which ones are less or more suitable than others (*same-type*), and 2) investigate how classifiers from the different families interact with one another and how their interaction influences the feature selection (*mixed-type*). In this way, both approaches will help us obtain a complete picture of the influences of number and nature of classifiers on feature selection. In addition to these approaches, WDT is used with two types of datasets. Thorough details on these two datasets are provided in the following section.

3.4.2 Datasets

The WDT method is used with two different types of datasets, each of which belong to the area of HCI and include the preferences of different users. The reason why such user preference datasets were chosen for use in this thesis lies within their nature. The nature of user preference datasets is different to that of other types of datasets. Datasets comprised of user preferences typically include a degree of fuzziness. This fuzziness comes from the fact that users may be uncertain of what they like or dislike, i.e., what they prefer. The presence of fuzziness can lead to noise within the data and such noisy data may be irrelevant to identify users' preferences. In other words, the data may include irrelevant features. In order to accurately distinguish between the preferences of different users, such noise and irrelevant features must be removed from the data. A way of removing such noise and irrelevant features from the data is to use a feature selection method, such as WDT. This explains why such datasets were chosen in this thesis to investigate the influences of number and nature of classifiers.

A detailed description of both user preference datasets is given in the next few pages.

1) Dataset Description of UP1

The first dataset includes users' preferences of search engine interface elements (Chen, 2000). The users were required to perform some searching and browsing tasks using the popular Google and Yahoo search engines, and then provide their preferences of the search engines. The preferences of 120 users were collected using a questionnaire with 90 statements, each of which had five possible answers, including 'very unimportant'; 'unimportant'; 'neutral'; 'important'; and 'very important'. In addition to their preferences, the questionnaire collected some personal details from the users, including their gender, cognitive style, level of computer experience, and level of Internet experience. These personal details are typically referred to as human factors (Treu, 1994; Frias-Martinez, et al, 2007). Each of these four human factors is shown in Table 3.3. These four human factors are popularly examined within user preference datasets. This is because much research has shown that gender (Ford and Miller, 1996; Roy and Chi, 2003), cognitive styles

(Chen and Macredie, 2004; Ford, Miller and Moss, 2005), level of computer experience (Lazander, et al., 2000; Mitchell, et al., 2005), and level of Internet experience (Liaw and Huang, 2006; Castañeda, Muñoz-Leiva, and Luque, 2007) significantly influence users' preferences.

Human Factor (Target Variable)	**Description**
Gender	*Male* *Female*
Cognitive Style	*Field Independent* *Field Dependent* *Intermediate*
Level of Internet Experience	*Little* *Average* *Good* *Excellent*
Level of Computer Experience	*Little* *Average* *Good* *Excellent*

Table 3.3. Description of the Human Factors / Target Variables

The users' preferences and their human factors mentioned previously constitute to the number of features and target variables of the dataset, respectively. As such, the dataset consists of 90 features, which represent the users' responses to the 90 questions in the questionnaire, and four potential target variables, which represent the four types of human factors. In addition, the number of instances in the dataset is 120, which corresponds to the number of users that answered the questionnaire. In summary, the user preference dataset includes 120 instances, 90 features, and four possible target variables. This user preference dataset will be referred to as UP1 from this point forward in the thesis.

2) Dataset Description of UP2

The second dataset used with WDT consists of users' preferences of a Web-based learning system (Chen and Macredie, 2004). The Web-based learning system aims to teach students how to use the Hyper Text Markup Language (HTML). Users were then required to document their preferences of the system using a questionnaire. More specifically, 65 users supplied their preferences using a questionnaire with 20 statements, each of which had five possible answers, including 'strongly agree'; 'agree'; 'neutral'; 'disagree'; and 'strongly disagree'. In addition to their preferences, users were also required to supply some of their personal details. In fact, users had to provide their gender, cognitive style and their level of computer and Internet

experience. These personal details (i.e., human factors) were identical to those collected in UP1 and shown in Table 3.3.

In summary, users' preferences of the Web-based learning system and their personal details make up the input features and target variables of this dataset, respectively. The dataset will thus have 65 instances, which correspond to number of users that used the system and 20 features, which correspond to the preferences of the users. In addition, the dataset includes four possible target variables, each of which corresponds to the users' human factors. This dataset will be referred to as UP2 dataset hereafter.

3) Determining Target Variables of Datasets

As described in the previous subsections, both the UP1 and UP2 datasets each include four possible target variables. Since a dataset may only have one target variable when doing classification and feature selection, it is necessary to identify a suitable target variable. In order to determine the target variable, we use the classifiers from each of the families previously mentioned and each of the four human factors presented in Table 3.3 to select relevant features from UP1 and UP2. The human factor that leads to the highest number of features will be chosen as the target variable of the dataset and used with WDT. This is because with a higher number of features the search strategy used in the WDT method will be able to evaluate and examine a larger number of feature subsets. Such a larger number of feature subsets can facilitate us to investigate how the number and nature of classifiers influence the results of feature selection.

The number of relevant features selected by each individual classifier for each human factor of both UP1 and UP2 datasets is shown in Table 3.4. In this table, we also present the mean number of features selected for each of the human factors. According to the mean number of features selected, we found different results for UP1 and UP2. In terms of UP1, we found that the classifiers selected the highest mean number of features when Computer Experience was the target variable. Due to the fact that Computer Experience led to the highest number of features, it will be chosen and used as the target variable of UP1 from here onwards. In terms of UP2, we found that the classifiers selected the highest mean number of features when

Cognitive Style was the target variable. As such, Cognitive Style will be used as the target variable of the UP2 dataset from this point forward.

	BN Family Classifiers			**DT Family Classifiers**			**NN Family Classifiers**			**SVM Family Classifiers**		**Mean no. of features selected**
UP1	*BNC*	*NB*	*AODE*	*C4.5*	*CART*	*CN2*	*NNC*	*KNN*	*K**	*SVMpoly*	*SVMrbf*	
Gender	8	20	24	28	20	19	20	19	19	21	17	19.31
Cognitive Style	15	2	2	1	2	1	2	6	1	3	5	3.63
Internet Experience	9	9	4	6	4	4	5	11	4	15	20	8.27
Computer Experience	9	19	26	32	27	18	18	22	21	21	22	21.36
UP2												
Gender	4	7	7	9	9	7	6	9	7	10	8	7.58
Cognitive Style	3	6	8	8	10	8	4	14	5	11	13	8.40
Internet Experience	4	3	2	3	4	3	3	4	3	2	1	2.91
Computer Experience	6	6	5	2	4	3	4	6	4	6	12	5.27

Table 3.4. Number of Features Selected by Individual Classifiers for Each Human Factor

The results from Table 3.4 showed that the UP1 and UP2 datasets have different target variables. In addition, it was shown that these two datasets have different number of input instances and features. A summary of the number of input instances, input features and target variables for these two datasets is shown in Table 3.5.

	No. of Instances	No. of Features	Target Variable
UP1	120	90	Users' Level of Computer Experience (little; average; good; excellent)
UP2	65	20	Users' Cognitive Style (Field Independent; Field Dependent; Intermediate)

Table 3.5. Summary of Characteristics of UP1 and UP2 Datasets

3.5 Conclusion

Wrapper feature selection approaches use a single classifier to select the subset of most relevant features. However, there is a problem with using a single classifier to do the feature selection. The problem is that each classifier is different because it will be of a different nature and possess different biases. This means that each classifier will select a different subset of relevant features. In order to overcome this problem of using a single classifier, this chapter proposed the Wrapper-based Decision Trees (WDT) method. The WDT method is a novel data mining method that merges consensus feature selection and decision tree construction. The novelty of the WDT lies within its ability to obtain a consensus among several different classifiers in order to reduce the biases of using single classifier so that an unbiased and mutually agreed set of relevant features are selected, and also construct decision trees to visualise and highlight the most important relationships between the selected features.

Since the WDT method combines multiple classifiers, the selection of classifiers for use with the method is crucial. In fact, the number of classifiers used and the nature of classifiers used may play a significant role in influencing the feature selection results. These two issues are very important and will be investigated using WDT throughout the

rest of the thesis. In addition to WDT, these two issues will be investigated using: 1) two approaches that consider different classifier arrangements and 2) two different types of datasets. The two arrangement approaches used are termed same-type approach and mixed-type approach. The same-type approach involves combining classifiers from the same family to do the feature selection with WDT. The mixed-type approach, however, involves combining classifiers from different families. These two approaches will help us fully understand the role of number and nature of classifiers in feature selection. Along with these approaches, two types of user preference datasets (UP1 and UP2) derived from field of HCI are used. Such datasets were chosen because of the high level of uncertainty associated with the users' preferences, which can often lead to noise and irrelevant features in the data, and thus are suitable for use with WDT method. The WDT method along with the classifier approaches and datasets, which make up the entire process involved in investigating role of number and nature of classifiers in feature selection, are illustrated in Figure 3.3.

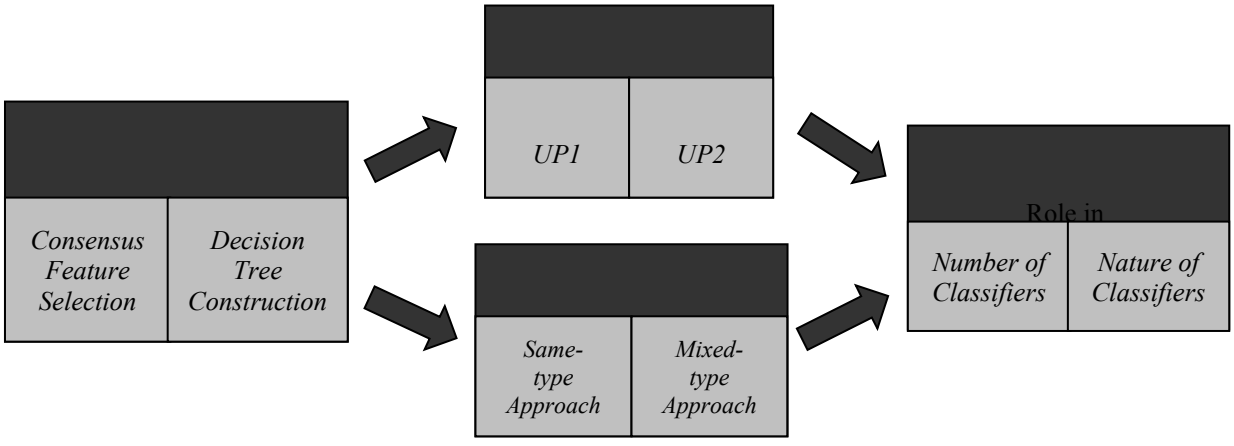

Figure 3.3. Summary of Entire Process of Thesis Investigation

The next chapter uses the entire process shown in Figure 3.3 to begin investigating the role of number of and nature of classifiers in feature selection. The chapter will employ the WDT method with both datasets in addition to the first of the two classifier arrangement approaches mentioned; the same-type approach.

Chapter 4 – The Combinations of Same-Type Classifiers

4.1 Introduction

A novel data mining method termed Wrapper-based Decision Trees, or WDT, was described in the previous chapter. The novelty of the WDT method lies within its ability to integrate the use of the Wrapper feature selection approach and decision tree classifiers. The WDT combines multiple classifiers to select relevant sets of features with the Wrapper and visualises the selected relevant features by constructing decision trees. The purpose of combining multiple classifiers for feature selection is to reduce the biases of each individual classifier. However, as already discussed in Chapter 3, the selection of classifiers can affect the feature selection results. In particular, the number of classifiers used and the nature of classifiers used may influence the number of relevant features selected and also the accuracy levels of the selected features. In terms of number of classifiers, decreasing or increasing the number of classifiers used for feature selection may have effects on the number of features selected. Varying the number of classifiers used may also have an effect on the accuracies of the features, i.e., how relevant the features are. In terms of nature of classifiers, there are many different types of classifiers, each of which has its own assumption and bias. The bias of a classifier may lead to differences in the number of features selected and also variations in the accuracies of features. As such, the nature of a classifier also has the potential of influencing the features selected.

The number of classifiers and the nature of classifiers are thus two important issues that may affect the way in which features are selected. In this thesis, the effects of both number and nature of classifiers are investigated using two different approaches. The first approach is the same-type approach and the second approach is the mixed-type approach. With regards to the former approach, classifiers belonging to the same classifier family (i.e., classifiers that have the same nature) are combined with the WDT method. For example, classifiers belonging to the DT family may be combined together to form such combinations. With regards to the latter approach, classifiers belonging to different classifier families (i.e., classifiers that have different nature) are combined

with the WDT method. For example, classifiers belonging to DT family and classifiers belonging to BN family may be combined together to form such combinations. In addition, these approaches also make use of different numbers of classifiers. This implies that both the number and nature of classifiers can be investigated by these two approaches.

This chapter will address the same-type approach. The same-type approach will be examined with the two different types of datasets that were described in the previous chapter. The two chosen datasets are derived from the field of HCI, where one dataset comprises of users preferences of Web search engines and the other dataset comprises of users' preferences of a Web-based learning system. The former dataset will be referred to as UP1 hereafter whereas the latter will be referred to as UP2. We will present the results from each dataset in terms of the number of features selected by the same-type combinations and the accuracy levels of the selected features. Decision trees formed using the features selected from these datasets will also be examined in order to better understand the relationships between the selected relevant features.

The chapter is organised as follows. It will start in Section 4.2 by giving a brief description of what classifiers are used to select the relevant features and how these classifiers are combined with WDT. Subsequently, Section 4.3 presents the results from same-type combinations regarding the number of relevant features selected and the accuracy levels generated by the selected features. Relationships between the number of features selected and the accuracy levels generated by combinations will also be presented. Finally, Section 4.4 presents an analysis of the decision trees with the highest levels of accuracies for both UP1 and UP2 datasets.

4.2 Same-type Classifier Combinations

As described in the previous section, this chapter investigates the influences of both number and nature of classifiers using the same-type approach. The same-type approach involves combining various classifiers that belong to the same classifier family. In this thesis, the same-type combinations use classifiers from four different families, namely the Bayesian Network (BN) family, Decision Tree (DT) family, Nearest Neighbour

(NN) family, and Support Vector Machine (SVM) family. More precisely, three classifiers from the BN, DT and NN families are used to do the feature selection with WDT, and one classifier is used from the SVM family. The chosen SVM classifier will be used with two different kernels, namely the polynomial kernel and radial basis function kernel (Vapnik, 1998). In this way, features in the datasets that relate to both linear aspects and non-linear aspects can be identified. All chosen classifiers belonging to the BN, DT, NN, and SVM families are summarised in Table 4.1.

BN Family	DT Family	NN Family	SVM Family
Bayesian Network Classifier (BNC) (Friedman, Geiger and Goldszmidt, 1997)	C4.5 (Quinlan, 1993)	Nearest Neighbour Classifier (NNC) (Dasarthy, 1991)	SVM with Polynomial and Radial Basis Function Kernel (Vapnik, 1995; 1998)
Naïve Bayes (NB) (Langley and Sage, 1994)	CART (Breiman et al., 1984)	*k*-Nearest Neighbor (KNN) (Cover and Hart, 1967)	
Average One Dependence Estimator (AODE) (Webb, Boughton, and Wang, 2002)	CN2 (Clark and Niblett, 1989)	K* (Cleary and Trigg, 1995)	

Table 4.1. Classifiers Belonging to Each Classifier Family

The classifiers shown in Table 4.1 were chosen for two main reasons. First, previous studies have shown that classifiers from these four families are suitable for analysing a variety of datasets, especially user preference datasets. For example, Kritikou, et al. (2008) developed an e-Learning system that utilised the BNC, which belongs to BN family, to capture users' preferences and overall behaviour of the system. The developed system was capable of adapting to each user's specific preferences. Liu and Kešelj (2007) proposed a method that used the C4.5 DT classifier to automatically classify users' Web navigation patterns and preferences to help predict which Web pages are more likely to be accessed next by users. In addition, Jian, Jian, and Jin (2005) developed an e-Commerce recommender system that used KNN, which belongs to the NN family, to automatically recommend new products whose characteristics match customers' preferences and interests. Finally, Bo and Luo (2007) proposed a personalised Web information recommendation algorithm that uses the SVM classifier. The algorithm was able to identify users' preferences and apply these preferences to a personalised information recommendation service so as to suit the information needs of

different users. Second, the chosen classifiers are probably the most popularly used classifiers for representing the nature and biases of the four different classifier families shown in the table. Using classifiers from these different families will help select different features that can then be combined with WDT to see their influences on feature selection results.

The classifiers mentioned shown in Table 4.1 were combined to analyse both UP1 and UP2 datasets. The classifiers were combined using an exhaustive approach so that each classifier was used with every other classifier within a same family. This exhaustive approach led to the construction of 2-classifier same-type combinations (e.g., BNC+NB, BNC+SVMpoly, BNC+SVMrbf), 3-classifier same-type combinations (e.g., BNC+NB+AODE, BN+NB+SVMpoly, BN+NB+SVMrbf), and 4-classifier same-type combinations (e.g., BNC+NB+AODE+SVMpoly and BNC+NB+AODE+SVMrbf). As shown in these examples, the SVM classifier is included in the combinations. The SVM classifier was included in the same-type combinations mainly because it is well known for its highly accurate performance and excellent generalisation ability on many different types of datasets (Chen and Hsieh, 2006; Barakat and Bradley, 2007). In this way, we cannot only investigate the influences of same-type combinations on feature selection, but also we can see if the inclusion and exclusion of the SVM classifier influences the feature selection results of the combinations.

In total, 54 same-type classifier combinations were formed in the manner described above, including 27 for 2-classifiers (9 without SVM, 9 with SVMpoly, and 9 with SVMrbf), 21 for 3-classifiers (3 without SVM, 9 with SVMpoly, and 9 with SVMrbf), and 6 for 4-classifiers (3 with SVMpoly, and 3 with SVMrbf). These 54 same-type combinations will be used to select relevant features from both the UP1 and UP2 datasets. The features selected by the combinations will subsequently be used to build decision trees to visualise the interactions between features and determine the classification accuracies of the selected features. The classification accuracies will give an indication of how relevant each of the different sets of features is in relation to the target variable. In this case, high classification accuracy implies that the selected

features are very relevant to the target variable whereas low classification accuracy implies that the features are not so relevant.

The analysis of the number of relevant features selected by the same-type combinations and their associated classification accuracies for both UP1 and UP2 datasets are carried out in the next section.

4.3 Results from UP1 and UP2 Datasets

In terms of the UP1 dataset, the combinations employed the users' level of computer experience as the target variable and their responses to the 90 questions of the questionnaire as the input features. In terms of the UP2 dataset, the same combinations adopted the user's cognitive style as the target variable and their responses to the 20 questions regarding the Web-based learning as input features. These two datasets used different target variables because they were previously shown (in Chapter 3) to select highest number of relevant features from the dataset. The fact that they selected higher number of features compared to other possible target variables increases the likelihood of identifying features that are of highest relevance to the target variable.

The number of relevant features chosen by each classifier combination, as shown in parentheses next to each combination, and the classification accuracies generated using the selected relevant features for both UP1 and UP2 are shown in Table 4.2 and Table 4.3 respectively. Each of the tables presents the classification accuracies generated by the same-type combinations from the lowest accuracies, which appear on the left side of the tables, to the highest accuracies, which appear on the right side of the tables. This helps identify the same-type classifier combinations which are least suitable and most suitable for identifying relevant feature subsets.

The mean number of features selected by the 2-classifier, 3-classifier and 4-classifier combinations and the mean accuracy levels generated by these combinations for the datasets are also shown in Table 4.4. In particular, Table 4.4 includes: 1) mean number of features selected and accuracies generated by combinations without SVM classifiers, 2) mean number of features selected and accuracies generated by combinations with

SVMpoly, 3) mean number of features selected and accuracies generated by combinations with SVMrbf, and 4) the overall number of features selected and overall accuracies generated by the 2-classifier, 3-classifier and 4-classifier combinations. This table will help us determine how combinations with and without the SVM classifier influence feature selection results in UP1 and UP2 datasets.

	Same-type Classifier Combinations for UP1 Dataset			
	Lowest Accuracies between 72.50 to 78.33% (N=8)	***Intermediate Accuracies between 80 to 84.17% (N=33)***		***Highest Accuracies between 85 to 88.33% (N=13)***
2-classifier	BNC+NB (12) BNC+SVMpoly (14) BNC+SVMrbf (17) NB+SVMrbf (17) KNN+K* (14) KNN+SVMpoly (12) KNN+SVMrbf (13)	BNC+AODE (18) NB+SVMpoly (17) AODE+SVMpoly (22) AODE+SVMrbf (21) C4.5+CART (22) C4.5+CN2 (22) CART+CN2 (22) C4.5+SVMpoly (22) C4.5+SVMrbf (23)	CART+SVMpoly (18) CART+SVMrbf (18) CN2+SVMpoly (20) CN2+SVMrbf (21) NNC+K* (13) K*+SVMpoly (17) K*+SVMrbf (18)	NB+AODE (20) NNC+KNN (10) NNC+SVMrbf (12) NNC+SVMpoly (15)
3-classifier	KNN+K*+SVMrbf (19)	BNC+NB+SVMrbf (15) BNC+AODE+SVMpoly (16) BNC+AODE+SVMrbf (16) NB+AODE+SVMpoly (23) NB+AODE+SVMrbf (25) C4.5+CART+SVMrbf (18) C4.5+CN2+SVMpoly (20) C4.5+CN2+SVMrbf (19) CART+CN2+SVMrbf (16)	NNC+KNN+SVMpoly (13) NNC+KNN+SVMrbf (14) KNN+K*+SVMpoly (18)	BNC+NB+AODE (10) BNC+NB+SVMpoly (13) C4.5+CART+CN2 (18) C4.5+CART+SVMpoly (18) CART+CN2+SVMpoly (15) NNC+KNN+K* (17) NNC+K*+SVMpoly (16) NNC+K*+SVMrbf (17)
4-classifier	-	BNC+NB+AODE+SVMpoly (10) BNC+NB+AODE+SVMrbf (11) C4.5+CART+CN2+SVMrbf (15) NNC+KNN+K*+SVMpoly (11) NNC+KNN+K*+SVMrbf (11)		C4.5+CART+CN2+SVMpoly (13)

Table 4.2. No. of Features Selected by Same-type Combinations and Associated Classification Accuracy Levels for UP1 Dataset.

	Same-type Classifier Combinations for UP2 Dataset			
	Lowest Accuracies of 90.77% and 92.31% (N=12)	***Intermediate Accuracies of 93.85% and 95.38% (N=40)***		***Highest Accuracies of 96.92% (N=2)***
2-classifier	BNC+SVMpoly (5) BNC+SVMrbf (7) AODE+SVMrbf (8) CART+SVMrbf (10) NNC+SVMrbf (6) K*+SVMrbf (6)	BNC+NB (3) BNC+AODE (4) NB+AODE (3) NB+SVMpoly (4) NB+SVMrbf (9) AODE+SVMpoly (4) C4.5+SVMrbf (9) C4.5+CART (10) C4.5+CN2 (6) C4.5+SVMpoly (7) CART+CN2 (10)	CART+SVMpoly (8) CN2+SVMpoly (6) CN2+SVMrbf (10) NNC+KNN (10) NNC+K* (2) NNC+SVMpoly (5) KNN+K* (10) KNN+SVMpoly (14) KNN+SVMrbf (16) K*+SVMpoly (4)	-
3-classifier	BNC+NB+SVMrbf (5) C4.5+CART+SVMrbf (10) C4.5+CN2+SVMrbf (8) NNC+KNN+SVMrbf (12) KNN+K*+SVMrbf (11)	BNC+NB+SVMpoly (3) BNC+NB+AODE (3) BNC+AODE+SVMpoly (3) BNC+AODE+SVMrbf (5) NB+AODE+SVMpoly (4) NB+AODE+SVMrbf (8) C4.5+CART+CN2 (7) C4.5+CART+SVMpoly (9) C4.5+CN2+SVMpoly (6)	CART+CN2+SVMpoly (8) CART+CN2+SVMrbf (9) NNC+KNN+SVMpoly (8) NNC+K*+SVMrbf (3) KNN+K*+SVMpoly (8)	NNC+KNN+K* (4) NNC+K*+SVMpoly (4)
4-classifier	NNC+KNN+K*+SVMrbf (4)	BNC+NB+AODE+SVMpoly (3) BNC+NB+AODE+SVMrbf (2) C4.5+CART+CN2+SVMrbf (9) C4.5+CART+CN2+SVMpoly (7) NNC+KNN+K*+SVMpoly (4)		-

Table 4.3. No. of Features Selected by Same-type Combinations and Associated Classification Accuracy Levels for UP2 Dataset.

	Mean Number of Features & Mean Accuracy Levels for UP1							
	Combinations Without SVM		*Combinations With SVMpoly*		*Combinations With SVMrbf*		*Overall*	
2-classifier	17.29	81.57	17	81.09	18.11	80.65	17.41	81.08
3-classifier	16.21	86.94	15.94	84.26	17.67	81.57	16.33	83.50
4-classifier	-	-	11.33	84.17	12.33	80.28	11.83	82.22
	Mean Number of Features & Mean Accuracy Levels for UP2							
	Combinations Without SVM		*Combinations With SVMpoly*		*Combinations With SVMrbf*		*Overall*	
2-classifier	6.46	94.42	6.33	94.81	7.33	92.48	7.26	93.94
3-classifier	5.73	95.18	5.41	95.89	7.88	93.34	6.57	94.36
4-classifier	-	-	4.67	94.87	5	93	4.83	94.10

Table 4.4. Mean No. of Features selected and Mean Accuracy Levels for UP1 and UP2

An initial inspection of the results presented in Tables 4.2, 4.3 and 4.4 shows differences among combinations with SVMpoly and combinations with SVMrbf. It was found that using combinations with SVMrbf resulted in slightly higher mean numbers of relevant features being selected compared to using combinations with SVMpoly. This was found in both UP1 and UP2 datasets. This implies that combinations with SVMrbf helped identify more features overall than combinations with SVMpoly, irrespective of the dataset used. An explanation for this finding may have to do with the nature of the polynomial and radial basis function kernels. The SVM classifier with the polynomial kernel seeks to identify linear aspects within datasets whereas the SVM with the radial basis function seeks non-linear in addition to linear aspects (Cristianini and Shawe-Taylor, 2000). Such a difference may explain why combinations which used the radial basis function kernel were able to identify a higher number of features. It may be that combinations comprised of SVM with radial basis function kernel were able to identify more features because they were able to select features that related to both non-linear and linear aspects in the data, rather than just linear aspects.

Furthermore, it was found in both UP1 and UP2 that using combinations with SVMrbf resulted in significantly lower mean classification accuracies than using combinations with SVMpoly. This means that the same-type combinations which used SVMrbf generated lower levels of accuracies than combinations which used SVMpoly. This finding suggests that the former selected features that were less relevant with respect to the target variable than the latter. The reason for such a finding may lie within the features selected by the kernels used by SVM. As

previously mentioned, the SVMpoly seeks to identify features relating to linear aspects within datasets whereas the SVMrbf mainly seeks features relating to non-linear aspects. It may have been the case that the features selected by SVMrbf, which related to non-linear aspects, were of little relevance or totally irrelevant to the target variables of the datasets. Selecting such irrelevant features may lead to low levels of accuracy, and may therefore explain why SVMrbf led to lower accuracies than SVMpoly.

Since combinations with SVMrbf selected features that led to lower accuracy levels than combinations with SVMpoly, the rest of this chapter will disregard the results from combinations with SVMrbf and instead focus on the results from the combinations with SVMpoly. The rest of this chapter will thus focus on the results from the same-type combinations including those combinations with the SVMpoly classifier for both UP1 and UP2 datasets.

4.3.1 Number of Relevant Features

This subsection presents the results of the influences of number and nature of classifiers on the number of relevant features selected by the same-type combinations from the UP1 and UP2 datasets. Three key results were found regarding these datasets, each of which is described in the following pages.

1) 2-classifier Combinations Select More Relevant Features Than 3-classifier and 4-classifier Combinations

On close examination of Table 4.2, Table 4.3 and Table 4.4, it seems that combinations with few classifiers selected more relevant features than combinations with many classifiers. More specifically, the 2-classifier combinations identified more relevant features than the 3-classifier and 4-classifier combinations. In order to see this more clearly, we analyse the mean number of features selected by these combinations for both datasets. In terms of UP1 dataset, the mean number of features selected by 2-classifier combinations (17.11) was higher than the mean number of features selected by 3-classifier (16.42) and 4-classifier combinations (11.33). In terms of UP2 dataset, the mean number of features selected by 2-classifier combinations (6.28) was also found to be higher than the mean number of features selected by 3-classifier (5.67) and 4-classifier combinations (4.67). In addition to

mean number of features selected, a further analysis was carried out to identify further differences in the number of features selected by the classifier combinations. The further analysis involved ranking the number of features selected by the classifier combinations from highest number (top half) to lowest number (bottom half). In this way, we will be able to see which combinations selected higher and lower number of features. The number of classifier combinations that appeared in the top half and bottom half of the ranking for UP1 and UP2 is shown in Table 4.5.

Combinations	**No. of combinations in top half of ranking (N=17)**		**No. of combinations in bottom half of ranking (N=16)**	
	UP1	*UP2*	*UP1*	*UP2*
2-classifier	11	10	7	8
3-classifier	6	5	6	7
4-classifier	-	1	3	2

Table 4.5. No. of Combinations that Appear in Top Half and Bottom Half of Ranking of Number of Features Selected for UP1 and UP2

The results from Table 4.5 show that nearly all of the 2-classifier combinations appeared in the top half of the ranking for both UP1 and UP2 datasets. However, in general, 3-classifier and 4-classifier combinations appeared in the bottom half of the ranking. These results show that 2-classifier combinations were able to select a higher number of relevant features than 3-classifier and 4-classifier combinations. A possible explanation for differences in number of features selected may lie within the fact that the median strategy was used to combine the results from the different classifiers. The median strategy computes the middle value for a given set of values. In this study, the median was calculated for every feature selected by each combination. To better understand the way in which the median strategy was used, the BNC+C4.5 2-classifier combination will be used as an example for explanation. For a feature, *f*, BNC may show a relevance value of 2, while C4.5 may not find the feature relevant (i.e., 0 is assigned). In this example, the median of *f* will be 1. This feature will therefore be classified as a relevant feature because any feature with a median of 1 or more is considered relevant by WDT.

This example shows that features with low relevance levels have the ability of being classed as relevant, which, in turn, results in more relevant features being identified. On the other hand, using three or four classifiers results in fewer relevant features being identified. This may be due to the fact that each feature will have more

relevance values. When using 3-classifier combinations each feature will have three associated relevance values. For a feature to be regarded as relevant, at least two of the three relevance values must be 1 or more so that the resulting median is above 1. This same concept also applies to 4-classifier combinations, but with the exception that three out of the four relevance values for each feature must be 1 or more for a feature to be considered as relevant. Overall, this shows that fewer relevant features can be identified when more than two classifiers are used because more classifiers are required to (mutually) agree on a particular feature. In other words, the number of classifiers used for feature selection influences the number of relevant features selected from both datasets.

2) Combinations With DT Family Classifiers Select More Relevant Features Than Other Combinations.

The number of relevant features selected by the same-type combinations was also influenced by combinations with classifiers from the DT family. Combinations that included DT family classifiers selected more relevant features overall than combinations with classifiers from the other families. This finding was observed in both UP1 and UP2 datasets. In order to see this finding more clearly, we consider the mean number of relevant features selected by combinations that include each classifier family for the UP1 and UP2 datasets. The mean number of features selected by each classifier family is shown in Table 4.6.

Mean number of features selected	**Combinations with BN family classifiers**		**Combinations with DT family classifiers**		**Combinations with NN family classifiers**	
	UP1	*UP2*	*UP1*	*UP2*	*UP1*	*UP2*
2-classifier combinations	17.17	3.83	21	7.50	13.17	7.50
3-classifier combinations	15.50	3.50	17.67	7.50	10.67	6
4-classifier combinations	10	3	13	7	11	4

Table 4.6. Mean Number of Features Selected by Different Classifier Families

In general, Table 4.6 shows that combinations with DT family classifiers selected a higher number of relevant features than combinations with BN and NN family classifiers. In addition to this table, a further analysis was conducted with both datasets in order to better understand the differences in the number of relevant

features selected by each classifier family. For each of the classifier families, i.e., BN, DT and NN family, we analyse the number of combinations that selected features above the total mean number of features and number of combinations that selected features below the total mean number of features. The total mean number of features for UP1 is 16.33 and the total mean number of features for UP2 is 5.91. Furthermore, we take into account the number of combinations that appeared in the top half (i.e., those that selected a high number of features) and bottom half (i.e., those that selected a low number of features) of the ranked list, where the ranked list represents number of features selected by all combinations sorted from highest to lowest. The results from this analysis are shown in Table 4.7.

	No. of combinations above total mean number of features		**No. of combinations below total mean number of features**		**No. of combinations that appear in top half of ranked list**		**No. of combinations that appear in bottom half of ranked list**	
	UP1	*UP2*	*UP1*	*UP2*	*UP1*	*UP2*	*UP1*	*UP2*
Combinations with DT Classifiers (N=11)	9	10	2	1	9	11	2	-
Combinations with BN Classifiers (N=11)	5	-	6	11	5	-	6	11
Combinations with NN Classifiers (N=11)	4	5	7	6	3	5	8	6

Table 4.7. Comparison of DT Family, BN Family and NN Family Combinations

The results from Table 4.7 show that nearly all of the combinations with DT family classifiers selected number of features much higher than the total mean number of features selected, and appeared in the top half of the ranked list. Once again, this shows that combinations with classifiers from the DT family were able to identify more relevant features than combinations with BN and NN family classifiers. The reason why classifiers belonging to the DT family identified more relevant features than classifiers belonging to the BN and NN families may have to do with the nature of these three types of classifiers. On the one hand, the nature of BN classifiers allows them to identify relationships between features which are typically presented in a graphical network structure (Grossman and Domingos, 2004). However, the number of relationships, and thus the number of features, identified may be affected by the

amount of prior knowledge available about the actual features in the dataset. In the event that prior knowledge is available, the number of relationships and the number of features selected may increase because there are more details concerning the features and the interactions among the features. On the contrary, the number of features may decrease when no or little prior knowledge is available. In essence, this means that prior knowledge may influence the number of features used in the network structure and thus the number of features selected by BN family classifiers (Niculescu, Mitchell, and Rao, 2006).

On the other hand, the nature of classifiers belonging to the NN family allows them to determine relevant features by using some distance metric, where the most relevant features are those that are deemed the closest by distance metric. However, the number of most relevant features identified greatly depends on a small number of features called neighbours (Ghosh, 2006). This is because the number of neighbours employed represents number of features used to determine the most relevant (i.e., closest) features. For example, using few neighbours may lead to fewer features being selected whereas more neighbours may result in more features being identified. Thus, using an inappropriate number of neighbours may lead to fewer features identified (Han and Kamber, 2006).

Conversely, the nature of DT classifiers is very different to the nature of BN and NN classifiers described above. Basically, DT classifiers use a statistical measure to determine the relevance of features, where the most relevant features are those with the highest statistical relevance values (Nikovski and Kulev, 2006). More importantly, the nature of DT classifiers does not require them to have prior knowledge about the features in the data (like BN classifiers) and does not require them to rely on a predetermined number of neighbours for finding the relevant features (like NN classifiers). The fact that DT classifiers do not need to deal with such issues, which can considerably decrease number of features selected suggests that combinations with DT classifiers are more likely to identify a higher number of relevant features than combinations comprising BN and NN classifiers. This may thus explain why the combinations with DT classifiers selected more relevant features than combinations with BN and NN classifiers.

3) Combinations with SVM Classifier Select Lower Number of Features

Closely examining the results from UP1 and UP2 revealed another interesting finding with respect to combinations comprised of the SVM classifier. In total, there were 21 same-type combinations that included the SVM classifier. The results from UP1 show that 11 of these 21 combinations selected number of features lower than the total mean number of relevant features selected from UP1, and the same number of combinations appeared in the bottom half of the ranking. These results from UP1 show that combinations with SVM selected a low number of features. The results from UP2 also show similar finding. The results from UP2 show that 12 of the 21 combinations selected fewer features compared to the total mean number of features. In addition, 12 of the 21 combinations with SVM appeared in the bottom half of the ranking. A closer look at Table 4.4, which presents the mean number of features selected by combinations with and without SVM classifier in both UP1 and UP2, also corroborates the fact fewer features were selected by combinations with SVM.

The findings from UP1 and UP2 suggest that including the SVM classifier in same-type combinations somewhat reduces the number of relevant features selected. The cause for this may lie within the nature of classifiers used in the same-type combinations. Same-type combinations predominantly include classifiers from one classifier family. This means that these classifiers will have very similar biases and thus select very similar features. However, when classifiers with the same biases are combined with the SVM classifier the results may be different. This is because the SVM classifier belongs to a very different classifier family and thus has very different biases (Wu, Huang and Meng, 2008). Because of these differences, there may be less agreement among the SVM classifier and the other family of classifiers. Lower agreement among the classifiers may result in fewer relevant features being selected, which may therefore explain why combinations with SVM selected fewer features than combinations without the SVM.

In summary, the abovementioned results have shown that the number of classifiers and nature of classifiers influence the number of relevant features identified by same-

type combinations. More specifically, it was found that combinations with few classifiers are able to identify a higher number of relevant features than combinations with many classifiers. In addition, the results also showed that combinations with DT family classifiers selected a higher number of features than combinations with other classifier families. The number and nature of classifiers thus influence number of features selected. The next section determines the effects of the number and nature of classifiers on the classification accuracies of the selected features.

4.3.2 Accuracy Levels of Relevant Features

This section presents the results regarding the influences of number and nature of classifiers on the classification accuracies. Analysis of the accuracies from both UP1 and UP2 datasets revealed several findings, each of which will be explained.

1) 3-classifier Combinations Generate Highest Classification Accuracies

The results from both UP1 and UP2 dataset revealed that 3-classifier combinations were able to generate accuracies much higher than the majority of the other combinations. In terms of the UP1 dataset, 3-classifier combinations were found to generate a mean accuracy level (84.93%) higher than the mean accuracy levels of 2-classifier (81.30%) and 4-classifier (84.16%) combinations. In terms of the UP2 dataset, the mean accuracy level of 3-classifier combinations (95.38%) was also found to be higher than the mean accuracy levels of 2-classifier (94.69%) and 4-classifier (94.87%) combinations.

A further analysis of the accuracies from the UP1 and UP2 datasets was additionally carried out so as to obtain a better understanding of the accuracies from the 3-classifier combinations. The analysis for UP1 dataset revealed that 9 of the 12 3-classifier combinations generated accuracies higher than total mean accuracy of all same-type combinations, which was found to be 82.88%. In addition, ranking the accuracies of all combinations from highest (top) to lowest (bottom) showed that 9 of the 12 combinations were in the top half of the ranking. The analysis for the UP2 dataset uncovered that 10 of the 12 3-classifier combinations generated accuracies higher than the total mean accuracy of all combinations used with UP2 (94.95%), and the same number of combinations were also found to be in the top half of the accuracy

ranking. The aforementioned findings suggest that 3-classifier combinations were able to produce higher accuracy levels than other combinations. In other words, features selected by the 3-classifier combinations are more relevant in relation to the target variable than the other classifier combinations.

2) Combinations with DT Family Classifiers Show a Higher Level of Classification Accuracy

It was found in both UP1 and UP2 datasets that combinations with the DT family classifiers generally led to higher accuracies than those with classifiers from BN and NN families. A further analysis of accuracies generated by each family of classifiers was carried out to show any differences. The results of this analysis are shown in Table 4.8.

	Mean accuracies generated by combinations with BN Family Classifiers		**Mean accuracies generated by combinations with DT family classifiers**		**Mean accuracies generated by combinations with NN family classifiers**	
	UP1	*UP2*	*UP1*	*UP2*	*UP1*	*UP2*
2-classifier combinations	80.55	94.36	82.22	94.87	81.11	94.87
3-classifier combinations	84.80	94.99	85.83	95.38	84.17	95.38
4-classifier combinations	83.33	93.85	85	95.38	84.17	95.38

Table 4.8. Classification Accuracies Generated by Each Classifier Family

In terms of UP1, Table 4.8 shows that 3-classifier combinations that made use of DT family classifiers generated higher accuracies than combinations with the other family classifiers. It also shows that combinations with DT family classifiers selected feature subsets that were more relevant with respect to the target variable. In terms of UP2, the results from Table 4.8 show a very interesting finding. It is shown that combinations with DT family classifiers and combinations with NN family classifiers generated higher accuracies than combinations with BN family classifiers. In fact, the accuracies generated by combinations with DT classifiers were identical to the accuracies of combinations with NN classifiers.

The reason why combinations with these two classifier families generated identical levels of accuracies may lie within similarities in their nature. Classifiers belonging to the DT family and NN family determine relevant features in a rather similar manner.

On the one hand, DT family classifiers determine the relevance of features according to the feature's position in the decision tree. For example, the feature that appears at the root (top) of the tree is deemed the most relevant to the target variable, while those on subsequent (lower) levels are deemed less relevant. On the other hand, NN family classifiers determine the relevance of the features according to distance, where the most relevant feature will be closest and the least relevant feature will be furthest away. This suggests that DT family classifiers and NN family classifiers use a fairly similar approach to determine the relevance of features. In this way, the two classifier families may select similar features that have similar levels of relevance. Selecting similar features with comparable relevance levels may lead to very similar levels of accuracy being generated, and may therefore explain why combinations with the DT and NN family classifiers were able to select features which generated identical accuracy levels.

In summary, the results from both UP1 and UP2 datasets showed a common result, which was that combinations with DT family classifiers generated accuracies that were in most cases higher than the other combinations. A possible explanation for this result may lie within the nature of DT classifiers. DT classifiers are sometimes regarded as another type of feature selection method called 'embedded methods' (Guyon and Elisseeff, 2003). Basically, embedded methods like DT classifiers are able to perform feature selection (Perner and Apte, 2004; Sugumaran, Muralidharan, and Ramachandran, 2007). This means that DT classifiers have the ability to select relevant features on their own. To be precise, DT classifiers use 1) a search strategy to search for potentially relevant features and 2) the splitting criterion they typically utilise in order to determine the relevance of features. The relevant features (i.e., features that satisfy the splitting criterion) are then used by the classifier to form a hierarchical tree structure, which helps with the classification of the data. Including classifiers with these abilities in WDT combinations suggests that they will select relevant features at two different stages. In the first stage, features are selected by the individual DT classifiers and in the second stage features are selected by combinations comprised of the DT classifiers. In this manner, only the features that are selected at both of these stages will form the final feature subset, which is very likely to include features of high relevance. Uncovering highly relevant features with respect to the target variable increases the likelihood of obtaining higher classification

accuracies. This may thus explain why combinations with DT family classifiers on the whole generated higher accuracy levels than other combinations.

3) Combinations with SVM Classifier Generate Different Accuracy Levels for UP1 and UP2

An interesting finding was found in the UP1 and UP2 datasets regarding combinations with the SVM classifier. According to the results from UP1, 11 of the 21 combinations with SVM generated accuracies lower than the total mean accuracy level. The same number of combinations was also found in the bottom half of the ranked accuracy list. The results from UP1 suggest that combinations with SVM classifier generally led to low levels of accuracy. However, the results from UP2 show a different aspect. The results from UP2 show that 14 of the 21 SVM combinations generated accuracies higher than the total mean accuracy of all same-type combinations, and the same number of combinations appeared in the top half of the ranked accuracy list. The results from UP2 suggest that combinations with SVM generated high levels of accuracy.

The results from UP1 and UP2 regarding influences of SVM classifier on accuracy levels are rather different. On the one hand, the results from UP1 are not clear enough to demonstrate the influences of SVM classifier on accuracy levels. This was shown by the fact that the number of combinations above/below total mean accuracy level and number of combinations in top/bottom half of accuracy ranking are practically equal. On the other hand, the results from UP2 clearly show how combinations with SVM influence the accuracy levels generated. This difference in clarity of results from UP1 and UP2 was also found in the previous section (Section 4.3.1). The previous section found that combinations with SVM classifier led to fewer relevant features being selected. However, results from UP2 showed the effect of SVM on number of features selected clearer than the results of UP1.

The rationale behind such differences may have to do with the size of the feature subsets selected from UP1 and UP2. On the one hand, same-type combinations selected large feature subsets (i.e., feature subsets that contain a high number of features) from the UP1 dataset. When dealing with large feature subsets, the influences of nature of classifiers on feature selection results may not be so clear because the presence of many features may mask the true influences of the classifier.

In the case of UP1, the influences of nature of SVM may have been masked by the high number of features selected. On the other hand, the same-type combinations selected small feature subsets (i.e., feature subsets that contain a low number of features) from UP2. When small feature subsets are considered, the influences of nature of classifiers may be clearer because there are fewer features present to hide the effects of a particular classifier. As such, the influence of nature of SVM classifier on accuracy levels may be more apparent when small feature subsets are selected. This may help explain the different findings obtained from the UP1 and UP2 datasets.

4.3.3 Relationships Between Number of Features Selected and Accuracy Levels of Features

The two abovementioned sections have shown that the number and nature of classifiers used in same-type combinations influence the number of relevant features selected (Section 4.3.1) as well as the accuracy levels of selected features (Section 4.3.2). When considered individually, the number of features selected and the accuracy levels of features show some interesting findings. However, little is known about the relationships that may exist between the number of features selected and the accuracy levels of the features. In this vein, this section considers the number of features selected by each same-type combination and the associated accuracy levels.

	Accuracy Levels Generated By Combinations		
	Lowest	*Intermediate*	*Highest*
2-classifier	13	18.10	15
3-classifier	-	18	12.71
4-classifier	-	10.50	13
Total	13	46.60	30.71

Table 4.9. Number of Features Selected for Classification Accuracies in UP1

	Accuracy Levels Generated By Combinations		
	Lowest	*Intermediate*	*Highest*
2-classifier	5	6.50	-
3-classifier	-	6	4
4-classifier	-	4.70	-
Total	5	17.20	4

Table 4.10. Number of Features Selected for Classification Accuracies in UP2

To identify potential relationships between these two issues, two tables are used. The first one, Table 4.9, shows the mean number of relevant features selected by combinations which generated lowest, intermediate and highest accuracy levels for UP1 dataset. The second one, Table 4.10, presents the mean number of relevant

features selected by combinations which generated lowest, intermediate and highest accuracy levels for UP2 dataset. By using these two tables, we will be able to see how the number of features selected changes as the accuracy levels change, or vice versa. A close examination of both tables reveals a common finding among UP1 and UP2. The finding relates to the fact that combinations which generated the lowest levels of accuracies were found to select a low number of features. In addition, it was also found that combinations which generated the highest accuracy levels selected a low number of relevant features. However, combinations which generated intermediate accuracy levels (i.e., accuracies between the lowest and highest accuracies) were found to select a higher number of relevant features than combinations which generated lowest and highest accuracies.

A plausible explanation for such findings may lie within the relevance of features within the selected feature subsets. On the one hand, combinations which generated lowest accuracy levels may have missed out some highly relevant features. In this way, they will mainly include features that are of little relevance or possibly irrelevant features. The fact that such combinations excluded highly relevant features may explain why a lower number of features were selected. On the other hand, combinations which generated highest accuracy levels may include highly relevant features. This means that there will be mainly highly relevant features but very few irrelevant features in selected subsets. The fact that combinations mainly included highly relevant features may explain why fewer features were selected by such combinations. Combinations which generated intermediate accuracies, however, may include a mixture of highly relevant features and features that are of little relevance to the target variable. This may explain why their accuracies were neither too low nor too high. More importantly, this may explain why a higher number of features were selected by combinations which generated intermediate accuracies.

These results from UP1 and UP2 suggest that the number of relevant features selected is to some extent determined by the level of accuracy generated. On the one hand, combinations which generate the lowest or highest accuracies are more likely to select small feature subsets. On the other hand, combinations which generate accuracies in between the lowest accuracies and highest accuracies are more likely to select large feature subsets. In general, these findings imply that the number of relevant features

selected by combinations has some relationships with the accuracy levels that the selected features generated. Such findings therefore improve our understanding of relationships between the number of features selected by the combinations and how accurate these features are in relation to the target variable.

In summary, the results obtained in this section have shown that the number and nature of classifiers used significantly influence the number of relevant features selected in addition to the accuracy levels of selected features. We have also found relationships between the number of features selected and the accuracy levels of the features, which improves our understanding of the selected feature subsets. The next section moves on to determine the relationships between the selected relevant features by examining the decision trees of same-type combinations with the highest accuracy levels.

4.4 Visualising Features with Decision Trees

The previous section looked at the influences of number and nature of classifiers on feature selection results. The influences of these two issues were examined using the same-type combinations. Each of the same-type combinations were used to select relevant sets of features from the UP1 and UP2 datasets, which were then used to build decision trees. Each decision tree had an associated level of accuracy which indicates how relevant the feature sets used to build the trees are in relation to the target variable. In this section, the decision trees with the highest level of accuracies will be analysed. Analysing decision trees with the highest accuracy will help uncover the most relevant relationships between selected feature sets and the target variable. First, we present the decision tree with the highest accuracy for the UP1 dataset and then present the decision trees with the highest accuracies for the UP2 dataset.

4.4.1 Decision Tree of UP1 Dataset

1) Analysis of Features Selected by DT Family Combination

The C4.5+CART+CN2 DT family combination was found to generate a decision tree with the highest level of classification accuracy (i.e., 88.33%) from UP1. The fact that this combination led to the highest accuracy implies that it selects features that are most relevant to the target variable. This section presents the relevant features selected by this combination. The relevant features selected by this combination are

shown in Table 4.11. Each of the features presented in Table 4.11 has an associated level of relevance shown in parenthesis. This relevance value indicates how relevant a feature is to the target variable, i.e., computer experience. As an example, consider features 'Q13' and 'Q14'. Although both features were selected by this classifier combination, they differ in their relevance to computer experience. In the combination, 'Q13' is selected with a relevance value of 1, while 'Q14' is selected with a relevance value of 10. These relevance values indicate that 'Q14' is much more relevant to determining users' level of computer experience than 'Q13'.

	Selected Features and Their Relevance Values
C4.5+CART+CN2 (DT Family Combination)	Q4(1), Q11(1), Q13(1), Q14(10), Q15(2), Q21(3), Q29(2), Q30(2), Q31(6), Q32(1), Q33(2), Q38(1), Q48(3), Q49(1), Q56(2), Q58(1), Q66(1), Q76(1).

Table 4.11. Features Selected by C4.5+CART+CN2

A close look at the features reveals 'The results are presented by the levels of the relevance' (Q14) and 'There are not too many types of icons' (Q31) to be the most relevant feature and the second most relevant feature respectively. As such, these two features, especially Q14 because it is assigned the highest relevance value, can be said to play important roles in distinguishing the class values of the target variable (i.e., determining a user' level of computer experience). These two relevant features, in addition to the other relevant features selected by the DT family combination, were used to construct the decision tree. The constructed decisions tree is presented and explained in the following section.

2) Constructed Decision Tree

As shown in Figure 4.1, the decision tree formed comprises of three levels. The first level indicates the most important feature typically known as the root node while the remaining levels indicate other relevant features. This means that features found in levels two and three are not as relevant as the feature found in the first level. The decision tree also includes the number of users in the dataset that follow each level of computer experience (see the key of the decision tree for description of each level). For example, D (4), which can be found on the far left of the second level of the decision tree in Figure 4.1, signifies that 4 users who found Q31 strongly unimportant and found Q30 strongly unimportant had excellent level of computer experience.

Analysing the decision tree in Figure 4.1 reveals two interesting issues, which are explained in the following pages.

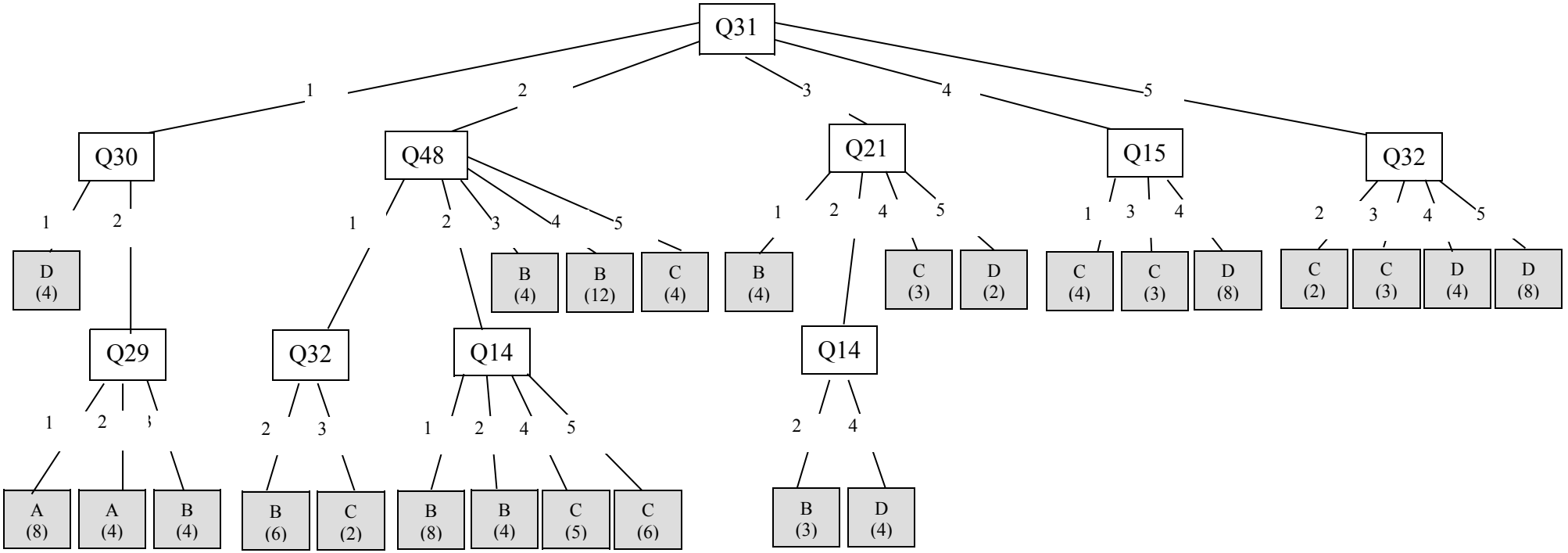

Figure 4.1. Decision Tree for C4.5+CART+CN2 Classifier Combination

Decision Tree Key	
Users' Preferences	*Level of Computer Experience*
1 = Very Unimportant	A = Little
2 = Unimportant	B = Average
3 = Neutral	C = Good
4 = Important	D = Excellent
5 = Very Important	

The first issue relates to the root node of the decision tree. In the decision tree, Q31 is the root node. This means that Q31 is considered as the most important feature by the decision tree classifier. Interestingly, this finding is somewhat different to what was found from the features selected by the C4.5+CART+CN2 combination. Analysing the features selected by this combination showed that Q14 was more relevant than Q31 since it had a much higher level of relevance (see Table 4.11). To further explore the differences among the relevance of Q31 and Q14, a deep analysis was carried out with the help of the ANalysis Of VAriance (ANOVA). The ANOVA, which is a useful method for analysing the statistical significances among three or more items, is applied to examine the significances of Q31 and Q14 in relation to the target variable (Miller and Neill, 2008). The result from ANOVA showed that both Q14 and Q31 are significant. However, the significance of Q14 is $p<.005$ while the significance of Q31 is $p<.001$. This shows that the significance of the latter is higher than the former. The analysis carried out here is based on statistical significances. On the other hand, decision tree classifiers also use statistical measures such as the information gain measure to determine the most relevant and least relevant features and their positions in the tree (Perner and Apte, 2004; Larrañaga et al., 2006). In this case, the decision tree classifier may have also deemed Q31 to be statistically more relevant to the target variable than Q14. This may therefore explain why Q31 is the root node of each decision tree and Q14 is found in the lower levels of the decision trees.

The second issue relates to the features used to form the decision tree. On close examination of the decision tree, three features were found to differentiate the preferences of users with low levels of computer experience and those with high levels of computer experience. The three features included: Q31, Q14, and Q48. In terms of Q31, this particular feature appears as the root node in the decision tree. This means that Q31 is a very important feature in relation to computer experience. The reason why the other two features are relevant lies within the fact that these features have a high number of users associated with them (as shown in parenthesis in each leaf of the decision tree). The implications of these three features for determining users' level of computer experience are further explained in the following pages.

Q31: There are not too many types of icons

Figure 4.1 shows that all users with little computer experience and the majority of users with average computer experience found this feature unimportant or very unimportant. On the other hand, the majority of participants with good and excellent levels of computer experience consider it important or very important. A possible explanation for these findings is that users with low levels of computer experience tend to have limited amount of knowledge so they may not be familiar with all the functionalities of search engines and may not know how to select suitable functionalities for their tasks. Thus, presenting a large selection of icons can enable them to easily identify various functionalities provided by the search engines and help them to choose the most suitable functionalities for their tasks.

Q14: The results are presented by the levels of the relevance

Many of the users with high levels of computer experience found this feature important while users with low levels of computer experience considered this feature unimportant. A possible interpretation for this may lie within the users' familiarisation with different search engines. Users with higher levels of computer experience are more likely to have used different types of search engines before. The wide use of search engines makes them more accustomed to using this function than those with less computer experience. This finding is also consistent to that shown by Brand-Gruwel, Wopereis, and Vermetten (2005) who found that users with high levels of computer experience had a tendency to use the relevance levels of search results to make a judgement.

Q48: Error messages let you know the cause of the problem

The majority of users with low levels of computer experience considered this feature important. In contrast, many of the users with high levels of computer experience considered it unimportant. When searching the Web for information, users with low levels of computer experience are more prone to making mistakes, for example, misspelling search terms. This is because such users possess only a limited amount of system knowledge when compared to individuals with higher levels of experience (Chen, Fan and Macredie, 2006). As such, it will be likely that users with less

experience will encounter more problems when searching the Web. Such users, therefore, preferred the idea of all errors being clearly explained. In this way, they will be able to interpret what they did to cause the errors. Once they know the nature of the errors, they can more easily and quickly find solutions.

The abovementioned findings from the three features suggest that users with low levels of computer experience had very different preferences to search engines compared to users with high levels of computer experience. Such features can be used to differentiate between the preferences of users with different levels of computer experience and in turn may be used to better understand how users with different computer experience locate information through search engines.

4.4.2 Decision Tree(s) of UP2 Dataset

1) Analysis of Features Selected by NN Family Combinations

Analysing the accuracies generated by all same-type combinations for the UP2 dataset showed that two NN family combinations, namely NNC+K*+KNN and NNC+K*+SVM, both generated the highest accuracy level, which was 96.92%. A deep analysis of these two combinations shows that both combinations select the same relevant features. The relevant features selected by these two combinations, and their associated relevance values, are: Q6(1), Q9(10), Q18(10), and Q19(1). By looking at the relevance values of the features, we also see that the relevance values of the features selected were identical between these two types of combinations. A plausible explanation for why both of these combinations selected exactly the same features with identical relevance values may lie within the classifiers used in the combinations.

The combinations have two classifiers in common which are NNC and K*. However, the combinations differ in the third classifier used. The first combination uses the KNN classifier while the second combination uses the SVM classifier. The fact that the combinations generated highest accuracies even though they included these different classifiers may suggest that these classifiers have some similarities. These two classifiers may be similar in the way they select relevant features. The KNN classifier

belongs to the NN family. NN family classifiers such as KNN use distance to determine which are most relevant (i.e., closest) and which are least relevant (i.e., furthest) to the target variable. The SVM classifier, which belongs to the SVM family, also uses distance to determine relevant features. Basically, SVM classifiers look at the distance of each feature in accordance to the hyperplane found, which best separates the values of the target variable. Features closest to the hyperplane will typically be regarded as highly relevant whereas features furthest away from the hyperplane will be regarded as not so relevant. In summary, the fact that KNN and SVM classifiers select relevant features in a similar manner may help explain why the two NN family combinations, which included these different classifiers, selected identical features with identical relevance values.

A closer look at the relevance values of the features selected by the two classifier combinations shows that 'It is hard to use the back/forward buttons' (Q9) and 'It is easy to find a route for a specific task with the index' (Q18) were assigned the highest relevance level of 10 by both combinations. This suggests that Q9 and Q18 are the most relevant features among the selected features with respect to the target variable. In other words, Q9 and Q18 are highly relevant to determining users' cognitive styles. In order to further examine the relevance of all selected features, including that of Q9 and Q18, and their relationships with the target variable, the next section looks at the decision trees constructed by the two classifier combinations.

2) Constructed Decision Trees

In addition to selecting identical relevant features, the NNC+K*+KNN and NNC+K*+SVM combinations also formed identical decision trees (Figure 4.2). This is the reason why only one decision tree is presented as opposed to two. As illustrated in Figure 4.2, the decision tree has two levels where the first level includes a single feature that is the root node. In addition, each decision tree also includes the number of users in the dataset that follow each type of cognitive style (see the key of the decision tree for description of each type). For example, FI (21), which appears on the far left of the second level, means that the 21 users who strongly disagreed with Q9 were Field

Independent. An analysis of the decision tree displayed in Figure 4.2 reveals some interesting findings, which are explained further down the page.

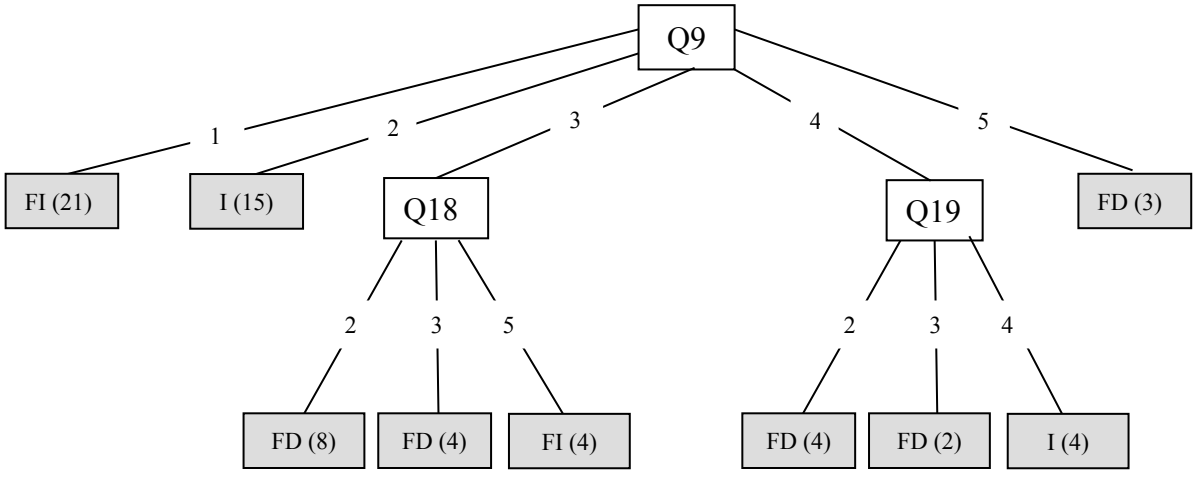

Figure 4.2. Decision Tree for NNC+KNN+K* and NNC+K*+SVMpoly Combinations

Decision Tree Key	
Users' Preferences	*Cognitive Style*
1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree	FI = Field Independent I = Intermediate FD = Field Dependent

An interesting finding relates to the features used in this decision tree. The decision tree makes use of three features namely, Q9, Q18, and Q19. On the other hand, Q6 was not included in the construction of the decision tree. The fact that Q6 was not included in the tree but Q19 was is rather surprising because both of these features were found to have identical relevance values of 1. In order to better understand the relevance of Q6 and Q19 in relation to the target variable, the ANOVA method was used. The result from ANOVA showed that both Q6 and Q19 are significant. However, the significance of Q19 ($F=11.88$, $p<.001$) is slightly higher than the significance of Q6 ($F=7.769$, $p<.005$). This means that Q19 is statistically more relevant to the target variable than Q6, which may explain why the DT classifier used Q19, instead of Q6 when building the tree.

The second interesting finding relates to the three features, i.e., Q9, Q18, and Q19, included in the decision tree. The implications of these features for determining users' cognitive style are discussed over the next few pages.

Q9: It is hard to use the back/forward buttons

In terms of 'It is hard to use the back/forward buttons' (Q9), this particular feature appeared as the root node of all decision trees. This implies that Q9 is the most relevant feature with regards to users' cognitive style. This feature was also found to be highly relevant feature by both of the classifier combinations. To see how users with different cognitive styles preferred this feature, we carry out a deep analysis of the decision tree. In general, the decisions trees showed that nearly all of the Field Independent (FI) and Intermediate (I) users strongly disagreed and disagreed with this feature, respectively, while majority of Field Dependent (FD) users agreed or strongly agreed with it. The reason for this different preference may be because FI users like to work on their own (Witkin et al, 1977) and find their own route on the Web (Liu and Reed, 1995). As a result, FI users may prefer to freely navigate the Web-based learning system. The back/forward buttons are navigation tools that may help FI users to do this. In contrast, FD users easily get disorientated (Chen and Macredie, 2004) so the back/forward buttons, which enable free and unguided navigation, may let them feel lost and are not very suitable for them.

Q18: It is easy to find a route for a specific task with the index

The other feature in the decision trees is 'It is easy to find a route for a specific task with the index' (Q18). This particular feature was also found to be a highly relevant feature among the combinations, which suggests that Q18 may play an important role in identifying a user's cognitive style. Studying the decision tree in Figure 4.2 confirms the importance of Q18. It was found that some FI users strongly agreed with this feature whereas quite a few FD users disagreed with it. This finding is in line with those of Ford and Chen (2000) who also found that FI users prefer using the index to locate a particular item. FI users prefer using the index because it provides them with a break down of all the information in the system. In this way, they will be able to easily find

specific information. On the other hand, FD users are individuals who prefer the system to provide them with a holistic view all the information so the index, which provides specific information may not be suitable to them. This may therefore explain why FI users and FD users had different preferences regarding the index.

Q19: This tutorial can be used sufficiently well without any instructions

In terms of Q19, some Intermediate users were found to agree with this particular feature while some FD users were found to disagree with this feature. This shows that Intermediate users could use the tutorial without the help of any instructions, whereas FD found the tutorial difficult to use without a set of instructions presented to them. This may have to do with the type of approach Intermediate and FD users adopt (Witkin, et al, 1977; Chen, 2002). Typically, Intermediate users exhibit some traits from both FI and FD users. In the case of Q19, Intermediate users exhibited traits of FI users. This is because FI users use an active approach in that they prefer to roam the system independently, use their own initiative to find the most relevant information for completing a task, and thus complete the task without any instructions or guidance. This may explain why Intermediate users were able to use the tutorial without the help of any instructions. On the other hand, FD users adopt a passive approach, which means that they rely on guidance or instructions provided by the system so that they can complete their task. As such, they may find it rather difficult to use the system without some instructions.

The aforementioned results showed that Q9, Q18 and Q19 are very important for determining users' cognitive styles. These features may therefore play a key role in understanding how users with different cognitive styles prefer using Web-based learning systems. In this way, Web-based learning systems can be designed to integrate these features in order to satisfy the needs of users with different cognitive styles.

4.4.3 Decision Trees with Highest Accuracies Formed by DT and NN Family Classifiers

The previous section presented the decision trees with the highest accuracies for both the UP1 and UP2 datasets. For the UP1 dataset, it was found that a combination

comprised of DT family classifiers produced the decision tree with the highest accuracy. For the UP2 dataset it was found that two combinations mainly comprised of NN family classifiers produced decision trees with the highest accuracy level. This suggests that combinations with DT classifiers and combinations with NN family classifiers are able to select features that form decision trees with the highest levels of accuracy. The reason why combinations with these two classifier families were able to produce decision trees with the highest accuracy may relate to similarities in their nature. As previously mentioned in Section 4.3.2, DT family classifiers determine relevant features according to their position in the tree. Features that appear at the top of the tree are considered as the most relevant while features towards the bottom of the hierarchical tree structure are considered the least relevant. The NN family classifiers determine the relevant features by considering their distance. The closest features (according to some distance metric) are regarded as the most relevant while features that are furthest away are regarded as the least relevant. This shows that DT classifier and NN family classifiers determine the relevant features in a similar manner. The fact that DT and NN families are similar may help explain why classifiers from these particular families were able to build decision trees with the highest accuracies.

However, the DT and NN families generated trees with highest accuracies in different datasets. The DT family combinations generated highest accuracies using UP1 dataset whereas NN family combination generated highest accuracies using UP2 dataset. The root of such a finding may lie within their ability to identify relevant features in datasets of different sizes. On the one hand, the UP1 dataset is a large dataset in that it consists of a large number of features. DT family classifiers may be more likely to determine highly relevant features within large datasets. This may be to do with the nature of DT classifiers. Typically, DT classifiers use feature selection themselves to search for features with the highest discrimination ability since these features are most relevant to the target variable (Guyon and Elisseeff, 2003). The possibility of identifying such features increases when there are many features in the dataset as there is a wider selection of features from which DT classifiers can choose from. Once identified, these highly relevant features are used to build a hierarchical tree structure, which shows the relationships between the relevant features and the target variable of the dataset. Since

the tree structure is built using features highly relevant to the target variable then the tree will most probably generated high accuracy levels. This may explain why DT family classifiers built decision trees with high accuracy levels for UP1, which consists of many features.

On the other hand, the UP2 dataset is a small dataset in that it consists of a small number of features. NN family classifiers may be more suited to small datasets. This is because NN family classifiers will only need to calculate the distances among few features in order to differentiate between the relevant (i.e., closest) and irrelevant (i.e., furthest) features. The fact that NN family classifiers will have to handle distances of fewer features may make it easier for them to differentiate the relevant features from the irrelevant features. In this way, the NN family classifiers will be able to identity the features most relevant to the target variable. Using most relevant features to build the decision trees may lead to high accuracy levels. This may therefore explain why NN family classifiers built decision trees that generated highest accuracies in UP2.

In summary, the DT family and NN family classifiers were found to build decision trees with the highest accuracies. This was attributed to similarities among these two classifier families. The similarities among these two classifier families may also suggest that such families are quite useful for selecting features that lead to decision trees with the highest accuracy levels. In other words, the nature of such classifier families enables them to form most accurate decision trees. Such decision trees can be used to identify the most relevant features and the most relevant relationships between features, which in this case can be used to improve our understanding of the preferences of different users.

4.5 Conclusions

This chapter investigated the influences of the number and nature of classifiers on feature selection results using the same-type approach. The results from this investigation revealed that the number and nature of classifiers significantly influence the number of features selected and the accuracy levels of these features.

In terms of the number of features selected, both datasets showed that combinations with few classifiers (i.e., 2 classifiers) selected many relevant features whereas combinations with many classifiers (i.e., 3 and 4 classifiers) selected few relevant features. In addition, it was found that the number of features identified was further influenced by the use of certain classifiers. More specifically, combinations with DT family classifiers were found to select a higher number of relevant features from both UP1 and UP2 datasets in comparison to the other classifier families. In terms of the accuracies of selected features, our results revealed that combinations comprised of three classifiers selected features which led to the highest classification accuracies in comparison to the other combinations. This was found in both UP1 and UP2 datasets. This indicates that features selected by 3-classifier combinations are of higher relevance to the target variable than features selected by other classifier combinations. However, the accuracy levels of 3-classifier combinations and many of the other classifier combinations were influenced by the DT family classifiers. It was found that combinations with DT family classifiers generated higher accuracies than combinations with classifiers from the other families. This suggests that combinations with DT family classifiers significantly influence number of features as well as the accuracies of features from UP1 and UP2. In fact, DT family classifiers seem to be able to select large, yet precise, feature subsets. These findings suggest that the nature of classifiers belonging to DT family is more influential than the nature of other classifier families used, which provides us with a better understanding of the importance and usefulness of this classifier family for feature selection.

In summary, the results from this chapter show that the number and nature of classifiers used influence feature selection results. However, the influences of these two issues were examined using the same-type approach. The same-type approach is one of the two approaches required to fully examine the influences of number and nature of classifiers. The other approach is the mixed-type approach. The following chapter therefore uses the mixed-type approach to investigate the influences of number and nature of classifiers on the number of features selected and the accuracy levels of the selected features. By using the mixed-type approach, we will be able to obtain a

complete picture of the influences of number and nature of classifiers on feature selection.

Chapter 5 – The Combinations of Mixed-Type Classifiers

5.1 Introduction

The number of classifiers and the nature of classifiers are two important issues that can influence feature selection results namely the number of relevant features selected and the accuracy levels of the features. These two issues are investigated in this thesis using two approaches, including the same-type approach and the mixed-type approach. The former approach was investigated in the previous chapter (Chapter 4) while the latter approach will be examined in this chapter. Likewise, the mixed-type approach will be examined using the two datasets (i.e., UP1 and UP2) that were used in the previous chapter. The results from the mixed-type approach for both of these datasets are presented in this chapter.

The chapter is organised as follows. Section 5.2 gives a brief description of what classifiers are used to form the mixed-type combinations for selecting the relevant features from the datasets. Subsequently, Section 5.3 presents the results from mixed-type combinations regarding influences on number of features selected and the accuracy levels generated by selected features. It then moves on to identify relationships between the number of features selected by combinations and the accuracies that they generated. Finally, a deep analysis of decision trees with highest accuracies is conducted in Section 5.4 to identify relevant relationships between selected features.

5.2 Mixed-type Classifier Combinations

The mixed-type approach involves combining classifiers with different types of nature. The mixed-type combinations are formed using the same classifiers and classifier families (i.e., BN family, DT family, NN family and SVM family) used in the previous chapter. The classifiers were combined using an exhaustive approach so that each classifier was used with every other classifier with a different nature. This led to the construction of 2-classifier mixed-type combinations (e.g., BNC+C4.5, BNC+SVMpoly, BNC+SVMrbf), 3-classifier mixed-type combinations (e.g., BNC+C4.5+KNN, BN+C4.5+SVMpoly, BN+C4.5+SVMrbf), and 4-classifier mixed-

type combinations where all the four families of classifiers were combined (e.g., BNC+C4.5+KNN+SVMpoly and BNC+C4.5+KNN+SVMrbf). The SVM classifier is used within the mixed-type combinations with a different kernel. For example, the BNC+SVMpoly combination uses the BNC classifier and the SVM classifier with the polynomial kernel (SVMpoly). On the other hand, the BNC+SVMrbf combination uses the features selected by the BNC classifier and the SVM classifier with the radial basis function kernel (SVMrbf).

In total, 180 mixed-type classifier combinations were formed, including 45 for 2-classifiers (27 without SVM, 9 with SVMpoly, and 9 with SVMrbf), 81 for 3-classifiers (27 without SVM, 27 with SVMpoly, and 27 with SVMrbf), and 54 for 4-classifiers (27 with SVMpoly, and 27 with SVMrbf). These 180 mixed-type combinations will be used to select relevant features from each dataset. The features selected by the combinations will subsequently be used to build decision trees in order to identify the relationships between the features and determine the classification accuracies of the selected features. The classification accuracies will help determine how relevant each feature is in relation to the target variable, where a high accuracy indicates high relevance and a low accuracy indicates low relevance.

5.3 Results from UP1 and UP2 Datasets

The mixed-type classifier combinations also use the UP1 and UP2 datasets to investigate the influences of number and nature of classifiers on number of features selected and accuracy levels. From the UP1 dataset, the mixed-type combinations will use the users' level of computer experience as the target variable and 90 input features. From the UP2 dataset, the combinations will use the user's cognitive style as the target variable and 20 input features. The results regarding the number of relevant features chosen by each mixed-type combination and the classification accuracies generated using the selected relevant features for UP1 and UP2 are shown in Table 5.1 and Table 5.2 respectively. These two tables organise the mixed-type combinations according to their classification accuracies. The combinations which produce low accuracy levels appear on the left side in the table whereas combinations which produce high accuracies appear on the right side. By organising the combinations in this manner, we can easily

and quickly discover those combinations which are less accurate and more accurate. Furthermore, each of the mixed-type combinations presented in the tables includes the number of features that they selected. The number of features selected by each combination is shown in parenthesis to the right of each combination.

In addition to Table 5.1 and Table 5.2, we present Table 5.3 to show the mean number of features selected by the 2-classifier, 3-classifier and 4-classifier combinations and the mean accuracy levels generated by the combinations for both UP1 and UP2. In fact, Table 5.3 includes the mean number of features selected and accuracies generated by combinations without SVM classifiers, mean number of features selected and accuracies generated by combinations with SVMpoly, mean number of features selected and accuracies generated by combinations with SVMrbf, and the overall number of features selected and overall accuracies generated by the combinations. This table can help us easily compare the number of features selected and the accuracy levels generated by the different types of classifier combinations.

	Mixed-type Classifier Combinations			
	Lowest accuracies between 72.50 to 79.17% (N=28)	***Intermediate Accuracies between 80 to 85% (N=116)***		***Highest accuracies between 85.83 to 90.83% (N=35)***
2-classifier	BNC+C4.5 (18) BNC+CART (15) BNC+K* (14) BNC+SVMpoly (14) BNC+SVMrbf (18) NB+CART (19) NB+SVMrbf (23) KNN+SVMpoly (12)	BNC+CN2 (17) BNC+NNC (9) BNC+KNN (13) NB+C4.5 (22) NB+CN2 (17) NB+KNN (12) NB+K* (20) NB+SVMpoly (18) AODE+CN2 (23) AODE+K* (19) AODE+SVMpoly (22) AODE+SVMrbf (24) C4.5+NNC (16) C4.5+KNN (18) C4.5+K* (21) C4.5+SVMpoly (21) C4.5+SVMrbf (25)	CART+NNC (18) CART+KNN (16) CART+K* (19) CART+SVMpoly (18) CART+SVMrbf (21) CN2+NNC (17) CN2+KNN (15) CN2+K* (19) CN2+SVMpoly (20) CN2+SVMrbf (22) NNC+SVMpoly (12) NNC+SVMrbf (16) KNN+SVMrbf (15) K*+SVMpoly (18) K*+SVMrbf (20)	NB+C4.5 (23) NB+CART (23) NB+NNC (15) AODE+NNC (18) AODE+KNN (16)

3-classifier	BNC+CN2+NNC (7) BNC+CART+KNN (11) BNC+CN2+KNN (5) BNC+CART+SVMpoly (13) BNC+CART+SVMrbf (15) BNC+CN2+SVMpoly (6) BNC+CN2+SVMrbf (9) BNC+K*+SVMrbf (15) AODE+CN2+K* (18) AODE+KNN+SVMrbf (18)	BNC+C4.5+NNC (14) BNC+C4.5+KNN (12) BNC+C4.5+K* (14) BNC+CART+K* (12) BNC+CN2+K* (8) BNC+C4.5+SVMpoly (13) BNC+C4.5+SVMrbf (17) BNC+NNC+SVMpoly (9) BNC+NNC+SVMrbf (9) BNC+KNN+SVMpoly (10) BNC+KNN+SVMrbf (11) BNC+K*+SVMpoly (14) NB+C4.5+NNC (15) NB+C4.5+KNN (16) NB+CART+KNN (16) NB+CN2+KNN (12) NB+C4.5+K* (18) NB+C4.5+SVMpoly (19) NB+C4.5+SVMrbf (20) NB+CART+SVMrbf (21) NB+CN2+SVMrbf (14) NB+NNC+SVMpoly (15) NB+NNC+SVMrbf (16) NB+KNN+SVMpoly (17) NB+KNN+SVMrbf (19) NB+K*+SVMpoly (17) NB+K*+SVMrbf (20) AODE+C4.5+SVMpoly (21)	AODE+C4.5+SVMrbf (23) AODE+CART+SVMpoly (22) AODE+CART+SVMrbf (23) AODE+CN2+SVMpoly (21) AODE+C4.5+NNC (17) AODE+CART+NNC (20) AODE+CN2+NNC (14) AODE+C4.5+KNN (17) AODE+CN2+KNN (14) AODE+CN2+SVMrbf (22) AODE+NNC+SVMrbf (17) AODE+KNN+SVMpoly (18) AODE+K*+SVMpoly (23) AODE+K*+SVMrbf (24) C4.5+NNC+SVMrbf (18) C4.5+KNN+SVMpoly (17) C4.5+KNN+SVMrbf (17) CART+NNC+SVMpoly (17) CART+NNC+SVMrbf (20) CART+KNN+SVMpoly (16) CART+KNN+SVMrbf (18) CN2+NNC+SVMpoly (14) CN2+NNC+SVMrbf (15) CN2+KNN+SVMpoly (13) CN2+KNN+SVMrbf (14) CN2+K*+SVMpoly (18) CN2+K*+SVMrbf (20)	BNC+CART+NNC (11) NB+CART+NNC (15) NB+CN2+NNC (13) NB+CART+K* (19) NB+CN2+K* (17) NB+CART+SVMpoly (19) NB+CN2+SVMpoly (12) AODE+CART+KNN (18) AODE+C4.5+K* (20) AODE+CART+K* (20) AODE+NNC+SVMpoly (16) C4.5+NNC+SVMpoly (17) C4.5+K*+SVMpoly (16) C4.5+K*+SVMrbf (18) CART+K*+SVMpoly (21) CART+K*+SVMrbf (24)
4-classifier	BNC+CN2+NNC+SVMpoly (8) BNC+CN2+NNC+SVMrbf (9) BNC+CART+K*+SVMpoly (13) BNC+CART+K*+SVMrbf (15) BNC+CN2+K*+SVMpoly (9) BNC+CN2+K*+SVMrbf (10) NB+C4.5+KNN+SVMrbf (15) NB+CART+K*+SVMpoly (16) NB+CART+K*+SVMrbf (18) AODE+CN2+K*+SVMrbf (14)	BNC+CART+NNC+SVMpoly (12) BNC+CART+NNC+SVMrbf (13) BNC+C4.5+KNN+SVMpoly (9) BNC+C4.5+KNN+SVMrbf (10) BNC+CART+KNN+SVMpoly (10) BNC+CART+KNN+SVMrbf (13) BNC+CN2+KNN+SVMpoly (5) BNC+CN2+KNN+SVMrbf (6) BNC+C4.5+K*+SVMpoly (11) BNC+C4.5+K*+SVMrbf (12) NB+CART+NNC+SVMpoly (13) NB+CART+NNC+SVMrbf (14) NB+CN2+NNC+SVMrbf (15) NB+C4.5+KNN+SVMpoly (13) NB+CART+KNN+SVMrbf (14) NB+CN2+KNN+SVMpoly (9)	NB+CN2+KNN+SVMrbf (9) NB+C4.5+K*+SVMpoly (13) NB+C4.5+K*+SVMrbf (14) NB+CN2+K*+SVMpoly (15) NB+CN2+K*+SVMrbf (18) AODE+CART+NNC+SVMpoly (13) AODE+CART+NNC+SVMrbf (14) AODE+CN2+KNN+SVMpoly (10) AODE+CN2+KNN+SVMrbf (11) AODE+C4.5+K*+SVMpoly (14) AODE+C4.5+K*+SVMrbf (16) AODE+CART+K*+SVMpoly (14) AODE+CART+K*+SVMrbf (17) AODE+CN2+K*+SVMpoly (11)	BNC+C4.5+NNC+SVMpoly (9) BNC+C4.5+NNC+SVMrbf (10) NB+C4.5+NNC+SVMpoly (13) NB+C4.5+NNC+SVMrbf (13) NB+CN2+NNC+SVMpoly (13) NB+CART+KNN+SVMpoly (13) AODE+C4.5+NNC+SVMpoly (12) AODE+C4.5+NNC+SVMrbf (14) AODE+CN2+NNC+SVMpoly (13) AODE+CN2+NNC+SVMrbf (15) AODE+C4.5+KNN+SVMpoly (11) AODE+C4.5+KNN+SVMrbf (12) AODE+CART+KNN+SVMpoly (12) AODE+CART+KNN+SVMrbf (14)

Table 5.1. No. of Features Selected by Mixed-type Combinations and Associated Classification Accuracy Levels for UP1 Dataset.

	Mixed-type Classifier Combinations			
	Lowest Accuracies of 90.77% and 92.31% (N=35)	***Intermediate Accuracies of 93.85% and 95.38% (N=136)***		***Highest Accuracies of 96.92% (N=9)***
2-classifier	BNC+CART (8) BNC+KNN (10) BNC+SVMpoly (5) BNC+SVMrbf (8) NB+KNN (12) AODE+KNN (11) AODE+SVMrbf (7) CART+SVMrbf (10) CART+KNN (15) CART+K* (7) NNC+SVMrbf (6) K*+SVMrbf (6)	BNC+C4.5 (4) BNC+CN2 (6) BNC+NNC (3) BNC+K* (3) NB+CART (9) NB+C4.5 (5) NB+CN2 (6) NB+NNC (3) NB+K* (3) NB+SVMpoly (4) NB+SVMrbf (9) AODE+C4.5 (5) AODE+CART (8) AODE+K* (3) AODE+CN2 (6) AODE+NNC (3) AODE+SVMpoly (4)	C4.5+KNN (12) C4.5+NNC (4) C4.5+K* (4) C4.5+SVMpoly (7) C4.5+SVMrbf (9) CART+NNC (8) CART+SVMpoly (8) CN2+KNN (12) CN2+SVMrbf (10) CN2+NNC (5) CN2+K* (6) CN2+SVMpoly (6) NNC+SVMpoly (5) KNN+SVMpoly (14) KNN+SVMrbf (16) K*+SVMpoly (4)	-

3-classifier	BNC+CART+SVMrbf (9) BNC+KNN+SVMrbf (11) NB+KNN+SVMrbf (11) NB+C4.5+SVMrbf (8) AODE+C4.5+SVMrbf (8) AODE+KNN+SVMrbf (13) AODE+K*+SVMrbf (5) C4.5+K*+SVMrbf (5) C4.5+NNC+SVMrbf (7) C4.5+KNN+SVMrbf (13) CART+KNN+SVMrbf (13) CART+K*+SVMrbf (8) CN2+KNN+SVMrbf (13)	BNC+CART+KNN (9) BNC+CART+K* (2) BNC+C4.5+SVMrbf (7) BNC+C4.5+NNC (4) BNC+CART+NNC (4) BNC+CN2+NNC (4) BNC+C4.5+KNN (7) BNC+CN2+KNN (6) BNC+CN2+K* (4) BNC+C4.5+SVMpoly (4) BNC+CART+SVMpoly (7) BNC+CN2+SVMrbf (6) BNC+NNC+SVMrbf (4) BNC+KNN+SVMpoly (9) BNC+K*+SVMpoly (2) BNC+K*+SVMrbf (3) NB+CART+KNN (8) NB+CN2+NNC (4) NB+C4.5+KNN (8) NB+CN2+KNN (7) NB+C4.5+K* (4) NB+CART+K* (5) NB+CN2+K* (5) NB+C4.5+SVMpoly (5) NB+CART+SVMrbf (10) NB+CART+SVMpoly (8) NB+CN2+SVMpoly (6) NB+CN2+SVMrbf (7) NB+NNC+SVMpoly (6) NB+NNC+SVMrbf (7)	NB+KNN+SVMpoly (9) NB+K*+SVMpoly (4) NB+K*+SVMrbf (5) AODE+CART+KNN (10) AODE+CART+K* (5) AODE+C4.5+SVMpoly (5) AODE+CART+SVMpoly (8) AODE+CART+SVMrbf (10) AODE+CN2+SVMrbf (9) AODE+NNC+SVMrbf (6) AODE+KNN+SVMpoly (11) AODE+CN2+NNC (5) AODE+C4.5+KNN (8) AODE+CN2+KNN (10) AODE+CN2+K* (5) AODE+CN2+SVMpoly (5) AODE+NNC+SVMpoly (4) AODE+K*+SVMpoly (4) C4.5+KNN+SVMpoly (12) C4.5+K*+SVMpoly (3) CART+NNC+SVMpoly (7) CART+K*+SVMpoly (7) CART+NNC+SVMrbf (9) CART+KNN+SVMpoly (11) CN2+K*+SVMrbf (5) CN2+NNC+SVMpoly (5) CN2+NNC+SVMrbf (6) CN2+KNN+SVMpoly (10) CN2+K*+SVMpoly (5)	BNC+C4.5+K* (4) BNC+CN2+SVMpoly (5) BNC+NNC+SVMpoly (4) NB+C4.5+NNC (4) NB+CART+NNC (6) AODE+C4.5+NNC (5) AODE+CART+NNC (5) AODE+C4.5+K* (6) C4.5+NNC+SVMpoly (4)

4-classifier	BNC+CART+KNN+SVMrbf (10) NB+C4.5+KNN+SVMrbf (10) NB+CART+KNN+SVMrbf (11) AODE+CN2+NNC+SVMrbf (3) AODE+C4.5+KNN+SVMrbf (9) AODE+CART+KNN+SVMrbf (11) AODE+CN2+KNN+SVMrbf (10) AODE+C4.5+K*+SVMrbf (3) AODE+CART+K*+SVMrbf (5) AODE+CN2+K*+SVMrbf (3)	BNC+C4.5+NNC+SVMrbf (2) BNC+CART+NNC+SVMrbf (4) BNC+C4.5+NNC+SVMpoly (3) BNC+CART+NNC+SVMpoly (5) BNC+CN2+NNC+SVMpoly (3) BNC+CART+KNN+SVMpoly (7) BNC+CN2+KNN+SVMpoly (6) BNC+CN2+NNC+SVMrbf (2) BNC+C4.5+KNN+SVMpoly (5) BNC+C4.5+KNN+SVMrbf (8) BNC+CN2+KNN+SVMrbf (9) BNC+C4.5+K*+SVMpoly (2) BNC+CART+K*+SVMpoly (3) BNC+CN2+K*+SVMpoly (2) BNC+C4.5+K*+SVMrbf (2) BNC+CART+K*+SVMrbf (4) BNC+CN2+K*+SVMrbf (3) NB+C4.5+NNC+SVMpoly (3) NB+CART+NNC+SVMpoly (5) NB+CN2+NNC+SVMpoly (3) NB+C4.5+NNC+SVMrbf (4) NB+CART+NNC+SVMrbf (7)	NB+CN2+NNC+SVMrbf (4) NB+CART+KNN+SVMpoly (7) NB+CN2+KNN+SVMrbf (10) NB+C4.5+K*+SVMpoly (3) NB+C4.5+K*+SVMrbf (3) NB+CART+K*+SVMrbf (6) NB+C4.5+KNN+SVMpoly (6) NB+CN2+KNN+SVMpoly (7) NB+CART+K*+SVMpoly (4) NB+CN2+K*+SVMpoly (3) NB+CN2+K*+SVMrbf (4) AODE+C4.5+NNC+SVMrbf (4) AODE+CART+NNC+SVMrbf (6) AODE+CN2+NNC+SVMpoly (3) AODE+C4.5+KNN+SVMpoly (9) AODE+CART+KNN+SVMpoly (11) AODE+CN2+K*+SVMpoly (4) AODE+C4.5+NNC+SVMpoly (3) AODE+CART+NNC+SVMpoly (5) AODE+CN2+KNN+SVMpoly (5) AODE+C4.5+K*+SVMpoly (3) AODE+CART+K*+SVMpoly (4)	-

Table 5.2. No. of Features Selected by Mixed-type Combinations and Associated Classification Accuracy Levels for UP2 Dataset.

	Mean Number of Features & Mean Accuracy Levels for UP1							
	Combinations Without SVM		*Combinations With SVMpoly*		*Combinations With SVMrbf*		*Overall*	
2-classifier	17.22	82.20	17.41	81.48	20.22	80.83	17.93	81.78
3-classifier	15.56	83.95	16.10	83.36	17.81	82.03	16.16	83.12
4-classifier	-	-	11.63	83.06	13.14	81.97	12.38	82.52
	Mean Number of Features & Mean Accuracy Levels for UP2							
	Combinations Without SVM		*Combinations With SVMpoly*		*Combinations With SVMrbf*		*Overall*	
2-classifier	6.70	94.19	6.33	94.36	9	92.82	7.10	93.97
3-classifier	5.70	95.44	6.30	95.15	8.07	93.16	6.70	94.59
4-classifier	-	-	4.30	94.81	5.81	93.45	5.20	94.13

Table 5.3. Mean No. of Features selected and Mean Accuracy Levels for UP1 and UP2

An initial examination of Tables 5.1, 5.2 and 5.3 reveals a common finding, which relates to the number of features selected and the accuracy levels generated by combinations with SVMpoly and combinations with SVMrbf. It was found that combinations with SVMrbf selected a higher number of relevant features than combinations with SVMpoly. This subsequently meant that higher mean numbers of relevant features were obtained by the different classifier combinations when combinations with SVMrbf were used. In addition, it was found in both UP1 and UP2 that combinations with SVMrbf usually generated significantly lower classification accuracies than the accuracies of combinations with SVMpoly. This meant that lower mean accuracies were observed when combinations with SVMrbf were used. The fact that combinations with SVMrbf generated lower accuracies than combinations with SVMpoly for both datasets suggests that the former selected less relevant features with respect to the target variable than those of the latter.

These findings tie in with those of the same-type combinations. Findings from the same-type combinations also showed that combinations with SVMrbf resulted in more features being selected but lower accuracies being generated. The findings suggest that the polynomial kernel may be more suitable to analyse such types of datasets when used with the SVM classifier. Consequently, SVMrbf will not be considered in the rest of this chapter.

5.3.1 Number of Relevant Features

This subsection presents the results of the influences of number and nature of classifiers on the number of relevant features selected by the mixed-type

combinations from the UP1 and UP2 datasets. The results revealed three key results regarding the number of features selected from these datasets, each of which is described in subsequent pages.

1) Few Classifiers Select More Relevant Features and Many Classifiers Select Few Relevant Features

A close look at the results from UP1 and UP2 shows that mixed-type combinations comprised of few classifiers selected a higher number of relevant features than mixed-type combinations with many classifiers. More specifically, the 2-classifier combinations identified more relevant features than the 3-classifier and 4-classifier combinations. In terms of the UP1 dataset, we found that the mean number of features selected by 2-classifier combinations (17.36) was significantly higher than the mean number of features selected by 3-classifier (15.31) and 4-classifier combinations (11.63). In terms of the UP2 dataset, the mean number of features selected by 2-classifier combinations (6.53) was also found to be higher than the mean number of features selected by 3-classifier (5.94) and 4-classifier combinations (4.30). A close examination of the number of features selected by the 2-classifier, 3-classifier and 4-classifier combinations was also carried out to better understand differences among the combinations. The examination involved identifying the number (i.e., frequency) of 2-classifier, 3-classifier and 4-classifier combinations that selected number of features above and below total mean number of features for each dataset. The total mean number of features for UP1 is 16.33 and total mean number of features for UP2 is 5.76. The results from this examination for UP1 and UP2 are presented in Figure 5.1 and Figure 5.2, respectively.

In general, the results from the two figures showed that many of the 2-classifier combinations selected number of features above total mean number of features while few were below total mean number of features. This suggests that 2-classifier combinations generally selected a high number of relevant features. On the other hand, we found that there are more 3-classifier and 4-classifier combinations that selected number of features below total mean but fewer combinations of this type that were above the total mean. This suggests that 3-classifier and 4-classifier combinations generally selected a low number of relevant features. The results from

these two figures support the fact that few classifiers select more relevant features and many classifiers select few relevant features.

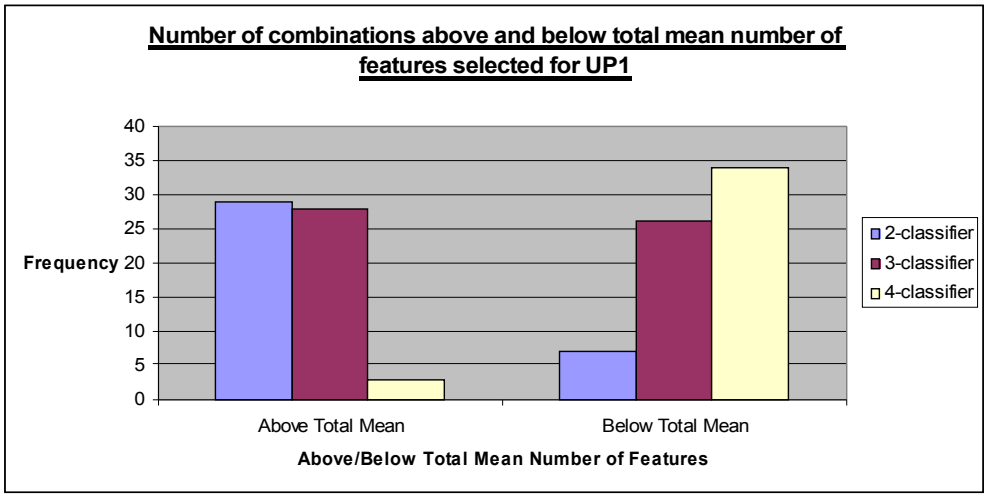

Figure 5.1. Number of Relevant Features Selected by Mixed-type Combinations for UP1

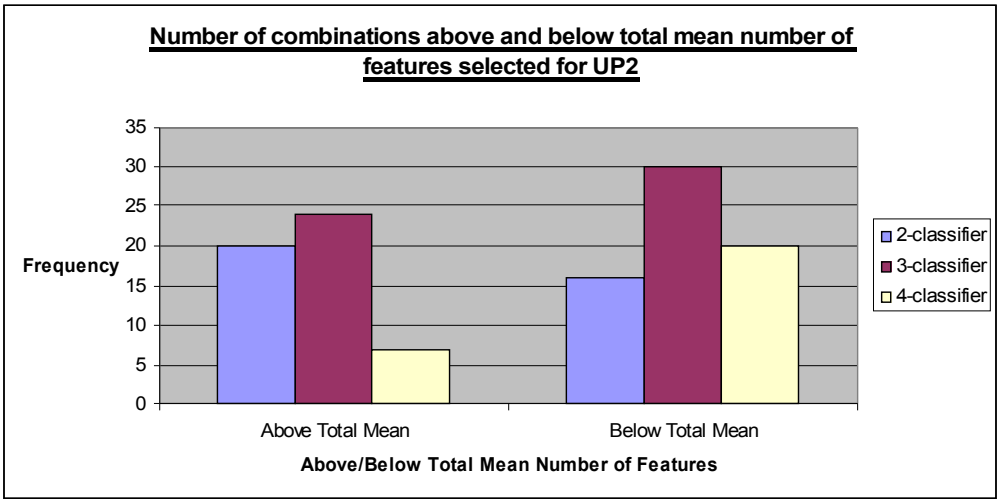

Figure 5.2. Number of Relevant Features Selected by Mixed-type Combinations for UP2

These findings suggest that using few classifiers for feature selection can lead to more relevant features being selected, whereas using many classifiers can often lead to few relevant features being selected. This finding may be attributed to the median strategy used to combine the classifiers. It was found that combining few classifiers with this strategy enabled more features to be selected because few classifiers were needed to agree on the relevance of a particular feature. More specifically, if few classifiers are used then it is easier for the classifiers to agree on a feature because there are few relevance values to consider, which will result in more features being selected.

However, combining many classifiers may make it slightly harder for the classifiers to agree on a particular feature because there are more relevance values to consider for each feature. This may subsequently result in fewer relevant features being selected.

2) Combinations with BN Family Classifiers Influence Number of Relevant Features Selected from UP1

In general, the results from the UP1 dataset show that combinations with BN family classifiers influence the number of relevant features selected. To better understand the influences on number of features, we examined the number of features selected by combinations with the different classifier families (Figure 5.3). A comparison of the number of features selected by the different classifier families shows that many of the combinations with BN family classifiers selected low number of relevant features. In fact, there are more combinations with BN family classifiers in number groups '5 to 10' and '11 to 15' than combinations with classifiers from the other families. Please note that we do not consider the results from combinations with SVM classifier since these combinations generally selected lowest number of features across all number groups. On the other hand, few combinations with BN family classifiers selected a high number of features, as shown by the fact there were generally fewer combinations with BN family classifiers in number groups '16 to 20' and '21 to 25' than combinations with classifiers from the other families. The differences in number of features selected as shown in this figure suggest that BN family classifiers have some influence on the number of features identified.

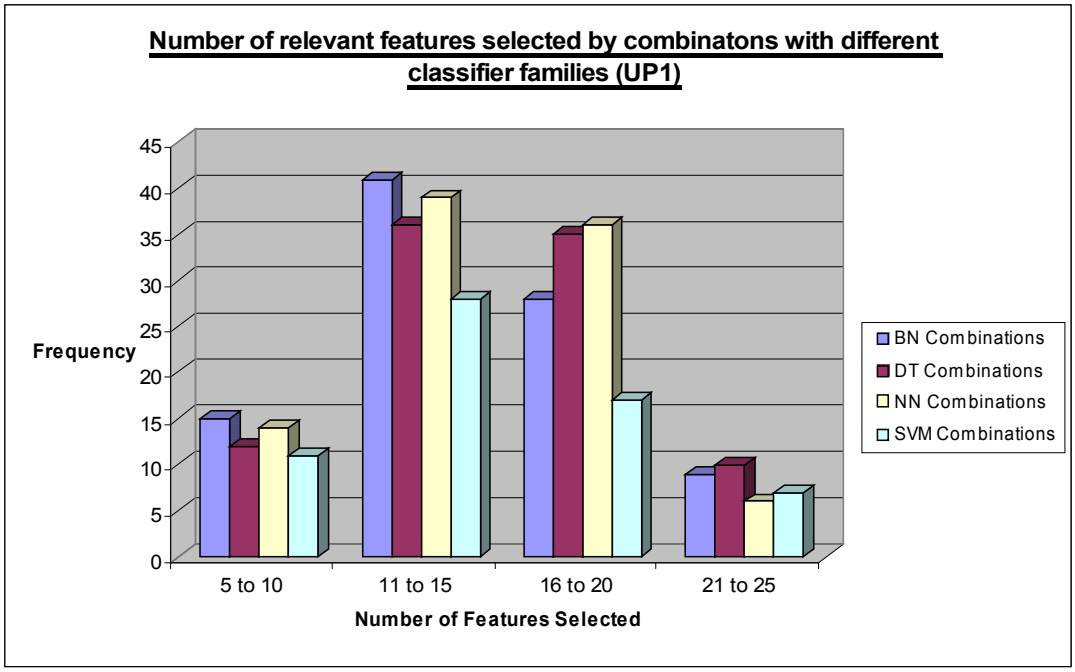

Figure 5.3. Number of Relevant Features Selected by Combinations with Different Classifier Families for UP1

An analysis of combinations with BN family classifiers, namely BNC, NB and AODE, is also carried out to determine how these classifiers influence the number of features selected. It was found that combinations with each of these classifiers selected different numbers of features. On the one hand, we found that combinations with BNC selected the lowest number of relevant features. In fact, 29 out of the 31 combinations with BNC selected feature subsets that contained lower number of features than the total mean number of relevant features selected by all mixed-type combinations (which was found to be 15). Moreover, ranking the number of features selected by all mixed-type combinations from the highest number (top half) to lowest number (bottom half) also showed that 28 of the BNC combinations appeared in the bottom half of the ranking. On the other hand, combinations with either the NB or AODE classifiers identified the highest number of relevant features. This was shown by that fact that 20 of the 31 combinations with NB classifier selected feature subsets that contained a higher number of features than the total mean number of relevant features. Similarly, 20 of the 31 combinations with AODE also selected number of features higher than the total mean number of features. In addition, it was found that 17 of the NB combinations and 20 of the AODE combinations appeared in the top half of the ranking.

An explanation for such differences in the number of selected features may be attributed to the nature of BN classifiers. Basically, the nature of classifiers from the BN family typically requires them to calculate the conditional probabilities of features in the dataset in order to find those features with the highest conditional probability values. This is because features with the highest conditional probabilities will be more relevant with respect to the target variable (Liao, et al, 2006). However, each BN classifier determines the conditional probability values of features in a different way. On the one hand, the NB and AODE classifiers simply calculate the conditional probabilities of each feature in relation to the target variable (and one other feature in the case of AODE). The features with the highest probabilities are then selected and used by the classifiers.

On the other hand, the BNC employs a more complex approach for determining the conditional probability values of features and finding those features that are most relevant. The BNC typically employs a heuristic search strategy (i.e., simulated annealing) as a way of searching through all the features in the dataset (Grossman and Domingos, 2004). The classifier also incorporates a scoring metric (i.e., Minimum Description Length) which judges the quality of the features selected by the search. The scoring metric works by penalising the classifier if features with low conditional probabilities (Wong, Lam, and Leung, 1999; Vinciotti, et al., 2006) or features that have complex interactions with other features are selected during the search. The fact that the scoring metric can penalise the classifier during the search suggests that fewer features may be found from the search and used by the BNC. As such, few features may be selected by the BNC during feature selection. This may therefore explain why combinations with BNC selected fewer relevant features than combinations with NB or AODE.

3) Combinations with NN Family Classifiers Influence Number of Relevant Features Selected from UP2

In the UP2 dataset, it was found that combinations with NN family classifiers influenced the number of relevant features selected. Figure 5.4 illustrates the influences of such combinations. This figure presents the number of relevant features selected by combinations with BN, DT, NN and SVM classifiers. A close examination of this figure shows that, in general, there are more combinations with

NN family classifiers in number groups '0 to 2', '3 to 5', '9 to 11', '12 to 14' and '15 to 17' than combinations with BN, DT and SVM family classifiers. This means that combinations with NN family classifiers selected many feature subsets comprised of low number of features but also selected many feature subsets comprised of high number of features. This shows that NN family classifiers caused differences in the number of features selected, which may suggest that combinations with NN family classifiers have some influence on the number of relevant features selected. In order to identify how combinations with classifiers from the NN family, including NNC, KNN and K*, influence the number of features selected, we perform a deep analysis of such combinations.

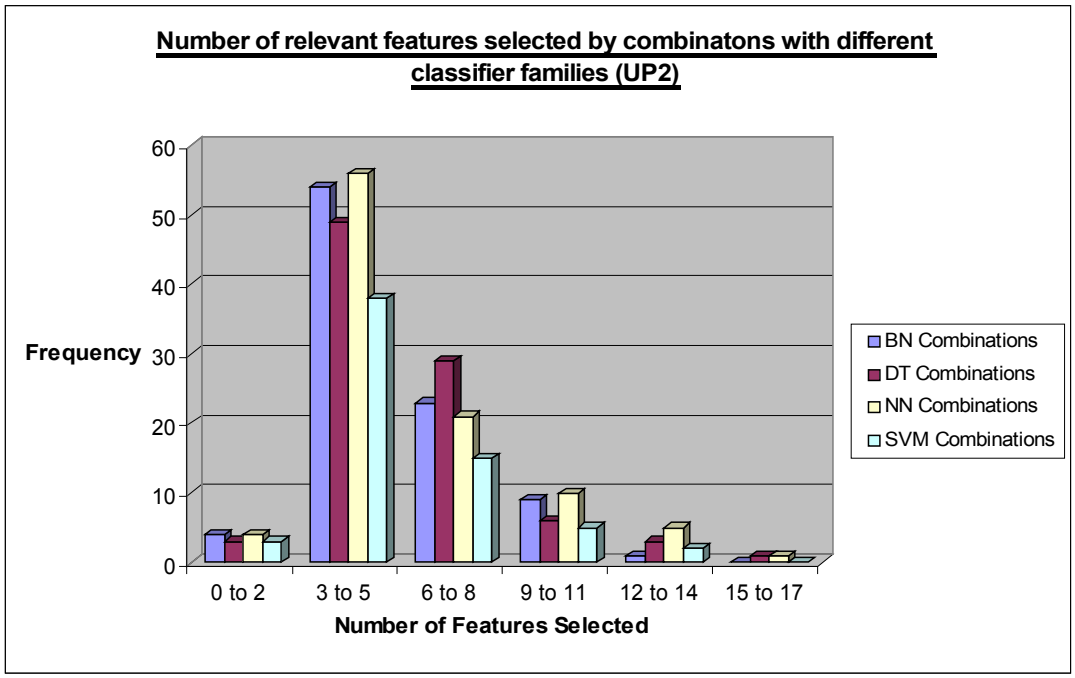

Figure 5.4. Number of Relevant Features Selected by Combinations with Different Classifier Families for UP2

The results from the deep analysis revealed differences in the number of features selected by combinations with the three NN family classifiers. It was found that combinations with NNC and K* selected very low numbers of features from the UP2 dataset. In terms of the combinations with these classifiers, it was uncovered that 28 out of the 31 combinations with NNC and 27 out of the 31 combinations with K* classifier selected feature subsets that had lower number of features than the total mean number of relevant features selected by all mixed-type combinations, which was found to be 5.76. Furthermore, the ranking of the number of features selected by all combinations showed that 28 of the 54 NN family combinations that appeared in the

bottom half of ranking included NNC and 26 of the 54 NN family combinations included the K* classifier. However, combinations with the KNN classifier selected very high numbers of features. In terms of the combinations with this classifier, it was found that 28 of the 31 KNN combinations selected number of features much higher than the total mean number of relevant features. In addition, we found that 30 of the 38 NN family combinations that appeared in the top half of ranking included the KNN classifier.

The reason why combinations with the NNC and K* classifiers selected lower number of features than combinations with KNN may lie within the number of neighbours employed by NN family classifiers to identify the relevant features. Basically, the NN family classifiers determine relevant features by using some distance metric (e.g., Euclidean distance), where the most relevant features are those that are deemed the closest by the distance metric and the least relevant are those with the furthest distance (Chrysostomou, Chen, and Liu, in press b). However, the number of most relevant features identified greatly depends on a small number of features known as neighbours (Han and Kamber, 2006). This is because the number of neighbours employed represents number of features used to determine the most relevant (i.e., closest) features. If few neighbours are employed by a NN family classifier then there is a high likelihood that fewer relevant features will be selected because fewer features will be used to determine the relevant features. On the other hand, using more neighbours can help select a higher number of relevant features.

The number of neighbours employed usually depends on the classifier used. In terms of the NNC and K* classifiers, these two classifiers typically employ a very small number of neighbours, e.g., one or two neighbours, when determining relevant features. On the other hand, the KNN classifier usually employs a larger number of neighbours when identifying the relevant features. Since the KNN classifier used a higher number of neighbours for feature selection than that of NNC and K*, it was able to select a higher number of features. This may help explain why combinations with the KNN classifier resulted in more relevant features being selected than combinations with NNC and K* classifier.

The abovementioned results have shown that the number of classifiers and nature of classifiers influence the number of relevant features identified by mixed-type

combinations. It was found that combinations with few classifiers were able to identify a higher number of relevant features than combinations with many classifiers. In addition, the results showed that nature of BN family classifiers and NN family classifiers led to differences in the number of features selected from the UP1 and UP2 dataset, respectively. The number and nature of classifiers thus influence number of features selected. The next section determines the effects of the number and nature of classifiers on the classification accuracies of selected features.

5.3.2 Accuracy Levels of Relevant Features

In this section, we determine the influences of number and nature of classifiers on the classification accuracies generated using the features identified by the mixed-type classifier combinations. Analysis of the accuracies from both UP1 and UP2 datasets revealed several findings. These findings are detailed in the following pages.

1) 3-classifier Combinations Generate Highest Classification Accuracies

The classification accuracies generated by all mixed-type combinations were examined. It was found in both UP1 and UP2 datasets that 3-classifier combinations were able to generate accuracies much higher than the majority of the other combinations. With respect to the UP1 dataset, 3-classifier combinations were found to generate a mean accuracy level (83.75%) higher than the mean accuracy levels of 2-classifier (82.01%) and 4-classifier (82.50%) combinations. With respect to the UP2 dataset, the mean accuracy level of 3-classifier combinations (95.37%) was also found to be higher than the mean accuracy levels of 2-classifier (94.27%) and 4-classifier (94.81%) combinations.

The accuracies of combinations, including those of 3-classifier combinations, for both UP1 (Figure 5.5) and UP2 (Figure 5.6) are also visualised to help see differences in the accuracies generated. More specifically, Figures 5.5 and 5.6 show the number of classifier combinations that generated accuracy levels higher and lower than the total mean accuracy level for the UP1 and UP2 datasets. The total mean accuracy level of UP1 is 83% and the total mean accuracy level of UP2 is 94.90%. The two figures show that the majority of 3-classifier combinations generated accuracy levels above the total mean accuracy while few generated accuracies below total mean accuracy. In addition, the figures also show that there are generally fewer 2-classifier and 4-

classifier combinations above the total mean accuracy levels but more of such combinations below the total mean accuracy level. The findings from these figures suggest that majority of 3-classifier combinations generated high classification accuracies. More importantly, it also suggests that 3-classifier combinations generated accuracies higher than the accuracies generated by 2-classifier and 4-classifier combinations.

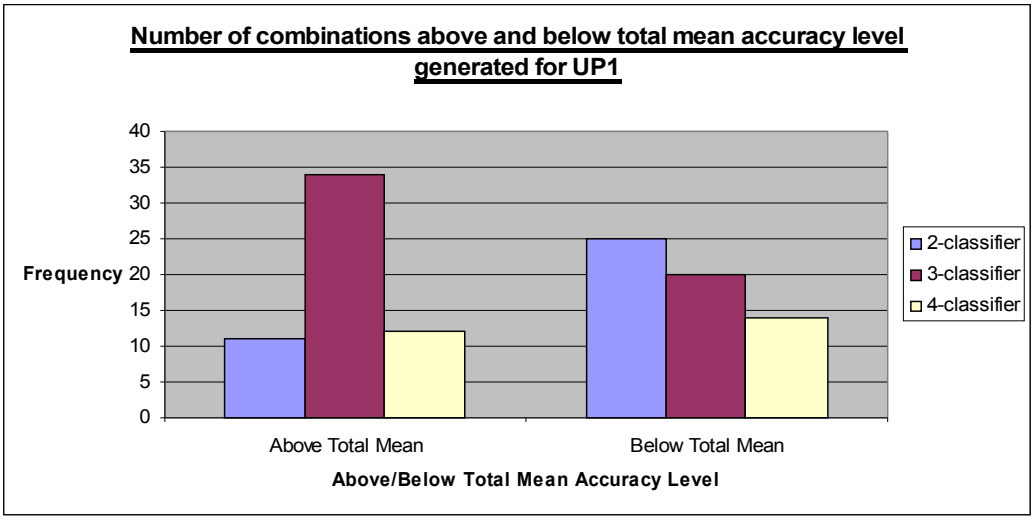

Figure 5.5. Classification Accuracies Generated by 2-Classifier, 3-Classifier, and 4-Classifier Combinations for UP1

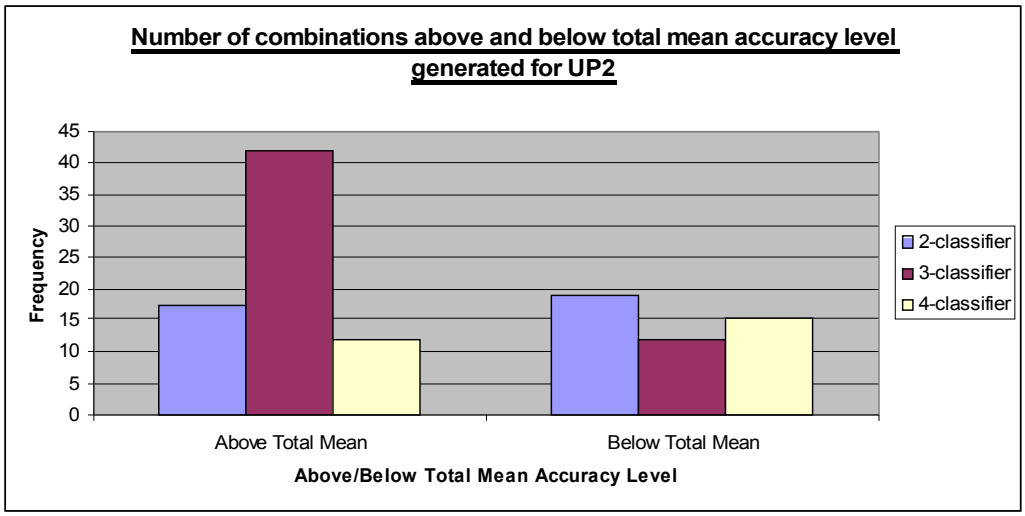

Figure 5.6. Classification Accuracies Generated by 2-Classifier, 3-Classifier, and 4-Classifier Combinations for UP2

In order to gain a better understanding of the accuracies generated by 3-classifier combinations, we carry out a deep analysis of the accuracies obtained from the UP1 and UP2 datasets. The analysis involved ranking the accuracies of all combinations from highest (top) to lowest (bottom) for both datasets. In terms of UP1, the analysis

showed that 34 of the 59 combinations in the top half of the ranking were in fact 3-classifier combinations. In terms of UP2, the analysis uncovered that 41 out of the 54 3-classifier combinations were in the top half of the accuracy ranking. All in all, the findings from UP1 and UP2 datasets showed that 3-classifier combinations were able to produce higher accuracy levels than other combinations. This suggests that 3-classifier combinations are more suited to selecting most accurate feature subsets than 2-classifier and 4-classifier combinations.

1) The Nature of BN Classifier Family Influences Level of Accuracy For UP1

An analysis of combinations with BN family classifiers, namely BNC, NB and AODE, was carried out to determine the influences of these classifiers. On the one hand, it was found that mixed-type classifier combinations with BNC generated the lowest levels of accuracy. More specifically, 25 out of the 31 combinations with BNC generated accuracies lower than the total mean accuracy of entire set of mixed-type combinations (i.e., 83%). Furthermore, it was found that 25 of the BNC combinations appeared in the bottom half of the accuracy ranking. On the other hand, we found that classifier combinations with the NB and AODE classifier generated the higher levels of accuracy. In detail, 20 of the 31 combinations with NB generated accuracies higher than the total mean accuracy. Similarly, 21 of the 31 combinations with AODE also produced accuracies higher than the total mean accuracy level. In addition, it was found that 19 of the NB combinations and 21 of the AODE combinations appeared in the top half of the accuracy ranking. According to these findings, features selected by combinations with BNC are less accurate with respect to the target variable than features selected by combinations with NB or AODE.

A possible explanation for such differences in accuracies may have to do with the number of features considered by BN family classifiers when building the network structures. The purpose of network structures is to show relationships among features in the dataset (Su and Zhang, 2006). In general, the number of features used to form the structures depends on the degree to which the conditional independence assumption of BN classifiers is enforced. A higher number of features used suggest that this assumption is weakly supported by the classifier while a lower number of features suggest that it is strongly supported by the classifier. In terms of the BN family classifiers used in the mixed-type combinations, the BNC usually considers a

large number of features when identifying relationships due to the fact that it weakly supports conditional independence among features. On the other hand, classifiers like NB and AODE consider a limited number of features in relation to the target variable because they strongly enforce the conditional independence. Because of these differences, the BNC may allow the network structure to grow without any stringent bounds since many features are involved in the structure building process, whereas the NB and AODE classifiers may impose some restrictions on the structure (Jiang, et al., 2007). Allowing a network to grow relatively unrestricted, as in the case of the BNC, may lead to overfitting of the data where the network structure includes features that are of little relevance to the target variable. Including such features may cause the BNC to have poor classifier performance and thus produce lower accuracy levels. This may explain why features selected by combinations with BNC generated low accuracies in comparison to combinations with NB or AODE.

Overall, BN family classifiers were found to significantly influence the accuracy levels of selected features. In addition, Section 5.3.1 also showed that BN family classifiers influence the number of relevant features selected. In summary, these findings suggest that the nature of BN family classifiers has substantial effects on the number of features selected and the accuracy levels of features. Such findings may be useful in choosing suitable classifiers for feature selection. On the one hand, one may want to avoid using the BNC since it leads to few features and these features are less relevant to the target variable. On the other hand, one may want to adopt the AODE or NB classifiers because these classifiers are able to select many features that are highly relevant to the target variable.

1) The Nature of NN Classifier Family Influences Level of Accuracy For UP2

A deep examination of combinations with each of the NN family classifiers revealed some interesting results. On the one hand, the results showed that combinations with the NNC or K* classifiers generally produced high accuracy levels. This was shown by the fact that 28 of the 31 combinations with NNC and 20 of the 31 combinations with K* produced classification accuracies above the total mean accuracy level. The ranking of all accuracies also showed that 24 of the NNC combinations and 19 of the K* combinations appeared in top half of ranking. On the other hand, results showed that combinations with the KNN classifier generated low accuracy levels. In fact, 17

of the 31 KNN combinations selected features that led to accuracies below the total mean accuracy and 19 of the 31 KNN combinations were found in the bottom half of ranking.

These findings suggest that combinations with NNC, K*, and KNN selected feature subsets that led to different levels of accuracy. The differences found may be attributed to the nature of these three classifiers. The NNC classifies a new unknown instance *n* by looking at the features and class value of the single closest instance to *n* (i.e., the single nearest neighbour to *n* is used). The K* classifies *n* by looking at the features of a small number of instances closest to *n*, i.e., it may use two or three closest neighbours. In contrast, the KNN classifies *n* by looking at the features and class values of several closest instances, i.e., several neighbours are used. In other words, the NNC and K* classifiers usually make use of a small number of neighbours when determining the class of *n* while the KNN classifiers makes use of a larger number of neighbours. However, using a large number of neighbours for establishing the class of *n,* as done by KNN, may lead to some problems.

In general, using a large number of neighbours may make it more difficult to determine and assign a class value to *n* compared to using the single closest neighbour or even a very small number of neighbours. This is because there will be more competition among the neighbours when deciding on the class value of *n.* As a result of more competition, it is possible that the wrong class value may be assigned to *n* (Larose, 2005). Assigning the wrong class value may cause the classifier to have poor performance, which may subsequently cause the classifier to miss out features that are highly relevant to the target variable. This in turn may lead to feature subsets that contain features of low relevance and that generate low accuracy levels. This may explain why features selected by combinations with KNN generated lower accuracy levels in comparison to combinations with NNC and combinations with K*.

The above finding shows that the number of neighbours employed by NN family classifiers influence the accuracy levels of selected features. Interestingly, the number of neighbours used by these classifiers was also found to affect the number of features selected (Section 5.3.1). The findings from these two sections suggest that the number of neighbours used for each NN family classifier may affect the way in which features

are selected. Thus, it may be worthwhile using different numbers of neighbours with each NN family classifier. In this way, we can improve the chances of finding subsets that include features most relevant to the target variable.

5.3.3 The Influences of BN Family and NN Family Classifiers on Feature Selection

The results from the previous two sections found that certain classifier families influenced the feature selection results of UP1 and UP2. More specifically, combinations with BN family classifiers were found to influence the number of relevant features and their accuracy levels of the UP1 dataset. On the other hand, combinations with NN family classifiers influenced the number of relevant features and their associated accuracies of the UP2 dataset. A plausible explanation for why these two classifier families influenced the feature selection results may lie within the way they consider each feature during feature selection.

Classifiers belonging to the BN family usually assign a weight to each feature so as to determine the relevance of each feature. The weights of features are typically provided in the form of prior knowledge, which is provided by experts in the field (Castelo and Siebes, 2000). Prior knowledge can be used to determine those features that are more relevant than the others. For example, features with high weights are seen as very relevant features while features with low weight values are seen as less relevant with respect to the target variable. However, prior knowledge about the features is not always available. In this thesis, no prior knowledge was available about the features in both UP1 and UP2 datasets, except the data themselves. In the event that little or no prior knowledge is available, the BN family classifiers assign each feature the same weight (Jiang, et al., 2007). As a result, each feature is considered equally relevant to the target variable. Classifiers from the NN family also assign weights to each feature in the dataset so as to distinguish the most relevant features from the least relevant features. This is usually done by considering the distance of the features. Those features which are closest are assigned a high weight and those features that are further are assigned a low weight. When no feature weighting is used, as done by the NN family classifiers used in this thesis, the classifiers consider each feature to be equally relevant to the target variable (Ghosh, 2006).

The NN family classifiers and BN family classifiers used in this thesis treated every feature with equal relevance, i.e., each feature was assigned equal probability. When features are treated equally then every feature has a chance of being selected. This may mean that the BN and NN classifier families offer each feature a similar chance of being selected which may result in similar effects on feature selection. Such similarities among the BN and NN family classifiers may help explain why they were both found to influence feature selection results of the mixed-type combinations. However, the influences of classifiers belonging to these two families were apparent in different datasets. On the one hand, BN family classifiers influenced results of UP1 dataset. On the other hand, NN family classifiers influenced results of UP2 dataset.

In terms of BN family classifiers, there are two possible reasons that may explain such findings. The first reason relates to the fact that BN classifiers typically look for features with the highest probability values since these features are most relevant to the target variable. The possibility of identifying such features increases when there are many features in the dataset as there is a wider selection of features from which to choose from. Once identified, these features are used to build a graphical network structure, which shows how these features are related to the target variable. It is this network structure which shows the relevancies within the dataset and thus determines the number of features that are selected and the accuracy levels generated by BN family classifiers. As such, influences of BN family classifiers on feature selection may be more obvious when network structures are built from datasets with large number of features. This may explain why BN family classifiers influenced feature selection results of UP1, which includes a large number of features.

The second reason relates to the conditional independence assumption typically made by BN family classifiers. Basically, BN classifiers assume that features within a dataset are conditionally independent of one another, i.e., two features are independent given another feature (Ling and Zhang, 2002). If features in a dataset do not satisfy this conditional independence assumption then BN family classifiers may perform poorly and thus may not cause differences in feature selection results. If, however, some or all features are conditionally independent then BN family classifiers are more likely to provide good performance and in turn show their influences on feature selection results. The fact that BN family classifiers influenced

feature selection results of UP1 may possibly suggest that features within this dataset were conditionally independent and thus satisfied the assumption made by BN classifiers. Hence, this is another reason why BN family classifiers influenced feature selection results of UP1 dataset.

In terms of NN family classifiers, a possible reason for the finding may have to do with the feature space of dataset. These types of classifiers usually consider the entire feature space of the dataset (i.e., number of features in dataset) to perform classification and determine relevant features. In addition, NN family classifiers must also compute the distances of all features and instances in the dataset so as to find which are relevant (i.e., closest) and which are not so relevant (i.e., furthest away). When the feature space of the dataset is large, the distances among the features may be very diverse which may make relevant and irrelevant features more difficult to distinguish. However, when the feature space of dataset is small, NN family classifiers will have to compute and handle the distances of fewer features. A small feature space may therefore enable NN family classifiers to clearly distinguish relevant features from irrelevant features. With this ability, NN family classifiers can show their influences on feature selection results. In other words, NN family classifiers may be able to show their influences on feature selection more clearly when there are few features present in dataset. This may explain why NN classifiers influenced feature selection results of UP2 dataset, which has small number of features.

5.3.4 Relationships Among Number of Features Selected and Accuracy Levels Generated

The mixed-type combinations have been shown in the previous sections to select different numbers of features and generate different levels of accuracies. In this section, we study the potential relationships between the number of features and accuracy levels of the combinations. To uncover such potential relationships we use two tables to present the number of features selected by the mixed-type combinations and the accuracies that these features generated. On the one hand, Table 5.4 shows the mean number of relevant features selected by mixed-type combinations which generated lowest, intermediate and highest accuracy levels for UP1 dataset. On the other hand, Table 5.5 shows the mean number of relevant features selected by mixed-

type combinations which generated lowest, intermediate and highest accuracy levels for the UP2 dataset.

	Accuracy Levels		
	Lowest	***Intermediate***	***Highest***
2-classifier	15.33	17.52	19
3-classifier	10	15.70	16.71
4-classifier	11.50	11.73	12
Total	36.83	44.95	47.71

Table 5.4. Number of Features Selected and Accuracy Levels of Features for UP1 Dataset

	Accuracy Levels		
	Lowest	***Intermediate***	***Highest***
2-classifier	9.71	5.86	-
3-classifier	8	6.38	4.78
4-classifier	-	4.59	-
Total	17.71	16.38	4.78

Table 5.5. Number of Features Selected and Accuracy Levels of Features for UP2 Dataset

Table 5.4 shows a positive relationship between number of features selected and accuracy levels. However, Table 5.5 shows a different aspect. Table 5.5 shows a negative relationship between the number of features selected and their accuracy levels. A plausible explanation for such a difference may lie within the classifiers used in the combinations. In terms of the UP1 dataset, it was previously found that BN family classifiers influenced both the number of features selected by combinations and the accuracy levels of the features. In fact, BNC was found to select low number of features and generate low accuracy levels whereas the NB and AODE classifiers were found to select high number of features and generate high accuracies. A close examination of Table 5.4 and Table 5.1, which presents the results from UP1, shows that the majority of combinations which selected low number of features with low accuracies included the BNC. Conversely, we found that majority of combinations which selected a high number of features and generated high accuracies included the NB and AODE classifiers. The differences among combinations with BNC, NB and AODE may help explain why we found a positive relationship between number of features selected and their associated accuracy levels for UP1.

In terms of the UP2 dataset, results from the previous sections showed that NN family classifiers influenced the number of features selected and the accuracies of the features. The NNC and K* classifiers were found to select few features but generate

high accuracy levels. On the other hand, KNN classifier was found to select a high number of features but generate low accuracy levels. Examining Table 5.5 and the results shown in Table 5.2, uncovered that many of the combinations which selected high number of features and generated low accuracies included the KNN classifier. Moreover, it was found that many of the combinations which selected low number of features and led to high accuracy levels included either the NNC or K* classifiers. Differences among the combinations with these classifiers may help explain why Table 5.5 shows a negative relationship between the number of features selected from UP2 and their associated accuracy levels.

These findings from Tables 5.4 and 5.5 seem to suggest that the classifiers used in the mixed-type combinations have some influences on the relationships between the number of features and accuracy levels of each feature subset selected. More specifically, BN family classifiers were found to influence the relationships of UP1 while NN family classifiers were found to influence relationships of UP2. An explanation for these differences may lie within the ability of such classifiers to influence feature selection results. As previously explained in Section 5.3.3, the influences of BN family classifiers on feature selection were more obvious in the UP1 dataset. Due to this issue, classifiers belonging to the BN family may also be responsible for influencing the relationships among features selected from the UP1 dataset. On the other hand, the influences of NN family classifiers were found to be clearer in the UP2 dataset, which may explain why NN family classifiers also caused differences in the relationships observed within this dataset.

The results obtained so far in this section have shown that the number and nature of classifiers used in mixed-type combinations greatly influence the number of relevant features selected in addition to the accuracy levels of selected features. More interestingly, we have found that the nature of BN and NN family classifiers influence these two issues.

5.4 Visualising Features with Decision Trees

This section examines the decision trees of combinations with the highest accuracy levels. Decision trees with the highest accuracy levels are most likely to include the most relevant features with respect to the target variable. Analysing decision trees

with the highest accuracy levels will thus help uncover the most relevant features and the most relevant relationships between the features and the target variable. The analysis of decision trees in this section is divided into two parts. The first part analyses the decision tree with the highest accuracy from the UP1 dataset. The second part analyses the decision trees with the highest accuracies from the UP2 dataset.

5.4.1 Decision Tree(s) of UP1 Dataset

1) Analysis of Features Selected by NB+CN2+K Classifier Combination*

The results from the UP1 dataset showed that the NB+CN2+K* classifier combination generated the decision tree with the highest level of classification accuracy (90.83) among all mixed-type combinations. The fact that this combination led to the highest accuracy implies that it selected features that are of highest relevance to the target variable. In this section, we present the relevant features selected by this combination. The relevant features selected by this combination are shown in Table 5.6. Each of the features presented in Table 5.6 has an associated level of relevance shown in parenthesis. This relevance value indicates how relevant a feature is to the target variable, i.e., users' level of computer experience. As an example, consider features 'Q14' and 'Q15'. Although both features were selected by this classifier combination, they differ in their relevance to determining users' level of computer experience. One the one hand, 'Q14' is selected with a relevance value of 10. On the other hand, 'Q15' is selected with a relevance value of 1. These relevance values indicate that 'Q14' is much more relevant to determining users' level of computer experience than 'Q15'.

	Selected Features and Their Relevance Values
NB+CN2+K* Classifier Combination	Q1 (1), Q3 (1), Q14 (10), Q15 (1), Q23 (1), Q24 (2), Q28 (1), Q29 (2), Q31 (8), Q33 (2), Q38 (1), Q42 (1), Q48 (1), Q56(5), Q58 (2), Q63 (1), Q70 (1).

Table 5.6. Features Selected by NB+CN2+K*

A closer look at the features and relevance values in Table 5.6 shows that 'The results are presented by the levels of the relevance' (Q14), 'There are not too many types of icons' (Q31) and 'The options that are used less frequently are located in less-convenient positions' (Q56) are particularly relevant. This suggests that these three features are highly relevant with respect to the target variable (i.e., users' level of computer experience). The three relevant features, in addition to the other relevant

features selected by the NB+CN2+K* combination, were used to construct the decision tree. The constructed decision tree is illustrated and explained in the next few pages.

2) Constructed Decision Tree

This section presents the decision tree formed using features selected by the NB+CN2+K* combination, which is in Figure 5.7. The decision tree formed comprises of three levels where the first level indicates the most important feature (the root node) while the remaining levels indicate other important features. The decision tree shown in Figure 5.7 also includes the number of users that follow each level of computer experience (see the key of the decision tree). For example, B (2), which can be found on the far left of the second level of the decision tree in Figure 5.7, signifies that 2 users who found Q31 strongly unimportant and found Q15 strongly unimportant had average level of computer experience. Examining the decision tree in Figure 5.7 reveals two interesting issues, which are detailed over the next few pages.

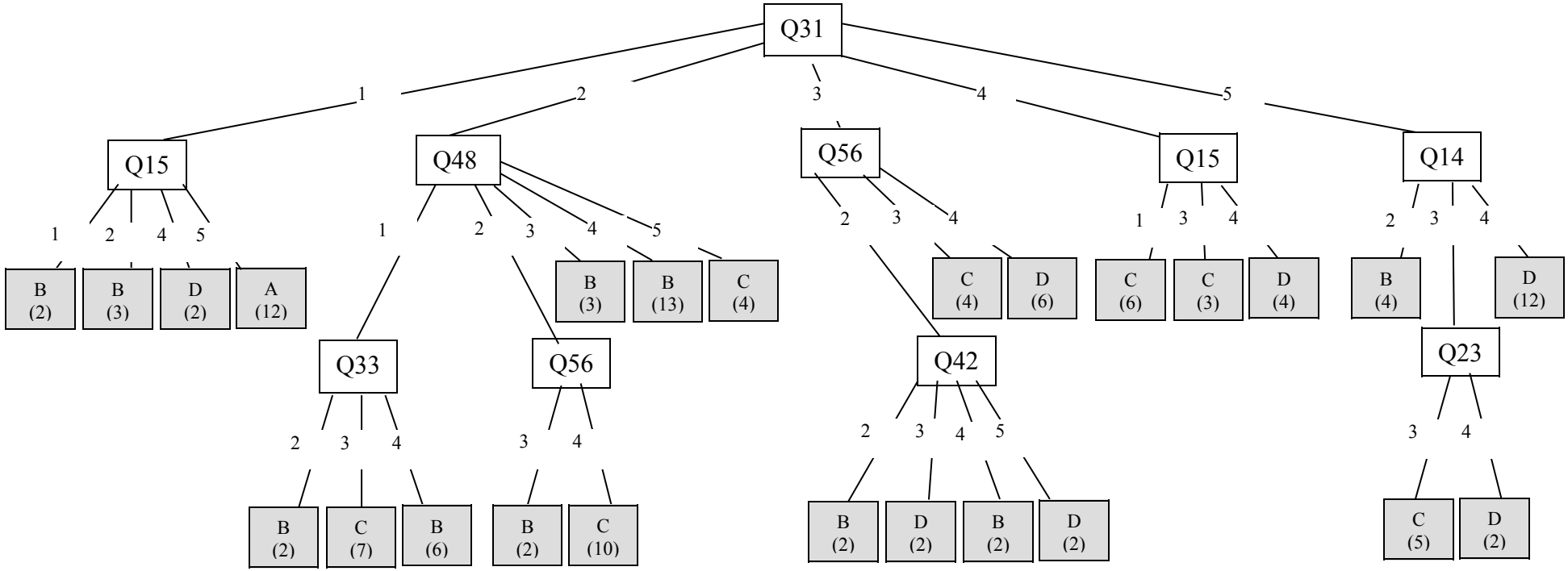

Figure 5.7. Decision Tree for NB+CN2+K* Classifier Combination

Decision Tree Key	
Users' Preferences	*Level of Computer Experience*
1 = Very Unimportant	A = Little
2 = Unimportant	B = Average
3 = Neutral	C = Good
4 = Important	D = Excellent
5 = Very Important	

The first issue found from Figure 5.7 relates to the root node of the decision tree. The figure clearly shows that Q31 is the root node of the decision tree, which implies that this feature is considered as the most important feature by the decision tree classifier. Interestingly, this finding is different to those found when considering the features selected by the NB+CN2+K* combination. As previously shown in Table 5.6, Q14 was found to be the most relevant feature among all selected features because it had the highest relevance level whereas Q31 was the second most relevant feature. This shows that Q14 was more relevant than Q31. The reason why Q31 was chosen as the root node of the tree, instead of Q14, may have to do with the statistical significances of these features. As previously stated in Chapter 4, Q31 had a higher statistical significance than Q14. This means that Q31 was statistically more relevant than Q14 with regards to target variable. Interestingly, decision tree classifiers also use and rely on the statistical significances of features to determine the positions of features when building the tree. The fact that Q31 had a higher significance than Q14 suggests that it would be positioned higher up in the tree. This may therefore explain why Q31 is the root node of the decision tree and Q14 is found in the lower level of the decision tree.

The second issue relates to some of the features used in the decision tree. On close examination of the decision tree, four features stand out from the rest because they were found to differentiate the preferences of users with low levels of computer experience and those with high levels of computer experience. The four features included: Q31, Q14, Q48, and Q56. The first of the four features, Q31, is a very important feature with regards to users' level of computer experience because it is the root node of the decision tree. The other three features are important because they have a high number of users associated with them as shown in parenthesis in the decision tree. The implications of these four features for determining users' level of computer experience are explained in detail in Table 5.7.

Feature	Findings from Decision Tree	Explanation of Findings
Q31 - There are not too many types of icons (root node)	All users with little computer experience and many of the users with average computer experience found this feature unimportant or very unimportant. However, the majority of users with good and excellent levels of computer experience considered it important or very important.	Users with low levels of computer experience typically possess less knowledge than users with high levels of computer experience. In addition, users with low levels of computer experience would not have used many search engines in the past. Due to their limited knowledge and inexperience regarding search engines, users with low levels of computer experience may not be familiar with all the functionalities provided by search engines. Thus, providing such users with a large selection of icons can enable them to easily and quickly differentiate between the various functionalities provided by the search engines.
Q14 - The results are presented by the levels of the relevance	The majority of the users with high levels of computer experience found this feature important while users with low levels of computer experience considered this feature unimportant.	The findings regarding this feature suggest that users with higher levels of computer experience preferred search results to be presented by the relevance levels in order to determine the most important from the least important result. The reason for this preference may lie within users past experience with search engines. Users with higher levels of computer experience are more likely to have used different types of search engines before. The fact that they have used search engines before may make them more familiar with using functions provided by search engines, especially the ordering of search results, than those with less computer experience.
Q48 - Error messages let you know the cause of the problems	A large number of users with low levels of computer experience considered this feature important. In contrast, a large number of users with high levels of computer experience considered it unimportant.	As abovementioned, users with less computer experience possess limited amount of knowledge compared to users with more computer experience. As such, users with less computer experience are more likely to make more errors when searching for information online. These users thus preferred the search engine to clearly explain the errors that they made. Explaining errors will help such users better understand the reasons for the errors occurring. By better understanding the errors, users will be able to rectify the errors by finding solutions.
Q56 - The options that are used less frequently are located in less-convenient positions	Users with less computer experience considered this feature unimportant and those with more computer experience found it important and very important.	Generally speaking, individuals with low levels of computer experience do not possess enough knowledge to determine the most effective way for completing some tasks. As such, they may consider that all options are equally used, which may explain why such users did not prefer options to be in different positions. On the contrary, individuals with higher levels of computer experience possess the relevant knowledge necessary to identify which options are of most relevance. In this way, they can focus their attention on using a small subset of the most relevant options to complete their searching tasks.

Table 5.7. Findings Relating to Q31, Q14, Q48 and Q56 Features from Decision Tree

5.4.2 Decision Tree(s) of UP2 Dataset

1) Analysis of Features Selected by Classifier Combinations

In this section, we present the features selected by the classifier combinations which generated the highest accuracies among all mixed-type combinations for UP2. In total, there were nine combinations which generated the highest accuracy of 96.92%. The combinations are shown in Table 5.8.

		Feature selected by combinations									
		Q2	***Q5***	***Q6***	***Q9***	***Q11***	***Q13***	***Q16***	***Q18***	***Q19***	***Q20***
CC1	***BNC+C4.5+K****				(10)		(1)		(10)	(1)	
CC2	***NB+C4.5+NNC***		(1)		(10)				(10)	(1)	
CC3	***NB+CART+NNC***		(1)	(1)	(10)			(1)	(3)	(1)	
CC4	***AODE+C4.5+NNC***				(10)	(1)	(1)		(10)	(1)	
CC5	***AODE+CART+NNC***			(1)	(10)	(1)			(10)	(1)	
CC6	***AODE+C4.5+K****				(10)	(1)	(1)		(10)	(1)	(1)
CC7	***BNC+CN2+SVMpoly***	(1)			(10)		(1)		(7)	(2)	
CC8	***BNC+NNC+SVMpoly***			(1)	(10)				(9)	(1)	
CC9	***C4.5+NNC+SVMpoly***			(1)	(10)				(10)	(1)	

Table 5.8. Features Selected by Classifier Combinations with Highest Accuracy

A close look at these combinations reveals an interesting finding. The finding relates to the fact that the combinations have some similarities. The similarities lie within the presence of some common classifiers across the combinations. An examination of the classifiers used in each combination revealed that six of the nine combinations included the NNC which belongs to the NN classifier family. Combinations with NNC were previously shown in Section 5.3.2 to select features that generated very high levels of accuracy. In addition, we found that two of the nine combinations included the K* classifier, which also belongs to the NN family. Combinations with this classifier were also found to generate very high levels of accuracy in Section 5.3.2. The remaining combination did not include NNC or K* but included another classifier worth noting. The classifier was SVMpoly which belongs to the SVM family. The reason why this combination with SVMpoly generated high accuracy may be attributed to its similarities with NN family classifiers.

As previously stated, NN family classifiers and SVM family classifiers are somewhat similar in that they rely on distances to determine the relevance of a feature. As such, they may select similar features which may subsequently lead to similar accuracy levels. These similarities may therefore explain why the combination with SVM generated same high accuracy like those of combinations with NNC and K*. In addition, the similarities between NN family (i.e., NNC and K*) and SVM family (i.e., SVMpoly) may also help explain why combinations which included these particular classifiers generated identical accuracy levels.

The relevant features selected by these nine combinations are also shown in Table 5.8. Each of the relevant features presented in this table has a relevance value which is indicated through use of parenthesis. Examining the relevance values of the selected features reveals that majority of the features have low relevance values, i.e., relevance of 1. In other words, most of the features selected are of little relevance to the target variable (i.e., users' cognitive style). However, Q9 and Q18 show different relevance values. It was found that 'It is hard to use the back/forward buttons' (Q9) was assigned the highest relevance level of 10 by all nine 3-classifier combinations. This suggests that Q9 is the most relevant feature with respect to the target variable. Furthermore, the table shows that 'It is easy to find a route for a specific task with the index' (Q18) was the second most relevant feature with a relevance value that varied from 3 to 10. It is also worth noting that these two features were selected by all of the nine combinations. In summary, these findings suggest that Q9 and Q18 are highly relevant to determining a user's cognitive style. In order to further examine the relevance of all selected features, including that of Q9 and Q18, and their relationships with the target variable, the next section looks at the decision trees constructed by the nine classifier combinations.

2) Constructed Decision Trees

The decision trees formed using the features shown in Table 5.8 are shown in Figure 5.8 and Figure 5.9. All decision trees have two levels where the first level includes a single feature that is the root node. In addition, each decision tree includes the number of users in the dataset that follow each type of cognitive style (see the key of Figure 5.8 for more

details on types of cognitive style). For example, consider the decision tree in Figure 5.8. In this decision tree, FD (3), which appears on the far right of the second level, means that the 3 users who strongly agreed with Q9 were Field Dependent. Analysing the decision trees shown in Figures 5.8 and 5.9 reveals several interesting findings. These findings are outlined on the following pages.

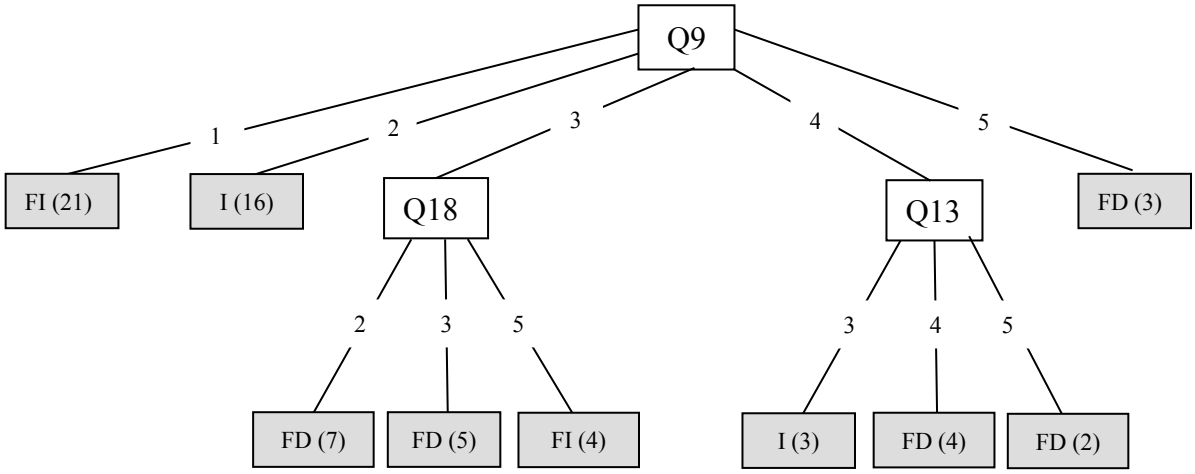

Figure 5.8. Decision Tree for CC1, CC4, CC6 and CC7 Combinations

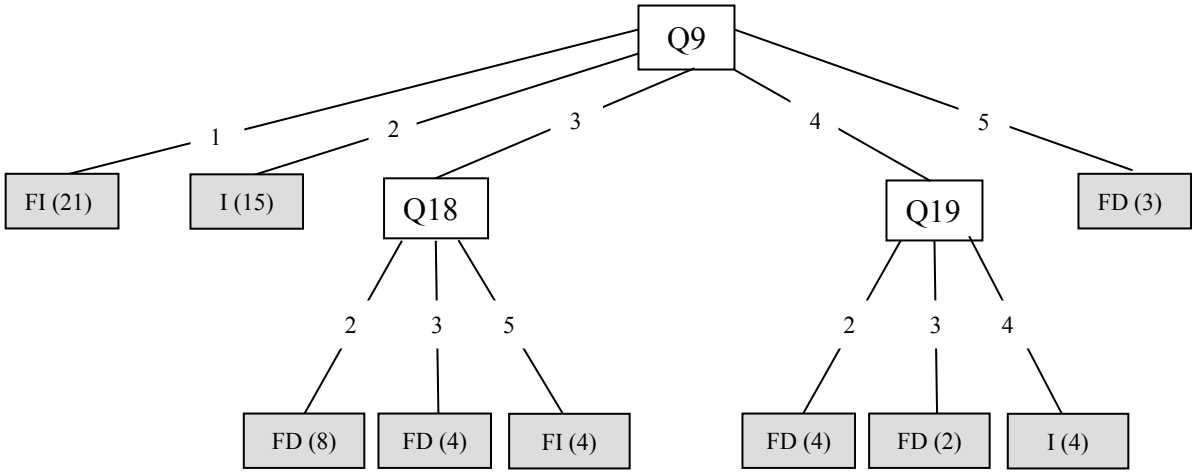

Figure 5.9. Decision Tree for CC2, CC3, CC5, CC8 and CC9 Combinations

Decision Tree Key	
Users' Preferences	*Cognitive Style*
1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree	FI = Field Independent I = Intermediate FD = Field Dependent

The most interesting finding relates to the construction of identical trees. A holistic view of all decision trees generated by the classifier combinations revealed that some of the combinations formed identical decision trees. On the one hand, the CC1, CC4, CC6, and CC7 combinations selected features that led to identical decision trees. On the other hand, the CC2, CC3, CC5, CC8 and CC9 combinations selected features that also led to identical decision trees. This is why only two decision trees are presented as opposed to nine. A plausible reason why half of the combinations led to one tree while the other half built another tree may have to do with the features selected by the classifier combinations.

A deep examination of the features selected by the nine classifier combinations revealed an interesting finding. The interesting finding related to Q13. On the one hand, the CC1, CC4, CC6, and CC7 combinations were found to select many features including Q13. These combinations led to the tree shown in Figure 5.8, which included Q13. On the other hand, the CC2, CC3, CC5, CC8 and CC9 combinations were found to select many features but not Q13. These combinations led to the tree shown in Figure 5.9, which excluded Q13. In essence, the combinations which selected Q13 produced one identical tree while the combinations which did not select Q13 produced another identical tree. A possible explanation as to why some combinations selected Q13 and other combinations did not select this feature may lie within the individual classifiers used in the combinations. A close examination of the features selected by the individual classifiers used in the combinations was conducted. The examination revealed differences among the features selected by classifiers from each classifier family.

In terms of the BN classifier family, the BNC and AODE were found to select many features including Q13, but NB classifier did not select this feature. In terms of the DT family, the C4.5 and CN2 classifiers selected Q13 but CART did not select it. However, NN family classifiers, namely NNC and K* did not select Q13 although KNN was found to select Q13. The SVMpoly was also a classifier that did not select Q13. The fact that individual classifiers differed in the features they selected, especially Q13, may explain

why some combinations did and did not select Q13 when building the trees. On the one hand, combinations comprised mainly of classifiers that selected Q13 (i.e., BNC, AODE, C4.5, CN2 and KNN) were more than likely to select Q13 and in turn use it to build the decision tree. On the other hand, combinations comprised mainly of classifiers that did not select Q13 (i.e., NB, CART, NNC, K* and SVMpoly) were more than likely to exclude Q13 from the decision tree building process. This may therefore explain why the combinations produced decision trees with different features.

As previously mentioned, combinations which selected Q13 were found to build one tree which included Q9, Q18 and Q13 (Figure 5.8). However, combinations that did not select Q13 were found to build another tree which included Q9, Q18 and Q19 (Figure 5.9). Interestingly, all nine of the combinations selected Q19. The fact that Q13 was used instead of Q19 for building the tree shown in Figure 5.8 may suggest that Q13 may be more relevant to the target variable than Q19. In order to further examine the significance of these two features, the ANOVA method was used. The output of ANOVA showed that the significance of Q13 ($F=21.31$, $p<.001$) is higher than the significance of Q19 ($F=11.88$, $p<.005$). This means that Q13 is statistically more relevant to the target variable than Q19. The fact that Q13 is more relevant than Q19 may explain why it was used in the decision tree building process.

The fact that classifier combinations were found to produce identical trees even though they selected different features suggests that the combinations were able to uncover a small subset of features that can help in identifying a user's cognitive style. In the case of the decision trees constructed by the combinations, the small subset of features comprises Q9, Q13, Q18 and Q19 because these features were the only features used to form the decision trees. A deep analysis of these four features is carried out so as to establish how they can help determine users' cognitive style. The results from this analysis are depicted in the next few pages.

The first and most important feature is 'It is hard to use the back/forward buttons' (Q9). This is because Q9 was the root node of all the decision trees. This implies that all decision trees deemed Q9 the most relevant feature with regards to users' cognitive

styles. Results from decision trees showed that nearly all of the Field Independent (FI) and Intermediate (I) users strongly disagreed and disagreed with this feature, respectively, while majority of Field Dependent (FD) users agreed or strongly agreed with it. A possible explanation for such different preferences may lie within the tendencies of users to navigate through the Web-based learning systems. On the one hand, users who follow the FI cognitive style typically prefer finding their own path or route when performing a particular task on the Web (Chen and Macredie, 2004). In this way, FI users would be more accustomed to using tools that can help them find a suitable path. Web-based learning systems include many different tools but an example of a tool that may help users find their own path through the tutorial is back/forward buttons. Such tools allow users to control where they have been and where they will go to next. This may explain why FI users preferred the back/forward tool. On the other hand, FD users prefer a guided approach. This means that such users prefer the system to find a path for them. As such, tools like back/forward buttons, which provide users with an opportunity to find a path, may not have appealed to FD users.

Another feature that appears in all identical decision trees is 'It is easy to find a route for a specific task with the index' (Q18). Interestingly, this feature was also found to be a common feature among all nine of the classifier combinations (as shown in Table 5.8) and was regarded as the second most relevant feature by the combinations. Examining Q18 within the decision trees uncovered that some FI users strongly agreed with this feature whereas several FD users disagreed with it. On the one hand, FI users are individuals who prefer to work on their own and use their own initiative to find and complete a task. As such, they may prefer to freely navigate the system and jump from one point to another within the system in order to find a suitable route for a particular task. The index is a tool that can provide FI users with such freedom to navigate through the system (Liu and Reed, 1995). This may explain why FI users preferred using the index tool. On the other hand, FD users rely on guidance from the system in order to find a suitable route for a task. More specifically, they may prefer to be guided in a structured manner so that they will be able to find a route for a task. The fact that FD users prefer to be guided by the system may help explain why they did not prefer using the index tool.

The remaining two features, namely 'I was confused which options I wanted, because it provided too many choices' (Q13) and 'This tutorial can be used sufficiently well without any instructions' (Q19), were previously found to be relevant features for determining users' cognitive style. However, users with different cognitive styles responded differently to these features. With regards to Q13, the tree in Figure 5.8 showed that users who responded to this feature were mainly FD users. Many of the FD users either agreed or strongly agreed with Q13. This shows that FD users were confused with the options provided to them because there were too many to choose from. This may have to do with the fact that FD users rely on the system to provide them with guidance in order to successfully complete some task (Wang, Hawk, and Tenopir, 2000). By relying on the system for some guidance, FD users expect to be shown and given the options that they need to complete the task. In the event that too many options or all of the options are presented to them, they will find it difficult to choose the one(s) that can help them with their task. This may therefore explain why FD users were confused when many options were provided to them by the tutorial.

With regards to Q19, a few Intermediate users were found to agree with this particular feature while a notable number of FD users were found to disagree with this feature. The results regarding Q19 suggest that users with different cognitive styles showed different attitudes to use of instructions. On the one hand, Intermediate users had little difficulties in using the tutorial without instructions. On the other hand, FD users had some trouble using the tutorial without instructions. The reason for this difference in preferences may have to do with the amount of guidance required by individuals. Intermediate users have the ability to combine the characteristics of both FI and FD cognitive styles. In this case, however, Intermediate users may have exhibited more of the characteristics from FI users. This is because users who follow the FI cognitive style are more likely to work on their own and actively find their own way around the system, thus avoiding the need to rely on guidance from the system (i.e., tutorial) to find what they are looking for. This shows that Intermediate users would have used the system without the help of any guidance, i.e., instructions. FD users, on the other hand, usually rely on external guidance to complete a task or locate a particular item in the system. In this case, external guidance

is guidance provided by the system. The fact that FD users prefer to be guided by the system may help explain why they found it difficult to use the system without instructions.

5.5 Conclusions

In this chapter, the mixed-type approach, and the corresponding mixed-type combinations, along with the UP1 and UP2 datasets were used to identify how the number and nature of classifiers influenced the number of features selected and the accuracy levels of the features. The mixed-type approach revealed some very interesting results.

With respect to the number of features selected, the mixed-type approach showed that 2-classifier combinations selected many relevant features and 3-classifier and 4-classifier combinations selected few relevant features. This therefore shows that combining few classifiers results in large feature subsets being selected but combining many classifiers results in small feature subsets. The reason for this difference in number of features was attributed to the strategy used to combine the classifiers. Furthermore, the results from mixed-type showed that the number of features identified was also influenced by the use of certain classifiers. In detail, BN family classifiers influenced the number of relevant features selected from the UP1 dataset while NN family classifiers influenced the number of relevant features selected from UP2 dataset. With respect to accuracy levels of selected features, the results from both UP1 and UP2 showed that combinations comprising three classifiers selected features which led to the highest classification accuracies in comparison to the other classifier combinations. The accuracies of 3-classifier combinations and the accuracies of the other combinations, however, were found to be influenced by the BN family classifiers (UP1) and NN family classifiers (UP2). Such findings suggest that the nature of classifiers belonging to the BN and NN family led to greater differences in feature selection than the nature of other classifier families used. Similarities between these two classifier families were used to explain the reason why they caused differences in number of features selected and the accuracy levels generated.

Subsequently, this chapter examined relationships between the number of features selected by mixed-type combinations and the accuracy levels that they generated. This was done for both UP1 and UP2 datasets. The results from UP1 showed there to be a positive relationship between number of features and accuracy levels of features. On the other hand, the results of UP2 showed there to be a negative relationship between these two issues. The reason for such different relationships lied within the classifiers that were used in the mixed-type combinations. Combinations with BN family classifiers were found to justify the positive relationships in UP1 while combinations with NN family classifiers were found to justify the negative relationships in UP2. Finally, the chapter examined the mixed-type combinations which formed decision trees with the highest accuracy levels. The decision trees with the highest accuracy levels from both UP1 and UP2 datasets were examined to reveal a small number of features that best described the target variables of each of the datasets.

The results from the mixed-type approach have showed that the number and nature of classifiers used influence feature selection results. The results from the same-type approach, which were presented in the previous chapter, also showed that number and nature of classifiers have considerable influences on feature selection. The mixed-type approach and same-type approaches were used to investigate the effects of number and nature of classifiers, but they are two different approaches. The two approaches differ in the type of classifiers that they combine. As such, the results from these two approaches will show the influences of number and nature of classifiers slightly differently. In order to better understand the influences of number and nature of classifiers as found by both approaches, the next chapter will synthesise the results from the same-type and mixed-type approaches.

Chapter 6 – The Influences of Classifiers: Number vs. Nature

6.1 Introduction

The purpose of this thesis was to investigate how the number and nature of classifiers influence feature selection. The investigation was carried out using two different approaches: same-type approach and mixed-type approach. These two approaches produce several interesting results, which are presented in the previous two chapters. In order to provide a deep understanding of classifiers, this chapter synthesises the results found from the previous two chapters.

More specifically, the results from these two approaches will be synthesised in three different parts. The first part, presented in Section 6.2, will focus on the number of classifiers and the results obtained from both approaches regarding this issue, which will provide answers to the first research question of the thesis (*RQ1: The influences of the number of classifiers on feature selection*). Section 6.3 presents the second part which will focus on the results regarding nature of classifiers, and will help provide answers to the second research question of the thesis (*RQ2: The influences of the nature of classifiers on feature selection*). Based on the findings from these two parts, we also identify which of the two issues (i.e., number of classifiers or nature of classifiers) has a greater effect on feature selection. This will provide answers to the third research question of the thesis (*RQ3: Whether number of classifiers or nature of classifiers has a greater influence on feature selection*). The third part of the chapter, presented in Section 6.4, will compare the decision trees formed by the same-type and mixed-type approaches for both UP1 and UP2 datasets. The comparison will consider the features used to build the decision trees from each approach and help identify a small number of features that best describe the target variable of each dataset. Finally, Section 6.5 uses the findings from all parts to propose some suggestions. The suggestions will act as a kind of reference to identify the suitability of different numbers of classifiers and classifiers with a particular nature for feature selection and decision tree construction.

6.2 Number of Classifiers

This section uses the results from the same-type and mixed-type approaches to provide a holistic view of the influences of number of classifiers on feature selection results. Firstly, we present the influences of number of classifiers on number of relevant features selected. Subsequently, we present the effects of number of classifiers on accuracy levels of selected features. Finally, we provide answers to the first research question (RQ1) based on the results from both same-type and mixed-type approaches.

6.2.1 Influences on Number of Features Selected

The combinations from the same-type and mixed-type approaches comprised of different numbers of classifiers. These combinations selected different numbers of relevant features from the UP1 and UP2 datasets. A brief summary of the results from combinations used in both approaches is shown in Table 6.1. The table shows the mean number of features selected by all same-type and mixed-type combinations for UP1 and UP2 datasets. In addition, the total mean number of features selected by these two types of combinations for the datasets is presented. The number of combinations that select number of features above and below total mean accuracy is also provided.

	Same-type Combinations						**Mixed-type Combinations**					
	UP1			***UP2***			***UP1***			***UP2***		
	Mean	*Above Total Mean*	*Below Total Mean*	*Mean*	*Above Total Mean*	*Below Total Mean*	*Mean*	*Above Total Mean*	*Below Total Mean*	*Mean*	*Above Total Mean*	*Below Total Mean*
2-classifier	17.36	10	8	6.28	10	8	17.11	29	7	6.53	20	16
3-classifier	15.31	6	6	5.67	6	6	16.42	28	26	5.94	24	30
4-classifier	11.63	-	3	4.67	1	2	11.33	3	34	4.30	7	20
Total Mean	15.05			5.91			16.33			5.76		

Table 6.1. Number of Features Selected by Same-type and Mixed-type Combinations

After a closer look at Table 6.1, a common finding was noted among the number of features selected by the combinations from the two datasets. It was found that combinations with few classifiers generally selected a high number of relevant features from UP1 and UP2 datasets while combinations with many classifiers generally selected a low number of relevant features. In fact, it was found that 2-classifier combinations selected more relevant features whereas 3-classifier and 4-classifier combinations selected fewer relevant features. In addition, it was found that more 2-classifier combinations generated number of features higher than the total mean number of features

for each UP1 and UP2 dataset. On the other hand, there were more 3-classifier and 4-classifier combinations which generated number of features lower than the total mean number of features for each dataset. These findings show that 2-classifier combinations selected a high number of features and 3-classifier and 4-classifier combinations selected a low number of features, irrespective of the nature of classifiers used in the combinations.

Interestingly, we found an additional finding that concerns the actual number of features selected by the same-type and mixed-type approaches. When comparing the mean number of relevant features selected by same-type and mixed-type combinations for both datasets (see Table 6.1), we generally found that the mean numbers of features selected by the same-type combinations are rather similar to the mean numbers of features selected by the mixed-type combinations. This result was apparent in both datasets. In addition, the results from Table 6.1 show that the total mean number of features selected by same-type and mixed-type combinations for both datasets are also very similar. The results suggest that the nature of classifiers used to form these two types of combinations had little (if any) influence on the overall number of relevant features selected. In other words, combining classifiers of the same nature and combining classifiers of a different nature made very small difference to the number of features selected from the datasets. These aforementioned results suggest that the number of classifiers used may have more of an influence on number of features selected than the nature of the classifiers used.

6.2.2 Influences on Accuracy Levels of Features

The number of classifiers used in the same-type and mixed-type combinations were also found to influence the classification accuracies generated. Table 6.2 presents a summary of the mean classification accuracies of all combinations belonging to the same-type and mixed-type approaches for UP1 and UP2. In addition, the table shows the number of classifiers combinations that generated accuracies higher and lower than the total mean accuracy level for each of the datasets.

	Same-type Combinations						Mixed-type Combinations					
	UP1			*UP2*			*UP1*			*UP2*		
	Mean	*Above Total Mean*	*Below Total Mean*	*Mean*	*Above Total Mean*	*Below Total Mean*	*Mean*	*Above Total Mean*	*Below Total Mean*	*Mean*	*Above Total Mean*	*Below Total Mean*
2-classifier	81.30	5	13	94.69	11	7	82.01	11	25	94.27	17	19
3-classifier	84.93	9	3	95.38	10	2	83.75	34	20	95.37	42	12
4-classifier	84.16	2	1	94.87	2	1	82.50	13	14	94.81	12	15
Total Mean	83			94.95			82.88			94.90		

Table 6.2. Classification Accuracies Generated by Same-type and Mixed-type Combinations

As shown in Table 6.2, 3-classifier combinations selected features which generated higher mean accuracy levels than those of 2-classifier and 4-classifier combinations. This was found across both approaches and in both datasets. In addition, the table shows that nearly all of the 3-classifier same-type and mixed-type combinations generated accuracies above the total mean accuracy of each dataset. However, in general, many of the 2-classifier and 4-classifier combinations generated accuracies below the total mean accuracy of each dataset. These results, along with the detailed results presented in Chapters 4 and 5, suggest that combinations comprising of three classifiers are more likely to identify the most accurate subsets of features in relation to the target variable, irrespective of the nature of classifiers used.

The reason for this may have to do with the number of features selected and their relevance. As previously found, the number of classifiers influenced the number of features selected. However, the features selected by the different numbers of classifiers may not necessarily be relevant to the target variable. On the one hand, 2-classifier combinations were found to select the highest number of features from the datasets. The fact that they selected the highest number may possibly suggest that some of the selected features are not very relevant to the target variables and may explain the low accuracy levels they generated. On the other hand, 4-classifier combinations were generally found to select the lowest number of features from the datasets. The fact that they selected lowest number may possibly suggest that such classifier combinations missed out some of the features that are highly relevant to the target variables, which may explain why they generated lower accuracies. The 3-classifier combinations were found to select number of features in between the number of features selected by 2-classifier and 4-classifier combinations. In addition, they selected accuracies higher than the accuracies of

2-classifier and 4-classifier combinations. This may suggest that combinations comprising of three classifiers are the right balance for excluding features of low relevance but including features of high relevance. This in turn may explain why 3-classifier combinations generated higher accuracies overall.

6.2.3 Differences in Accuracies Generated by Same-type and Mixed-type Combinations

The findings revealed two more interesting issues relating the classification accuracies generated by the same-type and mixed-type approaches.

1) Mean Accuracy Levels

The first issue relates to the mean accuracy levels generated by same-type combinations and mixed-type combinations for the two datasets. In terms of UP1, Table 6.2 showed that in general same-type combinations generated higher mean accuracies than mixed-type combinations. In terms of UP2, it was also found that same-type combinations generated higher mean accuracies than mixed-type combinations although the difference between the two was marginal. In order to better understand this issue, we carry out a deep analysis of the accuracies generated using the two datasets. The deep analysis will consider the accuracy levels and frequencies of accuracy levels generated (i.e., number of times each accuracy level was generated) by same-type and mixed-type combinations for each of the UP1 and UP2 datasets. The results from the deep analysis concerning UP1 and UP2 can be found in Table 6.3 and Table 6.4, respectively.

Level of Accuracy (%)	**Frequency of Accuracies Selected by Combinations (UP1)**	
	Same-Type Combinations	*Mixed-Type Combinations*
90.83	-	1
89.17	-	1
88.33	1	5
87.50	-	2
86.67	3	10
85.83	4	9
85	3	11
84.17	4	8
83.33	1	11
82.50	6	17
81.67	3	10
80.83	2	7
80	2	7

79.17	-	7
78.33	2	4
76.67	-	1
75	1	1
75.83	-	2
72.50	1	1

Table 6.3. Frequency of Accuracies Generated by Same-type and Mixed-type Combinations for UP1

Level of Accuracy (%)	**Frequency of Accuracies Selected by Combinations (UP2)**	
	Same-Type Combinations	*Mixed-Type Combinations*
96.92	2	9
95.38	21	67
93.85	9	34
92.31	1	7

Table 6.4. Frequency of Accuracies Generated by Same-type and Mixed-type Combinations for UP2

The results from Table 6.3 show that the same-type combinations generated far fewer kinds of accuracy levels than mixed-type combinations for UP1, as indicated by the fact that the former generated six types of accuracy levels less than the latter. The fact that same-type combinations produced fewer types of accuracy levels may suggest that same-type combinations produced less diverse accuracy levels than mixed-type combinations in the UP1 dataset. Interestingly, the results from Table 6.4 are different to that of Table 6.3. More specifically, results in Table 6.4 do not show clear differences in the accuracies generated using UP2 because the same-type and mixed-type combinations generated exactly the same accuracy levels. Therefore the results from UP1 show large differences between accuracies generated by same-type and mixed-type combinations but results from UP2 show very small differences between accuracies of these two types of combinations. A possible reason for this may lie within the number of features present in these two datasets. On the one hand, the UP1 dataset includes a large number of features, 90 to be precise. The fact that there are many features in this dataset means that it is more likely for the different types of combinations to select very different relevant features which may subsequently lead to very different levels of accuracies. On the other hand, the UP2 dataset includes a small number of features, i.e., 20 features. The fact that few features exist in this dataset implies that it may be more likely that the combinations will select similar features from the dataset which may lead to similar levels of accuracy.

Further analysis of the accuracies generated by same-type and mixed-type combinations was also carried out. The analysis only dealt with the accuracies from the UP1 dataset (shown in Table 6.3) because it was the only dataset to show major differences among accuracies generated by both types of combinations. The analysis revealed further differences among the accuracies of same-type and mixed-type combinations. These differences related to the range of accuracies generated by the combinations, which is the difference between the highest accuracy and the lowest accuracy for the 2-classifier, 3-classifier, and 4-classifier combinations. The range values for these combinations are presented in Table 6.5. In general, the range values presented in this table show that the difference between the highest and lowest accuracies of same-type combinations is smaller than that of mixed-type combinations. The smaller differences in accuracies once again suggest that there was less diversity in the accuracies generated by the same-type in comparison to mixed-type combinations.

Combinations	**Range of Accuracies (%)**	
	Same-Type Combinations	*Mixed-Type Combinations*
2-Classifier	13.33	12.50
3-Classifier	6.66	18.33
4-Classifier	2.50	10.83

Table 6.5. Range of Accuracies for Same-type Combinations and Mixed-type Combinations in UP1

A possible reason for these differences in accuracies may lie within the nature of the classifiers considered by same-type combinations. As previously mentioned, same-type combinations utilise classifiers from the same family. Therefore, the biases of these classifiers will be very similar or even identical. In other words, same-type combinations take into account the bias associated with only one type of classifier when selecting features. In this way, classifiers with similar biases may in turn select very similar features. If they select similar features then it is highly likely that the features will generate similar levels of accuracies. In contrast, mixed-type combinations make use of classifiers from different families, each of which has a very different bias. This means that the selected features will be different depending on the classifiers (and their biases) used in combinations, subsequently leading to a larger difference in the accuracies generated. In summary, same-type combinations generate similar levels of accuracies whereas mixed-type combinations generate diverse levels of accuracies. As a result of

this difference, these two types of combinations will generate different mean accuracy levels. The mean accuracy level is determined by calculating the mean of the individual accuracy levels generated by the same-type and mixed-type combinations. Because of the way in which the mean accuracy is calculated, considering similar individual accuracy levels (i.e., same-type) will result in a higher mean accuracy than if diverse individual accuracy levels (i.e., mixed-type) were considered. This may explain why same-type combinations were generally found to produce higher mean accuracy levels than mixed-type combinations.

2) Individual Accuracy Levels

The second issue identified relates to the individual accuracy levels produced by same-type and mixed-type combinations. To have a deep understanding of this issue, we consult Table 6.3 and Table 6.4 which were previously presented in this section. A closer look at these two tables reveals some interesting findings.

With regards to Table 6.3, which represents results from UP1, accuracies generated by same-type combinations seem to be evenly spread. On the other hand, accuracies generated by mixed-type combinations are somewhat unevenly spread. It was found that the majority of accuracies generated by mixed-type combinations were located towards the top half of the table as opposed to the bottom half. A deep analysis of accuracies in the top and bottom halves of Table 6.3 was conducted. The averages of accuracies in the top half and bottom half for both same-type and mixed-type combinations were computed. In terms of same-type combinations, the average of the accuracies in the top half was 85.25% and the average of the accuracies in the bottom half was 80.31%. In terms of mixed-type combinations, the top half average was 86.98% and the bottom half average was 80.86%. These findings show that the average top half accuracy of mixed-type combinations was higher than the average top half accuracy of same-type combinations. It also shows a higher proportion of accuracies generated by mixed-type combinations to be in the top half of the table. These findings therefore suggest that mixed-type combinations were able to generate slightly higher individual accuracies than those of same-type combinations.

With regards to Table 6.4, which represents results from UP2, accuracies generated by same-type and mixed-type combinations are identical. Considering the number of times each accuracy level was generated by these two types of combinations also shows similarities. These similarities are clearly shown in Table 6.6. This table is an extension of Table 6.4 in that it shows the percentage of combinations which generated each type of accuracy level. The fact Table 6.6 shows similarities in the percentage of accuracy levels generated suggests that same-type and mixed-type combinations were not really different in the context of UP2 dataset. The reason why these two combinations were not so different may have to do with the UP2 dataset itself. As previously mentioned, the UP2 dataset contains a significantly lower number of features compared to UP1. As such, classifier combinations used with this dataset may be more likely to select similar features and thus produce similar accuracy levels.

Level of Accuracy (%)	**Percentage of Accuracies Selected by Combinations (UP2)**	
	Same-Type Combinations	*Mixed-Type Combinations*
96.92	6%	8%
95.38	64%	57%
93.85	27%	29%
92.31	3%	6%

Table 6.6. Percentage of Accuracies Generated by Same-type and Mixed-type Combinations for UP2

Although results regarding UP2 dataset do not show clear findings, the results from UP1 dataset are quite clear. The results regarding accuracies of UP1 suggested that mixed-type combinations were able to generate slightly higher individual accuracies than those of same-type combinations. The reason for these results may be attributed to the nature of classifiers used in the combinations. On the one hand, same-type combinations combine classifiers from a single family. This means that the classifiers will be of the same nature and have the same or very similar biases. However, this is only one type of bias. Considering only one type of bias when selecting features may affect the accuracy of the features because the features are selected from the perspective of only one type of classifier family. On the other hand, mixed-type combinations make use of multiple classifiers from very different families. This means that the classifiers will be of different nature and subsequently have different types of biases. In other words, diverse types of

biases are considered when selecting relevant features. Considering several different types of biases in this manner can help lower the impact that they may have on the selection of features. This is because the different classifiers will mutually agree on the relevant features as a bid to overcome their individual biases. Reducing the impact of classifier biases on the feature selection process may therefore aid in the identification of feature subsets that contain highly relevant features with regards to the target variable. Such highly relevant features can subsequently lead to higher accuracy levels, which may therefore explain why mixed-type combinations were able to generate higher accuracy levels than same-type combinations using the UP1 dataset.

The two aforementioned findings showed that same-type combinations were able to generate higher mean accuracy levels than mixed-type combinations, but mixed-type combinations were able to generate higher individual accuracy levels. Such findings show that using different classifier arrangements (i.e., same-type and mixed-type) can lead to different feature selection results. In fact, the findings suggest that these classifier arrangements are capable of generating different levels of classification accuracy. Thus, there is a need to be aware of these different capabilities when performing feature selection so as to help them in uncovering the feature subset with the highest accuracy level.

6.2.4 The Role of Number of Classifiers in Feature Selection (RQ1)

The previous sections presented the findings from both same-type and mixed-type combinations regarding the influences of number of classifiers on number of features selected and the accuracy levels generated. These findings can help provide answers to the first research question (RQ1) of this thesis. Figure 6.1 summarises the key answers for RQ1. The figure presents the answers regarding influences of number of classifiers on number of features selected on the left side. The answers regarding influences of number of classifiers on accuracy levels of features are shown on right hand side of the figure. These answers can significantly improve our understanding of the role of number of classifiers in feature selection tasks.

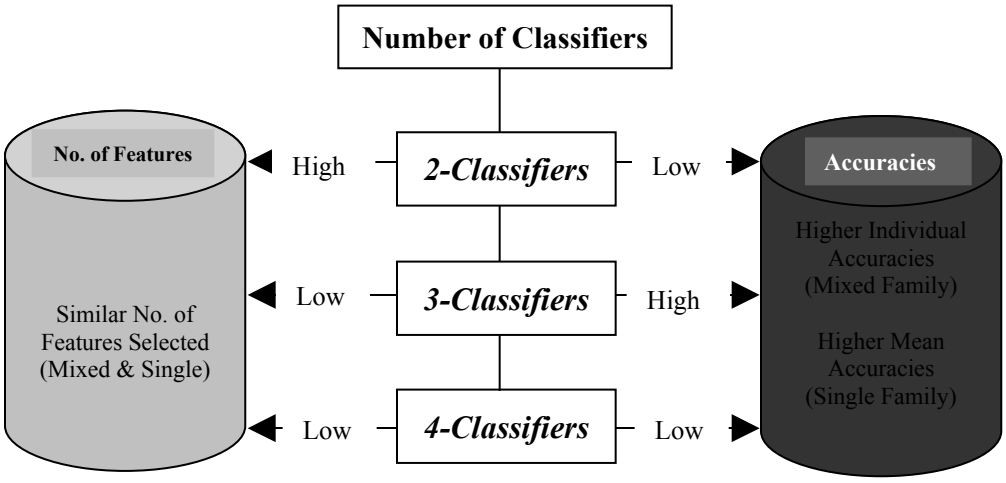

Figure 6.1. Summary of Answers to RQ1

6.3 Nature of Classifiers

This section uses the results from the same-type and mixed-type approaches to identify the overall influences of nature of classifiers on feature selection results. Initially, we present the influences of nature of classifiers on number of relevant features selected. The influences of nature of classifiers on accuracy levels of selected features are also presented. The section concludes by providing answers to the second research question (RQ2) of the thesis based on the results from both same-type and mixed-type approaches.

6.3.1 Influences on Number of Features Selected

The same-type and mixed-type approaches combined classifiers of different nature. The fact that these two approaches used classifiers of different nature helped us determine the influences of nature of classifiers on feature selection results.

In terms of same-type approach, it was found that combinations with DT family classifiers (i.e., C4.5, CART and CN2) selected a higher number of relevant features than combinations with the other classifier families. This was found in both UP1 and UP2 datasets. In terms of the mixed-type approach, it was found that combinations which included BN family classifiers (i.e., BNC, NB and AODE) influenced number of features

selected from UP1 dataset and combinations that included NN family classifiers (i.e., NNC, KNN and K*) influenced number of features selected from UP2 dataset. With regards to BN family classifiers, combinations with BNC were found to select low number of features whereas combinations with NB or AODE classifiers selected high number of features. With regards to NN family classifiers, combinations with NNC or K* classifiers selected low number of relevant features whereas combinations with KNN selected high number of features.

These results show that different families of classifiers influenced number of features selected. On the one hand, the findings suggest that DT family classifiers may only affect number of features when they are combined together. However, the influences of DT family classifiers are less apparent when combined with classifiers from other different families. Interestingly, BN family and NN family classifiers were found to cause differences in number of features selected when used and combined with classifiers from other families.

6.3.2 Influences on Accuracy Levels of Features

An examination of the accuracies generated by same-type and mixed-type combinations revealed interesting issues. With regards to the same-type combinations, we found that combinations with DT family classifiers generated higher accuracies than combinations with the other classifier families. This finding was observed in both UP1 and UP2 datasets. With regards to the mixed-type combinations, we found that combinations with BN family classifiers influenced the accuracy levels of features selected from UP1 dataset and combinations with NN family classifiers influenced the accuracy levels of features selected from UP2 dataset. With respect to former classifier family, combinations with BNC were found to generate low accuracy levels whereas combinations with NB or AODE were found to generate high accuracy levels. With respect to the latter classifier family, combinations with NNC or K* classifiers generated high levels of accuracy whereas combinations with KNN generated low levels of accuracy.

These findings suggest that DT classifiers have a greater effect on the accuracy levels when used together. On the other hand, the nature of BN and NN family classifiers cause differences in accuracy levels when used in conjunction with other classifiers from different families. In other words, the nature of BN and NN classifier families seems to be enhanced in the presence of other classifiers that have different nature. Such findings tie in with those obtained from the previous section. The previous section found that DT family classifiers influenced the number of features selected by same-type combinations whereas BN and NN family classifiers influenced the number of features selected by mixed-type combinations. The findings of the previous section and the findings from the current section collectively suggest that the nature of these three classifier families is stronger in different contexts. On the one hand, the nature of DT classifier is stronger in same-type combinations but weaker in mixed-type combinations. On the other hand, the nature of BN and NN classifiers is stronger in mixed-type combinations but weaker in same-type combinations. The reasons why the classifiers families are stronger in different contexts are explained in detail.

1) DT Family Classifiers Strong in Same-type Approach

As previously stated, DT classifiers possess two advantages over the other classifier families used. The first advantage is that DT classifiers do not require the presence of prior details regarding the features in the dataset to make decision as to which feature is more relevant than others (like BN family classifiers do), nor do they rely on a small number of features to determine set of relevant features; neighbours in the case of NN classifiers and support vectors in the case of SVM classifiers. The fact that DT classifiers do not possess such characteristics, which may reduce the number of features suggests that they may be able to select a higher number of relevant features. The second advantage relates to the fact that DT classifiers perform feature selection on their own. This means that they will perform feature selection on two occasions when used with WDT. The fact that DT classifiers perform feature selection on two occasions suggests that the features that are selected may be of very high relevance, which may subsequently lead to high levels of accuracy.

These advantages show that individual DT classifiers are able to select a high number of features which generate high levels of accuracy. However, the DT classifiers were not able to show such results when they were used in mixed-type combinations. There are two possible reasons that may explain this result. The first reason may lie within the assumptions made by the different classifiers that were used. On the one hand, DT classifiers make no assumptions about the data they use owing to the fact that they are non-parametric classifiers. On the other hand, classifiers belonging to other families, like BN and SVM, tend to make some assumptions about the data. The fact that classifiers make different assumptions suggests that there may be conflicts when such diverse classifiers are combined and used together. Such conflicts may mean that DT classifiers will have little if any influence on the feature selection results. However, when many DT classifiers are combined the result may be different. In this case, the DT classifiers that are combined will be similar in that they make no assumptions about the data. This means that there will be hardly any conflict between classifiers. As such, DT classifiers may influence results, which may explain why DT classifiers were found to be stronger in same-type combinations.

The second possible reason may lie within the number of trees that are built by DT classifiers. DT classifiers build trees to illustrate relationships among relevant features in dataset. Each DT classifier will build a single tree comprised of a certain number of relevant features. However, the trees built by different DT classifiers may not be the same since each DT classifier is slightly different. In this case, a single DT classifier may build a tree comprised of features very relevant to the target variable but the other DT classifier may build a tree with features that are not so relevant to target variable of dataset. By combining several trees, there is a better likelihood of identifying a tree in addition to features most relevant to the target variable. In other words, combining several DT classifiers may help produce better results than that of a single one. This may explain why the nature of DT classifiers was strengthened when several DT classifiers were used together.

Collectively, the findings from the same-type approach suggest that DT family classifiers have the ability to select large feature subsets that generate high levels of accuracies. The

ability to select such large yet precise feature subsets may appeal to experts who wish to use only one type of classifier family for their feature selection task. However, as explained previously, DT classifiers can only form such precise feature subsets when combined and used together.

2) BN and NN Family Classifiers Strong in Mixed-type Approach

The results from the mixed-type approach showed that classifiers belonging to the BN family and NN family influenced the number of feature selected and the accuracy levels generated. The fact that BN and NN family classifiers showed influences in the mixed-type approach may imply that they have a strong nature when used with different classifier families. In addition, the results may also imply that BN and NN family classifiers work better when used together. More specifically, they may be more influential in the presence of each other. This may be due to the fact that they have some similarities. As previously stated, BN and NN family classifiers have some similarities when determining the relevance of a feature. This may mean that they can influence both the number of features selected and thus the accuracies of the selected features. As such, similarities among BN and NN classifier families may enhance their strengths when they are combined together.

The similarities among the BN and NN family classifiers helped explain why they were found to influence feature selection results in mixed-type approach. However, BN classifiers caused different feature selection results for the UP1 dataset whereas NN classifiers caused different feature selection results for the UP2 dataset. As previously explained in Chapter 5, the reason for such findings may have to do with the nature of BN and NN family classifiers. With regards to BN family classifiers, there are two possible reasons why they influenced feature selection results of UP1. The first relates to the fact that BN family classifiers are more able to show relevancies in a dataset when the graphical network structures are built using a large number of features (i.e., UP1). Revealing the relevancies in the UP1 dataset may in turn make the influences of BN family classifiers more obvious. The second reason relates to the conditional independence assumption normally employed by BN classifiers. It may have been the

case that the features in the UP1 dataset consisted of features that were conditionally independent, which satisfies the assumption made by such BN classifiers. In this way, BN classifiers would most likely perform well on the dataset, which in turn can show their influences on feature selection results.

With regards to NN family classifiers, the reason they influenced results of UP2 may relate to the size of the feature space. The feature space of UP2 is rather small, i.e., UP2 contains a small number of features. When the feature space of a dataset is small, NN family classifiers only need to compute and handle the distances of few features when determining the relevant features. A small feature space may therefore enable NN family classifiers to clearly distinguish relevant features from irrelevant features. With this ability, NN family classifiers may be able to show their influences on feature selection results. This may explain why NN family classifiers caused differences in feature selection results of UP2 dataset.

In summary, one may need to be aware of the BN and NN classifier families when doing feature selection because of their strong influential nature in the presence of other classifier families.

6.3.3 The Role of Nature of Classifiers in Feature Selection (RQ2)

The results from the same-type and mixed-type combinations showed that the nature of classifiers greatly affects the number of features selected and accuracy levels of features. In fact, the strength of a classifier's nature was found to influence the feature selection results. Classifiers were found to possess either a *weak* nature or a *strong* nature, which reflects the classifier's ability to influence feature selection results. On the one hand, a classifier that has a *weak* nature is unable to influence feature selection results when used with classifiers of different nature, but able to influence results when used with classifiers that are similar to it in nature. An example of a classifier with a weak nature is the DT classifier. On the other hand, a classifier that has a *strong* nature is unable to influence feature selection when used with classifiers of a similar nature to it, but is able to influence results when used in conjunction with classifiers of a different nature.

Examples of classifiers with a strong nature include BN and NN classifiers. Figure 6.2 summarises the influences of classifiers with a weak nature (highlighted in a light shade of grey) and a strong nature (highlighted in a dark shade of grey) on feature selection results. The findings from this figure provide answers to the second research question of the thesis (RQ2), which can offer a better understanding of the role of nature of classifiers in feature selection.

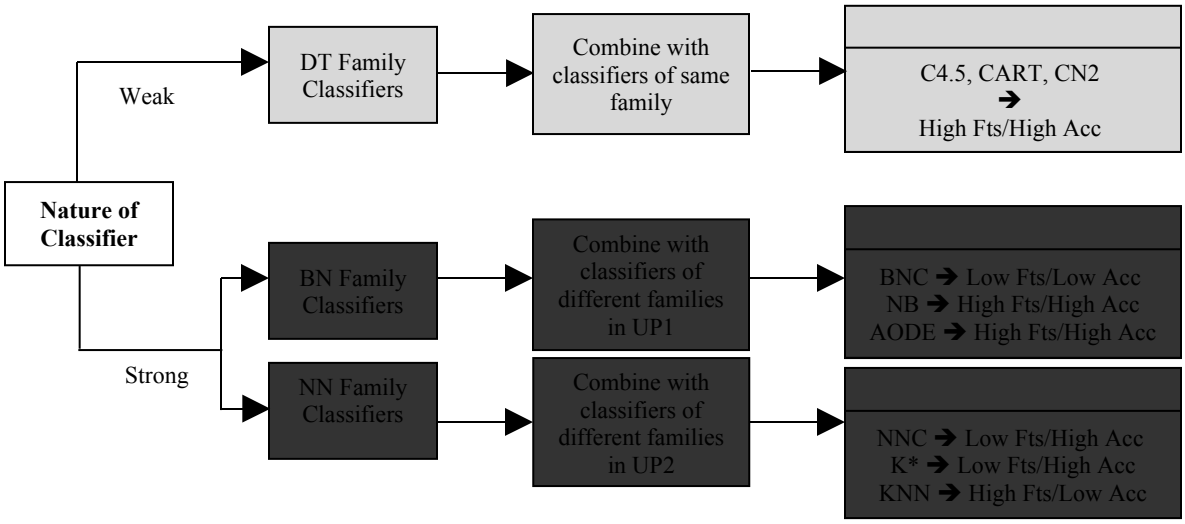

Figure 6.2. Summary of Answers to RQ2

6.3.4 Number of Classifiers vs Nature of Classifiers (RQ3)

The two abovementioned sections determined the influences of the number of classifiers and the nature of classifiers on feature selection. As a result, we were able to provide answers to the first (RQ1) and second (RQ2) research questions of the thesis. However, the third and final research question of the thesis remains unanswered. The third research question (RQ3) determines whether number of classifiers or nature of classifiers has a greater impact on feature selection. In order to provide answers to RQ3, this section compares the results obtained from number of classifiers and nature of classifiers.

In terms of number of classifiers, it was found that: 1) combinations with few classifiers consistently selected a high number of relevant features while combinations with many classifiers consistently selected low number of features and 2) combinations with three

classifiers generated higher classification accuracies than combinations with two or four classifiers. These two findings were observed among both same-type and mixed-type approaches and were found in both UP1 and UP2 datasets. In terms of nature of classifiers, the results differed according to the approach that was used. The results from same-type approach showed that combinations with DT family classifiers selected higher number of features than other families, and also generated higher accuracies than the other families. The results from the mixed-type approach, however, showed that classifiers from BN family influenced feature selection results of UP1 dataset and classifiers from NN family influenced feature selection results of UP2 dataset.

In summary, the findings from number of classifiers were observed in both approaches and across both datasets used in the thesis. On the other hand, the findings from nature of classifiers were specific to the approach used and in some cases the datasets used. This shows that the number of classifiers influenced all aspects of feature selection whereas nature of classifiers influenced certain aspects of feature selection depending on the approach and dataset. The fact that number of classifiers influenced feature selection results irrespective of the approach and dataset used may suggest that it has a greater effect than the nature of classifiers.

A possible explanation for why number of classifiers had a greater effect than nature of classifier on feature selection may be to do with the way in which WDT reduces effects of biases of individual classifiers. WDT combines multiple classifiers so as to reduce the effects of biases of individual classifiers on feature selection. However, the number of classifiers combined using WDT may have a major impact on this issue. In fact, the agreement among the number of classifiers used may influence this issue. As previously mentioned, combining few classifiers using WDT is somewhat easier to do than combining many classifiers, irrespective of the nature of classifiers used. This is because a small number of classifiers are used to agree on the relevance of a feature. The fact that a small number of classifiers are used may suggest that it is a ‘relaxed’ approach for selecting relevant features. On the other hand, combining many classifiers is often more difficult because there are more classifiers that need to agree on whether a feature is relevant or irrelevant. This may suggest that using many classifiers for selecting relevant

feature subsets is a 'strict' approach. In summary, the results seem to suggest that the number of classifiers used and the agreement among the classifiers (i.e., relaxed or strict approach) is a very important issue with regards to feature selection. The fact that the number of classifiers and the agreement among classifiers were found to be unaffected by nature of classifiers also implies that number of classifiers is more important and thus more influential than the nature of classifiers used in the WDT combinations. As such, we may need to pay more attention to the number of classifiers used when doing feature selection rather than the nature of classifiers used.

6.4 Comparison of Decision Trees

The same-type combinations and mixed-type combinations selected a set of relevant features from datasets, which were then used to build decision trees. The decision trees each had a level of classification accuracy which indicated how accurate the selected feature sets were in relation to target variable of dataset. The decision trees with highest classification accuracies were subsequently chosen and analysed. This was done for both UP1 and UP2 datasets. In this section, we compare the decision trees with the highest accuracies obtained from both same-type and mixed-type approaches for UP1 as well as UP2. By comparing the decision trees from both approaches, we will be able to identify any similarities or differences among the trees built and also identify a small number of features from the trees that are most reliable in describing the target variable of each dataset.

This section is divided into two subsections. The first subsection compares the decision trees from the UP1 dataset, whereas the second subsection compares the decision trees from the UP2 dataset.

6.4.1 Comparison of Decision Trees for UP1 Dataset

A total of two decision trees were formed for the UP1 dataset. One was formed by a same-type combination with all three decision tree classifiers (C4.5, CART, and CN2). The other tree was formed using the mixed-type approach, in which three classifiers from the BN, DT, and NN families were used. A close examination of these two decision trees

was carried out, which revealed some interesting results. The results related to the features that were used in the decision trees. The features used in the trees are summarised in Table 6.7.

	Features Used in the Decision Trees											
	Q14	***Q15***	***Q21***	***Q23***	***Q29***	***Q30***	***Q31***	***Q32***	***Q33***	***Q42***	***Q48***	***Q56***
Same-type Combination	√	√	√		√	√	√	√			√	
Mixed-type Combination	√	√		√			√		√	√	√	√

Table 6.7. Features Used in Decision Trees for UP1

The first interesting result concerns Q31. This feature appeared as the root node (i.e., the feature at the first level) of both trees. This showed that Q31 was regarded as the most relevant and most important feature by the decision trees with regards to the target variable (i.e., user's level of computer experience). This suggests that both approaches were able to identify Q31 as the most relevant feature for differentiating the preferences of users with low and high levels of computer experience.

The second interesting finding relates to a small number of features also presented in Table 6.7. In addition to Q31, other features were found to help differentiate the preferences of users with low and high levels of computer experience. In terms of the decision tree constructed using the same-type approach, the small number of features included Q31, Q14, and Q48. In terms of the decision tree constructed using the mixed-type approach, the features included Q31, Q14, Q48, and Q56. Comparing these features selected by the different approaches shows that there are common features, namely Q31, Q14 and Q48. The fact that common features were found among the decision trees may suggest that these features are highly relevant to users' level of computer experience. Such highly relevant features can therefore be used to differentiate the preferences of users with different levels of computer experience. As such, it may be essential to consider these features when developing personalised search engines that can accommodate the needs of users with different levels of computer experience.

6.4.2 Comparison of Decision Trees for UP2 Dataset

Several decision trees were formed using the UP2 dataset. One decision tree was formed using the same-type approach and two decision trees were formed using the mixed-type approach. In terms of the former approach, the tree was constructed by a combination which included all three of the NN family classifiers. In terms of the latter approach, the two decision trees were constructed by combinations which included NN family classifiers, namely NNC and K*. Interestingly, the same-type approach and the mixed type approach formed identical decision trees. This meant that only two different decision trees were produced from both same-type and mixed-type approaches. On the surface, the two decision trees appeared to be different but a deep analysis of the trees and the features used within the trees showed that they shared some common features. In fact, the two decision trees shared two common features, namely Q9 and Q18. In terms of Q9, this particular feature was used as the root node of each decision tree that was constructed implying that it is the most relevant feature among all features in the UP2 dataset. In addition, this feature was consistently assigned the highest level of relevance by the combinations used to build the decision trees. In terms of Q18, results from both types of combinations showed that this feature was the second most relevant feature in the dataset since it had high relevance levels. This may explain why it was included in the decision trees of both approaches.

All in all, the findings from the decision trees generated using the same-type and mixed-type approaches suggest that there are four key features present in UP2. The four key features include: Q9, Q18, Q13 and Q19. This is because these four features represent a small number of features that helped differentiate the preferences of users with different cognitive styles. Such features may therefore be very useful for designing personalised Web-based learning systems that suit the requirements of users with different cognitive styles.

6.4.3 Differences Between Same-type and Mixed-type Decision Trees

The comparison of decision trees from same-type and mixed-type approaches enabled us to identify a small number of features that are highly relevant to determining the target variable values of each dataset, i.e., users' level of computer experience (UP1) and users' cognitive style (UP2). However, the number of such highly relevant features varied according to the approach that was used. In general, slightly more highly relevant features were identified from trees of the mixed-type approach in comparison to trees of the same-type approach. In the case of UP1, four features were extracted from the decision tree constructed by mixed-type approach, whereas three features were extracted from the decision tree constructed by same-type approach. In the case of UP2, we were able to extract one more highly relevant feature from the trees of the mixed-type approach in comparison to the tree of same-type approach. It suggests that using mixed-type approach helped identify more highly relevant features than using the same-type approach. In other words, mixed-type approach found features that same-type approach may have missed out.

A possible reason for this may lie within the nature of classifier combinations used in these two approaches. As aforesaid, same-type combinations combine classifiers from a single family, which means that the classifiers will be of similar nature and have similar biases. In this way, same-type combinations consider the bias of a single classifier family. Considering only one type of bias when selecting features may limit the selection of highly relevant features because features will be selected from the perspective of only one type of classifier family. On the other hand, mixed-type combinations make use of multiple classifiers from different families which will be of different nature and will have different types of biases. Considering different types of biases may help lower the impact that they may have on the selection of features. In other words, mixed-type combinations may be able to select other relevant features because the perspectives of different classifier families are considered. As such, mixed-type combinations may be more likely to find a greater number of highly relevant features than same-type combinations, which will be used to build the decision trees.

6.5 Suggestions

The previous sections showed that different numbers of classifiers and classifiers with different nature affect feature selection results. In addition, it was found that different classifiers were used to build decision trees with highest accuracies. It is, therefore, essential to propose some general suggestions for identifying the suitability of different classifiers for feature selection and decision tree construction. This section proposes such suggestions in two sections. The first section will identify the suitability of classifiers for the task of feature selection. The second section will also identify the suitability of classifiers for building decision trees of highest accuracies.

6.5.1 Feature Selection

Based on the findings obtained from the first and second parts of this chapter, we were able to establish that different numbers of classifiers resulted in different number of features and accuracy levels being generated. In addition, we found that the nature of classifiers led to different number of features selected and accuracy levels. In fact, classifiers had different strengths when combined in same family (i.e., same-type approach) and when combined with different families (i.e., mixed-type approach). A detailed examination of the strengths of classifiers in these different approaches is given over the next few pages.

1) Single Family

An examination of the nature of classifiers showed that DT family classifiers seemed to have a weak nature because they were not found to influence feature selection results when used in the presence of other classifiers. On the other hand, it was interesting to discover that the nature of DT family classifiers was strengthened when several of these classifiers were used together. The fact that these classifiers showed a stronger nature when combined together meant that they influenced feature selected results. More precisely, DT family classifier combinations selected higher number of features than other classifier families and also generated higher levels of accuracies than other families.

Interestingly, a deeper analysis of the each of the DT family classifiers used in the combinations revealed two additional issues. The first issue relates to the number of

features selected by the DT classifiers. It was found that DT combinations with CART classifier selected a higher number of relevant features than combinations without CART. This was found across both UP1 and UP2 datasets. The second issue relates to the accuracy levels generated by the DT classifiers. It was found that DT combinations with CART were able to generate slightly higher accuracy levels than combinations without CART. Once again, this was found across both datasets. The results of these two issues are shown in Table 6.8. This table shows the mean number of features selected by combinations with and without CART. In addition, it shows the mean accuracy levels generated when CART was included in and excluded from the DT combinations.

	Mean no. of features selected by combinations with CART		**Mean no. of features selected by combinations without CART**		**Mean Accuracy Level of combinations with CART**		**Mean Accuracy Level of combinations without CART**	
	UP1	***UP2***	***UP1***	***UP2***	***UP1***	***UP2***	***UP1***	***UP2***
2-classifier	21.33	9.33	20.67	6.33	83.33	95.38	81.11	94.16
3-classifier	20	8	18	6	85.83	95.38	84.17	95.38
4-classifier	13	7	13	7	85	95.38	85	95.38

Table 6.8. Results from DT Family Classifiers

These two issues may suggest that the CART classifier may have been more responsible for the fact that DT family classifiers selected higher number of features and higher accuracy levels. The reason for this may relate to the characteristics of CART and the other two DT classifiers including C4.5 and CN2. These three DT classifiers are similar in that they use the data to build hierarchical trees which include the most and least relevant features with regards to target variable and display the relationships among these features. However, CART uses a slightly different way to build trees compared to C4.5 and CN2.

The CART classifier is different to the other classifiers in that it employs 10-fold cross validation during the tree building process. As such, CART can reduce the level of error in the resulting tree and utilise more of the data (i.e., features and instances) when building the tree (Kohavi, 1995a; Kohavi and Quinlan, 2002). The fact that CART can utilise more of the data whilst building the tree suggests that it is more likely to handle a higher number of features and instances. This may subsequently lead to a higher number

of features being used in the tree, which may explain why combinations which included it were able to select a higher number of relevant features than combinations without it. The fact that 10-fold cross validation helps CART reduce the error level of the tree also suggests that the resulting trees built using this classifier may be of higher accuracy than those built by the other classifiers. This may help explain why combinations with CART were able to generate higher accuracies than combinations without this classifier.

In summary, the DT family classifiers work well when used in conjunction with each other. As a result, they are able to select a high number of features. More specifically, using the CART classifier in combinations may help select a high number of features. To obtain a higher number of features, one may also want to use CART in 2-classifier combinations since these combinations were found to select a higher number of features than combinations with three or four classifiers. Interestingly, Table 6.8 shows that 2-classifier combinations with CART are able to select higher number of features than 2-classifier combinations without CART in both UP1 and UP2 datasets. Furthermore, DT family classifiers are able to produce higher accuracies when used together. In this context, it might be a better idea to include the CART classifier when doing feature selection since it was found to generate higher accuracies. Incorporating the CART classifier into a 3-classifier combination may also increase the chances of obtaining higher accuracy levels since combinations with three classifiers were previously shown to generate highest accuracy levels. In fact, the 3-classifier combinations with CART were shown in Table 6.8 to generate higher accuracy levels than those without the CART classifier for both of the datasets used.

These suggestions regarding the DT family classifiers may prove useful when such classifiers are used together. Their union creates a strong nature which results in significant effects on feature selection outputs.

2) Mixed Family

Classifiers from the BN and NN families were shown to possess a different nature to that of the DT family classifiers. It was previously found that BN family and NN family classifiers were unable to influence feature selection results when used on their own (i.e.,

single family). This seemed to suggest that they have a weak nature when used on their own. However, their nature strengthened when such classifiers were combined with classifiers from other families (i.e., mixed family). In other words, these classifiers were able to influence the feature selection results when combined with other classifier families. It may be that the nature of BN and NN family classifiers strengthens when these classifiers are used together in the same combination. It suggests that these two types of classifiers complement one another. Since there is a very high possibility that classifiers from these two families complement each other, we suggest that they be used together for the purpose of feature selection.

The suggestions are made according to the number of features that are selected by classifiers from these two families and the accuracy levels that are generated by them. The suggestions based on these two issues are shown in Figure 6.3.

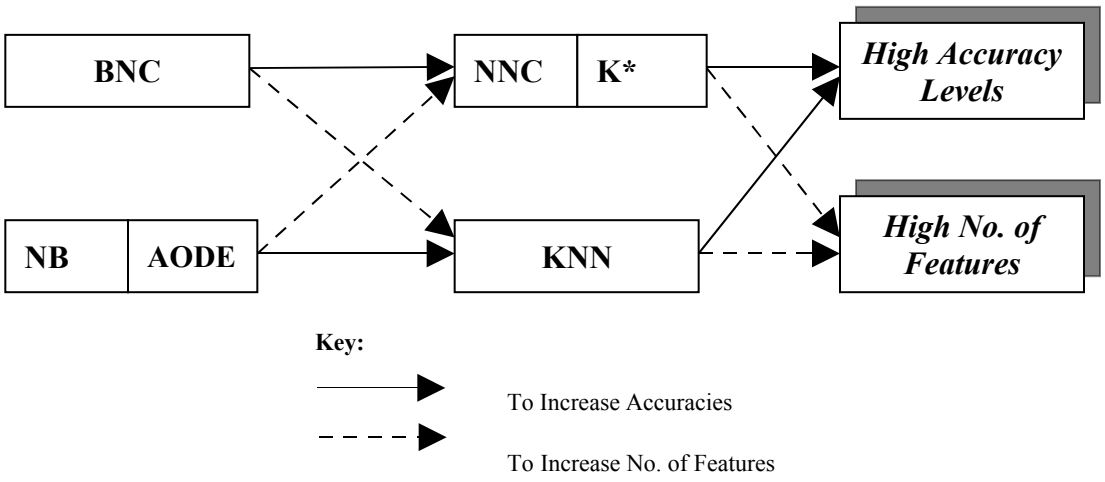

Figure 6.3. Suggestions for BN and NN Family Classifiers

With regards to number of features, classifiers from BN and NN family showed different results. In terms of the BN family, BNC led to low number of features whereas NB and AODE led to high number of features. In terms of NN family, NNC and K* led to low number of features while KNN led to high number of features. This shows that different classifiers selected different numbers of relevant features, some of which were lower or higher than others. A way of assisting the manner in which these classifiers select

relevant features may be to combine those which select low number of features with those that select high number of features. In this case, we may want to combine BNC with KNN in order to increase the number of features that are selected by BNC. In addition, we may also combine NNC or K* with either NB or AODE in order to increase the number of features selected by NNC and K*. These suggestions are shown in Figure 6.3 by dashed arrow lines. In order to further enhance the effects of these suggestions, one may also want to alter the number of BN and NN family classifiers that are used. It may be wise to combine two classifiers from these families in order to further increase the number of features selected (see Section 6.2.2).

To see whether the above suggestions work in practise, we examined their influences on the number of features selected by the combinations. The results from this examination are shown in Tables 6.9 and 6.10. The former table shows the results for increasing the number of relevant features selected by combinations with BNC. The latter table shows the results for increasing the number of features selected by combinations with NNC and combinations with K*. A close look at both of these tables reveals that the number of relevant features selected by the BNC, NNC and K* classifier combinations can in general be increased by combining them with the previously mentioned classifiers. This therefore shows that the suggestions outlined above do work in practice.

	Mean no. of features selected by BNC combinations with KNN	**Mean no. of features selected by BNC combinations without KNN**
UP1	13	12.36
UP2	10	4.83

Table 6.9. Increasing Number of Features Selected by BNC Combinations

	Mean no. of features selected by NNC combinations with NB or AODE	**Mean no. of features selected by NNC combinations without NB or AODE**	**Mean no. of features selected by K* combinations with NB or AODE**	**Mean no. of features selected by K* combinations without NB or AODE**
UP1	17	15	19	18.20
UP2	4.20	4	4	4.10

Table 6.10. Increasing Number of Features Selected by Combinations with NNC and Combinations with K*

With regards to accuracy levels, BNC generated low levels of accuracy whereas combinations with NB and AODE generated high levels of accuracy. For the NN family classifiers, NNC and K* generated high accuracy levels while KNN generated low accuracy levels. This shows that different classifiers selected different levels of accuracies. In this case, we may want to match classifiers which generated low accuracies with those classifiers that generated high accuracies. On the one hand, we may want to use BNC with either NNC or K* in order to increase the accuracy of BNC. On the other hand, we may want to combine KNN with either NB or AODE so as to increase the accuracy levels generated by KNN classifier. These suggestions are also shown in Figure 6.3 by solid arrow lines. Such suggestions may be further enhanced by taking into account the number of classifiers that are used. For the purpose of increasing accuracy levels, it may also be worthwhile combining three classifiers since such combinations were found to generate higher accuracy levels (see Section 6.2.3). For example, one may want to use KNN, NB, and one other classifier from a different family. This may help improve the chances of obtaining higher accuracy levels. In order to identify the effects of such suggestions on increasing accuracy levels, we present Table 6.11. In this table, we identify whether the inclusion of NNC or K* classifiers increase accuracy levels of BNC combinations and whether the inclusion of NB or AODE classifiers increase accuracy levels of KNN combinations. Analysing Table 6.11 shows that the accuracy levels of BNC combinations and KNN combinations increase with the addition of the aforementioned classifiers, thus supporting the suggestions previously mentioned.

	Mean accuracy levels of BNC combinations with NNC or K* (%)	**Mean accuracy levels of BNC combinations without NNC or K* (%)**	**Mean accuracy levels of KNN combinations with NB or AODE (%)**	**Mean accuracy levels of KNN combinations without NB or AODE (%)**
UP1	82.71	79.64	84.48	81.43
UP2	95.38	95	94.81	94.28

Table 6.11. Increasing Accuracy Levels of BNC Combinations and KNN Combinations

6.5.3 Decision Tree Construction

The previous section showed that classifiers from the DT family and NN family had different nature and led to different feature selection results. Interestingly, the findings

from the third part of this chapter showed that classifiers belonging to these families were able to select features that generated decision trees with the highest levels of accuracies for UP1 and UP2. For UP1, the decision tree with the highest accuracy was built using a combination with all three DT family classifiers. For the UP2, the decision trees with the highest accuracies were built by combinations comprised of NN family classifiers, namely NNC and K*. On the one hand, these findings suggest that DT classifiers are only able to build most accurate decision tree when used together. The fact that DT classifiers are influential when combined together ties in with the findings from Section 6.3.2. On the other hand, the findings regarding NN family classifiers imply that such classifiers, especially NNC and K*, should be incorporated into combinations so as to uncover the most relevant features in the dataset. Such findings tie in with those of the previous section, which showed that including these two classifiers resulted in high levels of accuracies. As a suggestion, it may therefore be worthwhile using DT family classifiers or classifiers from NN family to select relevant features from a dataset. This is because the features that are selected will more than likely form decision trees with highest accuracies. Subsequently, these kind of decision trees will help illustrate the most reliable and important relationships among the selected features.

All in all, the suggestions presented can be used to guide a choice of which number of classifiers and/or nature of classifiers to use when performing feature selection tasks and building decision trees. The suggestions cannot only be used to select suitable classifiers for such tasks but can also be used to identify those classifiers that are not so suitable. Such knowledge may prove vital when performing a wide variety of tasks. A summary of the key suggestions are illustrated in Figure 6.4.

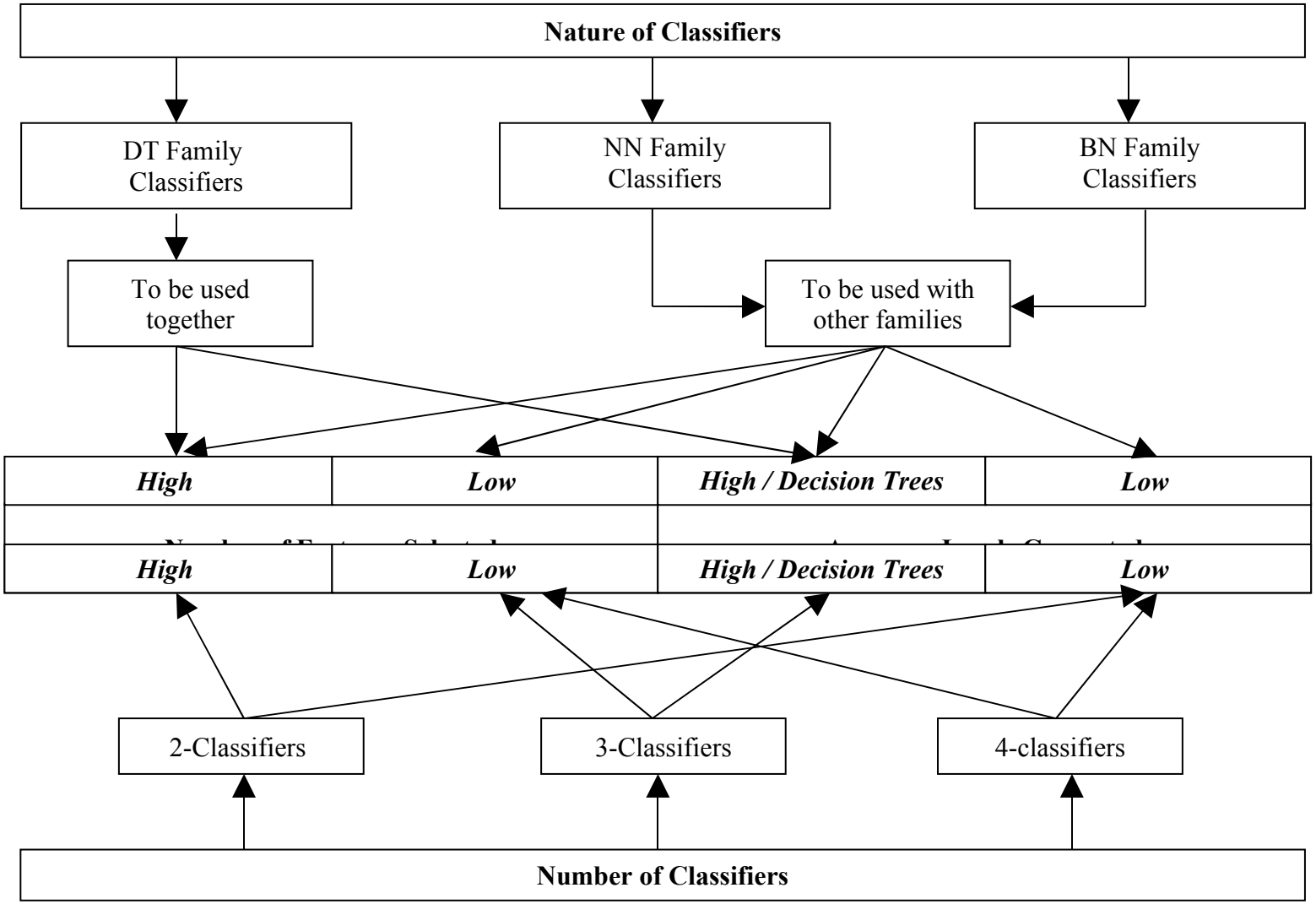

Figure 6.4. Summary of Key Suggestions

6.6 Conclusions

This chapter compared the findings from same-type and mixed-type approaches in order to determine the role of number of classifiers and nature of classifiers in feature selection. The results from the comparison helped provide answers to all three research questions of this thesis, namely RQ1, RQ2, and RQ3. In terms of RQ1, we found that combinations with few classifiers resulted in many relevant features being selected but many classifiers resulted in few features being selected regardless of the nature of classifiers used. In addition, we found that combinations with three classifiers were able to generate highest levels of accuracy regardless of what types of classifiers were used. In terms of RQ2, we found that DT family classifiers led to different feature selection results in same-type approach. However, it was found in the mixed-type approach that BN family classifiers influenced feature selection results of UP1 while NN family classifiers influenced feature selection results of UP2. According to the answers of both RQ1 and RQ2, we established that the number of classifiers had a greater effect on feature selection results than nature of classifiers, which answered RQ3. This was attributed to the fact that results regarding number of classifiers were consistent among all approaches and datasets used, whereas results regarding nature of classifiers were specific to the approach and dataset used. In other words, number of classifiers influence feature selection as a whole whereas nature of classifiers influence feature selection depending on some issues.

Subsequently, the chapter compared the decision trees generated by the same-type and mixed-type approaches. This comparison was done for both the UP1 and UP2 datasets. The results from the comparison showed similarities in the decision trees generated. The similarities helped identify a small number of highly relevant features that best describe and differentiate the different values of target variables. However, differences were also found. The differences lied within the number of highly relevant features that were identified. It was found that slightly more highly relevant features were found from mixed-type decision trees than same-type decision trees. Differences in the nature of classifiers used in the mixed-type and same-type approaches helped explain the occurrence of this finding. Finally, some general suggestions based on the comparisons in this chapter were presented. The suggestions considered the suitability of classifiers for feature selection tasks and building decision trees with the highest accuracy levels.

In summary, this chapter has provided answers to the three research questions of the thesis. The following chapter will summarise the answers to the research questions and present the key contributions of the thesis. In addition, it will describe possible limitations of the thesis and discuss directions for further work.

Chapter 7 – Conclusions

7.1 Introduction

Wrappers are probably the most popularly used feature selection approaches for reducing the dimensionality of datasets and identifying the most relevant sets of features. To identify relevant sets of features, Wrappers make use of classifiers. However, existing Wrapper approaches use a single classifier for this task. The problem with using a single classifier is that each one is different; each one will have a different nature and possess different biases. The fact that each classifier is different means that they will select different feature sets. More specifically, various classifiers may select different numbers of features which may produce different levels of accuracy. Interestingly, little is known about using different numbers of classifiers and classifiers with a different nature. On the one hand, the number of classifiers used may play a part in influencing the number of features selected or the accuracy levels generated by the selected features. On the other hand, the nature of classifiers may play a role in influencing the feature selection results.

The number of classifiers and nature of classifiers are thus two important issues that may influence feature selection results. With this in mind, this thesis investigated the role of number of classifiers and nature of classifiers in feature selection with the help of a novel data mining method called Wrapper-based Decision Trees (WDT). The WDT method is able to combine multiple classifiers for feature selection and use decision trees to visualise relationships between selected features. Therefore, the main novelty of the WDT method lies within its ability to combine multiple classifiers for feature selection. As such, the WDT method can be used with different numbers of classifiers and classifiers that have a different nature. This thesis used the WDT method along with three research questions to better understand the role of number and nature of classifiers in feature selection. The three research questions of this thesis are detailed below:

- To what extent does the number of classifiers used influence the number of features selected and the accuracy levels of the features (RQ1);

- To what extent does the nature of classifiers used influence the number of features selected and the accuracy levels of the features (RQ2); and

- Which of the two issues (i.e., number of classifiers or nature of classifiers) has a greater affect on feature selection (RQ3).

The aim of this chapter is to present the key answers to the three abovementioned research questions proposed in this thesis. The key answers to RQ1 (Section 7.2) will be presented first along with the answers to RQ2 (Section 7.3) and RQ3 (Section 7.4). The significance of the results found in this thesis is also discussed in Section 7.5. The chapter then moves on to describe the limitations of this thesis in Section 7.6. Finally, Section 7.7 discusses ideas for future work.

7.2 Number of Classifiers in Feature Selection (RQ1)

The WDT method proposed in this thesis was used with different numbers of classifiers. In fact, WDT combined two, three and four classifiers to perform feature selection. The number of classifiers that were combined was found to influence the feature selection results. There were two key findings that emerged regarding the role of number of classifiers:

1. Few classifiers (2-classifiers) selected higher number of relevant features whereas many classifiers (3-classifiers and 4-classifiers) selected lower number of relevant features.

2. Features selected by 3-classifier combinations generated higher classification accuracies than those of 2-classifier and 4-classifier combinations.

On the one hand, these findings show that different numbers of classifiers lead to different numbers of relevant features being selected. On the other hand, they show that different numbers of classifiers lead to different levels of classification accuracy. Interestingly, both findings were observed, irrespective of the nature (i.e., type) of classifiers that were used to select the relevant features. The reason why number of classifiers plays a significant role in feature selection may be attributed to the agreement among the classifiers used. The agreement among classifiers can vary because different classifiers will agree on features differently. Agreement may be high,

which may result in more features being selected, or low, which may result in fewer relevant features being selected by the classifiers. In addition, if the agreement among the classifiers leads to highly relevant features being selected, then this may lead to high levels of classification accuracy. Otherwise, agreeing on features of little or no relevance may lead to low accuracy levels. In summary, the number of classifiers and, in turn, the agreement among the classifiers can considerably affect the feature selection results generated.

7.3 Nature of Classifiers in Feature Selection (RQ2)

Classifiers with a different nature were also used with the WDT method. In fact, four families of classifier were used and these four families are Bayesian Networks, Decision Trees, Nearest Neighbour, and Support Vector Machines. The nature of classifiers belonging to three of these four families, namely Decision Tree, Bayesian Network and Nearest Neighbour families was found to influence the feature selection results. There were two main results relating to the nature of these three families:

1. Decision Tree classifiers influenced the number of features selected and accuracy levels of features when combined together.

2. Bayesian Network and Nearest Neighbour classifiers influenced the number of features selected and accuracy levels of features when combined with classifiers of different nature.

The aforementioned results suggest that the nature of classifiers belonging to these three families was stronger in different contexts. On the one hand, the nature of Decision Tree classifiers was stronger when several of these classifiers were combined and used together to do the feature selection. This strength meant that combinations with several Decision Tree classifiers influenced number of relevant features selected and accuracy levels generated. On the other hand, the nature of Bayesian Network and Nearest Neighbour classifiers was found to be stronger when such classifiers were used with classifiers from other families. Subsequently, combinations with Bayesian Network or Nearest Neighbour classifiers and classifiers from other families were shown to influence feature selection results. Collectively, these results show that classifiers of a different nature influence feature selection results in a different manner.

Such results can improve our understanding of the effects of using classifiers with different nature on the feature selection process.

7.4 The Importance of Number of Classifiers (RQ3)

The two aforementioned sections showed that the number of classifiers and nature of classifiers influence feature selection. However, the influences of these two issues were observed in different contexts. In terms of number of classifiers, the two findings mentioned in Section 7.2 were found in both of the user preference datasets used in this thesis (i.e., UP1 and UP2) and both of the classifier arrangement approaches used (i.e., same-type and mixed-type). In terms of the nature of classifiers, the two findings presented in Section 7.3 were specific to the dataset used and also the classifier approach used. For example, the first finding shown in the previous section (relating to the Decision Tree classifiers) was observed in both datasets but only in the same-type approach. These findings regarding these two issues seem to suggest that the number of classifiers influenced all aspects of feature selection whereas nature of classifiers influenced certain aspects of feature selection depending on the classifier approach and dataset employed. The fact that number of classifiers influenced feature selection results irrespective of the classifier approach and dataset used may suggest that it has a greater effect than the nature of classifiers. In other words, the number of classifiers plays a bigger role in feature selection than the nature of classifiers used. The reason for this was attributed to the level of agreement among the classifiers used for feature selection.

In summary, the findings suggest that the number of classifiers used and the agreement among the classifiers is a very important issue with regards to feature selection. In fact, the number of classifiers is more of an important issue than the nature of classifiers. With this in mind, there may be a need to pay more attention to the number of classifiers used for feature selection. In doing so, they can improve the chances of finding features most relevant to the feature selection task.

7.5 Significance of This Study

The significance of the results presented in this study lies within three different aspects including theory, methodology and application. Each of these aspects is explained in detail:

- With regards to *theory*, the study contributes to the understanding of the effects of using multiple classifiers for feature selection, in particular the number of classifiers and nature of classifiers. The results in this study showed that number and nature of classifiers are two important issues that can significantly affect the feature selection results. Suggestions based on these results were proposed to assist in the selection of suitable number and nature of classifiers for feature selection tasks.

- With regards to *methodology*, the study developed a new data mining method called WDT to help analyse the effects of the two abovementioned issues. The novelty of the WDT method lies within its ability to: 1) combine multiple classifiers in order to select relevant sets of features and 2) visualise the interactions among selected features using decision tree. Due to these two abilities, the WDT method was able to identify highly accurate sets of relevant features.

- With regards to *application*, the WDT method was applied to datasets consisting of users' preferences of search engines and Web-based learning systems. From these datasets, WDT was able to identify sets of highly relevant features. These highly relevant features helped differentiate the preferences of users regarding search engines and Web-based learning systems. Therefore, the relevant features selected in this study may prove useful for better understanding the preferences of different users when interacting with such systems.

7.6 Limitations of Thesis

As with any piece of research, there are limitations which may affect the results and conclusions obtained. A summary of the limitations of the research presented in this thesis are described in the next few pages.

- This thesis investigated the role of number and nature of classifiers in feature selection using the WDT method. However, this investigation was carried out using only two types of user preference datasets. The fact that this investigation only considered datasets from one area of research (i.e., HCI) may somewhat

limit the findings obtained and the suggestions proposed. Further works are needed to conduct experiments with a greater number of datasets so that the findings of this thesis can be strengthened and validated.

- The WDT method was able to: 1) combine several classifiers to perform feature selection and 2) visualise relationships between selected relevant features. In this thesis, a small selection of classifiers from only four types of classifier families was used to perform these two tasks. Employing a small selection of classifiers may limit the feature selection findings obtained. Several other types of classifiers are needed to expand on the findings.

- The WDT method was also able to combine different numbers of classifiers. But only a small number of classifiers were combined (i.e., two, three and four). Combining a small number of classifiers may limit the number of features selected and affect the accuracy levels generated by the features.

- In addition to the classifiers used, the WDT method made use of the forward search strategy to help identify relevant feature subsets. However, this is only one type of searching strategy. There are many different types of strategies that can be used, each of which may lead to different features being selected and subsequently affect the accuracy of the results. Experiments with other types of search strategies are needed to reinforce the results of this thesis.

Some of these limitations can be used as the starting point for future work, which are discussed in the following section.

7.7 Directions for Future Work

The research presented in this thesis has identified new directions for future research, some of which are summarised in the following pages.

- Chapter 3 proposed the WDT method which combines multiple classifiers for feature selection. However, a small number of classifiers were used with WDT in this thesis. It may be worthwhile using a greater variety of classifiers to do the feature selection. For example, Neural Networks (May et al., 2008) and

Genetic Algorithms (Sikora and Piramuthu, 2007), which are prolific in the data mining community, may be used with WDT. Employing such additional classifiers with WDT may further enhance our understanding of the role of nature of different classifiers in feature selection.

- In order to further understand the role of number of classifiers in feature selection, it may also be an idea to increase the number of classifiers that are used with WDT. In other words, use five or more classifiers with the WDT method. This may help us better understand how the relevance of features is affected as the number of classifiers increases. In addition, we may also be able to find a point at which the number of classifiers makes little difference to the relevance of the features selected.

- Interactions among features selected by WDT were visualised using decision trees. Since decision trees are only one type of classifier that builds visual models, it may be valuable to use a different classifier for this task. Bayesian Networks classifiers, for example, may be used for this task as they are capable of building visual structures that show relevant dependencies and relationships between features. By using other classifiers that build visual models, we will be able to gain a deeper look at the relationships among selected relevant features.

- As previously mentioned, this thesis investigated the role of number and nature of classifiers using two types of user preference datasets. However, these datasets were relatively small in size, i.e., less than 100 features. It is therefore necessary to use datasets that are larger in size in order to establish the validity of the results in the thesis. For example, bioinformatics datasets, which consist of thousands of features, may be suitable. The results from this process can be combined with the results presented in this thesis in order to uncover how the role of the number and nature of classifiers in feature selection differs according to the size of the datasets used.

- Employing the use of several different datasets, each of a different size, will not only help generalise the suggestions proposed in this thesis but will also help develop additional suggestions that can collectively assist experts in choosing

classifiers suitable to particular feature selection tasks. However, there is a need to conduct further empirical work to validate such suggestions.

References

Abbasi, A., Chen, H., and Salem, A., (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. *ACM Transactions on Information Systems,* 26, 3, 1-34.

Abraham, R., Simha, J.B., and Iyengar, S.S., (2007). Medical Datamining with a New Algorithm for Feature Selection and Naive Bayesian Classifier. In *10th International Conference on Information Technology, (ICIT 2007)*, Orissa, pp 44-49.

Aha, D.W. and Bankert, R.L., (1995). A comparative evaluation of sequential feature selection algorithms, In D. Fisher & J.-H. Lenz (Eds.), *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics, New York: Springer-Verlag*, pp 1-7.

Almuallim, H. and Dietterich, T.G., (1991). Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, *Anaheim*, pp 547-542.

Angiulli, F. and Folino, G., (2007). Distributed Nearest Neighbor-Based Condensation of Very Large Data Sets. *IEEE Transactions on Knowledge and Data Engineering*, 19, 12, 1593-1606.

Aula, A., Jhaveri, N., and Käki, M., (2005). Information search and re-access strategies of experienced web users In W*orld Wide Web (WWW) Conference 2005, Chiba, Japan*, pp 583-592.

Barakat, N.H. and Bradley, A.P., (2007). Rule Extraction from Support Vector Machines: A Sequential Covering Approach. *IEEE Transactions on Knowledge and Data Engineering*, 19, 6, 729-741.

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

Bell, D.A., Guan, J.W., and Bi, Y., (2005). On combining classifier mass functions for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 17, 10, 1307-1319.

Benbrahim, H. and Bensaid, A., (2000). A comparative study of pruned decision trees and fuzzy decision trees. *NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society, 2000*, pp 227-231.

Berger, H., Merkl, D., and Dittenbach, M., (2006). Exploiting partial decision trees for feature subset selection in e-mail categorization. In *Proceedings of the 2006 ACM Symposium on Applied Computing, SAC '06, Dijon, France,* pp 1105-1109.

Bhavani, S.D., Rani, T.S., and Bapi, R.S., (2008). Feature selection using correlation fractal dimension: Issues and applications in binary classification problems. *Applied Soft Computing*, 8, 1, 555-563.

Blanco, R., Inza, I., Merino, M., Quiroga, J., and Larrañaga, P., (2005). Feature selection in Bayesian classifiers for the prognosis of survival of cirrhotic patients treated with TIPS. *Journal of Biomedical Informatics*, 38, 5, 376-388.

Blum, A.L. and Langley, P., (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97, 1-2, 245-271.

Bo, Y. and Luo, Q., (2007). Personalized Web Information Recommendation Algorithm Based on Support Vector Machine. *In the 2007 International Conference on Intelligent Pervasive Computing, IPC2007, Jeju City*, *South Korea*, pp 487-490.

Bohen, S.P., et al. (2003). Variation in gene expression patterns in follicular lymphoma and the response to rituximab. In *Proceedings of the National Academy of Sciences*, 100, 4, 1926-1930.

Brand-Gruwel, S., Wopereis, I., and Vermetten, Y., (2005). Information problem solving by experts and novices: analysis of a complex cognitive skill. *Computers in Human Behavior*, 21, 487-508.

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

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J., (1984). *Classification and Regression Trees*, Wadsworth Int. Group/Probability Series, Belmont, California, USA.

Buckinx, W., Moons, E., den Poel, D.K., and Wets, G., (2004). Customer-adapted coupon targeting using feature selection. *Expert Systems with Applications*, 26, 4, 509-518.

Caruana, R. and Freitag, D., (1994). Greedy attribute selection. In *Machine Learning: Proceedings of the Eleventh International Conference*, Morgan Kaufmann.

Castañeda, J.A., Muñoz-Leiva, F., and Luque, T., (2007). Web Acceptance Model (WAM): Moderating effects of user experience. *Information & Management*, 44, 4, 384-396.

Castelo, R. and Siebes, A., (2000). Priors on network structures. Biasing the search for Bayesian networks. *International Journal of Approximate Reasoning*, 24, 1, 39-57.

Castellano, G., Fanelli, A.M., Mencar, C., Torsello, M.A., (2007). Similarity-Based Fuzzy Clustering for User Profiling. *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*, *Silicon Valley, CA*, pp. 75-78.

Chan, K.C. and Wong, A.K., (1991). A Statistical Technique for Extracting Classificatory Knowledge from Databases, *Knowledge Discovery In Databases*, Piatetsky-Shapiro, G., Frawley, W., (Eds.), AAAI/MIT Press, 107-124.

Chang, C.L., (2007). A Study of Applying Data Mining to Early Intervention for Developmentally-dalayed Children. *Expert Systems with Applications,* 33, 2, 407-412.

Chen, S.Y., (2000). *The Role of Individual Differences and Levels of Learner Control in Hypermedia Learning Environments.* PhD Thesis, University of Sheffield, UK.

Chen, S.Y., (2002). A Cognitive Model for Non-linear Learning in Hypermedia Programmes. *British Journal of Educational Technology,* 33, 4, 453-464.

Chen, S. Y., (2002). The Relationships between Individual Differences and the Quality of Learning Outcomes in Web-based Instruction. In *Proceedings of the ICEB Second International Conference on Electronic Business*, pp 345-351.

Chen, S.Y. and Macredie, R.D., (2004). Cognitive Modelling of Student Learning in Web-based Instructional Programmes. *International Journal of Human-Computer Interaction*. 17, 3, 375-402.

Chen, S.Y., Fan, J., and Macredie, R.D., (2006). Navigation in Hypermedia Learning Systems: Experts vs. Novices. *Computers in Human Behavior,* 22, 2, 251-266.

Chen, X.-W., (2003). An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters,* 24, 12, 1925-1933.

Chen, D., He, Q., and Wang, X., (2005). The Infinite Polynomial Kernel for Support Vector Machine. *Lecture Notes in Computer Science In Advanced Data Mining and Applications*, 3584, 267-275.

Chen, R.-C. and Hsieh, C.-H. (2006). Web page classification based on a support vector machine using a weighted vote schema. *Expert Systems with Applications*, 31, 2, 427-435.

Chen, Y. and Liginlal, D., (2007). Bayesian Networks for Knowledge-Based Authentication. *IEEE Transactions on Knowledge and Data Engineering*, 19, 5, 695-710.

Chien, C.F., Wang, W.C., and Cheng, J.C., (2007). Data Mining for Yield Enhancement in Semiconductor Manufacturing and An Empirical Study. *Expert Systems with Applications,* 33, 1, 192-198.

Chiu, S.-H., Chen, C.-C., and Lin, T.H., (2008). Using support vector regression to model the correlation between the clinical metastases time and gene expression profile for breast cancer. *Artificial Intelligence in Medicine*, 44, 3, 221-231.

Chrysostomou, K.A., Chen, S.Y., and Liu, X., (in press a). The Effects of Multiple Classifiers on Feature Selection. In *Proceedings of the 14th International Conference on Automation & Computing, Brunel University, UK.*

Chrysostomou, K.A., Chen, S.Y., and Liu, X., (in press b). The Influences of Number and Nature of Classifiers on Consensus Feature Selection. In *Proceedings of the 2008 International Conference on Data Mining, Las Vegas, Nevada*.

Chrysostomou, K.A., Lee, M., Chen, S.Y., and Liu, X., (2008). Wrapper Feature Selection. *Encyclopedia of Data Warehousing and Mining,* 4*,* 2103-2108.

Clark, P. and Niblett, T., (1989). The CN2 induction algorithm. *Machine Learning*, 3, 261-283.

Cleary, J.G. and Trigg, L., (1995). K*: an instance-based learner using an entropic distance measure. *In Proceedings of International Conference on Machine Learning*, *Tahoe City, CA, USA, Morgan Kaufmann*, pp 108-114.

Cornforth, D.J., Jelinek, H.F., Teich M.C. and Lowen, S.B. (2004), Wrapper subset evaluation facilitates the automated detection of diabetes from heart rate variability measures, in Mohammadian (ed.), *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA'2004),* University of Canberra, Australia, pp 446-455.

Cortes, C. and Vapnik, V., (1995). Support Vector Networks. *Machine Learning*, 20, 273-297.

Cover, T. and Hart, P., (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13, 1, 21-27.

Cristian, M.M. and Dan, B.D., (2006). From Decision Trees to Classification Rules with Data Representing User Traffic from an e-Learning Platform. In *2nd Conference on Information and Communication Technologies, 2006, ICTTA'06*, 1, pp 702-707.

Cristianini, N. and Shawe-Taylor, J., (2000). *An Introduction to Support Vector Machines*, Cambridge University Press.

Czekaj, T., Wu, W., and Walczak, B., (2008). Classification of genomic data: Some aspects of feature selection. *Talanta*, 76, 3, 564-574.

Dasarathy, B.V., (1991). *Nearest Neighbor (NN) Norms-NN Pattern Classification Techniques*. Los Alamitos, CA: IEEE Computer Society Press.

Dash, M. and Liu, H., (2003). Consistency-based search in feature selection. *Artificial Intelligence,* 151, 1-2, 155-176.

de Souza, J.T., Matwin, S. and Japkowicz, N., (2006). Parallelizing Feature Selection. *Algorithmica*, 45, 3, 433-456.

Denis, F., Gilleron, R., Letouzey, F., (2005). Learning from positive and unlabeled examples. *Theoretical Computer Science*, 348, 1, 70-83.

Džeroski, S. and Ženko, B., (2004). Is Combining Classifiers with Stacking Better than Selecting the Best One?. *Machine Learning*, 54, 3, 255-273.

Fan, W., Gordon, M.D., and Pathak, P., (2005). Effective profiling of consumer information retrieval needs: a unified framework and empirical comparison. *Decision Support Systems*, 40, 2, 213-233.

Florez-Lopez, R., (2007). Modelling of insurers' rating determinants. An application of machine learning techniques and statistical models. *European Journal of Operational Research*, 183, 3, 1488-1512.

Ford, N. and Miller, D., (1996). Gender differences in Internet perception and use. *Aslib Proceedings,* 48, 183-192.

Ford, N. and Chen, S.Y., (2000). Individual Differences, Hypermedia Navigation and Learning: An Empirical Study. *Journal of Educational Multimedia and Hypermedia*, 9, 4, 281-312.

Ford, N., Miller, D., and Moss, N., (2005). Web search strategies and human individual differences: Cognitive and demographic factors, internet attitudes and approaches. *Journal of the American Society for Information Science and Technology,* 56, 741–756.

Freund, Y., (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, 121, 2, 256-285.

Frias-Martinez, E., (2007). *User Modelling for Digital Libraries: A Data Mining Approach*, PhD Thesis, Brunel University, UK.

Frias-Martinez, E., Chen, S. Y., and Liu, X. (2006). Survey of Data Mining Approaches to User Modeling for Adaptive Hypermedia. *IEEE Transactions on Systems, Man, and Cybernetics: Part C,* 36, 6, 734-749.

Frias-Martinez, E., Chen, S. Y. Macredie, R, and Liu, X. (2007). The Role of Human Factors in Stereotyping Behavior and Perception of Digital Library Users: A Robust Clustering Approach. *User Modelling and User Adapted Interaction,* 17, 3, 305-337.

Friedman, N., Geiger, D., and Goldszmidt, M., (1997). Bayesian network classifiers. *Machine Learning*, 29, 131-163.

Gammerman, A.J., (1997). *Machine Learning: Progress and Prospects*. Royal Holloway, University of London, Egham, Surrey, ISBN 0 900145 93 5.

Gammerman, A.J., (1998). Learning by Support Vector Machine, Ridge Regression and Transduction. In *NTTS98: International Conference on New Techniques and Technologies for Statistics*, *Sorrento, Italy,* pp 175-181.

Ghosh, A.K., (2006). On optimum choice of k in nearest neighbor classification. *Computational Statistics & Data Analysis,* 50, 3113–3123.

Gibbons, J.M., Cox, G.M., Wood, A.T.A., Craigon, J., Ramsden, S.J., Tarsitano, D., and Crout, N.M.J., (2008). Applying Bayesian Model Averaging to mechanistic models: An example and comparison of methods. *Environmental Modelling & Software*, 23, 8, 973-985.

Grossman, D. and Domingos, P., (2004). Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood. In *Proceedings of the 21st International Conference on Machine Learning, Banff, Alberta, Canada*, 69, pp 361-368.

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

Hall, M.A. and Smith, L.A., (1997). Feature Subset Selection: A Correlation Based Filter Approach. In *Proceedings of the 4th International Conference on Neural Information Processing and Intelligent Information Systems, Berlin: Springer*, pp 855-858.

Hall, M.A., (2000). Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, *Morgan Kaufmann, San Francisco, CA*, pp 359-366.

Han, J. and Kamber, M., (2006). *Data Mining: Concepts and Techniques,* Second Edition, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers.

Hand, D.J., Mannila, H., and Smyth, P., (2001). *Principles of data mining*, MIT Press.

Hanczar, B., Courtine, M., Benis, A., Hennegar, C., Clément, K., and Zucker, J., (2003). Improving classification of microarray data using prototype-based feature selection. *SIGKDD Explorer Newsletter,* 5, 2, 23-30.

Hashemi, S. (2005). Linear-time wrappers to identify atypical points: two subset generation methods. *IEEE Transactions on Knowledge and Data Engineering*, 17, 9, 1289-1297.

Heckerman, D., Geiger, D., and Chickering, D.M., (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20, 3, 197-243.

Hu, H.-L. and Chen, Y.-L., (2008). Mining typical patterns from databases. *Information Sciences*, 178, 19, 3683-3696.

Huan, L. and Lei, Y., (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17, 4, 491-502.

Huang, J., Cai, Y., and Xu, X., (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information, *Pattern Recognition Letters*, 28, 1825–1844.

Huang, C.J., Liu, M.-C., Chu, S.-S., and Cheng, C.-L., (2007). An intelligent learning diagnosis system for Web-based thematic learning platform. *Computers & Education,* 48, 4, 658-679.

Huang, C.-J., Yang, D.-X., and Chuang, Y.-T., (2008). Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications*, 34, 4, 2870-2878.

Huang, D. and Chow, T., (2007). Effective Gene Selection Method With Small Sample Sets Using Gradient-Based and Point Injection Techniques. *IEEE/ACM Transactions on Computer Biology and Bioinformatics,* 4, 3, 467-475.

Inza, I., Larrañaga, P., Blanco, R., and Cerrolaza, A.J., (2004). Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence in Medicine*, 31, 2, 91-103.

Jian, C., Jian, Y., and Jin, H., (2005). Automatic content-based recommendation in e-commerce, *In Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, pp 748-753.

Jiang, L., Wang, D., Cai, Z., and Yan, X., (2007). Survey of Improving Naive Bayes for Classification, *Lecture Notes in Computer Science In Advanced Data Mining and Applications*, 4632, 134-145.

John, G.H., Kohavi, R., and Pfleger, K., (1994). Irrelevant features and the subset selection problem. *In Proceedings of the Eleventh International Conference on Machine learning*, pp 121-129, New Brunswick, NJ, Morgan Kaufmann.

Kapetanios, G., Labhard, V., and Price, S., (2008). Forecasting Using Bayesian and Information-Theoretic Model Averaging: An Application to U.K. Inflation. *Journal of Business & Economic Statistics*, 26, 1, 33-41.

Kira, K. and Rendell, L., (1992). A practical approach to feature selection. In: *Proceedings of Ninth International Conference on Machine Learning, Aberdeen, Scotland*, pp 249-256.

Kirkos, E., Spathis, C., and Manolopoulos, Y., (2007). Data Mining Techniques for the Detection of Fraudulent Financial Statements. *Expert Systems with Applications,* 32, 4, 995-1003.

Kittler, J., (1978). Feature set search algorithms. In C. H. Chen, editor, *Pattern Recognition and Signal Processing*. Sijhoff an Noordhoff, The Netherlands, 41-60.

Kittler, J., (1998). Combining classifiers: A theoretical framework. *Pattern Analysis and Applications,* 1, 18-27.

Kittler, J., Hatef, M., Duin, R.P.W., and Matas, J., (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 226-239.

Kohavi, R., (1995a). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of 14th International Joint Conference on Artifical Intelligence* (*IJCAI'95), Montréal, Canada*, pp 1137-1143.

Kohavi, R., (1995b). The power of decision tables. In *Proceedings of the European Conference on Machine Learning (ECML'95)*, *Lecture Notes in Artifical Intelligence*, 914, pp 174-189.

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

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

Kohavi, R. and Quinlan, J.R., (2002). Decision-tree discovery. In Will Klosgen and Jan M. Zytkow, editors*, Handbook of Data Mining and Knowledge Discovery*, chapter 16.1.3, pp 267-276. Oxford University Press.

Koller, D. and Sahami, M., (1996). Toward optimal feature selection. In *Proceedings of International Conference Machine Learning, Bari, Italy*, pp 284-292.

Kritikou, Y., Demestichas, P., Adamopoulou, E., Demestichas, K., Theologou, M., and Paradia, M., (2008). User Profile Modeling in the context of web-based learning management systems. *Journal of Network and Computer Applications*, 31, 4, 603-627.

Kuri-Morales A. and Rodríguez-Erazo, F., (in press). A search space reduction methodology for data mining in large databases. *Engineering Applications of Artificial Intelligence*. Available online at: http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V2M-4SY5WWP-1&_user=10&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=c1f474d3b0a69d6f069bb7ca37c433a1.

Langley, P. and Sage, S., (1994). Induction of selective Bayesian classifiers. In *Proceedings of the 10th Annual Conference on Uncertainty in AI*, *Seattle, Washington, USA,* pp 399-406.

Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J.A., Armañanzas, R., Santafé, G., Pérez, A., and Robles, V., (2006). Machine Learning in Bioinformatics. *Briefings in Bioinformatics*, 7, 1, 86-112.

Larose, D.T., (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley & Sons Inc, Hoboken, New Jersey.

Lazander, A.W., Biemans, H.J., and Wopereis, I.G., (2000). Differences between novice and experienced users in searching information on the World Wide Web. *Journal of the American Society for Information Science*, 51, 76–581.

Lee, C. and Lee, G.G., (2006). Information gain and divergence-based feature selection for machine learning-based text categorization. *Information Processing & Management*, 42, 1, 155-165.

Li, S. and Shue, L. (2004). Data mining to aid policy making in air pollution management. *Expert Systems and Applications*, 27, 3, 331-340.

Li, X., Rao, S., Wang, Y., and Gong, B., (2004). Gene mining: a novel and powerful ensemble decision approach to hunting for disease genes using microarray expression profiling. *Nucleic Acids Research*, 32, 9, 2685-2694.

Li, Y. and Guo, L., (2008). TCM-KNN scheme for network anomaly detection using feature-based optimizations. In *Proceedings of the 2008 ACM Symposium on Applied Computing SAC '08, Fortaleza, Ceara, Brazil*, pp 2103-2109.

Li, Y.F., Xie, M., and Goh, T.N., (in press). A study of mutual information based feature selection for case based reasoning in software cost estimation. *Expert Systems with Applications*. Available online at: http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V03-4T2DKTJ-1&_user=10&_rdoc=1&_fmt=&_orig=search&_sort=d&view=c&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=365e1c8ea3602618a21c71704e932f24

Liao, S.S., Wang, H.Q., Li, Q.D., and Liu, W.Y., (2006). A Functional-Dependencies-Based Bayesian Networks Learning Method and Its Application in a Mobile Commerce System. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, 36, 3, 660-671.

Liaw, S.-S. and Huang, H.-M., (2006). Information retrieval from the World Wide Web: a user-focused approach based on individual experience with search engines. *Computers in Human Behavior*, 22, 3, 501-517.

Ling, C.X. and Zhang, H., (2002). The Representational Power of Discrete Bayesian Networks. *Journal of Machine Learning Research*, 3, 709-721.

Liu, M. and Reed, W.M., (1995). The effect of hypermedia assisted instruction on second-language learning through a semantic-network-based approach. *Journal of Educational Computing Research*, 12, 2, 159-175.

Liu, X. and Kellam, P. (2003) Mining Gene Expression Data. In: *Bioinformatics: Genes, Proteins and Computers*. C Orengo, D Jones and J Thornton (eds), pp 229-244.

Liu, H. and Kešelj, V., (2007). Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests. *Data & Knowledge Engineering*, 61, 2, 304-330.

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

Liu, H., Li, J., and Wong, L., (2002). A Comparative Study on Feature Selection and Classification Methods Using Gene Expression Profiles and Proteomic Patterns. *Genome Informatics*, 13, 51-60.

Liu, H., Sun, J., Liu, L. and Zhang, H., (in press). Feature selection with dynamic mututal information. *Pattern Recognition*, Available online at: http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V14-4TW6HMB-1&_user=10&_coverDate=11%2F08%2F2008&_alid=843301104&_rdoc=173&_fmt=high&_orig=search&_cdi=5664&_st=13&_docanchor=&view=c&_ct=9180&_acct=C000050221&_version=1&_urlVersion=0&_userid=10&md5=37749d25b236c9e0ee63f3e78da80152

Lu, C.-L. and Chen, T.C., (2009). A study of applying data mining approach to the information disclosure for Taiwan's stock market investors. *Expert Systems with Applications*, 36, 2, 3536-3542.

Mak, M.-W. and Kung, S.-Y., (2008). Fusion of feature selection methods for pairwise scoring SVM. *Neurocomputing*, 71, 16-18, 3104-3113.

Marill, T. and Green, D. M., (1963). On the effectiveness of receptors in recognition systems. *IEEE Transactions on Information Theory*, 9, 11–17.

May, R.J., Maier, H.R., Dandy, G.C., and Fernando, T.M.K.G., (2008). Non-linear variable selection for artificial neural networks using partial mutual information. *Environmental Modelling & Software*, 23, 10-11, 1312-1326.

Miller, A. J., (1990). *Subset Selection in Regression*. Chapman and Hall, New York, 1990.

Miller, F. R. and Neill, J. W., (2008). General lack of fit tests based on families of groupings. *Journal of Statistical Planning and Inference*, 138, 8, 2433-2449.

Mitchell, T.M. (1997). *Machine Learning*. McGraw-Hill Higher Education.

Mitchell, T.J.F., Chen, S.Y., and Macredie, R.D. (2005). Hypermedia learning and prior knowledge: domain expertise vs. system expertise. *Journal of Computer Assisted Learning*, 21, 53–64.

Mladenic, D. and Grobelnik, M., (2003). Feature selection on hierarchy of web documents. *Decision Support Systems*, 35, 1, 45-87.

Moore, A.W. and Lee, M.S. (1994). Efficient algorithms for minimizing cross validation error. In *Proceedings of the Eleventh International Conference on Machine Learning*, *New Brunswick, NJ, USA*, pp 190-198, Morgan Kaufmann.

Ni, B. and Liu, J. (2004). A hybrid filter/wrapper gene selection method for microarray classification. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, 4, 2537-2542.

Niculescu, R.S., Mitchell, T.M., and Rao, R.B., (2006). Bayesian Network Learning with Parameter Constraints. *Journal of Machine Learning Research*, 7, 1357–1383.

Nikovski, D. and Kulev, V., (2006). Induction of compact decision trees for personalized recommendation. In *Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon, France,* pp 575-581.

Osei-Bryson, K.-M., (2008). Post-pruning in regression tree induction: An integrated approach. *Expert Systems with Applications*, 34, 2, 1481-1490.

Pazzani, M. and Billsus, D. (1997). Learning and Revising User Profiles: The identification of interesting web sites. *Machine Learning,* 27, 313-331.

Pearl, J., (1998). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Perner, P. and Apte, C., (2004). Empirical evaluation of feature subset selection based on a real-world data set. *Engineering Applications of Artificial Intelligence*, 17, 3, 285-288.

Piramuthu, S., (2004). Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*, 156, 2, 483-494.

Piramuthu, S. and Sikora, R.T., (2009). Iterative feature construction for improving inductive learning algorithms. *Expert Systems with Applications*, 36, 2, 3401-3406.

Pirooznia, M., Yang, J., Yang, M. Q., and Deng, Y., (2008). A comparative study of different machine learning methods on microarray gene expression data, *BMC Genomics*, 9, 1, 1-13.

Polat, K. and Güneş, S. (2006). The effect to diagnostic accuracy of decision tree classifier of fuzzy and *k*-NN based weighted pre-processing methods to diagnosis of erythemato-squamous diseases. *Digital Signal Processing*, 16, 6, 922-930.

Polat, K. and Güneş, S., (2009). A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems. *Expert Systems with Applications*, 36, 2, 1587-1592.

Prabowo, R. and Thelwall, M., (2006). A comparison of feature selection methods for an evolving RSS feed corpus. *Information Processing & Management*, 42, 6, 1491-1512.

Puig, D. and Garcia, M.A., (2006). Automatic texture feature selection for image pixel classification. *Pattern Recognition*, 39, 11, 1996-2009.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

Ritthoff, O., Klinkenberg, R., Fischer, S., and Mierswa, I., (2002). A Hybrid Approach to Feature Selection and Generation Using an Evolutionary Algorithm. In *Proceedings of the UKCI-02*, Birmingham, UK, pp. 147-154.

Roy, M., and Chi, M.T.C., (2003). Gender differences in patterns of searching the web. *Journal of Educational Computing Research*, 29, 3, 335–348.

Ruan, X., Li, Y., Li, J., Gong, D., and Wang, J., (2006). Tumor-specific gene expression patterns with gene expression profiles. *Science in China Series C: Life Sciences*, 49, 3, 293-304.

Ruiz, R., Riquelme, J. C., and Aguilar-Ruiz, J. S., (2006). Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition*, 39, 12, 2383-2392.

Sartore, L., Papanikolaou, G.E., Biancari, F., and Mazzoleni, F., (2008). Prognostic factors of cutaneous melanoma in relation to metastasis at the sentinel lymph node: A case-controlled study. *International Journal of Surgery*, 6, 3, 205-209.

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

Saunders, C., Gammerman, A.J., Brown, H., and Donald, G., (2000). Application of Support Vector Machines to Fault Diagnosis and Automated Repair, In: *Proceedings of the Eleventh International Workshop on the Principles of Design (DX'00), Morelia, Mexico*, pgs 5.

Sboner, A., Eccher, C., Blanzieri, E., Bauer, P., Cristofolini, M., Zumiani, G., and Forti, S., (2003). A multiple classifier system for early melanoma diagnosis. *Artificial Intelligence in Medicine*, 27, 1, 29-44.

Schapire, R.E., (1990). The strength of weak learnability. *Machine Learning*, 5, 2 197–227.

Selen, Y., Larsson, E.G., Stoica, P., and Sandgren, N., (2004). A model averaging approach for equalizing sparse communication channels. In *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, 1, pp 677-681.

Sikora, R. and Piramuthu, S., (2007). Framework for efficient feature selection in genetic algorithm based data mining. *European Journal of Operational Research*, 80, 2, 723-737.

Sima, C. and Dougherty, E.R., (2008). The peaking phenomenon in the presence of feature-selection. *Pattern Recognition Letters*, 29, 11, 1667-1674.

Somol, P., Baesens, B., Pudil, P., and Vanthienen, J., (2005). Filter- versus wrapper-based feature selection for credit scoring. *International Journal of Intelligent Systems*, 20, 10, 985-999.

Stein, G., Chen, B., Wu, A. S., and Hua, K. A. 2005. Decision tree classifier for network intrusion detection with GA-based feature selection. In *Proceedings of the 43rd Annual Southeast Regional Conference, Kennesaw, Georgia*, 2, pp 136-141.

Stitson, M.O., Weston, J., Gammerman, A.J., Vovk, V., and Vapnik, V., (1996). Theory of Support Vector Machines. *Technical Report CSD-TR-96-17, Department of Computer Science, Royal Holloway, University of London*.

Su, J. and Zhang, H., (2006). Full Bayesian Network Classifiers. In *Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA*, 148, pp 897-904.

Sugumaran, V., Muralidharan, V., and Ramachandran, K.I., (2007). Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing. *Mechanical Systems and Signal Processing*, 21, 2, 930-942.

Szpunar-Huk, E., (2006). Classifier Building by Reduction of an Ensemble of Decision Trees to a Set of Rules. *International Conference on Computational Intelligence for Modelling Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, *Sydney*, pp 144-148.

Tai, W.-S. and Chen, C.-T., (2006). A Web User Preference Perception System Based on Fuzzy Data Mining Method. *Lecture Notes in Computer Science* In *Information Retrieval Technology*, 4182, pp 615-624.

Talavera, L., (2005). An evaluation of filter and wrapper methods for feature selection in categorical clustering. *In Sixth International Symposium on Intelligent Data Analysis, IDA05*, *in Lecture Notes in Computer Science,* 3646, pp 440-451.

Takahashi, H., Murase, Y., Kobayashi, T., and Honda, H., (2007). New cancer diagnosis modeling using boosting and projective adaptive resonance theory with improved reliable index. *Biochemical Engineering Journal*, 33, 2, 100-109.

Thawornwong, S. and Enke, D., (2004). The adaptive selection of financial and economic variables for use with artificial neural networks. *Neurocomputing*, 56, 205-232.

Torres, J., Saad, A., and Moore, E., (2007). Application of a GA/Bayesian Filter-Wrapper Feature Selection Method to Classification of Clinical Depression from Speech Data. *Soft Computing in Industrial Applications*, 39, 115-121.

Treu, S., (1994). *User interface design*, Plenum Press, New York.

Tso, S. K. and Gu, X. P., (2004). Feature selection by separability assessment of input spaces for transient stability classification based on neural networks. *International Journal of Electrical Power & Energy Systems,* 26, 3, 153-162.

Ture, M., Tokatli, F., and Kurt, I., (2009). Using Kaplan–Meier analysis together with decision tree methods (C&RT, CHAID, QUEST, C4.5 and ID3) in determining recurrence-free survival of breast cancer patients. *Expert Systems with Applications,* 36, 2, 2017-2026.

Vapnik, V., (1998). *Statistical Learning Theory*. New York, NY: John-Wiley.

Vapnik, V., (2000). *The Nature of Statistical Learning Theory*, 2nd Edition. New York, NY: Springer-Verlag.

Vinciotti, V., Tucker, A., Kellam, P., and Liu, X., (2006). The Robust Selection of Predictive Genes via a Simple Classifier. *Applied Bioinformatics*, 5, 1-12.

Wang, P., Hawk, W.B., and Tenopir, C. (2000). User's interaction with world wide web resources: an exploratory study using a holistic approach. *Information Processing and Management,* 36, 229-251.

Wang, Y., Tetko, I.V., Hall, M.A., Frank, E., Facius, A., Mayer, K.F.X., and Mewes, H.W., (2005). Gene selection from microarray data for cancer classification--a machine learning approach. *Computational Biology and Chemistry*, 29, 1, 37-46.

Webb, G.I., Boughton, J.R., and Wang, Z., (2005). Not So Naive Bayes: Aggregating One-Dependence Estimators. *Machine Learning, 58*, 5-24.

White, K.J. and Sutcliffe, R.F.E., (2006). Applying incremental tree induction to retrieval from manuals and medical texts. *Journal of the American Society for Information Science and Technology*, 57, 5, 588-600.

Williams, N., Zander, S., and Armitage, G., (2006). A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification, *ACM SIGCOMM Computer Communication Review*, 36, 5, 7-15.

Wittmann, T. and Ruhland, J., (1999). Target Group Selection in Retail Banking through Neuro-Fuzzy Data Mining and Extensive Pre- and Postprocessing. *Lecture Notes in Computer Science*, *Data Warehousing and Knowledge Discovery,* 359-368.

Witkin, H.A., Moore, C.A., Goodenough, D.R., and Cox, P., (1977). Field-dependent and field independent cognitive styles and their educational implications. *Review of Educational Research*, 47, 1-64.

Wong, M.L., Lam, W., and Leung, K.S., (1999). Using evolutionary programming and minimum description length principle for data mining of Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, 2, 174–178.

Wu, T.-K., Huang, S.-C., and Meng, Y.-R., (2008). Evaluation of ANN and SVM classifiers as predictors to the diagnosis of students with learning disabilities. *Expert Systems with Applications*, 34, 3, 1846-1856.

Yang, J. and Olafsson, S., (2006). Optimization-based feature selection with adaptive instance sampling. *Computers & Operations Research*, 33, 11, 3088-3106.

Yeung, K.Y., Bumgarner, R.E., and Raftery A.E., (2005). Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data. *Bioinformatics*, 21, 10, 2394-2402.

Yu, E. and Cho, S., (2006). Constructing response model using ensemble based on feature subset selection. *Expert Systems with Applications*, 30, 352–360.

Yu, L. and Liu, H., (2004). Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research*, 5, 1205-1224.

Zhang, D., Chen, S., Zhou, Z.H., (2008). Constraint Score: A new filter method for feature selection with pairwise constraints. *Pattern Recognition*, 41, 5, 1440-1451.

Zheng, H. and Zhang, Y., (2008). Feature selection for high-dimensional data in astronomy. *Advances in Space Research*, 41, 12, 1960-1964.

Zhu, Z., Ong, Y.-S., and Dash, M., (2007). Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework. *IEEE Transactions Systems Man and Cybernetics: B Cybernetics*, 37, 1, 70-76.