An Empirical Study of Wrappers for Feature Subset Selection based on a Parallel Genetic Algorithm: The Multi-Wrapper Model

Thesis by

Othman Mohammad Soufan

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

King Abdullah University of Science and Technology

Thuwal, Kingdom of Saudi Arabia

September 17, 2012
EXAMINATION COMMITTEE APPROVALS FORM

The thesis of Othman M. Soufan is approved by the examination committee.

Committee Chairperson Dr. Vladimir Bajic

Committee Member Dr. Panos Kalnis

Committee Member Dr. Xin Gao
ABSTRACT

An Empirical Study of Wrappers for Feature Subset Selection based on a Parallel Genetic Algorithm: The Multi-Wrapper Model

Othman Mohammad Soufan

Feature selection is the first task of any learning approach that is applied in major fields of biomedical, bioinformatics, robotics, natural language processing and social networking. In feature subset selection problem, a search methodology with a proper criterion seeks to find the best subset of features describing data (relevance) and achieving better performance (optimality). Wrapper approaches are feature selection methods which are wrapped around a classification algorithm and use a performance measure to select the best subset of features. We analyze the proper design of the objective function for the wrapper approach and highlight an objective based on several classification algorithms. We compare the wrapper approaches to different feature selection methods based on distance and information based criteria. Significant improvement in performance, computational time, and selection of minimally sized feature subsets is achieved by combining different objectives for the wrapper model. In addition, considering various classification methods in the feature selection process could lead to a global solution of desirable characteristics.
ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Prof. Vladimir Bajic for the continuous support of my master’s thesis work, for his patience, motivation, enthusiasm, and immense knowledge. I do not see him as a great researcher but also as a leader inspiring people through gentle prestigious personality.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Panos Kalnis and Prof. Xin Gao for their guidance and evaluation of the work.

My appreciation also goes to my friends and colleagues and the department faculty and staff for making my time at King Abdullah University of Science and Technology a great experience. I also want to extend my gratitude to all the King Abdullah University of Science and Technology Professors and students who were willing to participate in the study.

Finally, my heartfelt gratitude is extended to my parents for their continuous supplications and encouragement, to my wife for her patience and support and to our kid for his wonderful lovely smile.
# TABLE OF CONTENTS

EXAMINATION COMMITTEE APPROVALS FORM.......................................................... 2

ABSTRACT ................................................................................................................. 4

LIST OF ABBREVIATIONS ......................................................................................... 9

LIST OF SYMBOLS .................................................................................................. 10

LIST OF ILLUSTRATIONS ......................................................................................... 11

LIST OF TABLES ....................................................................................................... 14

Chapter 1 .................................................................................................................. 15
Introduction ............................................................................................................... 15
1.1 Contribution ....................................................................................................... 16
1.2 Structure of the Thesis ....................................................................................... 16

Chapter 2 .................................................................................................................. 18
Related Work ............................................................................................................ 18

Chapter 3 .................................................................................................................. 24
Background ............................................................................................................... 24
3.1 Problem Definition ............................................................................................ 24
  3.1.1 Relevance and Optimality ........................................................................... 25
  3.1.2 Redundancy ................................................................................................ 27
3.2 Revealing Examples .......................................................................................... 27
  3.2.1 Redundant Variables and Separability ....................................................... 28
  3.2.2 Useless Variables and Complementarity ................................................... 29
  3.2.3 Relevance and Optimality ........................................................................ 31
Chapter 4 ............................................................................................................................................... 32
Feature Selection Model .................................................................................................................. 32
  4.1 Filter Methods ....................................................................................................................... 34
  4.2 Wrapper Methods .................................................................................................................. 35
  4.3 Embedded Methods ............................................................................................................... 36

Chapter 5 ............................................................................................................................................... 37
Supervised Machine Learning: Concepts and Definitions .......................................................... 37
  5.1 The Classification Task ......................................................................................................... 37
  5.2 Learning Algorithms ............................................................................................................ 37
    5.2.1 Naive Bayes Classifier ................................................................................................. 38
    5.2.2 C5.0 Decision Tree Generator .................................................................................... 39
    5.2.3 Multi-Support Vector Machine .................................................................................. 41
  5.3 Performance Evaluation ...................................................................................................... 43

Chapter 6 ............................................................................................................................................... 45
Genetic Algorithm ............................................................................................................................ 45
  6.1 Parallel Genetic Algorithm ................................................................................................. 46
  6.2 PGAPack Parallel Model ..................................................................................................... 47
    6.2.1 Initialization .................................................................................................................. 49
    6.2.2 Population Replacement .............................................................................................. 49
    6.2.3 Selection ........................................................................................................................ 50
    6.2.4 Crossover ....................................................................................................................... 50
    6.2.5 Mutation ........................................................................................................................ 51
    6.2.6 Stopping Criteria ......................................................................................................... 51

Chapter 7 ............................................................................................................................................... 52
Proposed Method ............................................................................................................................ 52
  7.1 Fitness Function ..................................................................................................................... 53
Chapter 8................................................................................................................. 57

Experimental Datasets .......................................................................................... 57

8.1 CorrAL .............................................................................................................. 59
8.2 m-of-n-3-7-10 ............................................................................................... 59
8.3 Monk1, Monk2 and Monk3 ............................................................................... 59

Chapter 9.................................................................................................................. 61

Experiments and Results ....................................................................................... 61

9.1 Design of the Objective Function for Wrappers for Feature Subset Selection .................................................................................................................. 61

9.1.1 How the function values in the generations of the PGA change given different values of $\alpha$? ....................................................................................... 61
9.1.2 Does $\alpha$ hold an effect on the CPU execution time? .............................. 67
9.1.3 Is there a significant difference in the size of the selected feature subsets given different wrapper models? ............................................................... 69

9.2 Comparison of Different Feature Selection Methods .................................... 70

9.2.1 Feature Selection Methods .......................................................................... 71
9.2.2 Classification Algorithms ........................................................................... 73
9.2.3 Performance Validation .............................................................................. 73
9.2.4 Relevance Validation .................................................................................. 81

Chapter 10............................................................................................................... 82

Conclusion ............................................................................................................. 86

REFERENCES ......................................................................................................... 88
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>JMI</td>
<td>Joint Mutual Information</td>
</tr>
<tr>
<td>MRMR</td>
<td>Maximum-Relevance Minimum-Redundancy</td>
</tr>
<tr>
<td>MSVM</td>
<td>Multi-class Support Vector Machine</td>
</tr>
<tr>
<td>NBC</td>
<td>Naïve Bayes Classifier</td>
</tr>
<tr>
<td>PGA</td>
<td>Parallel Genetic Algorithm</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>$D$</td>
<td>Dataset</td>
</tr>
<tr>
<td>$S$</td>
<td>Sample of data</td>
</tr>
<tr>
<td>$A$</td>
<td>Subset of features</td>
</tr>
<tr>
<td>$X_i$</td>
<td>The $i$th feature or variable in a dataset</td>
</tr>
<tr>
<td>$Y$</td>
<td>The target classification variable or label</td>
</tr>
<tr>
<td>$L$</td>
<td>Learning algorithm</td>
</tr>
<tr>
<td>$x^{(i)}$</td>
<td>The $i$th sample of a dataset</td>
</tr>
<tr>
<td>$\omega_i$</td>
<td>The $i$th discrete class label</td>
</tr>
<tr>
<td>$C$</td>
<td>The total number of classes</td>
</tr>
<tr>
<td>$M$</td>
<td>The total number of features</td>
</tr>
<tr>
<td>$N$</td>
<td>The total number of samples</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Weight parameter of minimally sized feature subset objective</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS

Figure 1: XY-Projection for Two IID Datasets (a) before rotation and (b) after 45 degree rotation (b); Figures from (Guyon & Elisseff, 2003) ................................................................. 29

Figure 2: (a) XY-Projection for Two Class Conditional Distributions with Identical Covariance Matrices (b) XY-Projection for XOR-like problem; Figures from (Guyon & Elisseff, 2003) ........................................................................................................ 29

Figure 3: Feature Selection Model .............................................................................................................. 32

Figure 4: PGAPack Master-Slave Model (Levine, 1996) ........................................................................ 47

Figure 5: Proposed System Model .............................................................................................................. 52

Figure 6: Comparison of the Best Error Rates Given the Generations of the PGA ............... 63

Figure 7: Comparison of the Size of the Selected Feature Subsets given the Generations of the PGA ........................................................................................................................................ 64

Figure 8: Effect of Alpha on Performance given Datasets on which an Improvement or Equal Progress was Realized ........................................................................................................ 65

Figure 9: Effect of Alpha on Performance given Datasets on which an Improvement or Equal Progress was Realized ........................................................................................................ 66

Figure 10: Effect of Alpha on Performance given Datasets on which Worse Performance was Realized ........................................................................................................ 66

Figure 11: Effect of Different Alpha Values on CPU Time ........................................................................ 68

Figure 12: Effect of Different Alpha Values on CPU Time Given Different Classification Methods ........................................................................................................ 69
Figure 13: Boxplot for the Sizes of the Selected Features Given Different Wrapper Models ................................................................................................................................................................. 70

Figure 14: Comparison of Different Feature Selection Methods based on KNN-3 Classifier .......................................................................................................................................................................................................... 76

Figure 15: Comparison of Different Feature Selection Methods based on KNN-5 Classifier .......................................................................................................................................................................................................... 76

Figure 16: Comparison of Different Feature Selection Methods based on SVM-Linear Classifier ........................................................................................................................................................................................................ 77

Figure 17: Comparison of Different Feature Selection Methods based on SVM-Poly of Degree 3 Classifier .................................................................................................................................................................................................. 77

Figure 18: Comparison of Different Feature Selection Methods based on AdaBoost Classifier .................................................................................................................................................................................................. 78

Figure 19: Comparison of Different Feature Selection Methods based on Bagging Classifier .................................................................................................................................................................................................. 78

Figure 20: Comparison of Different Feature Selection Methods based on Mean Error Rate of C5.0, MSVM and NBC ................................................................................................................................................................................................... 79

Figure 21: Comparison of Different Feature Selection Methods based on Error Rate of MSVM .................................................................................................................................................................................................. 79

Figure 22: Comparison of Different Feature Selection Methods based on Error Rate of C5.0 ................................................................................................................................................................................................... 80

Figure 23: Comparison of Different Feature Selection Methods based on Error Rate of NBC .................................................................................................................................................................................................. 80
Figure 24: Relevant Feature Selection Comparison of Different Feature Selection Methods based on Artificial Datasets .................................................................................................................. 82

Figure 25: Comparison of Misclassification Error Rate between C5.0 and C5.0 based on Wrapper Feature Selection .............................................................................................................. 84

Figure 26: Comparison of Generated Tree Size between C5.0 and C5.0 based on Wrapper Model for Feature Selection .............................................................................................................. 84

Figure 27: Comparison of Relevance Score using Benchmark Artificial Datasets between C5.0 and C5.0 based on Wrapper Feature Selection ........................................................................ 85

Figure 28: Percentage of Progress in CPU Time for C5.0 with Wrapper Feature Selection over C5.0 .............................................................................................................................. 85
LIST OF TABLES

Table 1: Parameters Setting for the GA .............................................................. 51

Table 2: List of Objective Functions for Wrapper-Based Feature Selection Methods ..... 54

Table 3: Summary of Experimental Datasets. Datasets 11-16 are artificial benchmark datasets .................................................................................................................. 58

Table 4: Target Concepts and Relevant Features for the Monk Datasets ................. 60
CHAPTER 1

INTRODUCTION

Data sources and generation techniques have been spreading into various disciplinary and reaching different segments of end users. This results in massive amounts of data that require advanced analysis tools to produce useful information. Machine learning is a field of study and a market trend dedicated to designing and building computational learning models for extracting information from data that would result in effective decision making. It has been applied into different core areas including medical diagnosis, robotics, bioinformatics, natural language processing and social networking.

As machine learning aims at treating larger and more complex tasks, the problem of defining the most relevant information increasingly constitutes a major phase (Blum & Langley, 1997). The first task of any learning approach or statistical pattern recognition is to solve the problem of feature selection (Somol, Novovicová, & Pudil, 2010). Feature selection attempts to select the minimally sized set of features that would maintain a good accuracy and as close as possible preserve the original class distribution, given all features (Dash & Liu, 1997).

The choice of features or variables to represent a set of patterns in a given domain affects many factors of the learning approach. The selected features can improve the prediction performance of the predictors, produce faster and more cost-effective ones
In addition, it can lead to a better description of the underlying process that generated the data and thus, a better understanding (Guyon & Elisseeff, 2003). Several feature selection methods have been introduced in the literature to produce the best feature subset (Dash & Liu, 1997; Guyon & Elisseeff, 2003; Kohavi & John, 1997; P. Somol et al., 2010).

1.1 Contribution

In this thesis, we follow an extensive empirical methodology to study the design of the objective function for wrappers for feature subset selection. In particular, we examine the effect of including selection of minimally sized feature subset as an objective on classification performance, size and CPU time. Furthermore, we propose a new formulation of the objective function based on multiple classifiers named as multi-wrapper and compare it to the individual based wrapper models. Several experiments performed on real and artificial benchmark datasets illustrates that a single feature subset minimizing more than one induction algorithm at the same time could be generated using the multi-wrapper approach. In addition, it is shown that a C5.0 decision tree generator based on PGA outperforms other wrapper and filter approaches given generalization ability, performance level and selection of relevant features.

1.2 Structure of the Thesis

The remainder of this thesis is organized as follows: In Chapter 2, we discuss previous publications related to our work. The problem of feature selection is presented with a set of examples revealing the complexity level of the problem in Chapter 3. Different
solutions to the problem are highlighted in Chapter 4. In subsequent chapters, a review for the different classification methods and parallel genetic algorithm is provided. The proposed model is, then, described as well as several fitness functions which are derived to guide feature selection for a wrapper-based approach in Chapter 7. A description of the experimental datasets is highlighted in Chapter 8. The results are discussed in Chapter 9 which is followed by a summary and conclusion in Chapter 10.
CHAPTER 2
RELATED WORK

Feature selection has been one of the major machine learning topics addressed and researched from early 1960s. Narendra and Fukunaga formulated the feature selection problem as a search method based on branch and bound techniques (Narendra & Fukunaga, 1977). Their algorithm is designed to avoid exhaustive enumeration by discarding direct evaluation of suboptimal subsets. The selected feature subset is, following their scheme, guaranteed to be global given a monotonic criterion function (i.e. a subset of attributes should be not better than any larger set containing the subset). The monotonicity condition, however, does not apply to many common evaluation functions. Foroutan and Sklansky exploited the concept of approximate monotonicity of the error rate with respect to the number of features to overcome this limitation (Foroutan & Sklansky, 1987).

Siedlecki and Sklansky proposed the use of genetic algorithms (GA) for feature selection (Siedlecki & Sklansky, 1989). The feature selection problem in their approach was formulated as a constrained optimization problem suitable for genetic search. The algorithm should look for the smallest or least costly subset of features maintaining a specified level of accuracy. The error evaluation was based on a derived error rate model which encounter for effects that occurs often in the K-NN rule. This approach
outperformed other non-exhaustive methods, in particular sequential search and branch and bound.

Relief is one of the most common techniques of feature selection that is based on weighting features in a given set (Kira & Rendell, 1992). The algorithm detects the statistically relevant features according to their relation to the target concept. Based on Euclidean distance measure, it picks Near-hit instance and Near-Miss instance for each sample in the dataset. For two given discrete feature values, Relief assigns a score of one if they match and zero otherwise. Given that Relief focuses on the target concept, redundant features may be part of the produced solution. The solution in this case is not optimal in size.

A variant of the feature selection based on genetic algorithm was described in (Punch et al., 1993). In order to search for a relative weighting of features that gives an optimal classification performance, the space of search was relaxed to allow for ranges over broader scale rather than the discrete one. The fitness function combines a misclassification measure as well as a cardinality percentage measure of the near-neighbor minority set of the K-NN classifier. The authors investigated a parallel implementation of the genetic algorithm that showed significant time improvement.

John, Kohavi and Pfleger examined the notions of relevance and irrelevance and they presented two degrees of relevance (John, Kohavi, & Pfleger, 1994). Guided by the
intuition that the selected features should consider the induction algorithm rather than just focusing on the features and the target concept, they popularized the wrapper model. In this model, the feature subset selection method is based on using cross-validation that is applicable to any induction algorithm existing. The accuracy did not improve significantly for some datasets but the induced trees were generally smaller in these cases using the wrapper model.

Classical sequential forward feature selection (SFS) and backward elimination (SBS) are sub-optimal methods that do not examine feature subsets in a diversified manner. This led to the design of floating search methods (Pudil, Novovičová, & Kittler, 1994). Given the sequential forward floating search, the proposed algorithm progresses as the basic SFS but with a conditional exclusion for the least significant features in the formed set. By means of conditional floating up and down of the included or excluded features, the method is allowed to correct wrong decisions made in previous steps towards approximating the optimal solution as much as possible. This method provides either the optimal or a close to optimal solution given less computational time than the branch and bound strategy.

Yang and Honavar applied the genetic algorithm for solving the feature subset selection problem based on a wrapper model of a constructive artificial neural network pattern classifier (Yang & Honavar, 1998). They showed that a fitness function that combined both classification’s accuracy and features subset’s cost outperformed that based on
accuracy alone concerning generalization accuracy, the number of features used, and the number of hidden neurons. The cost measure, nevertheless, is not maintained for many datasets since the field expert need to define associated cost or risk for every feature. Other studies have proposed the application of genetic algorithms to the problem of feature selection based on different frameworks (Kuncheva & Jain, 2000; Raymer, Punch, Goodman, Kuhn, & Jain, 2000).

Approaches based on optimization formulation of the feature selection problem often combines different optimization objectives into a single objective function. This is carried out through assigning specific weights to every objective, usually, based on the practitioner’s intuition or experience. This kind of strategy does not consider different possibilities of trade-offs between the different objective of the optimization function. Emmanouilidis et al. introduced a multi-objective evolutionary for feature selection to derive a set of non-dominated solutions with objective values and to explore a wide range of possible trade-offs (Emmanouilidis, Hunter, & MacIntyre, 2000; Emmanouilidis, Hunter, MacIntyre, & Cox, 1999).

Several studies have surveyed feature selection methods based on different approaches and analyses. Dash and Liu discuss different definitions for feature selection as well as their own one (Dash & Liu, 1997). Thirty two existing feature selection methods are categorized in the study based on the combinations of evaluation function and search procedure. Blum and Langley explored different definitions of relevance and they
examined characterization of different works on filter and wrapper approaches to feature selection (Blum & Langley, 1997). They highlighted, also, feature weighting methods and the problem of irrelevant examples of a dataset. Guyon and Elisseeff describe various ranking criteria applied for feature selection (Guyon & Elisseeff, 2003). They present a series of examples that illustrate the usefulness and the limitations of variable ranking techniques followed by an outline of main directions in feature selection including filter, wrapper and embedded methods.

The first work to apply a parallel genetic algorithm for feature selection, probably, appeared in (Punch et al., 1993). The authors reported the speed increase for selecting features over a dataset of 768 features and 300 samples with regard to different machines and showed a near linear runtime. Later on Liu et al. investigated the use of PGA for gene selection which can naturally correspond to the features (attributes) in the machine learning areas (Liu, Iba, & Ishizuka, 2001). The fitness function in their study was defined to include the classification performance and the size of the subset. In order to deal with the trade-off setting i.e. weighting the objectives of minimum classification error rate and minimum number of features, the multi-objective genetic algorithm was proposed by Oliveira et al. (Oliveira, Sabourin, Bortolozzi, & Suen, 2003).

Recently, Brown et al. formulated the feature selection task as a conditional likelihood problem and derived several mutual information heuristics based on this novel formulation (Brown, Pocock, Zhao, & Luján, 2012). Those feature selection criteria are,
thus, viewed as approximate maximisers of the likelihood. In this study, various questions related to the stability and relation between the given criteria are empirically discussed and evaluated. Based on the experiments, the authors recommend the joint mutual information (JMI) criterion for feature selection since it has the best trade-off between accuracy and stability.
CHAPTER 3
BACKGROUND

In this section, a brief overview of the problem of feature selection is given, followed by a set of examples that reveal the complication level of the feature selection problem. In addition, the feature selection model is reviewed with attention to the wrapper feature selection approach.

3.1 Problem Definition

Many heuristic criteria for feature selection try to optimize a definition of feature ‘relevance’ and ‘redundancy’ (Brown et al., 2012). At a conceptual level, removal of irrelevant or redundant features and selection of relevant ones is the basis of feature selection problem. Tasks of feature selection originated on this basis, nevertheless, may not infer optimality given a particular induction algorithm.

Let the input to a supervised learning algorithm be a dataset \( D \) of \( N \) training samples and \( M \) features. Each instance \( X \) is an element of the set \( F_1 \times F_2 \times ... \times F_M \) where \( F_i \) is the domain of the \( i \)th feature \( X_i \). Training instances are tuples \( (X, Y) \) where \( Y \) is the target classification variable or label. Here, \( X \in \mathbb{R}^M \) and \( Y \in \{\omega_1, \omega_2, ..., \omega_C\} \) for which \( C \) is the total number of classes. The \( i \)th sample of a dataset \( D \) is represented by \( x^{(i)} \).
Given a learning algorithm $L$, a sample of data $S$, a and a feature subset $A$, the feature selection problem is to find from the $M$-dimensional space, $\mathbb{R}^M$, a subspace of $m$ features, $\mathbb{R}^m$, that ‘optimally’ characterize and accurately predict the label $Y$.

The total number of subspaces is $2^M$, and with a predefined subspaces size of $m$, the number of subspaces with dimensions no larger than $m$ is $\sum_{i=1}^{m} \binom{M}{m}$.

Based on different definitions of relevance, many studies introduced different feature selection criterion to optimize the definition. Brown et al., on the other hand, propose to derive the selection criterion starting from a clearly specified objective function instead of trying to define it (Brown et al., 2012).

3.1.1 Relevance and Optimality

The definition of relevance has been explored by many different studies (Blum & Langley, 1997; Kohavi & John, 1997; Yu & Liu, 2004). The definition of Kohavi and John was based on two degrees of relevance which are required: weak and strong (Kohavi & John, 1997). A feature $X_i$ is strongly relevant if removal of $X_i$ changes the performance of an optimal Bayes classifier. A feature $X_i$ is weakly relevant if it is not strongly relevant and there exists a subset of features $A$ such that the performance of a Bayes classifier on $A \cup \{X_i\}$ is not worse than the performance on $A$. In other words, a feature $X_i$ is weakly relevant if it is possible to remove a subset of the feature so that $X_i$ becomes
strongly relevant (Blum & Langley, 1997). A feature is irrelevant if it is not strongly or weakly relevant.

**Definition 1** (*Strong relevance*). A feature $X_i$ is strongly relevant iff there exists some values $x_i, y,$ and $a_i$ for which $p(X_i = x_i, A_i = a_i) > 0$ such that

$$p(Y = y | X_i = x_i, A_i = a_i) \neq p(Y = y | A_i = a_i)$$  \hspace{1cm} (1)

**Definition 2** (*Weak relevance*). A feature $X_i$ is weakly relevant iff it is not strongly relevant, and there exists a subset of features $A_i'$ of $A_i$ for which there exists some $x_i, y,$ and $a_i'$ with $p(X_i = x_i, A_i' = a_i') > 0$ such that

$$p(Y = y | X_i = x_i, A_i' = a_i') \neq p(Y = y | A_i' = a_i')$$  \hspace{1cm} (2)

These definitions as stated are independent of the learning algorithm responsible for inducing a structure capable of accurately recognizing new instances. Therefore, a definition that takes a learning algorithm $L$ was presented (Caruana & Freitag, 1994b). Blum and Langley rephrased it as stated in Definition 3 (Blum & Langley, 1997).

**Definition 3** (*Feature Usefulness*). Given a learning algorithm $L$, a sample of data $S$, and a feature set $A$, a feature $X_i$ is useful to $L$ with respect to $A$ if using the feature set $\{X_i\} \cup A$ allows the hypothesis that $L$ produces to achieve better accuracy than using just the feature set $A$.  

26
Based on a general view of the definitions, two terms appear to affect the design of a feature selection method: relevance and optimality. Relevance from one side refers to the set of features preserving the original class distribution. Optimality, on the other side, is restricted by the hypothesis spaces and the performance of a given induction algorithm on a subset of features. If a feature is relevant, this does not necessarily imply that it is part of the optimal features set given a specific induction algorithm and vice versa (Kohavi & John, 1997).

3.1.2 Redundancy

Redundant features are those which are highly correlated and may lead to extra computational and processing time (Brown et al., 2012). Given a perfectly correlated feature, no additional information is gained by adding it to the dataset (Guyon & Elisseeff, 2003). Removing the redundant features would preserve the same information about the dataset and help to overcome any computational burden. Several measures can be used to estimate the degree of correlation between variables such as Pearson correlation coefficient and mutual information measures.

3.2 Revealing Examples

Feature selection is a problem that does not hold clear and specific rules to generate an exact and complete solution. A set of examples that reveal the complexity level of the problem are introduced in this section.
3.2.1 Redundant Variables and Separability

Given two random variables that are independently and identically distributed over a Gaussian distribution with a standard deviation of 1, the scatter plot for these two random variables is shown in Figure 1 (a) such that they are centered at (-1, -1) and (1, 1), respectively.

After a 45 degree rotation as in Figure 1 (b), if projected on the x-axis the variables becomes more separable and better convenient for classification since given the rotation matrix $\begin{bmatrix} \cos(45) & -\sin(45) \\ \sin(45) & \cos(45) \end{bmatrix}$, the distance between the centers becomes $2\sqrt{2}$ instead of 2. Through rescaling the x-axis by dividing by $\sqrt{2}$, the distance between the centers is going to be 2 given the new feature, yet, the within class standard deviation is reduced by a factor of $\sqrt{2}$. In General, averaging $n$ i.i.d random variables results in a new variable that preserves a reduction of standard deviation by a factor of $\sqrt{n}$ (Guyon & Elisseeff, 2003). This variable can be viewed as a redundant one since it is just a different representation of the information presented by other variables. However, it has led to a better separability between the classes in the given example. Thus, a feature that is redundant does not necessarily degrades the performance of detecting classes.
Figure 1: XY-Projection for Two IID Datasets such that the upper side of the figure shows a regular XY-plot and the lower one is the plot for the same datasets with flipped axes so that we get a YX-plot (a) before rotation and (b) after 45 degree rotation (b); Figures from (Guyon & Elisseeff, 2003)

3.2.2 Useless Variables and Complementarity

Figure 2: Two Different Examples are Illustrated in Parts (a) and (b); (a) XY-Projection for Two Class Conditional Distributions with Identical Covariance Matrices (b) XY-Projection for XOR-like problem; Figures from (Guyon & Elisseeff, 2003)
Classification tasks, sometimes, are represented by a large number of features to distinguish different classes or diagnostic groups. In such cases, reducing the dimensionality of the data through filtering a subset of features viewed as useless is tempting. The decision based on evaluating a feature independently from others, however, may not lead to improved class separability.

In Figure 2, two examples were constructed to clarify how useless variables can be useful if grouped with other ones. Figure 2 (a) illustrates two class conditional distributions having identical covariance matrices. Projection to one of the axes produces separate centers. Yet, the second axes would be useless. So, while one variable could be useful, the two variables together show a complete separability.

Regarding Figure 2 (b), the classes are constructed given the famous XOR problem. At coordinates (0,0), (0,1), (1,0), and (1,1) of a unit square, four Gaussian distributions were placed. Specified by values of the truth table of the XOR function, the distributions are assigned to different classes such that \(((0, 0), (1,1)) \in \omega_1\) and \(((0, 1), (1,0)) \in \omega_2\). As in the previous example, it can be noted that neither projection on the axes would achieve class separation. Nevertheless, a non-linear function is capable of easily grouping the classes in the two dimensional space.
3.2.3 Relevance and Optimality

Over a Boolean space of three variables \(X_1, X_2, X_3 \in \{0,1\}^3\), let the target concept function be defined such that \(f(X_1, X_2, X_3) = (X_1 \land X_2) \lor X_3\) (Kohavi & John, 1997). Given this setting, all the features are considered relevant given any practical definition. If the hypothesis function is defined as the conjunction of literals, then, the only optimal feature subset is \(\{X_3\}\) which achieves an accuracy of 87.5%.
CHAPTER 4

FEATURE SELECTION MODEL

The feature selection problem has been approached from different perspectives. Prior to exploring the different approaches, presenting a general model from which the approaches originate is helpful.

![Feature Selection Model Diagram](image)

*Figure 3: Feature Selection Model*

The feature selection model is illustrated in Figure 3 which is similar to the model presented by Dash and Liu (Dash & Liu, 1997). Several steps need to be defined for a feature selection algorithm in order to achieve its task.

*Search procedure* is the step responsible for the strategy of searching and exploring the different solutions in the space of features. The search can initially start from an empty set, given the full features, or using a subset of features. The search procedure, then, progresses according to a pre-defined operator such as compound operators (Kohavi &
Sommerfield, 1995), branch and bound (Narendra & Fukunaga, 1977) and evolutionary operators (Siedlecki & Sklansky, 1989).

Selection criterion evaluates the subset of features under examination and returns a function value. Based on the defined function, the best value refers to best selected features’ subset. Large number of selection criteria was introduced in the literature and applied for feature selection including monotonic functions such as Mahalanobis distance and Bhattacharyya distance (Narendra & Fukunaga, 1977), correlation measure as in (Hall, 1999) and mutual information feature selection as introduced in (Hanchuan, Fuhui, & Ding, 2005). In addition, the classification error rate and number of features were applied as an evaluative criterion for a given feature subset (Yang & Honavar, 1998).

For a base case or a trivial solution, the search would stop after checking a stopping criterion as the dashed line demonstrates in Figure 3. In addition, following every function evaluation, the stopping criterion is checked. For example, time limit, number of iterations or generation, and number of selected features can fit for a stopping criterion. Finally, the selected feature subset is preferably validated via holding different tests as relevance test, error rate or insignificance of the solution. Established benchmarks of artificial datasets or real ones could serve for the validation step as well as comparison between various methods.
The presented feature selection model can be utilized for categorizing feature selection methods into different groups as presented in (Dash & Liu, 1997; P. Somol et al., 2010). For the purpose of this study, we introduce the categorization with respect to selection criteria. Based on this categorization, feature selection methods may be divided into filter methods, wrapper and embedded methods.

4.1 Filter Methods

Methods that select features according to a specific criterion independent of the classification error are called filter methods. Generally, they constitute a pre-processing step for a supervised learning task. Measures based on distance, correlation, mutual information could serve as a selection criterion for the filtering method (Dash & Liu, 1997; Guyon & Elisseeff, 2003). The measures can, also, be formulated to reflect different degrees of redundancy and relevance (Dash & Liu, 1997). In Equation (3), for example, the filtering measure is designed to reflect reducing redundancy and increasing relevance. Minimizing this equation would result in minimizing the correlation between every feature $X_i$ and all the remaining features $X_j$ such that $i \neq j, j > i$. Moreover, minimizing the negative term representing the correlation between every feature $X_i$ and the target variable $Y$ leads to a maximized relevance.

$$\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \text{corr}(X_i - X_j) - \sum_{i=1}^{m} \text{corr}(X_i - Y) \quad (3)$$
RELIEF (Kira & Rendell, 1992), FOCUS (Almuallim & Dietterich, 1991) and mRMR (Hanchuan et al., 2005) are three well-known filter methods for feature selection. While RELIEF weights features individually according to their relevance with the target, FOCUS involves a greater degree of search for selecting a minimal subset of features that perfectly discriminate between classes. Similar to FOCUS, the mRMR considers relevance as well as redundancy but based on an information-theoretic metric for feature selection.

4.2 Wrapper Methods

Methods that select features based on the classification error of a given induction algorithm are called wrapper methods. Several works have studied the wrapper model of feature selection (Siedlecki & Sklansky, 1989; John, Kohavi, & Pfleger, 1994; Dash & Liu, 1997; Kohavi & John, 1997). In this study, we, also, focus on the wrapper model of feature selection with consideration of the proper design of the objective function. In addition, the study highlights an effort to measure the generalization ability of the wrapper model.

Wrapper methods are often criticized because they seem to require massive computational time. Every feature subset in the search space is to be evaluated by measuring the classification error after training the induction algorithm. This would require more time, in general, when compared to simple filter or ranking methods such
as the single variable ranking ones. The choice, however, of the search and learning methods may lead to an improved computational time (Guyon & Elisseeff, 2003).

4.3 Embedded Methods

Methods that incorporate filtering or ranking features into the model estimation process are called embedded methods (Guyon & Elisseeff, 2003). A well-known example for embedded feature selection methods would be decision trees. Decision trees such as CART and C4.5 have a built in feature selection (Breiman, 1984; J.R. Quinlan, 1993). While the split nodes of the tree are chosen based on a selection criterion, the training error rate serves the cause of validating or pruning the model. Another example of the embedded methods is the support vector machine recursive feature elimination (SVM RFE) for feature selection (Guyon, Weston, Barnhill, & Vapnik, 2002). In this method, a feature ranking method is applied on the dataset to generate a surviving subset of features after removing the irrelevant ones. Then, the computed weight vector generated by training a linear support vector machine is used to build a ranked list of features.

In general, filter methods are faster than embedded methods which are, in turn, faster than wrapper ones. Regarding overfitting, wrapper methods are more likely to overfit given and they are biased to a specific learning algorithm. On the other hand, embedded methods are less likely to overfit but given their learning component, they would be more likely to overfit than the filter methods (Brown et al., 2012).
CHAPTER 5

SUPERVISED MACHINE LEARNING: CONCEPTS AND DEFINITIONS

The problem of discovering patterns in data and extracting information is the core aim of machine learning. The pattern recognitions problems can be categorized as supervised learning, unsupervised learning and reinforcement learning tasks (Bishop, 2006). The supervised learning task is the focus of our study with the aim of enhancement and improvement by the means of feature selection.

5.1 The Classification Task

In general, given a set of examples forming a dataset or a database with their corresponding target vectors, this is called a supervised learning problem. If the target values belong to one of a finite number of discrete categories, then the problem is known as classification problem. The goal of classification, thus, is to take an input vector \( x \) and assign it to one of \( C \) discrete classes \( \omega_c \) such that \( c = 1, ..., C \) (Bishop, 2006). Based on this assignment, the space is divided into decision regions with decision boundaries based on a hypothesis function or a learning algorithm \( L \).

5.2 Learning Algorithms

A learning algorithm or induction algorithm forms a model that represents the knowledge induced from example data (Hall, 1999). Different learning algorithms represent this induced knowledge differently. The C4.5 (J.R. Quinlan, 1993), for
example, represent the learning model as a decision tree from which a set of decision rules can be generated.

Throughout this thesis, three classification methods that are widely used are applied as a basis for comparing different wrapper-based feature selection methods. These classification methods or classifiers include Naïve Bayes Classifier, Decision Trees and Support Vector Machines SVM. These classifiers are well-known in the field of machine learning and they have been applied for feature selection in many different studies (Bradley & Mangasarian, 1998; Huang & Wang, 2006; Kohavi & John, 1997; Koller & Sahami, 1996). The following sections briefly review the applied variant of each algorithm.

5.2.1 Naive Bayes Classifier

Based on the Bayes probability rule which assumes that features are independent of each other, the Naïve Bayes Classifier NBC assigns for each sample a target label to maximize the posterior probability. In other words, the probability of observing a conjunction of feature values for one instance i.e. $x_1, x_2, ..., x_m$ given the target value for this instances $\omega_c$ is just the product of the probabilities for the individual attributes such that $P(x_1, x_2, ..., x_m | \omega_c) = \prod_i P(x_i | \omega_c)$ (Mitchell, 1997). Since for each instance, the most probable target is to be assigned, the NBC can be represented as in Equation (4).
\[ \omega_{NB} = \arg\max_{\omega_c \in \Omega} P(\omega_c) \prod_i P(x_i|\omega_c) \]  

(4)

where \( \omega_{NB} \) denotes the target value output by the naïve Bayes classifier.

Estimating the probability \( P(x_i|\omega_c) \) for discrete values is straightforward. However, if the feature values are continuous, then the data distribution associated with each class should be estimated. A common assumption is that, within each class, the data follows a Gaussian normal distribution (John & Langley, 1995). In our implementation, this assumption was satisfied such that \( P(x_i|\omega_c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(x_i-\mu_c)^2}{2\sigma_c^2}} \) where \( \sigma_c^2 \) is the variance of the values in \( x \) associated with class \( c \) and \( \mu_c \) is the mean for these values within class \( c \).

The naïve Bayes classifier is to some extent immune to irrelevant features but its performance degrades quickly if correlated features which violate the NBC assumption are added. It should be noted, additionally, that the performance of NBC improves with the removal of relevant features in some cases (Kohavi & John, 1997).

5.2.2 C5.0 Decision Tree Generator

Building decision trees, in general, is based on finding the feature \( X_i \) that best divides the training data according to some index such as the information-based measures (Kotsiantis, Zaharakis, & Pintelas, 2007). After splitting the data samples, recursively, the process is applied on the generated sub-lists.
The most well-known algorithm for generating and building decision trees is the C4.5 algorithm introduced by Quinlan (J.R. Quinlan, 1993). It is an extension to the iterative dichotomizer (ID3) algorithm that deals with discrete as well as continuous values and applies pruning after creating the tree (J.R. Quinlan, 1986, 1993, 1996).

Gain ratio is an information-based splitting criterion for C4.5 (J.R. Quinlan, 1996). According to Quinlan (J.R. Quinlan, 1996), let \( p(D, j) \) be the proportion of cases in \( D \) that belongs to the \( j \)th class, then, the residual uncertainty or entropy about the class to which a case in \( D \) belongs can be stated as

\[
entropy(D) = - \sum_{j=1}^{c} p(D, j) \times \log_2(p(D, j))
\]  

(5)

Given a test \( T \) with \( k \) outcomes, the information gain is, then, expressed as

\[
Gain(D, T) = entropy(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times entropy(D_i)
\]  

(6)

The potential information of splitting a set of cases depends on knowing the subset \( D_i \) to which a case falls as Equation (7) highlights.

\[
Split(D, T) = - \sum_{j=1}^{k} \frac{|D_i|}{|D|} \times \log_2\left(\frac{|D_i|}{|D|}\right)
\]  

(7)
Based on the gain ratio which is the ratio of the information gain of a test to its split information, the variable or feature with the maximum gain ratio is selected as the split.

Faster and more memory efficient implementation compared to C4.5 was introduced in C5.0 by Quinlan (J. R. Quinlan, 2012). In our study, we used the available single-threaded version of C5.0 for Linux.

Several studies proposed feature selection methods based on decision trees as a wrapper-model in which different search strategies were applied including hill climbing and genetic algorithms and were compared with other approaches (Caruana & Freitag, 1994a; Cherkauer & Shavlik, 1996; Das, 2001; Kohavi & John, 1997).

5.2.3 Multi-Support Vector Machine

Support vector machine (SVM) was formulated as an optimization problem to build a classifier with a maximum margin and minimum error rate and deal with non-separable training data through high-dimensional mapping (Boser, Guyon, & Vapnik, 1992; Cortes & Vapnik, 1995). While the original formulation was intended for two-group classification problems, the case of multi-class data can be viewed as one-against-one or one-against-all to exploit the original formulation (Hsu & Lin, 2002). The original formulation of SVM, however, can be extended to solve \( C \)-class problems (Weston & Watkins, 1998). This extension may help in capturing the correlation between different
classes since it does not break the multiclass problem into multiple independent binary problem (Crammer & Singer, 2002).

Given classifier $L(x)$ parameterized by a matrix $W$ of size $C \times M$ such that $W_r$ is the $r$th row of $W$, Crammer and Singer (Crammer & Singer, 2002) formulated the M-SVM optimization problem as defined by Equation (5)

$$
\min_{W, \xi} \frac{1}{2} \beta ||W||^2 + \sum_{n=1}^{N} \xi_n
$$

subject to: $\forall_{n,c}$ $W_{y_n} \cdot x_n + \delta_{y_n,c} - W_{c} \cdot x_n \geq 1 - \xi_n$

In Equation (5), $\beta > 0$ is a regularization constant, $W_{c} \cdot x_n$ represents the confidence and the similarity score for the $c$ class with instance $x_n$ while $W_{y_n} \cdot x_n$ is the confidence value of the correct label. If $y_n = c$, then $\delta_{y_n,c} = 1$ and 0 otherwise. The $||W||_2^2$ is the $l_2$-norm of $W$ since a matrix $W$ of a small norm is desired and $\xi_n$ is a slack variable associated with each sample that a multiclass machine is not able to separate linearly.

The constraint in the given formulation is derived from a linear upper bound that reflects a value of zero if the confidence value for the correct label is larger by at least one than the confidences assigned to the rest of the labels (Crammer & Singer, 2002).

Guermeur and Monfrini introduced a variant of this formulation M-SVM² presenting the main property of the 2-norm SVM by inferring the corresponding generalized radius-margin bound (Guermeur & Monfrini, 2011).
In this study, we use the M-SVM\(^2\) which was implemented in MSVMpack, an open source software package with parallel implementation for several M-SVMs (Lauer & Guermeur, 2011).

5.3 Performance Evaluation

The decision on the quality of a machine learning algorithm is based on evaluating its performance. Likewise, performance metrics are needed to measure the effectiveness of a feature selection method and compare it to other approaches. A comparative study of twelve feature selection metrics was presented by Forman (Forman, 2003). The study highlighted that in case of substantial class skewedness, error rate and accuracy can be weak indicators of performance. Since many induction algorithms are focused on the goal of accuracy or error rate, it is still desirable to use these measures as performance evaluation metrics for feature selection. In order to overcome their limitation when class imbalance or skewedness are present, techniques such as under-sampling, over-sampling and cost-modifying can be applied (Japkowicz & Stephen, 2002). In this study, error rate measure is used as a performance evaluation for the feature selection method. In case of high class skewedness, under-sampling is intended to maintain the error rate as a good performance indicator.

The statistical error or accuracy measures can be estimated by various estimation methods such as cross-validation, holdout, bootstrap and the jackknife (Efron & Gong, 1983; Kohavi, 1995). The holdout estimation method partitions the data into two
mutually exclusive subsets called a training set and a test set or holdout set (Kohavi, 1995). While other estimation methods may be more stable and indicate the generalization ability more than this method, the holdout method is more convenient in the context of large datasets. Given the need to experiment with large datasets, train classifiers many times, hide a portion of data for the sake of comparing generalization power of different feature selection methods, the holdout estimation method was the choice of sampling in our study.

In the experiments, 70% of the data was used for training, 15% for validation and 15% is left for testing. Whenever there is a comparison between different learners, the testing set which was not part of any selection process or optimization is used.
CHAPTER 6
GENETIC ALGORITHM

Genetic algorithm (GA) is an optimization search methodology working in binary spaces or continuous ones (if modified) and based on analyzing and modifying a population of candidate solution at a given time. Genetic algorithms were developed initially by Holland et al. in the 1960s and 1970s and their application to machine learning was highlighted by his student Goldberg (Goldberg, 1989; Holland, 1975). They are robust techniques able to attack problems with complex search spaces of multiple local optima and non-differentiable functions (Storn & Price, 1997; Whitley, 1994).

A major trend in feature selection is the design and development of approaches based on genetic algorithms. The problem setting is originally based on representing a candidate subset of features by a string with $M$ binary genes in which a value of 1 indicates inclusion of the corresponding feature while 0 indicates exclusion. Guided by applying a set of randomized operators including selection, mutation and crossover to a pre-defined fitness function, the GA algorithm progresses for selecting the best feature subset. Bautista and Vila surveyed approaches presented in the literature for selecting relevant features using Genetic Algorithms and highlighted different utilized parameters as well as various defined fitness functions (Martin-Bautista & Vila, 1999).
Inherently, the GA can be designed in several ways to exploit parallel machines. A feature selection approach based on parallel genetic algorithm would have the advantage of being applied on large datasets and the ability to explore additional solutions in tolerable time limits.

6.1 Parallel Genetic Algorithm

In general, the design approaches for parallel genetic algorithms (PGA) can be classified into four categories: global parallelization, coarse and fine grained algorithms, and hybrid approaches (Cantú-Paz, 1995). The global approach is a master-slave parallel model based on distributing the evaluation of fitness of a single population to several computing node. Since selection, mutation and crossover consider the entire population, the approach is known as global parallel GAs (Cantú-Paz, 1998). Fine grained model is, also, based on a single population but the operators are restricted to a small neighborhood (Cantú-Paz, 1998). On the other hand, coarse grained GAs are multiple-population GAs which consider migration operator between the population (Cantú-Paz, 1998). Finally, the hybrid method uses some combination of the first three methods (Cantú-Paz, 1995). Some implementation considerations for parallel GAs over parallel computing paradigms are further discussed in (Munawar, Wahib, Munetomo, & Akama, 2008).
The first widely distributed library for parallel genetic algorithm is PGAPack (Levine, 1996; Levine, Hallstrom, Noelle, & Walenz, 1999). The implementation of the examined feature selection methods in this study is based on PGAPack.

6.2 PGAPack Parallel Model

PGAPack supports the parallel implementation of a master/slave single population algorithm in which the master is responsible for executing all steps of the genetic algorithm and function evaluations are sent to the slave processes as illustrated in Figure 4. In addition, the PGAPack library supports four native data types: binary-valued, integer-valued, real-valued, and character-valued strings.

![Figure 4: PGAPack Master-Slave Model (Levine, 1996)](image-url)
PGAPack’s implementation is based on the message-passing interface (MPI) standard and in particular, the single program, multiple data (SPMD) model (Levine, 1996; Snir, Otto, Walker, Dongarra, & Huss-Lederman, 1995). MPI is a standardized and portable message passing system that defines a set of functions and their behavior. PGAPack exploits several useful features of MPI including collective communication operations, barriers, and the ability to send and receive arbitrary structures. It is worthy to note that the parallel implementation of PGAPack will produce the same results as the sequential implementation but usually faster.

In this study, PGAPack was installed on a Beowulf class heterogeneous computer cluster. This cluster called Noor is based on Intel Xeon (Nehalem) processor in which each node has 4 dual-core processors equipped with high-performance, low latency Infiniband interconnect ideal for a stable, high performance resources to run a wide diversity of scientific application (Team, 2010).

Several aspects from the GAs must be determined when they are used to the feature selection problem. Following, is a discussion of the parameter settings in our study for each aspect of the GAs including their randomized operators. In Table 1, a summary for the chosen parameters is presented.
6.2.1 Initialization

The representation of the chromosomes of the population is based on the binary codification. The $i$th position or gene of a selected chromosome $r$ is equal to 1 if the feature is present and 0, otherwise. The length of the chromosomes is determined by the number of the features of the dataset. Normally, the population size is set to 50 or 100 and in some cases to 200 (Martin-Bautista & Vila, 1999). In our experiments, we use a population size of 100 such that each gene is set to 1 or 0 randomly with an equal probability. Nevertheless, a chromosome with all gene values set to 1 is always present in the initial population to preserve the case if the existence of all features is more desirable than selecting a subset.

6.2.2 Population Replacement

Given the master/slave parallel model over a dedicated 64-core machine, the new population in each generation is created consisting of sixty three individuals or strings via recombination. The remaining most fit strings are preserved in the new generation. This population generation scheme in which parents and offspring can typically co-exist is a variant of the standard genetic algorithm (Whitley, 1989). For the portion to be generated, selection step of the GA is based on tournament type of selection.
6.2.3 Selection

In genetic algorithms, a selection mechanism is applied to insert selected individuals into a mating pool which is used to generated new offspring (Brad & David, 1995). In the case of tournament selection, a number of individuals with the highest fitness among the competitors are selected and thus, the formed mating pool has a higher average fitness value than the whole population (Brad & David, 1995). The tournament selection scheme is the selection choice for preparing the newly generated feature subsets in our experiments.

6.2.4 Crossover

The crossover operator produces two new offspring by taking bits from each parent string and combining them. Different kinds of crossover define the combination scheme for the bits. In our choice of two-point crossover, the intermediate segments of one parent are placed into the middle of the second parent string (Mitchell, 1997). The probability of crossover is usually specified by values close to 1 while the probability of mutation is based on lower values between 0.01 and 0.15 (Siedlecki & Sklansky, 1989). Various studies considered 0.8 as a proper setting for the crossover rate (Deb, Agrawal, Pratap, & Meyarivan, 2000; Hong & Cho, 2006; Leung, Lam, Ling, & Tam, 2003). In our study, a crossover of 0.8 is used.
6.2.5 Mutation

The mutation operator is applied to produce small random changes in a given parent. By default, PGAPack applies mutation to chromosomes that did not undergo crossover (Levine, 1996). Based on some preliminary runs, the parameter setting for the mutation rate was 0.08. This value is close to the typical values used in standard genetic algorithm and to the value setting introduced in (Siedlecki & Sklansky, 1989).

<table>
<thead>
<tr>
<th>Pop. Size</th>
<th>Replacement</th>
<th>Mutation</th>
<th>Crossover</th>
<th>Stopping Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>63</td>
<td>0.08</td>
<td>0.8</td>
<td>100 ltr</td>
</tr>
</tbody>
</table>

Table 1: Parameters Setting for the GA

6.2.6 Stopping Criteria

The iteration limit or number of generations serves as the stopping criteria in the designed experiments. While it might be desirable to run genetic algorithms for hundreds of generations, a reasonable choice in our case given the training time of different classifiers was 100 generations. Based on this choice, the number of criterion function evaluations reaches about 6,400 function evaluations.
CHAPTER 7

PROPOSED METHOD

The proposed system has four main components; search procedure, selection criterion, stopping criterion and validation step. The exact framework for connecting these four steps has been implemented in our system as depicted in Figure 5. Briefly, the search procedure is based on a master/slave single population genetic algorithm. The selection criterion is based, in general, on the error rate of a specific classification method or induction algorithm. For a specified number of generations, the feature selection model seeks to select the best optimal feature subset. After the search procedure terminates, a separate validation set is used to estimate the final performance of the classification method.

Figure 5: Proposed System Model
7.1 Fitness Function

The fitness function for wrapper-based feature selection methods is generally an error rate or accuracy and in some cases, with other penalty factors. Some formulae that might depict some variation of fitness functions applied in wrapper-based feature selection approaches are shown in Table 2.

Siedlecki and Sklansky proposed a fitness function such that greater values of the error rate, a penalty function rises toward infinity as in Equation (9) of Table 2 (Siedlecki & Sklansky, 1989). The cost term of their function refers to the smallest subset of features. Feature subsets with error rate above threshold $t$ and below $t + s$ receive a small penalty between 0 and 1.
<table>
<thead>
<tr>
<th>Authors</th>
<th>Objective Function</th>
<th>Parameters Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Siedlecki &amp; Sklansky, 1989)</td>
<td>$p(e(a)) = \frac{\exp((e(a) - t)/s) - 1}{\exp(1) - 1}$, $J(a) = l(a) + p(e(a))$</td>
<td>$p(e(a))$: penalty function †, $s$: scaling factor or tolerance margin, $l(a)$: num. of features</td>
</tr>
<tr>
<td>(Punch et al., 1993)</td>
<td>$J(a) = ye(a) + \alpha \left(\frac{nmin/K}{N}\right)$</td>
<td>$\alpha, y$: weighting factors, $nmin$: near neighbor minority set, $K$: num. of nearest neighbors</td>
</tr>
<tr>
<td>(Kohavi &amp; John, 1997)</td>
<td>$J(a) = e + \alpha \ast p(e(a))$</td>
<td>$\alpha$: weighting factor equals 0.1%, $p(e(a))$: num. of features</td>
</tr>
<tr>
<td>(Yang &amp; Honavar, 1998)</td>
<td>$J(a) = e(a) + \frac{p(e(a))}{(1 - e(a)) + 1 + p(e(a))_{\text{max}}}$</td>
<td>$p(e(a))$: sum of measurement costs, $p(e(a))_{\text{max}}$: sum of costs of all features</td>
</tr>
<tr>
<td>(Huang &amp; Wang, 2006)</td>
<td>$J(a) = ye(a) - \alpha \left(\sum_{i=1}^{m} p_i \times w_i \right)^{-1}$</td>
<td>$p_i$: cost of feature $i$, $w_i$: 1 if feature is selected and 0, otherwise</td>
</tr>
<tr>
<td>(Zhu, Ong, &amp; Dash, 2007)</td>
<td>$J(a) = e(a)$</td>
<td>Cost or penalty is considered in the algorithm</td>
</tr>
<tr>
<td><strong>Proposed</strong></td>
<td>$J(a) = \frac{e_{\text{C5.0}}(a) + e_{\text{NBC}}(a) + e_{\text{SVM}}(a)}{3} + \alpha \ast p(e(a))$</td>
<td>Equal Contribution of Different Classifiers</td>
</tr>
<tr>
<td><strong>†</strong></td>
<td>$e(a) = \frac{FP + FN}{TP + FP + TN + FN}$</td>
<td></td>
</tr>
</tbody>
</table>

TP: the number of true positives in $a$
TN: the number of true negatives in $a$
FP: the number of false positives in $a$
FN: the number of false negatives in $a$

Table 2: List of Objective Functions for Wrapper-Based Feature Selection Methods
Given the K-Nearest Neighbor (KNN) algorithm, Equation (10) in Table 2 was formulated to account for a weight setting that gives the maximum class separation (Punch et al., 1993). So, a feature subset that results in getting the majority of the nearest neighbors to the highest value possible (K) is preferable.

The remaining equations in Table 2 define penalty or cost terms based on similar intuition. Yang and Honavar refers to the penalty term as actual costs and risks associated with medical diagnosis, for example (Yang & Honavar, 1998). Others refer to the penalty term as the number of selected features, or actual features’ generation costs if present (Huang & Wang, 2006; Kohavi & John, 1997). In some equations like Equation (10), Equation (11) and Equation (13), weighting factors that define trade-offs between the different objectives are explicitly stated. There is, however, no clear evidence on setting those values for selecting enhanced feature subsets.

In general, the fitness function or objective function based on wrapper feature selection, can be defined as in Equation (16)

\[ J(a) = ye(a) + ap(e) \]  

(16)

In our research, the penalty term is the percentage of the number of the selected features since actual costs are not present for the examined datasets. Specifically, we try to investigate the effect of \( \alpha \) on selecting feature subsets with improved accuracy. In addition, we propose the fitness function presented in Equation (15) which we call
multi-wrapper approach and try exploring its properties when compared to other wrapper-based models. The mean of the percentages of the misclassified instances (i.e. error rates) of C5.0, SVM and NBC is used to evaluate a given feature subset. This function reflects that more than one classifier equally shares the decision on the examined feature subset. Hence, the selected feature subset is one that minimizes several classifiers at the same time.
CHAPTER 8

EXPERIMENTAL DATASETS

It is claimed that in the mining field, there is a lack of experiments with examples for which the size of features in a dataset may vary from one hundred to several thousands (Martin-Bautista & Vila, 1999). Also, a deeper analysis is needed in these cases. Thus, we consider different types of datasets covering low and high dimensional data.

The datasets are chosen to have features of different characteristics and different levels of difficulties observed by example-feature ratios and number of target classes. Validation for feature selection methods is commonly carried out through using artificial datasets and using real-world datasets (Dash & Liu, 1997). Table 3 gives summary of the characteristics of the chosen datasets. The ratio measure in the last column is an indicator of the difficulty level such that a smaller value indicates a more challenging feature selection problem (Brown et al., 2012). It is based on the number of points ($N$), the median arity or the number of categorical values for the discretization ($m$), and the number of classes ($c$). Since we do not discretize the data, $m$ becomes the number of features and the ration would be $\frac{N}{mc}$. All datasets except for the Protein Disulfide dataset were acquired from the University of California at Irvine repository at which complete documentation for every dataset is supplemented (Frank & Asuncion, 2010). The Protein Disulfide dataset was obtained from the computation biology research center at
King Abdullah University of Science and Technology (Ibrahim & Bajic, 2012). A brief overview for only the selected artificial datasets is provided.

<table>
<thead>
<tr>
<th>No.</th>
<th>Dataset</th>
<th>Size</th>
<th>Features (No.)</th>
<th>Feature Type</th>
<th>Sparsity</th>
<th>Output Classes (No.)</th>
<th>Class Balance Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Breast Cancer Wisconsin (Diagnostic)</td>
<td>569</td>
<td>30</td>
<td>Real</td>
<td>Condense</td>
<td>2</td>
<td>212:357</td>
</tr>
<tr>
<td>2</td>
<td>Hill-Valley</td>
<td>1212</td>
<td>100</td>
<td>Real</td>
<td>Condense</td>
<td>2</td>
<td>1:1</td>
</tr>
<tr>
<td>3</td>
<td>Madelon</td>
<td>2600</td>
<td>500</td>
<td>Integer</td>
<td>Condense</td>
<td>2</td>
<td>1:1</td>
</tr>
<tr>
<td>4</td>
<td>Internet Advertisements</td>
<td>3279</td>
<td>1558</td>
<td>Integer, Real, Categorical</td>
<td>Semi Sparse</td>
<td>2</td>
<td>459:2820</td>
</tr>
<tr>
<td>5</td>
<td>Protein Disulfide</td>
<td>7401</td>
<td>2808</td>
<td>Integer, Categorical</td>
<td>Semi Sparse</td>
<td>2</td>
<td>2832:4569</td>
</tr>
<tr>
<td>6</td>
<td>Gisette</td>
<td>7000</td>
<td>5000</td>
<td>Integer</td>
<td>Semi Sparse</td>
<td>2</td>
<td>1:1</td>
</tr>
<tr>
<td>7</td>
<td>Dexter</td>
<td>600</td>
<td>20000</td>
<td>Integer</td>
<td>Sparse</td>
<td>2</td>
<td>1:1:1</td>
</tr>
<tr>
<td>8</td>
<td>Waveform Database Generator (Version 2)</td>
<td>5000</td>
<td>40</td>
<td>Real</td>
<td>Condense</td>
<td>3</td>
<td>1:1:1:1</td>
</tr>
<tr>
<td>9</td>
<td>Statlog (Landsat Satellite)</td>
<td>6435</td>
<td>36</td>
<td>Integer</td>
<td>Condense</td>
<td>6</td>
<td>1533:703:1358:626:707:1508</td>
</tr>
<tr>
<td>10</td>
<td>Semeion Handwritten Digit</td>
<td>1593</td>
<td>256</td>
<td>Categorical</td>
<td>Semi Sparse</td>
<td>10</td>
<td>Equal</td>
</tr>
<tr>
<td>11</td>
<td>Corral</td>
<td>160</td>
<td>18</td>
<td>Categorical</td>
<td>Condense</td>
<td>2</td>
<td>70:90</td>
</tr>
<tr>
<td>12</td>
<td>m-of-n-3-7-10</td>
<td>1324</td>
<td>20</td>
<td>Categorical</td>
<td>Condense</td>
<td>2</td>
<td>292:1032</td>
</tr>
<tr>
<td>13</td>
<td>Monk1</td>
<td>556</td>
<td>24</td>
<td>Categorical</td>
<td>Condense</td>
<td>2</td>
<td>1:1</td>
</tr>
<tr>
<td>15</td>
<td>Monk2</td>
<td>601</td>
<td>24</td>
<td>Categorical</td>
<td>Condense</td>
<td>2</td>
<td>206:395</td>
</tr>
<tr>
<td>16</td>
<td>Monk3</td>
<td>554</td>
<td>24</td>
<td>Categorical</td>
<td>Condense</td>
<td>2</td>
<td>1:1</td>
</tr>
</tbody>
</table>

Table 3: Summary of Experimental Datasets. Datasets 11-16 are artificial benchmark datasets.
8.1 CorrAL

Given 32 instances and six Boolean features \((A_0, A_1, B_0, B_1, I, C)\), this data was designed to represent a Boolean target concept of \((A_0 \land A_1) \lor (B_0 \land B_1)\) (John et al., 1994). Nevertheless, a uniformly random irrelevant feature \(I\) was generated and added to the dataset as well as a correlated feature \(C\) that matches the class label 75% of the time. In our case, these features were duplicated three times resulting in 18 features. This duplication is, purposely, applied so that an exhaustive enumeration is not easily possible using GA.

8.2 m-of-n-3-7-10

This is a binary dataset composed of 10 attributes including 3 irrelevant features. The relevant features are \(X_2, X_3, X_4, X_5, X_6, X_7, X_8\). There is neither duplicate data nor noise introduced in this dataset. For a convenient setting for the GA, a multiple of two of the features was considered and thus, dealing with 20 features.

8.3 Monk1, Monk2 and Monk3

These datasets have binary classes and six discrete attributes \((A_1, ..., A_6)\). Table 4 lists the target concept and predefined relevant features for each dataset. The six discrete attributes were duplicated to have twenty four features which is more preferable in the context of GA.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Target Concept</th>
<th>Relevant Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monk1</td>
<td>((A_1 = A_2) \lor (A_5 = 1))</td>
<td>(A_1, A_2, A_5)</td>
</tr>
<tr>
<td>Monk2</td>
<td>((A_n = 1)) for EXACTLY TWO choices of (n) (in ({1,2,...,6}))</td>
<td>(\text{All})</td>
</tr>
<tr>
<td>Monk3</td>
<td>((A_5 = 3 \land A_4 = 1) \lor (A_5 \neq 4 \land A_2 \neq 3))</td>
<td>(A_2, A_4, A_5)</td>
</tr>
</tbody>
</table>

*Table 4: Target Concepts and Relevant Features for the Monk Datasets*
CHAPTER 9
EXPERIMENTS AND RESULTS

Given the reviewed variants of different objective functions for wrappers for feature selection, we intend in this section to draw conclusions given the effect of the weight parameter for the objective of selecting minimum number of features. In addition, various types of comparisons with other feature selection methods are included in this section.

9.1 Design of the Objective Function for Wrappers for Feature Subset Selection

The objective function for wrapper-based feature selection methods is generally an error rate or accuracy and in some cases, with other penalty factors as defined in Equation (16). Through extensive experiments over 15 datasets and based on different classification methods, we draw conclusions for the appropriate setting of the objective function.

9.1.1 How the function values in the generations of the PGA change given different values of $\alpha$?

In order to explore the effects of different values of $\alpha$, weight associated with optimizing the number of features as in Equation (15), the best error rate in every generation of the PGA is considered. For every dataset, four models of wrappers based on PGA for feature subset selection are run. Their best achieved error rates given a
specific value of $\alpha$ are, then, grouped together. Finally, for every generation, the mean is computed over all the datasets.

Figure 6 shows that several values of $\alpha$ progressively improve the error rate through initial generations of the PGA. When $\alpha = 0.25$, there is a continuous improvement in the best error rate through even the later generations of the PGA. The effect of $\alpha$ on performance still varies given different datasets as discussed later. Nevertheless, when $\alpha > 0$, the PGA is still able to challenge the achieved performance given the case of optimizing the error rate only i.e. $\alpha = 0$. These values can be viewed as exploration degrees in the search space.
Minimizing only the classification error rate does not guide the search method to consider minimally sized feature subsets. Thus, in case two different selected feature subsets result in the same error rate, the search method fails to consider the one with the less number of features as in the case of $\alpha = 0$ demonstrated in Figure 7. Thus, there is a persistent need to include the minimum number of selected features in the design of the objective function.
Given that the average number of features of the 15 datasets is around 2029, the size of the selected features is reduced by about 70% as illustrated in Figure 7 for the case of $\alpha = 0.25$. For larger values of $\alpha$, higher percentage rates are attained.

![Comparison of Effect of Different Values of Alpha on the Size of Selected Features in Generations of PGA](Image)

*Figure 7: Comparison of the Size of the Selected Feature Subsets given the Generations of the PGA*
By examining the effect of the weighting values on the aggregate mean error rate of different classification methods, it can be shown that for only 5 datasets, there is no realized improvement as shown in Figure 10 compared to Figure 8 and Figure 9. These datasets are WDBC, Ad., Protein-Disulfide, Wave-form and Statlog. Four dataset out of those has classes that are not well-balanced suggesting that minimizing the number of features for datasets in which classes are not equally distributed may not lead to selecting feature of better performance. Minimizing the error rate instead could lead to the best feature subset. For the Wave-form dataset, the actual performance of the individual classifiers did not improve by minimizing the number of features.

![Effect of Alpha on Performance given Different Real-World Datasets](image)

*Figure 8: Effect of Alpha on Performance given Datasets on which an Improvement or Equal Progress was Realized*
Figure 9: Effect of Alpha on Performance given Datasets on which an Improvement or Equal Progress was Realized

Figure 10: Effect of Alpha on Performance given Datasets on which Worse Performance was Realized
9.1.2 Does $\alpha$ hold an effect on the CPU execution time?

Minimizing the number of the selected features with a reasonable weighting value leads to dramatic savings in the CPU execution time. This is illustrated in Figure 11 in which $\alpha = 0.25$ achieves the best time saving when compared to the other values. When $\alpha \to 0$, additional CPU time is required since smaller feature subsets are not receiving any preference. Drilling down in the plot which is an aggregate of several classification methods, we are able to see that the reason for extra CPU time when including less number of features (i.e. $\alpha = 0.5$, $\alpha = 0.75$ and $\alpha = 1$) is particularly because of the support vector machine (SVM) classifier as demonstrated in Figure 12. SVM fails to converge easily to a desirable specified training error rate when less information about the problem is present.
Figure 11: Effect of Different Alpha Values on CPU Time
9.1.3 Is there a significant difference in the size of the selected feature subsets given different wrapper models?

We would like to examine if similar values of $\alpha$ leads to the selection of feature subsets with various sizes when different wrapper models are considered. For every dataset, each observation in $K_{15 \times 4}$ is the mean of the sizes of the selected features given all values of alpha for a specific wrapper model. Under the null hypothesis that all samples in $K$ are drawn from populations with the same mean, the $p$-value is 0.6995 based on the balanced one-way ANOVA test. Based on the non-parametric Kruskal-Wallis test, under the null hypothesis that all samples belong to the same population or
equivalently, samples are drawn from different populations with the same distribution, the $p$-value is 0.5635. These two statistical tests clearly show that there is no statistical difference in the sizes of the feature subsets when different wrapper model are considered. Figure 13 shows the boxplot of observations in $K$. The result suggests that the major factor that affects the size of the selected feature subset is the $\alpha$ value rather than the classifier type.

![Boxplot for the Sizes of the Selected Features Given Different Wrapper Models](image)

**Figure 13:** Boxplot for the Sizes of the Selected Features Given Different Wrapper Models

### 9.2 Comparison of Different Feature Selection Methods

In order to validate a feature selection algorithm, usually, two validation techniques are followed including (a) comparison of performance reported on real data sets and (b) comparison of selection of relevant features given artificial datasets (Dash & Liu, 1997).
Dash and Liu stated that unless a variety of classifiers are chosen and many statistical different datasets are used, the first validation procedure may lead to wrong conclusions and thus, not suitable (Dash & Liu, 1997). Thus, we attempt in this section to consider a wide variety of classification methods that were not part of any wrapper feature selection model in our study. Several datasets with different statistical properties were, also, considered. Moreover, the second validation procedure is followed to provide a complete view of the chosen feature selection methods. Nevertheless, there is no aim at generalizing that specific methods are better than others and rather the aim is to explore where some wrapper feature selection model stands between other well-known ones.

We now describe the feature selection methods, the classification algorithms used for comparison, and the experimental results based on the two validation procedures.

9.2.1 Feature Selection Methods

Several distinguished feature selection methods were chosen for the sake of comparison and benchmarking in this study. FEAST, a feature selection toolbox, provides a set of implementations of information theoretic filter feature selection algorithms, and an implementation of RELIEF (Brown et al., 2012). In this recent study, the authors found that the best overall trade-off for accuracy/stability was found in the joint mutual information (JMI) and maximum-relevance minimum-redundancy (MRMR) criteria. Thus, we choose them as a basis of comparison. RELIEF, one of the
distinguished methods in feature selection, was also one of the techniques applied for comparison. Specifically, a slightly modified version from the original one implemented in FEAST was used. It considers the near-miss of one class of the available classes and, thus, not efficient as RELIEF-F which is an extension of the main algorithm to deal with multi-class problems (Kononenko, 1994).

A threaded 3-Nearest Neighbor classifier wrapper-based feature selection with Floating search served as an additional feature selection method. A threaded very-high-dimensional feature selection based on multinomial Bhattacharyya distance criterion was, additionally, considered in comparing different models of feature selection. These latest methods were based on an open source available feature selection toolbox FST3 (Petr Somol et al., 2010). They were chosen to explore feature selection guided by floating search and oscillating search. As a base line, a random feature selection is applied and all available features are taken in consideration, also. For the purpose of discovery, the union of the features generated by the individual wrapper-models is an additional selected method for comparison.

General filter selection methods require specifying the number of features to select. In such cases, the size of the features subset generated by the multi-wrapper approach is used as an input parameter for the filter methods.
9.2.2 Classification Algorithms

For evaluating the performance of the proposed wrapper models, a set of classifiers different than those used in the wrappers were chosen. In particular, the $K$-Nearest Neighbor classifier, support vector machine with linear and polynomial kernels, ensemble methods such as AdaBoost and Bagging. The AdaBoost ensemble classifier is based on the work introduced in (Freund & Schapire, 1995; Friedman, Hastie, & Tibshirani, 2000). The Bagging ensemble model is a bootstrap method for sampling the training data based on the work of Breiman (Breiman, 1996, 2001). Classifiers like the linear or quadratic discriminant analysis are not included because of the ill-conditioned covariance matrix for some datasets. Artificial Neural Networks are, also, not included since more parameters need to be optimized such as number of hidden units and hidden layers for every dataset. This would require excessive computational time and complexity level.

9.2.3 Performance Validation

In order to view the performance of every feature selection technique, boxplots for observations of classification’s error rates over 15 datasets are viewed in Figure 14, Figure 15, Figure 16, Figure 17, Figure 18 and Figure 19. In every boxplot, a testing error rate of a specific classifier is considered for comparison.
The boxplots show that there is no single feature selection method which could be regarded as a globally appropriate method for any kind of classifier. Nevertheless, MRMR, JMI and C5.0 based wrapper generalize well compared to other techniques.

In general, the boxplots show that feature selection does not always lead to a better performance as can be observed in Figure 16, Figure 17, Figure 18 and Figure 19. Through examining the observations of different datasets, KNN classifiers are more affected by feature selection than SVM which, in turn, more affected than ensemble models. Regarding SVM, parameter tuning seems to hold more effect on the classification performance.

Interestingly, in cases of AdaBoost and Bagging, feature selection methods have less effect on the classification error rate as demonstrated in Figure 18 and Figure 19. Thus, an ensemble based classifier is less likely to get affected by the type of selected features.

Regarding the multi-wrapper feature selection, it has an advantage over all other methods when the same classifiers incorporated in the wrapper model are considered for comparing the error rates as in Figure 20. Compared to the case when given all features, the error rates are improved by 31.3% on average. The multi-wrapper approach is an expensive method that requires excessive computational time compared
to the other methods. Yet, it shows that a single feature subset could be selected which at the same time minimizes the error rate for different set of heterogeneous classifiers.

Building a multi-wrapper model does not lead to a significant loss in the performance of the classification methods when compared to the individual models as illustrated in Figure 21, Figure 22 and Figure 23. In all cases, the multi-wrapper model achieves median values closest to those of the individual wrapper models. The skewedness of the distributions is almost the same of the other wrapper models. In Figure 22, C5.0 classifier was considered for comparing the different feature selection methods. Clearly, the wrapper model based on C5.0 achieves the best results. However, this model fails to select feature subsets of such optimal performance when NBC is considered as in Figure 23. This shows that the individual wrapper model could be biased in their selection of the feature subset to the selected classifier. The multi-wrapper model is able to overcome this bias without significant loss in finding features that optimally enhance the considered classifiers.
Figure 14: Comparison of Different Feature Selection Methods based on KNN-3 Classifier

Figure 15: Comparison of Different Feature Selection Methods based on KNN-5 Classifier
Figure 16: Comparison of Different Feature Selection Methods based on SVM-Linear Classifier

Figure 17: Comparison of Different Feature Selection Methods based on SVM-Poly of Degree 3 Classifier
Figure 18: Comparison of Different Feature Selection Methods based on AdaBoost Classifier

Figure 19: Comparison of Different Feature Selection Methods based on Bagging Classifier
Figure 20: Comparison of Different Feature Selection Methods based on Mean Error Rate of C5.0, MSVM and NBC

Figure 21: Comparison of Different Feature Selection Methods based on Error Rate of MSVM
Figure 22: Comparison of Different Feature Selection Methods based on Error Rate of C5.0

Figure 23: Comparison of Different Feature Selection Methods based on Error Rate of NBC
9.2.4 Relevance Validation

An artificial dataset in supervised learning can be prepared by generating a set of features or variables and defining a target function that assigns the label for each instance based on these variables. The features that are part of the function defining the target label are the relevant ones and known to the experimenter in prior. These artificial datasets can, thus, be the basis for validating the ability of different classification methods to select those features describing the data the most. A simple relevance score is defined to measure the ability of each method in selecting those pre-defined relevant features. Given the number of the selected relevant features as \( n_{rel} \) out of all relevant ones with size \( a_{rel} \) and the number of irrelevant features \( n_{irrel} \), the relevance score \( r \) is defined as in Equation (16).

\[
    r = [1 + \left( \frac{n_{rel}}{a_{rel}} - \frac{n_{irrel}}{m} \right)]/2
\]  

(17)

Wrapper approach for feature selection results in selecting a subset of features biased to the learner. This subset, however, could be a relevant subset. C5.0 and NBC based wrappers achieve the best scores of relevance over the artificial datasets as illustrated in Figure 24. As a filtering approach, MRMR shows a comparable effect with the wrappers based on C5.0 and NBC. The feature selection process is, actually, embedded in the C5.0 based on gain ratio as a criterion. In NBC, also, the Bayes rule is a representative measure for evaluating relevant features. Thus, these two wrapper models, successfully, were able to detect the relevant subsets of features. The JMI criterion which generalizes well to different classification models fails in some cases to select the relevant subset of
features. For the multi-wrapper model, it achieves an acceptable overall relevance when compared to other methods.

Figure 24: Relevant Feature Selection Comparison of Different Feature Selection Methods based on Artificial Datasets

9.2.5 Comparison of C5.0 and C5.0 based on Wrapper Feature Selection Model

Since C5.0 could be viewed as an embedded feature selection method, we are interested in measuring its ability compared to the PGA wrapped version (C5.0W). Overall, the optimized wrapper version of C5.0 minimizes the classification testing error as highlighted in Figure 25. Only for three datasets which are Statlog, Semeion and CorrAL, the performance degrades. The Statlog and Semeion datasets are multi-class ones while CorrAL holds not well-balanced class distribution. For C5.0, the gain ratio probably, is more successful in cases of multi-class and ill-balanced class distribution.
than the wrapper model based solely on optimizing the error rate and number of features.

Regarding the generated tree size, the C5.0W almost always leads to a smaller decision tree as illustrated in Figure 26. In addition, the C5.0W is successful in achieving considerable time savings in the computational time over the C5.0 according to Figure 28.

Given the benchmark artificial datasets, Figure 27 shows that C5.0 is more capable of selecting the relevant features. This, also, could be the result of not explicitly including a relevance scoring function in the objective function in the case of C5.0W. For C5.0, the gain ratio is the main judging factor for selecting a feature.
Figure 25: Comparison of Misclassification Error Rate between C5.0 and C5.0 based on Wrapper Feature Selection

Figure 26: Comparison of Generated Tree Size between C5.0 and C5.0 based on Wrapper Model for Feature Selection
Figure 27: Comparison of Relevance Score using Benchmark Artificial Datasets between C5.0 and C5.0 based on Wrapper Feature Selection

Figure 28: Percentage of Progress in CPU Time for C5.0 with Wrapper Feature Selection over C5.0
CHAPTER 10

CONCLUSION

Feature selection is the first task of any learning approach to define a relevant set of features. Several methods are proposed to deal with the problem of feature selection including filter, wrapper and embedded methods. In this study, we focus on the wrapper method for feature subset selection to select a minimally sized subset of optimal features. Following an extensive empirical methodology and exploring variations of the objective function, we addressed various design issues related to the wrapper model.

Including the objective of minimizing the number of selected features gives an advantage for the wrapper model for selecting the features. The size of the selected features can be reduced by about 70% while preserving good performance. In many cases, the error rate was improved by 31.3% with dramatic savings in computational time. Nevertheless, the weight of this objective should be chosen carefully since large values would restrict the search space to small regions affecting proper discovery of the solution. We found that a weight value of 0.25 is superior to others in the examined set of experiments. In addition, given wrapper models based on different classification methods, the difference in the sizes of the selected feature subsets is insignificant if the same weight setting for the objective function is applied.

Modifying the objective function to consider minimizing the error rate of more than one classifier, was introduced for the first time in this work. This new formulation, multi-
wrapper model, can be viewed as an effective selection technique when several good feature subsets are present. The multi-wrapper model is not biased to a single classifier as the single-wrapper model for which the generalization ability is restricted.

There is no single feature selection method which could be regarded as a globally appropriate method for any kind of classifier and datasets of dissimilar characteristics. Moreover, feature selection may not always lead to better performance. Nevertheless, information theoretic based feature selection methods such as MRMR, JMI and C5.0 are powerful methods that can select relevant features that generalizes good given different classification methods.

Different extensions for the current work can be possible. We have examined the parallel genetic algorithm for searching for the optimal subset of features. Other search methods such as parallel particle swarm, parallel direct search and multi-objective genetic algorithms could be examined. It might be worthwhile to include some other objectives for the wrapper model and elaborate more on the relation between relevant and optimal features. Objectives that explicitly represent relevant feature and reflect characteristics of the problem (e.g. multi-class classifications’ problems and not well-balanced classes’ problems) may lead to selection of better feature subsets.
REFERENCES


