FEATURE SELECTION USING GENETIC ALGORITHM

Kanyanut Homsapaya

A Dissertation Submitted in Partial
Fulfillment of the Requirements for the Degree of
Doctor of Philosophy (Computer Science and Information Systems)
School of Applied Statistics
National Institute of Development Administration
2017

FEATURE SELECTION USING GENETIC ALGORITHM

Kanyanut Homsapaya

School of Applied Statistics

Associate Professor..Major Advisor

(Ohm Sornil, Ph.D.)

The Examining Committee Approved This Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy (Computer Science and Information Systems)

Associate Professor..Committee Chairperson

(Pipat Hiranvanichakorn, Ph.D.)

Associate Professor..Committee

(Ohm Sornil, Ph.D.)

Associate Professor..Committee

(Surapong Auwatanamongkol , Ph.D.)

Assistant Professor..Committee

(Tanasai Sucontphunt, Ph.D.)

Assistant Professor..Dean

(Pramote Luenam, Ph.D.)

January 2018

ABSTRACT

Title of Dissertation	Feature Selection Using Genetic Algorithm.
Author	Miss Kanyanut Homsapaya
Degree	Doctor of Philosophy (Computer Science and Information Systems)
Year	2017

In this dissertation, a method of feature selection in machine learning, and more particularly supervised learning is presented. Supervised learning is a machine learning task that infers answers from a training data set. In machine learning, training datasets are employed in order to create a model which enables reasonable predictions, while in supervised learning, each training example is a training set consisting of instances and labels, and the learning objective is to be able to predict the label of a new unseen instance with as few errors as possible. In recent years, many proposed learning algorithms that perform fairly well have been proposed. The factors to accomplish successful model building depend on many aspects such as noise and size of data. Most often for learning algorithms, it is assumed that training data is represented by a vector of numerical data for which each measurement is a feature, and an important question related to machine learning is how to represent instances using vectors of these to yield high learning performance.

Nowadays, data volumes are tremendously large in terms of aspects such as the number of features and most machine learning and data mining techniques may not be productive for high dimensional data, query accuracy and efficiency lessen swiftly as the dimension increases, the so-called curse of dimensionality. One of the requirements of good representation is conciseness since representation that uses too many features incurs major computational difficulties and may lead to poor prediction performance. Attribute selection is one of the significant methods in which the objective is to choose a small subset to predict the target sufficiently well. Feature selection selects the most importance features, eliminates irrelevant and redundant features from the entire set of attributes, reduces the computational complexity of any learning and prediction algorithm used in the process, and reduces cost by excluding unselected features.

A floating search is commonly used for the searching process. They are heuristic search methods which dynamically change the number of attributes included or eliminated at each step; they have produced very good results. The principal improvement of this thesis is focused on filter-based feature selection using genetic algorithm technique. Filters are normally less computationally intensive than wrapper method because wrappers apply a predictive model to score feature subsets. This approach is selected to be fast to reckon, whereas rooted to spot apprehending the goodness of the feature subsets. GA method can help to gain more diversity of population and provides us a way of reducing search space. Moreover, the contributions related to improves the contemporary sequential forward floating selection algorithm. In this thesis, an improving feature step using genetic algorithm is proposed as an additional step in a floating search. The objective is to eliminate weak features and replace a predominant one at each sequential step. From the research observations, the proposed method was discovered to be beneficial in selecting features that can boost the accuracy of data classification. Moreover, the experimental outcomes show that the proposed method with the genetic algorithm enhanced classification correctness and cut down data dimensionality for supervised learning problems.

ACKNOWLEDGEMENT

I would like to thank all the people who contributed in some way to the work described in this thesis. Principally, I would like to express my appreciation to my academic advisor, Professor Dr. Ohm Sornil. You have been tremensdous mentor me. During my responsibility, he supported to a graduate school knowledge by contributing me intellectual flexibility in my work, supporting my attendance, engaging me in new ideas, providing me with many good opportunities to accomplish my work and requiring a strong condition of work in all my effort. Furthermore, I would like to thank my committee members Professor Surapong for their interest in my work.

I would like to extend special thanks to my mother, grandmother, and grandfather for their constant love and support. Words cannot express how appreciate I am for all of the sacrifices that you have made on my behalf. Your prayers for me are what has sustained me thus far. I would also like to thank my beloved friend. He has benevolent given more to me than I could ever have expected. I cannot explain how fortunate I am to have him in my life. Thank you for supporting me through everything, and especially, I can never thank you enough for encouraging me to fulfill my dream.

Kanyanut Homsapaya
December 2017

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figures **Page**

LIST OF TABLES

Tables **Page**

SYMBOLS AND ABBREVIATIONS

Symbols

J	Criterion function
S	Original feature set
cand S	Candidate set
d	Predefined number of selected feature
D	Total number of original features
sel $_{d}$	Number of selected features in current set
cand $_{d}$	Number of features in a candidate set
sel S	Selected feature subset
S^{+}_{cand}	Candidate set in the inclusion step
S^{-}_{cand}	Candidate set in the exclusion step

Abbreviations	**Equivalence**
BAVE	Bhattacharyya Distance
CART	Classification and Regression Tree
CV	Cross Validation
CMI	Conditional Mutual Information
E	Entropy
GA	Genetic Algorithm
JMBH	Jeffery-Matusita Distance Bound to Bayes Error
KNN	K Nearest Neighbor
MAHA	Mahalanobis Distance

MI	Mutual Information
SFFS	Sequential Forward Floating Search
SFS	Sequential Forward Search
SBS	Sequential Forward Search

CHAPTER 1

INTRODUCTION

With the rapid growth rate of data collection becomes larger in both dimension (Number of Features) and volume. Right now, there is obtainability of data hundreds of features leading to data with very high element. A lot of data collection is lack of important information such as irrelevant, noise, ambiguity and redundant to the target concept. These can lead to misinterpret to machine learning results, especially when there are more unrelated attributes than relevant features. It may lead to insufficient and inaccurate performance of data mining model. There are many advantages of attribute selection methods such as decreasing calculation time, enhancing prediction effectiveness, and a superior interpreting of the data in machine learning and pattern recognition applications. In this dissertation, we concentrate on the filtering feature selection of attributes to boost classification accuracy the learning process, enhance the model generalization capability, and reduce the problem of the curse of dimensionality.

1.1 Classification

Classification is arguably the most important task in data mining and offers the capability to process with huge volume of data. It can be employed to forecast group of class label based and categorizes data found on training data set and class labels. The algorithm attempt to reveal relationships of attributes which have possibility to forecast the outcome. The classification problem is the problem that for many real-world objects and systems. To determine if an object is a member of a set or not, or which of several sets) is a hard problem. For example, try to discover a definition of chair. An object that meets the condition of chair, this mean it is a chair but fails to meet the definition that is not chair.

Classification has two step processes. The first step is to employ classification algorithm on training data set and second step is tested against a predefined test dataset to appraise the model. Therefore, classification model is the method to define or categorize class label from dataset whose class label are conceal. A number of methods are commonly used for data classification containing decision trees; rule-based, probabilistic and instance-based methods; support vector machines and neural networks. There are two main obstacles to data mining: Noisy and irrelevant problem. Most notably, it adversely affects system effectiveness in terms of classification accuracy, execution time, size of feature subsets, and understandability of the model obtained (Wu and Zhu, 2008; Sáez, Galar, Luengo and Herrera, 2013) because these issues are likely to present new characteristics in the problem area. For instance, noise often leads to small example clusters in a specific class in domain areas belonging to another class or can cause data in examples located in key areas within a specific class to be missed (Sáez et al., 2013).

The principle purpose of attribute method is to choose subsets of highly appropriate dimensions by eliminating irrelevant and excessive features. It is critical first step in classification, especially when applied to a large data set. Variable selection can significantly improve the computation time of a machine learning algorithm as well as enhance the model performance.

Selecting features relevant to the problem is a critical first step in classification, especially when applied to a large dataset. The aim is to select a representative subset of highly relevant dimensions while removing irrelevant and redundant one (Dash and Liu, 1997). Attribute selection can considerably boost the running time of a machine learning algorithm as well as improve the quality of the model.

Consequently, Bins and Draper (2001) proposed a method to decrease a original size of attributes, from 1,000 to a much smaller subset, without removing any highly important features or decreasing classification accuracy. There are three steps in the algorithm: first, irrelevant features are removed using a modified form of the Relief algorithm (Kira and Rendell, 1992); second, redundant features are eliminated using K-means clustering (MacQueen, 1967), and lastly a combinatorial feature selection algorithm is employed to the current feature subsets using the Sequential Floating Backward Selection (SFBS) algorithm. The basic concept is to filter feature subsets on each step until only the smallest one is obtained.

A floating search method dynamically expands and diminishes the number of features until the desired target is accomplished. Instead of fixing the number of forward/backward steps, we can allow values to float so that they can be flexibly adjusted without the requirement of setting parameters, which is different from Plus-l-Minus-r method. Nonetheless, a floating search has the tendency to become struck at a local optimum solution since there is almost no chance to improve the solution's quality (Somol, Novovičová and Pudil, 2006). For this reason, we present a more complicated version of the floating search algorithm with the aim of removing some of its potential drawbacks and to aid finding a solution closer to the optimal one.

1.2 Feature Selection

With the emergence of extremely large-scale data and the consequential necessity for favorable machine learning method, new problems continually surface requiring ever evolving approaches to feature selection. Therefore, the time and spaces required for processing the data increase. To ameliorate the problem of the dimensionality, techniques to reduce them are constantly sought, which has become increasingly important to the fields of machine learning and data mining research. Practically, attribute selection is a commonly applied method to deduce dimensionality, the aim being to select a small subset of pertinent features from the original ones by applying certain evaluation criteria. This often accomplishes improved classification accuracy, better learning performance and model interpretability, and lower computational cost. Attribute selection or dimensionality reduction plays a crucial rule to solve these problems. The principal drawback of feature selection is the possibility of information loss. Useful information can be discarded if dimensionality reduction is done poorly. We can state that variable selection is an algorithm to choose the most significant variables and discard the least significant variables to reduce evaluation time and sometimes improve effectiveness while minimizing the information loss.

The three main variable selection procedures are filter, wrapper, and hybrid . Wrapper methods rely on a classification algorithm employed as the subset evaluation process of feature subsets (Guyon and Elisseeff, 2003). Filter approaches use an independent criterion to evaluate the data using general characteristics and then selects feature subsets without applying a classification algorithm. Common evaluation functions usually are measures such as distance, mutual information (MI), dependency or entropy, calculated directly from the training data. A filter-based technique in a cascade fashion with a genetic algorithm (GA) has been developed using a correlation-based criterion (Karegowda, Jayaram and Manjunath, 2011).

Typically, Feature selection process in general is shown in Figure 1.1 with two important components: subset generation and subset evaluation. In the initial step, a candidate feature subset is selected depend on the search strategy of interest. In the second step, the subset is calculated according to predetermined evaluation criteria; the one that fits best is chosen from all of the candidates after the terminating criterion has been found. In the last step, the chosen subset is confirmed using either domain information or a validation set.

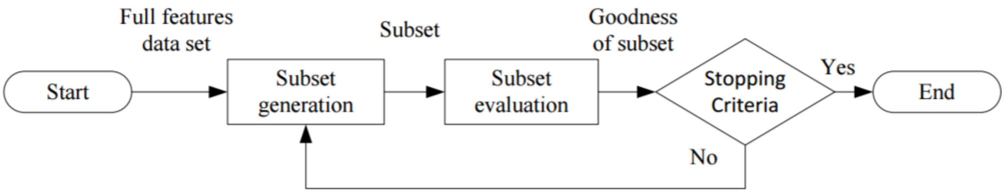


Figure 1.1 A Generalized Feature Selection Procedure

Search approach can be subsequent search, random search or complete search. Sequential search strategy will increase or discard one attribute at a time until terminating condition is found. This is a hill climbing strategy to generate selected subset. Random search strategy randomly selects feature subset and then perform sequential search. Another way of this strategy is to totally select feature subset randomly to evaluate.

Examples of sequential searching are sequential forward selection (SFS) and sequential backward selection (SBS), and their generalized versions GSFS and GSBS, which belong to the group of greedy algorithms, are most broadly utilized because of their general easiness and short running time.

The SFS method operates in a forward search manner starting with a blank set, then adds one attribute subset during each round until a new feature subset that maximizes the criterion function value is found, whereas the SBS method initiates with an original attribute subset and eliminates an attribute on each iteration until a predetermined criterion is satisfied. A drawback of both methods is that they have a nesting effect problem, which means that the features discarded are not eligible for reselection and the removal of selected features later on is not permitted.

1.3 Thesis Structure

In this thesis, data mining techniques to address the research objectives stated previously are presented. In Chapter 2, a literature review on mining data consisting of a number of stages is addressed: 1) supervised and unsupervised learning, 2) feature selection, 3) search methodology, 4) cross validation, 5) genetic algorithms, 6) niching methodology, 7) discretization, 8) measurements, and 9) classifiers. In Chapter 3, a new methodology for selection of features hinged on the use of mutual information is developed. In this chapter, the principles for floating searches and genetic algorithms are discussed. This chapter also covers feature selection approaches and the discretization process, and the results of data mining using three classifiers are shown and discussed in terms of how discretization, genetic algorithms, and feature selection affect classification. In Chapter 4, a new methodology for the selection of attributes rest on the use of niching technique is developed and a discussion of the results on the benefits of this new method is presented. Chapter 5 concludes the thesis with a summary of the main beneficial of the study and some advice for future work are provided.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Classification is the function whereby the correct class label for a given input is selected. In fundamental classification tasks, a set of labels is provided in advance and each input is considered independently from all of the others, after which a label is assigned to it. Some examples are

1) Classifying whether an email is spam or not.

2) Classifying the subject of a news article (e.g. sport, politics, etc.).

3) Classifying the meaning of each occurrence of a particular word (e.g. *bank* can refer to a financial institution, the act of depositing something in a financial institution, the act of tilting to the side, or a river bank).

The main task in feature selection classification is to discover the optimal feature subset from the initial attribute set that ameliorates the efficiency in generating the classification model, and thus enhances classification performance. The large number of high-dimensional data that occurs and is publically available on the online system has vastly boosted in the past recent years. Thus, machine learning approaches have obstacles in dealing with the huge amount of input attributes, which is posing an interesting challenge for scholars. In order to apply machine learning approaches completely, preprocessing of the data is indispensable.

Feature selection, also known as variable selection, attribute selection, or variable subset selection in machine learning and statistics (Kumar and Minz, 2014), is one of the most prevalent and significant techniques in data preprocessing, and has develop into a necessary fundamental of the machine learning process. It is the procedure of discovering useful features and discarding impertinent, inessential, or noisy data. This process deceases cost-timing of data mining algorithms, obtains a considerable improvement of classification accuracy, and enhances understandability. Unrelated features are features that cannot distinguish samples into classes or clusters as they are not significant with respect to the class concept. Redundant features demonstrate features that co-present with other features. This obstacle may lead to poor performance in many situations. In recent times, in-depth research into feature selection methods has been accomplished by scholars from a number of multidisciplinary fields including pattern recognition, data mining, machine learning, and statistics. In terms of the interaction between feature selection and the respective classification model, many of these have been categorized as filter, wrapper, or hybrid methods.

2.2 Supervised and Unsupervised Learning

2.2.1 Supervised Learning

Supervised learning is frequently yield to solve classification problems because the objective is often to analyze data in order to determine a target variable of future data. Therefore, if you are training your machine learning task for every input with corresponding target, it is named supervised learning, which will be able to contribute target for any new input after adequate data training. It is named “supervised” on account of in the training task of the learning procedure the algorithm has connect to the basic fact by using certain inputs and output prediction. If the output target can be a class label, it is called classification problem. Otherwise, if the output target is continuous or real number, it is called regression problem.

A training dataset consisting of example data is initially used to fit the classification model. In supervised learning, each example comprises an input record and its accompanying decision output value. A supervised learning algorithm examines and determines the training data and constructs a training algorithm, which is an inferred function used to assess new outputs. On most occasions, the basic scenario enables the algorithm to analyze and then accurately resolve the class labels for invisible instances. This step enables the learning algorithm to assess unseen situations in a suitable way by generalizing based on the training data.

Table 2.1 Example of Training Data

	Inputs					Output
	Gender	**Married**	**Job**	**Age**	**Salary**	**Trust**
Customer1	Male	No	Teacher	45	300,000	Good
Customer2	Female	Yes	Lawyer	23	200,000	Bad
Customer3	Male	Yes	Doctor	40	350,000	Good
Customer4	Female	No	Programmer	30	140,000	Bad
…	…	…	…	…	…	…
Customer n	Male	No	Doctor	30	180,000	????

2.2.2 Unsupervised Learning

Unsupervised learning is the task in classification that you don’t have any kind of target outputs to discover. It is related solely with features themselves but we do not have an associated response. The purpose is to explore patterns in the data or try to divide the data into groups or specific clusters. Commonly, an unsupervised learning the machine simply receives input feature data sets from its environment without supervised target outputs or rewards.

Unsupervised learning techniques are usually inspired by the fact that the limitation of time and financial to create "label" feature data, which would grant it to considered as employing supervised techniques. The other inspiration is owing to the fact that pictures, video, natural language documents and scientific research data (such as gene expressions), once quantified, obtains extremely large dimensionality and generated low results of accuracy rate. Even though it is difficult to imagine how a machine can possible learn to build a working model, creating a legal groundwork for unsupervised learning is nevertheless possible depended on the concept that the machine’s aim is to assemble representations of the input to be used for determination making, forecasting future inputs, and productively communicating said inputs to another machine.

2.3 Feature Selection

There are two main approaches of Dimensionality Reduction, namely feature extraction and feature selection. Feature extraction is the procedure whereby a set of different features is obtained from the full features sets using a mapping function, with the objective of demonstrating the original data as succinctly as possible. However, the computational cost desired to search for an acceptable mapping function and the loss of comprehensibility in the outcome are major disadvantages. Nevertheless, no new features are generated and a most favorable set of the initial features are chosen in accordance with satisfied condition.

There are four significant benefits of feature selection (Navot, 2006). First, the principle objective of this selection process is to deduce the calculation time and complexity of the learning algorithms by decreasing the element of the feature space. Second, the identity of the selected features can contribute insight into the nature of the problem at hand, and attribute selection lessens the cost of evaluating unselected features. Because we have found a small set of features that yield an efficient prediction score, it is no longer necessary to measure the rest of the features. Thus, only a few features in each instance need to be assessed in the prediction stage. Third, feature selection can also improve prediction accuracy because after the discovery of only a tiny set of good features, even very simple learning algorithms are able to perform well. Hence, feature selection is an essential step toward efficient learning when dealing with large multi-featured datasets. On a more general level, feature selection research has clearly become crucial to solving the fundamental issue of data representation. Lastly, feature selection gives a more detailed understanding of the problem at hand because it attains the most informative features.

Feature selection approaches are broadly classified into wrapper, filter, and hybrid techniques. In wrapper methods, variable selection involves applying a wrapper around a particular learning algorithm to assess the fitness of the attribute subsets. In filter methods, the attribute selection method removes unrelated and/or unnecessary attributes in a preliminary processing data step applied before any particular learning algorithm (Sanchez-Marono, Alonso-Betanzos and Castillo, 2005).

The filter approaches are in common computationally more effective, even though wrapper techniques frequently return to higher quality.

In the mechanism of attribute selection, unrelated and/or repetitious features (or other noise sources in the data) may be impede in many circumstances, as they are not pertinent and significant with respect to the class concept such as microarray data analysis. When the number of samples is much less than the features, then machine learning gets especially complication, because the search space will be sparsely populated. Thus, the model will not able to discriminate precisely between noise and appropriate data. There are two fundamental methods to attribute selection. The first is Individual Generation, and the second is Assessment of Feature Subset. Ranking of the features is recognized as Individual Assessment. In Individual Assessment, the score of an individual feature is reckoned in agreement with its degree of relevance. In Subset Evaluation, candidate attribute subsets are composed using search approach. The common process for attribute selection has four significant components as shown in Figure

1) Subset Generation
2) Assessment of Feature Subset
3) Terminating Condition
4) Outcome Validation

2.4 Feature Selection for Classification

In supervised learning, an optimum scenario will grant the method to precisely conclude the class labels for conceal instances; whereas, in unsupervised learning, this information is missing. The target of this scenario is thus to identify the natural grouping structure of the data. In semi supervised learning, only some of the data objects are labeled. It uses both labeled and unlabeled to train the model. Many researchers in the field of machine learning have discovered that unlabeled data used together with a small amount of labeled data is able to improve learning accuracy considerably.

Significantly, real-world classification difficulties desire supervised learning to cope with situations where the underlying class probabilities are unidentified and each instance is involved with a class label. Class label of training data is provided as a guideline to generate model. The information helps verify if the prediction is correct or not. The supervised learning algorithm examines the training data and generates a rigid rules-based model, which can be used to map new examples. In real-world situations, we often have little knowledge about relevant features. Thus, many candidate features are proposed to represent the domain more clearly, some of which are unimportant and/or excessive to the outcome. A relevant feature is neither unimportant nor redundant or else is not precisely associated with the outcome but affects the learning process. Simply put, an excessive feature does not add anything new. In many classification problems, it is troublesome selecting good classifiers before discarding undesired attributes owing to the gigantic volume of the data. Decreasing the number of insignificant and/or excessive features yields a more common but effective classifier as well as drastically decreasing the execution time of a learning algorf5ithm, which enables better insight into solving real-world classification problems.

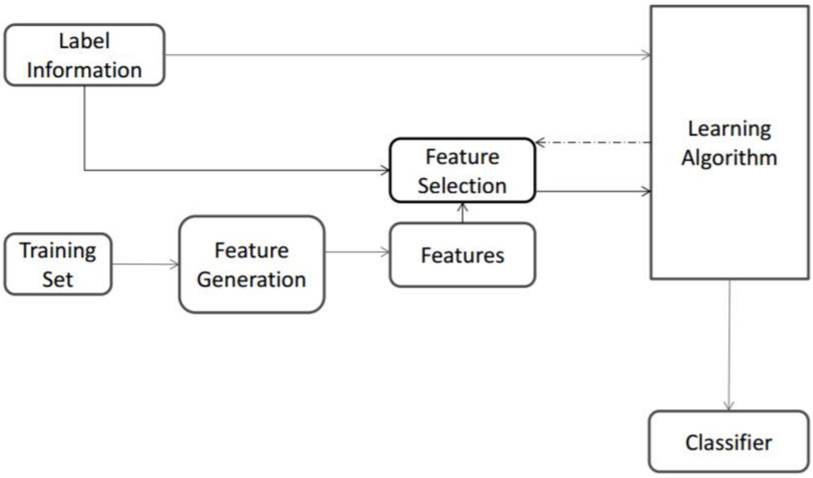


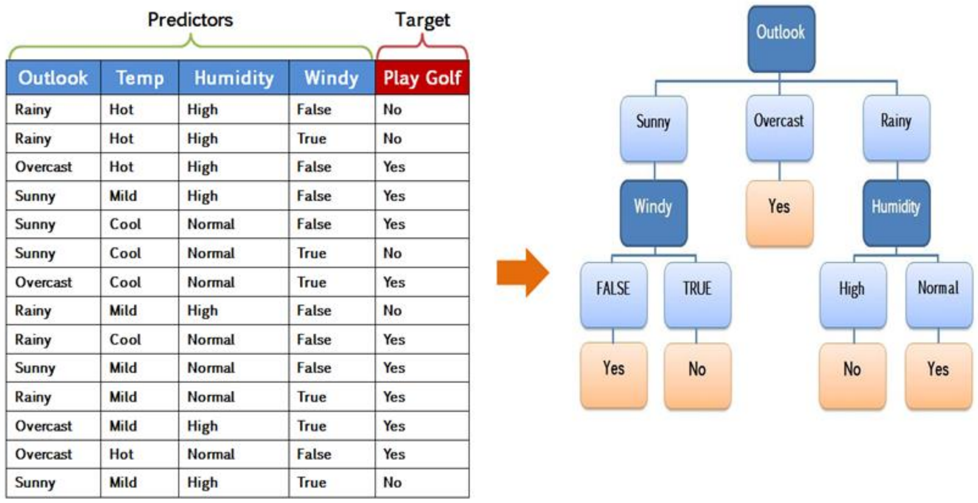
Figure 2.1 A General Framework of Feature Selection for Classification

Source: Tang, Alelyani and Liu, 2014: 37.

2.5 Decision Tree Classification

2.5.1 How a Decision Tree Works?

Suppose we would like to classify mammal or a non-mammal animal or classify customers of bank who are safe or risky to lend money (Kavitha, Kangaiammal and Satheesh, 2015). How can we tell whether it is a mammal or a non-mammal and safe or risky customer? One approach is to pose is d a series of questions about characteristics of animal type or customer category. The first question we may ask is whether the species is cold or warm blooded or how old of customer. If it is cold blooded, it is not a mammal. In the latter case, if customers who are age more than 45, it seems to have more chance to lose dept. Each time we receive the answer, a follow-up question is asked until we can reach a conclusion about class label of the record. Normally, a series of question and their potential answers can be organized in the format of decision tree which is a hierarchical structure consists of nodes and edges. Decision trees are formed on regression models or classification used to create a tree structure that can handle both categorical and numerical data. In this process, a dataset is reduced to smaller and smaller subsets while simultaneously building the associated decision tree incrementally, the outcome of which is a tree consisting of decision nodes and leaf nodes. The topmost decision node (the root node) in a tree corresponds to the best predictor.



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Figure 2.2 Example of Decision Tree
Source: Saedsayad, 2017.

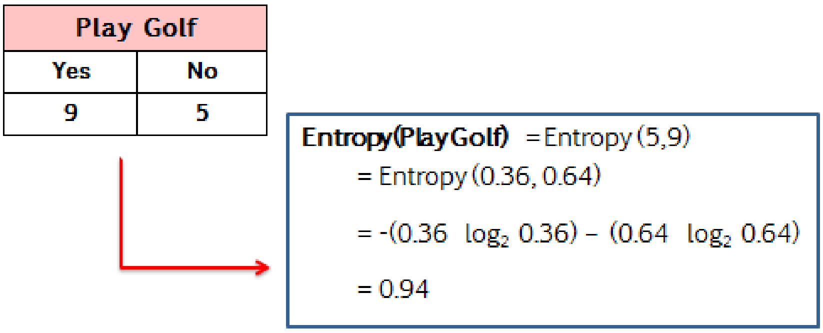
In Figure 2.2, a decision node (e.g. Outlook) has two or more branches (e.g., Sunny, Overcast, and Rainy), and a leaf node (e.g., Play) demonstrates a classification or decision based on the outcomes of the decision nodes and branches.

2.5.2 The ID3 Algorithm

This is an important algorithm used to construct a decision tree from a dataset (Quinlan, 1986). It employs entropy and information gain properties to create a decision tree and employs top down greedy search without backtracking, although that can be utilized if necessary.

To build a decision tree, the calculation of two types of entropy using frequency tables is required. Using the data in Figure 2.2, this can be accomplished as follows:

2.5.2.1 Entropy Using a Frequency Table with One Attribute:



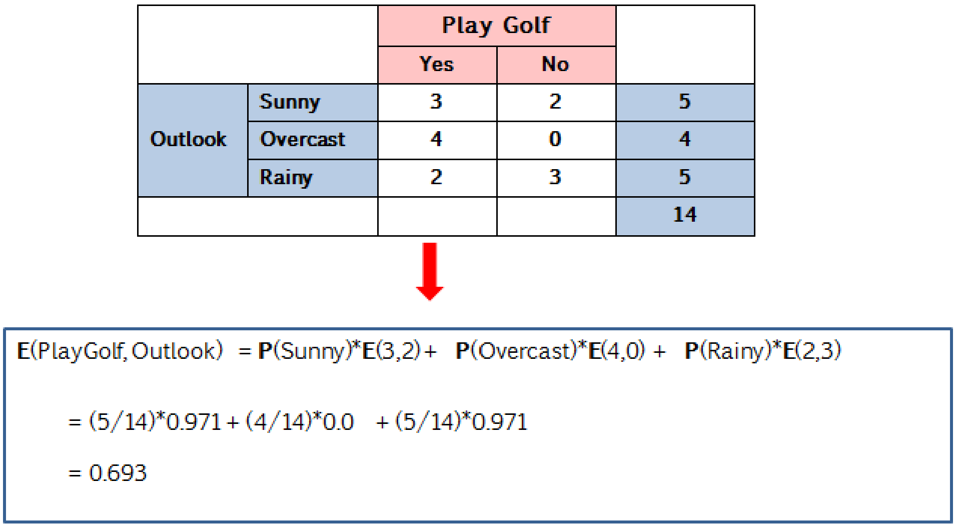
Play Golf	
Yes	No
9	5

$$\begin{aligned} \text{Entropy(PlayGolf)} &= \text{Entropy}(5,9) \\ &= \text{Entropy}(0.36, 0.64) \\ &= -(0.36 \log_2 0.36) - (0.64 \log_2 0.64) \\ &= 0.94 \end{aligned}$$

Figure 2.3 Frequency Table of One Feature

Source: Saedsayad, 2017.

2.5.2.2 Entropy Using a Frequency Table with Two Attributes:



		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

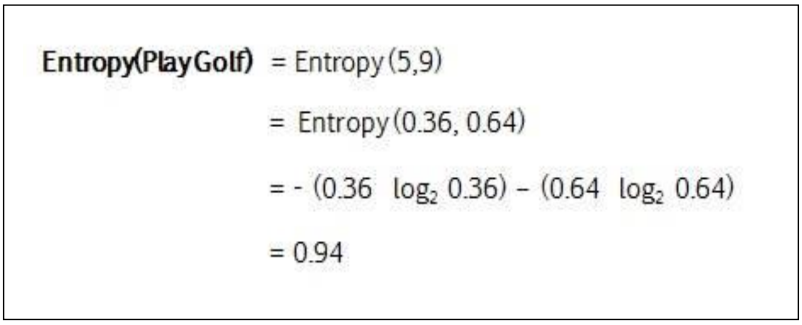
$$\begin{aligned} \text{E(PlayGolf, Outlook)} &= \text{P(Sunny)*E(3,2)} + \text{P(Overcast)*E(4,0)} + \text{P(Rainy)*E(2,3)} \\ &= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971 \\ &= 0.693 \end{aligned}$$

Figure 2.4 Frequency Table of Two Features

Source: Saedsayad, 2017.

2.5.2.3 Information gain is a measure of the difference in entropy before and after a dataset is split on an attribute. One that retrieves the maximal information gain (the biggest reduction in uncertainty) is used in the construction of the decision tree, e.g.

1) Calculate the entropy of the target.

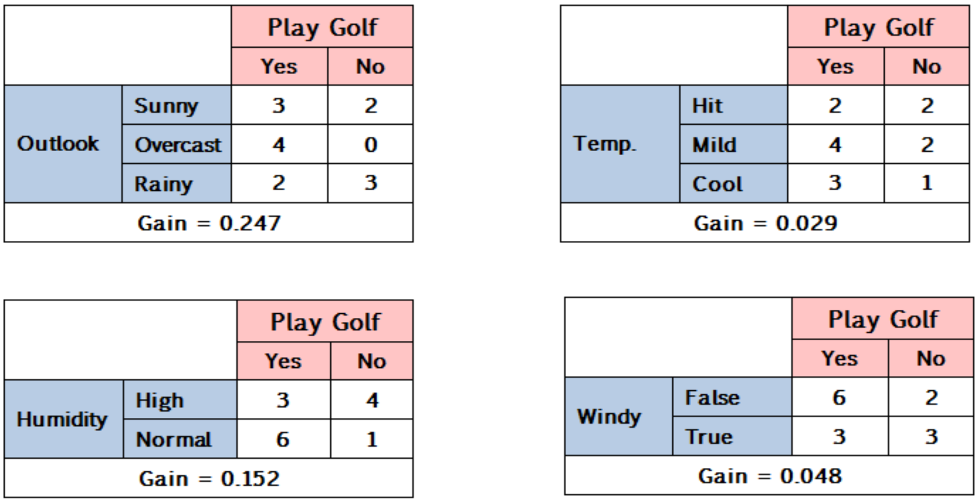


$$\begin{aligned} \text{Entropy(PlayGolf)} &= \text{Entropy}(5,9) \\ &= \text{Entropy}(0.36, 0.64) \\ &= -(0.36 \ \log_2 0.36) - (0.64 \ \log_2 0.64) \\ &= 0.94 \end{aligned}$$

Figure 2.5 Entropy of PlayGolf

Source: Saedsayad, 2017.

2) Subsequently, the dataset is split on the attribute and each branch of entropy is measured. The outcome entropy is subtracted from the entropy before the split, thus the result of information gain is deduced.



		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hit	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

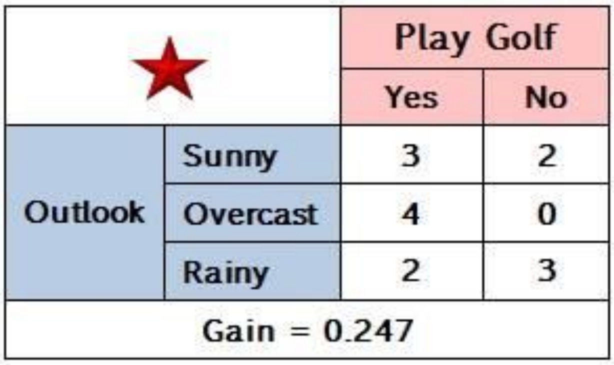
		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Figure 2.6 Information Gain

Source: Saedsayad, 2017.

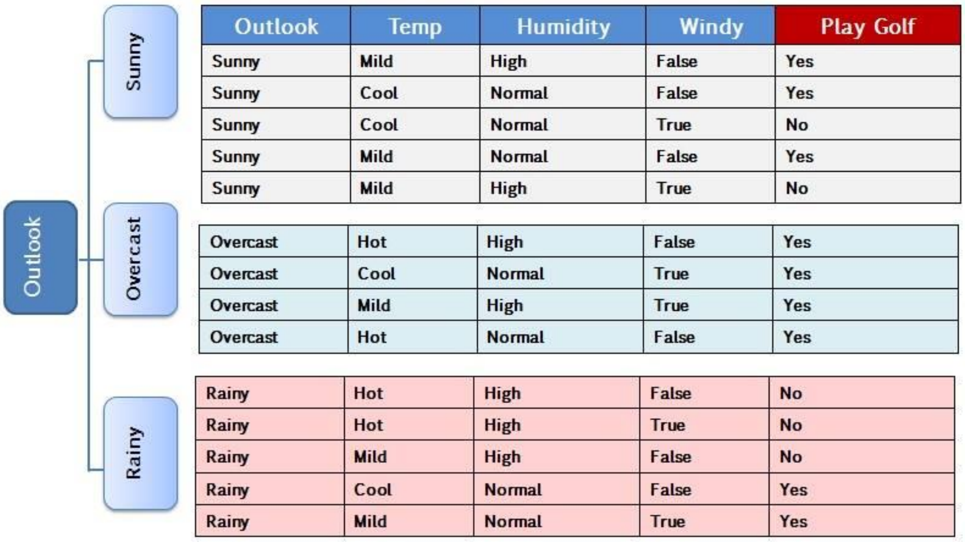
3) Selecting the highest information gain as the decision node and the data set is divided using this branch. A branch with zero entropy is a leaf node and one with non-zero entropy needs to be split further.



★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

Figure 2.7 Root Node

Source: Saedsayad, 2017.



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Sunny	Mild	Normal	False	Yes
Sunny	Mild	High	True	No

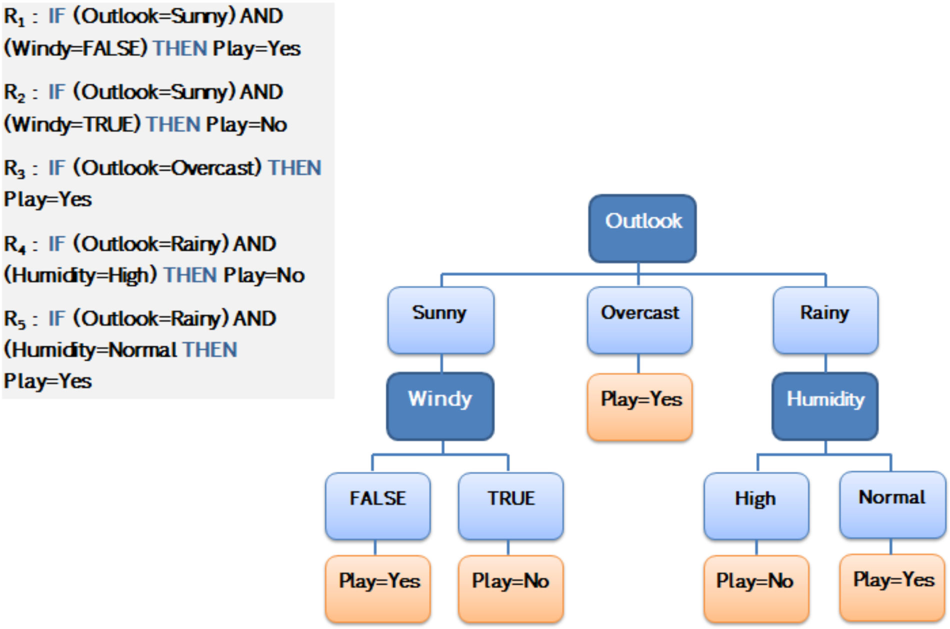
Overcast	Hot	High	False	Yes
Overcast	Cool	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Rainy	Mild	Normal	True	Yes

Figure 2.8 Decision Tree

Source: Saedsayad, 2017.

4) The ID3 algorithm is processed repeatedly on the remaining non-leaf branches until all of the data has been categorized.



R_1 : IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2 : IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3 : IF (Outlook=Overcast) THEN Play=Yes

R_4 : IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5 : IF (Outlook=Rainy) AND (Humidity=Normal THEN Play=Yes

Figure 2.9 Decision Tree to Decision Rules

Source: Saedsayad, 2017.

Typically, a decision tree can effortlessly be adapted to a set of rules or commands with pattern of if then clauses by mapping from the root node to the leaf nodes one by one.

2.6 Feature Selection Models

2.6.1 Filter Methods

Filter Methods, as its name imply, are algorithms which exude out insignificant attributes which have little value to analysis data. Normally, filter methods are not considered about classifiers. Thus, these methods are less computed time consuming than wrapper and hybrid methods.

As described before, filter methods employ to variable selection as a preliminary processing method with no induction procedure. The important part of filter methods search is to consider the traits of each feature subset using evaluation measures such as information gain and entropy described earlier, or the distance between a class and a statistical dependence test. This model trends to run faster than the wrapper approach.

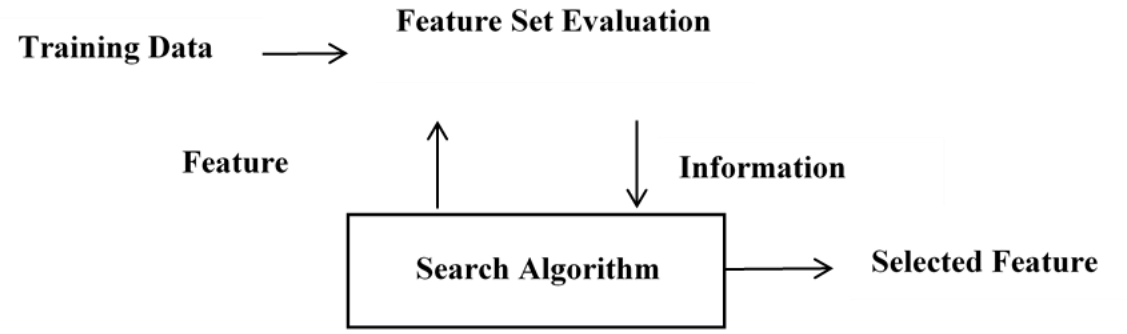


Figure 2.10 Filter Algorithm
Source: Juan Tapia Farias, 2017.

As shown in Figure 2.3, a filter algorithm begins a search from a feature subset and a blank set before searching through the feature space using the consecutive processes:

1) Evaluate the current feature subset X using measurement methods such as mutual information or normalized mutual information.

2) Compare it with the outstanding attribute subset acquired in the previous step (Xk1) and if better, assign the new one as the current best subset.

3) The search procedure is iterated until a pre-defined stop condition d is accomplished. The criterion d could be one or more of following: (1) subsequent addition or deletion of a feature does not generate a more appropriate feature subset, (2) the performance obligation is accomplished, or (3) a predetermined bound (stopping criterion) is attained, such as the maximum number of search iterations or the minimum number of features. At the end of the process, the current efficient subset is retained.

2.6.2 Wrapper

The wrapper method (Sanchez et al., 2005) is a type of dependent criterion that incorporates the variables themselves into the feature evaluation process. To ascertain the importance of a candidate attribute subset, a classification model is constructed and utilized to evaluate the particular set. The algorithm outputs the last current best subset.

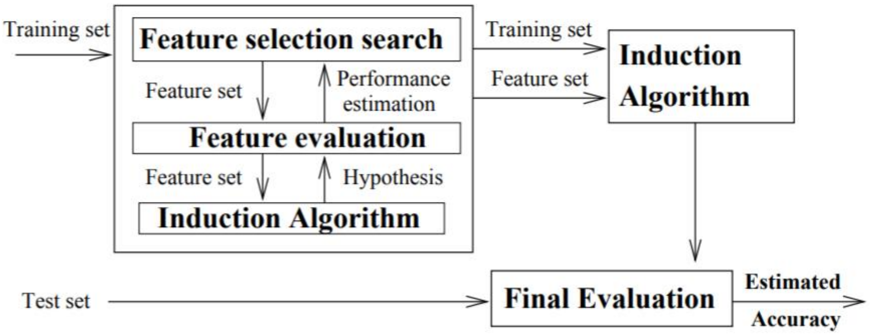


Figure 2.11 Wrapper Approach

Source: Kohavi and John, 1997: 273-324.

Figure 2.4, the wrapper method to attribute subset selection. The induction mechanism is managed as a “black box” by the subset selection method.

In wrapper approach, classification algorithm is employed as the evaluation function. The feature selection algorithm is existed the classification algorithm. For each subset a classifier is constructed and this classifier is used for evaluating that subset. The benefit of this approach is that it helps to increase reliability of the evaluation function. If evaluation algorithm and classification algorithms are different, different biases they have make the results less reliable. The disadvantage is that it increases the cost of the evaluation function.

Contrary to employing an independence test as in filter methods, the wrapper approach (Figure 1(b) employs a machine learning algorithm such as a decision tree or support vector machine and gauges the corresponding classification performance to direct the attribute selection process. The classification performance of X is approximated with the most efficient attribute subset previously attained. Subsequently, X_k = X if the classification efficiency is better than the former one. This procedure is repeated until a stop condition is satisfied, as explained previously in other filter methods.

2.6.3 Hybrid

To employ the benefits of filter and wrapper methods, the hybrid approach has been currently presented (Dash and Liu, 1997). A classic hybrid method applies both an independent test and a performance evaluation function of the attribute subset.

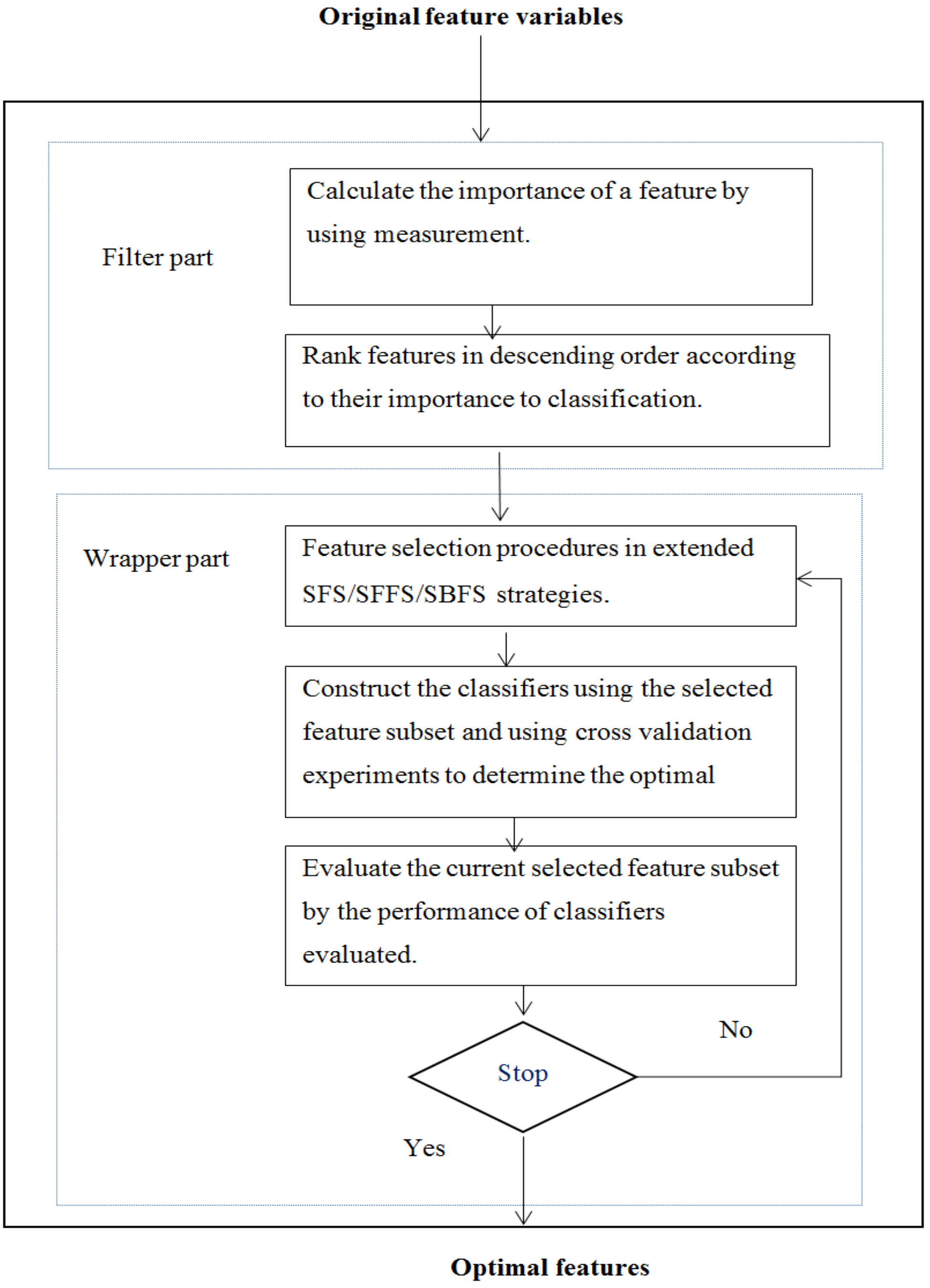

Original feature variables
Filter part
Calculate the importance of a feature by using measurement.
Rank features in descending order according to their importance to classification.
Wrapper part
Feature selection procedures in extended SFS/SFFS/SBFS strategies.
Construct the classifiers using the selected feature subset and using cross validation experiments to determine the optimal
Evaluate the current selected feature subset by the performance of classifiers evaluated.
No
Stop
Yes
Optimal features

Figure 2.12 The Hybrid Approach

Source: Lee, 2009: 10896-10904.

As shown in Figure 2.5, a hybrid search begins with a predetermined feature subset and combines filter and wrapper approaches to discover the finest subsets as cardinality increases. Because the filter approach comprises an independence test and an association criterion, it is initially used to select candidate features after which the wrapper approach reexamines them using a specific learning algorithm and another association criterion. After selecting the most efficient subset with cardinality k, the overall classification performance is evaluated with respect to a predetermined (stopping) criterion. If the performances match, the feature selection procedure has come to an end and the current best feature subset is outputted as the optimum attribute subset, else the searching continues after incrementing the cardinality to k+1 by increasing an attribute from the surviving ones and reiterating the previous steps. In improving the classification efficacy of the filter method by include a particular learning algorithm in the selection process and enhancing the effectiveness of the wrapper method by narrowing the exploration scope, the hybrid approach is an effort to overwhelm the respective weaknesses of the two.

2.7 Search Method

2.7.1 Sequential Feature Selection Algorithms

These comprise a group of greedy search methods employed to decrease the primary d-dimensional attribute space to a k-dimensional variable subspace where $k < d$. The incentive in seeking a suitable attribute selection algorithm is the automatic selection of a subset of attributes most relevant to the problem at hand.

2.7.2 Branch and Bound Algorithms

Branch and bound (BB) algorithms (Songyot Nakariyakul, 2009) are known to present the optimal solutions. Generally speaking, an exhaustive search is conducted to discover the efficient feature subset out of the original number of attributes by measuring a given criterion function for all attainable attribute subsets and then choosing the best feature subsets corresponding to the criterion function. Although an exhaustive investigation is suitable for low-dimensional data, it is impractical for a large-dimensional database because all the number of candidate attribute subsets that require to be exhaustively investigated can be huge. The BB algorithm explores the search space more efficiently than an exhaustive search. The basic concept is to generate the search tree. B&B algorithms branch and create two new nodes, therefore separating the solution space into a set of smaller subsets and attaining the relative upper and lower bound for each node. If the branch length at this search tree node is more than the current lower bound on the optimal tree length, this search tree path is aborted and later the search is backtracked and then continued to the next tree path. When the search tree reaches the magnify node, the tree is either optimal or sub-optimal rejected.

There are two objectives in a B&B search: finding the optimal solution and proving its optimality (He, Daume and Eisner, 2014). Normally, we have to trade-off between the two goals; we can find the optimum solution faster if we do not need to prove its optimum or reduced execution time. Consequently, we can find a potential solution without extensive proof of optimality, so the search time can be greatly reduced.

2.7.3 Sequential Search Algorithms

Sequential search algorithms including sequential forward selection (SFS) and sequential backward selection (SBS), and their generalized versions GSFS and GSBS, are greedy algorithms most widely used because of their general simplicity and short running time.

The SFS method operates in a forward search manner starting with an empty set and increases one variable subset during each round until a new attribute subset that maximizes the criterion function value is found, on the contrary, the SBS method begins with a full attribute subset and eliminates an attribute on each iteration until a predetermined criterion is satisfied. A drawback of both methods is that they have a nesting effect problem, which means that discarded features are not eligible for re-selection nor selected features for removal later. Since these algorithms do not analyze all candidate feature subsets, there is no promise of them yielding an optimum solution. Generalized forms GSFS and GSBS based on group collection feature testing discover exceptional solutions but at the cost of increased calculation time.

The plus l take away r (PTA) approach was presented to take care of the nesting problem (Zhang and Sun, 2002).

Sequential Forward Selection is the uncomplicated greedy search method. -Beginning from the empty set, continuously increase the attribute x^{+} that produces in the highest objective function $J(Y_{k+}x+)$ when incorporated with the attributes Y_k that have already been chosen .

Algorithm:

1) Begin with the blank data set $Y_0=\{\phi\}$
2) Pick the next best attribute $X^{+}=\text{argmax}[J(Y_{k+}X)];x\text{¢}Y_k$
3) Renew $Y_{k+1}=Y_{k+}\ X^{+}$; $k=k^{+1}$
4) Go to 2

SFS delivers the efficient performance when the optimum feature subset has the smallest possible number of attributes. When the search is close to an empty set, there is the potential to evaluate a large number of states. However, the domain analyzed by SFS is smaller for a full set since most of the attributes have already been determined. The search space can be visualized as an ellipse to give priority to the fact that there are fewer states toward complete or blank data sets. For instance, in the state space for four features, the number of states is largest in the middle of the search tree. Nevertheless, a main drawback of SFS is its inability to eliminate attributes that have develop into out-of-date due to the inclusion of other attributes

2.7.4 Sequential Backward Search

SBS is conducted the other way round to SFS. Commencing from a full set, feature x^{-} resulting in the smallest decline in the value of objective function $J(Y\text{-}x^{-})$ is sequentially eliminated. Note that feature elimination possibly leads to an enhance in objective function $J(Y_{k\text{-}x-)} > J(Y_k)$.

Algorithm:

1) Begin with the entire set $Y_0=X$
2) Eliminate the worst attribute $X^{-}=\text{argmax}[J(Y_k{}^{-}X)];x\ \ Y_k$
3) Renew $Y_{k+1}=Y_k{}^{-}\ X^{-}$; $k=k+1$
4) Go to 2

SBS manages optimally when the optimum variable subset contains a considerable number of attributes, since most of the time during its execution is spent visiting large subsets. The predominant drawback of SBS is the incompetence to reassesses a feature once it has been eliminated.

2.7.5 Plus-L Minus-R

Plus-L Minus-R is a generalization version of SFS and SBS. The objective is to prevent the limitation or weakness of re-selection or re-deletion feature subset. The value of L and R are presumed with constant value. If L>R, LRS initial with the blank set and continually reiteration increments ‘L’ features and eliminates ‘R’ features. The disadvantage is lack of regulation to define the optimal value of L and R.

Algorithm:

1) If LR then begin with the empty set Y={ϕ} else start with the complete set Y=X Go to step3.

2) Reiterate L times X^{+} =argmax[J(Y_{k+}X)];x¢Yk and Y_{k+1}=Y_{k}+ X^{+} ; k=k+1

3) Reiterate R times X^{-} =argmax[J(Y_{k-}X)];x Yk and Y_{k+1}=Y_{k}- X^{-} ; k=k+1

4) Go to 2

2.7.6 Sequential Forward Floating Search (SFFS)

Somol, Pudil, Novovičová, and Paclík (1999) and Pudil, Novovičová and Kittler (1994) proposed a sequential forward floating search (SFFS) algorithm by applying a criterion function to choose small feature subsets and compare them with candidate subsets. The aforementioned SFS and SBS algorithms can be extended to more complex floating variants SFFS and SFBS that have a further respective inclusion or exclusion step to discard (or increase) features once they have been contained (or removed) so that a larger number of attribute subset combinations can be analyzed. It must be focused that this task is conditional and only happens if the deriving attribute subset is evaluated as an improvement by the criterion function after respectively removing (or adding) a specific feature.

The SFFS and SBFS approaches were initially created to conquer the so-called ‘nesting effect’ difficulty of the simpler SFS and SBS algorithms cause by their respective inability to re-select a discarded feature or to discard a previously selected feature.

By far the most profitable method so far is the thorough floating search method reported by Somol et al. (1999) and Pudil et al. (1994). The floating search technique combines the ‘Sequential Forward Floating Search (SFFS)’ and the ‘Sequential Backward Floating Search (SBFS)’ based on two main categories: the search process in a forward direction. These methods use a criterion function to select a feature and compare candidate subsets. SFFS and SBFS can be classified as a wrapper or a filter approach depending on the criterion function used. They perform well but the computational time is long, especially with large datasets. The floating search methods can be considered the PTA algorithm without the use of a fixed parameter. They have been shown to give very good performance, close to optimum results, and to overcome the nesting problem. SFFS, SBFS, and bidirectional selection as a combination of both are greedy search methods that include or discard features one at a time. The floating search method comprises of two phases: forward and backward. SFFS begins with an empty set and sequentially add one attribute at a time. The structure of the floating search methodology is displayed in Figure 2.6

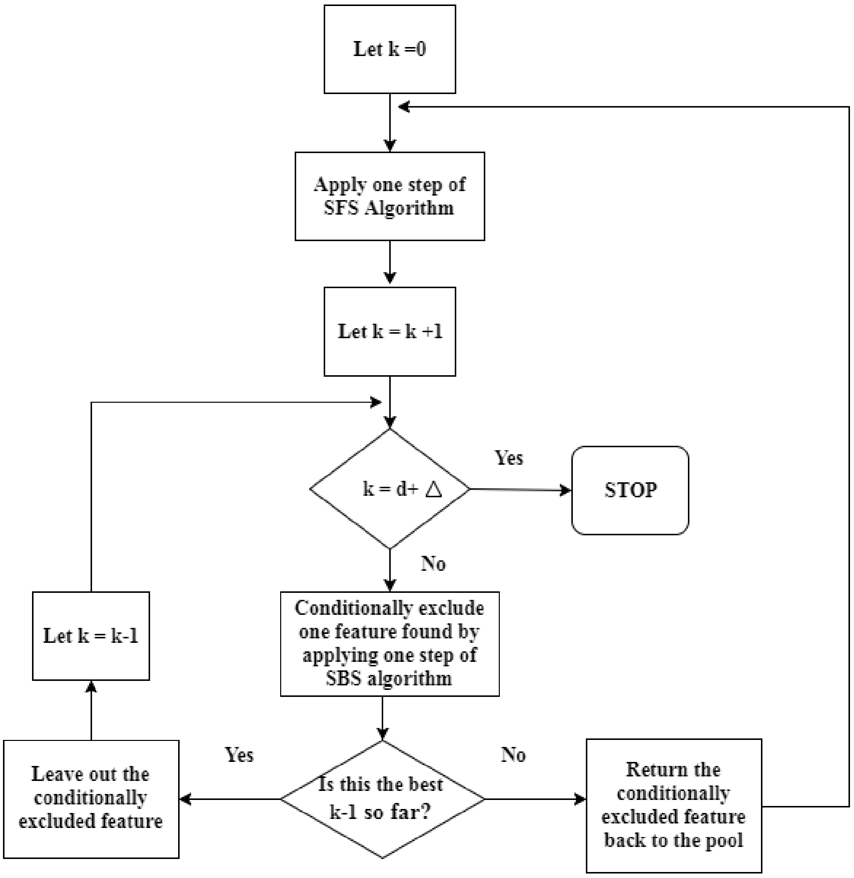

Let k =0
Apply one step of SFS Algorithm
Let k = k +1
k = d+ Δ
Yes
STOP
No
Conditionally exclude one feature found by applying one step of SBS algorithm
Let k = k-1
Is this the best k-1 so far?
Yes
No
Leave out the conditionally excluded feature
Return the conditionally excluded feature back to the pool

Figure 2.13 Structure of the Floating Search

Source: Chandrashekar and Sahin, 2014: 16-28.

SBFS, the counterpart of the forward search, is initialized with a complete set and sequentially eliminates one attribute at a time after execution of SFFS. An SFFS search selects the best unselected feature according to a criterion function to form a new feature subset, and an SBFS search iteratively determines which members of the selected subset are to be removed if the remaining set improves performance according to the same criterion function in the forward search. The algorithm loops back to a forward search until the stopping condition is reached The stopping condition k= d+ Δ determines whether the search algorithm can be allowed to continue on to the original dimensionality D, with d being the number of feature subsets containing desired values. When D is a very large dimension, the value of Δ needs to be decided upon carefully. There are disadvantages when using either algorithm. With SFFS, it is not possible to succeed in eliminating repetitious attributes produced in the search method, whereas SBFS cannot re-calculate evaluation feature efficiency together with other attributes at the same time.

2.7.7 Improved Forward Floating Selection

The Improved Forward Floating Selection (IFFS) projected by Songyot Nakariyakul and Casasent (2008) contains an additional step to determine whether replacing a weak feature will improve the criterion function value. Improved versions of SFFS have been proposed in many researches to obtain better performance. A new version of IFFS method for choosing a small subset of attributes is demonstrated. The scholars improved on SFFS by adding a exploration stage called “replacing the weak feature” to determine whether eliminating any of the features in the currently selected feature subset and adding a new one in each subsequent step will enhance the current feature subset. The extra step is performed after the removing step. This step conditionally eliminates one feature at a time and employs the SFS approach to select an unselected feature and add it to each resultant feature set. Their finding indicate that this approach provided the optimum solution (or very nearly to it) for many chosen subsets better than had previously been demonstrated by other suboptimal feature selection methods.

2.7.8 Adaptive Sequential Forward Floating Selection

Somol et al. (1999) presented the Adaptive Sequential Forward Floating Selection (ASFFS) algorithm with a parameter *r* which specifies the number of attributes to be included in the forward phase calculated dynamically. Parameter *o* is used in the exclusion phase to remove the maximum number of features if it improves performance. The benefit of ASFFS is in providing a less redundant subset than the SFFS algorithm.

Finally, Jitwadee Chaiyakarn (2013) propose a filter-based method to return a small subset of features for classification problems by employing two different criterion functions in the forward and backward steps. The functions help remove redundant features, maximize inter-class distances, and minimize intra-class distances.

All searching methods require an assessment proof to evaluate the quality of each feature before addition to or removal from the current set. With this in mind, several evaluation criterions involving distance, information and dependency measures have been proposed (Molina, Belanche and Nebot, 2002; Chotirat Ratanamahatana and Gunopulos, 2002). Mutual Information (MI) is one of the most frequently used information measures. Its aim is to appraise the mutual dependence between two variables, characterized as the dissimilarity between the total of their entropy values and their joint entropy value. MI is zero when the two variables are liberate and increases with an increment in the reliance of one on the other.

2.8 Cross Validation

Cross validation is a model assessment approach that is better than residuals. Data used for model generation is divided into 2 groups, training data and test data. Training data is for model training and test data is for model evaluation. Subsequently, once training has been completed, the eliminated data can be employed to analyze the efficiency of the learned model on new data, which is the fundamental concept behind an entire class of cross validation model assessment approaches.

Normally, data set does not provide independent test set separately; we have to split it into these two groups. The popular way that is often used for splitting is the k-fold cross validation (k-fold CV). K-fold cross validation is one technique to enhance over the holdout method. The data set is subdivided into *k* subsets, after which the holdout method is reiterated *k* times. During each iteration, one of the *k* subsets is employed as the test set and the other *k-1* subsets are combined to form the training set. Next, the average error across all *k* trials is calculated. The benefit of this method is that how the data is actually divided is inconsequential. Every data point is selected one time only in the test set and *k-1* times in the training set. The disadvantage of this approach is that the training method must be repeat *k* times, which implied that an inordinately long time is required to evaluate the results.

2.9 Genetic Algorithm

A genetic algorithm (GA), introduced by Goldberg and Holland (1988), is an adaptive optimization search algorithm to find an optimal solution inspired by natural selection in biological systems. The genes of an organism are gathered into structures called chromosomes, and a collection of chromosomes is referred to as a population. In general, there are three operations employed in GAs. First, selection is an operator for selecting potentially useful solutions for recombination, and is achieved by either tournament or roulette wheel selection (see Figure 3). Second, crossover ascribes to the process of creating an offspring chromosome from two matching parent chromosomes (see Figure 4). There are various categories of crossover: single point crossover, two point crossover, and uniform crossover. Crossover is an operation to produce child subsets recombined from parental chromosomes that consist of splitting chromosome pairs at random. Third, mutation causes genetic diversity of chromosomes by making random binary changes in a chromosome (Cedeño and Vemuri, 1999), thus adversely affecting their fitness value (see Figure 5). These principles have led to new solutions in the pursuit of better search solutions.

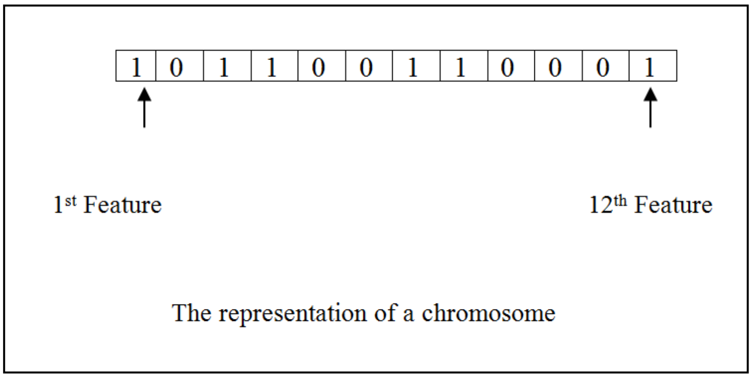


Figure 2.14 The Representation of a Chromosome

Source: Saha, 2017.

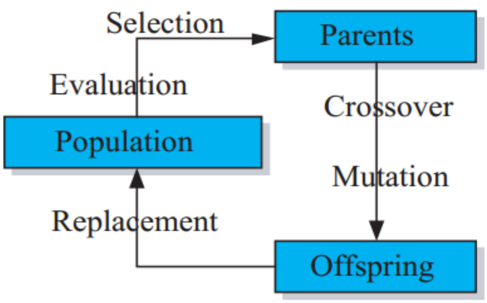


Fig. 2. Evolutionary cycle.

Figure 2.15 Simple of GA Algorithm

Source: Huang and Wang, 2006: 231-240.

There are many type of GA operation. Below is description.

2.9.1 Selection

In principle, the selection operator concludes which individuals are selected from the search space for generating (reproduction and how many offspring each selected individual produces) serves as candidate solutions to optimize the problem. The individuals in this population are measured for goodness of fit ("fitness") through a function. The selection strategy is "the better the individual, the higher its probability of being a parent." The next generation is chosen by an alternative mechanism between parents and their offspring. This method is repeated until a certain condition is satisfied.

Normally, crossover and mutation operators attempt to find the new search space, while the selection operator reduces the search area within the population by eliminating poor solutions. On the other hand, poor individuals should not be removed and they may have the favorable circumstances to be chosen because this may drive to beneficial genetic material.

2.9.1.1 Roulette Wheel Selection

In a roulette wheel selection, this is a simply method. The circular wheel is divided by fitness values. Each individual is assigned a segment of roulette wheel. A fixed point is selected on the wheel. The area of the wheel which appears in front of the fixed point is selected as the parent. To find the subsequent parent, the similar procedure is reworked.

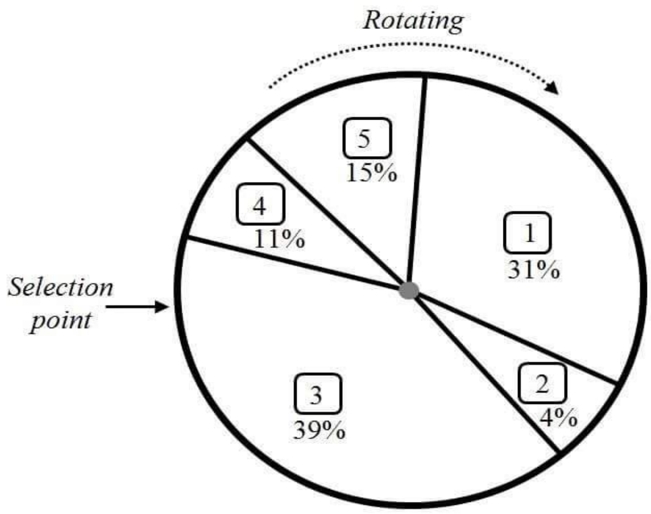


Figure 2.16 Roulette Wheel Selection

Source: Newcastle University, 2017.

In conclusion, it is shown that a fitter individual has a better pie on the wheel and consequently a larger opportunity of landing in front of the fixed point when the wheel is spun. Hence, we can imply that the probability of selecting parents depends on theirs fitness. This way will have a difficulties when the fitness value differs very much. If the foremost fitness chromosome is 98%, the rest of chromosome has a little chance to be selected.

```
Algorithm: ROULETTEWHEELSELECTION()
r := random number,
where $0 \le r < 1$;
sum := 0;
for each individual i
 {
            sum := sum + P(choice = i);
    if r < sum
               { return i; }
           }
```

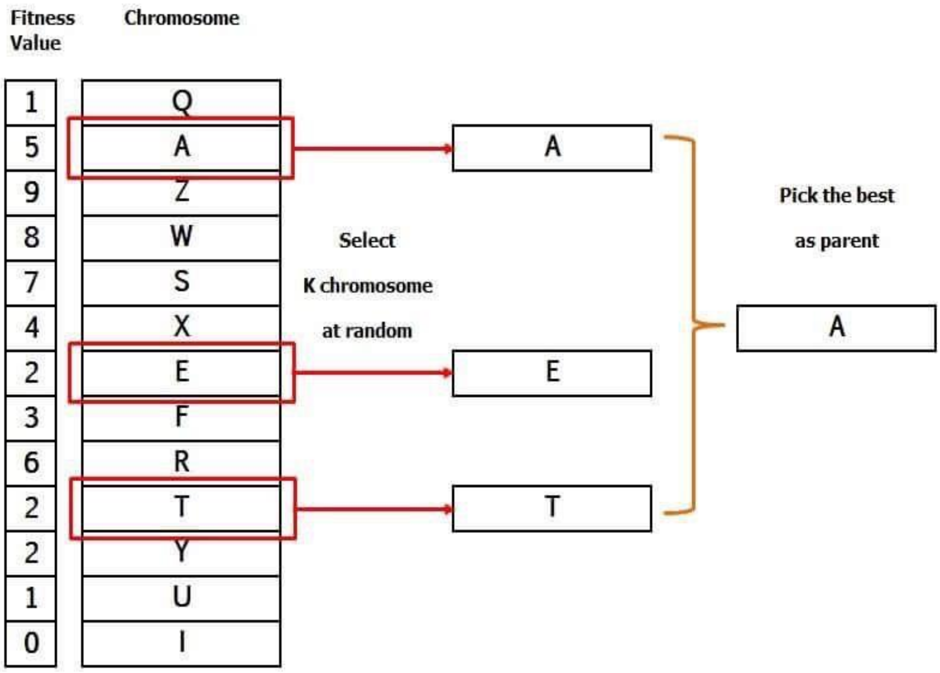
2.9.1.2 Stochastic Universal Sampling (SUS)

Stochastic Universal Sampling is closely the same as to Roulette wheel selection, the different point is that we can have multiple fixed points. As the result, all the parents are selected in just one spin of the wheel. SUS is another method of RWS that attempts to decrease the risk of premature convergence.

2.9.1.3 Tournament Selection

Tournament selection is the most prevalent selection method for genetic algorithms because of its efficiency and simple implementation. It is a modified version of rank-based selection methods, and its strategy combines to randomly select a set of k individuals. When these individuals are measured by a fitness function, the individual with the highest fitness wins and becomes part of the next generation's population. The whole process is repeated n times for the entire population.

Repeating selection of individual chromosomes, we have k-way tournament selection at random and choose the best one to become parents.



Fitness Value	Chromosome
1	Q
5	A
9	Z
8	W
7	S
4	X
2	E
3	F
6	R
2	T
2	Y
1	U
0	I

Figure 2.17 Tournament Selection

Source: Tutorialspoint, 2017.

2.9.1.4 Rank Selection

The method of rank Selection is to sort the population first in accordance with measuring quality and ranks them. Then every individual chromosome is divided selection probability corresponding with its grade. The selection of the parents depends on their rank rather than their fitness (Mangano, 2008). Thus, higher ranked individuals are selected more often than lower sorted ones. Moreover, scaling problems such as stagnation or premature convergence are overcome by rank selection since it controls selection pressure by uniformly spreading scaling across the population. Below table is the example of ranking population.

Table 2.2 Rank Selection

Chromosome	Fitness Value	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

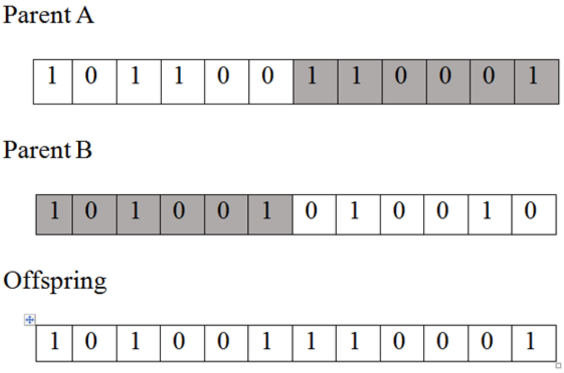
Source: Tutorialspoint, 2017.

2.9.2 Crossover

Crossover is a procedure of bringing more than one parental solutions and creating a new offspring solution from them. The idea behind crossover is that new offspring will possess good characteristics if the best components from each of the parents are exploited. Crossover probability is used to indicate a ratio of how many bits will be selected in the selection step. Mostly, crossover is employed in a Genetic Algorithm with a great probability – $\boldsymbol{p_c}$.

2.9.2.1 One Point Crossover

One point crossover is the most well-known of these methods and is widely applied. Following selection, a crossover operator selects two mating chromosomes. Afterwards, a distinct crossover point on both parental organism's strings is randomly selected and both parental chromosomes are split at the random crossover point. Consequently, the tails of the two parents are exchanged to obtain different offspring.



Parent A

1	0	1	1	0	0	1	1	0	0	0	1

Parent B

1	0	1	0	0	1	0	1	0	0	1	0

Offspring

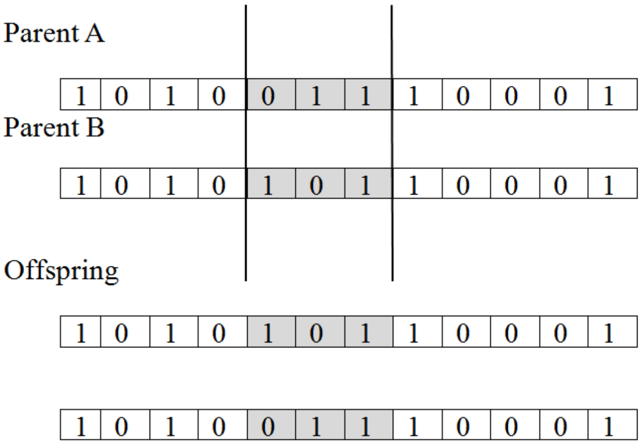
1	0	1	0	0	1	1	1	0	0	0	1

Figure 2.18 Example of One Point Crossover

Source: Kaya and Uyar, 2011: 1105-1355.

2.9.2.2 Two Point Crossover

Similar to single point crossover, two points are selected on the originator organism strings and everything between them is changed between the forerunner organisms, thus displaying two child genomes:



Parent A

1 0 1 0 0 1 1 1 0 0 0 1

Parent B

1 0 1 0 1 0 1 1 0 0 0 1

Offspring

1 0 1 0 1 0 1 1 0 0 0 1

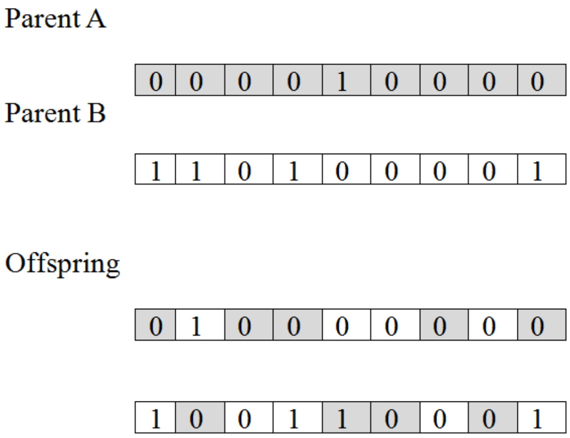
1 0 1 0 0 1 1 1 0 0 0 1

Figure 2.19 Example of Two Point

Source: Kaya and Uyar, 2011: 1105-1355.

2.9.2.3 Uniform Crossover

In a uniform crossover, we operate each gene separately so we do not divide the chromosome into segments. Instead, each chromosome is randomly selected to decide whether to include or not it in the off-spring.



Parent A

0 0 0 0 1 0 0 0 0

Parent B

1 1 0 1 0 0 0 0 1

Offspring

0 1 0 0 0 0 0 0 0

1 0 0 1 1 0 0 0 1

Figure 2.20 Uniform Crossover

Source: Umbarkar and Sheth, 2015.

2.9.3 Mutation

In common terms, mutation is the process to change in the genetic sequence, to introduce a novel solution in the potential search space. It is recognized to control and present a cause of variety in the genetic population and is normally employed with a small probability – $\boldsymbol{p_m}$. If the probability is very high, the GA gets decreased to a random search.

Mutation is a genetic operator which provide investigation of the population. It has been recognized that mutation is vital process to the convergence of the GA albeit crossover is not.

2.9.3.1 Bit Flip Mutation

Bit inversion -selected bits are inverted

1	0	1	0	0	1	1	1	0	0	0	1

After Mutation

1	0	1		0	1	0	1	1	0	0	0	1

Figure 2.21 Example of a Mutation Operator

2.9.3.2 Interchanging Mutation

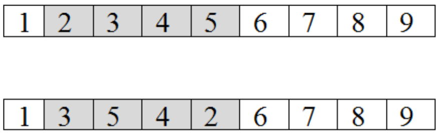
Interchanging mutation is processed by choosing two random positions of the string. And positions are selected and the bit strings according to those positions are exchanged.

2.9.3.3 Swap Mutation

It is also used in Permutation encoding. To perform swap mutation we randomly select two alleles and exchange their positions. In this way, most of the adjacency information is preserved but the broken links markedly disrupt order.

2.9.3.4 Scramble Mutation

Scramble mutation is also used with permutation-encoded chromosome. In this mutation, we unintentionally select a subset of genes, after which the alleles are randomly rearranged in those positions without requiring the subset to be adjacent.



1	2	3	4	5	6	7	8	9

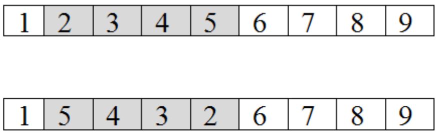
1	3	5	4	2	6	7	8	9

Figure 2.22 Example of Scramble Mutation

Source: Tutorialspoint, 2017.

2.9.3.5 Inversion Mutation

Similar to scramble mutation, the entire string in the subset is simply inverted instead of being shuffled. In the first step, we have to select two points at random position of chromosome and invert substring between them.



1	2	3	4	5	6	7	8	9

1	5	4	3	2	6	7	8	9

Figure 2.23 Example of Inversion Mutation

Source: Tutorialspoint, 2017.

The general scheme of evolutionary in genetic along with Pseudocode is shown below:

```
Input: Population, set of individuals
       Fitness_FN, a function which determines a
       Quality of individuals.
repeat
     new_population ← emptyset
     loop for i from 1 to size(Population) do
     x ← random_selection (Population,Fitness_FN)
     y ← random_selection (Population, Fitness_FN)
    child ← crossover (x,y)
    child ← mutate(child)
    add child to new_population
```

```
population ← new_population
until some individual is fit enough
return the best individual.
```

Genetic algorithms (GAs) have been favorably employed to attribute selection (Brill, Brown and Martin, 1992) with the objective to save on computational time without processing in an exhaustive fashion, by finding promising regions and selecting quality feature subsets. Furthermore, hybrid Gas (Oh, Lee and Moon, 2004) are involved in a new search method with local search operators that enhance the fine-tuning quality of a native GA search.

The fitness function, formed on the significant of survival of the fittest, is the process whereby GA assesses each individual's fitness and obtains the optimal solution after applying the genetic operators. This process is executed repeatedly over many generations until the stopping criterion has been met. For feature selection, the feature subsets are represented as a binary; a feature is either contained or not incorporated in the attribute subset.

2.10 Niching Method

Niching is an improved version of the simple genetic algorithm. It is necessary if the problem of interest is discovering multiple solutions and it is permitted to investigate the finding of better solutions or different optimum subsets. The characteristics of the feature selection problem are multimodal because results will be shown in either the local or global multiple optima solutions and a classical GA cannot yield an adequate outcome without the premature convergence problem. To address some of the multimodal problems (Pedroso, 1996), the performance of a genetic algorithm (GA) (Drchal, Šnorek and Kordík, 2006) cannot effectively yield or maintain an optimal solution. Thus, a population can become easily trapped by a premature condition with no possibility of generating better outcomes. Premature convergence is one of the most important difficult obstacles that occur when employing a GA to a complicated situation related to the diversity of a population. Hence, too much population diversity can bring about poor GA efficiency, and so the necessary trade-off between exploitation of the outstanding individuals and a more thorough search of the search space is significant. Moreover, niching methods (Ye, Qi and Xiao, 2011) have been generated to decreased genetic drift impacts resulting from the replacement operator in a native GA. Niching techniques that extend the genetic algorithm domain are known for their capability to investigate multiple optima, which means grant the GA to investigate many peaks in parallel. In each generation, some poorly adapted niches are eliminated while better adapted ones remain, and niches which vanish are replaced by new ones; these come into view as a choosing of individuals migrated from the current population.

Due to the problem of a lack of population diversity, all chromosomes in the population become almost similar and effect premature convergence in that the genetic operators can no longer create children that surpass their originators. Consequently, a niching method is a crucial controlling factor that helps to maintain a diverse population of candidate solutions, and this requirement ensures that the solution space is adequately searched. In other words, they stop the GA from becoming fastened in the local optimum in the exploration space. Significant methods of niching GAs reviewed in the literature are discussed here.

2.10.1 Fitness Sharing

Fitness sharing was first proposed by Goldberg (1989). Niches are maintained in a population by adjusting individuals' fitness, which diminishes each population component’s fitness by a value corresponding to the amount of comparable individuals. Typically, the corrected fitness is called *shared fitness* and can be expressed as an individual's fitness divided by its *niche count*:

$$f'(i) = \frac{f(i)}{\sum_{j \in P} sh(d(i,j))}.$$

where *P* is the set of all individuals in the populations and *sh* is the sharing function. The latter evaluation the similarity level between two elements: it should return values near to *1* for similar individuals and converge to *0* for dissimilar ones. The effect of this method is to promote searching in unexplored regions.

By boosting scaling fitness or by changing the fitness competence rule, niche approaches customize a native GA by handling convergence so that numerous peak solutions can be controlled in the potential investigation area. The ability to detect multiple peaks frequently contributes niche GAs the powerful and efficacy needed to investigate optimum assorted multimodal optimization difficulty. However, when accustomed to resolve optimization difficulties, most niche algorithms need former information such as the niche radius or the distance threshold.

2.10.2 Deterministic Crowding

This was originally proposed by DeJong (1975) as a method for maintaining population variety and avoiding premature convergence (Mengshoel and Goldberg, 2008). Crowding is employed in the choosing part of a GA algorithm to help make the decision concerning which individuals among those in the ongoing population and their children will be executed in the next generation. Individuals who only have close neighbors are approached to develop niching.

Crowding consists of pairing and replacement. In the pairing stage, the children are compared to individuals in the current population using a likeness measure such as hamming distance. In the replacement phase, the results of the paring phase are attained for each couple of individuals to help determine which of them will continue in the population. A review of crowding methods for GAs can be discovered in Mahfoud (1995).

There are three predominant types of crowding counting on how the replacement stage is achieved, namely deterministic (Mengshoel, Galán and Dios, 2014; Chen, Liu and Chou, 2014), probabilistic (Mengshoel and Goldberg, 2008), and those based on simulated annealing (Likas, Blekas and Stafylopatis, 1996). From each couple in the replacement phase, deterministic crowding chooses the fittest individual, while probabilistic crowding chooses the surviving individual’s chromosome according to a probabilistic formula that considers a fitness metric. Lastly, a simulated annealing based niche genetic algorithm was demonstrated to strength the optimization capability of a niche genetic algorithm. This method employs well-known rules such as Metropolis or Boltzmann, which bring a temperature parameter in the replacement phase.

Deterministic crowding is a niching algorithm proposed by Mahfoud (1995). This technique inserts new elements into the population by proposing tournament between the offspring and forerunner originating from indistinguishable niches. After the crossover and finally mutation procedure have completed, each offspring replaces its nearest forerunner if it has larger measure value. EA using deterministic crowding can be written in pseudocode as:

```
for i := 0 to N/2
  (p1, p2) := choose_two_random(P);
  (c1, c2) := crossover(p1, p2);
  c1' := mutate(c1);
  c2' := mutate(c2);
  if([d(p1,c1')+d(p2,c2')] <= [d(p1,c2')+d(p2,c1')]) then
      if(f(c1') > f(p1)) insert(c1') else insert(p1) end if;
      if(f(c2') > f(p2)) insert(c2') else insert(p2) end if;
  else
      if(f(c2') > f(p1)) insert(c2') else insert(p1) end if;
      if(f(c1') > f(p2)) insert(c1') else insert(p2) end if;
  end if;
 end for;
```

2.10.3 Clearing

The clearing method was presented by Petrowski (1996). The basic idea is to restrict environmental resources, and it is suitable for solving the extremely difficult search space optimization problem. Furthermore, it has been put in a favorable light for solving difficult multi-objective problems. The mechanism is almost the same as sharing but utilizes the restricted resources concept pertaining to the environment. As opposed to sharing resources among the individuals in a subpopulation from the same niche, the clearing procedure only assigns attributes to the best members of each subpopulation or species. These methods are productive for decreasing the genetic drift caused by the selection operation in traditional GAs and delivering multiple stable solutions.

2.11 Discretization

Discretization (Tsai, Lee and Yang, 2008; Boulle, 2004) or binning is the process of transforming continuous values into discrete ones so that there are a limited number of intervals, and is usually utilized as a preprocessing step. An example is to bin values by age into categories such as 20-39, 40-59, and 60-79. Initially, we find the number of discrete intervals and their boundaries, and then associate each interval with a numeric value. Most often, we must specify the number of intervals. This method is the solution to improving a predictive efficiency model because it can help to deduce the amount of level consideration to discard during modeling, thereby reducing noise or non-linearity. Moreover, binning allows easy detection of outliers and replacement of missing numerical values. When discretizing attributes using a supervised learning algorithm by regarding the class-attribute interdependence, the most common technique is to find the intervals which maximize the information gain by discretizing features with respect to target variables.

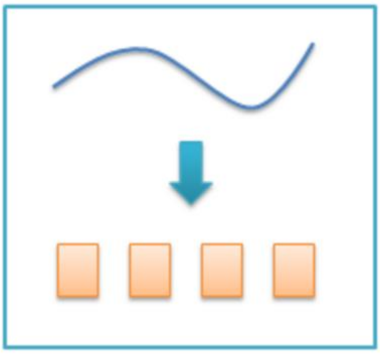


Figure 2.24 Discretizing Features

2.11.1 Discretization Method

2.11.1.1 Unsupervised Methods

Unsupervised binning methods (Dougherty, Kohavi and Sahami, 1995), blind class, convert numerical variables into categorical counterparts but do not apply the target (class) knowledge. These approaches count on presumptions of the dispersion of the feature values. Equal Width and Equal Frequency are two unsupervised binning methods.

2.11.1.2 Unsupervised Methods

1) Equal width Discretization

The approach divides the data into k intervals of equal size. The width of intervals is

$$w = (max\text{-}min)/k$$

And the boundaries range are:

$$min+\ w, min+2w, \ldots , min+(k\text{-}1)w$$

2) Equal frequency discretization

The approach categorizes the data into *k* groups with each assortment containing around the same number of values. With either the equal width or equal frequency discretization algorithms, the most outstanding solution of discovery *k* is to examine the resulting histogram and try out dissimilar intervals or groups.

Data: 0,4,12,16,16,18,24,26,28

Equal width

- Bin 1:0, 4	[-, 10)
- Bin 2:12, 16, 16, 18	[10, 20)
- Bin 3:24, 26, 28	[20, +)

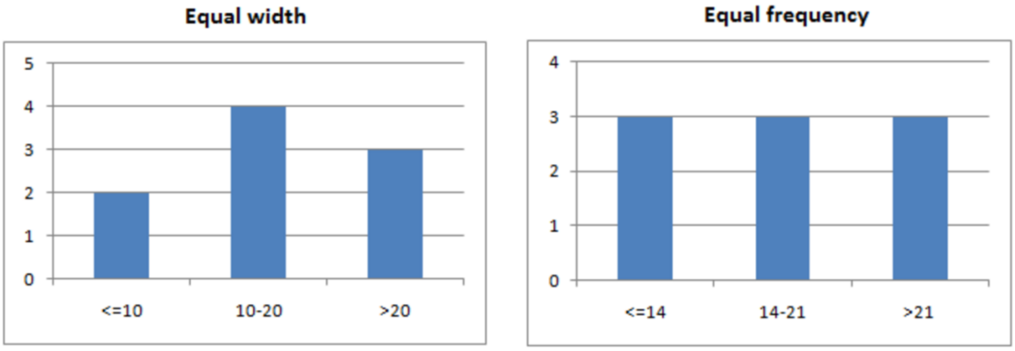


Figure 2.25 Equal Frequency Discretization

Source: Saedsayad, 2017.

2.11.1.3 Supervised Methods

Classification knowledge is attainable, and this information can be taken into attention when discretizing the data. Supervised binning approaches change numerical variables into categorical identical parts and relate to the target (class) information when choosing discretization cut points (Al-Ibrahim, 2011). Entropy-based binning is an instance of a supervised binning algorithm.

1) Entropy-based Binning

Entropy based calculates the homogeneity of a sample which applies a split approach. The entropy (or the knowledge content) is computed counted on the class label (Dougherty et al., 1995). If the data is thoroughly homogeneous the entropy is zero. Usually, it discovers the best split in order that the bins are as perfect as attainable that is the greater number of the values in a bin belong to have the equivalent target information. Normally, it is distinguished by involving the partition of the data into subsets with the maximal information gain. The discretization proceeds by choosing a bin borderline that minimizes the entropy in the resultant partitions. Afterwards, the method is employed repeatedly to both new partitions until the terminating condition is satisfied.

There are two key ideas behind the method:

(1) Data requires splitting into intervals that maximize the information measured by entropy.

(2) Partitioning intervals should not be too small because the problem of over-fitting will occur.

In the primary part of the approach, the data set is divide into two halves contingent on whether the continuous value is above or below the predetermined splitting value, after which the gain in entropy is computed. Out of all of the possible splitting values, the one that generates the best gain is selected, and the process is then repeated recursively.

2.12 Measurements

2.12.1 Mutual Information

In order to perform feature selection with the filter approach, criterions are required to calculate the relevance of the subset to the classification process. MI is a widely used measure to evaluate candidate feature subsets. Battiti (1994) used MI on candidate feature subsets to select a quality subset to be applied as input data for a neural network classifier. MI measures absolute dependencies between random variables and can be calculated as follows:

$$I(X ,Y) = H(X) + H(Y) -H(X ,Y),$$

where *H* is an entropy function, *Y* is a class attribute, and *X* is the feature to select .Given a random variable *X* such that

$$X= \begin{cases} 0 \text{ with probability } p \\ 1 \text{ with probability } 1 - p, \end{cases}$$

$$H(X) = -p \log p - (1 - p) \log(1 - p) = H(p)$$

Note that the entropy does not rely on the values that the random variable takes (0 and 1 in this case), but only counts on the probability distribution *p(x).*

2.12.2 Fuzzy Mutual Information

FMI is applied as the second criterion function which has been extended to handle imprecise data (Grande, Suárez and Villar, 2007). Similar to MI, we represent the *Fuzzy Mutual Information* (FMI) of two fuzzy variables X_s and X_t as:

$$FMI(X_s, X_t) = H(X_s) + H(X_t) - H(X_s, X_t)$$

Let us assume that P_s and P_t are strong fuzzy partitions consisting of T_s and T_t fuzzy sets, respectively, defined on X_s and X_t. Then, the fuzzy entropy $H(X_s)$ of the variable X_s can be calculated as

$$H(Xs) = -\sum_{T_s \quad i=1} P(A_{s,i}) \cdot \log P(A_{s,i})$$

where $P(A_{s,i})$ is the probability of the fuzzy set $A_{s,i}$ and is defined for a distribution $\{x_1,\ldots,x_N\}$ with respect to a probability distribution $P = \{p_1,\ldots,p_N\}$ *as* $P(A_{s,i}) = N\ i=1\ \mu A_{s,i}\ (x_i) \cdot p_i$ where $\mu A_{s,i}\ (x_i)$ is the membership degree of x_i to the fuzzy set $A_{s,i}$. Similarly, the fuzzy joint entropy $H(X_t, X_s)$ can be computed as:

$$H(X_t, X_s) = -(\sum_{i=1}^{T_t} \sum_{j=1}^{T_s} P(A_{t,i}, A_{s,j}) \cdot \log P(A_{t,i}, A_{s,j}))$$

The joint probability $P(A_{t,i}, A_{s,j})$ is computed as in :

$$P(A_{t,i}, A_{s,j}) = \sum_{N=1}^{k=1} \sum_{N=2}^{h=1} \mu_{At,i} \cap A_{s,j} (x_{k,t}, x_{h,s}) \cdot p(x_{k,t}, x_{h,s})$$

where N_1 and N_2 are the numbers of different values for the variables X_s and X_t in the dataset, respectively, and $\mu A_{t,i} \cap A_{s,j} = \mu A_{t,i}\ (x_{k,t}) \cdot \mu A_{s,j}\ (x_{h,s})$

2.12.3 Pearson Correlation

The full name is the Pearson Product Moment Correlation or PPMC, and it is the most commonly used correlation measure in statistics (Chee, 2015). The objective is to measure the strength of association between sets of data that identify how well they are related, for example age and blood pressure. Normally, it is used to uncover a potential linear relationship between them, and is defined as

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\,n\Sigma x^2 - (\Sigma x)^2\,]\,[\,n\Sigma y^2 - (\Sigma y)^2\,]}}$$

We can classify the group of correlation by considering as one variable increases what happens to the other variable:

2.12.3.1 Positive correlation – the other attribute has an inclination to also boost.

2.12.3.2 Negative correlation – the other attribute has an inclination to decline.

2.12.3.3 No correlation – the other variable does not tend to either enhance or diminish.

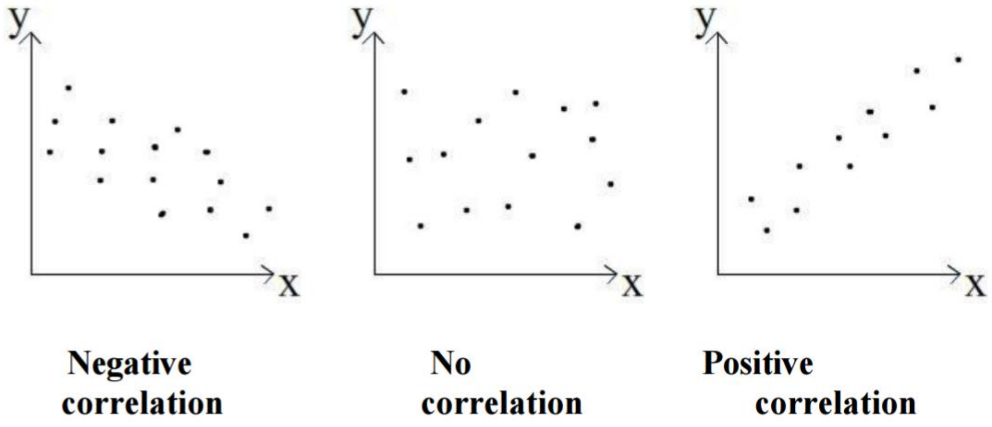


Figure 2.26 Examples of Negative, No and Positive Correlation

The main drawback of Pearson Correlation is that does not take into consideration whether variable is dependent or independent. For instance, calculating the correlation between a high calorie diet and diabetes would result in a high correlation of 0.8. Nevertheless, you could exchange the variable around in the correlation coefficient formula and conversely infer that diabetes causes a high calorie diet. That looks no uses at all. As a result, as a researcher you have to be mindful of the data you are plugging in. Moreover, the Pearson Correlation will not contribute you any information about the slope of the line.

2.12.4 Euclidean

Euclidean Distance is the most ordinary and fundamental use of distance (McCune, Grace and Urban, 2002). In most common measures when people consider about distance, they will bear on Euclidean distance. Euclidean distance or easily 'distance' examines the root of square differences between coordinates of a pair of objects. In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" distance between two points. The Euclidean distance between two points A = (x1, x2, x3, …, xn) and B = (y1, y2, y3, …, yn) is defined as:

$$d(A,B)=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+(x_3-y_3)^2+\ldots+(x_n-y_n)^2}=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

However, the Euclidean distance only works for data measured on the same scale. Usually, this measure is most often used to compare profile of respondents respectively with variables. For instance, suppose a sample consists of compounded demographic data of individuals arranged as a respondent-by-variable matrix. Each row of the matrix is a vector of *n* numbers, where *n* is the number of variables. We can evaluate the similarity or distance between any pair of rows and the variables are the column. Each column has its own scale that using to determine type and size of number. For instance, a measure of income might yield numbers between 0 and 1000 million, while another variable, a measure of education, might consist of numbers from 0 to 100. Even though the income numbers are larger in general than the education numbers, the difference is not purposeful because the attributes are calculated on dissimilar scales. In order to compare columns, we must consider the topic of differences in scale. On the other hand, the row vectors are different. They are not variables. Even if one variable has larger numbers than the other, this is not a cause for concern because rows are unaffected by scale. With regard to taking dissimilarities among the rows into account, there is no need to attempt to fine-tune for differences in scale. Therefore, Euclidean distance is usually the suitable measure for comparing cases.

2.12.15 Mahalanobis Distance

The Mahalanobis distance was first discovered by the Indian statistician P. C. Mahanobis in 1936 (Sapp, Obiakor, Gregas and Scholze, 2007). The Mahalanobis distance accounts for the variance of each variable and the covariance between variables. This is achieved by transforming the data to make them standardized and uncorrelated and then calculating the Euclidean distance for the transformed data. In this way, the Mahalanobis distance acts like a univariate z-score, thereby offering a solution to measuring distances by taking into account the scale of the data.

The Mahalanobis distance is the distance from *X* to the quantity μ defined as:

$$d2M(X,\mu)=(X-\mu)t\sum-1(X-\mu).dM2(X,\mu)=(X-\mu)t\sum-1(X-\mu).$$

This distance is based on the correlation between variables or the variance–covariance matrix. It calculates different from the Euclidean distance because Mahalanobis considers the correlation of the data set and does not reckon on the scale of measurement. Mostly, Mahalanobis distance is famous in cluster analysis and other classification method.

2.13 Classifier

2.13.1 Decision Tree

Typically, Classification and Regression Trees is a classification method which uses historical data to create decision trees. Decision trees are then used to classify new data. In order to apply CART, the number of pre-assigned class for all of the observed data is required. For example, a learning sample for a credit scoring system would require information on previous loans (variables) matched with actual repayments (classes). Two broadly used methods for constructing decision trees are

Classification and Regression Trees (Guyon and Elisseeff, 2003) and ID3/C4.5 (Hssina, Merbouha, Ezzikouri and Erritali, 2014). Decision trees are demonstrated by a set of question. CART always ask “Yes/No” question such as “Is age greater than 50?” . The tree attempt to find the best split of the training data based on the quality of data which has maximum purity of information gain. Each leaf node corresponds to a class label. A new example is classified by following a path from the root node to a leaf node. The leaf node reached is considered the class label for that example. The process is then repeated for each of the resulting data fragments. The algorithm can naturally handle binary or multiclass classification problems.

CART methodology comprises of tree parts: 1. Construction of maximum tree 2. To select choice of the right tree size 3. Classification of new data using constructed tree

Here is an example of San Diego Medical Center for classification of their patient.

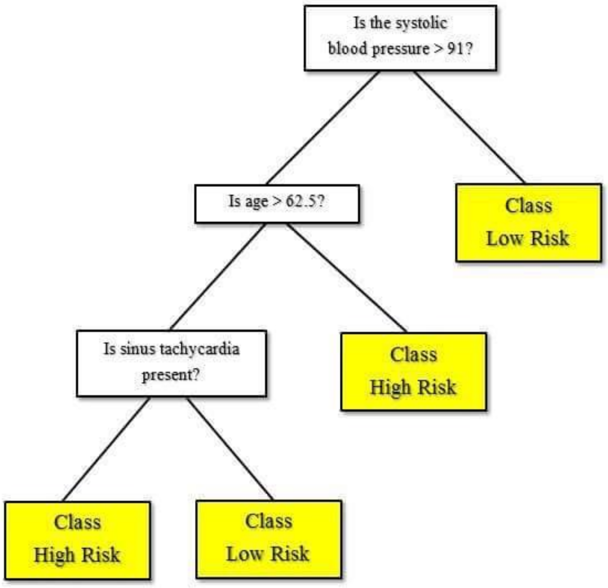


Figure 2.27 Classification Tree of San Diego Medical Center Patients

First introduced by Breiman, Friedman, Olshen and Stone (1984), CART is a well-known decision tree algorithm for supervised machine learning that is applied to both classification and regression problems. A decision tree represents a series of decisions. The key components of the tree are a collention of rules for splitting each node in the tree, and assigning a class outcome to each terminal node. In this study, CART employs the Gini impurity index as a assess to construct a decision tree. Consider parent node l that contains data belonging to the jth class, then the impurity function for node l is derived as

$$i(l) = 1 - \sum_{j} p^{2}(j|l),$$

and the declination of impurity of the split is denoted as

$$\Delta\, i(l) = i(l) - p_{L}\, i(n_{L}) - p_{R}\, i(n_{R}),$$

where l is a parent node which is split into nodes n_L and n_R .After this, the CART strategy is applied by choosing the feature that maximizes the decrease of impurity $\Delta\, i(l)$ at each subsequent node

2.13.2 Naïve Bays

The naive Bayes algorithm is family of simple probabilistic based on Bayes formula to decide which class a novel instance belongs to. The basic idea is that all naive Bayes classifiers presume that the value of a specific feature is unconnected of the value of any other feature, given the class variable. For instance, a fruit may be examined to be an apple if it is red and round with an approximate diameter of 10 cm. In order to ascertain the probability of this being an apple, a naive Bayes classifier considers each of these attributes individually regardless of any possible correlations between them (Dangi and Prashant Ahlawat, 2015).

Naive Bayes algorithms are a statistical classifier for supervised learning (Hsu, Chang and Lin, 2016). They are based on the principle of conditional probability and can forecast class membership probabilities, such as the probability that a given sample belongs to a specific class .Their performance is shown to be excellent in some domains but poor on specific domains, e.g., those with correlated features .The classification system is based on Bayes ’rule under the assumption that the effect of a particular attribute on a given class is independent of the other one. This supposition that makes computation simple is referred to as class conditional independence .A conditional probability model for the classifier is given as P (C_i |x) Using Bayes' theorem, we can write

$$P(C_i \mid x) = \frac{(P(C_i) * P(x|C_j)}{P(x)}$$

where C_i is the i^{th} class and x is the input vector .In this case, class variable C is conditional on several features: variable x =x1,… , xn.

2.13.3 Support Vector Machine

SVM, originally proposed by Cortes and Vapnik (1995) and Hsu et al. (2016) becomes important in many classification problems for a variety of reasons, such as their adjustability, calculation capability, and capacity to deal with large structural data. VMs are a recent method to extract information from a dataset in which classification is accomplished by applying a linear or nonlinear separating surface in the input space of the dataset. They have been employed in a number of fields including bioinformatics, face recognition, text categorization, and handwritten digital recognition, among others. They are a binary classifier assigning a new data to a class by minimizing the probability of error .

Given a training set of instance-labelled pairs (xi , yi), i = 1, . . . , l, where xi ∈ Rn and y ∈ {1, −1} 1 , the SVM requires the solution of the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned}$$

Its dual is

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha$$
$$\text{subject to } y^T\alpha = 0$$
$$0 \le \alpha_i \le C, i = 1, \ldots, n$$

where e is a vector of all ones, $C > 0$ is the upper bound, Q is an n by n positive semidefinite matrix, and $Q_{ij} = y_i y_j K(x_i, x_j)$, where $K\left(x_i, x_j\right) = \emptyset\,(x_i)^T\,\emptyset\,(x_j)$ is the kernel. Here, training vectors are implicitly mapped into a higher (maybe infinite) dimensional space by the function $\emptyset$.

.

2.13.4 K-nearest

k-Nearest Neighbors kNN (Bins and Draper, 2001) is an understandable method that keeps all feasible cases and classifiers and is contemplated among the oldest non-parametric classification algorithms in the beginning of 1970. KNN has been used in statistical estimation and pattern recognition. To classify an unknown example, the distance function is based on similarity measure such as Euclidean, Manhattan measures. After identifying the k smallest distances, and the most descripted class in the resultant k classes becomes the output class label. The value of k is most often decided either on using a validation set or by cross validation. All two distance measures are only apply for continuous variables but for the instance of categorical variables of Hamming distance must be utilized and also the technique of standardization of numerical variables between 0 and 1.

2.14 Confusion Matrix

A confusion matrix is particular layout table that gives visualization of effectiveness of method or illustrate the efficiency of a classification model or classifier on a set of test data for which the true values are known. The confusion matrix is demonstrated in an understandable way, but the related terminology can be complex. It is a various kind of contingency table, with each column of the matrix demonstrates the instances in a predicted class while each row demonstrates the instances in an actual or true and predicted class.

CHAPTER 3

IMPROVING FLOATING SEARCH FEATURE SELECTION USING GENETIC ALGORITHM

Filter methods seem to perform more poorly than wrapper methods but are a reasonable compromise for feature selection problems. Wrapper methods are broadly known to be greater in supervised learning problems since they use an inductive algorithm to assess opportunity and thus take the bias of a particular algorithm into account. Nevertheless, these algorithms show complexity; for instance, the number of executions that the search procedure can result in expensive computational cost, especially when shifting to more exhaustive search approaches. A hybrid approach is an attractive method due to acquiring the best characteristics of both approaches while limiting the influence of their drawbacks.

Accordingly, supervised strategies for filtering mainly depend on the characteristics and relationships between the data and a predefined class label. On the contrary, for clustering tasks, this turns out to be a difficult problem because we need to establish what is going to be relevant to disclose a structure not known in advance. Currently, the aspiration of this research was to develop a new approach to improve the quality of the attribute set selected from a filter-based method for reaching the gold of high classification accuracy, and so an additional step of filter-based SFFS by employing a genetic algorithm was applied, which allowed more variety in the candidate feature sets.

3.1 Proposed Method

We now discuss our algorithm for its use in selecting the outstanding subset of size d of the total of D attributes, as shown in Figure 3. The inclusion step using MI as the measure function (*J*) is executed to create a set of candidates for inclusion. In the exclusion step, a candidate feature subset is used to create smaller subsets from the result of the inclusion step by removing one feature and re-evaluating them. A selection subset of size k+1 is created and compared to the previously best subset of size k+1 from the inclusion part. If evaluation of the new subset is more qualified than the formerly selected set, the exclusion step retains the better one and iterates to smaller subsets, or else the algorithm goes back to the inclusion step.

Our feature improvement step based on GA is included after the exclusion step at each iteration. The objective is to replace the weakest attribute to examine whether eliminating any feature in the currently picked feature subset and including a promising one at each continuous step potentially improves the current attribute subset. The chromosome structure consists of binary genes, corresponding to individual features. The value of 1 at the i^{th} gene means that the i^{th} feature is chosen; otherwise it is 0.

The initial population is generated from the resulting feature exclusion subsets of size k+1 from the exclusion step by first removing the weakest features from the best subset resulting in a subset of size. k based on the number of feature subsets in each round is referred to. Each remaining feature is thus added to that subset generating the niched initial population for GA. The fitness function used in this study is MI. Then, a new population is produced by selection, crossover and mutation operations. The approach is terminated when the current feature set reaches the size of D-2 features.

Parameter tuning in this experiment is also suggested: a population size of 4-100 individuals, a bit-flip mutation rate of 0.01, and for a single point crossover, a rate of 0.75 and the number of generations was 500. Besides, crossover is helpful against local optimum solutions because it allows the algorithm to try combinations of innovations from different solutions. Hence, it is assumed that the experimental results can solve the unwanted problem of arriving at the local optimum solution.

We now provide an illustrative example of how the proposed algorithm works and how it improves SFFS. Presume that the first five feature sets chosen by the SFS approach at each size are {f1}, {f1, f4}, {f1, f4, f5}, {f1, f4, f5, f7} with the corresponding J values of 4.1, 6.2, 9.1 and 10.2, respectively, and the next iteration is to determine subsets with five features.

3.1.1 Cross Validation

Cross validation is a model assessment approach that is better than residuals. Data used for model generation is divided into 2 groups, training data and test data. Training data is for model training and test data is for model evaluation. Then when training is done, the data that was eliminated can be employed to test the performance of the learned model on new data. This is the fundamental idea for a whole class of model evaluation methods called *cross validation*.

Normally, data set does not provide independent test set separately; we have to split it into these two groups. An improvement over the holdout method that is often used for splitting is k-fold cross validation (k-fold CV). In this method, the data set is divided into *k* subsets and the holdout method is repeated *k* times. During each repetition, one of the *k* subsets is utilized as the test set and the other *k-1* subsets form the training set. Afterwards, the average error across all *k* trials is computed. The advantage of this approach is that it is immaterial how the data is divided since every data point gets to be in the test set one time only and *k-1* times in the training set. The drawback of this approach is that the training approach has to be repeat *k* times, thus considerable time is required to evaluate the results.

3.1.2 Discretization for Continuous Data

Discretization is the procedure of converting continuous values into discrete ones so that there are a limited number of intervals, and is usually utilized as a preprocessing step. Initially, we find the number of discrete intervals and their boundaries, and then associate each interval with a numeric value. Most often, we must specify the number of intervals. This method is essential to improving a predictive performance model because it can help to deduce the amount of level consideration to discard during modeling.

This research will focus on discretized technique which begins with sorting the data set and selecting only duplicate value for cutting point bin. At this step, they find the number of discrete values to represent each bin. The range associated with an interval must be divided into k interval depend on number of replicate values. We give an illustrative example of discretization process. Suppose that we have one dimension data and then data was sorted so that the results of cutting bin value are 1,2,5,7.

0	1	1	2	2	3	4	5	5	6	7	7	8	9

Afterwards, a discretized process makes the partition decision based on cutting bin value. It begins by replacing observed data value into its own bin number.

1	1	1	2	2	3	3	3	3	4	4	4	5	5

3.1.3 Inclusion

A feature is added to the feature subset. The SFS method adds a feature to the subset up to a total of five: J(f1,f4,f5,f7,f6) = 13. Presume that feature f6 is chosen using the SFS method and J for the 5th features is 14.

3.1.4 Exclusion

A feature is removed from the feature subset. The SBS method is applied in this step by backtracking and conditionally removing one feature from the subset selected in Step 1 and returning an improved subset, e.g. (f1,f5,f6,f7) j value = 11, (f1,f4,f5,f7) j value = 9, (f1,f4,f7,f6) j value = 9.5, and (f4,f5,f7,f6) j value = 10. In this case, the best feature subset of size 4 is (f1, f5, f6, f7).

3.1.5 Feature Improvement

The weakest feature is removed from the subset from of size k the previous step which is (f1, f5, f6, f7) by iteratively evaluating the smaller subsets: (f1, f5, f7), (f1, f5, f6), (f5, f6, f7) and (f1, f7, f6). In this case, we assume that the best performance subset of size 3 is (f5, f7, f6). Then, each feature is added to each subset of (f5, f7, f6) in order to find the best four feature subset, either (f5, f7, f6, f1), (f5, f7, f6, f2), (f5, f7, f6, f3), (f5, f7, f6, f4), (f5, f7, f6, f8), or (f5, f7, f6, f9.) Top n chromosomes is selected as the initial population for GA and passed through the crossover and mutation operations. Note that n is a number between 5 and 10.

3.1.5.1 Crossover Operation

Crossover is a genetic operator mainly responsible for creating new answer domains in the search area to be investigated; it is a random mechanism for swapping information among strings in the mating pool (Huang and Wang, 2006). Once a pair of chromosomes has been selected, crossover can take place to produce child chromosomes. A crossover point is indiscriminately selected from two randomly chosen individuals (parents). This point happens between two bits and separates each individual into left and right segments. Crossover then exchanges the left (or the right) part of the two individuals, which is referred to as mating with a single crossover operation:

Parent A – (f5, f7, f6, f2)

0	1	0	0	1	1	1	0	0	0

Parent B – (f5, f7, f6, f1)

1	0	0	0	1	1	1	0	0	0

Suppose the crossover point randomly occurs after the sixth bit, then each new child receives one half of each parent's bits:

Offspring1 – (f1, f5, f7, f6)

0	1	0	0	1	1	1	0	0	0

Offspring2 – (f2, f5, f7, f6)

1	0	0	0	1	0	1	1	0	0	0

This algorithm continues to select parental chromosomes to apply the crossover operation. Child chromosomes may have one bit more than the current size of k features subset. In this case, a random bit is automatically flipped to preserve the size of the chromosome (i.e. current feature set size).

3.1.5.2 Mutation Operation

The mutation operation is applied to all of the offspring chromosomes from the crossover step. Mutation manages at the bit level by randomly flipping bits in the new chromosome within the current population (turning a '0' into '1', and vice versa).

Offspring1 – (f5, f7, f6, f1)

1	0	0	0	1	1	1	0	0	0

After mutation – (f5, f7, f6, f2)

0	1	0	0	1	1	1	0	0	0

After all child chromosomes have passed through the mutation operator, the resultant chromosomes are evaluated by the fitness function. After this, we can discover the best performing features subset, which is (f5, f7, f6, f2). We assume that J({f5, f7, f6}) = 8.35, and that J({f5,f7, f6, f2}) = 12, which is larger than the prior largest value for four features, J = 11 Thus, the best 4-feature subset becomes {f5, f7, f6, f2} with J = 12, whereas the best 3-feature subset remains {f1, f4, f5} since J({f1, f4, f5}) = 9.1 > J({f5, f7, f6}) = 8.35

The improvement step helps discover subsets not discoverable by the greedy nature of SFFS. From the above example, the SFFS algorithm is not able to produce this best 4-feature four subset because it cannot backtrack to the set {f5, f7, f6} as a result of J({f1, f4, f5}) = 9.1 > J({f5, f7, f6}) = 8.35., thus could not add feature f2 to subset {f5, f7, f6}. Note that f2 is never selected in the first best four feature sets of the SFFS method: {f1}, {f1, f4}, {f1, f4, f5} and {f1, f5, f6, f7}. Moreover, after finishing the mutation step, the algorithm control results in feature size subsets to correspond with the k-feature subset of each iteration. If results are an equal k-feature subset, then the algorithm will randomly position itself to swap bit string values depending on situations of 0 to 1 or 1 to 0.

The example above demonstrates the advantage of our proposed algorithm. The algorithm replaces the weak feature (feature f1 in our example) in the feature set {f1,f5,f7,f6} with feature 2, which results in a new set of four features {f5, f7, f6, f2} which has a larger J value. Therefore, the search approach of our proposed method is more thorough than the SFFS algorithm, so it is more effective.

3.1.6 Terminating Condition

After each iteration, the selection / crossover / mutation cycle continues until all possible combinations of chromosomes in the population have been evaluated. The greater the fitness value, the higher the probability of that chromosome being selected for reproduction. This generational procedure is reiterated until a pre-determined stopping condition has been discovered. We discontinue the algorithm when the current feature set reaches $d < D$ features, where D is the total number of features in the dataset). The pseudo-code is depicted in Figure 3.

A fitness function is commonly required in GAs to appraise a candidate chromosome of an individual to assess whether the latter should survive or not. At each iteration, calculation of the fitness function is processed repeatedly, which, because of its simplicity, is a fast process, although it still impacts performance. In our model, we use the MI criterion as a fitness function. Basically, it calculates the amount of an information feature set in a group of variables for the sake of predicting the dependent data. In addition, the fitness function to be calculated includes the calculation of the classification rate, which requires a classifier.

Input: Y_m is a feature set, m is a predefined number of selected features, J is a criterion function. P_c is probability of crossover, P_m is probability of mutation, *Population* is set of individuals, *max_generation* is the maximum number of generations, and Fitness is a function which determines quality of individuals.

Output: The best solution in all generation.

```
(1) Inclusion
        Initialize: $Y_0 = \{\emptyset\}$; $m = 0$
          Find the best feature and update $Y_m$
          $x^+ = \arg\max_{x_i \in Y_m} [J(Y_m - x)]$
          $Y_m = Y_m + x^+$; $m = m + 1$
(2) Conditional Exclusion
          Find the worst feature
          $x^- = \arg\max_{x_i \in Y_m} [J(Y_m - x)]$
           If $J(Y_m - x^-) > J(Y_m)$ then
           $Y_m+1 = Y_m - x$;
      Go to Step 3 Else Go to Step1
(3) Feature Improvement Step.
**Repeat**
    *population* ← SBFS feature subsets $Y_m$
   *generation = 0;*
    **loop for** i **from** 1 **to** size(*Population*) **do**
         ***s1*** ← **selection** (*Population,* Fitness*)*
         **s*2*** ← **selection** (*Population,* Fitness*)*
        *child* ← **crossover** (*s1*,*s2*) with *pc* and check feasibility of *n* element
       *child* ← **mutate**(*child*)  with *pm* and check feasibility of *n* element
       *generation* = *generation* +1
     **until *generation* < *max_generation***
  $m = m + 1$
return the best individual solution $Y_m$
```

Figure 3.1 Pseudo-Code of the Proposed Algorithm

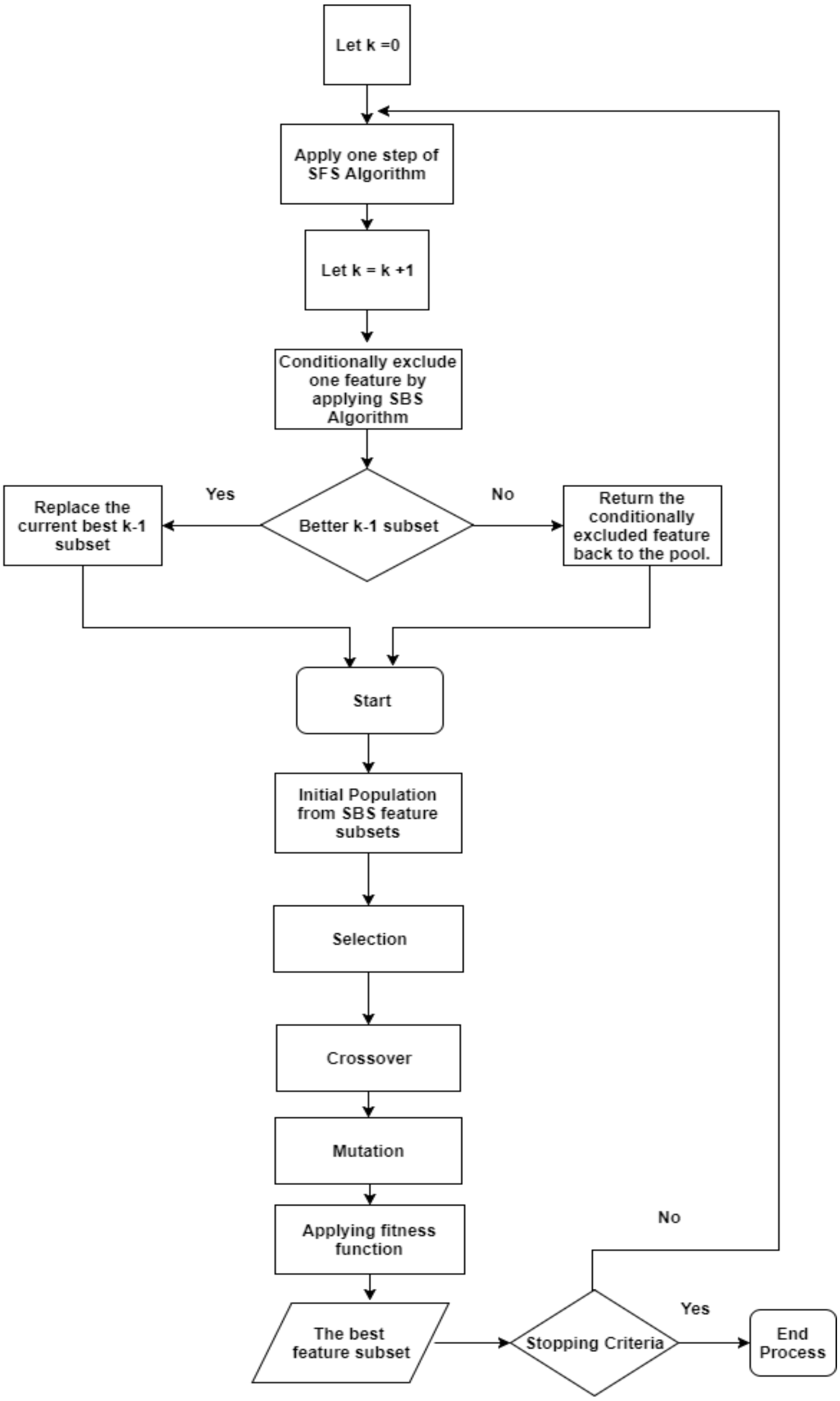

Let k =0
Apply one step of SFS Algorithm
Let k = k +1
Conditionally exclude one feature by applying SBS Algorithm
Better k-1 subset
Yes
Replace the current best k-1 subset
No
Return the conditionally excluded feature back to the pool.
Start
Initial Population from SBS feature subsets
Selection
Crossover
Mutation
Applying fitness function
The best feature subset
Stopping Criteria
No
Yes
End Process

Figure 3.2 Structure of Proposed Algorithm

3.2 Evaluation

3.2.1 Data Sets

In this chapter, the datasets used in this work are described and the efficiency of the proposed method is empirically evaluated. To evaluate the proposed feature selection algorithm, 20 standard datasets of various sizes and complexities from the UCI machine learning repository (https://archive.ics.uci.edu/ml/datasets.html) are used in the experiments. These datasets were chosen because they have been extensively used for classification and in other research on the feature selection task, and have been frequently applied as a benchmark to compare the performance of classification methods and consist of a mixture of numeric, real and categorical attributes. Numeric features are pre-discretized by the method demonstrated by Tsai et al. (2008). The focus of this study is on a discretization technique which begins by sorting a dataset and selecting only duplicate values for the cutting point bin. At this step, the number of discrete values to represent each bin is found. The range associated with an interval is divided into k intervals depending on the number of replicated values. This modification enables the discretization process to be faster and to yield a higher performance than is otherwise possible. Three classification modeling techniques are used in the experiments which consist of Classification and Regression Tree CART, Support Vector Machine SVM, and Naïve Bayes .Training and testing data is used as provided in the datasets. For those not providing separate testing data, a 10-fold cross validation is applied .To appraise a feature subset, MI is applied as the criterion function.

Each instance in the training set contains one class label and several feature variables. The objective of a classifier is to generate a model (based on the training data) which predicts the target values of the test data given only the test datan attributes. Three classification approaches are employed in the experiments which consist of classification and regression tree (CART), Naïve Bayes and support vector machine (SVM).

3.2.2 Experimental Setup

From the experiments, we found that a suitable set of parameters are follows: a population size of 4-100 individuals. We choose crossover operator from two operators (Single- point and Two-point operator). Crossover rate is varied between 0.75. Uniform mutation operator is used and mutation rate is 0.01. Lastly, the number of generation is 500.

Table 3.1 Data Used in Experiment

Dataset	Attribute Characteristics	No. of Instances	No. of attributes	No. of Classes
Wine	Integer	178	13	3
Breast Cancer (Original)	Integer	699	10	2
Breast Cancer (WDBC)	Real	569	32	2
Breast Cancer (WPBC)	Real	198	34	2
Iris	Real	150	4	3
Pima-Indian diabetes	Integer, Real	768	8	2
Abalone	Categorical, Integer, Real	4,177	8	3
Dermatology	Categorical, Real	366	34	6
Heart	Categorical, Real	270	13	2
German (Credit card)				

Table 3.1 (Continued)

Dataset	**Attribute Characteristics**	**No. of Instances**	**No. of attributes**	**No. of Classes**
Lung cancer	Integer	32	56	3
Soybean	Integer	307	35	4
Spambase	Integer, Real	4,601	57	2
Glass Identification	Real	214	10	7
Contact Lens	Categorical	24	4	3
Sonar	Real	208	60	2

Table 3.2 Experimental Results with CART, SVM and Naïve Bayes Classifier

Dataset	Original Datasets CART	Original Datasets SVM	Original Datasets Naïve Bayes	Proposed Method with CART	Proposed Method with SVM	Proposed Method with Naïve Bayes	No. of Attributes
Wine	89.87%	62.00%	89.00%	100.00%(7)	100.00%(7)	97.00%(7)	13
Breast Cancer (Original)	93.13%	93.13%	89.55%	97.82%(5)	98.00(6)	95.68%(6)	10
Breast Cancer (WDBC)	92.23%	85.11%	80.00%	95.49%(9)	96.00%(11)	93.00%(11)	32
Breast Cancer (WPBC)	72.00%	74.23%	70.00%	83.00%(6)	85.00%(6)	82.00%(6)	34
Iris	94.00%	94.00%	92.02%	98.44%(3)	100%(3)	95.68%(3)	4
Pima- Indian Diabetes	72.51%	75.00%	69.23%	73.18%(4)	78.00%(4)	72.00%(4)	8
Abalone	49.07%	55.00%	48.02%	52.00%(3)	61.00%(3)	50.26%(3)	8
Dermatology	95.08%	95.00%	91.00%	98.83%(26)	99.00%(18)	95.10%(18)	34
Heart	76.67%	79.00%	73.00%	80.00%(6)	81.11%(5)	79.00%(5)	13
German	68.50%	69.4%	60.00%	73.50%(6)	71.50%(7)	69.00%(7)	20

Table 3.2 (Continued)

Dataset	Original Datasets CART	Original Datasets SVM	Original Datasets Naïve Bayes	Proposed Method with CART	Proposed Method with SVM	Proposed Method with Naïve Bayes	No. of Attributes
Lung cancer	59.67%	60.00%	57.20%	75.00%(21)	83.33%(21)	72.00%(21)	56
Soybean	85.00%	85.00%	83.02%	100.00%(22)	100.00%(20)	98.28%(20)	35
Spambase	93.26%	85.00%	81.00%	96.00%(26)	92.56%(26)	91.76%(26)	57
Glass Identification	62.00%	75.00%	66.00%	63.13%(5)	78.00%(6)	68.00%(6)	10
Teaching Assistant	54.92%	53.02%	53.10%	58.03%(2)	61.86%(3)	62.00%(3)	5
Contact Lens	76.00%	77.00%	72.00%	80.00%(2)	100.00%(2)	85.00%(2)	4
Sonar	69.50%	61.00%	62.65%	76.86%(7)	62.98%(7)	67.00%(7)	60
Statlog (Australian)	65.45%	66.03%	59.00%	74.30%(7)	79.04%(7)	75.24%(7)	14
Ionosphere	84.00%	85.00%	84.00%	88.00%(5)	90.62%(8)	90.10%(8)	34
Image Segmentation	85.00%	83.10%	79.34%	90.95%(14)	88.57%(12)	85.10%(12)	19

Table 3.3 Comparison on Classification Accuracy with Other Recently Reported Methods on Common Datasets

Dataset	**PM* CART**	**PM* SVM**	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Breast Cancer (Original)	**98.10%**	**98.00%**	-	97.40%	94.40%	96.50%	-	-	94.80%
Breast Cancer (WDBC)	**97.0%**	**96.0%**	95.40%	-	-	-	-	-	93.00%
Iris	**98.00%**	**100.00%**	97.30%	-	-	97.30%	96.70%	96.60%	-
Pima Indian Diabetes	78.00%	78.00%	73.80%	79.90%	76.00%	73.20%	-	-	-
German	74.00%	72.00%	72.60%	76.20%	-	74.50%	-	69.90%	-
Soybean	**100.00%**	**100.00%**	-	88.30%	-	97.80%	-	-	-
Wine	**100.00%**	**100.00%**	-	-	91.60%	98.30%	-	-	-
Sonar	**85.86%**	**85.00%**	-	-	83.70%	-	-	-	-
Abalone	**58.00%**	**61.00%**	54.50%	-	-	-	30.00%	25.70%	-
Dermatology	**98.85%**	**99.9%**	-	-	-	95.40%	-	-	-
Contact Lenses	**80.00%**	**100.00%**	-	-	-	-	75.00%	-	-
Lung Cancer	96.24%	96.00%					96.875%		

Note: *Proposed Method

Source: A: Liu, Motoda and Yu, 2004; B: Chotirat Ann Ratanamahatana and Gunopulos, 2002; C: Anwar, Qamar and Qureshi, 2014; D: Yang, Cao and Zhang, 2010; E: Gupta and Ghafir, 2012; F: Tsai, Lin, Hong and Hsieh, 2011; G: Lavanya and Rani, 2011.

3.2.3 Discussions

We studied the effectiveness of the proposed feature selection using three different classification methods: CART, SVM and Naïve Bayes on 20 standard UCI datasets. Two performance measures are evaluated: classification accuracy and number of selected features.

Classification accuracy is the most common and simplest measure to evaluate a classifier. It is defined as the proportion of the total number of predictions that are correct. Furthermore, a good feature selection chooses a small feature subsets from the original features that is sufficient to predict the target label. The 10-fold cross validation procedure is applied to report the result figures.

The results in Table 2 show that the classification accuracy was noticeably enhanced by the proposed algorithm with all classifiers compared to that without feature selection. The best performance was where the accuracy achieved 100% with 7, 22, 3, and 3 features selected for the Wine, Soybean, Contact Lenses and Iris datasets, respectively, using SVM. Additionally, high classification accuracy is achieved with small feature subsets were Ionosphere, Soybean, Breast Cancer (WDBC), and Statlog (Australian).

It can be seen that the classification accuracies using SVM, CART, and Naïve Bayes significantly improved from 5% to 60% after applying the proposed algorithm with feature subsets for the Wine, Breast Cancer, Statlog (Australian), Soybean, Ionosphere, Heart and Lung cancer datasets. We also noted that Naïve Bayes yielded lower classification accuracy than SVM or CART.

In 97.70% of the cases, the proposed technique improved classification effectiveness and greatly reduced the number of features selected, thus increasing classification efficiency, for all of the classification methods. We actually achieved 100.00 %selection accuracy in four datasets with the proposed method.

Regarding the classification methods, SVM yielded the highest classification accuracy in 60 %of the datasets and yielded equal classification accuracy 10% while CART gave the highest accuracy in 30 %of the datasets.

As shown in table 3, the proposed algorithm based on SVM and CART outperformed the others for 8 out of 12 datasets and 7 out of 12 datasets, respectively. The SVM classifier achieved better results with the Wine, Soybean and Iris datasets by 1.60%, 12.00% and 17.85%, respectively, compared with recent research on feature selection by (Yang et al., 2010), and a 2.6% improvement with the Iris dataset compared with a study of Mohit, Verma, Katoch, Vanjare, and Omkar (2015).

Not only did the proposed algorithm reduced features from 13 to 7, 35 to 20, 34 to 8, 13 to 5 and 56 to 12 for the Wine, Soybean, Ionosphere, Heart and Lung cancer datasets, respectively, but also the classification accuracies improved by 12.35%, 17.64%, 7.14%, 26% and 60% when compared with the accuracy using full datasets. With the Soybean dataset, the proposed algorithm decreased the number of features from 35 to 20 and the classification accuracy using SVM was 100.00%, which is much higher compared to the others methods. Moreover, the proposed algorithm also decreased the number of attributes from 8 to 3 and 4 to 2 with the Abalone and Contact Lens datasets, respectively, and accuracy was again higher compared to the other methods.

The proposed algorithm based on a feature selection algorithm produced effective and small feature sets with higher classification accuracy on several different datasets because of the feature improvement step using a genetic algorithm that replaced the weakest features. The algorithm performed a more comprehensive search with a more valuable chance of finding the optimum solution. Our proposed algorithm was able to extract a more relevant and effective feature set from the original feature set by employing the genetic operations of selection, crossover and mutation operators to discover efficient and effective feature subsets.

CHAPTER 4

IMPROVING FLOATING SEARCH FEATURE SELECTION USING NICH GENETIC ALGORITHM

As aforementioned, genetic algorithms can be extended by niching techniques, and in recent years, a lot of research has been accomplished in this area. De Jong (1975) invented the niching concept (Mahfoud, 1995; Pedroso, 1996). A niche can be considered as a subspace in the environment that has various subpopulations in the investigation scope. The niche method generally employs a substitution approach when creating the new generation, which makes the individuals develop in their special search space. The niche technology maintains population diversity and allows a GA to investigate many peaks in parallel. Moreover, they avoid the GA from becoming captured in the local optimum of the search area. In this chapter, an attribute selection technique for classification employing a niching method to attain the optimum solution more closely is proposed.

4.1 The Proposed Method

We now discuss our approach to choose the best subset of size d of the total of D attributes. The inclusion step using MI as the criterion function (J) is executed to create a set of candidates for inclusion. In the exclusion step, a candidate attribute subset is used to create smaller subsets from the result of the inclusion step by removing one feature and re-evaluating them. A selection subset of size k+1 is generated and compared to the previously best subset of size k+1 from the inclusion part. If evaluation of the new subset is more qualified than the formerly chosen set, the exclusion step retains the better one and iterates to smaller subsets, or else the method goes back to the addition stage.

Our feature improvement step based on GA is incorporated after the exclusion step at each iteration. The chromosome structure consists of binary genes, corresponding to individual features. The value of 1 at the *i*th gene means that the *i*th feature is selected; otherwise it is 0.

The initial population is generated from the resulted subsets of size k+1 from the exclusion step by first removing the weakest features from the best subset resulting in a subset of size k. Each remaining feature is thus added to that subset generating the niched initial population for GA. The fitness function used in this study is MI. Then, a new population is created by selection, crossover and mutation operations. The procedure is terminated when the current attribute set reaches the size of D-2 features.

We now provide an illustrative example of how the proposed algorithm works and how it improves SFFS. Suppose that the first five attribute sets chosen by the SFS method at each size are {f1}, {f1, f4}, {f1, f4, f5}, {f1, f4, f5, f7} with the corresponding J values of 4.1, 6.2, 9.1 and 10.2, respectively, and the next iteration is to determine subsets with five features.

4.2 Step 1: Inclusion

An attribute is included to the variable subset. The SFS approach adds a feature to the subset up to a total of five: J(f1,f4,f5,f7,f6) = 13. Suppose that attribute f6 is selected using the SFS method and J for the 5th features is 14.

4.3 Step 2: Exclusion

A feature is removed from the feature subset. The SBS method is applied in this step by backtracking and conditionally removing one feature from the subset selected in Step 1 and returning an improved subset, e.g. (f1,f5,f6,f7) j value = 11, (f1,f4,f5,f7) j value = 9, (f1,f4,f7,f6) j value = 9.5, and (f4,f5,f7,f6) j value = 10. In this case, the best feature subset of size 4 is (f1, f5, f6, f7).

4.4 Step 3: Feature Improvement

The weakest feature is removed from the subset from of size k the previous step which is (f1, f5, f6, f7) by iteratively evaluating the smaller subsets: (f1, f5, f7), (f1, f5, f6), (f5, f6, f7) and (f1, f7, f6). In this case, we assume that the best performance subset of size 3 is (f5, f7, f6). Then, each feature is added to each subset of (f5, f7, f6) in order to find the best four feature subset, either (f5, f7, f6, f1), (f5, f7, f6, f2), (f5, f7, f6, f3), (f5, f7, f6, f4), (f5, f7, f6, f8), or (f5, f7, f6, f9.) Top n chromosomes is selected as the initial population for GA and passed through the crossover and mutation operations.

4.4.1 Crossover Operation

Once a pair of chromosomes has been chosen, crossover can take place to generate child chromosomes. A crossover location is unexpectedly chosen from two randomly selected individuals (parents). This point happens between two bits and separates each individual into left and right sections. Crossover then exchanges the left (or the right) segment of the two individuals therefore:

Parent A – (f5, f7, f6, f2)

0	1	0	0	1	1	1	0	0	0

Parent B – (f5, f7, f6, f1)

1	0	0	0	1	1	1	0	0	0

Suppose the crossover point randomly occurs after the sixth bit, then each new child receives one half of each parent’s bits:

Offspring1 – (f1, f5, f7, f6)

0	1	0	0	1	1	1	0	0	0

Offspring2 – (f2, f5, f7, f6)

1	0	0	0	1		1	1	0	0	0

This algorithm continues to select parental chromosomes to apply the crossover operation. Child chromosomes may have one bit more than the current size of k features subset. In this case, a random bit is automatically flipped to preserve the size of the chromosome (i.e. current feature set size).

4.4.2 Mutation Operation

The mutation operation is employed to all of the offspring chromosomes from the crossover step. Mutation performs at the bit level by unexpectedly flipping bits in the new chromosome within the current population (turning a ‘0’ into ‘1’, and vice versa).

Offspring1 – (f5, f7, f6, f1)

1	0	0	0	1	1	1	0	0	0

After mutation – (f5, f7, f6, f2)

0	1	0	0	1	1	1	0	0	0

After all child chromosomes have passed through the mutation operator, the resultant chromosomes are evaluated by the fitness function. After this, we can discover the best performing features subset, which is (f5, f7, f6, f2). We assume that J({f5, f7, f6}) = 8.35, and that J({f5,f7, f6, f2}) = 12, which is greater than the prior largest value for four features, J = 11 Therefore, the best 4-feature subset becomes {f5, f7, f6, f2} with J = 12, whereas the best 3-feature subset remains {f1, f4, f5} since J({f1, f4, f5}) = 9.1 > J({f5, f7, f6}) = 8.35

The improvement step helps discover subsets not discoverable by the greedy nature of SFFS. From the above example, the SFFS algorithm is not able to produce this best 4-feature four subset because it cannot backtrack to the set {f5, f7, f6}, thus could not add feature f2 to subset {f5, f7, f6}. Once the mutation step has ended, the algorithm ensures that the results in the feature size subsets correspond to the k-feature subset during each iteration. In the case where the results are an equal k-feature subset, the algorithm will swap bit string values (from 0 to 1 or 1 to 0) at random positions.

The example above demonstrates the advantage of our proposed algorithm. The algorithm substitutes the powerless attribute (feature f1 in our example) in the feature set {f1,f5,f7,f6} with feature 2, which results in a new set of four features {f5, f7, f6, f2} which has a larger J value. Therefore, the search approach of our proposed method is more complete than the SFFS algorithm, so it is more effective.

4.5 Step 4: Deterministic Crowding Step

After all of the children have already mutated, the basis of crowding is to employ tournament selection to the parent-child pairs with a hamming distance measure. In sample-based crowding, we determine which of the two will remain in the population (replacement phase) using tournament selection decided by the smallest hamming distance value between the parent and the child.

Parent A – (f5, f7, f6, f2)

0	1	0	0	1	1	1	0	0	0

Parent B – (f5, f7, f6, f1)

1	0	0	0	1	1	1	0	0	0

Child A – (f5, f7, f4, f2)

0	1	0	1	1	0	1	0	0	0

Child B – (f5, f7, f6, f2)

0	1	0	1	1	0	1	0	0	0

Hamming Distance (parent a, child a) = 1.2;
Hamming Distance (parent b, child b) = 1.1;
Hamming Distance (parent a, child a) = 1.3;
Hamming Distance (parent b, child b) = 1.5;
Fitness value (parent a) = 9.3
Fitness value (parent b) = 8.5
Fitness value (child a) = 9.0
Fitness value (child b) = 7.4

In this case, the total distance of ([d (parent a, child a) + d (parent b, child b)] is 2.3, which is less than [d(parent a, child b)+d(parent b, child a)]) 2.8; hence, the fitness of child a is less than parent a. The algorithm replaces parent with a child from the current population pool. Consequently, a new niche will occur and yield more diversity in the population. Finally, new feature subset (f5, f6, f2, f8), which has a larger J value than (f5, f7, f6, f2) is discovered. For this reason, the search approach of our proposed method is more comprehensive than the SFFS approach, so it is more effective.

4.6 Step 5: Terminating Condition

After each iteration, the selection / crossover / mutation cycle continues until all possible combinations of chromosomes in the population have been evaluated. The better the fitness value, the greater the probability of that chromosome being chosen for replication. This generational procedure is reiterated until a pre-determined termination condition has been reached. We terminate the algorithm when the current feature set accomplishes d < D attributes, where D is the total number of attributes in the dataset. The pseudo-code is depicted in Figure 3.

A fitness function is commonly needed in GAs to evaluate a candidate chromosome of an individual to assess whether the latter should survive or not. At each iteration, calculation of the fitness function is processed repeatedly, which, because of its simplicity, is a fast process, although it still impacts performance. In our model, we use the MI criterion as a fitness function. Basically, it reckons the amount of an information feature set in a group of variables for the sake of predicting the dependent data. In addition, the fitness function to be calculated includes the calculation of the classification rate, which requires a classifier.

```
        Input: $Y_m$ is a feature set, m is a predefined number of selected features, J is a criterion function. $P_c$ is probability of crossover, $P_m$ is probability of mutation, Population is set of individuals, max_generation is the maximum number of generations, and Fitness is a function which determines quality of individuals.
        Output: The best solution in all generation.
  (1)   Inclusion
           Initialize: $Y_0 = \{\emptyset\}; m = 0$
        Find the best feature and update $Y_m$
        $x^{+} = \underset{x/\in Ym}{\arg\max}\,[J(Y_m - x)]$

        $Y_m = Y_m + x^{+}; m = m + 1$
   (2) Conditional Exclusion
        Find the worst feature
         $x^{-} = \underset{x/\in Ym}{\arg\max}\,[J(Y_m - x)]$
        If $J(Y_m - x^-) > J(Y_m)$ then
      $Y_m+1 = Y_m - x$;
  Go to Step 3 Else Go to Step1
  (3) Feature Improvement Step.
        Repeat
          population ← SBFS feature subsets $Y_m$
          generation = 0;
          loop for i from 1 to size(Population) do
              p1 ← selection (Population, Fitness)
              p2 ← selection (Population, Fitness)
              child ← crossover (p1,p2) with pc and check feasibility of n element
              child ← mutate(child)  with pm and check feasibility of n element
              d1 ←hamming_distance(p1, child1)
              d2 ←hamming_distance(p2, child2)
              d3 ←hamming_distance(p2, child1)
              d4 ←hamming_distance(p1, child2)
                If (d1 + d2) <= (d3+d4)
                    If (Fitness(child2) >Fitness(p2) then replace p2 with child2;
                    If (Fitness(child1) > Fitness(p1) then replace p1 with child1;
               ELSE
                   If (Fitness(child1) >Fitness(p2) then replace p2 with child1;
                   If (Fitness(child2) > Fitness(p1) then replace p1 with child2;
              End if
              Fitness(child);
           generation = generation +1
        until generation < max_generation
      m = m + 1
      return the best individual solution $Y_m$
```

Figure 4.1 Pseudo-Code of the Proposed Algorithm

4.7 Hamming Distance

The hamming distance (Norouzi, Fleet and Salakhutdinov, 2012) is the number of digit positions in which the matching digits of two binary words of the same range are dissimilar. The approach can be broaden to other notation systems. For instance, the Hamming distance between 1011101 and 1001001 is two.

Here is another example of string value:

if

s1 = ‘CATS’

s2 = ‘DOGS’

then

distance = 3

For this reason, it is necessary to change all three substitutions, thereby transforming from s1 to s2.

cats =>dats

dats => dots (replace 'o' for 'a')

dots => dogs (replace 'g' for 't')

4.8 Experimental Results

4.8.1 Data Sets

Data used in the experiments are 20 standard data sets with various sizes from the UCI machine learning repository. Details of the data sets are shown in Table 4.1

Table 4.1 Data Used in Experiment

Dataset	Feature Characteristics	No. of Instances	No. of Features	No. of Classes
Wine	Integer	178	13	3
Breast Cancer (Original)	Integer	699	10	2
Breast Cancer (WDBC)	Real	569	32	2
Breast Cancer (WPBC)	Real	198	34	2
Iris	Real	150	4	3
Pima-Indian diabetes	Integer, Real	768	8	2
Abalone	Categorical, Integer, Real	4,177	8	3
Dermatology	Categorical, Real	366	34	6
Heart	Categorical, Real	270	13	2
German (Credit card)	Categorical, Integer	1,000	20	2
Lung cancer	Integer	32	56	3
Soy bean	Integer	307	35	4
Spambase	Integer, Real	4,601	57	2
Glass Identification	Real	214	10	7
Teaching Assistant	Categorical, Integer	151	5	3
Contact Lens	Categorical	24	4	3
Sonar	Real	208	60	2
Statlog (Australian)	Categorical, Integer, Real	690	14	2
Ionosphere	Integer, Real	351	34	2
Image Segmentation	Real	2,310	19	7

4.8.2 Experiment Evaluation

Three classification modeling techniques are used in the experiments which consist of Classification and Regression Tree (CART), Support Vector Machine (SVM), and Naïve Bayes. Training and testing data is used as provided in the datasets. For those not providing separate testing data, a 10-fold cross validation is applied. To evaluate a feature subset, MI is applied as the criterion function.

Table 4.2 Classification Effectiveness

Dataset	**Original Datasets**	**No. of Attributes**	**PM* CART**	**PM* SVM**	**PM* Naïve Bayes**
Wine	89.87	13	100.00 (7)	100.00(7)	97.14(7)
Breast Cancer (Original)	93.13	10	97.82(5)	97.85(5)	95.68(5)
Breast Cancer (WDBC)	92.23	32	95.499(9)	96.13(9)	91.00(9)
Breast Cancer (WPBC)	72.00	34	83.00(6)	86.26(6)	80.00(6)
Iris	94.00	4	98.44(3)	100(3)	95.68(3)
Pima- Indian Diabetes	72.51	8	73.178(4)	76.04(4)	71.89(4)
Abalone	49.07	8	52.00(3)	58.00(3)	49.26(3)
Dermatology	95.08	34	98.83(26)	98.85(26)	94.15(26)
Heart	76.67	13	80.00(6)	81.11(6)	79.00(6)
German	68.50	20	73.50(6)	71.50(6)	69.00(6)
Lung cancer	59.67	56	75.00(21)	83.33(21)	72.00(21)

Table 4.2 (Continued)

Dataset	**Original Datasets**	**No. of Attributes**	**PM* CART**	**PM* SVM**	**PM* Naïve Bayes**
Soy bean	85.00	35	100.00(22)	100.00(22)	98.28(22)
Spambase	93.26	57	96.00(26)	92.00(26)	91.76(26)
Glass Identification	62.00	10	63.13(5)	66.67(5)	65.00(5)
Teaching Assistant	54.92	5	58.03(2)	61.86(2)	62.00(2)
Contact Lens	76.00	4	80.00(2)	100.00(2)	85.00(2)
Sonar	69.50	60	76.86(7)	62.98(7)	67.00(7)
Statlog (Australian)	65.45	14	74.30(7)	79.04(7)	75.24(7)
Ionosphere	84.00	34	88.00(5)	90.62(5)	90.10(5)
Image Segmentation	85.00	19	90.95(14)	88.57(14)	85.10(14)

*Proposed Method

Note. Results Expressed as Percentages

4.8.3 Comparisons with Other Approaches

Table 4.3 Comparison with Other Previously Report Method on Common Datasets

Dataset	**PM***	**PM***	**A**	**B**	**C**	**D**	**E**	**F**	**G**
	CART	**SVM**							
Breast Cancer (Original)	**97.8**	**97.9**	-	97.4	94.4	96.5	-	-	94.8
Breast Cancer (WDBC)	**95.5**	**96.1**	95.4	-	-	-	-	-	93.0
Iris	**98.4**	**100**	97.3		-	97.3	96.7	96.6	-
Pima Indian Diabetes	73.2	76.0	73.8	79.9	76.0	73.2	-	-	-
German	73.5	71.5	72.6	76.2	-	74.5	-	69.9	-
Soybean	**100**	**100**	-	88.3	-	97.8	-	-	-
Wine	**100**	**100**	-	-	91.6	98.3	-	-	-
Heart	80.0	81.1	-	-	61.1	84.8	87.1	-	-
Sonar	76.8	62.9	-	-	83.7	-	-	-	-
Abalone	52.0	**58.0**	54.5	-	-	-	30.0	25.7	-
Dermatology	**98.8**	**98.9**	-	-	-	95.4	-	-	-
Contact Lenses	**76.0**	**100**	-	-	-	-	75.0	-	-

*Proposed Method

Source: Results Expressed as Percentages, A: Liu et al., (2004); B: Chotirat Ann Ratanamahatana and Gunopulos (2003); C: Anwar et al., (2014); D: Yang et al., (2010); E: Gupta and Ghafir (2012); F: Tsai et al., (2011); G: Lavanya and Rani (2011).

4.8.4 Discussion

The proposed technique employs the crowding method, which is an additional algorithm continuing from the genetic method to the filter-based feature selection method proposed in Chapter 3. This technique replaces similar individuals in a population and thus yields more diversity, which assures the slowing down of the premature problem of a traditional GA. The algorithm employs mutual information and hamming distance as a feature subset evaluation function by which an individual with higher fitness will survive and stay in the next generation. The proposed technique was evaluated using 20 standard datasets from the UCI repository with three different classification methods, and from the results, was found to enhance the accuracy of feature selection. In this study, the proposed method was only applied to classification problems in which a class attribute guides the search for features that are related or relevant to it. Moreover, it performed significantly well in comparison with other previously reported studies.

CHAPTER 5

CONCLUSION AND RECOMMENDATION FOR FUTURE

5.1 Conclusion

Attribute selection is critical to the performance of classification, and thus a feature selection algorithm that improves the performance of SFFS by incorporating a feature improvement step based on a genetic algorithm was proposed. This step helps to discover important subsets that are not possible using SFFS alone. The algorithm employs mutual information as a feature subset evaluation function. The proposed technique was evaluated using 20 standard datasets from the UCI repository with three different classification methods. To enhance the performance, preprocessing was applied and many definitions of feature relevance, feature selection, and optimal feature subsets were combined. The common process of attribute selection was presented with subset generation and assessment, and terminating criteria. There are three mechanisms of feature selection methods, namely filter, wrapper and hybrid, all of which are described in detail in the literature evaluation section. The drawbacks, benefits, and characteristics of feature selection are reviewed, which helps to understand all aspects of dimensional categorizations of feature selection algorithms and the history of their development, to yield insight into future challenges and research directions.

The contributions of this work are as follows. This research was based on filter-based feature selection because it has been widely accepted that it yields moderate accuracy with low computation cost. Our proposed algorithm enhanced the most up-to-date SFFS algorithm by adding an additional search step which incorporated a genetic algorithm at each sequence to check whether it can enhance the current feature subset. Lastly, our method contributes closer to the optimum solution for many selected subsets and its execution required considerably less time compared to previously reported methods. The experimental outcomes show that the proposed attribute selection technique significantly improved classification accuracy and resulted in a much smaller feature subset, thus improved efficiency and outperformed other feature selection methods, especially when compared with other feature selection algorithms, and also yielded good results with a SVM classifier.

5.2 Future Work

Even though we applied a genetic algorithm to introduce more candidate feature subsets and MI to capture important information of an individual chromosome for the GA, there is no guarantee of achieving the highest classification results. Moreover, many factors are required to move closer to the optimal solution, some of which could be further studied.

The number of criterion functions could be more than two. We could apply a number of simple criterion functions instead of two complicated criterion ones; however, they must be consistent and should be suitable for the problem. Moreover, the number of candidate features must decrease continuously from the first criterion function until the best one is obtained from the last one.

Fitness sharing of niching techniques is an alternative solution to optimize an objective function. GAs are efficient algorithms for solving a wide range of optimization problems, but there is no guarantee that the results will end up close to the local or global optimum solution. Nevertheless, a simple way to increase the probability of finding the global optimum is to adjust the probability of crossover, mutation, and the generation parameter for the decision-maker to select from the alternative solution. Moreover, there is another approach called the niching genetic algorithm for which its concept in brief is a repeated search of the same region of space. The benefit of this is to aid the formation of stable subpopulations in the search area so that each subpopulation is formed and located closer to the global or local optimum solutions. Although there are many publications on niching research, it has not yet been brought together with SFFS.

In our experiments, preprocessing was applied through the first steps in the form of discretization. Nevertheless, there are several ways to discretize that affect accuracy. The proposed method includes the sorting of the dataset and selecting only duplicate values for cutting point binning, and during this step, the number of discrete values to represent each bin are found. The range associated with an interval is divided into k interval depending on the number of replicate values. However, for other datasets, there could be different preprocessing steps; for example, instance-wise normalization and then feature-wise normalization, or vice versa. Further study on how preprocessing should be achieved and how it affects training performance and classification accuracy needs to be carried out.

To summarize, the study targeted understanding of the importance of the feature selection process and various ways of performing feature selection. The majority of the research focused on a labeled dataset, for which a genetic algorithm was introduced in supervised classification models to assist feature selection. This helped by removing irrelevant features and creating a subset of prominent features. Lastly, mutual information is a measure to capture significant feature subsets incorporated with the SFFS process.

BIBLIOGRAPHY

BIBLIOGRAPHY

Al-Ibrahim, Ali. 2011. Discretization of Continuous Attributes in Supervised Learning algorithms. **The Research Bulletin of Jordan ACM.** 2 (October): 158-166.

Anwar, Hina, Qamar, Usman and Qureshi, Abdul Wahab Muzaffar. 2014. Global Optimization Ensemble Model for Classification Methods. **The Scientific World Journal**. 2014 (April): 1-9.

Battiti, Roberto. 1994. Using Mutual Information for Selecting Features in Supervised Neural Net Learning. **IEEE Transactions on Neural Networks**. 5 (July): 537-550.

Bins, Jose and Draper, Bruce A. 2001. Feature Selection from Huge Feature Sets. In **Proceedings Eighth IEEE International Conference on Computer Vision.** Vancouver, BC: IEEE. Pp. 159-165.

Boulle, Marc. 2004. A Statistical Discretization Method of Continuous Attributes. **Machine learning**. 55 (April): 53-69.

Breiman, Leo, Friedman, Jerome H., Olshen, Richard A. and Stone, Charles J. 1984. Classification Algorithms and Regression Trees. In **Classification and Regression Trees.** Leo Breiman, Jerome H Friedman, Richard A Olshen and Charles J Stone, eds. Monterey, CA: Chapman and Hall/CRC. Pp. 246-280.

Brill, F. Z., Brown, D. E. and Martin, W. N. 1992. Fast Generic Selection of Features for Neural Network Classifiers. **IEEE Transactions: Neural Networks.** 3 (March): 324-328.

Cedeño, Walter and Vemuri, V. Rao. 1999. Analysis of Speciation and Niching in The Multi-Niche Crowding GA. **Theoretical Computer Science.** 229 (November): 177-197.

Chandrashekar, G. and Sahin, F. 2014. A Survey on Feature Selection Methods. **Computers & Electrical Engineering**. 40 (1): 16-28.

Chee, Jennifer D. 2015. **Pearson's Product-Moment Correlation: Sample Analysis.** Retrieved December 1, 2017 from https://www.researchgate.net/publication/277324930_Pearson%27s_Product-Moment_Correlation_Sample_Analysis

Chen, Chiu-Hung, Liu, Tung-Kuan and Chou, Jyh-Horng. 2014. A Novel Crowding Genetic Algorithm and its Applications to Manufacturing Robots. **IEEE Transactions on Industrial Informatics**. 10 (April): 1705-1716.

Chotirat Ann Ratanamahatana and Gunopulos, Dimitrios. 2002. Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection. In **Proceedings of Workshop on Data Cleaning and Preprocessing at IEEE International Conference on Data Mining.** Maebashi, Japan: IEEE. Pp. 1-10.

Cortes, Corinna and Vapnik, Vladimir. 1995. Support-Vector Networks. **Machine Learning.** 20 (September): 273-297.

Dangi, Nidhi and Prashant Ahlawat. 2015. A Naive Bayes Data Mining Approach for Classification of Cancer Dataset. **International Journal of Enhanced Research in Management & Computer Applications.** 4 (July): 31-36.

Darwen, Paul and Yao, Xin. 1996. Every Niching Method has Its Niche: Fitness Sharing and Implicit Sharing Compared. In **The 4th International Conference on Parallel Problem Solving from Nature.** Berlin, Germany: Springer. Pp. 398-407.

Dash, Manoranjan and Liu, Huan. 1997. Feature Selection for Classification. **Intelligent Data Analysis.** 1(1-4): 131-156.

De Jong, K. A. 1975. **An Analysis of the Behavior of a Class of Genetic Adaptative Systems.** Doctoral dissertation, University of Michigan.

Dougherty, James, Kohavi, Ron and Sahami, Mehran. 1995. Supervised and Unsupervised Discretization of Continuous Features. In **Machine Learning: Proceedings of the Twelfth International Conference.** Tahoe City, CA: Elsevier. Pp. 194-202.

Drchal, Jan, Šnorek, Miroslav and Kordík, Pavel. 2006. Maintaining Diversity in Population of Evolved Models. In **Proceedings of 40th Spring International Conference MOSIS 06, Modelling and Simulation of Systems.** Prerov, Czech Republic: MARQ. Pp. 113–120.

Goldberg, David E. 1989. **Genetic Algorithms in Search, Optimization and Machine Learning.** Boston, MA: Addison-Wesley Longman Publishing.

Goldberg, David E. and Holland, John H. 1988. Genetic Algorithms and Machine Learning. **Machine Learning.** 3 (October): 95-99.

Grande, Javier, Suárez, María del Rosario and Villar, José Ramón. 2007. A Feature Selection Method Using a Fuzzy Mutual Information Measure. In **Innovations in Hybrid Intelligent Systems**. Emilio Corchado, Juan M. Corchado and Ajith Abraham, eds. New York: Springer-Verlag Berlin, Heidelberg. Pp. 56-63.

Gupta, Deepti and Ghafir, Shabina. 2012. An Overview of Methods Maintaining Diversity in Genetic Algorithms. **International Journal of Emerging Technology and Advanced Engineering**. 2 (May): 56-60.

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

He, He, Daume, Hal, III and Eisner, Jason M. 2014. Learning to Search in Branch and Bound Algorithms. In **Advances in Neural Information Processing Systems 27 (NIPS 2014).** Montréal, Canada: Neural Information Processing Systems (NIPS). Pp. 3293-3301.

Hssina, Badr, Merbouha, Abdelkarim, Ezzikouri, Hanane and Erritali, Mohammed. 2014. A Comparative Study of Decision Tree ID3 and C4.5. **International Journal of Advanced Computer Science and Applications.** 4 (February): 13-19.

Hsu, Chih-Wei, Chang, Chih-Chung and Lin, Chih-Jen. 2016. A Practical Guide to Support Vector Classification. **Technical Report of Department of Computer Science, National Taiwan University, Taipei 106, Taiwan**. Pp. 1-16. Retrieved December 1, 2017 from https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Huang, Cheng-Lung, and Wang, Chieh-Jen. 2006. A GA-based feature selection and parameters optimizationfor support vector machines. **Expert Systems with Applications.** 31 (August): 231-240.

Jitwadtt Chaiyakarn. 2013. **A Filter-Based Feature Selection Using Two Criterion Functions and Evolutionary Fuzzification.** Doctoral dissertation**,** National Institute of Development Administration.

Juan Tapia Farias. 2017. **Features Selection Methods.** Retrieved from https://jtapiafarias.wordpress.com/about/

Kaya, Y. and Uyar, M. 2011. **A Novel Crossover Operator for Genetic Algorithms: Ring Crossover**. Retrieved from https://arxiv.org/ftp/arxiv/papers/1105/1105.0355.pdf

Karegowda, Asha Gowda, Jayaram, M. A. and Manjunath, A. S.. 2011. Feature Subset Selection using Cascaded GA and CFS: A Filter Approach in Supervised Learning. **International Journal of Computer Applications.** 23 (June): 1-10.

Kavitha, K. K., Kangaiammal, A. and Satheesh, K. 2015. Analysis on Classification Techniques in Mammographic Mass Data Set. **International Journal of Engineering Research and Applications.** 5 (July): 32-35.

Kira, Kenji and Rendell, Larry A. 1992. The Feature Selection Problem: Traditional Methods and a New Algorithm. In **Proceedings of the Tenth National Conference On Artificial Intelligence.** San Jose, CA: AAAI Press. Pp. 129-134.

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

Kumar, Vipin and Minz, Sonajharia. 2014. Feature Selection: A Literature Review. **Smart Computing Review**. 4 (June): 211-229.

Lavanya, D. and Usha Rani, K. 2011. Analysis of Feature Selection with Classification: Breast Cancer Datasets. **Indian Journal of Computer Science and Engineering.** 2 (October-November): 756-763.

Lee, M. C. 2009. Using Support Vector Machine with a Hybrid Feature Selection Method to the Stock Trend Prediction. **Expert Systems with Applications**. 36 (8): 10896-10904.

Likas, A., Blekas, K. and Stafylopatis, A. 1996. Parallel Recombinative Reinforcement Learning: A Genetic Approach. **Journal of Intelligent Systems.** 6 (June): 145-170.

Liu, Huan, Motoda, Hiroshi and Yu, Lei. 2004. A Selective Sampling Approach to Active Feature Selection. **Artificial Intelligence.** 159 (November): 49-74.

MacQueen, J. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In **Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability**. University of California, USA: UC Press. Pp. 281-297.

Mahfoud, Samir W. 1995. **Niching Methods for Genetic Algorithms.** Doctoral dissertation, University of Illinois at Urbana-Champaign.

Mangano, Salvatore. 2008. Genetic Algorithms. In **Soft Computing Techniques and its Applications in Electrical Engineering, Studies in Computational Intelligence (SCI) 103.** D. K. Chaturvedi, eds. Warsaw, Poland: Springer-Verlag Berlin Heidelberg. Pp. 363-381.

McCune, Bruce, Grace, James B. and Urban, Dean L. 2002. **Analysis of Ecological Communities.** Beach, Or: MjM Software Design.

Mengshoel, Ole J. and Goldberg, David E. 2008. The Crowding Approach to Niching in Genetic Algorithms. **Evolutionary Computation.** 16 (September): 315-354.

Mengshoel, Ole J., Galán, Severino F. and de Dios, Antonio. 2014. Adaptive Generalized Crowding for Genetic Algorithms. **Information Sciences.** 258 (February): 140-159.

Mohit, Rohit, Verma, Ranjan, Katoch, Sameeksha, Vanjare, Ashoka and Omkar, S. N. 2015. Classification of Complex UCI Datasets Using Machine Learning Algorithms Using Hadoop. **International Journal of Computer Science and Software Engineering.** 4 (July): 190-198.

Molina, L. C., Belanche, L. and Nebot, A. 2002. Feature Selection Algorithms: A Survey and Experimental Evaluation. In **Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on.** Maebashi City, Japan: IEEE. Pp. 306-313.

Navot, Amir. 2006. **On the Role of Feature Selection in Machine Learning.** Doctoral dissertation, Hebrew University.

Newcastle University. 2017. **Roulette Wheel Selection**. Retrieved from http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php

Norouzi, Mohammad, Fleet, David J. and Salakhutdinov, Ruslan R. 2012. Hamming Distance Metric Learning. In **Advances in Neural Information Processing Systems 25.** Red Hock, NY: Curran Associates. Pp. 1061-1069.

Oh, Il-Seok, Lee, Jin-Seon and Moon, Byung-Ro. 2004. Hybrid Genetic Algorithms for Feature Selection. **IEEE Transactions: Pattern Analysis and Machine Intelligence.** 26 (November): 1424-1437.

Pedroso, João Pedro. 1996. Niche Search: An Evolutionary Algorithm for Global Optimisation. In **International Conference on Evolutionary Computation - The 4th International Conference on Parallel Problem Solving from Nature.** Berlin, Germany: Springer. Pp. 430-440.

Petrowski, A. 1996. A Clearing Procedure as A Niching Method for Genetic. In **Evolutionary Computation, 1996., Proceedings of IEEE International Conference on.** Nagoya, Japan: IEEE. Pp. 798–803.

Pudil, P., Novovičová, J. and Kittler, J. 1994. Floating Search Methods in Feature Selection. **Pattern Recognition Letters**. 15 (November): 1119-1125.

Quinlan, J. R. 1986. Induction of Decision Trees. **Machine Learning.** 1 (March): 81-106.

Saedsayad. 2017. **Decision Tree-Classification**. Retrieved from http://www.saedsayad.com/decision_tree.htm

Saedsayad. 2017. **Equal Frequency Discretization**. Retrieved from http://www.saedsayad.com/unsupervised_binning.html

Sáez, José A., Galar, Mikel, Luengo, Julián and Herrera, Francisco. 2013. Tackling the Problem of Classification with Noisy Data Using Multiple Classifier Systems: Analysis of The Performance and Robustness. **Information Sciences.** 247 (October): 1-20.

Saha, Sriparna. 2017. Sriparna Saha. Retrieved from https://www.researchgate.net/profile/Sriparna_Saha

Sanchez-Marono, Noelia, Alonso-Betanzos, Amparo and Castillo, Enrique F. 2005. A New Wrapper Method for Feature Subset Selection. In **ESANN'2005 Proceeding - European Sysposium on Artificial Neural Networks.** Bruges, Belgium: D-Side Publication. Pp. 515-520.

Sapp, Marty, Obiakor, Festus E., Gregas, Amanda J. and Scholze, Steffanie. 2007. Mahalanobis Distance: A Multivariate Measure of Effect in Hypnosis Research. **Sleep and Hypnosis.** 9 (2): 67-70.

Somol, P., Pudil, P., Novovičová, J. and Paclík, P. 1999. Adaptive Floating Search Methods in Feature Selection. **Pattern Recognition Letters.** 20 (November): 1157-1163.

Somol, Petr, Novovičová,Jana and Pudil, Pavel. 2006. Flexible-Hybrid Sequential Floating Search in Statistical Feature Selection. In **Structural, Syntactic, and Statistical Pattern Recognition**. Dit-Yan Yeung, James T. Kwok, Ana Fred, Fabio Roli and Dick de Ridder, eds. Hong Kong, China: Springer-Verlag. Pp. 632-639.

Songyot Nakariyakul. 2009. A Review of Suboptimal Branch and Bound Algorithms. In **International Conference on Knowledge Discovery.** Manila, Philippines: IACSIT Press. Pp. 566-570.

Songyot Nakariyakul and Casasent, David P. 2008. Improved Forward Floating Selection Algorithm for Feature Subset Selection. In **International Conference on Wavelet Analysis and Pattern Recognition.** Hong Kong: IEEE. Pp. 793-798.

Suárez, María del Rosario, Villar, José Ramón and Grande, Javier. 2010. A Feature Selection Method Using a Fuzzy Mutual Information Measure. **International Journal of Reasoning-based Intelligent Systems.** 2 (2): 133-141.

Tang, J., Alelyani, S. and Liu, H. 2014. Feature Selection for Classification: A Review. **Data Classification: Algorithms and Applications**, 37 (1): 96-111.

Tsai, Cheng-Jung, Lee, Chien-I. and Yang, Wei-Pang. 2008. A Discretization Algorithm Based On Class-Attribute Contingency Coefficient. **Information Sciences.** 178 (Febuary): 714-731.

Tsai, Chih-Fong, Lin, Wei-Yang, Hong, Zhen-Fu and Hsieh, Chung-Yang. 2011. Distance-Based Features in Pattern Classification. **EURASIP Journal on Advances in Signal Processing.** 2011 (September): 1-11.

Tutorials Point Simply Easy Learning. 2017. **Genetic Algorithms - Parent Selection**. Retrieved from https://www.tutorialspoint.com/genetic_algorithms/ genetic_algorithms_parent_selection.htm

Umbarkar, A. J. and Sheth, P. D. 2015. Crossover Operators in Genetic Algorithms: A Reviwe. **ICTACT Journal on Soft Computing**. 6 (1): 1083-1092.

Wu, Xindong and Zhu, Xingquan. 2008. Mining with Noise Knowledge: Error-Aware Data Mining. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans.** 38 (June): 917 - 932.

Yang, Tao, Cao, Longbing and Zhang, Chengqi. 2010. A Novel Prototype Reduction Method for The K-Nearest Neighbor Algorithm with K≥ 1. In **Advances in Knowledge Discovery and Data Mining**. Mohammed J. Zaki, Jeffrey Xu Yu, B. Ravindran and Vikram Pudi, eds. Hyderabad, India: Springer-Verlag, Berlin Heidelberg. Pp. 89-100.

Ye, Feng, Qi, Weimin and Xiao, Jie. 2011. Research of Niching Genetic Algorithms for Optimization in Electromagnetics. **Procedia Engineering.** 16 (2011): 383-389.

Zhang, Hongbin and Sun, Guangyu. 2002. Feature Selection Using Tabu Search Method. **Pattern Recognition**. 35 (March): 701-711.

BIOGRAPHY

NAME Miss Kanyanut Homsapaya

ACADEMIC BACKGROUND Bachelor’s Degree with a major in Computer Science from Srinakarinwirot University, Bangkok, Thailand in 2004 and a Master’s Degree in Computer Science at National Institute of Development Administration, Bangkok, Thailand in 2007

PRESENT POSITION Senior Technical Consultant of Tectura Corp in 2004-2005
Senior Business Analyst at DTAC Co. Ltd in 2007-2013