TEXT FEATURE SELECTION USING ENHANCED BINARY BAT ALGORITHM

AISHA ADEL AHMED AL-HAJJANA

THESIS SUBMITTED IN FULFILMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2019

PILIHAN CIRI TEKS BERDASARKAN ALGORITMA KELAWAR BINARI DIPERTINGKAT

AISHA ADEL AHMED AL-HAJJANA

TESIS YANG DIKEMUKAKAN UNTUK MEMPEROLEHI
IJAZAH DOKTOR FALSAFAH

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2019

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

05 March 2019

AISHA ADEL AHMED AL-HAJJANA
P75140

ACKNOWLEDGEMENT

First and foremost praise be to Almighty Allah for all his blessings for giving me patience and good health throughout the duration of this PhD research.

I am very fortunate to have Associate Professor Dr. Nazlia Omar as a research supervisor. Also, I would like to express my high appreciation to my co-supervisor Prof. Salwani Abdullah.

This research project would not have been completed without the support of many people. Most notably, I would like to express my sincerest appreciation and gratitude to my supervisors Assoc. Prof. Dr. Nazlia Omar and Prof. Salwani Abdullah for their generous and continuous support; for their excellent advice and guidance throughout this research and for all the time, cooperation and effort they have dedicated to the supervision of this research. I would like again to thank them for their valuable suggestions, corrections and unlimited encouragement throughout the different stages of this research. I am also grateful to Dr. Mohammed Albared for his help and his valuable suggestions regarding this work.

I would express, as well, my deepest thanks to my examiners Assoc. Prof. Dr. Zalinda Othman and Prof. Naomie Salim. Thank you for your valuable comments and suggestions that have improved the contents of this thesis.

I would also like to express my high appreciation, my love and gratitude to my family; my dear parents, my brothers, my sister, my husband and my daughters for their unlimited support and patience during the different stages of this work. Also, many thanks to all my friends who helped and supported me throughout this research.

Last but not least, I am very grateful to UKM and to all members and staffs in the Faculty of Information Science and Technology for their help, friendship, and for creating a pleasant working environment throughout my studies at UKM.

ABSTRAK

Memandangkan banyak data bertulis yang dijana dan dikongsi di internet seperti laporan berita, artikel, tweet dan ulasan produk, keperluan untuk Pilihan Ciri Teks (PCT) yang berkesan menjadi semakin penting. Hal demikian dikatakan sangat mencabar berikutan data tersebut mempunyai nilai dimensi yang tinggi. Kaedah PCT terkini dikatakan tidak mementingkan kecenderungan ciri teks yang mana menyebabkan kurangnya kualiti set ciri teks yang dipilih serta mempengaruhi prestasi klasifikasinya. Kaedah yang lain pula bergantung kepada algoritma meta-heuristik berasaskan populasi, yang akan meningkatkan mutu set ciri teks yang dipilih dan hasil klasifikasinya. Walau bagaimanapun, jenis kaedah ini bergantung kepada pengelas dan menghasilkan bilangan ciri yang lebih tinggi. Di samping itu, algoritma ini didedahkan kepada penumpuan pramatang yang dipengaruhi oleh kekurangan kepelbagaian populasi. Selain itu, prestasi meta-heuristik kurang cekap apabila menangani masalah berdimensi tinggi, manakala kepelbagaian populasi perlu dikawal semasa pengoptimumannya. Untuk mengatasi masalah tersebut, kajian ini adalah bertujuan untuk membangunkan satu kaedah berasaskan Algoritma Kelawar Binari (AKB) bagi meningkatkan kecekapan PCT. Untuk permulaan, teori set secara kasar disesuaikan dan digunakan untuk menilai penyelesaian yang dihasilkan oleh AKB. Kaedah yang dicadangkan ini dibandingkan dengan versi pembalut (wrapper) AKB berdasarkan kaedah PCT. Kemudian, versi sampel Latin Hypercube Sampling (LHS) dicadangkan untuk memberi nilai awal kepada populasi yang pelbagai. Kaedah yang dicadangkan dibandingkan dengan nilai permulaan secara rawak dari segi prestasi kaedah PCT semasa pengoptimuman dan hasil klasifikasinya. Eksperimen menunjukkan bahawa kaedah permulaan yang dicadangkan mampu meningkatkan kepelbagaian populasi awal dan memberikan penyelesaian akhir, tetapi kepelbagaian populasi agak longgar pada peringkat awal proses pengoptimumannya. Maka, AKB berevolusi kooperatif diperkenalkan untuk mengawal kepelbagaian populasi semasa proses pengoptimuman untuk meningkatkan kualiti AKB berdasarkan kaedah PCT. Ini dilakukan dengan membahagikan dimensi masalah kepada beberapa bahagian dan mengoptimumkan setiap satunya di dalam sub-populasi yang berasingan. Untuk menilai kesesuaian umum dan keupayaan kaedah yang dicadangkan, tiga pengelas dan dua set data piawai standard dalam bahasa Inggeris dan satu lagi dalam bahasa Arab telah digunakan. Hasil keputusan menunjukkan bahawa kaedah yang dicadangkan sebelum ini semakin mantap meningkatkan prestasi pengelasan berbanding dengan hasil terbaik yang dilaporkan di dalam sorotan susastera. Peningkatan ini diperolehi untuk kedua-dua set data bagi Bahasa Inggeris dan Bahasa Arab menunjukkan kesesuaian umum kaedah PCT yang dicadangkan di dalam set data bagi kategori bahasa.

ABSTRACT

Given the huge amount of the textual data generated and shared on the internet such as news reports, articles, tweets and product reviews, the need for effective Text-Feature Selection (TFS) becomes increasingly important. This is challenging due to the high dimensionality of text data. Most of the current TFS methods ignore the feature dependencies, which reduces the quality of the selected feature set and affect the classification performance. Other methods depend on population-based meta-heuristic algorithms, which improve the quality of the selected feature set and the classification results. However, this type of methods is classifier dependent and produce higher number of features. In addition, these algorithms are exposed to premature convergence due to poor population diversity. Moreover, the performance of meta-heuristics is less efficient when tackling high-dimensional problems, and the population diversity needs to be controlled during the optimization process. To handle these problems, this research aims to develop a method based on Binary Bat Algorithm (BBA) to improve TFS. First, rough set theory is adapted and used to evaluate the solutions produced by BBA. The proposed method is compared with a wrapper version of BBA based TFS method. Then, a modified version of Latin Hypercube Sampling (LHS) method is proposed to initialize a diverse population. The proposed method is compared with random initialization in terms of the performance of TFS method during optimization process and the classification results. Experiments show that the proposed initialization method improves the diversity of the initial population and the final solution, but the population diversity decrease during early stages of the optimization process. Thus, a cooperative co-evolutionary BBA is introduced to control the population's diversity during the optimization process and to improve the performance of BBA based TFS method. This is done by dividing the dimension of the problem into several parts and optimizing each of them in a separate sub-population. To evaluate the generality and capability of the proposed method, three classifiers and two standard benchmark datasets in English and one in Arabic have been used. The results show that the proposed method steadily improves the classification performance in comparison with best results reported in literature. The improvement is obtained for both English and Arabic datasets which indicates the generality of the proposed TFS method in terms of the dataset language.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS

ACO	Ant Colony Optimization
ARE	Average Relative Error
BA	Bat Algorithm
BBA	Binary Bat Algorithm
BRE	Best Relative Error
F	F1 measure
FS	Feature Selection
GA	Genetic Algorithm
KNN	K Nearest Neighbour
KRS	Knowledge Representation System
LHS	Latin Hypercube Sampling
ML	Machine Learning
NB	Naïve Bayes
P	Precision
PSO	Particle Swarm Optimization
R	Recall
RI	Random Initialization
RST	Rough Set Theory
SN	Signal-to-Noise
SVM	Support Vector Machine
TFS	Text Feature Selection
WRE	Worst Relative Error

CHAPTER I

INTRODUCTION

1.1 RESEARCH BACKGROUND

Feature Selection (FS) has been an active research area in the pattern recognition, machine learning, statistics, and data mining communities. The main idea of feature selection is to choose a subset of the original variables by eliminating redundant features and those with little or no predictive information. The central premise when using a feature selection technique is that the data contains many features that are either redundant or irrelevant, and thus can be removed without incurring much loss of information. Feature selection is very important process, as it can make-or-break a classification engine (Pervaiz et al. 2018). Feature selection is considered an optimization problem (Alijla et al. 2018; Zhang et al. 2018) where the aim is to select the most representative features that give the highest prediction performance.

During the past decades, many conventional feature selection methods have been proposed. These methods could be proudly classified into two main categories: filters and wrappers. Filters measure the feature's relatedness using various scoring methods that are independent from the utilized classifier, and selects top-N features attaining the highest scores. However, most of these methods takes into account the relation between each feature with its corresponding category and ignore the effect of other features. In other words, those filter methods didn’t consider the features dependencies (Belhaouari et al. 2015; Tejada et al. 2017; Labani et al. 2018). On the other hand, a significant drawback in some feature selectors that they require user-supplied information. Some simply rank features leaving the user to choose their own subset. There are those that require the user to state how many features are to be chosen, or they must supply a threshold that determines when the algorithm should terminate.

All of these methods require the user to make a decision based on their own (possibly faulty) judgment (Cuevas et al. 2016).

Among many methods which are proposed for FS, population-based meta-heuristic methods such as genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO) and bat algorithm (BA) have attracted a lot of attention (Rodrigues et al. 2014; Xue et al. 2014; Aghdam & Heidari 2015; Bakar & Hamdan 2016; Zhang et al. 2017; Jiang et al. 2017; Fei 2017; Dong et al. 2018; Fallahzadeh et al. 2018; Ghareb, Hafiz et al. 2018; Niu et al. 2018) . These methods try to gather better solutions by using knowledge from previous steps. Therefore, focus on search strategies have shifted to meta-heuristic algorithms, which are well suited for searching among a large number of possibilities for solutions. However, most of the existing meta-heuristic based feature selection algorithms are classifier-based, which are argued to be less general, that is the selected features may obtain low performance in classification algorithms rather than the internal classification algorithm used in the evaluation function (Al-Ebbini et al. 2017; Sharmin et al. 2017; Alim et al. 2018).

Bat algorithm (BA) is a meta-heuristic method proposed by Yang (2010) based on the fascinating capability of micro-bats to find their prey and discriminate different types of insects even in complete darkness. The algorithm is formulated to imitate the ability of bats to find their prey. Such approach has demonstrated to outperform some well-known nature-inspired optimization techniques. The main advantage of the BA is that it combines the benefits of population-based and single-based algorithms to improve the quality of convergence (Mirjalili et al. 2014; Jaddi et al. 2015). BA and its variants have been successfully applied to solve many problems such as optimization, classification, feature selection, image processing and scheduling (Alihodzic et al. 2017; Chakri et al. 2017; Parashar et al. 2017; Tuba et al. 2017; Chaturvedi et al. 2017; Fei 2017; Nandy & Sarkar 2017; Dao et al. 2018; Niu et al. 2018)

Text classification is the process of automatic grouping of documents into some predefined categories. The idea of text classification is to assign one document to one class (i.e., category), based on its contents. It can provide conceptual views of document collection and has important applications in the real world. For example, news stories are typically organized by subject categories (topics) or geographical codes; academic papers are often classified by technical domains and sub-domains; even patient reports in health-care organizations are often indexed from multiple aspects, using taxonomies of disease categories, types of surgical procedures, insurance reimbursement codes and so on.

Text Feature Selection (TFS) is an important part of text classification, and much research has been done on various feature selection methods. A document usually contains hundreds or thousands of distinct words that are regarded as features. However, many of them may be noisy, less informative, or redundant with respect to class label. This may mislead the classifiers and degrade their performance in general (Deng et al. 2018). Feature selection can be thought of as selecting the best words of a document that can help classify that document. The idea of TFS, in simple words, is to determine the importance of words using a defined measure that can keep informative words, and remove non-informative words, which can then help the text classification engine.

Although many TFS methods have been proposed in literature, most of them focused either on conventional filter methods, or classifier-based meta-heuristic methods. Most of TFS that based on filter methods ignores feature dependencies, which affects the quality of the resulted feature set (Tejada et al. 2017; Labani et al. 2018). On the other hand, most of the meta-heuristic-based TFS methods utilized a classification algorithm as an evaluation criteria, which make the resulted feature sets biased to the choice of classifier (Sharmin et al. 2017; Al-Ebbini et al. 2017; Alim et al. 2018). Therefore, this research focuses on investigating and enhancing the performance of binary bat algorithm as a text feature selection method.

1.2 RESEARCH PROBLEM

One of the key problems in text feature selection is the high dimensionality of feature space which affects the accuracy of text classification. Selection of distinctive feature set is therefore essential in order to enhance the accuracy of the text classification. One way to guarantee the optimality of any search is to evaluate every possible solution. For feature selection in textual data, an exhaustive search is very computationally intensive and is not feasible even for tiny datasets. Therefore, the focus on search strategies has shifted to meta-heuristic algorithms, which are well suited for searching among a large number of possibilities of solutions.

i. Limitation of existing text feature selection methods

Most existing methods for TFS problem are filter ranking methods, which take into account the relation between each feature with its corresponding class. These methods simply evaluate each feature individually based on certain evaluation criteria, and filter out the low ranked features and ignore the effect of the other features . In other words, these methods do not consider the dependencies and interactions among features, resulting in poor quality feature subset that degrade the classification performance (Tejada et al. 2017; Labani et al. 2018). However, a feature which is weakly relevant to the target class by itself, could significantly improve the classification accuracy if it is used together with some complementary features (Labani et al. 2018). In contrast, an individually relevant feature may become redundant when used together with other features. The removal or selection of such features may miss the optimal feature subset.

On the other hand, some research work tried to tackle this issue utilizing either wrapper or filter methods. Wrappers for feature subset selection have been developed in which an optimal feature subset is searched that is tailored to a particular learning algorithm and a particular training set. It tends to find features better suited to the predetermined learning algorithm resulting in superior learning performance. However, wrapper approaches for feature selection are classifier dependent, and this makes the resulted feature sets biased to the choice of classifier (Sharmin et al. 2017; Al-Ebbini et al. 2017; Alim et al. 2018). Other methods that consider features' dependencies are multivariate filter methods, where the main limitation of these methods is that they rely heavily on greedy search techniques to generate the feature subsets, such as sequential forward selection (SFS) and sequential backward selection (SBS). Both of these techniques suffer from "nesting effect" because a feature that is removed or selected cannot be selected or removed in later stages (Xue et al. 2016; Hafiz et al. 2018).

Meanwhile, dependency measure of Rough Set Theory (RST) have the ability to accurately estimate the feature relevancy and redundancy (Chebrolu & Sanjeevi 2015a; Varma et al. 2015; Xue et al. 2016) and expected to improve the evaluation strategy and enhance the goodness of selected subset of features independently from any classifier. However, since little guidance are available about how to apply RST for row text data, an explorative study has to be conducted in order to determine how it could be effectively used for the problem domain.

Among the population based meta-heuristic algorithms, Binary Bat Algorithm (BBA) has the advantages of easier implementation, having fewer parameters and converging more quickly (Yang et al. 2018). However, the investigation of using BBA for feature selection has much less work and much shorter history than other population based meta-heuristic algorithms. It is needed to further investigate and improve the performance of BBA for TFS. Therefore, developing a TFS method based on BBA that appropriately adapted the dependency measure of RST is expected to select more discriminative feature set and therefore, improve the text classification performance.

ii. Premature convergence due to poor diversity of initial population.

Meta-heuristic algorithms are classified to single-based and population-based approaches with respect to the number of solutions. Population-based approaches have been successfully implemented by many researchers to solve feature selection problems. However, one of the important issues in a population-based algorithms is the diversity of solutions in the initial population in order to better explore the search space as it affects the convergence and the quality of the final solution (Pandey et al. 2014; Bajer et al. 2016). This seems to be harder with TFS because of the huge dimensionality, which make it difficult to cover the search space (Bajer et al. 2016). While random initialization can improve the performance on small to moderate dimensional problems, in high dimensional problems as in text feature selection, it might not allow sufficient diversity to provide better exploration of the search space (Bajer et al. 2016). Although multiple constructive heuristic methods were proposed in literature and successfully used as initialization methods (Jawarneh, & Abdullah 2015; Alssager et al. 2017; Viagas et al. 2018), most of them are designed for constrained problems and they are more suitable for single-solution meta-heuristics (Talbi 2009; Stützle & Ruiz 2017). Another type of initialization methods includes parallel diversification such as Latin Hypercube Sampling (LHS) method, is able to provide a good coverage to the search space (Talbi 2009; Gendreau & Potvin 2010). However, the only found study that utilized LHS as an initialization method is the study of Hamdan and Qudah (2015), which is applicable for continuous problems only. Therefore, introducing a modified LHS-based initialization method for the problems with binary representation, and investigate its performance with TFS is needed.

iii. Loosing diversity during search process and less efficient performance of meta-heuristics when tackling high-dimensional problems as text feature selection.

Furthermore, one of the limitations with many meta-heuristic algorithms is that their deficient performance with high-dimensional problems as the search space is not effectively explored due to losing population diversity during the search process (Ebrahimpour et al. 2018). Many methods were proposed in literature to control the population diversity including cooperative algorithms (Crainic 2017; Dao et al. 2018)(Crainic 2017). However, for high-dimensional problems, co-evolutionary algorithms are preferred as they able to divide the dimension of the solution into multiple parts, and optimize each part separately (Ebrahimpour et al. 2018). Moreover, as the text data is represented as a sequence of terms where each term considered one feature, this aggravate the problem of high dimensionality. Therefore, one of the challenging tasks is developing a co-evolutionary algorithm in a way that it is able to control the population's diversity and handle a TFS problem more efficiently.

iv. Generality in handling different languages.

Applicability of feature selection approach to handle different language datasets is another important issue as the structure of the different languages is not the same. In particular, Arabic language has a rich nature and very complex morphology and thus, it is not a trivial task to process and classify Arabic text dataset (Ghareb et al. 2018). Therefore, investigating the generality of the TFS method in terms of text language, by evaluating its performance on Arabic dataset is needed.

1.3 RESEARCH QUESTIONS

Based on the recently elaborated problems for optimizing feature selection, the main research question of this study is: How can Binary Bat Algorithm (BBA) be effectively designed for solving TFS problem? The detailed questions that need to be answered are:

1. How the dependency measure of RST could be used as an objective function in BBA to improve text feature selection?

2. How to develop an initialization method based on LHS to generate a diverse population for the TFS method?

3. How the co-evolutionary strategy could be used to improve the performance of BBA with TFS and to control the diversity of population during the search process?

4. Is the developed method general enough to handle the feature selection problem for Arabic text classification?

1.4 RESEARCH OBJECTIVES

The main aim of this research is to develop a method based on BBA and investigate its performance as a TFS method with ultimate goal to improve text classification performance. In order to achieve this major aim, several objectives are outlined as follows:

1. To develop a TFS method based on BBA and adapt dependency measure of RST which consider the features' dependencies in order to improve the selected feature set and the classification performance.

2. To develop a modified Latin Hypercube Sampling (LHS) method to generate a diverse initial population for BBA in order to avoid premature convergence.

3. To develop a co-evolutionary BBA method within a multi-population in order to improve its performance in TFS and control the population's diversity during the search process.

4. To evaluate the generality of the proposed text feature selection approach on Arabic dataset.

1.5 RESEARCH SCOPE

This research focuses on developing a text feature selection method with ultimate goal of improving the text classification performance. For this purpose, a population-based meta-heuristic algorithm namely Binary Bat Algorithm (BBA), is used. In terms of evaluation criteria, the fitness function of BBA will be limited to dependency measure of rough set theory, in order to consider the dependencies and interactions among features. Regarding enhancing the search technique, initialization method will be used which in turn is limited to the random initialization and Latin Hypercube Sampling (LHS) initialization. Furthermore, a co-evolutionary strategy will be adapted on BBA in order to improve the performance of the proposed TFS method when handling text-feature selection and controlling the population's diversity during the search process. The proposed method is applied on two standard English datasets namely WebKB and Reuters-21578. In addition to that, an Arabic dataset namely, Al-jazeera news, will be used to evaluate the generality of the proposed TFS method in handling Arabic language. Other issues such as dataset imbalance, enhancing the learning algorithm and computational complexity are beyond the scope of this research.

1.6 THESIS ORGANIZATION

This thesis contains seven chapters, including the current chapter. Chapter I is the introduction section and covers an introduction to the research, research problem, questions, objectives and scope.

Chapter II presents the literature review on various aspects related to this work, like feature selection methods, classification algorithms, bat algorithm and rough set theory. Then, it concentrates on the reviews and analyses of currently available published studies on TFS problem. The key challenges related to meta-heuristics for text feature selection are also presented in this chapter.

Chapter III describes the research methodology used in this thesis. The chapter starts with the research structure that defines the factors of the research. Then the research design is introduced that show the main phases of the research with datasets and evaluation metrics.

Chapter IV investigates the implementation of BBA as a text feature selection method with Naïve Bayes classifier. To improve the evaluation, the dependency measure of rough set is adapted instead of the classification method to provide better evaluation for the candidate solutions. The proposed method is compared with the baseline version (i.e., BBA with Naïve Bayes), in terms of classification performance and dependency on classifier.

Chapter V introduces a modified Latin Hypercube Sampling (LHS) method for initializing a diverse population in order to avoid premature convergence. The proposed method works with the problems with binary representation, while the original one works with continuous data.

Chapter VI investigates the performance of multi sub-population BBA as a co-evolutionary method. The main idea of this method is to divide the solution to multiple parts and optimize each part with one population according to divide-and-cooperative strategy. This chapter also evaluates the generality of the proposed method in terms of the language of dataset, by applying those methods on Arabic dataset. The results are compared with the recent studies that used the same dataset.

Finally, the conclusions of the work presented in this thesis and future works in this area are presented in Chapter VII.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

Feature selection is the most important step in any classification system. Feature selection is commonly used to reduce the dimensionality of datasets with tens or hundreds of thousands of features which would be impossible to process. In the last decade, hundreds of feature selection algorithms have been proposed. However, relatively small portion of them devoted to text classification.

This chapter gives an overview of the research undertaken in the field of feature selection. Then, due to the scope of the thesis, it focuses on text feature selection. Recent work related to the problems and objectives mentioned in the previous chapter are critically reviewed in order to determine the gap. It starts by overview about the feature selection including definition of feature selection, general feature selection process, feature selection method and feature selection for text classification. Next, the meta-heuristic algorithms for feature selection are presented, followed by bat algorithm. Then, the basic concepts of rough set theory and its applications and related work have been introduced. After that, machine learning and classification and some related aspect are described. Finally, the limitations found in the literature are discussed and suggest an innovative method to solve the problems.

2.2 FEATURE SELECTION

This section provides the definition of feature selection and the general process of feature selection. Then, the feature selection approaches are reviewed and discussed. Due to the scope of this work, more emphases have been given to text feature selection.

2.2.1 Definition of Feature Selection

Feature Selection (FS) has been defined by authors in many forms by looking at different perspectives, but most of them are similar in intuition and/or content. This section attempts to consolidate various terms and definitions of FS that were used in the literature. The feature selection problem can be found in all supervised and unsupervised machine learning tasks such as classification, clustering, regression and time-series prediction. Throughout this thesis, the emphasis of feature selection is around the text classification problem. In feature selection for classification, due to the availability of class label information, the relevance of features is assessed as the capability of distinguishing different classes. For example, a feature f_i is said to be relevant to a class c_j if f_i and c_j are highly correlated.

Jain et al. (2000) defined feature selection, is to select a subset of size *m*, given a set of *d* features that gives the minimum classification error. They further state that the straightforward method to the FS problem is based on two aspects: (1) to examine all possible subsets of size *m* and (2) to select the subset with the largest value of classification. Another definition introduced by John et al. (1994). They define Feature selection that, is to improve the classification accuracy or to reduce the size of the data structure diminution by choosing a subset of features without sharp drop in the classification accuracy. From another point of view, Liu & Yu (2005) defined feature selection as a process that finds the minimal size features subset from the feature set based on some evaluation criteria. Similarly, Guyon (2008) defined feature selection as making good classification with as small number of features as possible.

Overall, feature selection is the process of finding a small subset of original features that is necessary and sufficient to solve a classification problem. Naturally, the optimal feature subset is the smallest subset that can obtain the highest classification performance, which makes feature selection an optimization problem (i.e. to minimize the number of features and to maximize the classification performance) (Alijla et al. 2018; Zhang et al. 2018).

By reviewing the terms, feature selection is also known as subset selection or variable selection or attribute selection. In this thesis, the term of feature selection and subset selection is interchangeably used.

2.2.2 Feature Selection Process

The feature selection algorithm usually involves four key stages. Figure 2.1 depicts the main steps of feature selection as in (Dash et al. 1997; Liu & Yu 2005). These steps are subset generation, subset evaluation, stopping criterion, and result validation. The four stages will briefly explained in the following paragraphs.

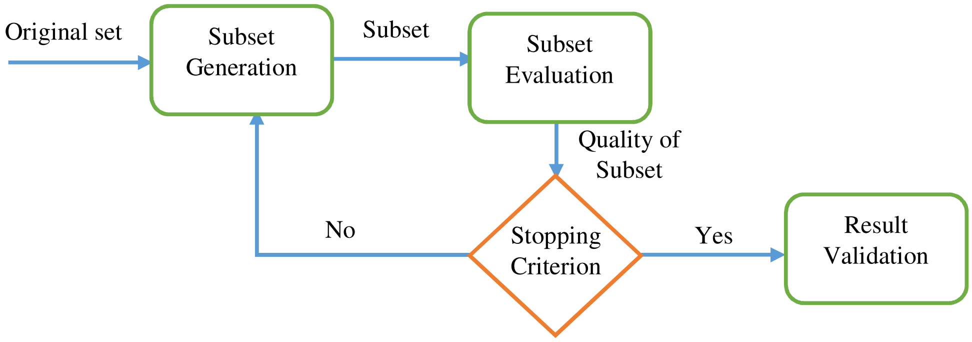

Figure 2.1 Main steps of feature selection

Source: Liu & Yu (2005)

a. Subset generation

Subset generation is basically a heuristic search process which lies under two fundamental issues: The starting point and the search strategy. Regarding the starting point, four methods are exist. The forward selection , where the search begins with no features and the features are added successively, or the backward selection, where the search begins with all features and the features are removed successively, or bidirectional selection which begin with both ends and remove or add features concurrently or begin with a random selected subset of features. In the exhaustive search, a dataset with F features generates 2^F potential subsets, which is exponentially laborious even with a moderate *F*. Regarding the search strategy, there are three different search strategies; complete, sequential or random.

i. Complete Search

Using this kind of search, obtaining the optimal feature set is guaranteed based on the evaluation criteria. The order of the complete search space is O(2F). There are very few feature selection methods that use an complete search such as the work of Liu & Zhao (2009) and Liu et al. (2010). Although exhaustive search is a complete search, there exist different heuristics such as beam search and best first search that can be employed without risking the opportunity of searching for the optimal result.

ii. Sequential Search

It ignores the completeness and expose to lose the optimal subsets. This technique is able to produce fast results and it is simple to implement. In sequential search, the order of the search space is usually $O(F^2)$ or less. In this technique, the features are added or removed one at a time. It has multiple variations to the greedy hill climbing such as sequential backward elimination, sequential forward selection, sequential backward floating selection and sequential forward floating selection (Pudil et al. 1994, Mao & Tsang 2013).

iii. Random Search

This strategy (Dong et al. 2018; Yiğit & Baykan 2014) starts the search by a feature subset that is selected randomly, and continue the search in two directions. The first direction follows the sequential search and the randomness is then inserted on the standard sequential search. This direction is also known as non-deterministic. The second direction randomly produces the subsequent subset. This direction is also known as Las Vegas algorithm.

Feature selection problems have a large search space, which is often very complex due to feature interaction. With such search space, applying complete search is computationally too expensive to perform even with small datasets. Therefore, different heuristic search techniques have been applied to feature selection such as sequential forward selection (SFS) and sequential backward selection (SBS). However, both of these techniques suffer from "nesting effect" because a feature that is removed or selected cannot be selected or removed in later stages (Xue et al. 2016). To overcome the limitations of the traditional subset generators, random search was successfully utilized by population-based meta-heuristics to produce multiple solutions in a single run (Aghdam & Heidari 2015, Xue et al. 2016).

b. Subset evaluation

After determining the subset of the features, the evaluation is needed and it is performed using an evaluation criterion. Two evaluation criteria are exist; including dependent criteria and independent criteria. The dependent criterion is applied in the wrapper methods. A prediction algorithm (i.e., learning algorithm) is required to use this criterion. The feature subset which provides the highest prediction performance is considered the best feature subset. Using this criterion, a superior result is obtained as the identified learning algorithm is used to guide the selection of the features. However, using this criterion, the selected feature set is biased to the utilized classifier.

The independent criterion is usually linked to the filter methods. The quality of the feature subset is based on the characteristics of the training data while ignoring any learning algorithm. Four absolute criteria were commonly used in the literature, including information measures, distance measure, consistency measure and dependency measure.

c. Stopping criterion

The stopping criterion is needed to decide when to stop the feature selection process. There are common methods that used as a stopping criteria to determine when the search process finishes like when a given threshold is met (e.g., maximum number of generations or minimum number of features), subsequent addition or deletion of the features does not generate a better subset, or a sufficiently good subset is selected.

d. Result validation

This criterion is used to check whether the subset is valid, either directly measuring using prior knowledge or indirectly monitoring the change in mining performance. This latter strategy is adopted in most feature selection work, where the performance of the FS method is monitored through the progress of classification accuracy. If the accuracy is maintained or improved despite features reduction, the features that have been selected is considered valid. Similar practices have been implemented by many past studies (Liang et al. 2015; Shunmugapriya & Kanmani 2017; Das et al. 2018).

2.2.3 Feature Selection Methods

In literature, feature Selection has been categorized by researchers in many forms by looking at different perspectives. This section attempts to consolidate various methods and approaches of feature selection that have reported in the literature. Feature selection is an important branch in the machine learning and data mining research area and essential step in any machine learning and data mining task such as classification, clustering and regression. In this study, the focus of feature selection is around the feature selection for text classification.

a. Supervision Perspective

Based on the level of supervision (i.e. the availability of class label information in classification problems), feature selection can be generally categorized as supervised, unsupervised, and semi-supervised methods. Supervised feature selection methods can further be broadly categorized into filter models, wrapper models and embedded models. With supervision information, feature goodness is usually evaluated by estimating the degree of correlation between feature and the targeted class.

The training phase in any classification model relay too much on the selected features. Usually, classifiers are trained based on a subset of features selected by supervised feature selection. The feature selection stage can be independent of the learning algorithms (filter methods) or it dependent on the learning algorithm as it may take the advantage of the classification model to evaluate the goodness of selected features (wrapper methods), or embed the evaluation of selected features in the learning algorithm (embedded methods). Finally, the trained classifier classifies the unseen instances in the test set with the selected features. In this work the focus is on supervised methods for classification problems.

b. Selection Strategy Perspective

Based on the utilized evaluation criteria, the feature selection methods are broadly classified into filter, wrapper and embedded models. Filter methods evaluates the selected subset using the general characteristics of data, while wrapper methods utilize the classification performance to evaluate the feature subset. Wrapper methods aim to improve the performance of the classification algorithm, and is computationally more expensive than filter model. In embedded methods, the feature selection is embed into the learning model. It should be noted that some literature classifies feature selection methods from the selection strategy perspective into four categories by including the hybrid feature selection methods (Saeys et al. 2007; Shen et al. 2012; Ang et al. 2016). Hybrid methods can be regarded as a combination of multiple feature selection methods (e.g., wrapper, filter, and embedded). Hybrid model tries to take advantage of the two models by utilizing the different evaluation criteria in different search stages (Liu & Yu 2005). The first two methods (i.e., wrapper and filter) are the most frequently used in literature and will be discussed in further details in next two subtitles.

i. Wrapper methods

Wrapper methods rely on the classification performance of a predefined classification algorithm in order to evaluate the quality of feature subset. Given a specific classification algorithm, a typical wrapper method performs two steps, including subset generation and subset evaluation. These two steps are repeated until some stopping criteria is met. The search component first generates a features subset, and then the classification algorithm acts as a black box to evaluate the quality of these features based on the classification performance. For example, the whole process works iteratively until the highest classification performance is achieved or the desired number of selected features is obtained. Then the feature subset that gives the highest classification performance is returned as the selected features. Unfortunately, a known issue of wrapper methods is that the search space for *N* features is 2^n, which is impractical when the dimensionality is very large. Therefore, different search strategies such as sequential search (Guyon & Elisseeff 2003), hill-climbing search, best-first search(Arai et al. 2016), branch-and-bound search (Narendra & Fukunaga 1977), and genetic algorithms (Golberg 1989) are proposed to yield a local optimum classification performance. However, the search space is still extremely huge for high-dimensional datasets. As a result, wrapper methods are seldom used in practice. Figure 2.2 illustrates the process of the wrapper.

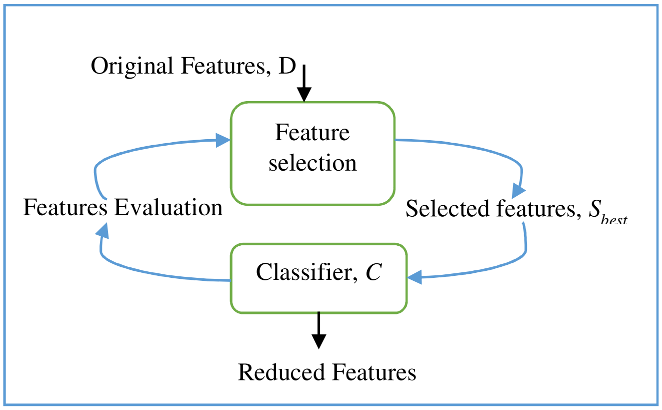

Figure 2.2 Wrapper feature selection

ii. Filter methods

The filter model separates feature selection from classifier learning so that the bias of a learning algorithm does not interact with the bias of a feature selection algorithm. It relies on measures of the general characteristics of the training data such as distance, consistency, dependency, information, and correlation. In filter method there are two main steps. In first step, feature assessed based on some evaluation criteria. The feature evaluation can be either univariate or multivariate. In the univariate scheme, each feature is assessed individually isolated of other features, while the multivariate approach assesses multiple features in a batch way. In the second step of filter method, features arranged in descending order and the low ranked feature are sorted out.

In the past decades, different evaluation criteria for filter methods have been proposed. Some representative criteria include feature discriminative ability to separate samples (Yang & Mao 2011; Du et al. 2013; Tang et al. 2014), feature correlation (Guyon & Elisseeff 2003; Koller & Sahami 1996), mutual information (Nguyen et al. 2014; Gao et al. 2016; Shishkin et al. 2016), feature ability to preserve data manifold structure (Jiang & Ren 2011; Gu et al. 2012), and feature ability to reconstruct the original data (Li et al. 2017). Figure 2.3 illustrates the process of the filter.

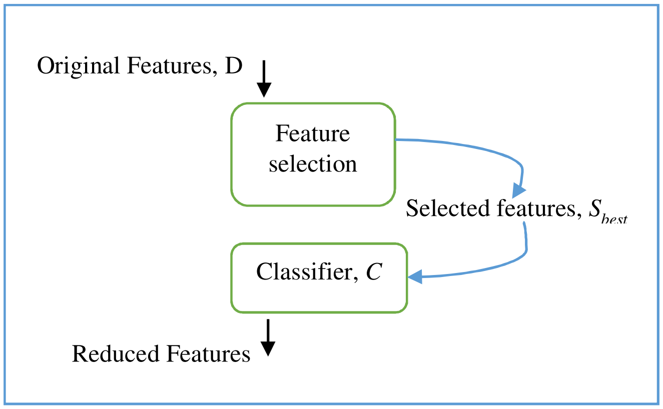

Figure 2.3 Filter feature selection

As mentioned earlier, the filter methods can be further divided into two types of methods; univariate and multivariate. Univariate is the most frequently used to address the feature selection problem. Univariate methods are to use a feature ranking method to filter out the least promising features before precoding the data to learning algorithm. These methods have been used extensively in many domains (Okun 2011; Manek et al. 2017; Wu et al. 2017). However, correlation filters could prompt some loss of relevant features that are meaningless by themselves but that can be useful in combination. The univariate methods evaluate the features individually. As a result, they ignore dependencies between the features, while the multivariate methods evaluate the quality of each subset using certain measurement criteria. The multivariate methods consider the feature dependencies during the selection process, however it is not fast as the univariate methods. Some examples of univariate filters are chi-square (Thaseen & Kumar 2017), Euclidean distance (Sharif et al. 2017), information gain (Zhu et al. 2017) and gain ratio (Nagpal & Gaur 2015). Some examples of multivariate filters are correlation-based feature selection (Jain et al. 2018), Markov-blanket filter (Yu et al. 2017) and fast correlation-based filter (Egea et al. 2018).

c. Single feature ranking and multi feature ranking

Single feature ranking is a relaxed version of feature selection (Guyon & Elisseeff 2003). Single feature ranking is computationally cheap, because it only requires the computation of the relative importance of the individual features and subsequently sorting them. In single feature ranking, a score denotes the relative importance of a single feature, which is measured by a predefined criterion. All the features are ranked according to the score and then feature selection can be accomplished by selecting a small number of top-ranked features. Normally, users specify the number of top-ranked features they need according to their requirements. There are also analytical methods to determine the best number of features (Duch et al. 2003).

Many measures have been proposed to evaluate the relative importance of each feature in single feature ranking algorithms, such as information gain, gain ratio and mutual information. Most single feature ranking methods fall into the filter approach category and not much work has been conducted on wrapper based single feature ranking. However, existing single feature ranking algorithms only measure the goodness of a single feature, not taking into account the interaction between groups of features (Guru et al. 2018).

The combination of one or more features to compute their correlation with target class are more likely to be complementary to each other and can typically achieve better classification performance. This kind of feature selection method called as multi feature ranking method or multivariate methods. multi feature ranking methods consider the feature dependencies in selecting the feature subsets, however it is generally slower than the univariate methods.

To sum up, there are many ways in which feature selection methods are categorised in the literature. The first categorized the FS methods from the supervision prospection into methods for supervised learning and methods for unsupervised learning while the second categorised them based on selection strategy into filter, wrapper and embedded (Kohavi & John 1997; Guyon & Elisseff 2003) the third categorised the FS methods into single feature ranking and multi feature ranking (Liu & Yu 2005).

2.2.4 Feature Selection for Text Classification

In text domains, effective feature selection is essential to make the learning task efficient and more accurate (Forman 2003). Although there are numerous feature selection algorithms proposed in the last decade, a relatively small portion of them are dedicated to text classification. This section reviews the recent studies that have been carried out for text feature selection.

These studies could be categorized in several ways by various criteria. For instance, the studies can be grouped from the supervision perspective into supervised or unsupervised or based on whether they return a ranking or subset to single ranking method and subset ranking method. From another prospective, these studies can be also categorized based on the used search strategy into wrapper (which typically use meta-heuristic method) methods or filter based methods. The filter methods can be further divided into single or subset ranking based on whether they return a ranking or subset. The later also known as univariate and multivariate. With such categorization, this studies can be explained more clearly.

a. Univariate and Multivariate -filter (single vs subset ranking method)

Single feature ranking method (aka univariate method) is a relaxed version of feature selection. In single feature ranking, a score denotes the relative importance of a single feature, which is measured by a predefined criterion. All the features are ranked according to the score and then feature selection can be accomplished by selecting a small number of top-ranked features. Normally, users specify the number of top-ranked features they need according to their requirements. Based on this approach Meng et al. (2011) proposed a two-stage feature selection method. In the first stage, they have chosen the features by a novel feature selection method named the Feature Contribution Degree (FCD) method. This method is used to minimize the number of features by choosing the features which have the higher degree of contribution towards classification. In the second stage they have employed Latent Semantic Indexing method (LSI) to develop a new conceptual vector space. Another work (Imambi & Sudha 2011), studied and compared various dimensionality reduction methods at the pre-processing phase. They also proposed a feature weighting scheme and named it global relevant weighting (GRW). Their architecture includes pre-processing layer and feature selection layer.

In another study, (Pinheiro et al. 2012) proposed a filter method for text feature selection, named as ALOFT (At Least One Feature). Their method focused on particular features to ensure that, every document in the training set is depicted by a minimum of one feature. In addition, Uysal and Gunal (2012) proposed a filter based probabilistic feature selection method, namely distinguishing feature selector (DFS), for text classification. The proposed method selects distinctive features while eliminating uninformative ones considering certain requirements on term characteristics.

Another study (Basu & Murthy 2012), proposed a feature selection method depends on a similarity between a term and a class, for text classification. In the proposed way, every term of the vocabulary will be assigned a score depending on its similarity with all the classes. All the terms will be ranked according to their individual score. Then a predefined number of terms having large score will be selected as important features. If a term never occur in a class then the proposed method will generate a negative score by which the term will never be associated with that class. The authors reported that their method performed better than the previous methods.

Later, Ren and Sohrab (2013) proposed class-indexing-based term-weighting approaches. The proposed class-based indexing is incorporated with term, document and class index. They have investigated the efficiency of proposed class-indexing-based TF.IDF.ICSdF (i.e., Term Frequency Inverse Term Frequency Inverse Class Space density Frequency) and TF.IDF.ICF (i.e., Term Frequency Inverse Term Frequency Inverse Class Frequency) approaches, with other term weighing approaches. They reported that their term weighting approaches are efficient in improving the classification task. Al-Thubaity et al. (2013) examined the effect of combining five feature selection methods, namely CHI (Chi Square), IG (Information Gain), GSS (Galavotti, Sebastiani and Simi), NGL (NG, Goh and Low) and RS (Relevancy Score) on Arabic text classification accuracy. Two approaches of combination were used, intersection and union. The experiments show slight improvement in classification accuracy for combining two and three feature selection methods. No improvement on classification accuracy was seen when four or all five feature selection methods were combined.

In another study, Shang et al. (2013) proposed a novel metric called global information gain (GIG). Based on the proposed metric, they also introduce a feature selection method called maximizing global information gain (MGIG). Wang et al. (2014) proposed a t-test feature selection approach based on term frequency. Their approach was compared with the state-of-the-art methods on two text corpora using three classifiers in terms of macro-average-F1 and micro-average-F1.

In later study, Zong et al. (2015) focused on selecting discriminative and semantic feature for text classification. Their method aimed to select more discriminative features by computing the semantic similarity of features and the similarity between features and documents. Another study (Lu et al. 2015), proposed a text feature selection method based on Category-Distribution Divergence (CDDFS). This method computes the degree of membership and degree of non-membership between the feature and the category. It was intended to filter the features having low degree of membership and high degree of non-membership.

Another approach, namely improved global feature selection scheme (IGFSS) was proposed by Uysal (2016). The proposed scheme has the same steps of a common feature selection scheme except the last step where it was modified in order to obtain a more representative feature set. The proposed scheme aims to improve the classification performance of global feature selection methods by creating a feature set representing all classes almost equally. A local feature selection method is used in IGFSS to label features according to their discriminative power on classes and these labels are used while producing the feature sets.

In another recent study, a feature selection algorithm based on gravitation, named GFS, was proposed (Yang et al. 2016). GFS consider a feature occurring in one category as an object, and all objects corresponding to a feature occurring in various categories can constitute a gravitational field, then the gravitation of a feature with unknown category label on which all objects in the gravitational field act is used for feature selection.

Zhen et al. (2016) proposed filter feature selection scheme based on class difference measure. The key idea of their method is difference between the frequencies of document of class in which a term occurs. Wu et al. (2017) proposed text feature selection algorithm by merging the classical methods of Gini index and term frequency (TF), which is named as Gini-TF.

Zhang et al. (2017) proposed a text feature selection method that used two degrees to measure the importance of the features. The first is feature dispersion degree of between-class documents, which used to measure the feature dispersion between categories (the greater its value, the larger the influence of the feature has). The second degree is the feature concentration degree of within-class documents, which used to measure feature concentration in the text of a category (the greater its value, the larger the influence of feature has). They reported that their method with these degrees improves the selection of representative feature set.

In another recent study, Fattah (2017) proposed a statistical feature selection approach for text classification task. This approach measures the term distribution in all collection documents, the term distribution in a certain class and the term distribution in a certain class relative to other classes. Rehman et al. (2017) proposed a feature ranking metric, called normalized difference measure (NDM), which unlike the conventional balanced accuracy measure (ACC2) (Forman 2003), takes into account the relative document frequencies. They reported that their metric outperformed seven previous metrics in more than half of the cases.

In another study, it was suggested to take the interactions of words into account when selecting features for text classification; in order to eliminate redundant terms (Javed et al. 2015). The proposed method work by combining the feature ranking and feature subset selection algorithms in two stages so that feature selection for text classification can have benefits from both these classes of algorithms. The proposed method was compared with two other methods namely DFS (Uysal & Gunal 2012) and IG+PCA (Uğuz 2011) and reported to give better performance in most trials.

It is worth mentioning that during the writing stage of this work, there was some recently published work that proposed multivariate feature selection methods. This first work by Jain et al. (2018) proposed two phase hybrid model for cancer classification. The proposed model integrates Correlation-based Feature Selection (CFS) with improved-Binary Particle Swarm Optimization (iBPSO). Their model selects a low dimensional set of prognostic genes to classify biological samples of binary and multi class cancers using Naïve Bayes classifier. Other work by Egea et al. (2018) proposed a modification of fast-based-correlation feature (FCBF). Their idea is to split the feature space in fragments with the same size. The aim of introducing this division is to improve the correlation and, therefore, the machine learning applications that are operating on each node.

In the domain of TFS, Guru et al. (2018) introduced a framework for selecting a most relevant subset of the original set of features for text classification. Their framework ranks the features in groups instead of ranking individual features. They reported the effectiveness of their framework over the conventional ones. Another recent study (Labani et al. 2018) proposed a text feature selection method called Multivariate Relative Discrimination Criterion (MRDC). Their method focuses on the reduction of redundant features using minimal-redundancy and maximal-relevancy concepts, and it takes into account document frequencies for each term, while estimating their usefulness. Table 2.1 summarizes the related filter method for text feature selection.

Table 2.1 Previous filter text feature selection methods

Author, year	Method	Single/Multi	Language
Meng et al. (2011)	FCD/LSI	Single	English
Imambi and Sudha (2011)	GRW	Single	English
Pinheiro et al. (2012)	ALOFT	Single	English
Uysal and Gunal (2012)	DFS	Single	English
Basu and Murthy (2012)	Term significance	Single	English
Ren and Sohrab (2013)	TF.IDF.ICSdF TF.IDF.ICF	Single	English
Shang et al. (2013)	GIG	Single	English
Wang et al. (2014)	T-test	Single	English
Zong et al. (2015)	Discriminative and semantic feature selection	Single	English
Lu et al. (2015)	CDDFS	Single	Chinese
Javed et al. (2015)	Markov blanket based	Multi	English
Uysal (2016)	IGFSS	Single	English
Yang et al. (2016)	GFS	Single	English
Zhen et al. (2016)	Class difference	Single	English
Wu et al. (2017)	Gini-TF	Single	Chinese
Zhang et al. (2017)	Dispersion degree	Single	Chinese
Rehman et al. (2017)	NDM	Single	English
Guru et al. (2018)	Alternative framework	Multi	English
Labani et al. (2018)	MRDC	Multi	English

Some conclusions can be drawn from the previous studies that are listed in Table 2.1. First, most of the discussed method are single (univariate) filter ranking method. Although single ranking methods are simple, fast and independent from the classification algorithm, these methods evaluate each feature individually (i.e. the features evaluated in isolation from other features). In high dimensional feature space, an individual feature may have a small correlation with the target class, thus it could be treated as irrelevant and removed from the feature space. However, when it is combined with other features, they form a subset of predictors which can be used to construct highly accurate and efficient predictive models. The loss of useful information by considering these features as irrelevant may degrade the classification performance. However, most of feature selection algorithms which have been applied to text data are based on filter single ranking approaches which assume that there is no potential interaction between features. Thus, the interaction among the features is totally ignored. Guyon & Elisseeff (2003) has conclude that features which are meaningless by themselves can be meaningful together.

Second, there are few works that have been developed to evaluate the features as subset to overcome the limitation of single ranking method (Javed et al. 2015; Guru et al. 2018; Labani et al. 2018). Filter-subset ranking (multivariate) methods are independent from the classifier and has the ability to consider the feature dependencies. However, the proposed filter multivariate methods relay heavily on greedy search techniques to generate the feature subsets, such as sequential forward selection (SFS) and sequential backward selection (SBS). Both of these techniques suffer from "nesting effect" because a feature that is removed or selected cannot be selected or removed in later stages (Xue et al. 2016).

b. Meta-heuristic-based methods (wrapper)

Meta-heuristics optimization techniques are computational methods that iteratively improve the candidate solutions regarding an optimization problem with respect to a particular evaluation criterion (Asghari & Navimipour 2016). Some of these meta-heuristics simulate some characteristics of the behaviour of living beings, as particle swarm optimization, ant colony optimization and genetic algorithms. The meta-heuristic techniques have been used for wrapper feature selection methods. Moreover, they have gained more attention in recent years (Bidi & Elberrichi 2016; Jiang et al. 2017; Alijla et al. 2018; Fallahzadeh et al. 2018; Sainte & Alalyani 2018; Niu et al. 2018) because they attempt to produce an improved solution by applying earlier knowledge gained from the previous solution. Wrapper approach for feature selection depend on the classification performance of a predefined learning classifier to evaluate the quality of selected features. Given a specific learning algorithm, a typical wrapper method performs two steps: (1) searches for a subset of features and (2) evaluates the selected features. These two steps are repeated until some stopping conditions are met.

It first starts with subset generation, then the classifier acts as a black box to evaluate the goodness of generated subset based on the classifier performance. Based on this approach, several works have been proposed in literature.

For example, Mesleh and Kanaan (2008) proposed a feature subset selection method based on Ant Colony Optimization (ACO) and Chi-square statistic. In their method, chi-square statistics is used to calculate the scores of each feature in the initialization step. Support vector machine classifier is used as an objective function. The method was compared with some traditional method on in-house Arabic text corpus and the authors reported that their method obtained better results in terms of classification accuracy.

Another study (Nii et al. 2008), proposed a genetic algorithm based feature selection method for generating numerical data from collected nursing-care texts in order to improve the classification performance. The SVM classifier was used to evaluate the feature sets. Aghdam et al. (2009) proposed a feature selection technique, based on ant colony optimization (ACO) and KNN classifier. They have compared the performance of the proposed method with genetic algorithm, information gain and chi-square on the task of feature selection and they reported the competitive performance of their method.

Later, Zhao et al. (2010) investigated the performance of combined genetic algorithm and k-means algorithm in feature selection for text classification. Their fitness function is based on the average similarity and the average weight of each feature. In another study, Zaiyadi and Baharudin (2010) proposed a hybrid approach for feature selection in text classification based on Ant Colony Optimization (ACO) and Information Gain (IG). Their approach proposed to generate feature sets by ACO and evaluate them using IG. However, no experimental results were reported in order to evaluate the proposed approach. Another studies including (Aghdamet al. 2008; Meena et al. 2012) used ACO for text feature selection as a wrapper model and reported either competitive or better results.

Chantar and Corne (2011) proposed BPSO-KNN based on Particle Swarm Optimization (PSO) as a feature selection method, aimed at finding a good subset of features, to facilitate the Arabic text classification task. In Uğuz (2011), two-stage feature selection and feature extraction was used to improve the performance of text classification. In the first stage, each term within the document is ranked depending on their importance for classification using the information gain (IG) method. In the second stage, genetic algorithm (GA) and principal component analysis (PCA) feature selection and feature extraction methods are applied separately to the terms which are ranked in decreasing order of importance, and a dimension reduction is carried out.

Lei (2012) proposed a text feature selection method based on Information Gain and Genetic Algorithm. First, the features are chosen using Information Gain. Then, the chosen features forms the input of Genetic Algorithm. Yiğit and Baykan (2014) introduced a feature selection method based on Information Gain and Particle Swarm Optimization algorithms.

Later, Aghdam and Heidari (2015) proposed a text feature selection method based on particle swarm optimization and KNN classifier. Another study (Ghareb et al. 2016) proposed a hybrid feature selection approach based on the Genetic Algorithm for Arabic text classification in a wrapper model. In the first step, one of six well-known feature selection methods is used at a time to select the feature subset. Then, an enhanced GA is used to optimize the selected subset. The authors reported that their method is more effective than single filter methods.

Recently, Sainte and Alalyani (2018) proposed a text feature selection method based on firefly algorithm for Arabic text classification. Majidpour & Gharehchopogh (2018) combined Flower Pollination Algorithm (FPA) with Ada-boost for text feature selection. In another recent study (Chen et al. 2018), a text feature selection was proposed, which is based on Water Wave Optimization (WWO) algorithm.

Table 2.2 Comparison between meta-heuristic based feature selection methods

Author/ year	MH method	Fitness function	Type of data	Language	initialization	Improvement strategy
Mesleh and Kanaan (2008)	ACO	SVM classifier	Text	Arabic	Random	-
Nii et al. (2008)	GA	SVM classifier	Text	English	Random	-
Aghdam et al. (2009)	ACO	KNN classifier	Text	English	Random	-
Zhao et al. (2010)	GA	Average similarity	Text	English	Regional growth initialization	-
Meena et al. (2012)	ACO	NB classifier	Text	English	Random	MapReduce parallelization
Chantar and Corne (2011)	PSO	KNN classifier	Text	Arabic	Random	-
Uğuz (2011)	GA	KNN and C 4.5	Text	English	Random	-
Lei (2012)	GA	Cosine similarity	Text	English	Random	-
Yigit and Baykan (2014)	PSO	Cosine similarity	Text	English	Random	-
Aghdam and Heidari (2015)	PSO	KNN classifier	Text	English	Random	-
Ghareb et al. (2016)	GA	NB classifier	Text	Arabic	Random	-
Chen et al. (2018)	WWO	Classification accuracy	Text	Chinese	Random	-
Derrac et al. (2009)	CHC algorithm	Classication accuracy	Integer/ real	-	Random	Co-evolutionary 3 population
Derrac et al. (2010)	CHC algorithm	Classication accuracy	Integer/ real	-	Random	Co-evolutionary 3 population
Tian et al. (2010)	GA	Multi-objective	Integer/ real	-	Random	Co-evolutionary 2 population
Wen & Xu (2011)	GA	Classication accuracy	Integer/ real	-	Random	Co-evolutionary 2 population
Ding et al. (2016)	EA	Classication accuracy	Integer/ real	-	Random	MapReduce model
Ebrahimpour et al. (2018)	BGSA	Information Gain	Micro array	-	Random	Co-evolutionary multi population

Table 2.2 summarizes meta-heuristic based feature selection methods from multiple aspects. Some general conclusions can be drawn from Table 2.2 and the literature review. Most studies have used the classification performance as an evaluation criterion. This leads to selecting a feature set that is bias to the employed classifier. Few studies used a different evaluation criterion, such as cosine similarity (Lei 2012; Yiğit, & Baykan 2014) and information gain (Ebrahimpour et al. 2018). Although these methods are independent from any classification algorithm, they weight each feature individually and then the average weight is considered as the quality of the feature set. This way, the dependencies and interactions between features are still ignored.

Regarding the initialization of the population, random initialization is the mostly used method to generate the initial population except in Zhao et al. (2010) study, where regional growth initialization is used instead. However, this initialization method based on seeding the population with high weight and this method is not enough to generate a diverse population.

Regarding the improvement strategy, few studies employed the co-evolutionary strategy with multi populations in order to improve the meta-heuristic performance. It is worth mentioning that the improvement strategy here referred only to the use of co-evolutionary strategy or parallel methods. The only study that employ parallelization for text feature selection is Meena et al. (2012) study. However, their method includes parallelization of multiple machines, which is not always available. Other studies that employed co-evolutionary strategy were devoted to solve a multi-objective problem, which is not applicable to single objective problems.

2.3 META-HEURISTICS FOR FEATURE SELECTION

Meta-heuristics are a broad family of non-deterministic optimization methods aimed at finding accurate solutions to complex optimization problems when exact methods are not applicable (Xue et al. 2016). Meta-heuristic algorithms are broadly classified into single solution based or population based algorithms. Population based meta-heuristic algorithms are applied successfully for feature selection problem. Population based meta-heuristic algorithms often perform well approximating solutions in different types of problems because they do not make any assumption about the underlying fitness landscape. Therefore, these techniques have shown successes in a variety of fields, ranging from practical applications in industry to leading-edge scientific research (Xue et al. 2016). This section briefly summarizes population based meta-heuristic from two aspects, which are the search techniques, and evaluation criteria. This section also present some concepts regarding population based meta-heuristic algorithms, which are initialization methods and co-evolutionary techniques. Then, the most utilized population based methods for feature selection are briefly discussed and the related work on feature selection is presented.

2.3.1 Search Techniques

There exist a very small number of feature selection methods that are based on an exhaustive search (Dash et al. 1997; Liu & Zhao 2009; Liu et al. 2010). The reason behind that is attributed to the expensive computation of such methods even when the number of features is relatively small (e.g., 50). Therefore, the feature selection utilized different heuristic search techniques, such as greedy search techniques, where the most known examples are sequential forward selection (SFS) and sequential backward selection (SBS). However, both of these techniques suffer from "nesting effect" because a feature that is removed or selected cannot be selected or removed in later stages.

Later, in order to search for the optimal feature subsets, Mao and Tsang (2013) proposed a two-layer cutting plane algorithm. Min et al. (2014) utilized a backtracking algorithm within a heuristic search, which performs an exhaustive search using rough set theory for feature selection problems. The results show that the performance achieved by the heuristic search techniques is similar to the backtracking algorithm but with shorter time. In recent years, population based meta-heuristic algorithms have been applied effectively to solve feature selection problems, such as Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO).

The search space for feature selection problems is large, which is often very complex due to the interactions between features. Ignoring the interactions between features leads to selecting redundant features and features that are not important for classification. That is because the feature that is ranked individually as a high related feature, may become weakly relevant when it is combined and evaluated with other features. Unlike traditional search methods, meta-heuristic algorithms do not make assumption about the search space and do not need knowledge about the domain. Another significant advantage of the population based meta-heuristic is that they can produce multiple solutions in a single run. However, population based meta-heuristic in general have a limitation of getting trap into local optima when the population is not enough diversified.

2.3.2 Evaluation Criteria

The classification performance of the selected feature set is utilized as the evaluation criteria in wrapper feature selection methods. Most of the well-known classification techniques, such as Naïve Bayes (NB), decision tree (DT), K-nearest neighbor (KNN), support vector machine (SVM), linear discriminant analysis (LDA), and artificial neural networks (ANNs), have been applied to wrappers for feature selection (Mesleh & Kanaan 2008; Chantar & Corne 2011; Ghareb et al. 2016). For filter approaches, different measures are used, including measures based on information theory, distance measures, correlation measures, and consistency measures (Xue et al. 2016).

Single feature ranking based on a certain criterion is a simple filter approach, where feature selection is achieved by choosing only the top-ranked features (Liu & Zhao 2009). Single feature ranking methods are computationally cheap but do not consider feature interactions, which often leads to redundant feature subsets (or local optima) when applied to complex problems such as text feature selection.

2.3.3 Initialization Method

One of the important factors for any population-based optimization algorithm is the initial population (Kazimipour et al. 2013; Łapa et al. 2018). Thus, a bad initialization that generates solutions close to each other could leave large areas with no solutions to explore. On the other hand, if the population is very disperse, a large number of iterations may be required to reach a local optimum (Melo & Delbem 2012).

Traditionally, basic random number generators are widely used to generate the initial population. Recent studies suggest that it is possible to significantly improve the performance of a population-based optimization algorithm just by using different initialization methods (Bajer et al. 2016; Kazimipour et al. 2013). Moreover, a large and growing body of literatures has proposed new ways of generating better initial populations.

In an early work, Gao et al. (2012) proposed an initialization approach for PSO in order to improve its performance and avoid premature convergence. The PSO with the initialization method was applied on complex multimodal problems. They employed opposition-based learning method and chaotic maps to generate the initial population. They reported the superiority of their method. Chang et al. (2013) generated the initial population based on cryptex to improve population diversity on weapon-target assignment problem, while Paul et al. (2013) proposed an innovative Vari-begin and Varidiversity (VV) population seeding technique on Travelling Salesman Problem. In another study (Kumar et al. 2013), initialization of population was carried out by repeatedly calling hill climbing approach. Orito et al. (2013) proposed an initialization approach using the extreme point of the bordered Hessian with GA for the portfolio optimization problems.

Later, Pan et al. (2014) proposed an initialization method by applying the concept of adaptive randomness (AR) to distribute the individuals as spaced out as possible over the search space. Delshad and Rahim (2014) proposed an initial technique to initialize the population based optimization methods for solving economic dispatch problems. In addition, Zhang et al. (2015) proposed an initialization method in order to reduce the number of population parameters and make the original population more close to the real strain distribution. Their method used polynomial function parameters to be the initialized population instead of the randomly distributed strain.

Another study (Jawarneh, & Abdullah 2015), examined the sequential insertion heuristic (SIH) as an initialization method to generate a diverse population. Their initialization method was applied with bee colony optimization (BCO) algorithm to tackle the vehicle routing problem with time window (VRPTW) and the results showed the improvement in terms of convergence and final solution. Recently, Chávez and Oropeza (2016) proposed a stochastic algorithm for obtaining feasible initial population to the Vehicle Routing Problem with Time Window. They reported that their method facilitated the generation of highly diverse populations. (Bajer et al. 2016) proposed a method for initializing the population that attempts to generate good solutions in their proximity. The method is based on clustering and a simple Cauchy mutation. They reported that their method increased the convergence rate compared to other methods.

It is shown that there are several initialization methods proposed and applied to different problems with different optimization algorithms. However, most of these methods are either for continuous problems or for low dimensional space. In addition, for feature selection problem, the only found initialization methods were in Xue et al. (2013) and Maini et al. (2017) studies. Xue et al. (2013) proposes an initialization strategy that based on two traditional methods, forward selection and backward selection. Forward selection starts with an empty set of features and it usually selects a smaller number of features, while backward selection starts with the full set of features and selects a large number of features. While in Maini et al. (2017) study, the authors proposed an initialization method for PSO and IDS in order to uniformly sample the search space. Their method divide the population into three parts, where the first part is initialized with small number of features, the second is initialized with medium number of features, and the last one is initialized with a large number of features. The limitations in both Xue et al. (2013) and Maini et al. (2017) studies are that they are not enough to generate a diverse population, do not ensure a uniform distribution and are not efficient with high dimensional problems.

2.3.4 Co-evolutionary Techniques

The simplest definition of a co-evolutionary algorithm is that it is an evolutionary algorithm (or a collection of evolutionary algorithms) in which the fitness of an individual depends on the relationship between that individual and other individuals (Ladjici & Boudour 2011). Such a definition immediately imbues these algorithms with a variety of views differing from those of more traditional evolutionary algorithms.

Therefore, the interaction between individuals of different populations is a key to the success of coevolution techniques.

In the literature, coevolution is often divided into two classes: cooperative and competitive, regarding the type of interaction employed. In cooperative coevolution, each population evolves individuals representing a component of the final solution. Thus, a full candidate solution is obtained by joining an individual chosen from each population. In this way, increases in a collaborative fitness value are shared among individuals of all the populations of the algorithm. In competitive coevolution, the individuals of each population compete with each other. This competition is usually represented by a decrease in the fitness value of an individual when the fitness value of its antagonist increases (Derrac et al. 2010).

Additionally, coevolution is a research field that has started to grow recently. Some research efforts have been applied to tackle the question about how to select the members of each population that will be used to evaluate the fitness function. One way is to evaluate an individual against every single collaborator in the other population. Although it could be a better way to select the collaborators, it would consume a very high number of evaluations in the computation of the fitness function .To reduce this number, there are other options, such as the use of just a random individual or the use of the best individual from the previous generation (Wen & Xu 2011).

In an early work, Bergh and Engelbrecht (2004) presented the cooperative particle swarm optimizer, employing cooperative behavior to improve the performance of the original algorithm. This is achieved by using multiple swarms to optimize different components of the solution vector cooperatively. Krohling and Coelho (2006) proposed an approach based on co-evolutionary particle swarm optimization to solve constrained optimization problems formulated as min–max problems. Another study (Yang et al. 2008), proposed a cooperative coevolution framework in order to optimize large scale non-separable problems. Goh et al. (2010) adapted a competitive and cooperative co-evolutionary approach for multi-objective particle swarm optimization algorithm design, which appears to solve complex optimization problems by explicitly modeling the co-evolution of competing and cooperating species. In another work, Li

& Yao (2012) proposed a cooperative coevolving particle swarm optimization algorithm in an attempt to address the issue of scaling up particle swarm optimization algorithms in solving large-scale optimization problems (up to 2000 real-valued variables).

Later, Jiao et al. (2013) proposed a direction vectors based co-evolutionary multi-objective optimization algorithm, that introduces the decomposition idea from multi-objective evolutionary algorithms to co-evolutionary algorithms. Wang et al. (2014) proposed an adaptive co-evolutionary algorithm based on genotypic diversity measure. In their work, the adaptive selection, mutation and substitution operators are designed to realize cooperative search among operators and dynamic pairing among sub-populations. Another study (Jiang et al. 2015), proposed a co-evolutionary improved multi-ant colony optimization algorithm for ship multi and branch pipe route design. They reported that their algorithm is better than the conventional method at avoiding the problem of local optimum and accelerating the convergence rate. Pan (2016) proposed a cooperative co-evolutionary artificial bee colony algorithm that has two sub-swarms, with each addressing a sub-problem. The sub-problems are charge scheduling problem in a hybrid flowshop, and cast scheduling problem in parallel machines.

Recently, Gong et al. (2017) proposed a multi-objective cooperative co-evolutionary algorithm to optimize the reconstruction term, the sparsity term and the total variation regularization term, simultaneously, for Hyperspectral Sparse Unmixing. Atashpendar et al. (2018) proposed a parallel multi-objective cooperative co-evolutionary variant of the Speed-constrained Multi-objective Particle Swarm Optimization algorithm. Their algorithm adopts a strategy for limiting the velocity of the particles that prevents them from having erratic movements. Jia et al. (2018) proposed a two-layer distributed cooperative co-evolution architecture with adaptive computing resource allocation for large-scale optimization. In another study, Yaman et al. (2018) proposed an approach utilizing Cooperative Co-evolutionary Differential Evolution algorithm to optimize high-dimensional ANNs. The aim of their algorithm is to optimize the pre-synaptic weights of each post-synaptic neuron in different subpopulations, and employs a limited evaluation scheme where fitness evaluation is performed on a relatively small number of training instances based on fitness inheritance.

For feature selection problem, a few studies in literature utilized the cooperative co-evolutionary. Two early works (Derrac et al. 2009, 2010) performed instance and feature selection by creating three populations in different sizes. The first population performs feature selection, while the second population performs instance selection, and the third population is for both feature and instance selection. Tian et al. (2010) presented a hybrid learning algorithm based on a cooperative co-evolutionary algorithm (Co-CEA) with dual populations for designing the radial basis function neural network (RBFNN) models with an explicit feature selection. In this algorithm, the first subpopulation used binary encoding masks for feature selection, and the second subpopulation tends to yield the optimal RBFNN structure.

Another study by Wen & Xu (2011) presented a cooperative coevolution framework to make the feature selection process embedded into the classification model construction within the genetic-based machine learning paradigm. Their approach has two coevolving populations cooperate with each other regarding the fitness evaluation. The first population corresponds to the selected feature subsets and the second population is for rule sets of classifier. Later, Ding et al. (2016) proposed an attribute equilibrium dominance reduction accelerator (DCCAEDR) based on the distributed co-evolutionary cloud model. The framework of N-populations distributed co-evolutionary MapReduce model is designed to divide the entire population into N subpopulations, sharing the rewards of different subpopulations' solutions under a MapReduce cloud mechanism. Recently, Ebrahimpour et al. (2018) proposed CCFS algorithm that divides vertically (on features) the dataset by random manner and utilizes the fundamental concepts of cooperation coevolution in order to search the solution space via Binary Gravitational Search Algorithm (BGSA).

It is noticed that in most of the mentioned studies (Derrac et al. 2009; Derrac et al. 2010; Tian et al. 2010; Wen & Xu 2011), the authors attempted to solve the feature selection problem as a multi-objective problem by creating two or more populations and each of them optimizes one objective. However, they are not applicable for single objective problems and they do not solve the high dimensionality problem. In the work of Ding et al. (2016), the focus was to distribute the optimization process on multiple machines in order to reduce the computational time. However, the requirement of their model, such as the hardware (e.g., multiple PC machines), the mechanism of distribution of the dataset, the way of communication between different machines, and the way of forming the complete solution, is not always available. In the work of Ebrahimpour et al. (2018), the dimension of the full solution is divided to smaller subsets where each of them is optimized in a separate population. Although their method is effective with high dimensional feature selection problem, there are multiple aspects that needs further improvement. For example, the method might has better parameter tuning in order to improve its performance. In addition, the solutions in the different sub-populations need to be combined with each other in each generation in order to be evaluated, which reduces the chance of each solution to be optimized separately from the other sub-populations.

2.3.5 Population Based Meta-heuristics for Feature Selection

Population-based meta-heuristic approaches have been successfully implemented by many researchers to solve feature selection problems. The main concept underlying population-based approaches is that the algorithm frequently improves the quality of the solutions. The most used approaches for solving feature selection problem include genetic algorithm, ant colony optimization and particle swarm optimization. These approaches are briefly described in the following subsections.

a. Genetic Algorithm (GA)

The basic idea of a GA is that it has a population of chromosomes (i.e., strings), that encode the individuals (i.e., candidate solutions) to an optimization problem. In general, the chromosomes are represented by bit strings (i.e., strings of 1s and 0s), that encode the solution. For reproduction, genetic operators are then applied to the population's solutions to generate a new population of solutions.

The main genetic operators are crossover and mutation. Crossover generates two offspring solutions from two parent solutions by copying selected bits from each parent. On the other hand, the mutation operator randomly changes the value of one bit. Additionally, a fitness function is used to evaluate the quality of each solution in order to keep and improve promising solutions.

GA has been utilized for solving the feature selection problem on a number of domains. Some of the studies that utilized GA for feature selection include (Uğuz 2011; Lei 2012; Oreski & Oreski 2014; Welikala et al. 2015; Soufan et al. 2015; Ghamisi & Benediktsson 2015; Cheng et al. 2016; Ghareb et al. 2016; Jiang et al. 2017; Das et al. 2017; Das et al. 2018; Dong et al. 2018; Murthy & Koolagudi 2018).

b. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population-based optimization technique, which was developed by Kennedy and Eberhart (1995). It was inspired by social behavior of bird flocking or fish schooling. PSO starts with a swarm of random particles where each particle is associated with a velocity. The velocity and position of the particle are described mathematically by the following equations:

$$V_i(t+1) = w.V_i(t) + c_1.r_1(t).[P_i(t) - X_i(t)] + c_2.r_2(t).[P_g(t) - X_i(t)] \quad \ldots(2.1)$$

$$Xi(t+1) = Xi(t) + Vi(t+1) \quad \ldots(2.2)$$

where c_1 and c_2 are positive constants, called learning rates; r_1 and r_2 are random values in the range [0, 1]; t indicates the iteration number; w is a inertia weight; the index g represents the best particle among all the particles in the population; the velocity of particle i (i.e., the rate of change of position) expresses as $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$, the position of particle i in the problem space with D dimensions expresses as $(x_{i1}, x_{i2}, ..., x_{iD})$, and the optimal position of particle i expresses as $P_i= (p_{i1}, p_{i2}, ..., p_{iD})$, it is also called *pbest.* The global optimum position of all particles expresses as $Pg= (p_{g1},p_{g2}, ..., p_{gD})$, it is also called *gbest.*

PSO and its variant were successfully applied to solve feature selection problem in multiple domains. Some of the studies that utilized PSO for feature selection are (Xue et al. 2014; Yiğit & Baykan 2014;Ghamisi & Benediktsson 2015; Abdul-Rahman et al. 2015; Aghdam & Heidari 2015; Chinnaswamy & Srinivasan 2016; Moradi & Gholampour 2016; Sheikhpour et al. 2016; Zhang et al. 2015, 2017; Hafiz et al. 2018; Jain et al. 2018; Majid et al. 2018)

c. Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) algorithm are stochastic algorithms which was inspired from the foraging behavior of real ants. When the ants found a food source, they lay some pheromone to mark the path. The quantity of the laid pheromone depends on the distance, quality and quantity of the food source. While an isolated ant moves at random, it can detect the previously laid trail and decide with to follow it, thus reinforcing the trail with its own pheromone.

The artificial ants construct a solution by a sequence of probabilistic decisions. The solution space is initially empty and is expanded by adding a solution component at every probabilistic decision. The transition probability used by ACO is based on the pheromone intensity (i.e., history of previous successful moves), and heuristic information (expressing desirability of the move). The ibest ant or global best ant or both of them deposit pheromone to mark the bath. After all ants have completed their solutions, pheromone evaporation on all edges triggered.

ACO was successfully applied for feature selection in multiple domains (Moradi & Rostami 2015; Tabakhi & Moradi 2015; Wan et al. 2016; Zhang et al. 2015; Aghdam & Kabiri 2016; Dadaneh et al. 2016; Alwan & Mahamud 2017; Shunmugapriya & Kanmani 2017; Sweetlin et al. 2017; Fallahzadeh et al. 2018; Jameel & Rehman 2018).

2.4 BAT ALGORITHM

Several meta-heuristic algorithms to solve various data mining tasks are exist in the literature. They can be classified into population-based and single-based approaches.

Population-based meta-heuristic methods use a number of solutions in an attempt to create a new solution that shares the superior qualities of the previous ones and is expected to have improved fitness function. The single-based approaches get better leading a specific solution by exploiting the neighbourhood with a set of moves. Both population-based and single-based methods are iterative procedures that regularly replace solutions with those of better quality. It has been proven that population-based methods have superior advantages and have better performance compared to single-based approaches (Prugel-Bennett 2010).

The Bat Algorithm (BA) was first proposed by Yang (2010) and is based on the echolocation activity of bats in the natural world. Echolocation is the making of very loud sound waves and echoes to recognize where objects are in space. When sound waves sent by a bat hit an object they generate echoes, which return to the bat's ears. Bats listen to the echoes to understand where the object is, its size and its character. Bats have this ability in darkness.

Using the echolocation behaviour, bats find insects the size of mosquitoes that they like to eat. Bats fly randomly using frequency, velocity and position to search for prey. In the BA, the frequency, velocity and position of each bat in the population is updated for further movements. The algorithm is formulated to imitate the ability of bats to find their prey. The BA follows many simplifications and idealization rules of bat behaviour that were considered and proposed by Yang (2010).

The BA has the advantage of combining a population-based algorithm with local search. This algorithm involves a sequence of iterations, where a collection of solutions changes through random modification of the signal bandwidth which is increased using harmonics. The pulse rate and loudness is updated only if the new solution is accepted. The frequency, velocity and position of the solutions are calculated based on following formulas:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \qquad \ldots(2.3)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_{bes}^t)\, f_i \qquad \ldots(2.4)$$

$$x_i^t = x_i^{t-1} + v_i^t \qquad \ldots(2.5)$$

where the value of β is a random number within the range of [0,1], fi is the frequency of the i^{th} bat that controls the range and speed of movement of the bats, v^i and x^i denote the velocity and position of i^{th} bat, respectively, and x_{bes}^{t} stands for the current global best position at time step *t*. In order to enhance the diversity of the possible solutions a local search approach is applied to those solutions that meet a certain condition in BA. If the solution meets the condition, then random walk (Equation 2.6) is employed to generate a new solution:

$$x_{new} = x_{old} + \epsilon\overline{A^t} \qquad \text{...(2.6)}$$

in which ϵ [-1,1] is a random number that efforts to the power and direction of the random walk and A^t denotes the average loudness of all bats so far

```
objective function $f(x), x = (x_1, ..., x_d)^T$
initialize the bat population $x_i$ ($i$ = 1,2, ..., $n$) and $v_i$
define pulse frequency $f_i$ at $x_i$
initialize pulse rates $r_i$ and the loudness $A_i$
while ($t$ <Max number of iterations)
        generate new solutions by adjusting frequency,
        and updating velocities and locations/solutions [Eq. (2.3) to (2.5)]
       if (rand > $r_i$)
              select a solution among the best solutions
              generate a local solution around the selected best solution Eq. (2. 6)
      endif
      generate a new solution by flying randomly
      if (rand < $A_i$ & $f(x_i) < f(x_*)$)
              accept the new solutions
              increase $r_i$ and reduce $A_i$ [Eq. (2. 7) and Eq. (2. 8)]
     endif
    rank the bats and find the current best $x_*$
endwhile
postprocess results and visualization
```

Figure 2.4 Pseudocode of bat algorithm

Source: Yang 2010

The loudness A_i and the pulse rate r_i have to be updated in each iteration. The loudness typically decreases when a bat find its prey while the pulse rate increases. The loudness A_i and pulse rate r_i are updated as follows:

$$r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \quad \ldots(2.7)$$

$$A_i^{t+1} = \alpha A_i^t \quad \ldots(2.8)$$

In which α and γ are constant values and both are equal to 0.9 as in (Yang 2010). The loudness and pulse rate are updated only if the new solution is accepted. The pseudocode of BA is shown in Figure 2.4.

2.4.1 Bat Algorithm Applications

The standard BA and its variants have been applied to solve many problems such as optimization, classification, image processing, feature selection, and scheduling (Yang & He 2013). In the following, some applications of the BA are briefly highlighted.

a. Continuous Optimization

The standard bat algorithm was mainly proposed to solve continuous constrained optimization problems (Yang 2010). Several studies employed bat algorithm or proposed another variants to solve continuous optimization problems. For example, Tsai et al. (2011) proposed an improved BA, which is called Evolved Bat Algorithm (EBA), by reanalyzing the characteristics of the bat and redefining the corresponding operations based on the basic framework of Bat Algorithm (BA). In addition, Yang (2011) investigated the application of the BA for multi-objective optimization. Simulation results suggest that the proposed algorithm works efficiently. In another study Yang and Gandomi (2012), a new bat algorithm was proposed for solving engineering optimization problems. The extensive comparison study carried out over seven different nonlinear constrained design tasks, reveals that BA performs superior to many different existing algorithms used to solve these seven benchmark problems. It is potentially more powerful than other methods such as GA and PSO as well as harmony search. Also, Bora et al. (2012) optimized the brushless DC wheel motors using bat algorithm with superior results.

In another study, Yilmaz and Kucuksille (2013) proposed three modification to enhance bat algorithm. Results indicate that proposed version has achieved better performance compared to BA on ten benchmark test functions. later, Sakib et al. (2014) compared the Flower pollination algorithm with the basic Bat algorithm on continuous problems and they suggested that the Flower pollination algorithm can perform much better than the Bat algorithm on the continuous optimization problems. Talatahari and Kaveh (2015) presented an improved bat algorithm for optimizing large-scale structures. The authors reported the efficiency of the proposed algorithm.

b. Data mining problems

Research on using BA for k-means clustering was performed by Komarasamy and Wahi (2012) and achieved an improvement in efficiency. Khan et al. (2011) proposed a fuzzy bat clustering method for ergonomic screening of office workplaces. A study of the clustering problem employing the BA and its extension was proposed in (Khan et al. 2012), while a comparison of the BA with PSO and GA in training FNNs in the e-learning context was presented in (Khan & Sahai 2012). Mishra et al. (2012) used the BA for microarray data classification. Natarajan et al. (2012) compared the cuckoo search and BA for Bloom filter optimization in spam filtering. Nakamura et al. (2013) proposed a binary BA for the feature selection problem. Damodaram and Valarmathi (2012) used a modified a BA for phishing website detection and optimization. An optimized approach using a modified BA to record deduplication was presented in (Banu & Chandrasekar 2013).

c. Scheduling problem

Musikapun and Pongcharoen (2012) used the BA to solve a multi-stage, multi-machine, multi-product scheduling problem. They studied a class of nondeterministic polynomial-time hard (NP hard) problems with a featured parametric study. This study showed improvement using an optimal set of parameters. Later, Sathya and Ansari (2015) applied the bat algorithm based dual mode PI controller to the multi-area interconnected thermal power system in order to tune the parameter PI controllers. They compared the proposed controller with those from conventional the PI controllers and Fuzzy gain scheduling of PI controllers. They show that their bat algorithm based controller provided better transient.

In another study, Xu and Zhang (2016) proposed a hybrid discrete bat algorithm with priority assignment rule initialization to avoid the premature convergence. They reported that their algorithm has good performance in solving the flexible job shop scheduling problem. Later, Xu et al. (2017) proposed a method of encoding strategy based on dual flexibility degree in order to express the relationship between the process and the bat population. They reported that their method performance was better than particle swarm optimization and genetic algorithm in solving the flexible job-shop scheduling problem. Recently, Dao et al. (2018) proposed an algorithm based on parallel bat algorithm to solve job shop scheduling problem.

d. Feature selection problem

Nakamura et al. (2013) developed a discrete version of bat algorithm called binary bat algorithm for feature selection. Taha et al. (2013) presented Bio-inspired method called bat algorithm hybridized with a Naive Bayes classifier. Rodrigues et al. (2014) presented a wrapper feature selection approach based on Bat Algorithm (BA) and Optimum-Path Forest (OPF) classifier. Another study, (Laamari & Kamel 2014) presented a feature selection approach for intrusion detection based on bat algorithm and SVM classifier. Later, Rozlini et al. (2015) proposed a feature selection framework based on enhancement of bat algorithm (BA) with Dempster-Shafer.

e. Image processing

Akhtar et al. (2012) presented a full body human pose estimation method using the BA. In their research the BA performed better in comparison with some other algorithms. Zhang and Wang (2012) proposed an image matching method using a BA with mutation. Nandy and Sarkar (2017) employed the concept of the bat algorithm to design an automatic clustering method, and applied it to image segmentation. They adopted a rule-based statistical hypothesis approach for merging similar clusters, which guides the optimization process to find an optimal number of clusters. They reported that their BA based method is faster and convergence is improved over of the method that compared with it.

f. Fuzzy logic

Lemma & Hashim (2011) used a combination of fuzzy systems and BA for exergy modelling while Khan et al. (2011) proposed a fuzzy modification of bat algorithm for clustering of company workplaces. Reddy and Manoj (2012) hybridized a BA with fuzzy logic to find optimal capacitor sizes to minimize losses. In another study, Tamiru and Hashim (2013) applied bat algorithm to study fuzzy systems and to model exergy changes in a gas turbine.

g. Other applications

Bat algorithm is able to be applied and hybridized with other algorithms in many application successfully. For instance, the standard version of Bat Algorithm has been combined with Scheduling Tool by Musikapun and Pongcharoen (2012) to propose a method called BAST to solve multi-stage multi-machine multi-product scheduling problems. The algorithm takes into account the Just-in-Time production philosophy by aiming to minimize the combination of earliness and tardiness penalty costs. The computational experiment on the BAST was conducted using data obtained from a collaborating company engaged in capital goods industry. The experimental results indicated that the BA performance can be improved up to 8.37% after adopting the appropriate parameters' setting.

In a later study, Wang and Guo (2013) proposed a hybrid meta-heuristic HS/BA method for optimization problem. They improved the BA by combining original harmony search (HS) algorithm and evaluated the improved algorithm on multimodal numerical optimization problems. They concluded that the HS/BA significantly improves the performances of the HS and the BA on most multimodal and unimodal problems. Recently, Jaddi et al. (2015a, 2015b) presented an optimization algorithm based on the cooperative bat inspired Algorithm in the first study, and proposed a modified bat algorithm with a new solution representation for both optimizing the weights and structure of ANNs in the second study.

In another study, a novel hybrid Bat Algorithm (BA) with the Differential Evolution (DE) strategy using the feasibility-based rules, namely BADE has been proposed by Meng et al. (2015) to enhance the performance of the basic BA. Experimental results demonstrated that the BADE perform more efficient and robust than the basic BA, DE, and a few other methods.

There are another studies in literature which employed bat algorithm or proposed another algorithms based on bat algorithms such as (Alihodzic et al. 2017; Chaturvedi et al. 2017; Fei 2017; Tuba et al. 2017; Niu et al. 2018)

2.4.2 Advantages of Bat Algorithm

Bat algorithm (BA) is a meta-heuristic method proposed by Yang (2010) based on the fascinating capability of micro-bats to find their prey and discriminate different types of insects even in complete darkness. The algorithm is formulated to imitate the ability of bats to find their prey. Such approach has demonstrated to outperform some well-known nature-inspired optimization techniques.

The main advantage of the BA is that it combines the benefits of population-based and single-based algorithms to improve the quality of convergence (Mirjalili et al. 2014; Jaddi et al. 2015a). The other benefits of the BA that motivate researchers to adopt it to solve a different types of problems are as follows (Yang & He 2013):

i. Frequency tuning: The BA employs echolocation and frequency tuning during the process of problem solving. Although echolocation is not directly used to imitate the right function in the real world, frequency alterations are used.

ii. Automatic zooming: The BA has the ability to automatically zoom into an area where potentially better solutions have been found. This zooming is performed by the automatic shifting from explorative movement to local intensive exploitation. Therefore, the BA has a fast convergence rate in the early stages of the iteration process.

iii. Parameter control: Most meta-heuristic algorithms employ fixed parameters which need to be tuned in advance. In contrast, the BA uses parameter control, whereby the values of the parameters (A and r) are differed as the iterations progress. This helps to automatically direct the BA to move from exploration to exploitation when the best solution is searching.

In addition to the advantages of bat algorithm, Khan and Sahai (2012) presented a comparison study of Bat Algorithm with Genetic Algorithm, Particle Swarm Optimization and other algorithms in the context of e-learning, and thus suggested that bat algorithm has clearly some advantages over other algorithms. In another study, Akhtar et al. (2012) presented a study for full body human pose estimation using bat algorithm, and they concluded that BA performs better than particle swarm optimization (PSO), particle filter (PF) and annealed particle filter (APF). In addition, Gandomi et al. (2013) applied BA to three benchmark constraint engineering problems: pressure vessel design, welded beam design, and spring design. The simulations indicated that BA was very efficient and the results obtained were superior to GA, PSO, and HS. Furthermore, preliminary theoretical analysis by Huang et al. (2013) suggested that BA has guaranteed global convergence properties under the right condition, and BA can also solve large-scale problems effectively.

The possible reason for the superiority of Bat Algorithm is that BA can be considered to be a combination of PSO and intensive local search controlled by loudness and pulse emission rate (Chawla & Duhan 2015). The capability of frequency tuning in BA provides some functionality that might be similar to the key feature used in PSO and harmony search (HS). It uses a good combination of major advantages of these algorithms and, thus, is potentially more powerful than they are (Yang & He 2013; Chawla & Duhan 2015).

As shown in the previous subsection, BA and its variants have been successfully applied to solve many problems such as optimization, classification, image processing and scheduling (Yang & He 2013; Chawla & Duhan 2015; Jaddi et al. 2015a). Although BA has been recently applied successfully to solve feature selection problems (Taha et al. 2013; Laamari & Kamel 2014; Rodrigues et al. 2014), its potential on text-feature selection has not been investigated.

2.5 ROUGH SET THEORY

Rough set theory provides a mathematical tool to find out data dependencies and reduce the number of features included in dataset by purely structural method. Many rough set algorithms for feature selection have been proposed. The complete solution to detect minimal reducts is to produce all possible reducts and choose one with minimal cardinality, which can be done by constructing a kind of discernibility function from the dataset and simplifying it (Emary et al. 2014). Unfortunately, the number of possible subsets of features is always very large. Hence, examining particularly all subsets of features for selecting the optimal one is NP-hard. In literature, multiple meta-heuristic algorithms such as genetic algorithm (Chen et al. 2014; Das et al. 2018), ant colony optimization (Chebrolu & Sanjeevi 2015a; Varma et al. 2015), particle swarm optimization (Chebrolu & Sanjeevi 2015b) and bat algorithm (Emary et al. 2014) successfully contributed with rough set theory to solve feature selection problems.

2.5.1 Basic Concepts of Rough Set Theory

Pawlak (1982) proposed the rough set theory as a formal framework for the automated transformation of data into knowledge. Given a collection of data objects from a universe of interest, a knowledge representation system (KRS) is employed to express observations about the objects collected. A decision table is a type of KRS that represents a relation, typically functional or partially functional, between a group of input values and a set of output values, known as condition and decision attributes respectively. Decision tables are learned or generated from the collection of data objects. Using the concept of rough sets, one can extract a generalized description of objects contained in such decision tables. Extracted descriptions in the form of rules are utilized to sort new objects.

Suppose K is a KRS such that $K = (U, A)$ where U is a non-empty finite subset of objects from a universe of interest, and A is referred to as attributes expressing observations acquired from objects in U . In other words, K can be viewed as a system that maps each attribute $a \in A$ to a value in $V_{a,}$ for every object in a defined universe. This mapping is denoted $a : U \rightarrow V_a$, where V_a is a finite set of values called the domain of the attribute a. The subsections that follow will discuss the basic concepts of the rough set theory, as well as the concept of set approximations.

a. Indiscernibility

Let $I=(U,A)$ be an information system (attribute-value system), where U is a non-empty, finite set of objects (the universe) and A is a non-empty, finite set of attributes such that $a:U \rightarrow V_a$ for every $a \in A$. V_a is the set of values that attribute a may take. The information table assigns a value $a(x)$ from V_a to each attribute a and object x in the universe U. With any $P \subseteq A$ there is an associated equivalence relation $IND\ (P)$:

$$IND(P) = \{(x, y) \in U^2 | \ \forall a \in P, a(x) = a(y)\} \qquad \ldots(2.9)$$

The relation $IND(P)$ is called a P-indiscernibility relation. The partition of U is a family of all equivalence classes of $IND(P)$ and is denoted by $U/IND(P)$ or U/P. If $(x, y) \in IND(P)$, then x and y are indiscernible (or indistinguishable) by attributes from P.

b. Set approximations

In rough set, the lower approximation and upper approximation are two essential operations. Given a random set $X \subseteq U$, the P-lower approximation of X, denoted as $\underline{P}X$, is the set of all elements of U, which can be definitely classified as elements of X based on the attribute set P. The lower approximation is also called the positive region. The P-upper approximation of X, denoted as $\overline{P}X$, is the set of all elements of U, which can be probably classified as elements of X derived from the attribute set P. The upper approximation is also called the negative region. These two definitions can be expressed as:

$$\underline{P}X = \{x | [x]_p \subseteq X\} \qquad \ldots(2.10)$$

$$\overline{P}X = \{x|[x]_p \cap X \neq \emptyset\} \qquad \ldots(2.11)$$

c. Dependency of Attributes

One of the most important aspects of database analysis or data acquisition is the discovery of attribute dependencies. The aim is to discover which variables (features) are strongly related to a decision attribute (class). In rough set theory, the notion of dependency is defined very simply. Let us take two (disjoint) sets of attributes, set P and set Q, and inquire what degree of dependency obtains between them. Each attribute set induces an (indiscernibility) equivalence class structure, the equivalence classes induced by P given by $[x]_P$, and the equivalence classes induced by Q given by $[x]_Q$.

Let $[x]_Q = \{Q_1, Q_2, Q_3, \ldots, Q_N\}$ where Q_i is a given equivalence class from the equivalence-class structure induced by attribute set Q. Then, the attribute set Q depends on the attribute set P in a degree k denoted by $P \Rightarrow_k Q$, which is given by:

$$k = \gamma(P, Q) = \frac{|\underline{P}Q_i|}{|U|} \qquad \ldots(2.12)$$

where $|\underline{P}Q_i|$ is the number of instances in positive region (lower approximation), and $|U|$ is the number of all instances in the search space.

2.5.2 Applications and Related Work

Based on rough set theory, application research mainly focuses on attribute reduction, rule acquisition and intelligent algorithm. Attribute reduction as a NP-Hard problem has been carried out a systematic research. Based on rough set model, the development of reduction theory provides a lot of new methods for data mining. For example, in the different information systems (coordinated or uncoordinated, complete or incomplete), with information entropy theory, concept lattice and swarm intelligence algorithm, the rough set theory has gained the corresponding achievements. At present, the research is mainly concentrated on three aspects, such as theory, application and algorithm (Zhang et al. 2016). In the following subsections, some applications of rough set theory are briefly highlighted.

a. Application of the rough sets in fault diagnosis

Rawat et al. (2016) provided a system for fault diagnosis using the status of an intelligent electronic device and circuit breakers which can be tripped by any kind of fault. To protect the system from vulnerabilities and different kinds of faults, they proposed a multilayered fault estimation classifier, based on the Dominance based rough set. They analyzed the status of different IEDs, which changes their status in the case of a fault and generates an alarm at the control center. In another study, Zhang et al. (2017) proposed a rough set model that combines interval-valued hesitant fuzzy sets with multi-granulation rough sets over two universes, called an interval-valued hesitant fuzzy multi-granulation rough set over two universes. Then, they developed a general approach to steam turbine fault diagnosis by using their model.

In a recent study, Oliveira et al. (2018) presented a method of diagnosing short-circuit faults performed with a digital circuit. Their method identifies short-circuit faults: hard switch fault and fault under load; that can be used with any switch regardless of its parameters. The digital diagnostic circuit is obtained with the use of rough sets theory, which optimizes and defines a minimum set of variables necessary to diagnose faults. A set of diagnostic rules were obtained by applying rough sets theory to the variables. Another recent studies that applied rough set theory for fault diagnosis include (Fei 2017; Niu et al. 2017; Suo et al. 2017; Zhao et al. 2017; Huang et al. 2018).

b. Application of the rough sets in pattern recognition

Liu (2014) proposed a method of character pattern recognition based on rough set theory. The author stated that the defining the location of the characteristic and abstracting the characteristic value, the knowledge table and table reduction, can be ascertained by giving the characters’ two dimensional image and then, the decision rules can be deduced. Guo et al. (2016) performed discretization and differential matrix reduction on the feature matrix based on rough set theory. reduced feature vectors,