A NEAR OPTIMAL SOLUTION FOR MAXIMUM RELEVANCE MINIMUM REDUNDANCY FEATURE SELECTION

by

Nguyen Tuan Lang

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
April 2020

This thesis is dedicated to my family and loved ones.

Table of Contents

List of Tables . . . . **v**

List of Figures . . . . **vi**

Abstract . . . . **viii**

List of Abbreviations and Symbols Used . . . . **x**

Acknowledgements . . . . **xi**

Chapter 1 Introduction **1**

Chapter 2 Literature Review **5**

2.1 Filter Feature Selection Methods 5

2.1.1 Standard Filter-based Feature Selection Methods 6

2.1.2 Mutual Information-based Feature Selection Methods 8

2.2 Wrapper Feature Selection Methods 11

2.3 Binary data 12

2.4 Summary 13

Chapter 3 Methodology **15**

3.1 Introduction to Information Theoretic Approach 15

3.1.1 Entropy and Mutual Information 15

3.1.2 Maximal Dependency Feature Selection Criterion 18

3.1.3 Maximum Relevance Minimum Redundancy 19

3.2 Overview of Submodular Function Optimization 20

3.3 Proposed Method 24

3.3.1 Limitation of Max Dependency and mRMR on Submodular Feature Set 24

3.3.2 Normalized Mutual Information 25

3.3.3 Alternative Feature Selection Strategy For Binary Data 26

3.4 Summary 33

Chapter 4 Experiments and Evaluations . . . 34

4.1 Datasets . . . 34
4.1.1 Document Classification Datasets . . . 35
4.1.2 Cyber Threat on Social Network Datasets . . . 38
4.1.3 Medical diagnosis datasets . . . 40

4.2 Performance Metrics . . . 41

4.3 Classifiers . . . 43
4.3.1 Naive Bayes . . . 43
4.3.2 Support Vector Machines . . . 44
4.3.3 Decision Trees . . . 47

4.4 Experimental Evaluations . . . 48
4.4.1 Comparison with Standard Filter Methods . . . 48
4.4.2 Comparison with Mutual Information-based Methods . . . 50
4.4.3 Comparison with Feature Selection Methods for Binary Data . 57
4.4.4 Comparison with Wrapper Methods . . . 61
4.4.5 Feature Similarity Insight . . . 62
4.4.6 Performance on Non-binary Data . . . 65
4.4.7 Summary . . . 67

Chapter 5 Conclusion and Future Work . . . 72

5.1 Conclusion . . . 72

5.2 Future Work . . . 73

Bibliography . . . 74

List of Tables

4.1 An overview of the datasets used 35

4.2 Accuracy comparison with Wrapper methods 62

4.3 F-measure comparison on SMS spam (tf-idf) 67

4.4 F-measure comparison on Spambase 67

4.5 F-measure comparison on Cora 69

4.6 F-measure comparison on CiteSeer 69

4.7 F-measure comparison on WebKB 69

4.8 F-measure comparison on SMS Spam (binary) 70

4.9 F-measure comparison on Terrorist Attacks 70

4.10 F-measure comparison on Terrorists 70

4.11 F-measure comparison on SPECT 71

4.12 F-measure comparison on HIVA 71

List of Figures

1.1 Shares of feature selection studies on multimedia 1

3.1 Illustration of the submodular function in sensor deployment: a) Adding a new sensor **A'** to a small contamination area. b) Adding a new sensor **A'** to a larger contamination area 21

4.1 t-SNE mapping of Cora dataset 35

4.2 t-SNE mapping of CiteSeer dataset 36

4.3 t-SNE mapping of WebKB dataset 36

4.4 t-SNE mapping of SMS spam (binary) dataset 37

4.5 t-SNE mapping of Terrorists dataset 39

4.6 t-SNE mapping of Terrorist Attacks dataset 39

4.7 t-SNE mapping of SPECT dataset 40

4.8 t-SNE mapping of HIVA dataset 41

4.9 A simple line divides two groups of samples 45

4.10 SVM hyperplane 46

4.11 Accuracy comparison with standard filter methods on document classification datasets using SVM 50

4.12 Accuracy comparison with standard filter methods on document classification datasets using Naive Bayes 51

4.13 Accuracy comparison with standard filter methods on document classification datasets using Decision Trees 51

4.14 Accuracy comparison with standard filter methods on Cyber threat and Medical diagnosis datasets using SVM 52

4.15 Accuracy comparison with standard filter methods on Cyber threat and Medical diagnosis datasets using Naive Bayes . . . 52

4.16 Accuracy comparison with standard filter methods on Cyber threat and Medical diagnosis datasets using Decision Trees . . 53

4.17 Accuracy comparison with MI-based methods on document classification datasets using SVM 53

4.18 Accuracy comparison with MI-based methods on document classification datasets using Naive Bayes 54

4.19 Accuracy comparison with MI-based methods on document classification datasets using Decision Trees 54

4.20 Accuracy comparison with MI-based methods on Cyber threat and Medical diagnosis datasets using SVM 55

4.21 Accuracy comparison with MI-based methods on Cyber threat and Medical diagnosis datasets using Naive Bayes 55

4.22 Accuracy comparison with MI-based methods on Cyber threat and Medical diagnosis datasets using Decision Trees 56

4.23 Accuracy comparison with methods for binary features on document classification datasets using SVM 57

4.24 Accuracy comparison with methods for binary features on document classification datasets using Naive Bayes 58

4.25 Accuracy comparison with methods for binary features on document classification datasets using Decision Trees 58

4.26 Accuracy comparison with methods for binary features on Cyber threat and Medical diagnosis datasets using SVM 59

4.27 Accuracy comparison with methods for binary features on Cyber threat and Medical diagnosis datasets using Naive Bayes . 59

4.28 Accuracy comparison with methods for binary features on Cyber threat and Medical diagnosis datasets using Decision Trees 60

4.29 Similarity measure of features selected on binary document classification datasets 63

4.30 Similarity measure of features selected on binary cyber threat and medical diagnosis datasets 64

4.31 Accuracy benchmark of comparative methods on non-binary SMS spam (tf-idf) dataset using SVM 65

4.32 Accuracy benchmark of comparative methods on non-binary SMS spam (tf-idf) dataset using Naive Bayes 66

4.33 Accuracy benchmark of comparative methods on non-binary SMS spam (tf-idf) dataset using Decision Trees 66

Abstract

In many real-world applications, a high number of features could result in noisy and redundant information, which could degrade the general performance of classification tasks. Feature selection techniques with the purpose of eliminating such features have been actively studied. In several information-theoretic approaches, such features are conventionally obtained by maximizing relevance to the class while the redundancy among the features used is minimized. This is an NP-hard problem and still remains to be a challenge. This research proposes an alternative feature selection strategy on binary text representation data based on the properties of submodular functions, with the purpose of providing a theoretical lower bound for finding a near optimal solution based on the Maximum Relevance-Minimum Redundancy criterion. In doing so, the proposed method can achieve a 2-approximation by a naive greedy search. Empirical experiments validated and benchmarked against different baseline methods show that the proposed technique is a promising approach on binary data in general.

List of Abbreviations and Symbols Used

B Limited budget acted as cardinality constrain.

$H(x)$ Entropy of variable x.

$I(x_i; x_j)$ Mutual information of x_i and x_j.

$NMI(\cdot)$ Normalized Mutual Information function.

Ω A ground set of n features.

β Hyperparameter for MIFS-based methods.

$\mathbb{E}_X$ n-dimensional vector of the expectation of each element in X.

$\mathcal{F}_{\mathcal{A}}$ Sensor coverage at zone $\mathcal{A}$.

$c(s)$ Cost of a sensor s.

$d(\cdot)$ A metric distance function.

$dp(U)$ Dispersion function of elements in U.

$p(X)$ Probability mass function of random variable X.

$p(x)$ Probability of outcome variable x.

$D_{KL}(p||q)$ Kullback–Leibler divergence between two distributions p and q.

χ^2 Chi-square statistic.

CDFE Class-dependent Density-based Feature Elimination.

CHI Chi-Squared Statistic.

DF Document Frequency.

DT Decision Trees.

FE Feature Extraction.

FS Feature Selection.

GAs Genetic Algorithms.

IG Information Gain.

MIFS Mutual Information Feature Selection.

MIFS-U Mutual Information Feature Selection under Uniform Information Distribution.

mRMR Maximum Relevance Minimum Redundancy.

NB Naive Bayes.

NLP Natural Language Processing.

PSO Particle Swarm Optimization.

SBFS Sequential Backward Floating Selection.

SBS Sequential Backward Selection.

SFFS Sequential Forward Floating Selection.

SFS Sequential Forward Selection.

SVM Support Vector Machine.

Acknowledgements

Firstly, I would like to show my profound appreciation to my supervisor, Dr. Nur Zincir-Heywood, for her continuous supports with enthusiastic guidance, motivation, great ideas, and constant patience that helped me complete the research project and this thesis. I would also like to thank Dalhousie University and Faculty of Computer Science for their supports during my study and research. Last but not least, I would appreciate all members of Network Information Management and Security (NIMS) Lab, who are very knowledgeable, supportive, and humorous. These cool things helped me a lot to overcome some stressful moments.

Chapter 1

Introduction

The explosion of social networks has required the need to process information effectively. Multimedia data is an extremely rich resource for service providers spanning many areas from entertainment, education, biomedical, etc. With such a massive amount of information, data pre-processing which aims to reduce data dimensionality by retaining useful and crux features becomes one of the most important steps impacting on the task performance. This enhances the processing time, accurate prediction, and result comprehensibility to smooth the way for profoundly understanding and human insights on data [1][2].

Data pre-processing generally has two main steps including feature extraction (FE) and feature selection (FS). FE facilitates processing by extracting manageable groups of features from the raw data to generate a dataset while FS tends to select

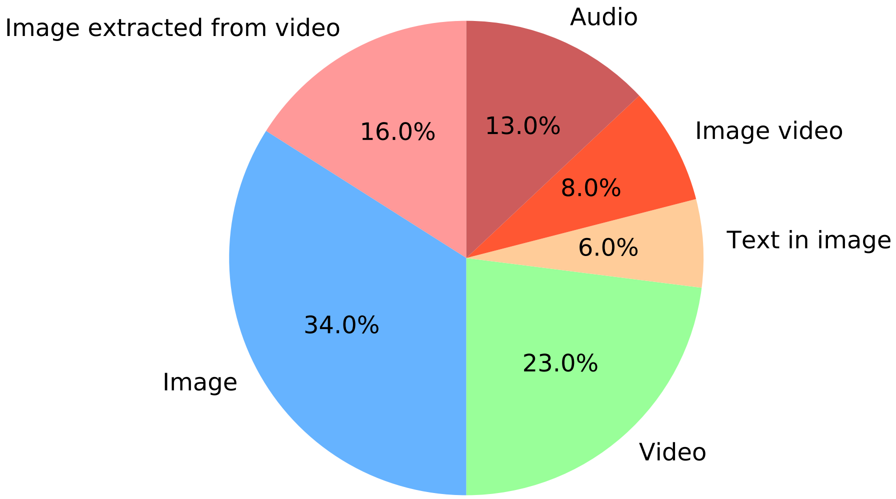


Figure 1.1: Shares of feature selection studies on multimedia

the most informative and non-redundant features from the initial set to build a model [3][4]. For example, in image steganalysis FE extracts images with undercover patterns, and FS is designed to distill salient features from these extracted images [3]. In general, the demand for feature extraction is not always as high as the feature selection, and a minimum of feature extraction is always needed. Take deep learning neural networks as an exemplar, we do not need to perform feature extraction when the neural network model can get a low dimensional representation of high dimensional input data by itself. Meanwhile, feature selection is always taken into consideration in order to facilitate interpretability and computing feasibility, or to avoid the case of a large number of irrelevant features describing not enough data.

The objective of FS is to select the smallest subset of features given a certain regularization, or alternatively finding the desired feature subset with a minimum generalization error. There are numerous feature selection methods being proposed to adapt the need of data processing in the information explosion era. Since 2000, approximately 25,912 and 245,796 articles were found on the IEEE Xplore[1] and ACM Digital Library[2] database when the key word "feature selection" was searched on January 2020. It can obviously be seen that feature selection has been a hot topic in pattern recognition over the last 20 years. Some fields required FS is illustrated by related works collected for a survey on multimedia in [5] (Figure 1.1), it seems that feature selection problems in Image and Video acquire the most attention from the research circle.

Generally, there are three families of feature selection methods which are categorized based on the association of their selection algorithm and the learning method [7]. Filter methods select features according to their statistical characteristics. These methods are usually effective in terms of computational cost (time) and resistant to overfitting. Wrapper methods select desired features by repeatedly evaluating performance over each possible subset for a selected classifier. By this way, wrapper methods generally outperform filter methods but they also have a high risk of overfitting and a computational cost. Embedded methods, finally, draw on advantages of

[1] https://ieeexplore.ieee.org/Xplore/home.jsp

[2] https://dl.acm.org/

both the filter and the wrapper models by optimizing the interaction between variables and the learning process. In this case, filter methods are usually used as the preprocessing step. The most typical embedded technique is decision tree algorithm.

Maximum Relevance - Minimum Redundancy (mRMR) feature selection is a well-known mutual information based-method which is a form of the filter feature selection family, originally introduced by Ding and Peng [27] with the aim of enhancing the degree of phenotype classification accuracy by choosing genes capturing abundant but salient characteristics of them. The original mRMR method tends to select features having the strongest relevance to the class vector while minimizing redundancy among selected features. By a simple heuristic approach, mRMR has shown its effectiveness in many classification tasks. However, it has some limitations. First, although feature selection is a simple but formidable strategy for dimensionality reduction and has proven to work well in text classification task [6][15], there are not many mRMR-based works dedicated to text mining. A tremendous amount of unstructured text data in the forms of emails, web pages, or social media posts places a heavy burden on text classification, which is a fundamental task in Natural Language Processing (NLP). A document may contain hundreds to thousands of unique words represented as features, but there is a small percentage of them that contributes to making a significant difference in the performance for text classification. Hence, an effective method to reduce such redundant features is essential. Second, there is still no evidence to theoretically approximate the optimal solution for its greedy objective function. This limitation is mainly due to the lack of the monotone submodular property of the objective function when solving by a greedy algorithm.

In this thesis, a novel approach that adapts the properties of submodular functions to determine a bound for the approximate solution by the greedy algorithm is proposed. The new scheme is validated and benchmarked with common filter methods such as DF, IG, CHI, ReliefF, and wrapper methods including SFS and SFFS on text classification tasks. The performance of the proposed method is also compared with mRMR-based methods to analyze how these methods work on text data.

The contributions of this thesis can be summarized as follows:

- Reviews of common feature selection methods and their limitations in some situations that can be addressed in the literature.

- This thesis proposes a new feature selection strategy which can be considered as a modified version of the traditional mRMR. At first, the common relevance measure is modified , then a distance metric function as the similarity measure for the selected set is applied to create a new objective function. In doing so, the proposed method could circumvent the limitation of the feature searching process of common MI-based methods by providing a 2-approximation solution for the proposed greedy search.

- Proposed methods, namely aNMI-DIST and sNMI-DIST, therefore can gain prediction accuracy and reduce a considerable amount of redundancy among features used compared to filter feature selection methods and wrapper feature selection methods for binary text representation. Performance on various datasets in social networks and medical diagnosis is also investigated to verify the efficiency of the proposed methods on binary data in general.

The rest of thesis is organized as follows. Chapter 2 summarizes several closely related works and concisely explains usual feature selection methods employed in previous studies which will be used as baselines. Chapter 3 elaborates the maximum relevance-minimum redundancy criterion and proposes adjustments to the feature selection strategy with the purpose of effectively utilizing the correlation between features and labels. The empirical results on different text datasets are validated and analyzed in Chapter 4. Finally conclusions and future works are discussed in Chapter 5.

Chapter 2

Literature Review

In practice, a high number of features could result in noisy and redundant information, which could degrade the general performance of classification tasks. Feature selection techniques with the purpose of eliminating such features have been actively studied. In this chapter, related works on various methods of filter feature selection family will be reviewed in Section 2.1, followed by Section 2.2 that provides an overview of wrapper techniques. Finally, summary of approaches mentioned in the literature is presented with current limitations.

2.1 Filter Feature Selection Methods

In high-dimensional datasets, it is essential to filter out the most redundant features by eliminating a subset of irrelevant features in order to avoid the overfitting problem and tackle the curse of dimensionality. Authors in [8] compared the classification accuracy of filter methods on malware detection data. A new filter feature selection method [9] was proposed to compare with established methods for cancer prediction. Authors in [10] developed an ensemble filter method that combines specific scores of several filter methods and then compare it to a single filter method. Likewise, Ghosh et. al [11] proposed a 2-stage model combining of filter methods such as ReliefF, or Chi-square, and genetic algorithm to get the fine-tuned result for feature selection in microarray datasets. An extensive comparison of 22 filter feature selection methods on high-dimensional data investigated in [12] showed that although some filter methods perform well on certain datasets, there is no group of these methods that is always superior to all other methods. To contribute to the expansive review on filter methods, this section focuses on two main approaches in the filter feature selection family: standard filter-based methods and mutual information-based methods.

2.1.1 Standard Filter-based Feature Selection Methods

The standard filter-based ranking methods evaluate a feature-goodness criterion based on a certain threshold, then decide which features should be retained or eliminated. Common standard techniques that will be reviewed in the literature are Document Frequency (DF), Mutual Information (MI), Information Gain (IG), Chi-square (χ^2) statistic (CHI), and ReliefF [13]. Note that in this literature MI method is grouped into the standard filter-based feature selection method. A comprehensive investigation of the most classic filter-based methods with different classifiers was conducted in [14] to analyze insightful impacts of feature selection algorithms. One of the earliest comparative feature selection studies for text classification [15] found the effectiveness of IG, CHI, and DF and they are strongly correlated to common terms while MI tends to bias rare terms which are sensitive to probability estimation errors. In general, the χ^2 statistic and IG feature selection method usually achieve bright results when compared with MI and DF thresholding on various text classification datasets with different classifiers [16][17], especially CHI often acts as a competitive baseline for imbalanced text data [18].

Document Frequency (DF)

Document frequency measures how relevant a document is to individual terms in a corpus. This is the simplest but reliable technique to reduce the feature dimension in text classification and is on par with IG or CHI in some tasks despite much lower running costs. With the purpose of discarding sporadic terms which are not significantly useful for prediction performance, a term contained in documents will be selected by the highest estimated probability, for example, term t_k appears in any document D_i in the corpus

$$DF(t_k, D_i) = P(t_k|D_i). \tag{2.1}$$

Document frequency is justified to be more efficient under the Bernoulli model [19]. Also, features with low DF are assumed to have relatively informative, therefore these features could not be aggressively removed. The investigation in [15] shows that DF should be considered as an ad hoc approach to enhance the prediction performance rather than a principal method.

Mutual Information (MI)

Mutual Information is a common criterion to find the most relevant features to the class vector. The expected mutual information of term t_k and class c_i measures how much information t_k contributes to accurately predicting c_i. Formally,

$$I(t_k; c_i) = \log \frac{P(t_k, c_i)}{P(t_k)P(c_i)}. \tag{2.2}$$

This criterion assumes terms with higher ratios effectively contribute to making the correct classification decision. A limitation of MI is that $I(t_k, c_i)$ tend to be influenced by the marginal probabilities of terms in which rare terms may have higher MI score than common terms.

Information Gain (IG)

Information gain can be seen as a principal criterion used in Decision Trees classifier [20]. IG is a similar approach to mutual information by taking the average of mutual information measured. The information gain of term t_k in the class vector c of n elements $c = \{c_1, c_2, ..., c_n\}$ is defined as

$$\begin{aligned} IG(t_k) = -\sum_{i=1}^{n} P(c_i) \log P(c_i) + P(t_k) \sum_{i=1}^{n} P(c_i|t_k) \log P(c_i|t_k) \\ +P(\overline{t_k}) \sum_{i=1}^{n} P(c_i|\overline{t_k}) \log P(c_i|\overline{t_k}), \end{aligned} \tag{2.3}$$

in which $P(\overline{t_k})$ corresponds to the probability of t_k to not appear in a document. By this way, IG tends to select terms with a large number of distinct values. In the context of text classification, the common term contributing a trivial discriminating influence has a high possibility to become a candidate of the desired feature set. The time complexity of IG is the same with $I(x_i; x_j)$, as $O(Vn)$ where V is the size of the feature set.

Chi-squared statistic (CHI)

The χ^2 statistic is a kind of statistical hypothesis testing for the purpose of measuring how far observed data is different to the expected result. In text processing, the association of term t_k and class c_i is calculated by

$$\chi^2(t_k, c_i) = \frac{\left[P(t_k, c_i)P(\overline{t_k}, \overline{c_i}) - P(t_k, \overline{c_i})P(\overline{t_k}, c_i)\right]^2}{P(t_k)P(\overline{t_k})P(c_i)P(\overline{c_i})}. \tag{2.4}$$

Terms with the highest values for the chi-squared test are most likely to be relevant to the class therefore they should be selected. One weakness of CHI is that if a term rarely appears in the corpus, the normalization in Equation 2.4 could not longer obey to the χ^2 distribution. As a consequence, terms with a low-frequency may be not reliable in the context of χ^2 statistic [21].

ReliefF

ReliefF is a best known variant of the original Relief feature selection method that is first proposed in [22][23]. Relief was originally designed for only binary classification problems. For a dataset with n samples and m features, weights of features that are assigned as a zero vector W_0 will be iteratively updated by the following expression:

$$W_{i+1} = W_i - (x_i - H_i)^2 + (x_i - M_i)^2, \tag{2.5}$$

At each cycle, Relief constructs two nearest neighborhoods of the target sample x_i. A group of samples in the same class with x_i is called the nearest hit (H) and samples out of this class are in the nearest miss (M) group. Then Relief updates the weight of a feature by scoring the difference between this feature and nearby samples of the same class and the other class. The relevance vector will be obtained by dividing each element of the weight vector by k after k iterations. A feature x_i having ${w^i}_k \in W_k$ becomes a candidate if ${w^i}_k \geq \tau$, in which τ is a relevance threshold.

ReliefF [13] was introduced to overcome the major limitation of its predecessor when Relief has to decompose the multi-class problem into multiple binary class problems. Generally, ReliefF finds k near misses from each different class and k nearest hits for the same class instead of a single hit and miss as in Relief. When increasing the number of miss/hit instances, the average difference score is more reliable and resistant to noisy data.

2.1.2 Mutual Information-based Feature Selection Methods

The intuition behind a good set of features selected is the strong correlation between features and the class label that is commonly measured by mutual information. This section will review several techniques based on this criterion. A simple approach which was reviewed in the previous section is ranking the mutual information scores calculated between an individual feature and the class label in order, which was referred as *Mutual Information Maximization* by Lewis in study [24]. More particularly, a feature x_i having high $I(x_i; c)$ with class vector c should be considered as a strong candidate for the feature subset. However, this approach only counts on the interaction between the class vector and a feature itself. The ideal objective should consider the correlation between the candidate feature subset S_m and the class vector c, that is $I(x_i; x_j)$. In practice, the computational cost for the exact solution of this function is intractable when the number of features becomes large. Authors in [25] proposed a scheme called *MIFS* which employed mutual information of feature-to-class, and feature-to-feature to identify the desired feature subset, while removing the features that are not relevant to the class. A new feature is selected by maximizing the following expression.

$$f_{MIFS}(x_i) = I(x_i; c) - \beta \sum_{x_i, x_j \in S_m} I(x_i; x_j), \tag{2.6}$$

where $\beta \in [0; 1]$ is a penalized hyperparameter on the redundancy term. If $\beta = 0$, only relevance between features and the class label is considered. When β increases, this measure is deducted by a quantity of mutual information among features selected as the redundancy term has a tendency to influence the relevance term. Investigations in [25] found that β works well in the range of 0.5 to 1 for many classification tasks.

One weakness of MIFS is that when β becomes too large, the redundancy term also tends to overshadow the correlation between features and the class vector. The study in [26] showed that if a feature is closely associated to the already selected features, performance of MIFS could be degraded. To circumvent this limitation, the authors supplanted the original idea of MIFS by an alternative objective function which takes into account the conditional mutual information between a new feature and the feature already chosen. The formal expression of this algorithm called MIFS-U is given as Equation 2.7.

$$f_{MIFS-U}(x_i) = I(x_i; c) - \beta \sum_{x_i, x_j \in S_m} \frac{I(x_j; c)}{H(x_j)} I(x_i; x_j), \tag{2.7}$$

where $H(x)$ is the entropy of variable x. MIFS-U was shown to be outperformed MIFS on many datasets investigated with the same computational complexity.

The performance of aforementioned methods is contingent on the selection of β, which is not an easy task. The same idea with MIFS and MIFS-U, mRMR [27] was also proposed to adapt the effect of redundancy. Instead of penalization by the hyper-parameter, this method normalizes the redundancy term by the number of features selected to balance relevance and redundancy when the large number of features could cause the increase in magnitude with respect to the redundancy term. By a simple greedy algorithm, mRMR has demonstrated competitive power in classification tasks, mainly in the bioinformatics applications [29]-[31], and for both discrete and continuous domains [28]. The original mRMR method tends to select features having the strongest relevance to the class vector while minimizing redundancy among selected features. Although this strategy empirically achieves considerable performance, there is still no evidence to theoretically approximate the optimal solution for its objective function. This limitation is mainly due to the lack of the monotone submodular property of the objective function when solving by a greedy algorithm. Authors in [33] implemented redundancy reduction based on mRMR for text classification and showed that their method outperformed others on the Reuters news datasets. The investigation in [34] showed that the mRMR-based method could proficiently boost performance of taxonomic classifiers. By comparison to the information gain (IG) method, mRMR achieved better results on several sentiment classification datasets [35].

There is a huge family of existing feature selection algorithms that are based on mutual information approaches. Algorithms in this family apply various heuristic filter criteria to gauge the importance of features. Though the aforementioned MI-based methods can exploit the feature redundancy that the standard filter methods fail to tackle, they are similar to these standard methods in terms of the independency of any learning processes. However, most of existing MI-based feature selection methods can only work in a supervised scenario. It means that without the knowledge of the class, the relation between features and the class is obscured. In addition, these methods can only perform on discrete data. For continuous numerical variables, the discretization preprocessing steps are required beforehand [36].

2.2 Wrapper Feature Selection Methods

Wrapper methods adapt particular learning techniques to evaluate the performance of a candidate feature subset. In general, these methods train the machine learning classifier on a single feature or a batch of features, then they select features that produce the highest classification accuracy. Nevertheless, they are limited to low dimensional data as an intensive computation is required when evaluating performance at every training cycle. One of common wrapper methods is a family of Genetic algorithms (GAs) for feature selection. GA was firstly introduced by John Holland in 1960. However, the first publish adapted for artificial systems was formally released in [39]. GA is a random search strategy when it randomly operates coding the population of candidates called chromosomes into binary sequences, in which where a digit embodies a gene. In general, GA includes three main steps during each iteration. Selection, Crossover, and Mutation. The operation is repeated until reaching a termination condition. There are many studies on feature selection using GAs such as [40]-[43]. However, finding the optimal solution to the high-dimensional population of genes requires very expensive computations. Particle Swarm Optimization (PSO) [44] feature selection method shares many resemblances with GAs. For example, they are both population-based search strategies and rely on information sharing among their population members to enhance their search processes using a combination of deterministic and probabilistic rules [45]. A major advantage of PSO over GA is that PSO requires less computational effort to derive such high quality solutions as GA because PSO is easy to implement and there are few parameters to adjust and less number of function evaluations than GA.

Sequential feature selection algorithms are also a family of wrapper methods. The motivation behind this approach is to sequentially select a subset of features that is most relevant to the problem by using the classifier to evaluate each subset in the whole feature set. Overall, sequential methods add or remove a feature at each iteration according to the performance of the classifier used until a feature subset of the desired size k is reached [46]. There are four common variants of sequential searching methods, namely Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), Sequential Forward Floating Selection (SFFS), and Sequential Backward Floating Selection (SBFS). In SFS, the algorithm commences with an empty set, then it selects a feature that allows to reach the best classification accuracy to add to the empty set. This procedure is repeated until the desired number of features is obtained. In contrast, SBS starts with the whole feature set, then performs relatively the same as SFS except that it eliminates features that degrade the overall performance. The obvious limitation of SFS and SBS is that once a feature in collected/eliminated, it could not be revised in the following steps. By this reason, the performance of the overall selected subset is not evaluated. The floating variants (SFFS and SBFS) are developed to combine the inclusion and exclusion process. Intuitively, these variants should gain the overall performance but improvement is only emphasized if the previous selected subset is asserted as “good”.

Generally, wrapper methods are usually slower than filter methods, but a comprehensive study in [47] that reported the comparison on evolutionary feature selection strategies shows that wrapper methods using a simple classification algorithm can be faster and often attain better classification performance than filter methods. In recent years, several studies working on wrapper methods have been proposed aiming at achieving a better classification performance and lower computational complexity such as recursive feature elimination [48] or using evolutionary and swarm intelligence algorithms for FS [49].

2.3 Binary data

Binary feature representation has been found in a wide range of applications such as text mining, handwriting classification, medicine domains, or biometrics matching applications [37][50][51]. Features in many types of data can be represented by a binary variable. For instance, in medicine domains where each attribute represented as a symptom can be triggered as 1 if a sample (e.g. a patient) has the abnormal clinical manifestation of this symptom, and vice versa it is switched to 0. Another example in text mining, the absence of a word in the document can be modeled by bit 0, while this bit will be fired to 1 if the feature appears in the document. Especially in some document classification tasks, TF-IDF representation may underperform binary representation and even consume much more time to process in high dimensional data. Besides, mutual information for binary data is much easier to compute compared to TF-IDF values. In image processing, binary normalization can reduce data transmission costs and processing costs (including mathematical transformations, quantization, etc.).

2.4 Summary

Feature selection has been actively studied in pattern recognition domains. However, there are several problems that need to be considered.

- Filter feature selection methods work well in classification task, especially in text mining. One advantage of these methods is computationally effective. However, standard methods such as IG or CHI are usually influenced by the feature frequency. Moreover, they generally do not consider the effect of redundancy so their performance on the selected feature subset may be deteriorated when there are a considerable number of redundant features in the dataset.

- Mutual information-based methods are a kind of filter feature selection family. They tend to choose the most informative features while eliminating redundant features. MIFS, MIFS-U and mRMR are popular MI-based techniques and they have been shown to perform well in many tasks. However, the problem of optimizing the maximum relevance minimum redundancy objective remains to be challenged.

- Wrapper methods take advantage of classifier training on each feature subset candidate to enhance the overall performance. Although these methods perform better than filter methods, they require expensive computational costs to evaluate the performance at every iteration.

- Despite a significant number of feature selection techniques have been proposed, a relatively small portion of them are applied for relatively short text categorization [32]. Also the maximum relevance minimum redundancy feature selection method and its variants have been proposed in many applications; however, a very limited number of mRMR versions are dedicated to NLP domains.

- Binary features are represented and shown the effectiveness in many real-world datasets. Nevertheless, feature selection methods designed for binary data are limited.

In an effort to partly find a way to overcome these limitations, this thesis proposes a new feature searching scheme based on MI-based methods on text data, then providing a theoretically approximate solution to the problem of maximum relevance-minimum redundancy by drawing on properties of binary data. Finally, experiments will be examined on several datasets in different domains such as social networks or medical diagnosis to verify the method proposed.

Chapter 3

Methodology

The primary orientation of this research is to propose a novel feature selection method that can theoretically provide a near-optimal solution to the problem of optimizing an objective function combined of maximizing the feature-class relevance criterion and feature-feature redundancy criterion, which is considered to be NP-hard. mRMR-based feature selection methods have been used and developed to perform classification tasks. Although they have proven to work well under various conditions of data and classifiers, finding a theoretical bound for an approximate solution by using heuristic searching still remains to be challenged. To this end, a new objective function is designed to overcome this limitation by exploiting the advantage of submodular functions under certain conditions. This section reviews the information theoretic approach of Maximum Relevance - Minimum Redundancy criterion, and offers an alternative feature selection strategy using the submodular property for binary data. Justifications show that the proposed method can derive a 2-approximation solution by a naive greedy search. In the following, the introduction of fundamental information theoretic concepts will be presented in Section 3.1. Section 3.2 reviews definitions and properties of submodular functions. The proposed method is described in Section 3.3.

3.1 Introduction to Information Theoretic Approach

3.1.1 Entropy and Mutual Information

This subsection introduces fundamental concepts from information theory which will be using in the remaining of the thesis. For the sake of simplicity and matching to the proposed method, definitions are only presented on the discrete domain.

Entropy

Entropy $H(X)$ of a random variable X and possible outcome x with probability

$p(x)$ having probability mass function $p(X)$, is common measured by uncertainty quantity as follow

$$H(X) := \mathbb{E}_X\left[p(X)\right] = -\sum_{x \in X} p(x) \log p(x). \tag{3.1}$$

where $\mathbb{E}_X$ is the n-dimensional vector of the expectation of each element in X. Given two mass distribution p and q defined on the same probability space, the divergence between two distributions determined by Kullback–Leibler divergence $D_{KL}(p||q)$ as

$$D_{KL}(p||q) := \mathbb{E}_X\left[\frac{p(X)}{q(X)}\right] = -\sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}. \tag{3.2}$$

Properties of Entropy:

1. A zero-probability event does not affect to entropy:

$$H(p_1, p_2, ..., p_n, 0) = H(p_1, p_2, ..., p_n).$$

2. The entropy of a discrete random variable is non-negative:

$$H(X) \geq 0.$$

3. If X and Y are two independent random variables, knowledge about Y doesn't impact on knowledge about X:

$$H(X|Y) = H(X).$$

4. Conditioning reduces entropy:

$$H(X|Y) \leq H(X).$$

5. The sum of entropies of two random variables is always larger or equal to the entropy of two variables occurring together:

$$H(X) + H(Y) \geq H(X, Y).$$

6. Chain rule:

$$H(X, Y) = H(X|Y) + H(Y) = H(Y|X) + H(X).$$

Mutual Information

The mutual information of two random variables measures the dependence between them. More precisely, it measures how much knowing one of these variables reduces uncertainty about the other. In case of discrete space, the mutual information of two jointly discrete random variables X and Y is calculated by

$$\begin{aligned} I(X;Y) :=& \; E_X\left[D_{KL}(p(Y|X)||p(Y)\right] \\ =& \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\tfrac{p(x,y)}{p(x)p(y)} \end{aligned} . \tag{3.3}$$

Properties of Mutual Information:

1. Mutual information is non-negative:

$$I(X;Y) \geq 0.$$

2. Mutual information holds symmetric property:

$$I(X;Y) = I(Y;X).$$

3. Conditional mutual information:

$$I(X;Y|Z) = \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} p(x,y,z)\log\frac{p(x,y,z)p(z)}{p(x,z)p(y,z)}.$$

Relationship between Mutual Information and Entropy

Following common relationships between mutual information and entropy are introduced to facilitate the proposed feature selection mechanism presented in the later sections.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ I(X;Y) &= H(Y) - H(Y|X) \\ I(X;Y) &= H(X) + H(Y) - H(X,Y) \\ I(X;Y) &= H(X,Y) - H(X|Y) - H(Y|X) \\ I(X;X) &= H(X) \end{aligned}$$

3.1.2 Maximal Dependency Feature Selection Criterion

A scheme which tends to find a feature set jointly having the maximum dependency to the class vector was proposed in [28]. This is formally expressed by maximizing the joint mutual information function between m-selected discrete feature set S_m in feature space Ω and class c as the lemma below

Lemma 3.1

$$\arg\max_{S_m\subseteq\Omega} I(S_m;c) = \arg\max_{S_m\subseteq\Omega} \sum_{k=1}^{m} \sum_{X\subseteq S_m;|X|=k} (-1)^{k+1} I_{k+1}(X;c). \tag{3.4}$$

The proof of Lemma 3.1 can be done by Lemma 3.2 as follow

Lemma 3.2

$$H(S_m) = \sum_{X\subseteq S_m} (-1)^{|X|+1} I_{|X|}(X). \tag{3.5}$$

The proof of Lemma 3.2 can be found in [68], pp. 1049.

Now we are ready to prove Equation 3.4.

Proof.

$$\begin{aligned} I(S_m;c) &= H(S_m) + H(c) - H(S_m,c) \\ &= \sum_{X\subseteq S_m} \Big((-1)^{|X|-1} I_{|X|}(X) + I(c)\Big) - \sum_{X\subseteq S_m} (-1)^{|X|-1} I_{|X|}(X) \\ &= \sum_{X\subseteq S_m} (-1)^{|X|+1} I_{|X|+1}(X;c) \\ &= \sum_{k=1}^{m} \sum_{X\subseteq S_m;|X|=k} (-1)^{k+1} I_{k+1}(X;c) \end{aligned} \quad . \tag{3.6}$$

The proof is completed.

The problem in Equation 3.4 becomes intractable when the computational cost increases exponentially with respect to k. For example, each feature simply has only two states in a binary dataset with N instances, then a set of m features could have a maximum $\min(2^m, N)$ joint states. It can be easily seen that the mutual information cannot be properly calculated when the number of joint states increases very rapidly as the number of samples. This problem even gets exacerbated in cases of multivariate discrete features, or continuous feature variables. For this reason, Max Dependency feature selection is not suitable for applications where the number of feature categories/values is significant or the cardinality of samples is not very large.

3.1.3 Maximum Relevance Minimum Redundancy

Strong relevance between features and a class vector can be indicated by a high correlation of the target space to the classification variables. Some of the features, however, do not significantly contribute to the overall performance, on the contrary degrade the proficiency for distinguishing classes. For example, in the context of text categorization, two terms *"beach"* and *"sea"* in a document may have a high possibility of falling into the *"ship"* category. But obviously, it is not necessary to have both of these words to increase discrimination accuracy.

Max Dependency criterion previously shows several limitations: (1) Maximizing mutual information $I(S_m; c)$ is only to select mutually independent features having strong relevance to the target. This does not suffice for a desired set high discriminative features. (2) Calculating the accurate joint mutual information between feature subsets and the class could be impossible for big data. To avoid these restrictions, mRMR feature selection method was proposed to select features which have largest dependency on the class vector (Max Relevance), while eliminating redundancy among selected features as much as possible (Min Redundancy). Also, the mutual information of a feature subset and the class vector as in Equation 3.4 is approximated to the average value of all mutual information values between individual features and the class, which will be discussed in Section 3.3.

By treating the max relevance and min redundancy criterion evenly, mRMR tends to optimize the combined objective function of two aforementioned criteria over a selected set S_m:

$$f(S_m) = \frac{1}{|S_m|} \sum_{x_i \in S_m} I(x_i; c) - \frac{1}{|S_m|^2} \sum_{x_i, x_j \in S_m} I(x_i; x_j), \tag{3.7}$$

where $I(\cdot)$ is the mutual information function, x_i is a feature and c is the category vector. The first term represents the average relevance of features in the selected set to the target class, while the second term acts for redundancy among these features. If we stop at finding a set of features that stops at satisfying the maximization of the first term, the problem construes as Max-Relevance feature selection with the relevance objective function $\sum\limits_{x_i \in S_m} I(x_i; c)$.

When the exact solution to the problem of finding a set of desired features among

N features in a dataset requires $O(N^{|S_m|})$ searches in the worst case, which is impractical for large datasets, a near-optimal solution can be reached by a heuristic algorithm. After selecting an initial feature having the highest mutual information score to the class vector, the selected feature set iteratively picks a new feature, assuming the m^{th} feature, so that the following condition is maximized:

$$\underset{x_j \in \Omega \setminus S_{m-1}}{\arg\max} \left(I(x_j; c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_i; x_j) \right), \tag{3.8}$$

with the assumption that the feature set S_{m-1} with $m-1$ features has already been selected. The computational complexity of this incremental search is $O(|S_m|.N)$.

3.2 Overview of Submodular Function Optimization

This section reviews fundamental concepts on a special class of set functions called *submodular function*. Submodular functions hold properties which could be comparable to convex functions although they are defined as concave functions. So the problem of optimizing a convex or concave function can also be considered as the problem of maximizing or minimizing a submodular set function. For the sake of visualization, take the problem of sensor placements in [56] as an example. Consider the problem of keeping the water distribution systems out of contamination, suppose $\mathcal{F}_{\mathcal{A}}$ measures the number of people being safe by placing sensors at zone $\mathcal{A}$. Different locations where contaminants are accidentally/deliberately released may affect different figures of people. It means that contaminants at some areas may spread faster or broader than at others, so that choosing appropriate locations to initially deploy sensors can facilitate earlier contamination detection. A good set of sensor placements $\mathcal{A}$ may detect contamination early, helping people being protected so the value of $\mathcal{F}_{\mathcal{A}}$ will be high. By contrast, a poor set of $\mathcal{A}'$ may get low $\mathcal{F}_{\mathcal{A}'}$. The illustration in Figure 3.1 indicates the diminishing returns effect of submodular functions when deploying sensors into contamination areas. A new sensor with coverage **A'** (brighter region) adding to the smaller area can gain the contamination detection coverage as in Figure 3.1a, while placing this sensor into an area where many sensors are already deployed to detect contamination (darker regions) may be less useful because there is more overlap (Figure 3.1b).

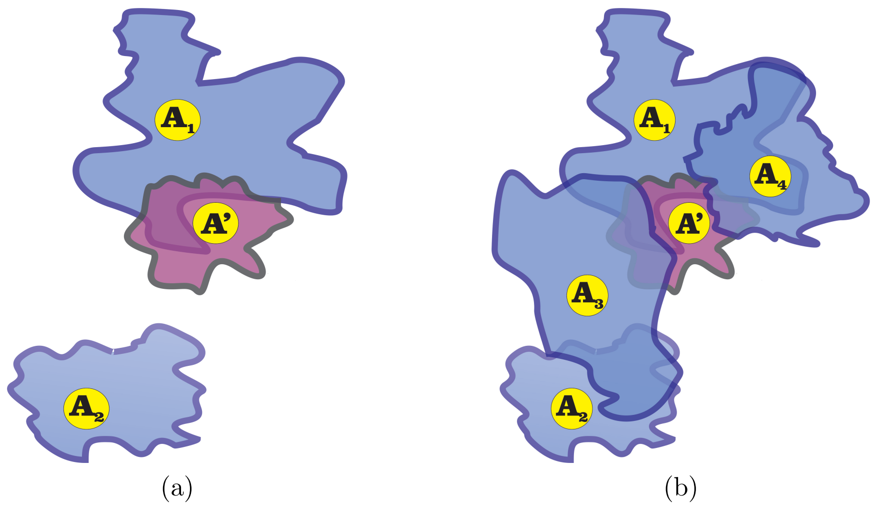


Figure 3.1: Illustration of the submodular function in sensor deployment: a) Adding a new sensor **A'** to a small contamination area. b) Adding a new sensor **A'** to a larger contamination area

In general, the problem of sensing coverage optimization is to find the best possible set of sensor placements $\mathcal{A}^*$ satisfying the optimum condition $\mathcal{A}^* = \arg\max_{\mathcal{A}} \mathcal{F}(\mathcal{A})$. We can intuitively maximize $\mathcal{F}_{\mathcal{A}}$ by placing sensors at every feasible location. In reality, however, costs for deploying sensors at different locations or environments are heterogeneous. For example, introducing sensors into a water pipe may be much more exorbitant than placing sensors at an available infrastructure, or fewer expensive sensors deployed may perform better a significant number of cheaper sensors. The problem of sensing optimization becomes the problem of optimizing $\mathcal{F}_{\mathcal{A}}$ under *cardinality constraint*: suppose we want to find an as good as possible set of sensor placements in the whole set of sensors $\mathcal{V}$ so that the number of sensors does not exceed k, the problem can be defined as

$$\mathcal{A}^* = \underset{\mathcal{A} \subset \mathcal{V}, |\mathcal{A}| \leq k}{\arg\max}\, \mathcal{F}(\mathcal{A}). \tag{3.9}$$

Let $c(s)$ be a cost of a sensor s, obviously the cost of a set of sensors $\mathcal{A}$ can be summed up by the individual costs of element sensors, $C(\mathcal{A}) = \sum_{s \in \mathcal{A}} c(s)$. If we have a limited budget, B, to spend, then our constraint is equivalent to $C(\mathcal{A}) \leq B$.

There is no evidence of an efficient algorithm to solve such problem exactly. Even the very simple problem in Equation 3.9 can lead to an NP-hard solution [52]. However, the answer could be approximated by a naive heuristic approach: *greedy algorithm*. This approach can be done thanks to a very interesting property of submodular functions, called *diminishing returns*. Formal definitions and remarks of submodular functions will be given in the following part.

Let Ω be a ground set of n elements. For a set $S \subseteq \Omega$ and an element $v \in \Omega$, we denote $S \cup \{v\}$ as $S + v$ and $S \setminus \{v\}$ by $S - v$. The next equation expresses a very common definition of submodular functions. Note that symbols and notations in the rest of this thesis related to the submodular property will be following up by its definitions and corollaries unless stated otherwise.

Definition 3.1 *A set function $f : 2^{\Omega} \to \mathbb{R}$ is submodular if for every subset $S, T \subseteq \Omega$ with $S \subseteq T$ and every $v \in \Omega \backslash T$, we have that:*

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T). \tag{3.10}$$

Definition 3.1 illustrates the diminishing returns property of submodular functions. More specifically, adding an element to a larger set produces smaller marginal increase in the value of f. This property of submodular functions is very powerful in modeling returns of investment, accuracy of a learning algorithm, etc. [69]

Definition 3.2 *A submodular function $f : 2^{\Omega} \to \mathbb{R}$ is non-negative if $f(S) \geq 0$ and monotone if $f(S) \leq f(T)$ for any $S \subseteq T \subseteq \Omega$.*

Corollary 3.1 *Let $f, g : 2^{\Omega} \to \mathbb{R}$ be submodular set functions, and $S, T \subseteq \Omega$ be subsets of Ω. Then,*

1. *$h(S) = c \cdot f(S)$ is a submodular function for every $c \in \mathbb{R}^{+}$.*

2. *$h(S) = f(S) + g(S)$ is a submodular function.*

3. *$h(S) = f(S \cap T)$ is a submodular function.*

4. *$h(S) = f(\Omega \setminus S)$ is a submodular function.*

5. *$h(S) = \min\{f(S), c\}$ is a monotone submodular function if f is monotone and $c \in \mathbb{R}^{+}$.*

6. $\sum_{v \in T} f(v|S) \geq f(S \cup T) - f(S)$.

Functions holding the submodular property have been showing their power in both theoretical and practical applications by heuristic schemes. A naive greedy algorithm performs surprisingly well in the problem of submodular function optimization thanks to the diminishing returns property. Empirical observations indicate that it is difficult to find out an algorithm which performs better than the greedy algorithm for maximization problems alike in Equation 3.9. A near-optimal solution guaranteed in the worse case for the problem in Equation 3.9 can be found in [70], which is revised as Theorem 3.1 below.

Theorem 3.1 *Let $S_k \subseteq \Omega$ is a set of k elements chosen by a greedy algorithm and a monotone submodular function $f : 2^\Omega \rightarrow \mathbb{R}$, then*

$$f(S_k) \geq \left(1 - \frac{1}{e}\right) f(S^*), \tag{3.11}$$

where $f(S^*) = \underset{S \subseteq \Omega, |S|=k}{\arg\max} f(S)$.

A naive greedy algorithm can begin with an empty set, then add elements with the largest marginal increase over the current candidate set. The process will iterate until the number of selected elements meets the cardinality constrain. The procedure of the greedy search is given as Algorithm 1.

Algorithm 1 Greedy Algorithm(Ω, f, k)

```
INPUT: A ground set $\Omega$, cardinality constrain $k$ .
OUTPUT: A subset $S_k \subseteq \Omega$.
$S_0 \leftarrow \emptyset$
for $i = 1$ to $k$ do
    $v_i \leftarrow \underset{v \in \Omega \setminus S_{i-1}}{\arg\max}(f(S_{i-1} \cup \{v\}) - f(S_{i-1}))$
    $S_i \leftarrow S_{i-1} \cup \{v_i\}$
end for
Return $S_k$
```

3.3 Proposed Method

The previous section revised the submodular property as well as its power in finding a near-optimal solution for maximizing monotone submodular functions. This section will elaborate limitations of objective functions shown in Max Dependency feature selection criterion and Maximum Relevance Minimum Redundancy method when applying the submodularity property, then propose an alternative feature selection strategy which can overcome these restrictions and provide a theoretical bound of 2-approximation by a naive greedy search specifically on binary data.

3.3.1 Limitation of Max Dependency and mRMR on Submodular Feature Set

Beside the intensively computational cost to solve the problem of maximizing Max Dependency criterion in Equation 3.4, it is very hard to apply a greedy algorithm to this objective function. Proposition 3.1 shows that Max Dependency criterion cannot meet the submodular property to guarantee a near-optimal solution when applying a greedy algorithm.

Proposition 3.1 *Mutual information of a set of dependence features is not submodular.*

Consider an example when two binary features X_1 and X_2, and a binary classification $Y = X_1 \oplus X_2$. Obviously, self-information of individual features in predicting Y is 0 as each feature cannot predict Y without another. However, the set these two features can exactly classify Y. So that $0 = f(\{X_1\}) - f(\emptyset) \leq f(\{X_1, X_2\}) - f(X_2) = 1$ violates submodular properties, with $f(\cdot)$ is the mutual information function.

In general, Proposition 3.1 indicates that greedily selecting features could not warrant the largest marginal benefit $\Delta_v(S) = f(S \cup \{v\}) - f(S)$ of adding variable v to the selected feature set S, where f is the mutual information function. Although Lemma 3.1 quantifies $f = I(S; c)$ into sums of the mutual information between feature subsets and the class vector, it is still difficult to estimate the maximal gain of such individual mutual information quantity, i.e. $\underset{X \subset S}{\Delta_u}(X) = f(X \cup \{u) - f(X)$.

To mitigate the weakness of Max Dependency feature selection criterion, mRMR avoids estimating multivariate densities of both $p(x_1, x_2, ..., x_m)$ and $p(x_1, x_2, ..., x_m, c)$

of the global mutual information on selected feature set S_m. Alternatively, it could be much more straightforward and accurate by calculating each bivariate density $p(x_i, x_j)$ and $p(x_i, c)$ (detailed explanation can be found in [28]). By doing this, the problem of maximizing $I(S_m, c)$ becomes maximizing the relevance function $\sum\limits_{x_i \in S_m} I(x_i; c)$. We can easily prove that this function is monotone and submodular when $I(x_i; c)$ is always non-negative. The mRMR objective function, however, does not hold these properties. mRMR tends to select high discriminative features (or reduces the relevance among features being selected) as penalizing its objective function by a redundancy term (Equation 3.7). This consequently neither makes the mRMR condition submodular nor monotone due to the repercussion of subtraction. Despite the original mRMR feature selection method is being used and attain very good performance in many applications by the simple greedy algorithm, there is still no guaranteed bound for this. Also, lacking the monotone and submodular property makes it challenging to employ accelerated greedy algorithms [65].

3.3.2 Normalized Mutual Information

The simple relevance measure $I(x_i | i = \overline{1, n}; c)$ is widely used for mutual information-based methods. However, in this research, this measure is replaced by two variations, normalized by:

$$NMI_{ave}(X;Y) = \frac{2I(X;Y)}{H(X) + H(Y)}, \tag{3.12}$$

and,

$$NMI_{sqrt}(X;Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}, \tag{3.13}$$

in which $I(X;Y)$ is the mutual information (MI) of two random variables X and Y; $H(X)$ and $H(Y)$ are entropy values of X and Y, respectively.

The reasons for choosing these normalized mutual information functions to quantify the relevance term are as the following: Firstly, there is no upper bound for $I(X;Y)$ that makes it seem facile to interpret and is not straightforward for comparison purposes. On the observation that $I(X;Y) \leq \min\{H(X), H(Y)\}$ [1], normalized mutual information can be ranged in $[0, 1]$. Secondly, a measure based on informational entropy is favourable in evaluating correlation between two nominal attributes

[1] $I(X;Y) \leq \min\{H(X), H(Y)\} \leq \sqrt{H(X), H(Y)} \leq \frac{1}{2}(H(X) + H(Y))$. For more details, see [74].

with higher confidence [71]. Thirdly, when the number of features becomes large, there is no penalization to $I(X;Y)$, but entropy tends to increase. Features with many values tend to have higher entropy than features with less values. This in turn might lead into higher mutual information and a bias into features of many values. An alternative solution to cope with this problem is normalizing mutual information by dividing it by the entropy of a feature. Mutual information (or information gain) prefers features with more values while symmetrical uncertainty (i.e. $NMI(\cdot)$) compensates for the MI's bias toward attributes [72].

Proposition 3.2 *Given a feature set $S \subseteq \Omega$ and class c, let $f(S) = NMI(S;c)$, then f is not submodular.*

Proposition 3.2 implied that the solution for maximizing the normalized mutual information is also unachievable. Therefore this quantity by itself cannot solve the problem stuck as Max Dependency. For this reason, an alternative feature selection strategy will be proposed to combine the advantage of $NMI(\cdot)$ function and a substitute redundancy term, called *dispersion*, to enhance feature selection performance and draw a theoretical approximation for the solution by greedy searching.

3.3.3 Alternative Feature Selection Strategy For Binary Data

Several mutual information-based feature selection methods [25][26][27] attempt to select a feature subset $S_m \subseteq \Omega$ such that

$$\begin{array}{ll} \max\limits_{S \subseteq \Omega} & (Relevance(S;c) - Redundancy(S)) \\ \text{s.t.} & |S| = m \end{array}, \tag{3.14}$$

where $Relevance(\cdot)$ measures joint mutual information among variables.

As described in Section 3.2, this problem is also NP-hard. The incremental search can heuristically obtain a desired subset. However, to the best of the author's knowledge, there is no theoretical approximate solution. Fortunately, binary variables own properties which are able to facilitate the submodularity that the aforementioned objective function is lacking. This section will take advantage of submodular properties on binary data to propose an alternative objective function which can theoretically achieve a near-optimal subset for the feature selection problem. Further more, binary representation where features are normalized as value 0 or 1 can accelerate calculating the mutual information among them, therefore computational efficiency is a benefit in high dimensional data.

Section 3.3.2 showed the justifications for the new mutual information quantity in terms of facilitating confidence of a selected feature set. However, Proposition 3.1 and Proposition 3.2 indicate that neither $MI(\cdot)$ nor $NMI(\cdot)$ holds the submodularity property, which might lead to the computational cost in a high dimensional feature space. An alternative objective function is a need in the context where the original mRMR method does not theoretically utilize the greedy search. This section will define another redundancy function as a combination of metric distance functions. Distance measure has been actively studied for clustering problems. They have numerous variants as information theoretic quantities to measure the similarity (or diversity) between clusters [73][74][75].

Definition 3.3 *Let $d : X \rightarrow \mathbb{R}^+$ be a metric on set X. For all $x, y, z \in X$, d is a distance function if satisfying the following conditions:*

1. $d(x, y) \geq 0$.
2. $d(x, y) = 0 \Leftrightarrow x = y$.
3. $d(x, y) = d(y, x)$.
4. $d(x, y) \leq d(x, z) + d(z, y)$.

Definition 3.4 *Let $d(\cdot)$ be a metric distance function on X, then dispersion function $dp(U)$ of elements in $U \subseteq X$ is defined as*

$$dp(U) = \sum_{\{u,v\}|u,v \in U} d(u, v). \tag{3.15}$$

Generally, dispersion implies the difference between all pairs of elements and has been being used in the context of databases, social media and web search. Nevertheless, the problem of optimizing the dispersion function is known as NP-hard. Equation 3.17 defines a new objective function as the integration of the normalized mutual information acting as a relevance term and the distance function (Definition 3.5) that plays a role of redundancy.

To diminish the solemn inclination of subtraction in the objective function which naturally affects the monotone submodular property, the redundancy criterion is replaced by a distance function $d(x_i, x_j | x_i, x_j \in S)$ between all pairs, which is $1 - I(x_i; x_j)$.

Definition 3.5 *$nd(x, y) = 1 - I(x, y)$ is the normalized distance function of $d(x, y)$ defined in Definition 3.3, and*

$$nd(U) := \sum_{\{u,v\}|u,v \in U} nd(u, v).$$

Lemma 3.3 *$nd(x_i, x_j)$ is a non-negative function for binary variables.*

Proof. *Obviously, $I(x_i, x_j) \leq \min\{H(x_i), H(x_j)\}$ that is not greater than 1 as the entropy of a binary variable is max out at 1.*

Definition 3.6 *$f_{rel}(S)$ measures the total relevance between features in subset S and class c by the normalized mutual information function*

$$f_{rel}(S) = \sum_{x_i \in S \subseteq \Omega} NMI(x_i; c). \tag{3.16}$$

An alternative greedy searching strategy will tend to optimize the following objective:

$$\begin{aligned} \max_{S \subseteq \Omega} \quad & f_{NMI\_DIS}(S) = \tfrac{1}{|S|} f_{rel}(S) + \tfrac{1}{|S^2|} nd(S) \\ \text{s.t.} \quad & |S| = k, \end{aligned} \tag{3.17}$$

where $|S|$ is the cardinality of subset S.

Algorithm 2 approximates but substitutes the relevance function akin to the original mRMR by a new f_{rel} relevance measure as in Equation 3.16. In case of using $NMI_{sqrt}(\cdot)$ function, it is scaled by a ratio of $\frac{1}{2}$. This should be appropriate for the greedy search solution that will be demonstrated later.

Lemma 3.4 *$f_{NMI\_DIS}(S)$ is a non-negative and monotone submodular function.*

Proof. Clearly if there exists two selected feature sets $f_{NMI\_DIS}(S)$ and $f_{NMI\_DIS}(T)$ so that $S \subseteq T \subseteq \Omega$ where Ω is the whole feature set, and a feature $u \in \Omega \backslash T$, then $f_{NMI\_DIS}(S \cup \{u\}) - f_{NMI\_DIS}(S)$ is always equal to $f_{NMI\_DIS}(T \cup \{u\}) - f_{NMI\_DIS}(T)$ when $nd(S)$ is consequently non-negative as Lemma 3.3. Therefore the monotone submodular and non-negative property are held when $f_{NMI\_DIS}(T) \geq f_{NMI\_DIS}(S) \geq 0$.

Lemma 3.5 *$nd(x_i, x_j)$ is a metric for binary variables.*

Proof. Because conditioning always decreases entropy [74], the conditional entropy of x_i given x_j, $H(x_i|x_j)$ holds the triangle inequality property since

$$\begin{aligned} H(x_i|x_j) &\leq H(x_i, x_k|x_j) = H(x_i|x_k, x_j) + H(x_k|x_j) \\ &\leq H(x_i|x_k) + H(x_k|x_j). \end{aligned} \tag{3.18}$$

for any x_k in the feature set.

To prove our distance function is a metric, we need to show that $nd(x_i, x_j) \leq nd(x_i, x_k) + nd(x_k, x_j)$ for any x_k in the given feature set. It becomes the following inequality

$$1 + I(x_i; x_j) - I(x_i; x_k) - I(x_k; x_j) \geq 0. \tag{3.19}$$

Using the relation between mutual information and entropy, the LHS of Equation 3.19 becomes

$$\begin{aligned} &1 - H(x_i|x_j) + H(x_i|x_k) + H(x_k|x_j) - H(x_k) \\ &= 1 - (H(x_i|x_j) - (H(x_i|x_k) + H(x_k|x_j))) - H(x_k). \end{aligned} \tag{3.20}$$

By the use of Equation 3.18 and $H(x_k) \leq 1$, the proof is hence completed.

Algorithm 2 Alternative Greedy Search

```
**INPUT:** A set of features $\Omega$, a class vector $c$, $k$ features need to be selected.
**OUTPUT:** A set of $k$ selected features.
$S \leftarrow \emptyset$
**for** $i = 1$ to $k$ **do**
    **if** $i = 1$ **then**
        $S \leftarrow \arg\max_{u \in \Omega} \; f_{NMI\_DIS}\left(\{u\}\right)$    In case of applying $NMI_{ave}(\cdot)$    $(*)$
        $S \leftarrow \arg\max_{u \in \Omega} \; f_{NMI\_DIS}\left(\{u\}\right)$    In case of applying $NMI_{sqrt}(\cdot)$    $(**)$
    **else**
        $u^* \leftarrow \arg\max_{u \in \Omega \backslash S} \; \left(f_{NMI\_DIS}(S \cup \{u\}) - f_{NMI\_DIS}(S)\right) + \frac{1}{|S|} nd(S)$    $(*)$
        $u^* \leftarrow \arg\max_{u \in \Omega \backslash S} \; \frac{1}{2}\left(f_{NMI\_DIS}(S \cup \{u\}) - f_{NMI\_DIS}(S)\right) + \frac{1}{|S|} nd(S)$    $(**)$
        $S \leftarrow S \cup \{u^*\}$
    **end if**
**end for**
**Return** $S$
```

Theorem 3.2 *Greedy algorithm 2 achieves a ratio of $\frac{1}{2}$ for the mRMR problem.*

Proof. Lemma 3.4 has already proven that f_{NMI_DIS} is non-negative and monotone submodular combined with a distance metric between pairs of features shown in Lemma 3.5, Borodin et al. [75] show that the linear time greedy algorithm achieves a 2-approximation for any kinds of the configuration in Algorithm 2 satisfying a cardinality constraint.

Proposition 3.3 *Greedily picking an element such that maximizing relevance measure $NMI_{sqrt}(\cdot)$ is equivalent to minimizing the redundancy among the selected feature set.*

To prove Proposition 3.3, the following lemmas are needed.

Lemma 3.6 *Let $H(X_A)$ is the Shannon entropy function over the feature set X_A, then $H(X_A)$ is a monotone submodular function, i.e., $H(X_S) \leq H(X_T)$ for any $\emptyset \not\subset S \subseteq T$.*

For the proof of Lemma 3.6, see [76].

Lemma 3.7 *If $H(X_A)$ is the entropy function defined in Lemma 3.6, then*

$$H(X_S) \approx \sum_{x_i \in S} H(x_i) - \sum_{x_i, x_j \in S; i \neq j} I(x_i; x_j). \tag{3.21}$$

Proof. Using the joint entropy chain rule and mutual information chain rule, we have:

$$\begin{aligned} H(X_S) &= H(x_1) + \sum_{i=2}^{|S|} H(x_i|x_1, ..., x_{i-1}) \\ &= H(x_1) + \sum_{i=2}^{|S|} (H(x_i) - I(x_1, ..., x_{i-1}; x_i)) \\ &= H(x_1) + \sum_{i=2}^{|S|} \left(H(x_i) - \sum_{j=1}^{i} I(x_j; x_i|x_1, ..., x_{j-1}) \right) \\ &= \sum_{i=1}^{|S|} H(x_i) - \sum_{i=2}^{|S|} \sum_{j=1}^{i} I(x_j; x_i|x_1, ..., x_{j-1}) \end{aligned} . \tag{3.22}$$

If we assume there is no high-order but the second-order interaction between features (i.e., only consider mutual information between pairs of features), Lemma 3.7 is proven.

Proposition 3.3 is ready to prove.

Proof. Let $f(A) = \frac{I(X_A;c)}{\sqrt{H(X_A)H(c)}}$ for any $A \subseteq \Omega$; a real vector c. For any $u \in \Omega \backslash A$. We have:

$$\begin{aligned} f(A \cup \{u\}) - f(A) &= \frac{1}{\sqrt{H(c)}} \left(\frac{I(X_{A\cup u};c)}{\sqrt{H(X_{A\cup u})}} - \frac{I(X_A;c)}{\sqrt{H(X_A)}} \right) \\ &= \frac{1}{\sqrt{H(c)}} \left(\frac{H(X_{A\cup u}) - H(X_{A\cup u}|c)}{\sqrt{H(X_{A\cup u})}} - \frac{H(X_A) - H(X_A|c)}{\sqrt{H(X_A)}} \right) \quad . \\ &= \frac{1}{\sqrt{H(c)}} \left(\left(\sqrt{H(X_{A\cup u})} - \sqrt{H(X_A)} \right) - \left(\frac{H(X_{A\cup u}|c)}{\sqrt{H(X_{A\cup u})}} - \frac{H(X_A|c)}{\sqrt{H(X_A)}} \right) \right) \end{aligned} \tag{3.23}$$

Let $h(A) = \frac{H(X_{A\cup u}|c)}{\sqrt{H(X_{A\cup u})}} - \frac{H(X_A|c)}{\sqrt{H(X_A)}}$, we have:

$$\begin{aligned} h(A) &= \frac{H(X_{A\cup u}|c)}{\sqrt{H(X_{A\cup u})}} - \frac{H(X_A|c)}{\sqrt{H(X_A)}} = \frac{H(c,X_{A\cup u}) - H(c)}{\sqrt{H(X_{A\cup u})}} - \frac{H(c,X_A) - H(c)}{\sqrt{H(X_A)}} \\ &\geq \frac{\max\{H(c),H(X_{A\cup u})\} - H(c)}{\sqrt{H(X_{A\cup u})}} - \frac{H(X_A)}{\sqrt{H(X_A)}} \end{aligned} \quad . \tag{3.24}$$

Case 1: If $H(c) \geq H(X_{A\cup u})$, then $h(A) \geq -\sqrt{H(X_A)}$, so

$$f(A \cup \{u\}) - f(A) \leq \frac{\sqrt{H(X_{A\cup u})}}{\sqrt{H(c)}}. \tag{3.25}$$

Case 2: If $H(c) < H(X_{A\cup u})$, then $h(A) \geq \sqrt{H(X_{A\cup u})} - \sqrt{H(X_A)} - H(c)$, so

$$f(A \cup \{u\}) - f(A) \leq \frac{H(c)}{\sqrt{H(c)}\sqrt{H(X_{A\cup u})}} < \frac{H(X_{A\cup u})}{\sqrt{H(c)}\sqrt{H(X_{A\cup u})}} = \frac{\sqrt{H(X_{A\cup u})}}{\sqrt{H(c)}}. \tag{3.26}$$

When sequentially selecting an element for the desired m-element subset, assumed $S_m \subseteq \Omega$. We have $f(S_m) - f(\emptyset) \leq \frac{1}{\sqrt{H(c)}} \sum\limits_{i=1}^{m} \sqrt{H(S_i)}$ indicating total joint information gain by the greedy search.

By utilizing Jensen's inequality and Lemma 3.6, we obtain:

$$\left(\sum_{i=1}^{m} \sqrt{H(S_i)} \right)^2 \leq \frac{\sum\limits_{i=1}^{m} H(S_i)}{m} \leq H(S_m). \tag{3.27}$$

When $H(c)$ is constant, combined with Lemma 3.7, the proof is hence completed.

3.4 Summary

This section provides an overview of information theory that is commonly used in feature selection tasks. Several studies working on mutual information-based methods in the literature failed to find a theoretical approximation solution to the problem of optimizing the relevance-redundancy criterion. The review of submodular set functions and their applications in practice is hence introduced to propose an alternative feature selection strategy. By theoretical analysis, submodular properties on binary variables can facilitate the greedy algorithm to find an approximation solution. As a result, the proposed method can achieve a 2-approximation by the heuristic search. Furthermore, this section also shows that greedily picking features into the feature selection subset can effectively diminish the redundancy among these features themselves.

Chapter 4

Experiments and Evaluations

This section will analyze the performance of the proposed methods to evaluate the effectiveness of new criteria. First, descriptions of datasets will be given, then several machine learning algorithms such as the statistical classifier (Naive Bayes), the algorithm using hyperplane for classification (Multiclass Support Vector Machines), and the decision-based algorithm (Decision Trees). Experimental results will be elucidated by further analysis and discussion. More details, the performance of the proposed methods, namely aNMI-DIST which uses the $NMI_{ave}(\cdot)$ function, and sNMI-DIST which applies the $NMI_{sqrt}(\cdot)$ function in the searching strategy of Algorithm 2, is compared with that of common standard filter feature selection methods including DF, CHI, IG, and ReliefF; mutual information-based methods such as mRMR [27], MIFS [25], and MIFS-U [26]; feature selection methods dedicated to binary data such as Max-criterion and Diff-criterion in [37]; and some well-known wrapper feature selection methods such as Sequential Forward Selection (SFS) and Sequential Forward Floating Selection (SFFS) [7]. Performance of the proposed methods on non-binary data is also investigated to verify the effectiveness of algorithms used. Finally, insights about the redundancy reduction of the proposed method will be discussed by the similarity of selected features.

4.1 Datasets

The experiments are conducted on several datasets under three different domains: Document classification datasets, Cyber Threat on Social network datasets, and Medical diagnosis datasets. For the first two groups of datasets, the results presented below are obtained by using 10-fold cross validation to evaluate the proposed models. For the Medical diagnosis group, training and testing will be performed and examined on two sets using the training and testing datasets used by the competition where these dataset were provided originally. Table 4.1 gives an overview of the datasets used.

Table 4.1: An overview of the datasets used

Dataset	Feature Representation	# Features	# Samples	# Training Samples	# Testing Samples	# Labels	# Zero Features	Density
Cora	Binary	1433	2708	2438	270	7	1	1.27%
CiteSeer	Binary	3703	3312	2981	331	6	0	0.86%
WebKB	Binary	1703	877	790	87	5	0	5.20%
SMS spam	Binary	1520	5574	5197	557	2	0	0.44%
	TF-IDF	1520	5574	5197	557	2	0	0.44%
Spambase	Frequency	57	4601	4141	460	2	0	22.59%
Terrorist Attacks	Binary	106	1293	1164	129	6	8	9.76%
Terrorists	Binary	1224	851	766	85	2	288	1.68%
SPECT	Binary	22	267	80	187	2	0	21.99%
HIVA	Binary	1617	42294	3845	38449	2	0	9.09%

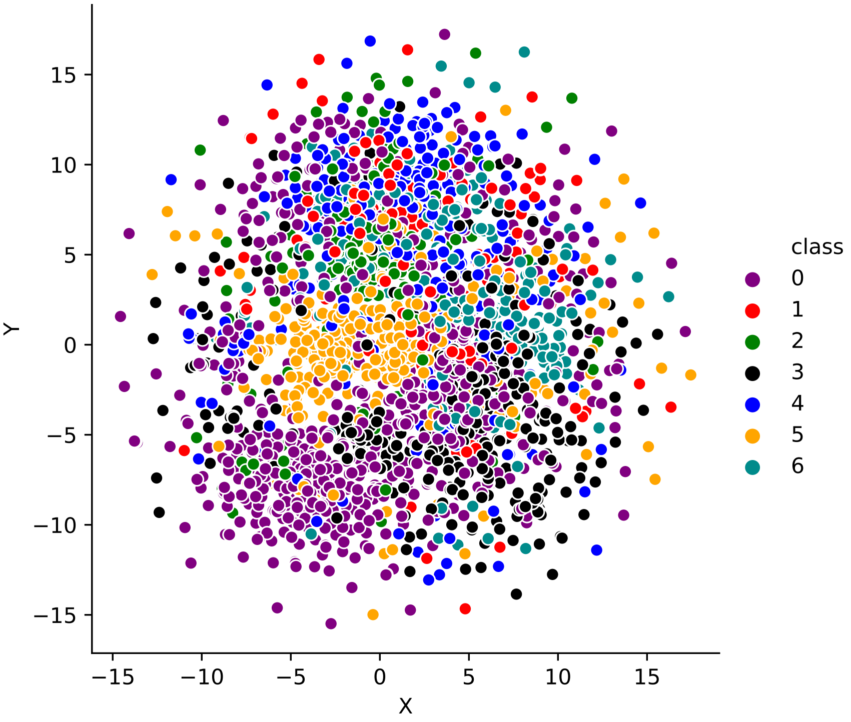


Figure 4.1: t-SNE mapping of Cora dataset

Note that a zero-feature indicates a missing feature where there is no sample in the dataset having this feature. Moreover, density of a dataset is calculated by the average percentage of non-zero feature in a sample over the total number of features. To better visualize how features and samples are distributed over every given datasets, t-SNE maps for those datasets are illustrated in Figure 4.1-4.8.

4.1.1 Document Classification Datasets

The experiments are conducted on publicly popular text classification datasets namely Cora, Citeseer, WebKB [77], and SMS Spam Collection [78]. In first three datasets,

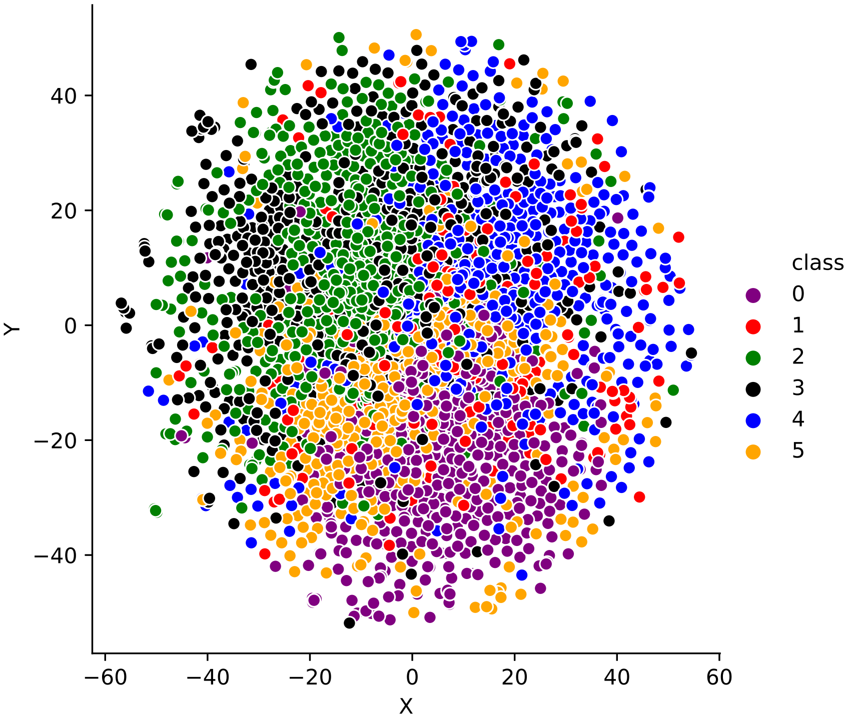


Figure 4.2: t-SNE mapping of CiteSeer dataset

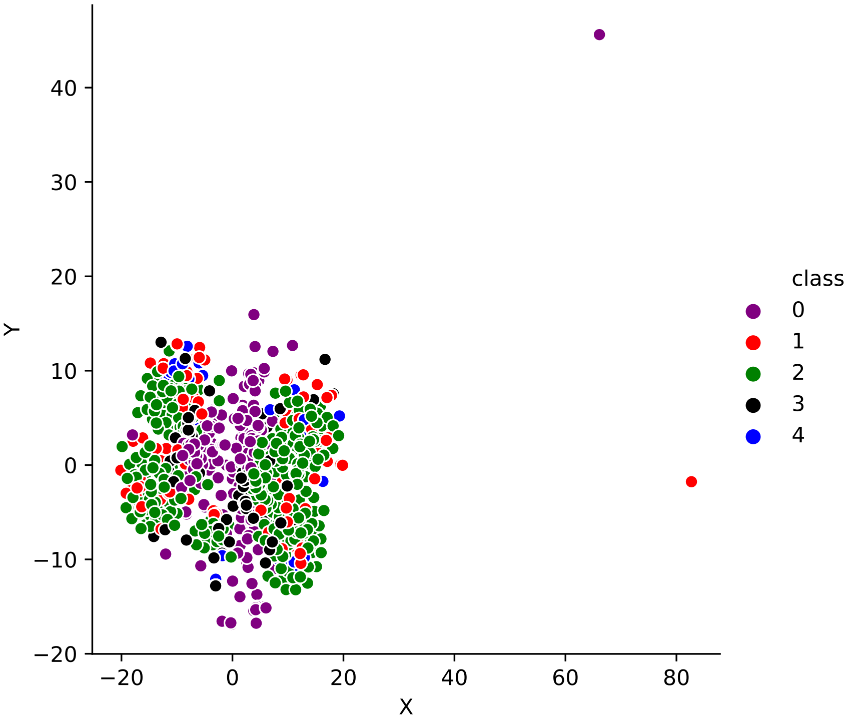


Figure 4.3: t-SNE mapping of WebKB dataset

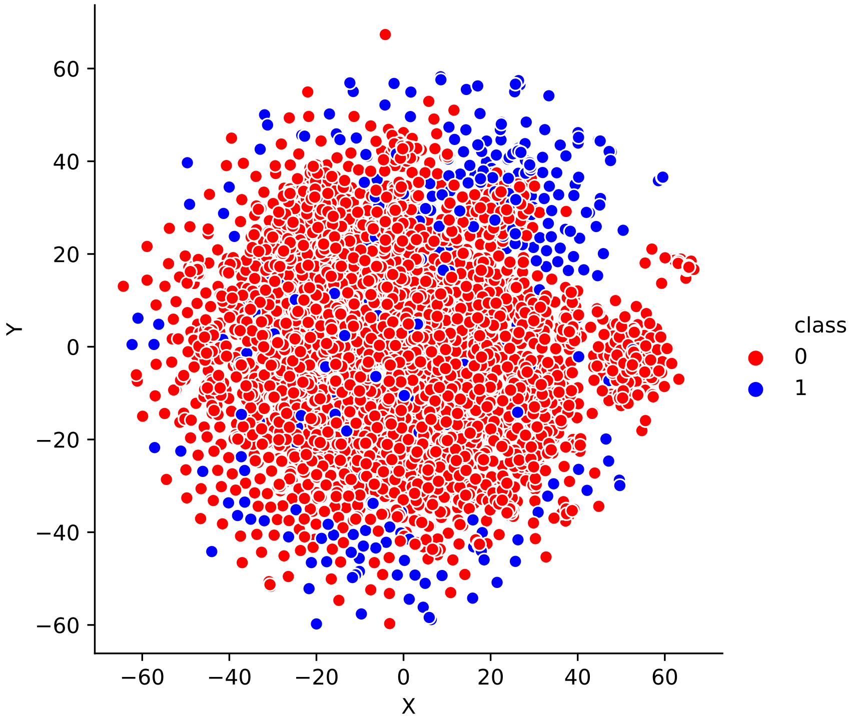


Figure 4.4: t-SNE mapping of SMS spam (binary) dataset

stop words and all words with document frequency less than 10 are eliminated. Remaining words as features are represented by binary values indicating the absence/presence of the corresponding word from the dictionary. More specifically:

Cora dataset contains 2708 scientific publications categorized into one of seven classes. The vocabulary consists of 1433 unique word in total with an average of around 18 words in a document;

Citeseer dataset is a selection of 3312 publications from six sub-domains in the computer science field. After stemming and removing stopwords, the dataset contains 3703 unique words left with around 31 words per document;

WebKB dataset collects 877 webpages from four different universities classified into one of five classes. Each document is described by a binary vector of 1703 dimensions and has around 90 words on average.

SMS Spam Collection is a set of SMS tagged messages that have been collected for mobile phone spam research. It contains one set of SMS messages in English of 5574 messages, tagged according being ham (legitimate) or spam. Different from email messages, SMS messages are usually in a short and non-standard form which contain slangs, abbreviations or even ”deliberate” typos aiming at fooling offensive text filtering systems. In the preprocessing step, all stop words, punctuation, stemming words and words with document frequency less than 5 are eliminated. Finally a set of features of 1530 unique words with an average of around 7 words per message is obtained. After the pre-processing step, features in this dataset are transformed into numerical values by both one-hot encoding vectors (SMS Spam binary) and *tf-idf* vectors (SMS spam tf-idf) to concretely demonstrate the efficiency of the proposed method on binary text representation.

4.1.2 Cyber Threat on Social Network Datasets

The rapid development of social networks and flexibility to propagate as well as spam cyber threat information on them are leading to significant demands on threat detection and prevention. An effective feature selection method might correct important attributes which are specialized for risky messages. This thesis uses two different datasets on cyber threats, including Terrorists and Terrorist Attack dataset [80] and Spambase dataset [79].

Terrorists dataset collects information and relationships about terrorists. The dataset was initially designed for classification tasks to categorize the relationships among terrorists. It contains 851 relationships distributed to 2 classes, each described by a binary valued vector of attributes where each entry indicates the absence/presence of a feature. There are a total of 1224 distinct features.

Terrorist Attack dataset consists of 1293 terrorist attacks each assigned one of 6 labels indicating the type of the attack. Each attack is described by a 0/1-valued vector of attributes whose entries indicate the absence/presence of a feature. There are a total of 106 distinct features.

Spambase dataset is a 4601-email collection of spam emails came from UCI’s postmaster and individuals who had filed spam, and non-spam emails came from filed work and personal emails. This dataset contains 57 input attributes of continuous format indicating whether a particular word or character was frequently occurring in the email, and 1 target attribute in the discrete format.

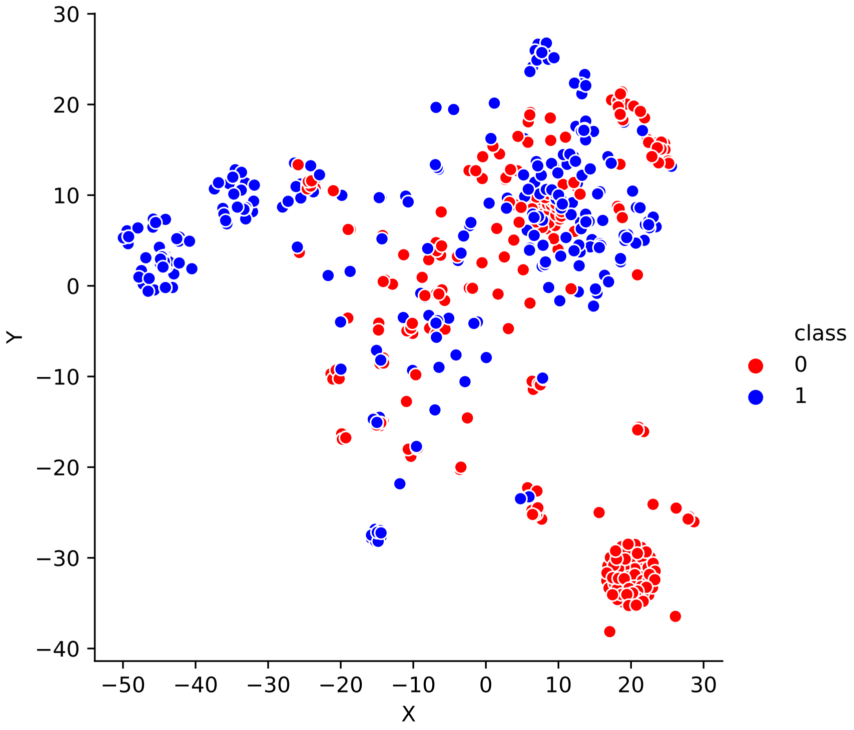


Figure 4.5: t-SNE mapping of Terrorists dataset

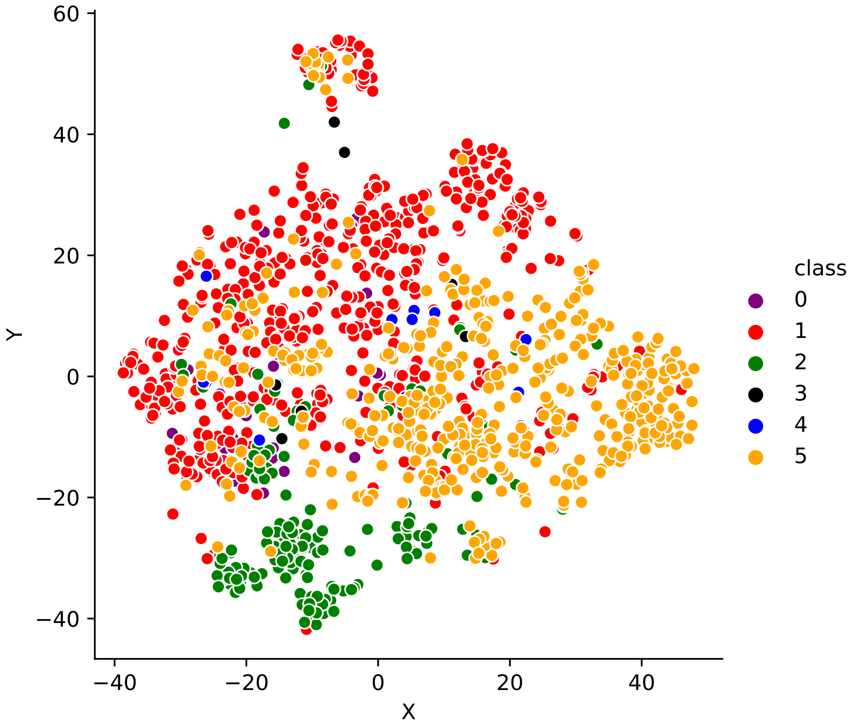


Figure 4.6: t-SNE mapping of Terrorist Attacks dataset

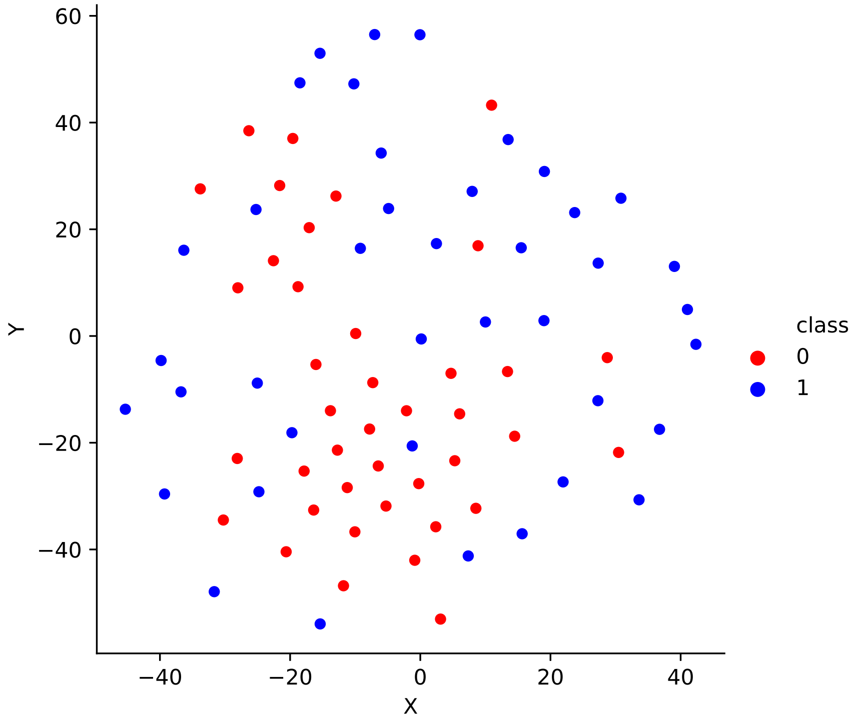


Figure 4.7: t-SNE mapping of SPECT dataset

4.1.3 Medical diagnosis datasets

Classification on Disease diagnosis datasets may act as clinical decision supports for physicians, enabling earlier prediction and identification of disease, thereby tailoring treatment plans to the needs of patients. Two medical diagnosis datasets will be used to verify the proposed method.

SPECT Heart dataset [81] contains 267 image sets of patients who are diagnosed of cardiac Single Proton Emission Computed Tomography (SPECT). The dataset was extracted into 22 binary feature patterns where each of the patients is classified into two categories: normal and abnormal. Hence there are totally 23 binary attributes being used including the binary target. This is an imbalanced dataset with only 55 samples in normal class compared to 212 samples in the another.

HIVA dataset [82] is to predict which compounds are active against the HIV/AIDS infection. Originally, the dataset is classified into 3 categories: active, moderately active, and inactive. Then authors quantized it into the binary classification problem (active and inactive). The dataset is represented by 1617 sparse binary input variables which appear for properties of the molecule inferred from its molecular structure.

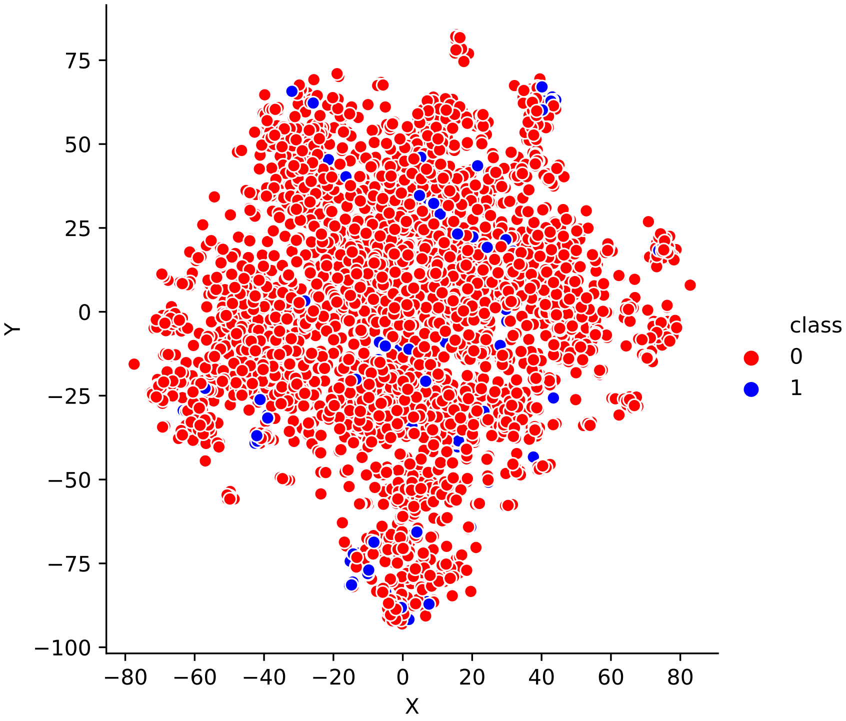


Figure 4.8: t-SNE mapping of HIVA dataset

This is an imbalanced dataset with total 1489 positive instances and 40805 negative instances.

4.2 Performance Metrics

This thesis uses Accuracy and F-measure, which are widely being used for classification tasks to evaluate the prediction performance. To assert the effectiveness of the proposed feature selection method, Jaccard similarity will also be investigated with the aim at analyzing how features selected affect to the redundancy reduction, as well as how redundancy reduction is important to the classification performance.

Accuracy

Testing accuracy measures all the correct prediction over the whole testing dataset. It can be expressed as

$$Acc = \frac{Number\,of\,testing\,samples\,that\,are\,correctly\,classified}{Total\,number\,of\,testing\,samples}. \tag{4.1}$$

The accuracy metric is generally useful in cases where data is balanced. In fact, there are many datasets which are extracted from imbalanced number of samples in the class vector. In spam email dataset in general, for example, the number of spam messages is notably smaller than that of non-spam messages. This might lead to misevaluation when most of correct predictions contributing to Accuracy were conducted on the non-spam samples, so the ability to detect spam emails is not as correct as a classification model makes it out to be.

F-measure

To alleviate the aforementioned problem, F-measure is a common term to replace Accuracy in evaluating the prediction performance. It is widely used in the domain of information retrieval such as measuring search, or document classification. Formally, traditional F-measure (or F-score) is calculated as

$$F - measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{4.2}$$

where:

- *Precision*: The ratio of the number of correct predictions over the number of all instances predicted.
- *Recall*: The ratio of the number of correct predictions over the number of all that should be identified as true labels by the classifier.

Similarity

As the proposed method, redundancy among features plays a role in selecting a good subset. Intuitively, more similarity among features selected, more redundant information included. To measure how feature selection methods influence to redundancy, this work considers Jaccard index as a similarity coefficient which is popular used in binary data, and the similarity measure is obtained by summing up the similarity values of pairs of features.

Let A and B are two non-empty feature subsets. Jaccard coefficient measures similarity between two subsets by dividing the size of the intersection divided by the size of the union of the subsets, which is defined as below

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}. \tag{4.3}$$

4.3 Classifiers

There are many machine learning algorithms being used in classification tasks. Deep learning methods have the ability to automatically extract features from input data with little or no preprocessing by regularization functions [83]. While deep learning algorithms using neural networks have achieved incredible performance, shallow classifiers such as Support Vector Machines, Naive Bayes, Random Forests, K-means Clustering or even Logistic Regression are showing their power as well. Although neural networks generally work better than shallow methods, intensive training should be taken into consideration [84]. Moreover, some shallow methods are able to be on par with neural networks in practice, especially on small to medium-sized datasets. For example, an empirical study of Twitter and Tumblr for sentiment analysis in [85] shows that the best performance is achieved using SVM and substantially outperforms that of the Multi-layer Perceptron Neural Network. In [86], SVM is able to discriminate images better than deep learning techniques used both raw and reconstructed inputs without regularization. Moreover, feature selection is a very important task to reduce uninformative features using shallow classifiers. For these reasons, instead of competing for state-of-the-art results on classification tasks, this thesis only conducts experiments on shallow classifiers to investigate how feature selection methods perform on them. After examining on several well-known classifiers, SVM and NB and DT are employed because they generally perform better than others in terms of prediction accuracy. SVM and NB, in particular, are more being paid attention in classification tasks due to easy implementation and good performance [87].

4.3.1 Naive Bayes

Naive Bayes is a probabilistic learning method which has been actively studied for nearly a half century, and maintains as a competitive baseline classifier for text categorization. According to the concise definition in [19], by assuming that each of the features it uses is conditionally independent of one another given some class, the probability of a document D_i being in class c_j is computed as:

$$P(c_j|D_i) \propto P(c_j) \prod_{1 \leq k \leq n}, P(t_k|c_j) \tag{4.4}$$

where $P(t_k|c_j)$ is the conditional probability of term t_k occurring in a document of class c_j, $P(c_j)$ is the prior probability of a document occurring in class c_j. Naive Bayes classifier selects the best class having a maximum estimated posterior probability above

$$\begin{aligned}\hat{c}_j &= \arg\max_{c_j \in c} \hat{P}(c_j|D_i) \\ &= \arg\max_{c_j \in c} \hat{P}(c_j) \prod_{1 \le k \le n} \hat{P}(t_k|c_j)\end{aligned}, \tag{4.5}$$

where the conditional probability $\hat{P}(t_k|c_j)$ as the relative frequency of term t_k in documents belonging to class c_j with the vocabulary V:

$$P(t_k, c_j) = \frac{T_{c_j t_k}}{\sum_{t'_k \in V} T_{c_j t'_k}}, \tag{4.6}$$

with $T_{c_j t_k}$ is the number of occurrences of t_k in training documents from class c_j, including multiple occurrences of a term in a document.

This work assumes that the posterior probability follows a multinomial distribution because this lends itself to data which can easily be transformed to counts like word counts in text.

4.3.2 Support Vector Machines

The most common way to train multiclass SVMs classification is decomposing an input vector $x \in \mathbb{R}^d$ into k classes satisfied the following rule

$$\hat{y} = \arg\max_{m \in [k]} \mathbf{w}_m^T x, \tag{4.7}$$

in which the inner product $\mathbf{w}_m^T x$ represents the score of the m^{th} class corresponding to x. The classifier selects any classes having highest scores $\hat{y}$ by finding the solution to the following optimization problem during the training process [88].

$$\underset{\mathbf{w}_1 \to \mathbf{w}_k}{\text{minimize}} \quad \frac{1}{2} \sum_{m=1}^{k} \|\mathbf{w}_m\|^2 + C \sum_{i=1}^{n} \left(1 - \max_{m \neq y_i} \mathbf{w}_m^T x_i - \mathbf{w}_{y_i}^T x_i\right)_+, \tag{4.8}$$

where $C > 0$ is the regularization parameter and $(u)_+ = 0$ if $u < 0$ and $(u)_+ \neq 0$ otherwise.

Figure 4.9 visualizes a simple example of two groups which can be separated by some sample hyperplanes in two dimensional space. In theoretical, there are an

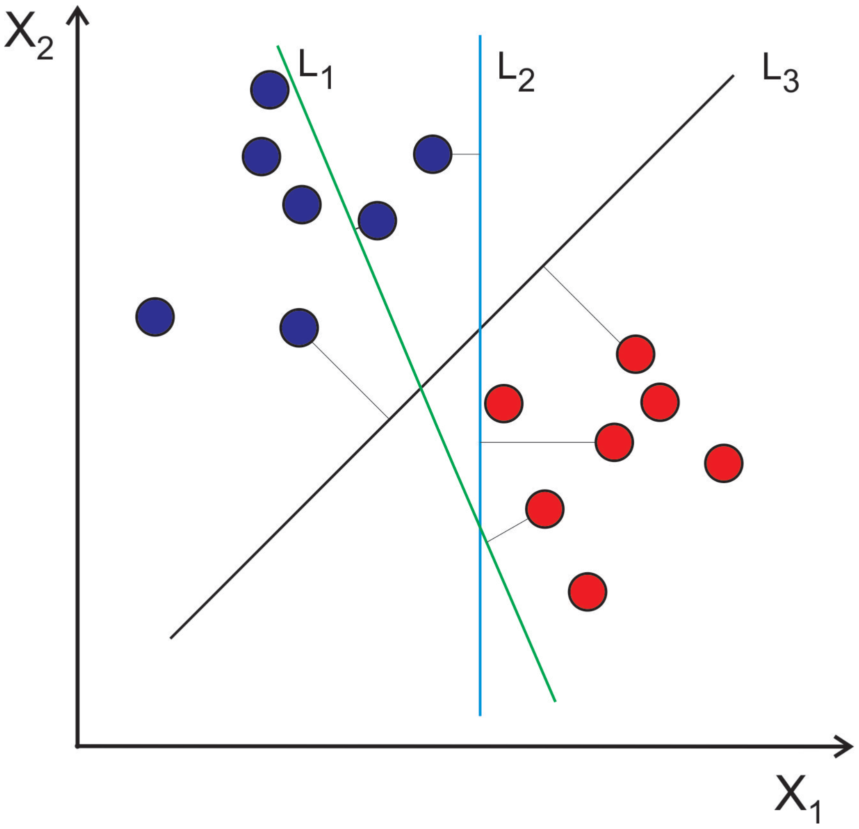


Figure 4.9: A simple line divides two groups of samples

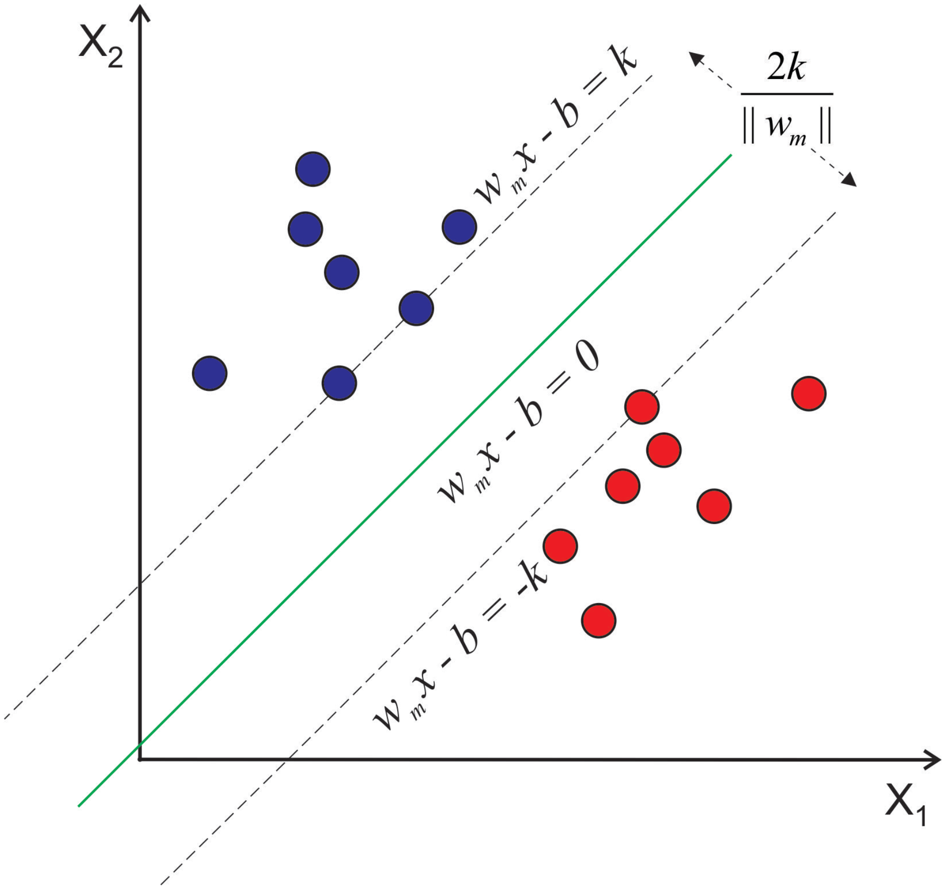


Figure 4.10: SVM hyperplane

infinite number of lines that can exactly cut the training data into two separate classes. However, SVM tends to choose one reasonable line as which represents the largest separation, or margin, between the two classes. More precisely, the hyperplane is chosen so that the distance from it to the nearest data point on each side is maximized. Such a hyperplane, if exists, is called *Maximum-margin hyperplane*. Figure 4.10 illustrates the Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

For the sake of simplicity, this thesis employs the popular LIBLINEAR library [89] to train the multiclass SVMs classifier. also, the linear kernel function that is commonly used in SVM classification tasks will be set as the default setup.

4.3.3 Decision Trees

Decision Trees are gained attention in the research circle many decades ago. However, the first time it was formally introduced and thoroughly investigated for Machine Learning applications is by J.R. Quinlan [20] in 1986. Basically, DT is a non-parametric supervised learning method used for classification and regression, with the purpose of building a model which can predict the value of a target variable by learning simple decision rules inferred from the data features. Tree-based decision algorithms have both advantages and limitations [90]. Several notable advantages of decision tree algorithms are listed as follow:

- Decisions can be built as tree elements, therefore tree-based algorithms are straightforward and interpreted for classification.
- Can reduce some pre-processing efforts such as data normalization or removing blank values.
- Logarithmic time complexity for the predicting process.
- Ability to work from both numerical and categorical data while some other methods only work on one certain type of data.
- The interpretation is intelligible and easy to understand and explain by boolean logic. A black box model like an artificial neural network makes itself obscure to interpret.
- Outcomes can be validated by statistical test models, therefore facilitating the reliability of the model.
- Working as a feature filter when irrelevant features can be eliminated through the training process, helping increase accurate decision.
- Learning process requires a small amount of resources compared to other algorithms, so it is useful when working with large datasets.

Major limitations of DT are:

- Easy to get overfitting when it creates over complex trees on some simple datasets.

- Sensitive to small change. A slight variation in the training data may cause a completely different tree.

- Finding an optimal decision tree is an NP-hard problem. A heuristic scheme to find every local optimal subtree can be applied. However, the global optimum for the decision tree could not guaranteed.

- Ability to work from both numerical and categorical data while some other methods only work on one certain type of data.

- Decision trees classifiers tend to bias dominant classes. Selecting some initial nodes for tree may result in poor performance.

This thesis implements the CART decision trees algorithm to generate binary trees using the feature and threshold that attain the largest information gain at each node. CART is very similar to C4.5 decision tree but it is able to work on both categorical and continuous variables, and constructs the tree based on a numerical splitting criterion recursively applied to the data whilst C4.5 takes more steps to calculate rule sets.

4.4 Experimental Evaluations

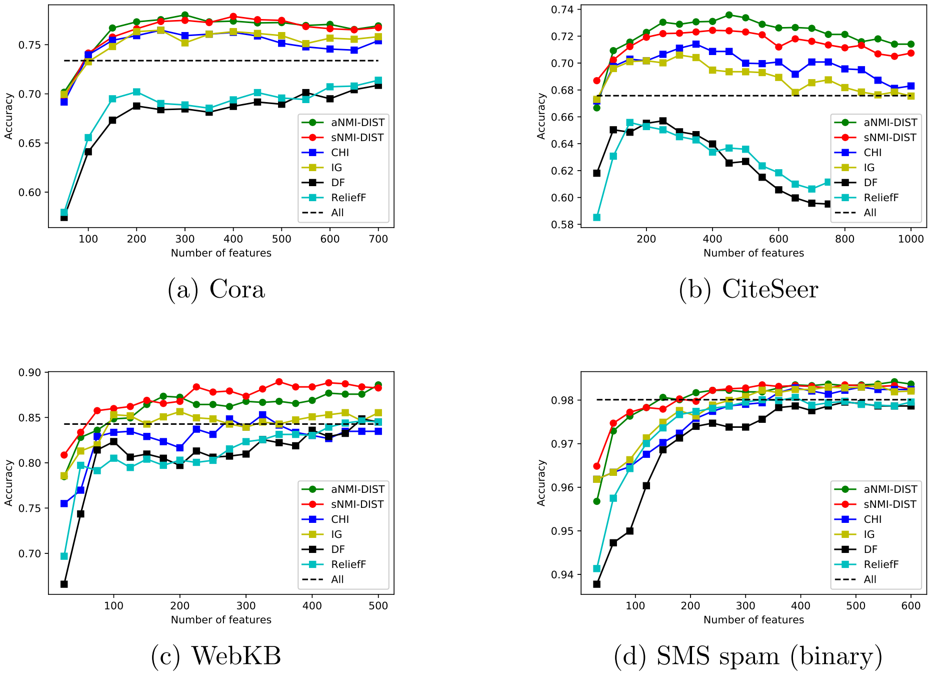
In this section, experimental results of the proposed methods, namely aNMI-DIST and sNMI-DIST, and benchmarking feature selection methods on various aforementioned datasets. Further analysis and discussion are also elaborated for each dataset based on its characteristics. Note that the "All" baseline indicates all features are used for classification.

4.4.1 Comparison with Standard Filter Methods

The performance of the proposed methods is compared with that of common standard filter feature selection methods. Figure 4.11 - 4.16 benchmark the classification accuracy of aNMI-DIST and sNMI-DIST against DF, IG, CHI and ReliefF using different classifiers. As the results, $NMI(\cdot)$-based algorithms mostly outperform standard filter methods in all cases, especially in document classification datasets. In detail, there is a negligible distinction between the classification accuracy of two proposed methods in Cora dataset and they attain the highest accuracy of nearly 79% using SVM and Naive Bayes, and 69% using Decision Trees classifier, while DF and ReliefF perform very poor classification predictions with the gaps of 8-13% less compared to NMI-based methods. This may be because DF retains most of informative features by discarding rare features based on only counting the number of terms in the corpus, while in the document context words having low frequencies may considerably contribute to the classification performance. Regarding ReliefF, it usually fails to identify interacting features in cases the feature space becomes large. This may be reasonable to explain the performance of ReliefF on several high dimensional feature such as CiteSeer, SMS spam, Terrorists/Terrorist Attacks, and HIVA dataset. It is not surprised when IG and CHI perform well on many datasets. Whereas IG is very similar to mutual information, CHI is known to be successful in retaining most relevant features. The results of aNMI-DIST and sNMI-DIST shown on the document classification datasets also emphasize the importance of the redundancy criterion when they can enhance the classification accuracy by eliminating irrelevant features, which basically are not applied in other standard filter feature selection methods. Noticeably, aNMI-DIST and sNMI-DIST are the unique methods that surpass the “All” baseline on Cora and CiteSeer dataset using Naive Bayes.

In Terrorist attacks dataset, there is a slight difference in the classification accuracy performed by NMI-based methods and standard filter methods when using SVM and NB classifier. As the aforementioned explanation, this dataset has only 106 features, which can be easy for simple ranking methods like DF or ReliefF. aNMI-DIST and sNMI-DIST only show predominance to DF and ReliefF by using Decision Trees. Although the proposed methods do not significantly perform better than IG and CHI in this case, they can reach the same performance with only 21 features rather than 30 features by CHI. A considerable distance between NMI-based methods and standard filter methods is more clearly presented in Terrorists dataset because of a large number of features. This pattern is obvious as the number of features selected is small (compared to the total number of features). When the number of features becomes large, IG seems to be on par with aNMI-DIST and sNMI-DIST.

NMI-based methods also work well on Medical diagnosis dataset using SVM where they perform 5% better on SPECT dataset and only 0.1% on HIVA dataset than the second best method, IG. They also outperform the other standard filter methods when using Decision Trees with nearly 78% classification accuracy compared to roughly 73% by IG and CHI. However, aNMI-DIST and sNMI-DIST are far inferior to the standard filter methods by using Naive Bayes. In HIVA dataset, IG can reach a peak of 91% by using Naive Bayes, creating a considerable gap of 3% and 4% to sNMI-DIST and aNMI-DIST respectively. In SPECT dataset, although DF and ReliefF surprisingly perform better than the others by using Naive Bayes, the classification accuracy is only 65%, far from nearly 82% by using SVM and 78% by using Decision Trees of NMI-based methods.



(a) Cora

(b) CiteSeer

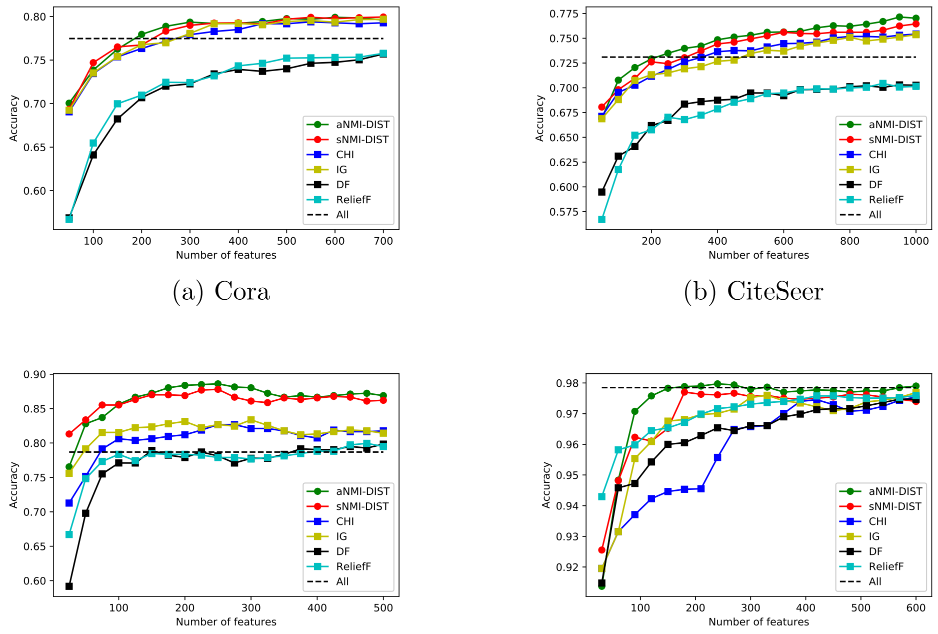
(c) WebKB

(d) SMS spam (binary)

Figure 4.11: Accuracy comparison with standard filter methods on document classification datasets using SVM

4.4.2 Comparison with Mutual Information-based Methods

This section benchmarks the proposed methods and mutual information-based methods, those are mRMR, MIFS, and MIFS-U. Note that in MIFS and MIFS-U, authors proposed the hyperparameter β to control the redundancy penalization. However, on the observation that the optimal value for this parameter is intractable in different



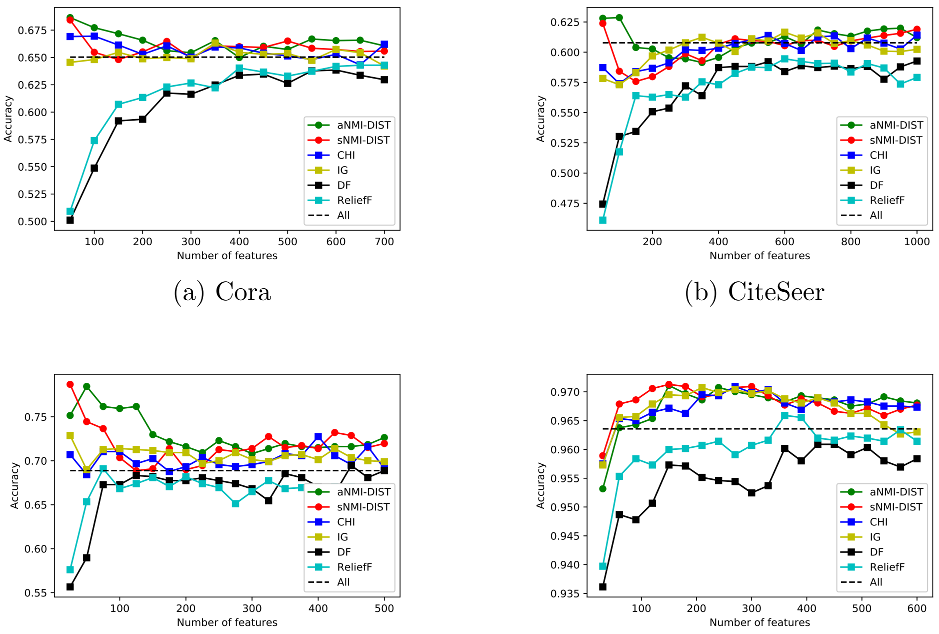
(a) Cora

(b) CiteSeer

(c) WebKB

(d) SMS spam (binary)

Figure 4.12: Accuracy comparison with standard filter methods on document classification datasets using Naive Bayes



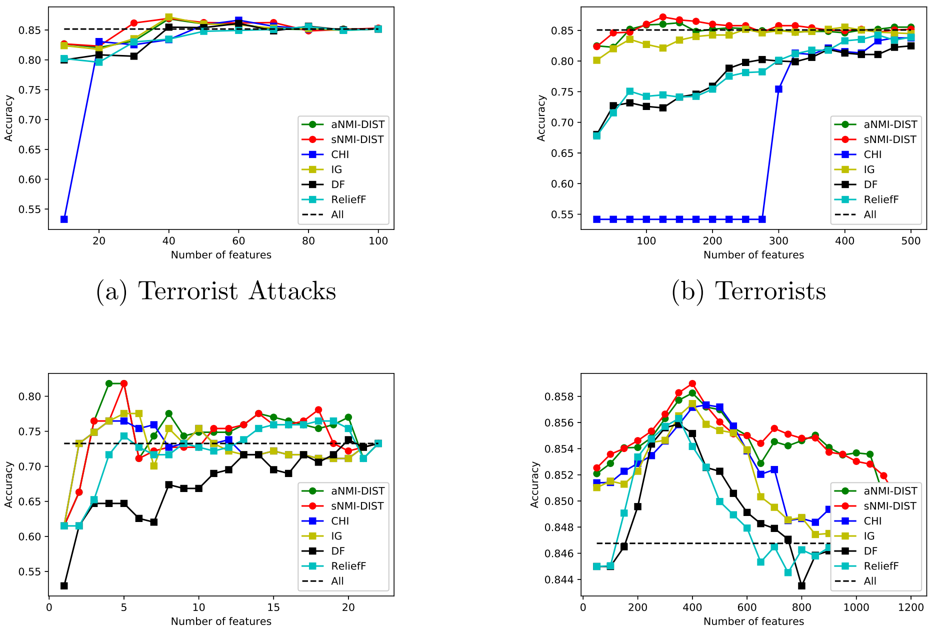
(a) Cora

(b) CiteSeer

(c) WebKB

(d) SMS spam (binary)

Figure 4.13: Accuracy comparison with standard filter methods on document classification datasets using Decision Trees



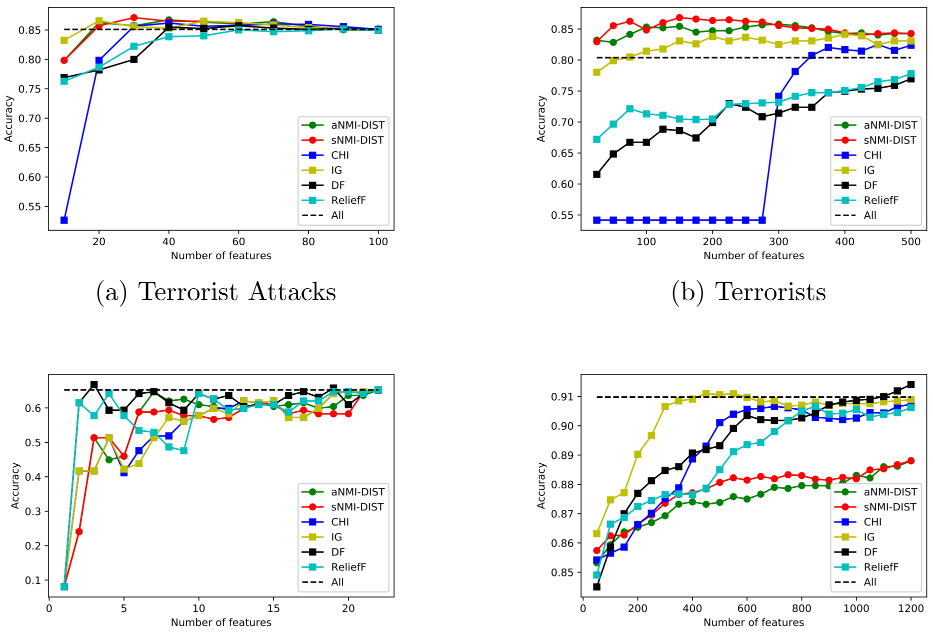
(a) Terrorist Attacks

(b) Terrorists

(c) SPECT

(d) HIVA

Figure 4.14: Accuracy comparison with standard filter methods on Cyber threat and Medical diagnosis datasets using SVM



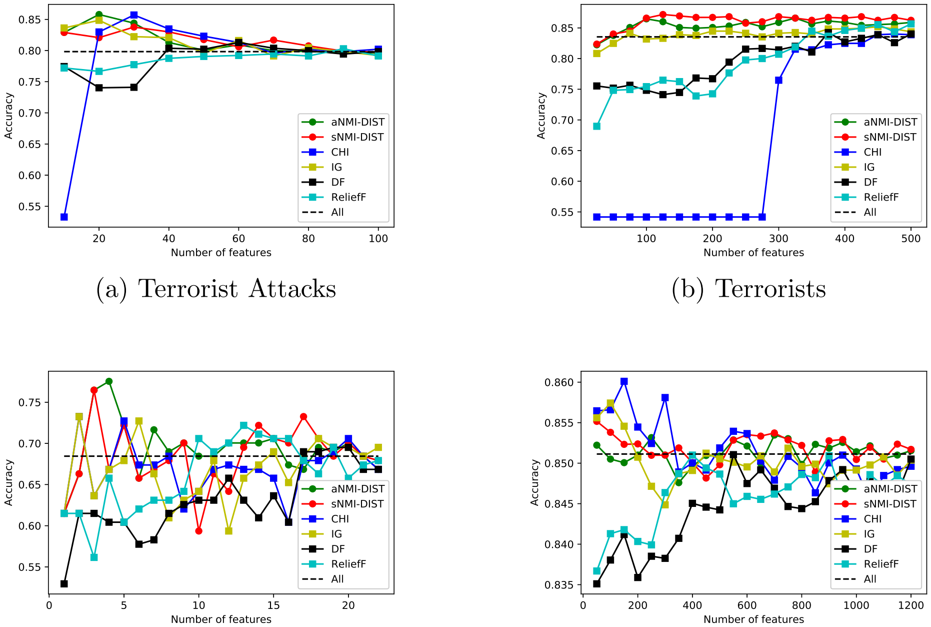
(a) Terrorist Attacks

(b) Terrorists

(c) SPECT

(d) HIVA

Figure 4.15: Accuracy comparison with standard filter methods on Cyber threat and Medical diagnosis datasets using Naive Bayes



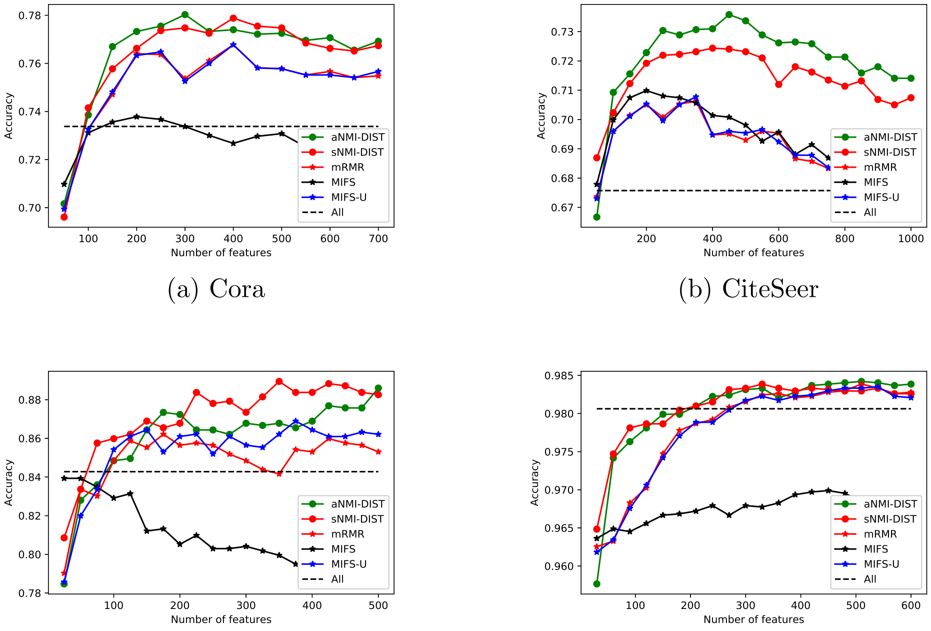
(a) Terrorist Attacks

(b) Terrorists

(c) SPECT

(d) HIVA

Figure 4.16: Accuracy comparison with standard filter methods on Cyber threat and Medical diagnosis datasets using Decision Trees



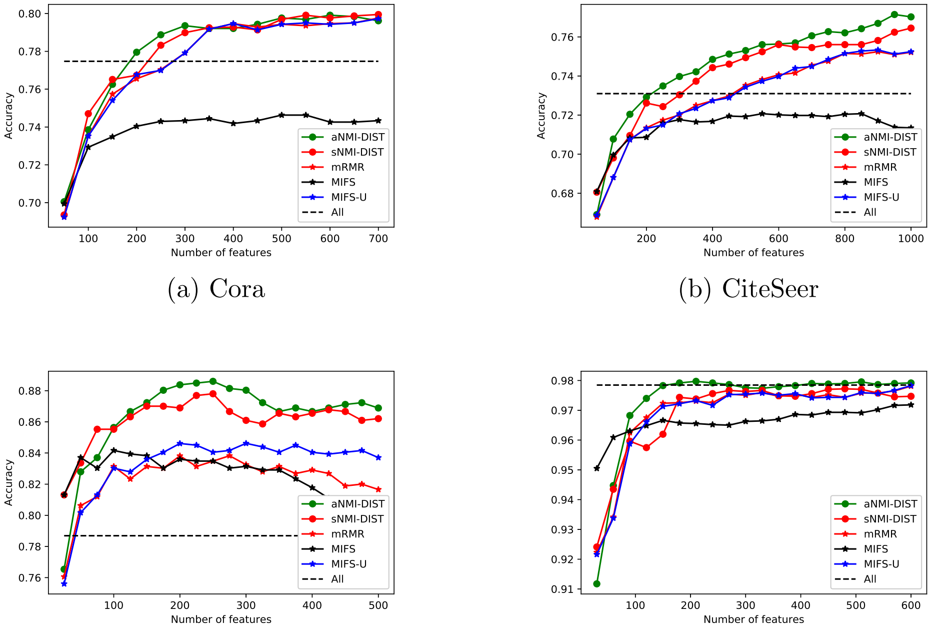
(a) Cora

(b) CiteSeer

(c) WebKB

(d) SMS spam (binary)

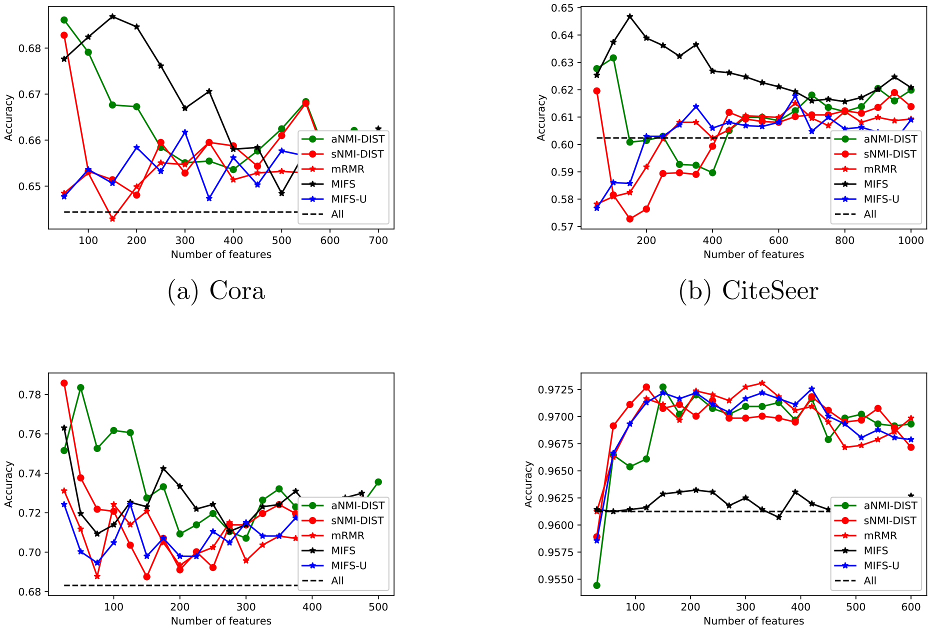
Figure 4.17: Accuracy comparison with MI-based methods on document classification datasets using SVM



(a) Cora (b) CiteSeer

(c) WebKB (d) SMS spam (binary)

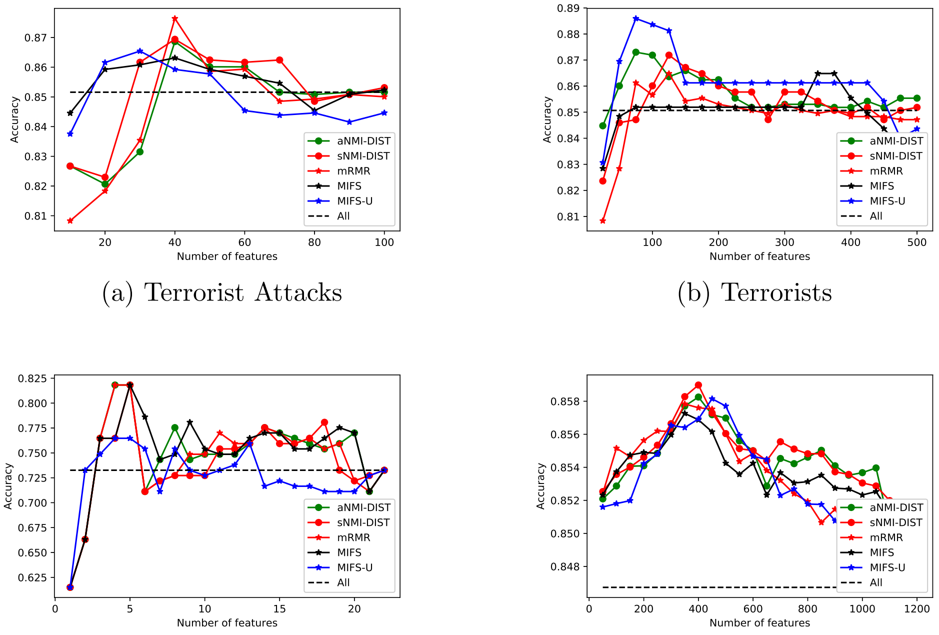
Figure 4.18: Accuracy comparison with MI-based methods on document classification datasets using Naive Bayes



(a) Cora (b) CiteSeer

(c) WebKB (d) SMS spam (binary)

Figure 4.19: Accuracy comparison with MI-based methods on document classification datasets using Decision Trees



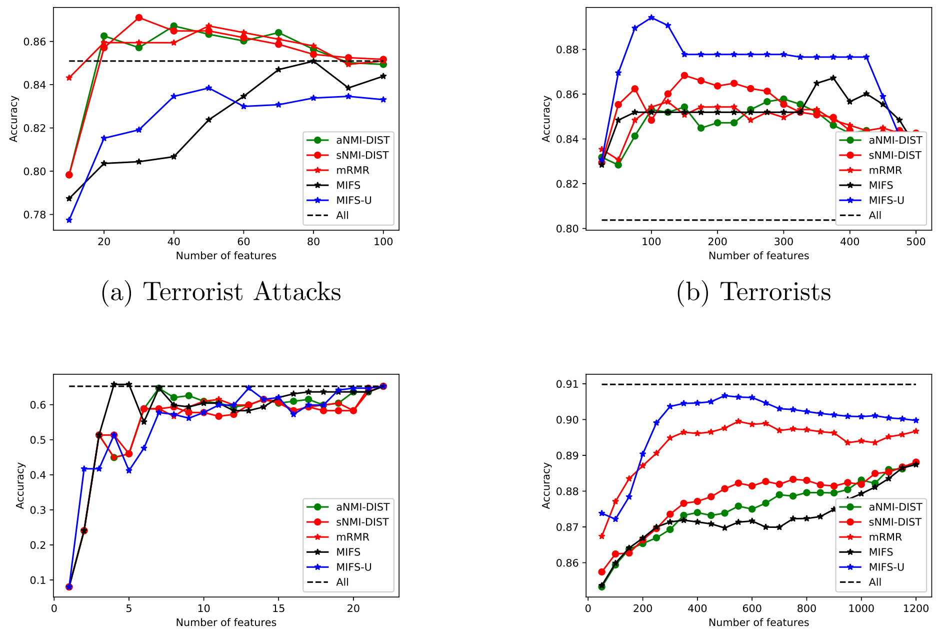
(a) Terrorist Attacks

(b) Terrorists

(c) SPECT

(d) HIVA

Figure 4.20: Accuracy comparison with MI-based methods on Cyber threat and Medical diagnosis datasets using SVM



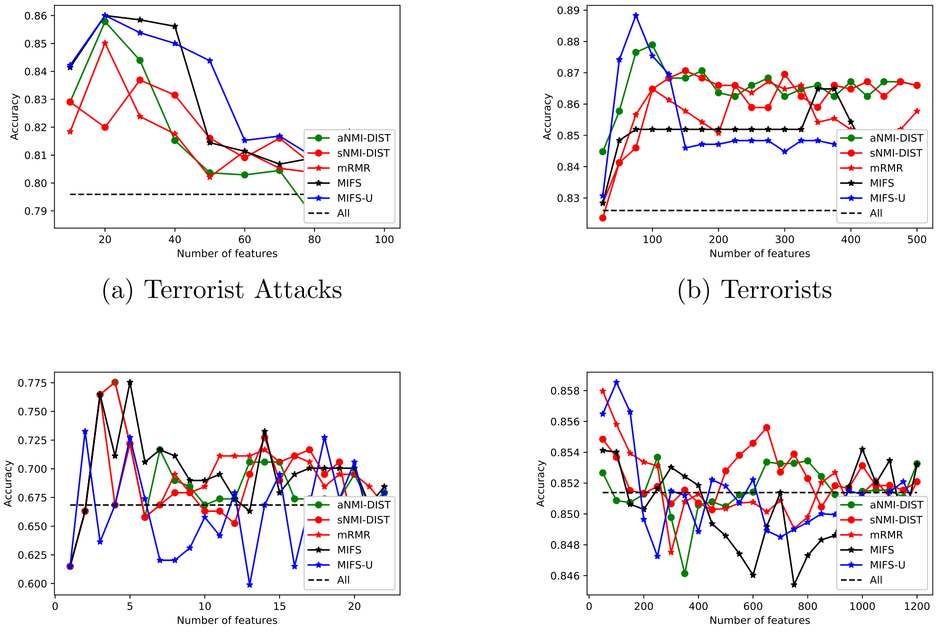
(a) Terrorist Attacks

(b) Terrorists

(c) SPECT

(d) HIVA

Figure 4.21: Accuracy comparison with MI-based methods on Cyber threat and Medical diagnosis datasets using Naive Bayes



(a) Terrorist Attacks

(b) Terrorists

(c) SPECT

(d) HIVA

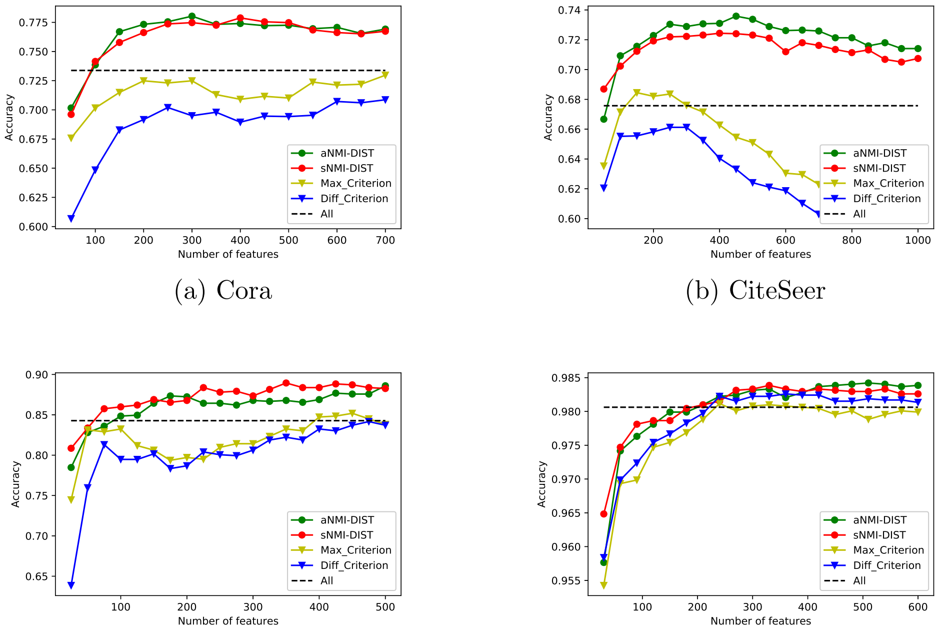
Figure 4.22: Accuracy comparison with MI-based methods on Cyber threat and Medical diagnosis datasets using Decision Trees

classifiers and datasets, and it does not properly express the significance of redundancy criterion to a selected feature set. For this reason, $\beta = 0.5$ is set for MIFS method and $\beta = 1$ is set for MIFS-U method as default setting in the original works to conduct experiments in this thesis.

On document classification datasets, aNMI-DIST and sNMI-DIST perform dominantly against mRMR and MIFS-based methods for all classifiers run except for Decision Trees on CiteSeer dataset. However, the classification accuracy in this case is no more than 65%, far from the best accuracy of around 74% and 77% conducted by SVM and Naive Bayes respectively. SMS spam binary dataset in particular, only aNMI-DIST can overcome the whole-feature baseline with only 200 feature selected using Naive Bayes. It also can be seen that MI-based methods apparently work better than standard filter methods in text classification tasks and outperform the "All" baseline because these datasets have a larger number of features than medical and terrorist attacks datasets.

In cyber threat and medical diagnosis datasets, MIFS-based methods seem to work better in classification prediction on Terrorists in all three classifiers, and on HIVA dataset using Naive Bayes and Decision Trees. Note that from the statistical property of Terrorist dataset, it has many zero-features (288 features). This leads to the entropy values of these features becoming to be very small, making the redundancy component expressed by MIFS-U method significantly larger. As the illustration of t-SNE mapping in Figure 4.5, there are a considerable number of training samples in the same neighborhood. In consequence, redundancy among features may become more important thereby MIFS-U with an entropy coefficient in the redundancy term is able to properly select a good set of features feeding into the classifier. In terms of SPECT dataset, NMI-based methods and MI-based methods achieve relatively similar highest results. The cause may be due to a small number of features, and these all features are very informative and there is not much redundancy among them. This is understandable because features collected from medical datasets are often distilled and make a significant contribution to medical classification tasks.

4.4.3 Comparison with Feature Selection Methods for Binary Data



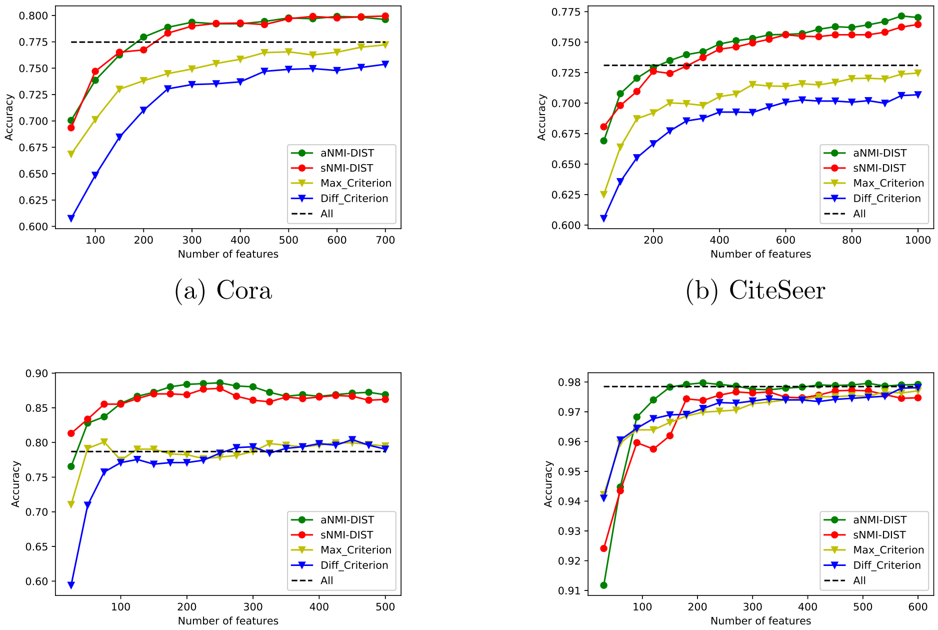
(a) Cora

(b) CiteSeer

(c) WebKB

(d) SMS spam (binary)

Figure 4.23: Accuracy comparison with methods for binary features on document classification datasets using SVM



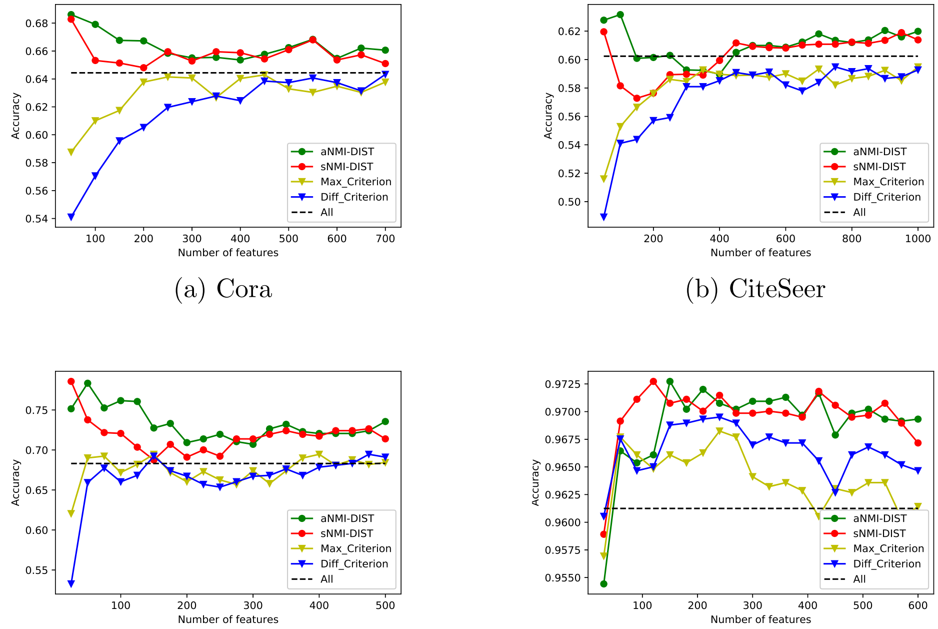
(a) Cora

(b) CiteSeer

(c) WebKB

(d) SMS spam (binary)

Figure 4.24: Accuracy comparison with methods for binary features on document classification datasets using Naive Bayes



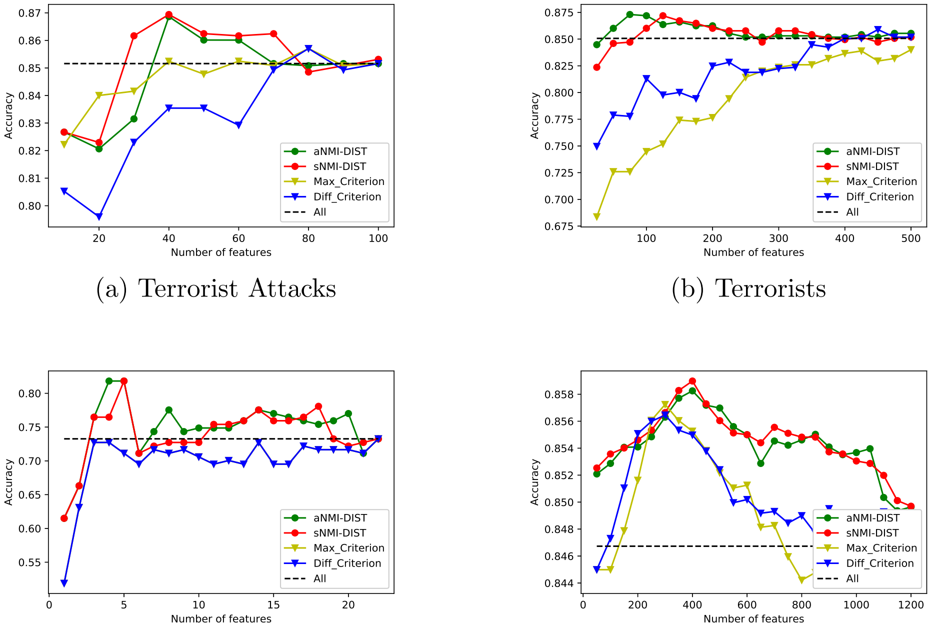
(a) Cora

(b) CiteSeer

(c) WebKB

(d) SMS spam (binary)

Figure 4.25: Accuracy comparison with methods for binary features on document classification datasets using Decision Trees



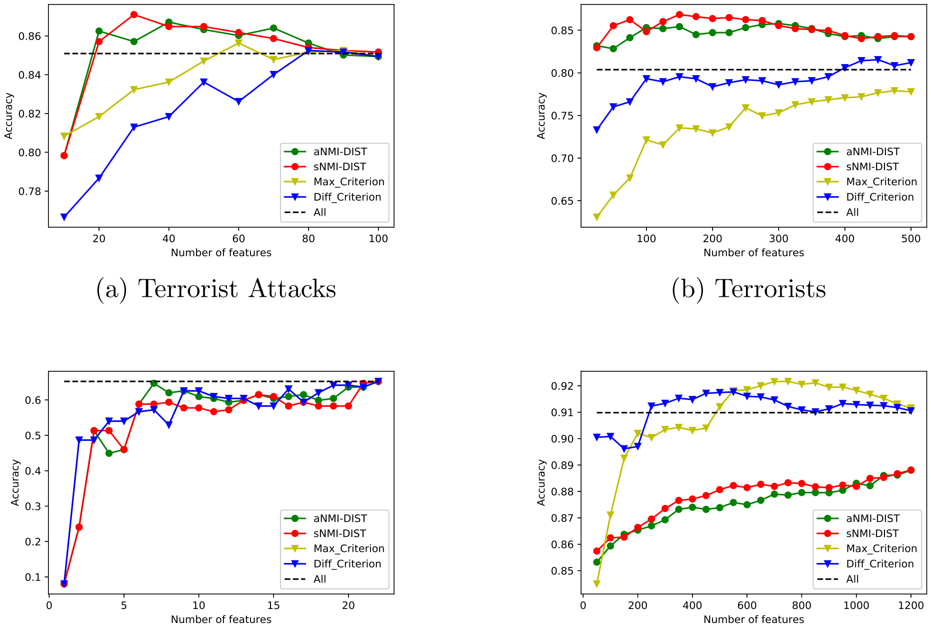
(a) Terrorist Attacks

(b) Terrorists

(c) SPECT

(d) HIVA

Figure 4.26: Accuracy comparison with methods for binary features on Cyber threat and Medical diagnosis datasets using SVM



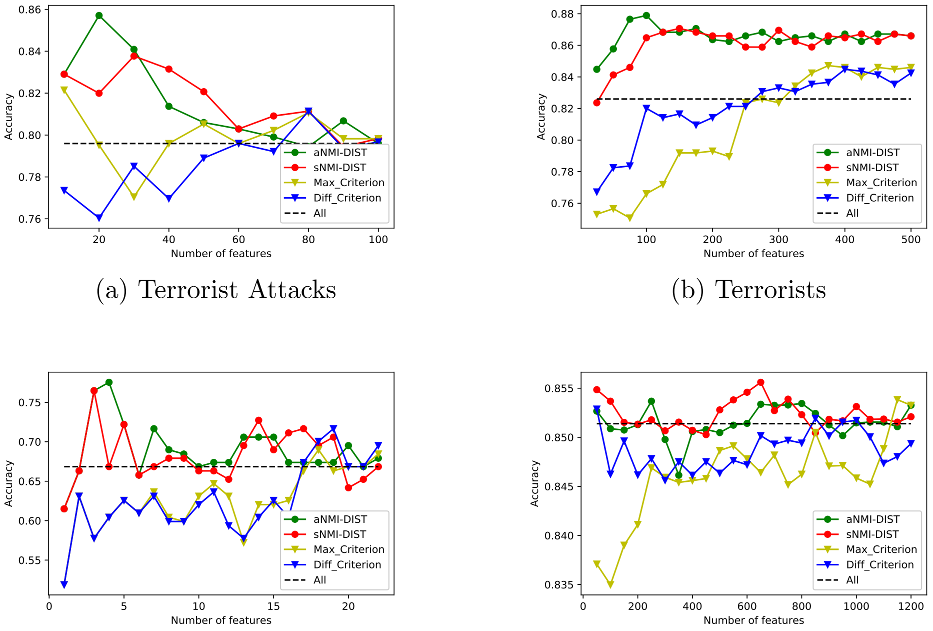
(a) Terrorist Attacks

(b) Terrorists

(c) SPECT

(d) HIVA

Figure 4.27: Accuracy comparison with methods for binary features on Cyber threat and Medical diagnosis datasets using Naive Bayes



(a) Terrorist Attacks

(b) Terrorists

(c) SPECT

(d) HIVA

Figure 4.28: Accuracy comparison with methods for binary features on Cyber threat and Medical diagnosis datasets using Decision Trees

Although many datasets in various real-world problems are represented by binary data, there are a limited number of feature selection methods dedicated to those features. Most of feature selection methods are proposed for more general types of data such as continuous or mixed discrete data. Class-dependent density-based feature elimination (CDFE) [37] is an efficient method specifically designed for high dimensional binary data by exploiting some statistical characteristics of binary variables. This scheme has shown a good performance in reducing the feature set size as well as the computational cost. However, a major weakness of CDFE is that it originated to rank binary features for two-class categorization problems. This section compares the proposed methods that can be applied for multi-class problems with two variants of CDFE, namely Max-criterion and Diff-criterion, which are limited for binary classification.

As results obtained, aNMI-DIST and sNMI-DIST perform much better than CDFE methods in most cases, except for on HIVA dataset using Naive Bayes due to the statistics of this dataset previously explained. The conspicuous difference between NMI-based methods and Max-criterion method and Diff-criterion method is mainly from multi-class datasets, where some restrictions of Max-criterion and Diff-criterion are manifested. In Cora and Citeseer dataset, aNMI-DIST seems to slightly work better than sNMI-DIST, and these both methods significantly exceed CDFE methods with accuracy gaps around 5%-13%. A similar pattern is presented on WebKB dataset, yet sNMI-DIST overcomes moderately aNMI-DIST when 200-500 features are selected using SVM.

Regarding binary class datasets, Max-criterion and Diff-criterion can be competitive against NMI-based methods in some cases. On SMS spam binary dataset and SPECT dataset, Diff-criterion relatively achieves the same accuracy as sNMI-DIST, and Max-criterion performs only 0.5% worse than aNMI-DIST when using SVM, Naive Bayes and Decision Trees. Although NMI-based methods outperform CDFE methods on HIVA datasets when using SVM and Decision Trees, difference between them is negligible. The only case the proposed methods remarkably outperform CDFE methods in the binary classification task is on Terrorists dataset, where redundancy plays a pivotal role in evaluating a set of features selected.

4.4.4 Comparison with Wrapper Methods

Wrapper methods have shown their power in feature selection. However, they are extremely time consuming because of the laborious process of simultaneously selecting features and training a classifier. This section shows comparisons between the proposed methods and two well-known wrapper methods, which are Sequential Forward Selection (SFS) and Sequential Forward Floating Selection (SFFS). Experiments are performed on an Intel(R) 64-bit, 4 Cores(TM) i5-4440 CPU, 3.10GHz machine, and are conducted on three datasets in different domains and different lengths of feature dimension including (WebKB: high dimension, Terrorist Attacks: medium dimension, and SPECT: low dimension). Note that only the best performance of comparative methods will be given. In SPECT dataset, the proposed methods achieve more accurate predictions than SFS and SFFS with much less training time when using SVM. Although these wrapper methods perform the same classification accuracy with the proposed methods when using Naive Bayes and Decision Trees, they take much more training time. The wrapper methods seem to work better than the proposed methods on Terrorist Attacks dataset. However they also need much more computational costs

Table 4.2: Accuracy comparison with Wrapper methods

Dataset	**Method**	**SVM**			**NB**			**DT**		
		d	*Acc*	*Time*	*d*	*Acc*	*Time*	*d*	*Acc*	*Time*
WebKB	*SFS*	75	88.61	1h48m	93	84.96	22m38s	98	81.21	45m23s
	SFFS	50	88.27	2h26m	270	87.25	5h47m	109	82.35	4h18m
	sNMI-DIST	430	89.29	3m56s	230	88.26	2m23s	25	78.69	2m48s
	aNMI-DIST	500	88.61	4m14s	210	88.72	2m23s	50	78.34	2m02
Terrorist Attacks	*SFS*	20	87.32	1m51s	54	89.64	0m51s	27	88.02	0m28s
	SFFS	40	90.27	8m12s	36	89.11	1m36s	19	87.95	2m32s
	sNMI-DIST	36	87.10	< 1s	30	87.10	< 1s	26	85.40	< 1s
	aNMI-DIST	36	88.26	< 1s	39	86.95	< 1s	21	85.94	< 1s
SPECT	*SFS*	14	77.01	0m26s	22	65.24	0m23s	3	77.54	0m21s
	SFFS	14	73.80	0m38s	22	65.24	0m20s	3	77.54	0m20s
	sNMI-DIST	5	81.82	< 1s	22	65.24	< 1s	3	76.47	< 1s
	aNMI-DIST	4	81.82	< 1s	22	65.24	< 1s	4	77.54	< 1s

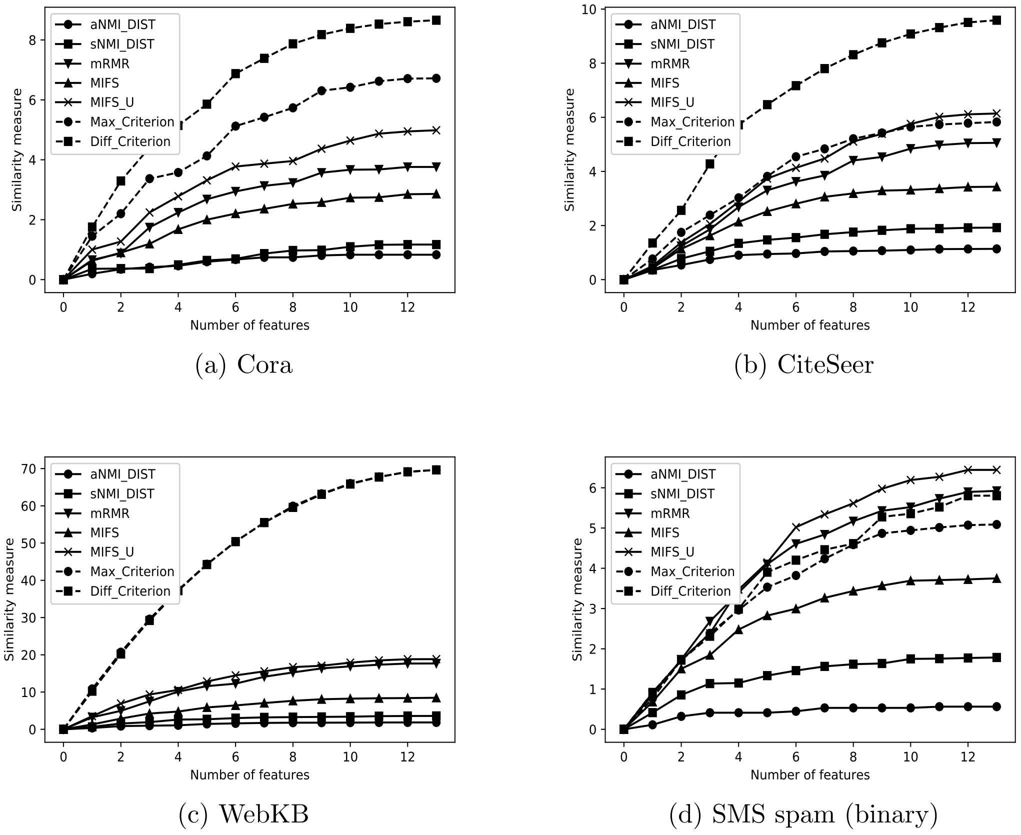
to perform the feature selection task and training the classifier simultaneously. Intuitively, the wrapper methods tend to rapidly increase running time when the number of features increases. This trend can be observed from WebKB dataset with more than 1700 features. Summary results are shown in Table 4.2, where d is the number of features selected, Acc is the classification accuracy, and $Time$ presents running time of a classifier on the given machine.

4.4.5 Feature Similarity Insight

After selecting features, this section investigates how methods affect the similarity of chosen features, which is an important component for the redundancy of the selected set.

Figure 4.30 shows similarity measures of selected features on document classification datasets. In all cases, similarity among features chosen by aNMI-DIST is the lowest followed by that of sNMI-DIST method. In contrast, the similarity measures of the others tend to significantly increase after a small number of features are obtained. This means that the proposed methods can considerably reduce the redundancy among selected features, as Proposition 3.3.

On the cyber threat datasets, MIFS seems to hold the smallest irrelevant features. However, it does not perform better than NMI-based methods and the other MI-based methods. This may be because classification prediction gets stronger influence from



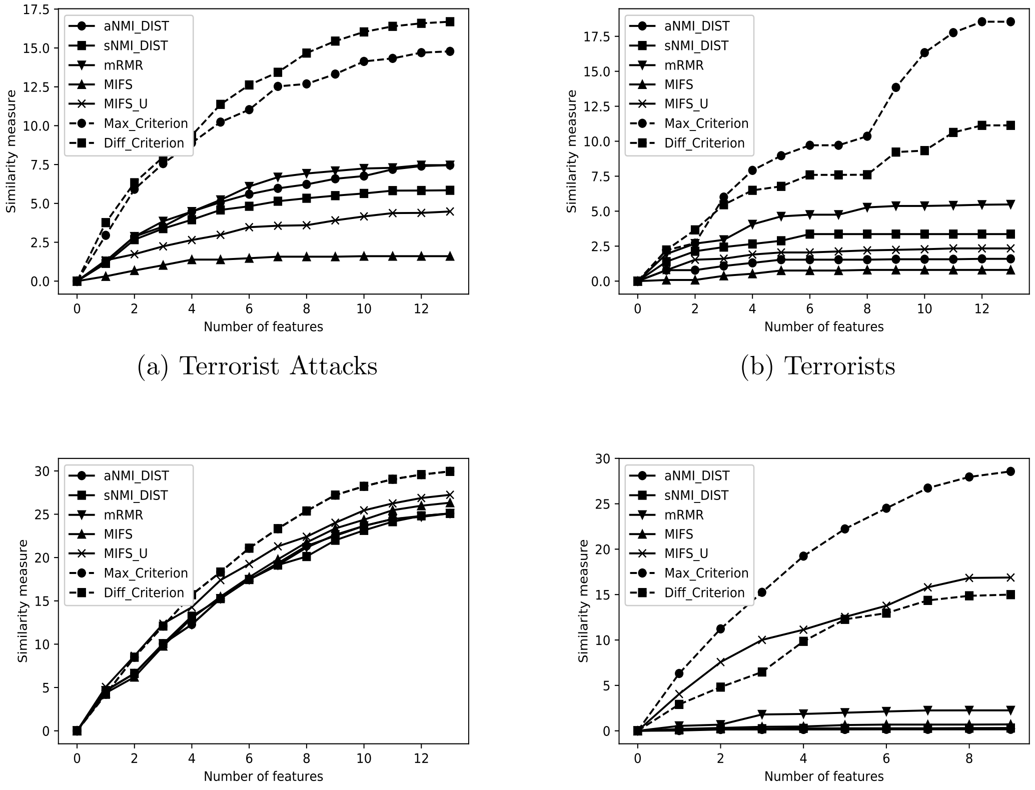
(a) Cora

(b) CiteSeer

(c) WebKB

(d) SMS spam (binary)

Figure 4.29: Similarity measure of features selected on binary document classification datasets



(a) Terrorist Attacks

(b) Terrorists

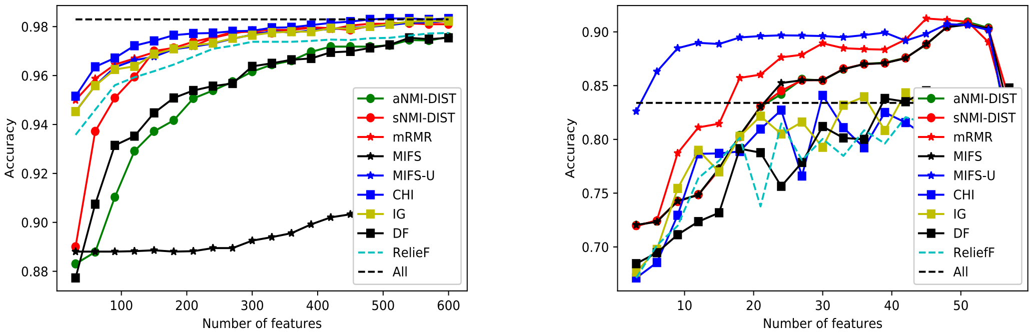
(c) SPECT

(d) HIVA

Figure 4.30: Similarity measure of features selected on binary cyber threat and medical diagnosis datasets

relevant features while MIFS tends to select features having less redundancy (i.e. the redundant term dominates the relevant term in MIFS algorithm). Note that the importance of redundancy can be varied in some situations and contingent to classifiers. For example, in some datasets the relevance among features are noticeable enough while variations among relevance values between features and the class vector are subtle. This may lead to MI-based methods are in favour of features having small redundancy. As in Terrorist Attack dataset, MIFS results in the best performance when using Decision Trees, but poor performance compared to the others when using SVM and Naive Bayes. Similar trends also can be recognized when performing on datasets Terrorists, SPECT, and HIVA. However there is not considerable difference between MI-based and NMI-based methods. Note that on SPECT dataset redundancy becomes significantly large by all feature selection methods examined. Furthermore, Max-criterion and Diff-criterion are not good methods to reduce redundancy among features.

4.4.6 Performance on Non-binary Data

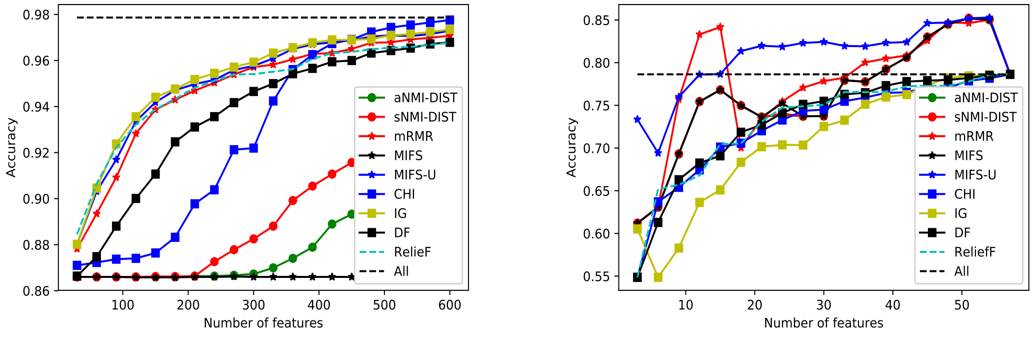


(a) SMS spam (tf-idf)

(b) Spambase

Figure 4.31: Accuracy benchmark of comparative methods on non-binary SMS spam (tf-idf) dataset using SVM

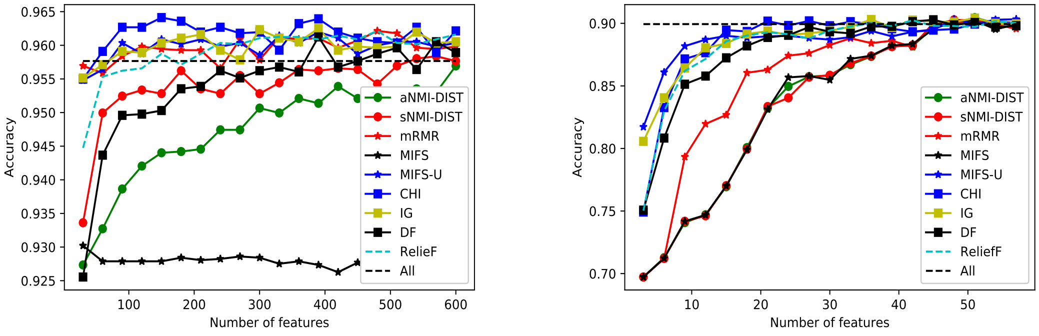
This section verifies how the proposed methods perform on non-binary datasets. Experiments are conducted on SMS spam that is represented by *tf-idf* vectors, and on Spambase dataset. As Accuracy and F-measure results shown in Figure Table 4.3 and 4.4, the proposed methods could not draw on the submodular property of binary data to facilitate feature selection effectiveness. Note that in the F-measure



(a) SMS spam (tf-idf)

(b) Spambase

Figure 4.32: Accuracy benchmark of comparative methods on non-binary SMS spam (tf-idf) dataset using Naive Bayes



(a) SMS spam (tf-idf)

(b) Spambase

Figure 4.33: Accuracy benchmark of comparative methods on non-binary SMS spam (tf-idf) dataset using Decision Trees

Table 4.3: F-measure comparison on SMS spam (tf-idf)

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	92.55	94.87	0.020	85.91	92.30	0.000	91.11	**92.19**	0.000
DF	88.61	94.47	0.603	81.07	92.39	0.000	90.03	91.42	0.003
IG	93.80	96.00	0.004	**86.98**	**93.75**	0.000	**91.13**	91.64	0.000
CHI	**94.65**	**96.25**	0.001	76.35	**94.81**	0.000	**91.38**	**91.99**	0.000
mRMR	**94.10**	96.00	0.003	85.30	93.06	0.000	91.04	91.56	0.000
MIFS	69.77	78.88	0.000	46.64	47.35	0.001	84.00	85.06	0.000
MIFS-U	93.81	**96.04**	0.004	**86.53**	93.61	0.000	91.03	91.56	0.000
sNMI-DIST	92.31	95.79	0.057	61.77	82.68	0.128	89.65	91.30	0.009
aNMI-DIST	87.15	94.38	1.000	55.44	77.00	1.000	88.20	90.28	1.000

Table 4.4: F-measure comparison on Spambase

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	75.62	83.34	0.007	71.82	77.57	0.630	87.58	89.82	0.001
DF	74.21	85.78	0.014	71.75	77.57	0.618	87.31	89.91	0.021
IG	76.48	83.48	0.014	**69.51**	77.57	0.258	**88.17**	**90.06**	0.003
CHI	77.53	83.97	0.277	69.51	77.57	0.596	87.95	**90.08**	0.009
mRMR	**83.72**	**90.48**	0.283	**75.14**	84.73	0.659	84.58	**90.06**	0.369
MIFS	81.22	**90.42**	0.917	73.43	**85.08**	0.986	82.49	89.80	0.995
MIFS-U	**87.68**	90.31	0.001	**80.05**	**85.07**	0.036	**88.08**	89.82	0.004
sNMI-DIST	81.24	90.35	0.995	73.36	**85.07**	0.998	82.38	89.79	0.998
aNMI-DIST	80.96	90.35	1.000	73.36	**85.07**	1.000	82.51	89.91	1.000

comparison tables, $Mean$ is defined as the average F-measure over the total F-measure values calculated, and Max indicates the maximal F-measure result performed by a classifier. The tables also report the statistical hypothesis Student's t-Test with significance threshold $= 0.05$, to check if these mean values of the baseline methods are significantly different from the proposed method aNMI-DIST, which are represented by *p-value* with 3 digits after the decimal separator.

4.4.7 Summary

In overall, the proposed methods, namely aNMI-DIST and sNMI-DIST, perform better than comparative methods on various binary datasets by different classifiers. More particular observations from experimental results can be summarized as follow:

- aNMI-DIST and sNMI-DIST show as the promising feature selection methods for binary representation data, especially on text classification tasks where the number of words is considerably large and having many redundant words, which can be effectively eliminated by the proposed methods. For synthesis or medical datasets where the number of features is rather small and features are informative, these proposed methods also seem to work well but are not markedly superior to other methods, especially MI-based methods.

- The proposed methods work well on both binary-class and multi-class problems. However, in some cases CDFE methods can be effective with high classification accuracy and less computational cost. Furthermore, CHI shows to be a competitive candidate for imbalanced data (e.g. Terrorist Attacks or HIVA dataset).

- aNMI-DIST method performs better than sNMI-DIST method for document classification datasets on average. Although both methods are penalized by balanced entropy quantities, the relevance term defined by sNMI-DIST tends to have higher influence to the objective function, leading to reduce the importance of the redundancy term that is reasoned as a crucial property of these datasets.

- SVM should be the recommended classifier for the proposed methods where they mostly achieve the best performance. Although the proposed methods may result in poor prediction in some cases by using Decision Trees, classification accuracy of other methods obtained is generally not good as that of SVM or Naive Bayes.

- Despite the effectiveness of the proposed methods has been shown by classification accuracy, F-measure should also be justified. In overall, F-measure results reflect the performance similar to that of the comparative methods examined by accuracy, except for unbalanced data such as Terrorist Attacks. Summary of F-measure of given methods conducted on various datasets is reported in Table 4.5-4.12.

Table 4.5: F-measure comparison on Cora

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	65.67	68.52	0.00	69.52	69.52	0.00	59.16	62.43	0.00
DF	64.72	67.90	0.00	68.74	73.52	0.00	58.02	62.15	0.00
IG	72.92	74.60	0.08	75.60	77.81	0.82	62.46	63.49	0.00
CHI	72.85	74.77	0.07	75.41	77.57	0.69	62.74	64.16	0.01
Max-criterion	69.07	70.47	0.00	72.96	75.45	0.02	60.49	61.88	0.00
Diff-criterion	66.24	68.40	0.00	70.21	73.58	0.00	59.11	61.79	0.00
mRMR	72.94	74.64	0.09	75.63	77.80	0.84	62.38	62.90	0.00
MIFS	70.62	71.66	0.00	71.58	72.50	0.00	62.95	65.08	0.31
MIFS-U	72.93	74.61	0.08	75.61	77.74	0.83	62.58	63.36	0.00
sNMI-DIST	**74.01**	**75.64**	0.79	**75.89**	**78.08**	0.98	**63.06**	**65.46**	0.32
aNMI-DIST	**74.23**	**75.68**	1.00	**75.86**	**77.80**	1.00	**63.34**	**65.66**	1.00

Table 4.6: F-measure comparison on CiteSeer

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	58.21	60.48	0.000	63.19	65.89	0.000	53.57	55.60	0.000
DF	58.15	61.50	0.000	63.68	66.08	0.000	53.42	55.63	0.000
IG	64.78	66.58	0.000	68.71	71.52	0.051	56.44	57.39	0.202
CHI	65.78	67.48	0.000	69.25	71.92	0.192	56.65	57.95	0.386
Max-criterion	60.86	64.34	0.000	66.38	68.70	0.000	54.71	56.58	0.000
Diff-criterion	58.77	62.12	0.000	64.25	66.68	0.000	53.74	55.96	0.000
mRMR	64.94	66.45	0.000	68.80	71.38	0.070	56.41	57.47	0.049
MIFS	65.33	66.80	0.000	66.90	67.82	0.000	**59.10**	**61.49**	0.000
MIFS-U	65.01	66.77	0.000	68.83	71.43	0.071	56.44	58.05	0.311
sNMI-DIST	**67.20**	**68.56**	0.066	**69.67**	**72.61**	0.383	56.48	58.20	0.243
aNMI-DIST	**68.18**	**69.80**	1.00	**70.48**	**73.53**	1.00	**56.91**	**59.15**	1.00

Table 4.7: F-measure comparison on WebKB

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	67.83	67.83	0.000	67.83	67.61	0.000	67.61	55.90	0.000
DF	66.43	72.28	0.000	62.69	67.96	0.000	53.19	56.52	0.000
IG	71.71	75.33	0.007	68.00	71.46	0.000	56.66	58.66	0.000
CHI	69.71	74.49	0.001	66.90	71.45	0.000	56.50	59.31	0.000
Max-criterion	69.44	74.21	0.000	65.91	68.15	0.000	54.05	57.43	0.000
Diff-criterion	66.16	71.25	0.000	63.55	68.98	0.000	52.74	55.41	0.000
mRMR	72.57	75.56	0.037	69.15	72.65	0.000	56.99	59.75	0.000
MIFS	69.16	72.13	0.000	68.76	71.23	0.000	**61.74**	63.49	0.016
MIFS-U	73.29	76.09	0.172	70.38	73.85	0.008	56.94	60.30	0.000
sNMI-DIST	**76.24**	**78.91**	0.205	**75.44**	**78.83**	0.771	58.47	**64.87**	0.143
aNMI-DIST	**74.81**	**78.22**	1.000	**75.05**	**79.22**	1.000	**60.11**	**66.27**	1.000

Table 4.8: F-measure comparison on SMS Spam (binary)

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	94.07	95.75	0.039	92.93	94.51	0.634	91.25	92.37	0.000
DF	93.16	95.47	0.005	91.61	94.71	0.224	90.54	91.89	0.000
IG	95.25	96.42	0.503	92.59	**95.34**	0.546	**93.10**	**93.92**	0.324
CHI	94.97	96.35	0.229	90.52	94.90	0.071	92.85	93.61	0.790
Max-criterion	94.83	95.81	0.133	93.28	95.02	0.833	91.92	92.98	0.003
Diff-criterion	95.22	96.12	0.442	**93.43**	95.25	0.926	92.42	93.23	0.246
mRMR	95.32	**96.50**	0.583	92.77	95.30	0.617	**93.10**	93.69	0.426
MIFS	92.65	93.22	0.000	92.27	93.74	0.304	91.05	91.35	0.000
MIFS-U	95.26	96.34	0.512	92.64	**95.34**	0.567	93.03	**93.81**	0.406
sNMI-DIST	**95.67**	96.38	0.848	92.73	95.11	0.590	93.04	93.49	0.273
aNMI-DIST	**95.59**	**96.47**	1.000	**93.56**	**95.60**	1.000	92.81	93.52	1.000

Table 4.9: F-measure comparison on Terrorist Attacks

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	47.94	51.57	0.137	47.95	51.23	0.349	47.18	49.87	0.484
DF	47.65	51.98	0.126	47.80	51.53	0.353	46.85	50.72	0.178
IG	50.18	53.88	0.826	49.60	52.18	0.959	49.24	**50.85**	0.980
CHI	48.50	**54.76**	0.493	47.73	51.74	0.525	45.97	49.57	0.194
Max-criterion	49.71	51.29	0.552	**50.15**	**53.27**	0.658	49.41	51.25	0.975
Diff-criterion	47.77	52.10	0.114	48.16	51.43	0.401	47.78	50.35	0.324
mRMR	49.98	**55.92**	0.745	49.57	52.02	0.977	48.28	50.02	0.339
MIFS	49.88	51.13	0.589	48.29	50.25	0.263	**50.26**	**53.07**	0.084
MIFS-U	50.51	51.61	0.976	48.61	50.28	0.417	**52.13**	**56.01**	0.014
sNMI-DIST	**51.08**	54.67	0.738	**49.77**	51.16	0.860	48.41	50.36	0.802
aNMI-DIST	**50.54**	53.73	1.000	49.54	**52.85**	1.000	49.18	51.18	1.000

Table 4.10: F-measure comparison on Terrorists

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	75.37	83.30	0.000	69.08	76.31	0.000	76.53	84.99	0.000
DF	76.18	81.56	0.000	66.51	75.22	0.000	78.16	83.05	0.000
IG	82.74	84.71	0.049	81.00	83.21	0.000	82.62	84.44	0.000
CHI	55.01	82.38	0.000	54.44	80.81	0.000	55.26	82.65	0.000
Max-criterion	77.76	83.15	0.000	70.12	76.32	0.000	79.42	84.01	0.000
Diff-criterion	80.37	85.28	0.000	76.32	80.52	0.000	80.24	84.00	0.000
mRMR	83.73	85.31	0.913	83.61	84.48	0.713	**84.80**	86.26	0.592
MIFS	83.61	85.39	0.643	83.86	**85.68**	0.278	83.60	85.35	0.024
MIFS-U	**85.08**	**87.59**	0.008	**86.21**	**88.67**	0.000	84.32	**87.90**	0.731
sNMI-DIST	**84.23**	**86.14**	0.305	**84.15**	**85.68**	0.086	**85.22**	86.11	0.153
aNMI-DIST	83.79	84.99	1.000	83.48	84.89	1.000	84.56	**86.40**	1.000

Table 4.11: F-measure comparison on SPECT

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	57.12	60.46	0.171	**45.87**	50.96	0.647	52.75	56.66	0.047
DF	54.45	58.72	0.000	**47.81**	**53.19**	0.267	50.80	55.09	0.000
IG	57.38	60.41	0.192	44.09	**51.71**	0.885	52.27	57.00	0.016
CHI	57.38	59.52	0.184	44.05	**51.71**	0.876	52.21	57.00	0.017
Max-criterion	55.83	58.72	0.007	45.75	50.95	0.675	50.31	54.32	0.000
Diff-criterion	55.83	58.72	0.007	45.75	50.95	0.675	50.47	55.48	0.000
mRMR	58.16	**64.30**	0.785	44.04	50.52	0.875	**55.00**	**61.36**	0.714
MIFS	**58.69**	**64.30**	0.751	45.50	51.68	0.753	**55.51**	**61.36**	0.236
MIFS-U	57.32	59.52	0.164	44.58	**51.71**	0.984	52.72	57.00	0.047
sNMI-DIST	57.94	**64.30**	0.602	43.95	**51.71**	0.851	54.29	59.52	0.390
aNMI-DIST	**58.41**	**64.30**	1.000	44.52	50.52	1.000	54.94	**61.36**	1.000

Table 4.12: F-measure comparison on HIVA

Method	**SVM**			**NB**			**DT**		
	Mean	*Max*	*p-value*	*Mean*	*Max*	*p-value*	*Mean*	*Max*	*p-value*
ReliefF	54.32	56.26	0.175	76.80	82.10	0.000	56.53	57.82	0.503
DF	54.21	56.74	0.159	77.39	**83.12**	0.000	56.59	**58.56**	0.282
IG	54.44	56.34	0.144	77.62	81.16	0.000	56.91	58.20	0.074
CHI	54.55	56.86	0.197	73.51	80.85	0.010	56.81	**58.94**	0.274
Max-criterion	54.43	56.77	0.237	**79.58**	**84.03**	0.000	56.79	58.34	0.224
Diff-criterion	54.52	57.02	0.244	**81.11**	82.85	0.000	**57.09**	58.47	0.100
mRMR	**55.57**	57.19	0.827	74.53	78.24	0.000	56.58	58.00	0.174
MIFS	55.42	57.74	0.797	66.48	75.09	0.731	56.27	58.08	0.960
MIFS-U	55.11	57.31	0.641	76.19	79.04	0.000	**58.08**	58.33	0.243
sNMI-DIST	**55.55**	**57.80**	0.878	68.15	75.18	0.569	56.28	57.76	0.819
aNMI-DIST	55.44	**58.05**	1.000	67.10	75.25	1.000	56.15	58.16	1.000

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Social media generates a huge amount of data every day, and most of them are formed of texts and images. This may cause a plethora of redundant features that need to be manipulated. An effective feature selection method to circumvent the limitations should be developed as the growth in demand for data mining has been getting bigger. This thesis reviews several popular feature selection methods in the both filter and wrapper family, and analyzes limitations of these methods in certain situations. Mutual information-based feature selection is shown to be the appropriate method in many tasks. However, the number of studies on text categorization using MI-based methods is limited.

The proposed method attempts to adapt the idea of MI-based methods and design a novel feature selection strategy to efficiently perform on text data. To this end, an approximation solution of the heuristic search is propounded by reaping the benefit of submodular properties on binary variables. In detail, the alternative greedy search strategy is based on a new normalized relevance measure that can alleviate the effect of mutual information increasing. A distance metric function is also used to substitute the redundancy functions in previous studies. By theoretical analysis, the proposed method can provide the approximation ratio of $\frac{1}{2}$ for the greedy algorithm. Performance of the proposed method is validated and benchmarked against different baseline methods. Experimental results show their effectiveness in the binary representation text classification task. Comparisons examined on additional datasets also show that the new proposal is a promising approach on binary data in various dataset domains.

5.2 Future Work

This study is demonstrated to perform efficiently on binary data. Accordingly, the aim of future work is to extend the proposed method to non-binary data. In terms of scalability, the computational complexity of the alternative greedy search in this research is the same with that of MI-based filter selection methods and is significantly lower than that of wrapper methods. To handle the problem of high dimensional data, a new searching mechanism needs to be developed. Note that the suggested objective function is proven to hold the submodular properties, hence an accelerated greedy algorithm can be applied to facilitate the computational process. Moreover, submodularity is very effective in social networks where nodes are linked by the influence weights. These are not investigated in the proposed method and will be explored in future work.

Bibliography

[1] S. García, J. Luengo, F. Herrera, Tutorial on practical tips of the most influential data preprocessing algorithms in data mining, Knowl.-Based Syst. 98 (2016) 1-29

[2] J. Han, M. Kamber, J. Pei, Data mining: concepts and techniques: concepts and techniques, Elsevier, 2011

[3] F.G. Mohammadi, M.S. Abadeh, Image steganalysis using a bee colony based feature selection algorithm, Eng. Appl. Artif. Intell. 31 (2014) 35-43

[4] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (2005) 491-502

[5] Lee, P.Y., Loh, W.P. and Chin, J.F., 2017. Feature selection in multimedia: The state-of-the-art review. Image and vision computing, 67, pp.29-42.

[6] Forman, G.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification. J. Machine Learning Research **3**(Mar), 1289-1305 (2003)

[7] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**(Mar), 1157-1182 (2003)

[8] Darshan, SL Shiva, and C. D. Jaidhar. "Performance evaluation of filter-based feature selection techniques in classifying portable executable files." Procedia Computer Science 125 (2018): 346-356.

[9] Ke, Wenjun, Chunxue Wu, Yan Wu, and Neal N. Xiong. "A new filter feature selection based on criteria fusion for gene microarray data." IEEE Access 6 (2018): 61065-61076.) was proposed to compare with established methods for cancer prediction.

[10] Hoque, Nazrul, Mihir Singh, and Dhruba K. Bhattacharyya. "EFS-MI: an ensemble feature selection method for classification." Complex & Intelligent Systems 4, no. 2 (2018): 105-118.

[11] Ghosh, Manosij, Sukdev Adhikary, Kushal Kanti Ghosh, Aritra Sardar, Shemim Begum, and Ram Sarkar. "Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods." Medical & biological engineering computing 57, no. 1 (2019): 159-176.

[12] Bommert, Andrea, Xudong Sun, Bernd Bischl, Jörg Rahnenführer, and Michel Lang. "Benchmark for filter methods for feature selection in high-dimensional classification data." Computational Statistics Data Analysis 143 (2020): 106839.

[13] Robnik-Šikonja, Marko, and Igor Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. In Machine learning 53, no. 1-2 (2003): 23-69.

[14] Vora, S.,Yang, H.: A comprehensive study of eleven feature selection algorithms and their impact on text classification. In: 2017 Computing Conference, pp. 440-449 (2017)

[15] Yang, Y., Pedersen, J. O.: A comparative study of feature selection in text categorization. In: Proc. 14th Int. Conf. Machine Learning (1997)

[16] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M.W. Mahoney, "Feature Selection Methods for Text Classification," in *Proc. 13th Int'l Conf. Knowledge Discovery and Data Mining (KDD'07)*, pp. 230-239, 2007

[17] M. Rogati and Y. Yang, "High-Performing Feature Selection for Text Classification," in *Proc. CIKM '02: Eleventh Int'l Conf. Information and Knowledge Management*, pp. 659-661, 2002

[18] Z. Zheng, X. Wu, and R. Srihari, "Feature Selection for Text Categorization on Imbalanced Data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 80-89, 2004.

[19] C. D. Manning, P. Raghavan, and H. Schtze, Introduction to Information Retrieval. Cambridge University Press, 2008.

[20] , J. R. Quinlan. Induction of decision trees. in Machine Learning. 1: 81–106, 1986.

[21] Dunning, Ted. Accurate methods for the statistics of surprise and coincidence. In Computational linguistics 19, no. 1 (1993): 61-74.

[22] K. Kira, L.A. Rendell, The feature selection problem: traditional methods and a new algorithm, In AAAI, vol. 2, 1992a, pp. 129-134.

[23] K. Kira, L.A. Rendell, A practical approach to feature selection, In Proceedings of the Ninth International Workshop on Machine Learning, 1992b, pp. 249-256.

[24] D. D. Lewis. Feature selection and feature extraction for text categorization. In Proceedings of the workshop on Speech and Natural Language, pages 212–217. Association for Computational Linguistics Morristown, NJ, USA, 1992.

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

[26] Kwak, N., Choi, C.-H.: Input feature selection for classification problems. IEEE Transactions on Neural Networks **13**(1), 143-159 (2002)

[27] Ding, C., Peng, H.C.: Minimum Redundancy Feature Selection from Microarray Gene Expression Data. In: Proc. Second IEEE Computational Systems Bioinformatics Conf, pp. 523-528 (2003)

[28] Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1226-1238 (2005)

[29] M. Mandal and A. Mukhopadhyay. An improved minimum redundancy maximum relevance approach for feature selection in gene expression data. In Procedia Technol., vol. 10, pp. 20-27, 2013.

[30] M. Radovic, M. Ghalwash, N. Filipovic, and Z. Obradovic. Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. In BMC Bioinformatics, vol. 18, no. 1, p. 9, 2017.

[31] P. A. Mundra and J. C. Rajapakse. SVM-RFE with MRMR filter for gene selection. In IEEE Trans. NanoBiosci., vol. 9, no. 1, pp. 31-37, Mar. 2010.

[32] Deng, X., Li, Y., Weng, J., Zhang, J.: Feature selection for text classification: A review. Multimed. Tools Appl., 1-20 (2018)

[33] Saleh, S. N., El-Sonbaty, Y.: A feature selection algorithm with redundancy reduction for text classification. In: 22th International Symposium on Computer and Information Sciences, pp.130-135 (2007)

[34] Garbarine, E., DePasquale, J., Gadia, V., Polikar, R., Rosen, G.: Information-theoretic approaches to SVM feature selection for metagenome read classification. Comput. Biol. Chem. **35**, pp. 199-209 (2011)

[35] Agarwal, B., Mittal, N.: Optimal Feature Selection for Sentiment Analysis. International Conference on Intelligent Text Processing and Computational Linguistics. Springer, Berlin, Heidelberg (2013)

[36] Li, Jundong, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. ”Feature selection: A data perspective.” ACM Computing Surveys (CSUR) 50, no. 6 (2017): 1-45.

[37] Javed, K., Babri, H.A., Saeed, M.: Feature selection based on class-dependent densities for high-dimensional binary data. TKDE **24**(3), 465-477 (2012)

[38] Fleuret, F.: Fast binary feature selection with conditional mutual information. The Journal of Machine Learning Research **5**, 1531-1555, (2004)

[39] Holland, J.H.: Adaptation in natural and artificial systems. MIT Press, Cambridge (1992)

[40] Siedlecki, Wojciech, and Jack Sklansky. A note on genetic algorithms for large-scale feature selection. In Handbook of pattern recognition and computer vision, pp. 88-107. 1993.

[41] Punch III, William F., Erik D. Goodman, Min Pei, Lai Chia-Shun, Paul D. Hovland, and Richard J. Enbody. Further Research on Feature Selection and Classification Using Genetic Algorithms. In ICGA, pp. 557-564. 1993.

[42] Yang, Jihoon, and Vasant Honavar. Feature subset selection using a genetic algorithm. In Feature extraction, construction and selection, pp. 117-136. Springer, Boston, MA, 1998.

[43] Li, S., Wu, H., Wan, D. and Zhu, J. An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine. Knowledge-Based Systems, 24(1), pp.40-48, 2011.

[44] Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948, 1995.

[45] Hassan, Rania, Babak Cohanim, Olivier De Weck, and Gerhard Venter. A comparison of particle swarm optimization and the genetic algorithm. In 46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, p. 1897. 2005.

[46] http://rasbt.github.io/mlxtend/user_guide/feature_selection/SequentialFeatureSelector/

[47] Xue, Bing, Mengjie Zhang, and Will N. Browne. “A comprehensive comparison on evolutionary feature selection approaches to classification.” International Journal of Computational Intelligence and Applications 14, no. 02 (2015): 1550008.

[48] Huang, X., Zhang, L., Wang, B., Li, F., Zhang, Z., 2018. Feature clustering based support vector machine recursive feature elimination for gene selection. Appl. Intell. 48 (3), 594–607.

[49] Brezočnik, Lucija, Iztok Fister, and Vili Podgorelec. “Swarm intelligence algorithms for feature selection: a review.” Applied Sciences 8, no. 9 (2018): 1521.

[50] J.Bringer and V.Despiegel. Binary feature vector fingerprint representation from minutiae vicinities. In 4th IEEE International conference on biometrics compendium, (2010)

[51] Vij, Akhil, and Anoop Namboodiri. Learning minutiae neighborhoods: A new binary representation for matching fingerprints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 64-69. 2014.

[52] Krause, Andreas, Leskovec, Jure, Guestrin, Carlos, VanBriesen, Jeanne, and Faloutsos, Christos.: Efficient Sensor Placement Optimization for Securing Large Water Distribution Networks. Journal of Water Resources Planning and Management, 134(6), 516-526.

[53] Krause, A., Singh, A. and Guestrin, C., 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. Journal of Machine Learning Research, 9(Feb), pp.235-284.

[54] Krause, Andreas, and Daniel Golovin. Submodular function maximization. (2014): 71-104.

[55] Krause, Andreas, and Carlos Guestrin. Near-optimal observation selection using submodular functions. In AAAI, vol. 7, pp. 1650-1654. 2007.

[56] Krause, Andreas, and Carlos Guestrin. Submodularity and its applications in optimized information gathering. ACM Transactions on Intelligent Systems and Technology (TIST) 2, no. 4 (2011): 1-20.

[57] M. Shamaiah, S. Banerjee, and H. Vikalo. Greedy sensor selection: Leveraging submodularity. In IEEE Conf. on Decision and Control, pages 2572-2577, 2010.

[58] Jawaid, Syed Talha, and Stephen L. Smith. Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems. Automatica 61 (2015): 282-288.

[59] Shulkind, Gal, Stefanie Jegelka, and Gregory W. Wornell. Sensor array design through submodular optimization. IEEE Transactions on Information Theory 65, no. 1 (2018): 664-675.

[60] Kempe, D., Kleinberg, J. M., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD (2003)

[61] Kempe, David, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In International Colloquium on Automata, Languages, and Programming, pp. 1127-1138. Springer, Berlin, Heidelberg, 2005.

[62] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos,J. VanBriesen, and N. S. Glance. Cost-effective outbreakdetection in networks. In Proceedings of the 13th ACMSIGKDD Conference on Knowledge Discovery and DataMining, pages 420-429, 2007.

[63] Mossel, Elchanan, and Sebastien Roch. On the submodularity of influence in social networks. In Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, pp. 128-134. 2007.

[64] Lin, H., Bilmes, J.: Multi-document summarization via budgeted maximization of submodular functions. In: NAACL-HLT 2010, pp. 912-920 (2010)

[65] Liu, Y., Wei, K., Kirchhoff, K., Song, Y., Bilmes, J.: Submodular feature selection for high-dimensional acoustic score spaces. In: Proc. IEEE Int. Conf. Acoust., Speech Signal Process, pp. 7184-7188 (2013)

[66] Khanna, Rajiv, Ethan Elenberg, Alexandros G. Dimakis, Sahand Negahban, and Joydeep Ghosh. Scalable greedy feature selection via weak submodularity. arXiv preprint arXiv:1703.02723 (2017).

[67] Ghadiri, M., Schmidt, M.: Distributed maximization of submodular plus diversity functions for multi-label feature selection on huge datasets. *arXiv preprint arXiv:1903.08351* (2019)

[68] Graham, R. L., Grötschel, M., Lovász, L.: Handbook of combinatorics **2**, Elsevier (1995)

[69] Buchbinder, Niv, and Moran Feldman. Submodular functions maximization problems. Handbook of Approximation Algorithms and Metaheuristics 1 (2017): 753-788.

[70] Nemhauser, G. L., Wolsey, L. A., Fisher, M. L.: An analysis of approximations for maximizing submodular set functions. I. Mathematical Programming **14**(1), 265-294 (1978)

[71] Kvålseth T.O.: Entropy and Correlation: Some comments. IEEE Trans. on Systems, Man and Cybernetics, SMC **17**(3), 517–519 (1987)

[72] Hall, MA.: Correlation-based feature selection for machine learning. PhD Thesis, University of Waikato, Hamilton (1999)

[73] Gollapudi, S., Sharma, A.: An axiomatic approach for result diversification. In: Proc. WWW (2009)

[74] Vinh, N. X., Epps, J., Bailey, J.: Information theoretic measures for clustering comparison: Variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**, pp. 2837-2854 (2010)

[75] Borodin, A., Lee, H. C., Ye, Y.: Max-sum diversification, monotone submodular functions and dynamic updates. In: Proc. 31st Symp. PODS, Scottsdale, AZ, USA (2012)

[76] Fujishige, S.: Polymatroidal dependence structure of a set of random variables. Information and control, **39**(1), pp.55-72 (1978)

[77] Sen, Prithviraj, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. AI magazine 29, no. 3 (2008): 93-93.

[78] kaggle.com/uciml/sms-spam-collection-dataset

[79] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[80] Zhao, Bin, Prithviraj Sen, and Lise Getoor. Entity and relationship labeling in affiliation networks. In ICML Workshop on Statistical Network Analysis. 2006.

[81] Kurgan, L.A., Cios, K.J., Tadeusiewicz, R., Ogiela, M., and Goodenday, L.S. Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis. Artificial Intelligence in Medicine, vol. 23:2 (2001), pp 149-169.

[82] http://www.causality.inf.ethz.ch//activelearning.php?page=datasets#cont

[83] LeCun, Y., Bengio, Y.,and Hinton, G. E. (2015). Deep learning. Nature 521, 436–444. doi: 10.1038/nature14539

[84] Kotsiantis, Sotiris B., Ioannis D. Zaharakis, and Panayiotis E. Pintelas. Machine learning: a review of classification and combining techniques. In Artificial Intelligence Review 26, no. 3 (2006): 159-190.

[85] Kumar, Akshi, and Arunima Jaiswal. "Empirical study of twitter and tumblr for sentiment analysis using soft computing techniques." In Proceedings of the world congress on engineering and computer science, vol. 1, pp. 1-5. 2017.

[86] Pasupa, Kitsuchart, and Wisuwat Sunhem. "A comparison between shallow and deep architecture classifiers on small dataset." In 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 1-6. IEEE, 2016.

[87] Caruana, Rich, and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning, pp. 161-168. 2006.

[88] K. Crammer and Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines. In Journal of Machine Learning Research, vol. 2, pp. 265-292, 2002.

[89] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, Liblinear: A Library for Large Linear Classification. In J. Machine Learning Research, vol. 9, pp. 1871-1874, 2008.

[90] https://scikit-learn.org/stable/modules/tree.html