University of Cincinnati

Date: 11/7/2017

I, Abhro Jyoti Mondal, hereby submit this original work as part of the requirements for the degree of Master of Science in Computer Science.

It is entitled:

Document Classification using Characteristic Signatures

Student's name: **Abhro Jyoti Mondal**

This work and its defense approved by:

Committee chair: Raj Bhatnagar, Ph.D.

Committee member: Ali Minai, Ph.D.

Committee member: Shomir Wilson, Ph.D.

UNIVERSITY OF Cincinnati

29118

Document Classification Using Characteristic Signatures

A thesis submitted to the

Graduate School of the

University of Cincinnati

in partial fulfillment of the requirements

for the degree of

Master of Science

in the **Department of Electrical Engineering & Computing Systems**

of the **College of Engineering & Applied Sciences**

By

Abhro Jyoti Mondal

B.Tech. Information Technology, West Bengal University of Technology, India, 2011

November 2017

Thesis Advisor & Committee Chair:

Dr. Raj Bhatnagar

Department of Electrical Engineering & Computer Science
College of Engineering and Applied Sciences

Abstract

Supervised document classification technique, proposes a model that is trained with a fixed number of labeled classes and can be effectively used to classify test documents under one of the labels in the training set. The major objective of our research was to identify text documents from labels or topics which are not present in the training set, yet appeared in the test set. We devised a method to identify and eliminate documents from such labels or topics that do not occur in the training set. This technique brings together the idea of template matching and document classification by creating characteristic signatures that are unique to each label in the training set. Using these signatures any unknown label could be detected and ignored in the test data set. Our results clearly show that, the proposed approach is useful in classification of documents into known categories or labels, as well as identifying labels that do not match with the predefined labels in training set.

Acknowledgements

I would like to extend my sincere gratitude to my thesis advisor and committee chair Dr. Raj Bhatnagar. He consistently steered me in the right direction whenever he thought I needed it or I had questions about my research. I would also take this opportunity to thank Dr. Ali Minai and Dr. Shomir Wilson for taking time and interest to serve on my defense committee. I express my profound gratitude to my parents and my brother for their constant encouragement and support. My joy has no bounds in expressing my appreciation to all my friends who were always there to encourage and motivate me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Document classification has been a field of research and study in the past in form of library science, information retrieval and text organization [1][2][3]. This upsurge is the result of the increasing trend in digital media in every form all around the world, which requires efficient techniques for documents to be stored, sorted, indexed, categorized and often made accessible to a large number of users. Some of the earliest document classification techniques can be traced back to 1968 which uses information retrieval techniques to find similar documents [4][5]. One of the approaches is to build a classifier that learns from a set of pre-classified document collection and this offers many advantages over manual classification which includes speed, scalability, savings in terms of labor and time [6][7]. Content based document classification can be implemented in numerous real life applications or problems, such as, spam filtering, email routing, sentiment analysis, recommendation system and genre classification [7]. The goal of this research is to study the efficiency of signature based document retrieval and comparing the performance of different ways of creating documentclass signatures. Text documents are collection of concepts or thoughts based on one or more topics, themes or subjects. Moreover, every collection or datasets of text documents could be categorized in to

a specific number of classes [8][9]. These classes are unique and the documents in each class would share certain characteristics that places them in a specific class. Hence, these documents can be classified according to their features or attributes which could be in the form of words, subjects, authors, publishers, year of publication [10][11]. The key aim of this work is to use words as features in a collection of documents and create a unique signature for each class of documents. We would further investigate different techniques of making these signatures which are able to classify the class to which a document belongs. There are several information retrieval techniques which are developed over the past few decades to classify documents into specific classes [11][12][13]. However, the general idea derived from these techniques is to train a model from a set of training set and thereafter use the model to classify a collection of test data-set. Interestingly, if an unknown class is introduced in the dataset these models fail to identify the foreign class because its an undefined element to the classifiers. This is where the idea of using characteristic signatures comes into play. Since signatures are unique to classes, they can be trained to identify any unknown classes in the test dataset that has different properties or features from the classes it in the training set. We will discuss in the later chapters how the performance of signatures tends to become better than other classification techniques when an unknown class is introduced in the test dataset. Document classification approaches can be arguably categorized into two different techniques, these are discrimination analysis and characteristic signature analysis [12][14].

1.2 Discriminative Models and Generative Models

Although in practice discriminative models and generative models are already established as two different approaches [15]. In discriminative approach we essentially try to model the boundary between the classes. The idea behind discrimination analysis is to distinguish two or more classes using their differentiating

features. The differences between two or more classes are identified based on their relative differences. In case of Support Vector Machines (SVM), it uses a subset of the training data called support vectors to build the decision function in order to classify various classes [14]. SVM are supervised learning algorithm that classifies data based on their differences. For a given set of pre-defined classes, this technique creates a classifier that is used to classify the documents into one of the predefined classes. Generative models on the other hand attempts to model the distribution of the classes. Discriminative models usually uses conditional probability to model the boundary between the classes, while in a generative model we use a joint probability distribution which ideally models the real distribution of the classes.

1.3 Characteristic Signature

A unique signature for each group is generated based on their isolating and recognizing features. Based on these distinct signatures one or more new observations are classified in to one of the known categories or classes. Since these signatures are unique to the pre-defined classes, this helps us to classify documents from the pre-defined classes even when unknown classes are introduced in to the corpus. Discrimination analysis for text documents are widely used in the research field. In order to realize the advantages of characteristic signature analysis we will take help of the following example. Figure 1.1(a) depicts a plot of documents in a dataset containing two classes based using two feature sets. The red triangles represent documents from one class and the green circles represent documents from another class. We can build a classifier using one of the many popular discrimination analysis techniques to classify the two classes. In case of SVM, it will build a hyperplane that would separate the two classes as shown in the Figure 1.1(b). On the other hand, characteristic signatures can be thought of as descriptive or representative illustration of classes. In general, when we use a clustering approach, the clusters are considered to be a prototype of the classes. Similarly, a signature

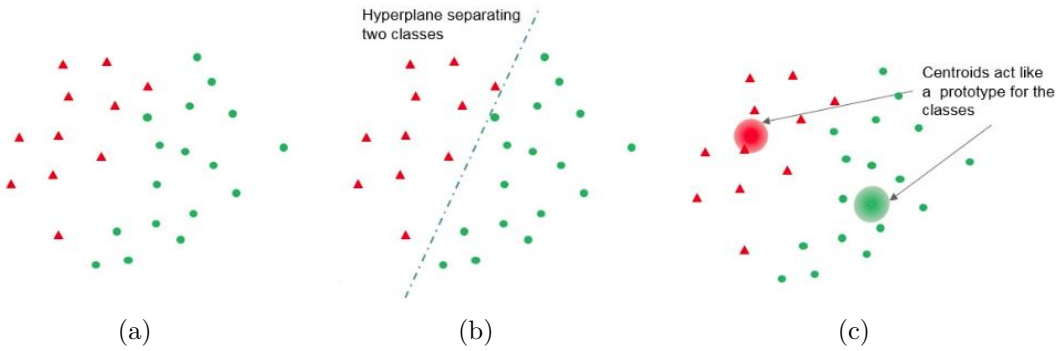

FIGURE 1.1: (a) Plot of data points from two classes (b) Classification of classes using discrimination analysis(eg. SVM) (c) Classification of classes using characteristic analysis

for a particular class can be visualized as the centroid of the class as shown in Figure 1.1(c).

Therefore, in case of documents the centroid is formed using the information obtained from the frequency of the words. The number of dimension is defined by the number of features or words used for classification. The entire set of words is not used for classification because we want to use a subspace of the set of words or features which would result in a more meaningful classifier. So far in the discussion, both the approaches are successful to classify the two classes. However, if we introduce an unknown class in the dataset and use the current classifier trained on two classes, our classifiers are going to perform poorly. Discrimination analysis does not take in to account the information contained in all the datapoints. In case of SVM, the hyperplane is built using a selected few datapoints called the support vectors. Thus, on adding an unkown class, SVM would be unable to correctly classify the unknown class. On the other hand, this can be easily countered using a signature based approach with a little modification. We need to feed a threshold to the classifier based on characteristic signatures, that will mark a boundary for each classes. Any data point on or within the boundary will be classified as a member of that class. Figure 1.2 illustrates the idea that we incorporate to differentiate the existing old classes from the new unknown class. The black square represents the unknown classes introduced in the data set.

We can observe that this approach will exclude any datapoint that does not satisfy

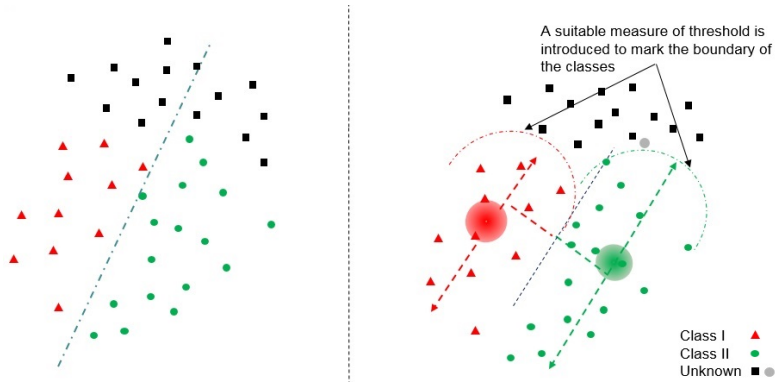

FIGURE 1.2: Introduction of unknown class and classifying the three classes using classifier trained on two classes. Misclassification using a discrimination analysis is depicted on the left pane and classification using characteristic signature is depicted on the right pane.

the threshold criteria (the grey circle shown in the right pane of FIgure 1.2. We will discuss in details regarding the implementation of the threshold and the results in later sections. The following section addresses the advantages and disadvantages of signatures based approach.

1.4 Advantages and Disadvantages of Signature Based Retrieval

Signature based information retrieval has been used in the past to search databases of DNA molecules [16], searching through databases of annotated images [17]. The idea of using signatures serves as a filter that provides a quick way to discard non-qualifying documents. Signature based document classification can be easily implemented for large number of classes. Signature based document retrieval also offers flexibility over the length of the signatures chosen for classification which would directly impact the time taken to create the classifier. In this study we have explored different ways of creating signatures which again creates an opportunity to pick the suitable class-signatures in order to classify documents with diverse context and background.

1.5 Outline of Signature Construction Method

Text documents belonging to various classes are fundamentally considered to have characteristics attributes, which are in a way unique to the contents of the document. This assumption is the incentive to build models based on characteristic attributes which are hidden in the text. Therefore, a characteristics signature of a class or topic of documents are assumed to share similarities among themselves. The main challenge lies in carefully selecting the attributes of a document to create a characteristic signature and to be able to efficiently extract useful information from the text. Essentially a characteristic signature could be made as long as needed. Hence, we need to experimentally find out an optimal length for the signatures. For the purpose of our experiment we used text datasets ranging across different topics or classes. The goal is to build characteristic signatures based on a training set and subsequently use the signatures to classify a fresh set of documents and measure performance metrics of the model. Fundamentally the characteristic signatures are expressed based on three properties of the features of a text document namely, frequency, entropy and z-score of the words in the document. Assuming that words are the most significant features in a document, we intend to create a characteristic signature based on those three properties for a selected set the words. To study the performance of signatures and their ability to classify documents in to various classes, we choose to vary the length of the signatures from 10 words through 140 words in varying steps.

Signatures based on Frequency: After preprocessing of a collection of documents every word in the collection is considered to be a part of the feature set of the collection. Based on the information obtained from absolute frequency, relative frequency and relative rest (which is discussed later) of each word across the collection we could essentially make a signature based on frequency of those properties of the words.

Signature based on Min Entropy: Continuing on similar lines, we can use absolute frequency, relative frequency of the words to calculate the entropy for each

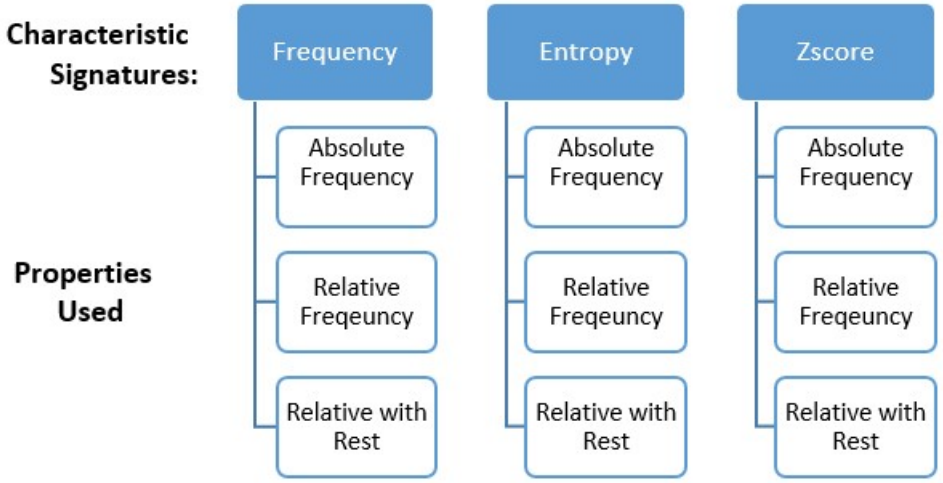

FIGURE 1.3: Outline of various modes of signature and features of words used

word and utilize that information to build signatures for each class which are fundamentally based on entropy.

Signatures based on Z-score: Similarly, we can use absolute frequencies, relative frequencies of the words and build signatures for each class based on the z-scores of each of these properties of the words.

1.6 Overview of Chapters

The rest of thesis is organized as described in the following section: Chapter 2 gives a synopsis of some the prominent algorithms used in supervised and unsupervised document classification and their advantages and disadvantages. It also explains the applications of template based classification in Optical Character Recognition and image processing. In our work we have devised a method that uses template based classification on document collections. Chapter 3 or signature generating algorithms section is focused on describing the steps to generate signatures based on various properties of the features or words of the document. These signatures are essentially templates which are matched with test data. This is further described in Chapter 4 which discusses experimental setup and results. Describes the experiments and datasets used to conduct the study on the signatures created using the methods described in the previous chapter. Finally, chapter 5 concludes

the work with and future work by summarizing the discussion, observations and insights obtained from the experiments described in the previous chapter. This chapter also outlines a possibility of future work.

Chapter 2

Literature Survey

Document classification has been an area of intensive research and various supervised and unsupervised learning algorithms have revealed significant performances in document classification. This section explores some of the popular document classification techniques.

2.1 Text Categorization with Support Vector Machines

Support Vector Machines (SVM) are supervised learning algorithms that can be used for document classification. SVM in their basic form can be used to learn linear threshold function. Each data point is viewed as p-dimensional vector and such data points are separated using a (p-1) dimensional hyperplane. In case of training set with separable two classes, as shown in Figure 2.1, there are lots of possible linear separators.

While some learning algorithms like perceptron, finds any linear separator between the data points. SVM specifically learns a decision surface that is maximally position from all the data points. The distance between the decision surface and the closest data points determines the margin of separation for the classifier. This

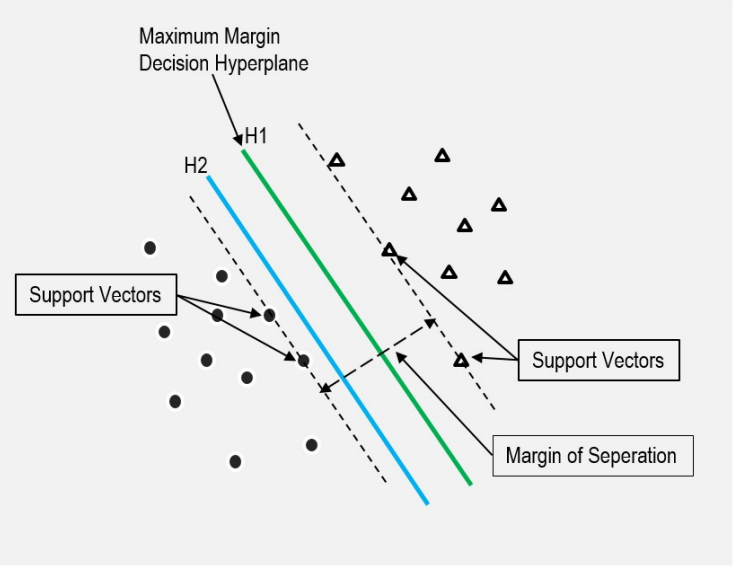

FIGURE 2.1: The figure depicts two hyperplanes separating two classes. H1 has larger margin of separation and H2 has smaller margin of separation. Support Vectors are the four points on the class boundary.

approach of constructing the decision boundary is influenced by the position of a subset of data points which lies in close neighborhood of the separator. These points are known as support vectors (in any n-dimensional space they can be thought of as n-dimensional vectors). SVM can be used to learn polynomial classifiers with the help of an appropriate kernel function.

SVMs are promising to learn text classifiers because of they can handle high dimensional input space. While handling text documents, one has to deal with high number of features. Since, SVM uses a subset of the training set, they have the potential to handle large feature spaces. This can be visualized as a way to avoid high dimensional input space by assuming the relevant features. Feature selection focuses on excluding the irrelevant features. Document vectors are essentially sparse which means for each document only few entries for the features are non-zero [17]. SVM are generally well suited for problems with sparse instances. The other factors that makes SVM a popular tool in document classification problem is that fact that the problem is linearly separable. The main intension is to find a linear separator using SVM. This approach is one of the examples of discrimination analysis and suffers from inability to adapt to random changes made by introducing new classes. The classifier needs to be re-learned in order to function

effectively and classify documents. SVM doesnt take in to account all the features in a document and hence this approach of feature selection often leads to loss of information [18]. However, using characteristic signatures could overcome these issues and varying the length of the signature would help us to take in to account large number of features and thus retaining more information regarding the specific class.

2.2 Latent Dirichlet Allocation (LDA)

LDA is a topic model that represents documents as a mixture of topics and outputs words with certain probabilities. In recent time LDA have grown in popularity and many applications have been proposed. LDA follows a distribution over distributions approach, where topic mixture proportions are drawn from a Dirichlet distribution. A document in LDA is a mixture of small number of topics and a topic is a distribution over a fixed set of vocabulary [19]. LDA assumes that documents are produced in the following manner:

1. Fix the number of words in the vocabulary
2. Select a topic mixture for the document from Dirichlet distribution over a fixed set of K topics
3. Generate each word in the document by
   (a) Select a topic from the multinomial distribution sampled above
   (b) Choose a word from the word distribution of the chosen topic

Figure 2.2 represents a probabilistic graphical model of LDA. This graphical model is sub divided in to three levels. The hyperparameters α and β are known as corpus level parameters. The variable θ is a document level variables and variables z and w are word level variables. The hyperparameters are assumed to be sampled once in document generation and variable θ is sampled per document. The variables z and w are sampled for each word in the document. LDA assumes that each θ is chosen from a Dirichlet distribution defined over the topic simplex and parametrized by

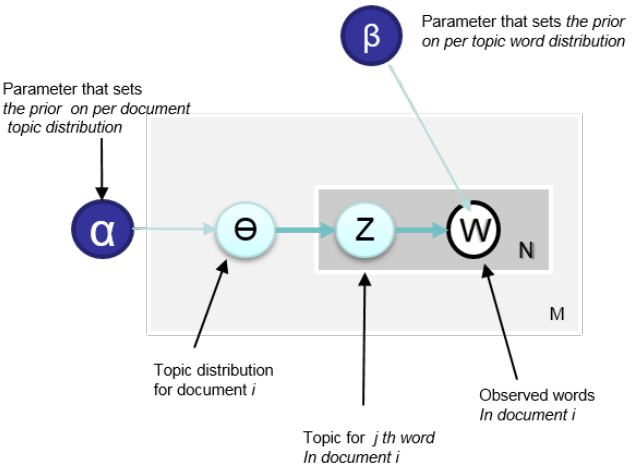

FIGURE 2.2: Plate diagram of LDA

a hyperparameter α, and words are chosen from a Dirichlet distribution which is parametrized by a hyperparameter β. Below is the algorithm that shows how a document is generated:

Algorithm 1 Generative Process

1: **for** document d_d in corpus D **do**
2: Choose $\theta_d \sim (\alpha)$
3: **for** position w in d_d **do**
4: Choose a topic $z_w \sim (\theta_d)$
5: Choose a word w_w from $p(w_w|z_w, \beta)$, a multinomial distribution over words conditioned on the topic and the prior β.
6: **end for**
7: **end for**

Here, $p(z_d, n|\theta)$ is the probability that topic z is drawn for the nth word token in document d, given the topic distribution, θ for document d and $p(w_d, n|Dir)$ is the probability of word w drawn from the nth word token in d assuming topic distribution $z_{d,n}$. A high value of α means a document d contains a large number of topics. Similarly, a high value of means a topic z contains large number of words. The output of the algorithm is a model with pre-defined number of topics and distribution of words for each topic. All the above analysis considers the occurrence of word patterns and derives an insight based on that. For short documents which are insufficient to reflect the true distribution of the words, this could cause poor topic assignments. It is used for unlabeled document collection and hence the number of topics to be generated has to be predefined.

2.3 Template Matching

Template matching is a well-known technique in the field of image processing and computer vision with a wide range of applications [20]. The aim is to match the template to an image and to check if the template lies inside the image. A template could essentially be a character, pattern or a simple object. The technique is widely accepted and used in detecting words from document images and also to recognize characters in OCRs. This idea is used to detect edges in images which could be used to navigate autonomous robots and in quality control in manufacturing works. Vijayarani et al. [21] had implemented template matching method to search words in an image of a document by the help of comparison using various performance index. Other studies have shown template matching algorithm has been used to develop prototype for Optical Character Recognition (OCR) [20]. In general, a OCR method involving template matching algorithm would take the following course of action as shown in the flowchart below. Various template matching techniques using different metrics have shown that OCR can be performed with decent accuracy [21][22]. Some of the evolutionary template matching techniques are cross-correlation, sum of absolute differences and sum of squared difference [23]. Nevertheless, these techniques are search exhaustive and computationally expensive, several newer techniques are also introduced to speed up the search and matching of templates [23]. To summarize the generic template matching algorithm, we can describe it as finding a match between the sub-image(template) and the region in the original image on which the sub-image in coincidental. Therefore, the key motivating aspect for template matching is to device an effective similarity measurement and an appropriate search strategy.

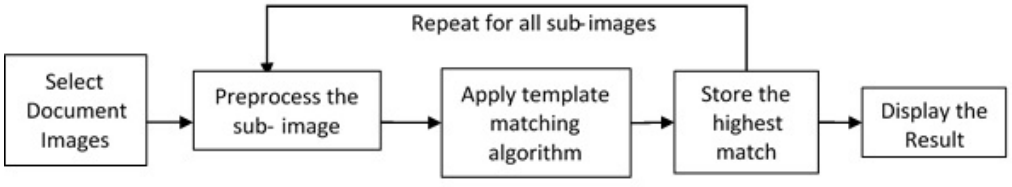

FIGURE 2.3: Flowchart depicting generic flow of Optical Character Recognition using template matching.

Chapter 3

Signature Generating Algorithms

3.1 Terminology

3.1.1 Bag of Words

This is a representation of documents which is very prominent in information retrieval and natural language processing. In this representation a text or sentence is represented a set of words without considering the grammar or the order of the words. This approach has been incorporated in other fields like computer vision [24]. This method is widely used in document classification problems where the words are used as features and helps to build the classifiers.

3.1.2 Document Term Matrix

A document-term matrix is a mathematical representation of a collection of documents in the form of a matrix. The rows correspond to documents and the columns correspond to features or words in the vocabulary. This kind of matrices are usually very sparse in nature. There are various schemes that can be utilized to determine the value in each cell of the matrix. Such examples schemes are tf-idf score, absolute frequency, and relative frequency of the respective features.

3.1.3 Term Frequency Inverse Document Frequency (tf-idf)

Term frequency inverse document frequency is a numeral statistic that helps us to determine the importance of a word to a document in a collection [25]. Tf-df is an important weighing factor in the field of document classification. The tf-idf value increase proportionally to the frequency of the word in the document, however it is offset by the occurrence of the word in the corpus or collection. This helps to counter the fact that some words occur more frequently in a document and yet it does not carry enough weight. Tf-idf score composed of two components: the first component is the term frequency(tf) and inverse document frequency(idf). TF is computed as the number of times a word has occurred divided by the total number of words in the document. The chances of a word to occur in longer documents is higher than occurring in shorter documents with fewer words. Hence, tf scores are divided by the length of the document in order to normalize the frequency. IDF scores are computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

$$tfidf = f_{t,d} \cdot log\frac{N}{n_t}$$

where, $f_{t,d}$ is the term frequency adjusted by the document length $log\frac{N}{n_t}$ is the inverse document frequency.

3.1.4 Correlation

In probability and statistics, correlation often called correlation coefficient which represents the similarity and trend of a linear relationship between random variables. There are number of correlation coefficient schemes which are used for different situation.

3.1.5 Stemming

In information retrieval stemming is the process of reducing the word to its root form. In case of information retrieval stem words need not be a valid root word as it is defined in that specific language. It is considered sufficient if all the words with the same root, indicate to the same stem word. There are many resources available online which offers stemmers that could be used for the purpose of document classification.

3.2 Creating Signatures

Before we are ready to use a text collection we need to run few standard text preprocessing techniques which involve removing stop words, stemming etc [26]. There are plenty of standard tools available as online resources which can used to perform text pre-processing. We need to learn the vocabulary from all the documents in the collection and create a document-term matrix. This could be achieved using the bag of words approach. We assign each word occurring in any of the documents in the collection a positive integer by creating a dictionary. This dictionary maps each word to their respective indices. Now for each document in the training set we gather the count of each word occurring in the document. Therefore, the document-term matrix would be a NxM matrix where N is the total number of documents and M is the total number of words in the collection as shown in the matrix below. It should be noted that we have used words, features and terms interchangeably in this section.

$$\text{Mat}_{\text{N}xM} = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n{-}1 \\ d_n \end{array} \begin{array}{c} \begin{array}{cccccc} f_1 & f_2 & f_3 & \cdots & f_m{-}1 & f_m \end{array} \\ \left(\begin{array}{cccccc} 1 & 1 & 1 & & 1 & 1 \\ 0 & 1 & 0 & & 0 & 1 \\ 0 & 0 & 1 & & 0 & 1 \\ \vdots & & & \ddots & & \\ 0 & 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & 0 & & 0 & 1 \end{array} \right) \end{array}$$

The document-term matrix is essentially a large sparse matrix which gives the count or occurrences of each word in the collection. The columns are the features and the rows are the documents. Therefore, the matrix depicts N rows and M columns to represent the N documents in the collection and the entire vocabulary of M words. Occurrences can be assumed to be a good metric to create signatures. Table 3.1 depicts a document-term matrix which has n documents and m features. Its not feasible to represent a document term matrix of a real text dataset due to its sheer dimension. In order to explain how feature subset selection is done in the first step lets consider Table 3.1. The document term matrix has m columns represent all the initial m features generated from the training set and 9 rows, each representing a document in the collection. The last row in Table 3.1 shows the sum of the occurrences of the features in the entire collection. We observed through repeated experiments that if a word occurs too less or too high in the document collection, it is adding noise to the model. Hence, we decided to select a subset of the entire feature set available by introducing a lower bound and an upper bound on the occurrences of a word in order to pick only useful words to build signatures. The lower bound for occurrence of a word was set to 8 and the upper bound was set at 1000. Therefore, all the features with occurrence of 8 and lesser or 1000 and more has to be eliminated. This would eliminate features like f_2 and perhaps many more features which could not be shown in Table 3.1. Following this screening of feature set, we are left with a shrunken document-term matrix which should have equal or lesser number of columns depending on the

Doc \ Feature	f_1	f_2	...	f_{k-2}	f_{k-1}	f_k	...	f_{m-1}	f_m
d1	2	0	...	0	0	1	...	67	0
d2	8	0		0	0	0		13	1
d3	0	0		0	0	1		42	0
d4	0	0		1	7	0		0	1
d5	0	0		2	5	0		2	0
d6	2	1		1	0	0		1	0
. . .	. . .			. . .				. . .	
d_{n-1}	0	0		18	1	0		0	1
d_n	0	0		50	0	48		1	13
Total Frequency	12	4		100	13	50		126	16

TABLE 3.1: Sample document-term matrix depicting documents from three classes.

...

number of features eliminated. This document-term matrix should serve as the basis for creating signatures.

3.3 Signatures Using Absolute Frequency of Words

The previous section showed us how to generate the document-term matrix based on feature subset selection. In sections 3.3, 3.4 and 3.5 we will walk through the steps used to create the characteristic signatures based on various properties of the features or words. For the sake of easiness and simplicity we will take help of a hypothetical document-term matrix which has nine rows denoting nine documents in the collection and ten column denoting ten features, after performing suitable feature subset selection. The document-term matrix shown in Table 3.2 will be used in the following sections to demonstrate the creation of characteristic signatures. The nine documents in the document-term matrix essentially belong to a hypothetical training set which contains documents from three different classes or topics. These documents representing each row are sorted class-wise, i.e. documents d_1, d_2, d_3 belong to class I, d_4, d_5, d_6 belong to class II and d_7, d_8, d_9 belong

Doc \ Feature	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
d_1	1	1	3	0	0	1	0	0	0	0
d_2	1	0	1	0	0	0	0	0	1	0
d_3	0	0	2	0	1	0	0	0	0	1
d_4	1	0	0	1	3	0	2	0	1	2
d_5	0	0	1	2	2	0	0	1	0	0
d_6	0	1	0	1	4	0	1	1	0	3
d_7	0	4	0	0	0	0	0	0	0	1
d_8	1	3	1	0	0	0	0	0	0	0
d_9	0	2	1	0	1	0	0	0	1	0

TABLE 3.2: Sample document term matrix depicting absolute frequencies of features/words.

to class III. In section 3.3 our aim is to demonstrate generating characteristic signatures based on the absolute frequency of the features or words. There are three kinds of signatures that would be generated using the absolute frequencies of the words in the document collection. The first kind of signature would be using the frequencies of the absolute frequency of words, secondly using the entropy of the absolute frequency of the words in the collection and lastly, using z-score as of the absolute value of the words in the corpus. We will start with describing the first case mentioned above which is creation of signature using frequencies of the absolute frequency of words or features.

In section 3.3 our aim is to demonstrate generating characteristic signatures based on the absolute frequency of the features or words. There are three kinds of signatures that would be generated using the absolute frequencies of the words in the document collection. The first kind of signature would be using the frequencies of the absolute frequency of words, secondly using the entropy of the absolute frequency of the words in the collection and lastly, using z-score as of the absolute value of the words in the corpus. We will start with describing the first case mentioned above which is creation of signature using frequencies of the absolute frequency of words or features.

Table 3.3 represents the same document-term matrix depicted in Table 3.2 along with some more intermediate results. In Table 3.3 we have introduced a row after

Doc \ Feature	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
d_1	1	1	3	0	0	1	0	0	0	0
d_2	1	0	1	0	0	0	0	0	1	0
d_3	0	0	2	0	1	0	0	0	0	1
ClassI	2	1	3	0	1	1	0	0	1	1
d_4	1	0	0	1	3	0	2	0	1	2
d_5	0	0	1	2	2	0	0	1	0	0
d_6	0	1	0	1	4	0	1	1	0	3
ClassII	1	1	1	3	3	0	2	2	1	2
d_7	0	4	0	0	0	0	0	0	0	1
d_8	1	3	1	0	0	0	0	0	0	0
d_9	0	2	1	0	1	0	0	0	1	0
ClassIII	1	3	2	0	1	0	0	0	1	1

TABLE 3.3: Sample document term matrix depicting occurrences of feature.

each class denoting the number of times each feature showed up in a class. In other words, we are keeping a count of how many documents the feature appeared in and not the number of occurrences of the feature. Hence, feature f_1 appeared in two documents in class I and similarly feature f_3 appeared in all the three documents in class I and in two documents in class III. Based on this information we are going to perform another level of feature subset selection. In order for a feature or word to further qualify for creating a signature, it must occur in more than at least k documents in ANY one of the classes. The value of k behaves as a threshold and helps in determining which features could participate in creating characteristic signatures. For a real text dataset, we would experimentally determine a suitable value for threshold k. But for the example at hand, lets set the threshold k at a value of 2. This means that for a feature to be able to qualify for forming characteristic signature, it must occur in 2 documents in any of the three classes at hand. In Table 3.3 the features f_6 and f_9 do not appear in any of the three classes for twice or more. While, among the rest of the features $f_1, f_2, f_4, f_5, f_7, f_8$ and f_{10} appear in at least two or more documents in one of the classes. Feature f_3 appears in twice or more documents in class I and II. This would further reduce the feature set since we have to eliminate features f_6 and f_9 for not satisfying the threshold k. Now, for all the features that satisfy the threshold k, we calculate the absolute frequency for all the three classes and sort them in descending order

Features	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
Absolute Frequency	4	11	9	4	11	-	3	2	-	7

(a)

Features	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
Absolute Frequency	4	11	9	4	11	-	3	2	-	7

(b)

TABLE 3.4: Sum of frequencies for all three classes for features satisfying threshold k . (a) Unsorted (b) Sorted in descending order of absolute frequency of features.

as shown in Table 3.4.

Table 3.4(b) shows the features which are sorted based on potentially most influencing to less influencing characteristics based on frequency of the words. Based on the desired length of the signature we choose the foremost features from the sorted list of features. It should be noted that since we sort the features in this step based on decreasing frequency and choose features in decreasing order of frequency, we refer this characteristic signature to be based on frequency of absolute frequency of the words. Let us assume, we are going to create three signatures of length 4, 6 and 8 words respectively. In order to create a signature of length n words, we will consider the first n words from Table 3.4(b). Hence, the characteristic signature (of length 4 words) using absolute frequency will include features f_2, f_5, f_3 and f_{10}. Similarly, characteristic signatures of length 6 words and 8 words would include features $f_2, f_5, f_3, f_{10}, f_1, f_4$ and $f_2, f_5, f_3, f_{10}, f_1, f_4, f_7, f_8$ respectively. So far, we have determined the features that are needed to create the characteristic signatures using absolute frequency for this hypothetical data-set. With the features necessary to generate the signatures being identified, our next step is to actually create the characteristic signatures from these features that will uniquely represent each class.

In order to create the signatures that are unique to the classes we need to evaluate the absolute frequency of each feature for a particular class. Therefore, absolute frequencies for all the features considering their occurrences in class I is denoted

Doc \ Feature	f_2	f_5	f_3	f_{10}	f_1	f_4	f_7	f_8
d_1	1	0	3	0	1	0	0	0
d_2	0	0	1	0	1	0	0	0
d_3	0	1	2	1	0	0	0	0
AF I	1	1	6	1	2	0	0	0
d_4	0	3	0	2	1	1	2	0
d_5	0	2	1	0	0	2	0	1
d_6	1	4	0	3	0	1	1	1
AF II	1	9	1	5	1	4	3	2
d_7	4	0	0	1	0	0	0	0
d_8	3	0	1	0	1	0	0	0
d_9	2	1	1	0	0	0	0	0
AF III	9	1	2	1	1	0	0	0

(a)

Class	Signature length (words)	f_2	f_5	f_3	f_{10}	f_1	f_4	f_7	f_8
I	4	1	1	6	1				
	6	1	1	6	1	2	0		
	8	1	1	6	1	2	0	0	0
II	4	1	9	1	5				
	6	1	9	1	5	1	4		
	8	1	9	1	5	1	4	3	2
III	4	9	1	2	1				
	6	9	1	2	1	1	0		
	8	9	1	2	1	1	0	0	0

(b)

TABLE 3.5: (a)Represents absolute frequencies for features that qualified for building characteristic signatures. The features are shown in sorted order of their relevance. (b) Characteristic signatures of class I, II and III with signatures of length 4,6 and 8 words.

by row AF-I (shown in Table 3.5(a)). Similarly, absolute frequencies for all the features taking in to account the occurrences in class II and class III are represented in row AF-II and AF-III respectively. As depicted in Table 3.5(b), the characteristic signature of length 4 words, unique to class I, would be represented by the first four elements in row AF-I. Therefore, on increasing the length of the signature we will select the corresponding values of the features in rows AF-I, AF-II and AF-III for each class. It should be noted that in this example the longest length of signature possible with the given threshold k is eight words. Table 3.5(b) shows

the characteristic signatures based on frequency using absolute frequencies of the features for all the three classes and with length of signature from 4 words to 8 words incremented in steps of 2. In the previous section we built signatures based on the frequencies of the absolute frequency of the words. We would now use the minimum entropy of the absolute frequency of the words in the corpus to build the signatures. Words for the signatures are selected from the document-term matrix based on the criteria that the word must appear in at least k (threshold) documents in ANY one class and must appear q times in the rest of the classes (where k and q are positive integers and $kgtq$). Entropy is the measure of randomness of a random variable. It can also be expressed as a measure of the amount of information a random variable contains. Entropy of a word or term for three classes can be defined as follows:

$$[Entropy]_f = -[p(a)log(p(a)) + p(b)log(p(b)) + p(c)log(p(c))] \tag{3.1}$$

where $p(a), p(b)$ and $p(c)$ are the different metric values the word w takes for each of the three classes. It should be noted that the threshold q introduced for selecting features makes sure that none of the three components for each class in the above equation is zero. Unlike the last method demonstrated to create signatures, we will use the entropy value for each word instead of their frequencies in the document-term matrix. Based on entropy value of the selected words we arrange them in increasing order and the top n words qualify to be the signature for the data collection. We will take help of a similar set up to illustrate how the signatures are made based on entropy using absolute frequency of the words. Lets consider the hypothetical dataset depicted in the document-term matrix shown in Table 3.2. Let us assume that k is 2 and q is 1, in other words for a feature to qualify to participate in characteristic signature formation using minimum entropy, it must occur at least twice in one of the three classes and must appear at least once in the other two classes. Table 3.6(a) shows the occurrences of the features across the three classes. Based on the threshold values set for k and q, the set features that qualify to participate in creating the signatures are f_1, f_2, f_3, f_5 and

f_{10}. Features f_4, f_6, f_7 and f_8 do not meet the condition of appearing at least q (1 in this example) times in all three classes. Feature f_9 satisfies the criteria with threshold q but does not satisfy the condition for threshold k, which requires the feature to appear at least k times in any one of the classes. Table 3.6(b) shows the maximum and minimum appearances among all the three classes. Therefore, the maximum appearance should be k or higher and the minimum Occurrence should at least be q times in order to be selected for creating signatures. The features that do not satisfy the thresholds are struck out in Table 3.6(b). The next step includes calculating the entropy of each feature using the entropy formula discussed earlier and the results are shown in Table 3.6(c).

Doc \ Feature	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
d_1	1	1	3	0	0	1	0	0	0	0
d_2	1	0	1	0	0	0	0	0	1	0
d_3	0	0	2	0	1	0	0	0	0	1
ClassI	2	1	3	0	1	1	0	0	1	1
d_4	1	0	0	1	3	0	2	0	1	2
d_5	0	0	1	2	2	0	0	1	0	0
d_6	0	1	0	1	4	0	1	1	0	3
ClassII	1	1	1	3	3	0	2	2	1	2
d_7	0	4	0	0	0	0	0	0	0	1
d_8	1	3	1	0	0	0	0	0	0	0
d_9	0	2	1	0	1	0	0	0	1	0
ClassIII	1	3	2	0	1	0	0	0	1	1

(a)

Feature	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
Occurrences [max,min]	[2,1]	[3,1]	[3,1]	[3,0]	[3,1]	[1,0]	[2,0]	[2,0]	[1,1]	[2,1]

(b)

Feature	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
Entropy	0.45	0.26	0.37	-	0.26	-	-	-	-	0.35

(c)

TABLE 3.6: (a) Sample document term matrix depicting absolute frequencies of features/words.
(b) Maximum and minimum appearances of the features among the three classes.
(c) Entropy calculation for features that satisfy threshold k and q.

The features are then sorted in ascending order of their entropy as shown in Table 3.7(a). We are going to select the features for the characteristic signatures from this sorted list. We will illustrate creating characteristic signature for length n based on minimum entropy using absolute frequency of the words. As shown in Table 3.7(b) we have consolidated the absolute frequencies for all the features that qualify for creating signatures in the previous steps. Rows AF-1, AF-II and AF-III denotes the sum of the absolute frequencies for the qualifying features for classes I, II and III respectively. As a part of this example we cannot create a characteristic signature based on minimum entropy using absolute frequency of words which can be of length greater than 5 words. Therefore, if we choose to create signatures unique to each class of length 3, 4 and 5 words, we should choose the cumulative absolute frequencies of the corresponding features for class I, II and III. Table 3.7(c) illustrates the characteristic signatures obtained for each class based on minimum entropy using absolute frequency of features. Therefore, a signature of length 3 words for class I, II and III should be [6,1,1], [1,1,9], [2,9,1] using features f_3, f_2, f_5 respectively. These signatures generated are unique to the characteristics of each classes.

Until now, we have discussed how to create characteristic signatures based on frequency and minimum entropy using absolute frequencies of features. In the following scenario, we will illustrate creating characteristic signatures where z-score is used to select the words that eventually form the signature. We follow a similar method of selecting features as we did in case of characteristic signatures based on frequency. We will demonstrate using the same hypothetical document-term matrix we have used in the earlier scenarios. A subset of the features are features are selected from the document-term matrix based on the criteria that the feature or word must appear in at least k (threshold) documents in ANY one class. We can recall that this step is identical to the first scenario. Table 3.8 show the features that satisfy the threshold criteria and the z-score values of each feature (i.e. excluding features f_6 and f_9). We calculate the absolute z-score value for each term in the collection for each class using the following formula: $AbsoluteZscore_f = ABS((x - \mu)/\sigma)$ where x is the value of feature f

Feature	f_2	f_5	f_{10}	f_3	f_1
Entropy	0.26	0.26	0.34	0.37	0.45

(a)

Doc \ Feature	f_2	f_5	f_{10}	f_3	f_1
d_1	1	0	0	3	1
d_2	0	0	0	1	1
d_3	0	1	1	2	0
AF I	1	1	1	6	2
d_4	0	3	2	0	1
d_5	0	2	0	1	0
d_6	1	4	3	0	0
AF II	1	9	5	1	1
d_7	4	0	1	0	0
d_8	3	0	0	1	1
d_9	2	1	0	1	0
AF III	9	1	1	2	1

(b)

Class	Signature length (words)	f_3	f_2	f_5	f_1	f_{10}
I	3	1	1	1		
	4	1	1	1	6	
	5	1	1	1	6	2
II	3	1	9	5		
	4	1	9	5	1	
	5	1	9	5	1	1
III	3	9	1	1		
	4	9	1	1	2	
	5	9	1	1	2	1

(c)

TABLE 3.7: (a) Features satisfying k and q threshold sorted based on entropy. (b) Absolute frequency of the qualifying features and based on classes. (c) Characteristic signatures based on entropy of features.

in the document term matrix, is the mean of all the values of the feature in the document-term matrix for a particular class and σ is the standard deviation of feature f for that specific class. We have illustrated how to calculate absolute z-score for one of the features i.e. f_1 in Table 3.8(a). We calculate the μ and σ for each feature for the specific classes. Based on the values of μ and σ for each class we compute the absolute z-score for every occurrences of the feature as shown in Table 3.8(a). Similarly, we compute the absolute z-score for all the features that satisfy the threshold k, represented in Table 3.8(b). It should be noted that since this document-term matrix is a dummy matrix, used to illustrate the signature making process, few of the features do not have a z-score. The last row in Table 3.8(b) total z-score shows the sum of the z-scores for each feature across all the classes in the document-term matrix. Our next step would be to sort the features based on increasing sum of absolute z-score values for each feature shown in Table 3.9(a). Thus we have the list of features which should be considered for creating characteristic signatures unique for each of the classes. Similar to the

previous scenarios, we will create signatures of length 4,6 and 8 words based on the absolute frequency of the features.

Doc \ Feature	f_1			
	AF	ZS	μ	σ
d_1	1	0.58		
d_2	1	0.58	0.66	0.58
d_3	0	1.15		
d_4	1	1.15		
d_5	0	0.58	0.33	0.58
d_6	0	0.58		
d_7	0	0.58		
d_8	1	1.73	0.33	0.58
d_9	0	0		

(a)

Doc \ Feature	f_1		f_3		f_5		f_8		f_1		f_3		f_5		f_8	
	AF	ZS	AF	ZS	AF	ZS	AF	ZS	AF	ZS	AF	ZS	AF	ZS	AF	ZS
d_1	2	0.58	1	1.15	3	1	0	-	0	0.58	0	-	0	-	0	0.58
d_2	1	0.58	0	0.58	1	1	0	-	0	0.58	0	-	0	-	0	0.58
d_3	0	1.15	0	0.58	2	0	0	-	1	1.15	0	-	0	-	1	1.15
d_4	1	1.15	0	0.58	0	0.58	1	0.58	3	0	2	1	0	1.15	2	0.22
d_5	0	0.58	0	0.58	1	1.15	2	1.15	2	1	0	1	1	0.58	0	1.09
d_6	0	0.58	1	1.15	0	0.58	1	0.58	4	1	1	0	1	0.58	3	0.87
d_7	0	0.58	4	1	0	1.15	0	-	0	0.58	0	-	0	-	1	1.15
d_8	1	1.73	3	0	1	0.58	0	-	0	0.58	0	-	0	-	0	0.58
d_9	0	0	2	1	1	0.58	0	-	1	1.15	0	-	0	-	0	0.58
Total z-score		6.93		6.62		6.62		2.31		6.62		2		2.31		6.8

(b)

Note:AF and ZS represents absolute frequency and Z-score respectively

TABLE 3.8: (a) Table illustrating z-score calculation for feature f_1.
(b) Table depicting z-score values for all features for each document.

Table 3.9(b) depicts the sum of absolute frequency of the features for each class. Therefore, a signature based on z-score using absolute frequency of features and of length four words would consider the first four values in row AF-1 which corresponds to the absolute frequencies of the first four features (refer to Tables 3.9(b) and (c)). In other words, the characteristic signatures (of length four) based on minimum z-score using absolute frequency of words for class I, II and III are [0,0,0,1], [3,2,4,1] and [0,0,0,9] respectively using features f_7, f_8, f_4 and f_2. We

have described how characteristic signatures can be built based on various properties of the absolute frequency of the words or features like frequency, entropy and z-score. In the following sections we will discuss how to build characteristic signatures unique to classes by using other metrics of the features like relative frequencies of the words or different forms of relative frequencies of the words (discussed later in section 3.5).

Feature	f_7	f_8	f_4	f_2	f_3	f_5	f_{10}	f_1
Z-score	2	2.31	2.31	6.62	6.62	6.62	6.8	6.93

(a)

Doc \ Feature	f_7	f_8	f_4	f_2	f_3	f_5	f_{10}	f_1
d_1	0	0	0	1	3	0	0	1
d_2	0	0	0	0	1	0	0	1
d_3	0	0	0	0	2	1	1	0
AF I	0	0	0	1	6	1	1	2
d_4	2	0	1	0	0	3	2	1
d_5	0	1	2	0	1	2	0	0
d_6	1	1	1	1	0	4	3	0
AF II	3	2	4	1	1	9	5	1
d_7	0	0	0	4	0	0	1	0
d_8	0	0	0	3	1	0	0	1
d_9	0	0	0	2	1	1	0	0
AF III	0	0	0	9	2	1	1	1

(b)

Class	Signature length (words)	f_7	f_8	f_4	f_2	f_3	f_5	f_{10}	f_1
I	4	0	0	0	1				
	6	0	0	0	1	6	1		
	8	0	0	0	1	6	1	1	2
II	4	3	2	4	1				
	6	3	2	4	1	1	9		
	8	3	2	4	1	1	9	5	1
III	4	0	0	0	9				
	6	0	0	0	9	2	1		
	8	0	0	0	9	2	1	1	1

(c)

TABLE 3.9: (a) Features sorted in increasing order of absolute z-score.
(b) Table showing sum of absolute frequency of features for class I, II and III
(c) Characteristic signatures based on minimum z-score.

3.4 Signatures Using Relative Frequency of Words

In the beginning of section 3.3 we introduced a dummy document-term matrix that depicts absolute frequency of the features for a list of nine documents belonging to three classes. This document-term matrix become the basis of creating signatures using absolute frequency and essentially three different approaches unfolded from the document-term matrix. In this section, we will discuss yet another method of creating signatures unique to each classes using the relative frequency of the words,

instead of their absolute frequencies. The main difference in creating signatures using relative frequencies of words would lie in the document-term matrix that we use in the first place. Let us consider the Table shown in 3.10(a). This is identical to the document-term matrix we have been using the previous section illustrating the absolute frequency for features, except that we have added an extra column to the far right which depicts the total word count in each document. Table 3.10(b) denotes the relative frequency of the features in each document. The relative frequency is obtained by dividing absolute frequency of each feature in a document by the total word count in that document.

Doc \ Feature	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	Word counts in document
d_1	1	1	3	0	0	1	0	0	0	0	6
d_2	1	0	1	0	0	0	0	0	1	0	3
d_3	0	0	2	0	1	0	0	0	0	1	4
d_4	1	0	0	1	3	0	2	0	1	2	10
d_5	0	0	1	2	2	0	0	1	0	0	6
d_6	0	1	0	1	4	0	1	1	0	3	11
d_7	0	4	0	0	0	0	0	0	0	1	5
d_8	1	3	1	0	0	0	0	0	0	0	5
d_9	0	2	1	0	1	0	0	0	1	0	5

(a)

Doc \ Feature	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
d_1	0.17	0.17	0.50	0	0	0.17	0	0	0	0
d_2	0.33	0	0.33	0	0	0	0	0	0.33	0
d_3	0	0	0.50	0	0.25	0	0	0	0	0.25
d_4	0.10	0	0	0.10	0.30	0	0.20	0	0.10	0.20
d_5	0	0	0.17	0.33	0.33	0	0	0.17	0	0
d_6	0	0.09	0	0.09	0.36	0	0.09	0.09	0	0.27
d_7	0	0.80	0	0	0	0	0	0	0	0.20
d_8	0.20	0.60	0.20	0	0	0	0	0	0	0
d_9	0	0.40	0.20	0	0.20	0	0	0	0.20	0

(b)

TABLE 3.10: (a) Document-term matrix depicting absolute frequency and word counts per document.
(b) Document-term matrix based on 3.4.1(a) representing relative frequency of features.

The Table shown in 3.10(b) is the main basis for creating characteristic signature using relative frequency of the features. This approach of using relative frequency normalizes the occurrences of the features in the document. In other words, larger document would not influence the signature with its own contents of words. After obtaining the document-term matrix using relative frequency, the characteristic signatures will be generated based on the frequency, entropy and z-score of the relative frequencies of the features. The method followed to generate the characteristic signatures for each class follows the same process discussed in section 3.3.

All the steps thus conducted on Table 3.2 should be replicated for Table 3.10 to create characteristic signatures based on frequency, entropy and z-score using relative frequency of features. In order to create the characteristic signatures based on frequency of the relative frequency of the features, we will use some findings that are already made in the section 3.3. The first step is to select a sub-set of the feature based on threshold k, where features not occurring at least k times in any of the three classes will be eliminated. For the sake of illustration, assuming a k value of 2 the features that satisfy the threshold k are $f_1, f_2, f_3, f_4, f_5, f_7, f_8$ and f_{10}. Features f_6 and f_9 are eliminated since they do not satisfy the condition for threshold k. We are skipping the detailed steps of selection of the features satisfying the threshold value because we have already established this outcome in the previous section (refer Table 3.4). Table 3.11(a) shows the selected features and the sum of the relative frequencies across all documents sorted in descending order.

Table 3.11 represents the characteristic signatures based on frequency of relative frequency of the words that are unique to class I, II and III. Its interesting to note that these signatures generated from relative frequency includes different features from the signatures generated using absolute frequency of the words. Therefore, they will show different degree of classification which we will observe and discuss in the next chapter. To illustrate the creation of characteristic signature based on entropy of relative frequency of the features, we will refer to table 3.6(b) which depicts the features that satisfy the threshold value of k and q, for qualifying to

Feature	f_2	f_3	f_5	f_{10}	f_1	f_4	f_7	f_8
RF	2.63	1.90	1.44	0.92	0.80	0.52	0.29	0.26

(a)

Doc \ Feature	f_2	f_3	f_5	f_{10}	f_1	f_4	f_7	f_8
d_1	0.17	0.50	0	0	0.17	0	0	0
d_2	0	0.33	0	0	0.33	0	0	0
d_3	0	0.50	0.25	0.25	0	0	0	0
RF I	0.17	1.33	0.25	0.25	0.50	0	0	0
d_4	0	0	0.30	0.20	0.10	0.10	0.20	0
d_5	0	0.17	0.33	0	0	0.33	0	0.17
d_6	0.09	0	0.36	0.27	0	0.09	0.09	0.09
RF II	0.09	0.17	0.99	0.47	0.10	0.52	0.29	0.26
d_7	0.80	0	0	0.20	0	0	0	0
d_8	0.60	0.20	0	0	0.20	0	0	0
d_9	0.40	0.20	0.20	0	0	0	0	0
RF III	1.8	0.40	0.20	0.20	0.20	0	0	0

(b)

Class	Signature length (words)	f_2	f_3	f_5	f_{10}	f_1	f_4	f_7	f_8
I	4	0.17	1.33	0.25	0.25				
	6	0.17	1.33	0.25	0.25	0.50	0		
	8	0.17	1.33	0.25	0.25	0.50	0	0	0
II	4	0.09	0.17	0.99	0.47				
	6	0.09	0.17	0.99	0.47	0.10	0.52		
	8	0.09	0.17	0.99	0.47	0.10	0.52	0.29	0.26
III	4	1.8	0.40	0.20	0.20				
	6	1.8	0.40	0.20	0.20	0.20	0		
	8	1.8	0.40	0.20	0.20	0.20	0	0	0

(c)

Note:RF represents Relative Frequency

TABLE 3.11: (a) Features satisfying threshold k, sorted in descending order of relative frequency.
(b) Sum of relative frequencies for features that qualified for building characteristic signatures based on class I, II and III. Depicted in row RF I, RF II and RF III
(c) Characteristic signatures of class I, II and III with signatures of length 4,6 and 8 words.

participate in signature creation. Features f_1, f_2, f_3, f_5 and f_10 will be used to create the signatures based on their entropies. Table 3.12(a) depicts the features in ascending order of their entropy. In order to build a signature unique to the

Feature	f_2	f_3	f_5	f_1	f_{10}
Entropy	2.03	0.13	0.29	0.39	0.44

(a)

Doc \ Feature	f_2	f_3	f_5	f_1	f_{10}
d_1	0.17	0.50	0	0.17	0
d_2	0	0.33	0	0.33	0
d_3	0	0.50	0.25	0	0.25
RF I	0.17	1.33	0.25	0.50	0.25
d_4	0	0	0.30	0.10	0.20
d_5	0	0.17	0.33	0	0
d_6	0.09	0	0.36	0	0.27
RF II	0.09	0.17	0.99	0.10	0.47
d_7	0.80	0	0	0	0.20
d_8	0.60	0.20	0	0.20	0
d_9	0.40	0.20	0.20	0	0
RF III	1.8	0.40	0.20	0.20	0.20

(b)

Class	Signature length (words)	f_3	f_2	f_5	f_1	f_{10}
I	3	0.17	1.33	0.25		
	4	0.17	1.33	0.25	0.50	
	5	0.17	1.33	0.25	0.50	0.25
II	3	0.09	0.17	0.99		
	4	0.09	0.17	0.99	0.10	
	5	0.09	0.17	0.99	0.10	0.47
III	3	1.8	0.40	0.20		
	4	1.8	0.40	0.20	0.20	
	5	1.8	0.40	0.20	0.20	0.20

(c)

TABLE 3.12: (a) Features satisfying k and q threshold sorted based on entropy. (b) Relative frequency of the qualifying features and based on classes. (c) Characteristic signatures based on entropy of features.

classes we will choose the features as they sorted based on the required length of the signature needed. Table 3.12(c) shows the signatures created for class I, II and III based on minimum entropy of the features using relative frequency of the features. The characteristic signatures based on z-score of the relative frequency of the words are generated using the same document-term matrix of relative frequency. The features must satisfy the threshold k, in order to participate in signature creation. Assuming a value of two for the threshold k, the features selected are $f_1, f_2, f_3, f_4, f_5, f_7, f_8$ and f_10. Table 3.13(a) represents the relative frequency of each feature and the z-score value of the relative frequency. The sum of z-score for each feature across all documents are sorted in to increasing order. This marks the order in which the features are chosen based on the required length of the characteristic signature. Table 3.13(c) shows the signatures of length 4,6 and 8 words based on the z-score of the relative frequency of the words.

Doc \ Feature	f_1		f_3		f_5		f_8		f_1		f_3		f_5		f_8	
	RF	ZS	RF	ZS	RF	ZS	RF	ZS	RF	ZS	RF	ZS	RF	ZS	RF	ZS
d_1	0.17	0.02	0.17	1.15	0.5	0.58	0	-	0	0.58	0	-	0	-	0	0.58
d_2	0.33	0.99	0	0.58	0.33	1.15	0	-	0	0.58	0	-	0	-	0	0.58
d_3	0	1.01	0	0.58	0.5	0.58	0	-	0.25	1.15	0	-	0	-	0.25	1.15
d_4	0.1	1.15	0	0.58	0	0.58	0.1	0.54	0.3	1	0.2	1.03	0	1.02	0.2	0.31
d_5	0	0.58	0	0.58	0.17	1.15	0.33	1.15	0.33	0	0	0.97	0.17	0.98	0	1.12
d_6	0	0.58	0.09	1.15	0	0.58	0.09	0.61	0.36	1	0.09	0.07	0.09	0.04	0.27	0.81
d_7	0	0.58	0.8	1	0	1.15	0	-	0	0.58	0	-	0	-	0.2	1.15
d_8	0.2	1.15	0.6	0	0.2	0.58	0	-	0	0.58	0	-	0	-	0	0.58
d_9	0	0.58	0.4	1	0.2	0.58	0	-	0.2	1.15	0	-	0	-	0	0.58
Total z-score		6.64		6.62		6.93		2.30		6.62		2.07		2.04		6.86

(a)

Feature	f_7	f_8	f_4	f_2	f_5	f_1	f_{10}	f_3
Z-score	2.07	2.04	2.3	6.62	6.62	6.64	6.86	6.93

(b)

Class	Signature length (words)	f_7	f_8	f_4	f_2	f_5	f_1	f_{10}	f_3
I	4	0	0	0	0.17				
	6	0	0	0	0.17	0.25	0.50		
	8	0	0	0	0.17	0.25	0.50	0.25	1.33
II	4	0.29	0.26	0.52	0.09				
	6	0.29	0.26	0.52	0.09	0.99	0.1		
	8	0.29	0.26	0.52	0.09	0.99	0.1	0.47	0.17
III	4	0	0	0	1.80				
	6	0	0	0	1.80	0.20	0.20		
	8	0	0	0	1.80	0.20	0.20	0.20	0.40

(c)

Note:RF and ZS represents relative frequency and Z-score respectively

TABLE 3.13: (a) Features sorted in increasing order of absolute z-score.
(b) Table depicting z-score values for all features for each document.
(c) Characteristic signatures based on minimum z-score.

3.5 Signatures Using Relative Rest

This approach is essentially an adaptation of the previous method where we use relative frequency with a little modification. In this approach the signatures of length n are created following the method described above. In addition to that we

add an extra element at the end of the signature. This element contains the sum of the non-zero relative frequencies of all the terms which were excluded from the signature in the first place. We adapted this method with the belief that the value in the last element (referred as rest) would further normalize the signature and would essentially use the information about relative frequencies of non-significant term to improve the weights of the terms selected in the signature. The relative frequency of the rest of the non-selected features (referred as relative-rest) is then used to create signatures using minimum entropy and minimum z-score. If we recall from the previous section 3.4, when creating signatures based on frequency of relative frequency of the features we discarded features f_6 and f_9 due to not satisfying the criteria for threshold k. The features to be used in creating signatures were sorted in the order $f_2, f_3, f_5, f_10, f_1, f_4, f_7, f_8$. To recall the way, we created signatures of length n in the earlier sections, was to select the first n features and ignore the rest of the features. For an example, to create a signature of length 4 we will consider the first four features in the sorted list i.e. f_2, f_3, f_5, f_10 and not consider features f_1, f_4, f_7, f_8 for a signature of length 4. The idea for building signatures based on the rest of the non-selected features is to use the information stored in the relative frequencies of the features that qualify due to the length of the signature necessary. Table 3.14 shows characteristic signatures of length 4,6 and 8 using the relative rest of the features. We can observe in Table 3.14, that characteristic signatures of length 4 and 6 for all the three classes contain an extra cell whose value is the sum of the features that are not used in the first n features of the signature. The sum of the relative frequency of the rest of the features could be computed from Table 3.11(b) in the previous section. Therefore, the characteristic signatures of length 4 based on frequency using relative-rest of features are [0.17, 1.33, 0.25, 0.25, 0.50], [0.09, 0.17, 0.99, 0.47, 1.17], [1.8, 0.40, 0.20, 0.20, 0.20] for classes I, II, III respectively. Similarly, we have generated the characteristic signatures of length 6 words. However, it should be noted that for signatures of length 8 words there were no remaining features that satisfied the k threshold. Hence, the relative rest is 0 in this example. In an actual dataset, this situation never arose since the number of features qualifying are significantly

Class	Signature length(words)	f_2	f_3	f_5	f_{10}	RR (sum of RF of f_1,f_4, f_7,f_8)				
I	4	0.17	1.33	0.25	0.25	0.50				
II		0.09	0.17	0.99	0.47	1.17				
III		1.8	0.40	0.20	0.20	0.20				
		f_2	f_3	f_5	f_{10}	f_1	f_4	RR (f_7 and f_8)		
I	6	0.17	1.33	0.25	0.25	0.50	0	0		
II		0.09	0.17	0.99	0.47	0.10	0.52	0.55		
III		1.8	0.40	0.20	0.20	0.20	0	0		
		f_2	f_3	f_5	f_{10}	f_1	f_4	f_7	f_8	RR
I	8	0.17	1.33	0.25	0.25	0.50	0	0	0	0
II		0.09	0.17	0.99	0.47	0.10	0.52	0.29	0.26	0
III		1.8	0.40	0.20	0.20	0.20	0	0	0	0

Note:RR and RF represents relative rest and relative frequency respectively

TABLE 3.14: Characteristic signatures for class I, II, III based on frequency using relative-rest.

more than the signature lengths.

Now, using similar approach we can create the characteristic signatures based on entropy using the relative rest of the features for all the three classes. Table 3.15 shows the characteristic signatures based on entropy. The relative rest value for the features are computed as a sum of the relative frequency of features which are not included in the original signature length. We can refer to Table 3.12(b) to compute the sum of the relative frequency of the rest of the features for each class. We can draw from Table 3.14 that characteristic signatures of length 4 based on entropy using relative rest values of the features are [0.17,1.33, 0.25, 0.50, 0.25], [0.09, 0.17, 0.99, 0.10, 0.47], [1.8, 0.40, 0.20, 0.20, 0.20] for class I, II, III respectively. We should note that in these signatures of length n, we have $n + 1$ elements, since the last element is the relative-rest value of the remaining features which otherwise would not be included in a signature of length n. So far we have a described how the characteristic signatures specific to each class are created by adding the relative rest values to the already created signatures using relative frequency. We will illustrate the last way we have devised to create characteristic signatures based on z-score using the same concept of relative-rest frequencies of the features. We will take a look back at Table 3.13(c) which exhibits

Class	Signature length(words)	f_3	f_2	f_5	RR (sum of RF of f_1,f_{10})		
I	3	0.17	1.33	0.25	0.50		
II		0.09	0.17	0.99	0.10		
III		1.8	0.40	0.20	0.20		
		f_3	f_2	f_5	f_1	RR (f_{10}	
I	4	0.17	1.33	0.25	0.50	0.25	
II		0.09	0.17	0.99	0.10	0.47	
III		1.8	0.40	0.20	0.20	0.20	
		f_3	f_2	f_5	f_1	f_{10}	RR
I	5	0.17	1.33	0.25	0.50	0.25	0
II		0.09	0.17	0.99	0.10	0.47	0
III		1.8	0.40	0.20	0.20	0.20	0

Note:RR represents relative rest

TABLE 3.15: Characteristic signatures for class I, II and III based on entropy of relative-rest.

the characteristic signatures based on z-score using only relative frequency of the features. It is apparent from the previous approaches discussed in this section, that we would add an extra element to the signatures of length n, whose value is the sum of the relative frequencies of the remaining features that are shown in the header of the Table 3.13(c).

In Table 3.16 we have represented the characteristic signatures created based on z-scores using relative-rest of the features. To create a signature of length n, we added an extra element to the signature whose value is the sum of remaining features i.e., the ones excluded from the initial signature of length n. To summarize the entire process of creating signatures, the first step involves creating a document-term matrix of all the words or features and the documents sorted in the order of their class. This is followed by a series of preprocessing of the features and vectors and selecting a subset of the features which is assumed to have useful information for creating unique signatures for each class. We used two key metrics of words or features to build the document-term matrix. These were absolute frequency of the words and relative frequency of the words. For each of these metrics used for features, we further consider selecting the feature on the basis

Class	Signature length(words)	f_7	f_8	f_4	f_2	RR (sum of RF of f_5,f_1, f_{10},f_3)				
I	4	0	0	0	0.17	2.33				
II		0.29	0.26	0.52	0.09	1.73				
III		0	0	0	1.8	1.00				
		f_2	f_3	f_5	f_10	f_1	f_4	RR (f_7 and f_8)		
I	6	0	0	0	0.17	0.25	0.50	1.58		
II		0.29	0.26	0.52	0.09	0.99	0.10	0.64		
III		0	0	0	1.8	0.20	0.20	0.60		
		f_2	f_3	f_5	f_10	f_1	f_4	f_7	f_8	RR
I	8	0	0	0	0.17	0.25	0.50	0.25	1.33	0
II		0.29	0.26	0.52	0.09	0.99	0.10	0.47	0.17	0
III		0	0	0	1.8	0.20	0.20	0.20	0.4	0

Note:RR and RF represents relative rest and relative frequency respectively

TABLE 3.16: Characteristic signatures based on z-score using relative-rest frequencies of features.

of the frequency, entropy and z-score of these metrics for each class in the collection. As discussed in section 3.5 we have also devised a method to tweaked the signatures created using relative frequency of the words. This signatures includes information of the features which were considered insignificant and thus ignored when create signatures. So far we have devised methods to create signatures from training data which will eventually be subjected to classify documents from test dataset. In the following section we will discuss how the test data are classified using the signatures created.

3.6 Classification of Documents Using Signatures

After obtaining the characteristic signatures from the training set, we move our focus to create a feature vector for each test document that we want to classify in to one of these three classes. We would use that feature vector and match it against each class signatures and determining their similarity. We use the same approach to create a document-term matrix for each of the test documents. In the next step, we match every single term in the characteristic signature with every term in a test document. If a feature in the characteristic signature exists in the test

document, we calculate the absolute frequency or relative frequency of that term in the test document and add it to the feature vector. If a term in the characteristic signature does not exist in the test document, we append an arbitrarily small value (ϵ) to the feature vector for that specific feature. Considering a signature of length n, we add an offset that essentially keeps count of the number of zeros or ϵ in a signature. Currently, we have four signatures of length n+1 which are essentially distributions of the same feature words used to build the characteristic signatures. The characteristic signatures represent the features of the distribution for their respective classes. The signature of the test document is then compared to each of the characteristic signatures of each class and their distances are measured. We used KullbackLeibler divergence (KL Divergence) to measure the distance between each distribution. KL divergence is given by the below formula

$$D(p||q) = \int p(x) log \frac{p(x)}{q(x)} \tag{3.2}$$

KL divergence is used to measure the distance between two given distributions and is always non-negative. Its is zero only when p is equal to q, and hence it serves as a distance measure. However, it should be noted that it is not a proper distance metric since it does not obey triangle inequality and generally $DKL(p||q)$ is not the same as $DKL(q||p)$. In our experiments, the characteristic signatures serve as our distributions. The characteristic signatures obtained from the training sets, which represent each of the classes are represented as $q(x)$ and the characteristic signature obtained from the test document is denoted by $p(x)$. Therefore, using KL divergence we obtain the distance between the characteristic signatures of each classes and the test signature. The pair with minimum KL divergence value indicates higher similarity among the two signatures. Hence, the test document is categorized in to the class which the characteristic signature represents. The KL divergence is calculated for all test documents and the three characteristic signatures belonging to three classes. Once all the test documents are categorized into one of the three classes, we evaluate the correctness and quality of the classification using accuracy, precision and recall. The KL divergence plays

an important role to establish the fact that signatures can be used to eliminate classes from labels which dont exist in the training set. In order to achieve this goal, we decided to introduce a KL divergence threshold in the scenario where we introduce an additional class in the test set. We classify a test document using KL divergence into one of the predefined classes, on the basis of the closest match with a signature from a specific class with the feature vector of the test document. The test document is assigned to the class which has the minimum KL divergence value among all the signatures. The idea is when a document from an unknown label is introduced in the test set, the KL-divergence value generated would be quite high due to its dissimilarity with the training set. We introduce the KL divergence threshold to detect such wide range of differences in KL divergence value and eliminate documents with large KL divergence value as unknown labels.

Chapter 4

Experimental setup and results

4.1 Datasets

Sentence Corpus- This collection consists of introductions and abstracts from scientific articles. The text is categorized in to three classes, PLoS computational Biology (PLOS), machine learning repository on arXiv (ARXIV) and the psychology journal judgement and decision making (JDM).

News articles originating from BBC news have served as a benchmark in text mining researches. This dataset consists of 2200 odd documents from the news website which corresponds to stories across five different topic areas from the year 2004-2005. This collection consists of 5 classes namely, business, entertainment, politics, sports and technology. In our experiments we have used politics, technology and entertainment as the three different classes. We built 7 different samples of training set collections and test set collection for each dataset. We run our experiments on each of these sampled datasets and then compute the average of the accuracy, precision and recall values obtained from each experiments. In the sections below we have tried to highlight the results of classification using characteristic signatures. It is worth to note that we have varied the signature length for different values of threshold (k) to study the classification of test documents.

4.2 Data Preprocessing

The above two corpora were then preprocessed in the following manner:

Words were stemmed using a NLTK Porter Stemmer which follows the algorithm presented by Porter in his work at Cambridge [27]. Stemming reduces words to similar roots and thus helps in removing redundancies and reducing the total number of features that would otherwise be considered separate.

Common stop words in that occurs in English were removed which would further filter the vocabulary. The inbuilt scikit learn stop words removal module was used to perform the operation.

Words that occurred less than 8 times or more than 1000 times were dropped from the vocabulary. Through a series of experiments, we observed that in the case of our approach these words were adding noise to the classifier.

4.3 Random Sampling

For both the datasets, we had created seven different samples of training set and test set to perform the experiments. As described earlier the BBC dataset consists of five different classes. However, in this study we decided to limit the total number of unique classes to include to three. Hence, from the BBC dataset we select documents from topic technology, entertainment and politics. The training set for both the dataset is supposed to consist of 200 documents of each class and the test set is supposed to consist of 100 documents of each class. Thus the entire training set would contain 600 documents for all the three classes and the test set should contain 300 documents for all the three classes. We have randomly sampled the training of each class for both the dataset. We randomly selected 200 documents for each class for Sentence Corpus and BBC dataset. We have repeated this step for seven times to generate seven samples of the training set for consisting of three classes. A similar approach was taken to create seven different samples for the

test set for consisting of three different classes for both sentence corpus and BBC dataset. Once we have generated seven samples of the training set and test set for both the datasets, we took up the experiments planned for these samples. These training sets were essentially used to create characteristic signatures and these signatures were eventually used to classify documents from the test dataset. The signatures were created and tested for each of the seven samples of the dataset and compiled in to one location. This procedure was followed for both sentence corpus and BBC dataset. During these experiments, performance metrics were captured for analysis like accuracy, precision and recall. Performance metrics were captured for each method of classification using a specific type of characteristic signature. The details will be described in section 4.4. Once the performance metrics were obtained for all the seven samples created for the datasets, we evaluate the average for all the metrics over the seven samples. The average accuracy, recall and precision for the seven samples will account for the performance for that particular dataset. All the analysis drawn in the next section between the two datasets are based on the averaged performance of seven samples.

4.4 Results

4.4.1 Plots and Analysis

Figure 4.1 depicts the performance of characteristic signatures based on frequency using Absolute Frequency (AF), Relative frequency (RF) and Relative Rest (RR) using a threshold of 50 words. The left column corresponds to Sentence Corpus (SC) dataset and the column to the right corresponds to BBC dataset. In each of the plots the x-axis represents the length of the characteristic signatures and the y-axis represents a performance metric (accuracy, precision or recall). In case SC dataset all three kinds of frequency signatures have performed very close to each other. In SC dataset with signature length of 10 words, the signatures based on RF and AF has an accuracy of around 72% where the accuracy for RR is around the 63% mark. Signatures based on RF does the best among all the three and

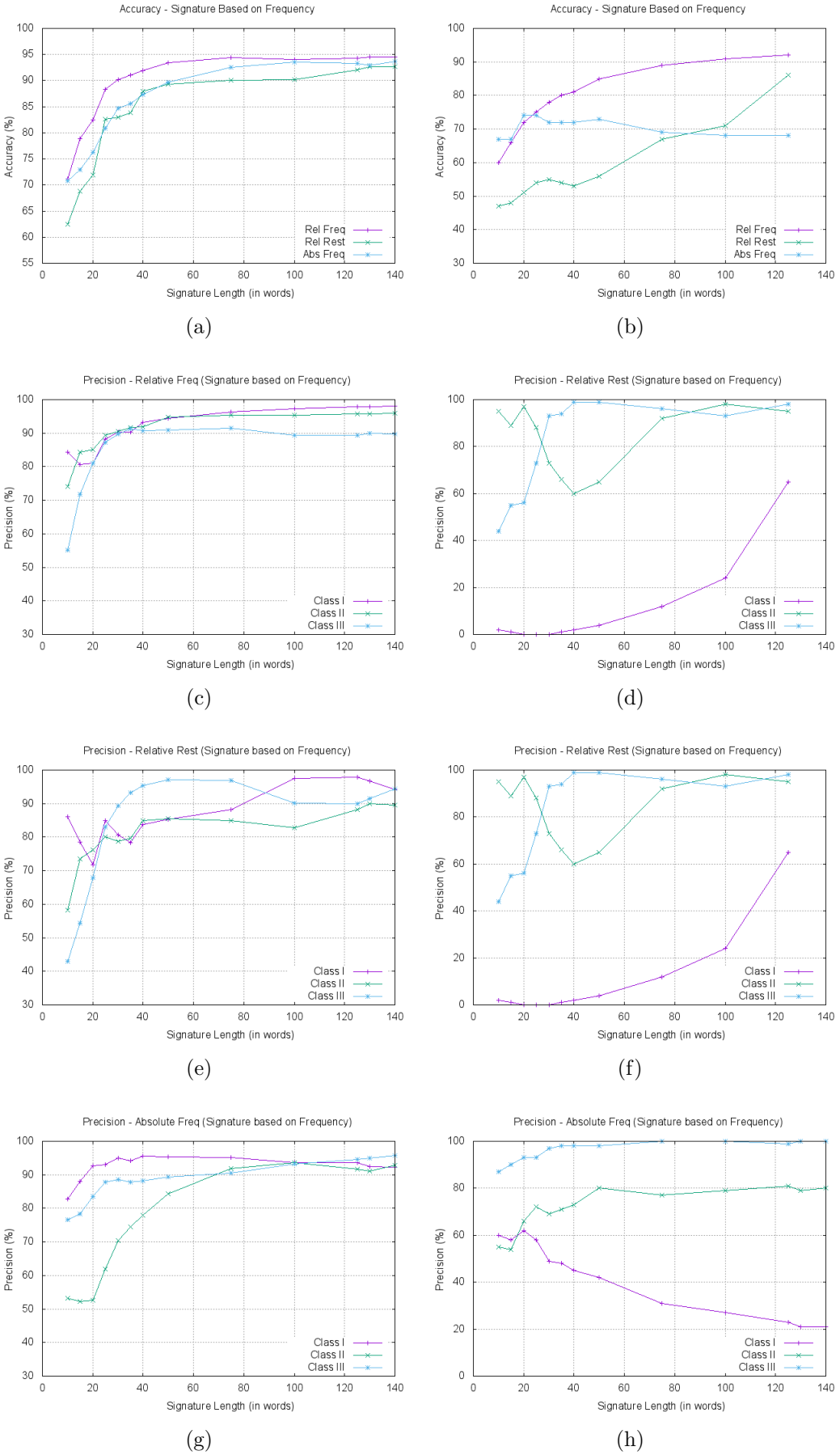

FIGURE 4.1: Comparison of accuracy and precision in SC(left column) and BBC (right column) datasets using characteristic signatures based on frequency (threshold $t = 50$).

attains a maximum accuracy of $\approx 95\%$ at a signature length of ≈ 75 words. It appears that signatures based on RF hit the 90% accuracy mark with a signature length of 30 words. The other two methods of signatures i.e. using RR and AF also reached a peak of $\approx 93\%$ and $\approx 94\%$ respectively. While in case of BBC dataset, signatures using RF has an accuracy of 60% when the signature length is at minimum, but eventually reaches a peak of $\approx 92\%$ accuracy. Signatures based on RR manages to reach a peak accuracy of $\approx 87\%$ at maximum signature length of 125 words. Looking closely at accuracy plot for SC dataset and BBC dataset, we could clearly the understand that the minimum length of the signature used is 10 words and have been gradually increased in finite steps. When the signature lengths are less (in the order of 40 words or lesser), any addition of features in the signature (i.e. increasing the signature length) increases the accuracy at a fast rate. This is very evident in the accuracy plot for SC dataset in figure 4.1 indicated by steep slope in the beginning of the curve when signature length is increased from 10 words to 30 words. This steep increase in the accuracy is attributed to the fact that the features used in the signature contain strong characteristic attributes that are unique to the classes. Interestingly, on further increasing the signature length in case of SC dataset from 30 words to approx. 60 words the rate of change in accuracy has slowed down and eventually reached 95% (for signatures based on RF). This is likely due to the lack of good features that are as effective as the earlier features in classifying the documents in to distinct classes. After a certain point (around the 70 words mark on x-axis) any further addition of words (features) to the signature doesnt enhance the accuracy. In other words, in case of SC dataset a signature length of 70 words would be sufficient to classify the documents to its best possible ability. We also observed that signatures created using RF of word frequency turned out to be more effective than signatures created using AF or RR of the word frequency. Now shifting our thoughts to the BBC dataset, it is evident that in case of RF the increase in the rate of accuracy on increasing the signature length was more consistent and gradual. The signatures using RR have an overall increasing trend in accuracy but the trend is not consistent throughout the variation of signature length. But, surprisingly the AF for word frequency in

case of BBC dataset did not perform at all. This is likely due to the nature of the contents of the documents. Absolute frequency of the words could not yield useful features with the ability to classify classes. However, in case of signatures using RF or RR, where both uses relative frequencies of the word to select the features, the overall trend in accuracy was increasing with increasing signature length. Hence, the strangeness in case of signatures using AF could be a result of varying document lengths and word counts, which got countered when relative frequency is used instead. Before discussing the precision plots generated by the signatures for each class, we should note that each signature type (based on AF, RF, RR) will generate separate plots for class I, II and III (as shown in figure 4.1). However, only some of the precision plots are presented in the figure in an effort to cover different trends depicted by these plots. The trend shown in the precision plots for signatures using RF for both SC and BBC dataset are consistent with the accuracy depicted by the Relative Frequency. Looking at the precision plot for signatures using RR in case of BBC dataset, we observe that the precision for class I and III have an increasing trend, not to mention that class III precision stayed around zero until the signature length reaches ≈30 words. Interestingly, the precision for class II took a significant dip between signature length of 20 through 40 words and then gradually started increasing. This sudden strangeness in the precision of class II, has resulted in the unevenness in the overall accuracy of the signatures using RR in case of BBC dataset. Moving on to the precision plots in case of signatures based on AF in BBC dataset, it is clear that all three classes indicated an overall increasing trend in precision until signature length reaches 20 words. This is aptly reflected in the accuracy plot for AF in BBC dataset. However, quite surprisingly the precision for class I in case of signatures using AF in BBC dataset kept declining and the other two classes did not improve much as well. Hence, the overall accuracy for AF in BBC dataset did not show much improvement.

Figure 4.2 represents the performance of characteristic signatures based on frequency using Absolute Frequency (AF), Relative frequency (RF) and Relative

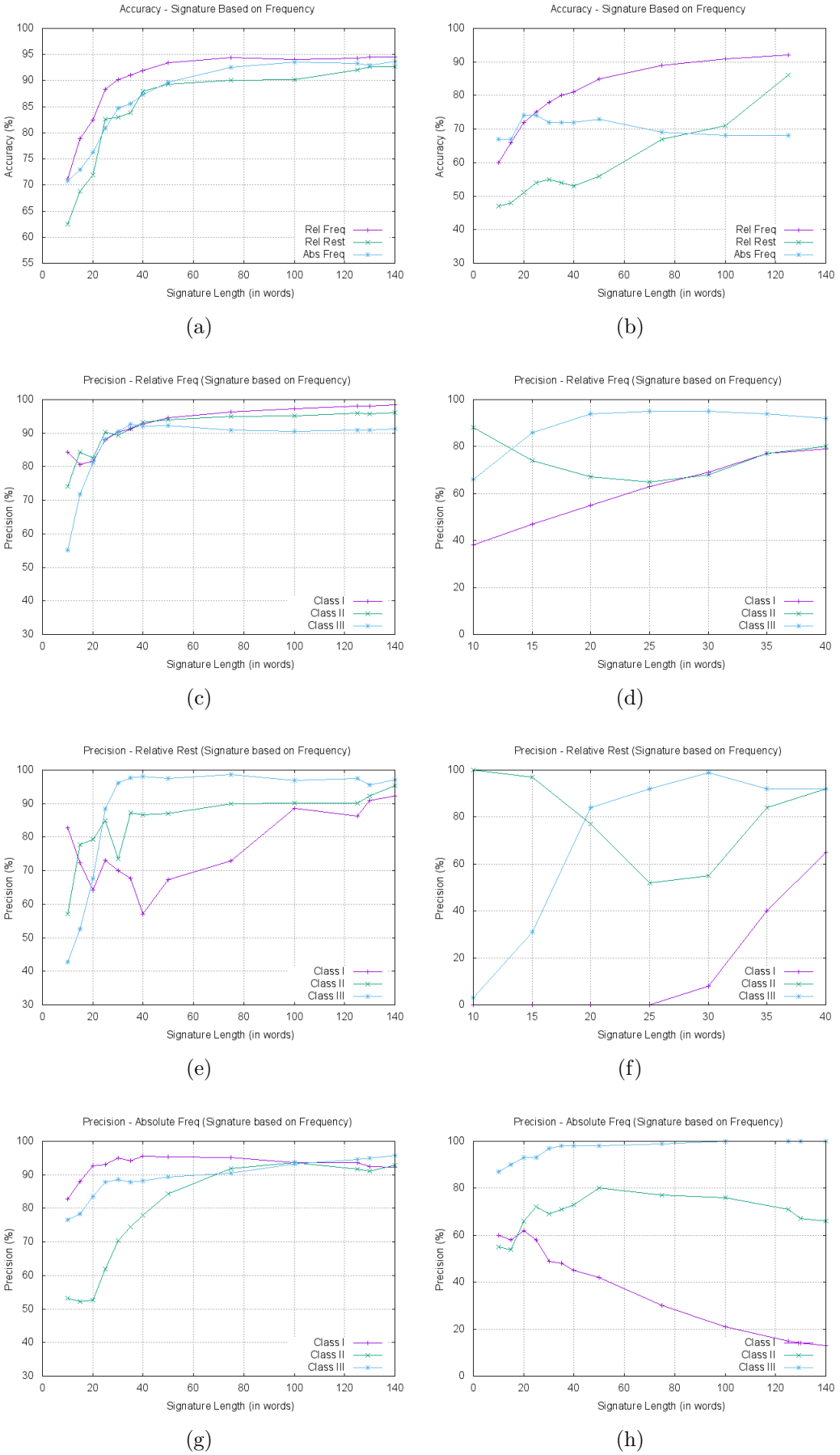

FIGURE 4.2: Comparison of accuracy and precision in SC(left column) and BBC (right column) datasets using characteristic signatures based on frequency (threshold $t = 75$)

Rest (RR) and using a threshold of 75 words. Similar to Figure 4.1, the left column shows plots obtained from Sentence Corpus dataset and the right column shows plots obtained from BBC dataset. In this experiment with threshold value set to 75, its evident that the overall trends in the plots are very similar to what we observed with threshold 50. Signatures using RF produces better accuracy for both the SC and BBC dataset. In case of SC dataset, the signatures based on frequency using AF of words performs close to signatures using RF. In general, the change in threshold value from 50 words to 75 words does not seem to alter the performance but any significant measure. Since the trends depicted are the same, the observations and explanations under figure 4.1 are valid in this scenario. In SC dataset with signature length of 10 words, the signatures based on RF and AF has an accuracy of 72% where the accuracy for RR is $\approx$ 62%. Signatures based on RF reaches a maximum of accuracy of $\approx$ 95% at a signature length of $\approx$ 75 words. The other two methods of signatures i.e. using RR and AF also reached a peak of $\approx$ 93% and $\approx$ 94% respectively. While in case of BBC dataset, signatures using RF has an accuracy of 60% when the signature length is at minimum, but eventually reaches a peak of $\approx$ 92% accuracy. Signatures based on RR manages to reach a peak accuracy of $\approx$ 87% at maximum signature length of $\approx$ 125 words. We would like to observe and study another method of building signature i.e. using entropy and understand its performance for the two datasets.

Figure 4.3 depicts the performance of characteristic signatures based on frequency using Absolute Frequency (AF), Relative frequency (RF) and Relative Rest (RR) using a threshold of 50 words. The left column corresponds to Sentence Corpus (SC) dataset and the column to the right corresponds to BBC dataset as described in the earlier figures. In figure 4.4.3 the accuracy plots for SC dataset indicates that the Minimum Entropy signatures based on AF appears to have performed better than in case of BBC dataset. However, in SC dataset signatures based on RF caught up with signatures based on AF around the 100 words mark for signature length. Nevertheless, all the three methods of minimum entropy signatures hit the same peak accuracy ($\approx$ 87%). In BBC dataset minimum entropy signatures based on RF had clearly performed better than AF and RR. We should recall

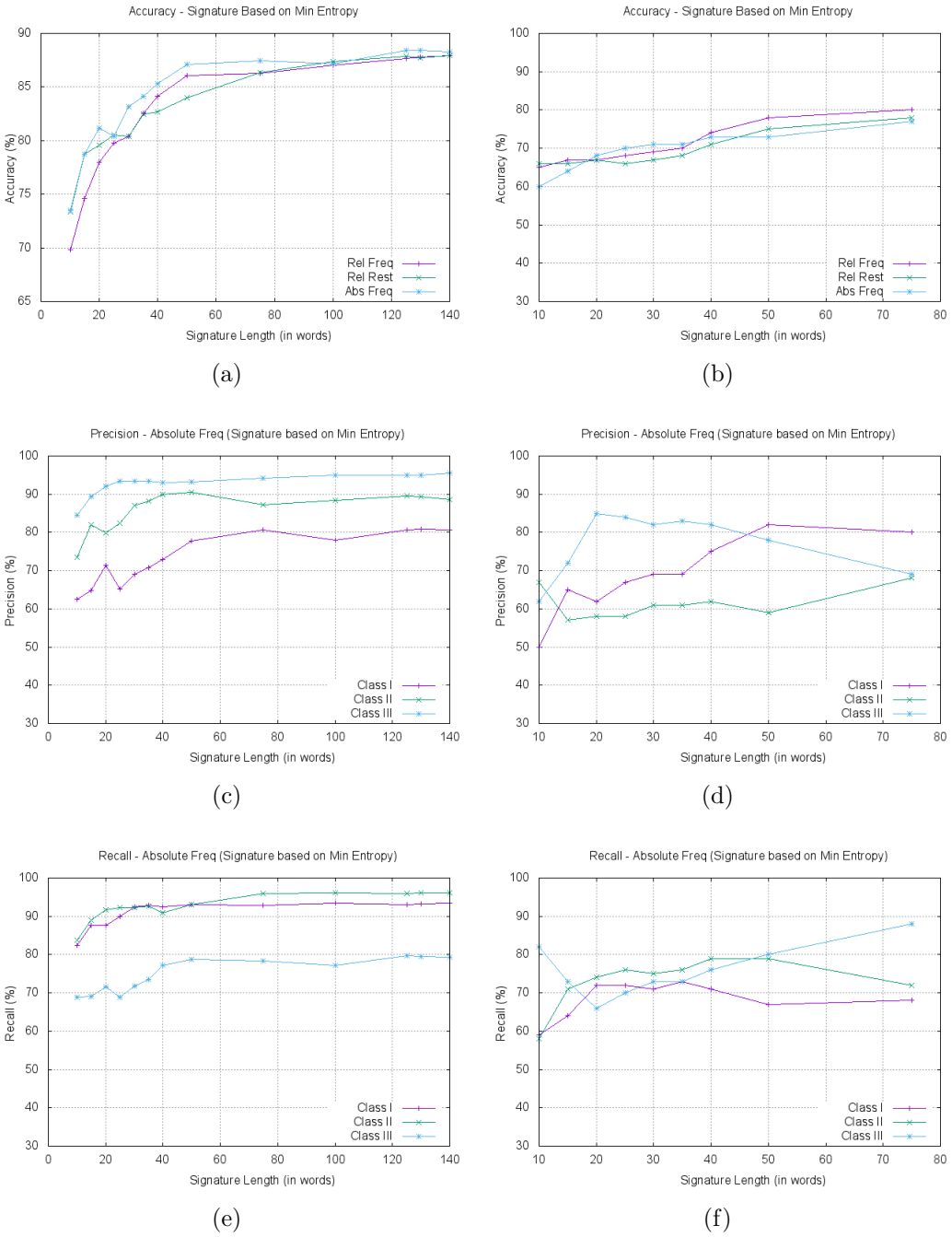

FIGURE 4.3: Comparison of accuracy and precision in SC(left column) and BBC (right column) datasets using characteristic signatures based on minimum entropy (threshold $t = 50$)

that in figure 4.1 and 4.2, the accuracy for BBC dataset with frequency signatures using AF had a strange downward trend which is missing in the case of signatures using minimum entropy. Thus, different properties (frequency, entropy or z-score) of collection of features or words in a text dataset is likely to portray different behavior. Again, we have not presented all the precision plots for the accuracy. The precision plot for minimum entropy signature using AF and RR is shown for SC and BBC datasets respectively. The third row in figure 4.3 represents the recall for minimum entropy signature using AF and RR for datasets SC and BBC respectively. The recall depicted in case of SC has been consistent and are above the 90% mark for classes I and II where as its close to $\approx$ 80% mark for class III. This indicates that most of the relevant documents were correctly classified by the signature. However, the recall in the BBC dataset for minimum entropy signatures using RR shows a little more interesting trend. The recall for class I and II kicks off at $\approx$ 70% mark and reaches $\approx$ 92% and $\approx$ 100% mark in the end. It is interesting to note that as the signature length increased, in case of recall for class III for signatures based on RR the gradual decrease in recall after the signature length crossed the 40 words mark is accompanied by a gradual increase in precision for class III.

In figure 4.4 we have shown the performance of z-score signatures based on RF, RR and AF for both datasets SC and BBC for a threshold of 50 words. In case of SC dataset z-score signatures based on RF has clearly performed much better than RR and AF. Signatures based on RF reached a peak accuracy of $\approx$ 90% at maximum signature length, whereas signatures based RR hit the 80% mark and went down further on increasing signature length from 125 to 140 words. In case of BBC dataset, the accuracy of the z-score signatures based on RF increased rapidly 60% to $\approx$ 80% while varying the signature lengths from 10 words to 20 words. This is imperative that the features added to the signature were rich in information and was effective to classify the classes. The accuracy for signatures using RF finally hit the $\approx$ 92% mark. Among the other signatures based on AF and RR, the later did moderately and attained a max accuracy of $\approx$ 82% whereas z-score signatures based on AF were not able to classify the classes efficiently.

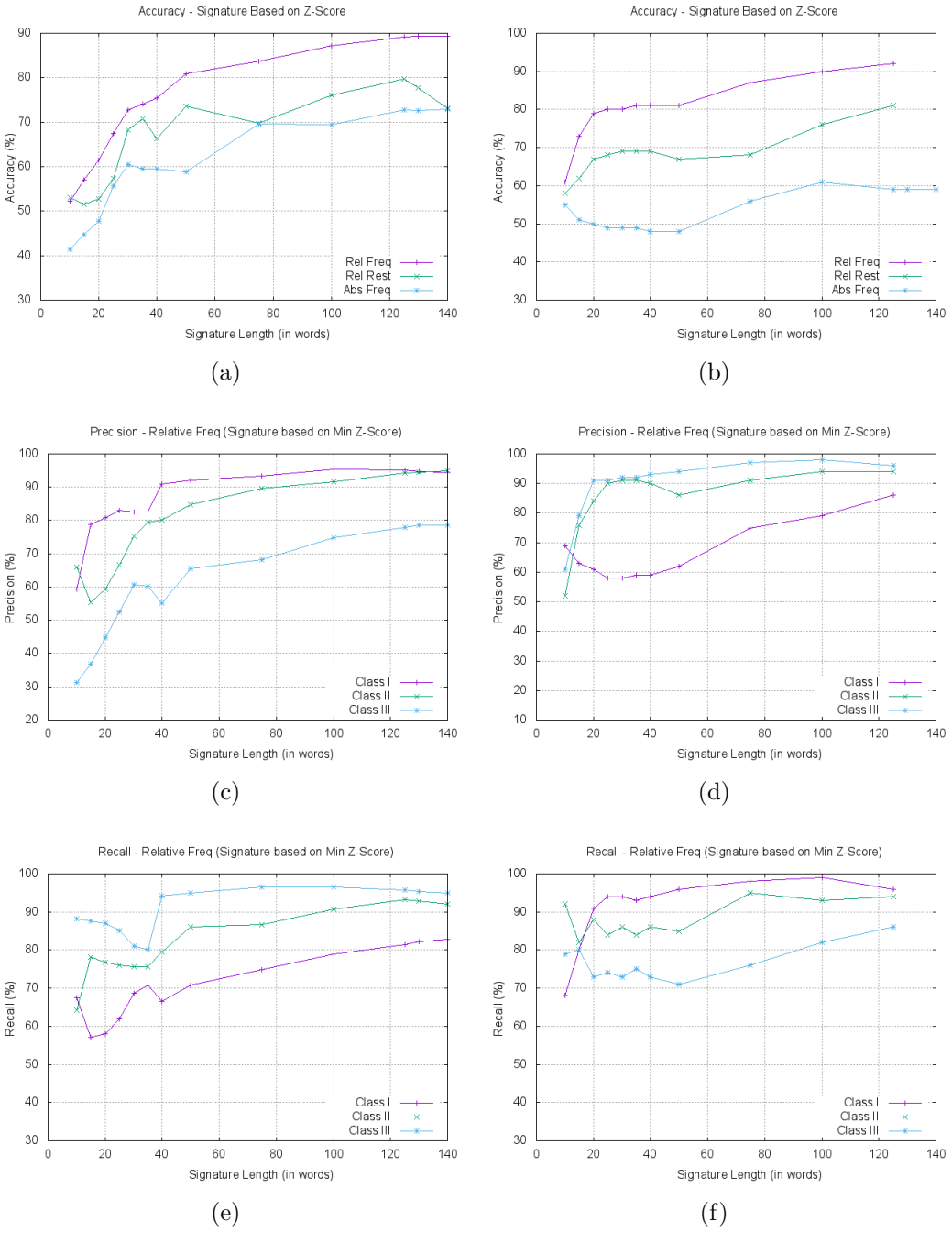

FIGURE 4.4: Comparison of accuracy, precision, recall in SC(left column) and BBC (right column) datasets using characteristic signatures based on minimum z-score (threshold $t = 50$)

The precision plots in figure 4.4 depicts precisions for z-score signatures based on RF for SC and BBC datasets. The precision plots for SC datasets are consistent with the way the accuracy has built for signatures based on RF. On the other hand, the precision plot in BBC dataset insists that the flatness in accuracy plot for RF is clearly due to the poor precision for class I. The recall represented in the figure are for signatures based on RF for both the datasets. The marks for all the three classes are consistent throughout the increase in signature length. Figure 4.5 depicts the performance of minimum z-scores based signatures using RF, RR and AF for datasets SC and BBC for a threshold of 75 words. The initial observation on increasing the threshold is that the accuracy has increased for RF and RR by almost 5% (from threshold value of 50, figure 4.4) and reached 95% at the maximum signature length for signatures based on RF in SC dataset. The accuracy for signatures based on RR has also increased considerably and peaked at $\approx 95\%$ in case of dataset SC. Ideally, we expect the signature to get access to a richer collection of features and hence pick features from a larger repository on increasing the threshold value from 50 to 75 words. This explains why the accuracy in case of SC dataset has increased by 5% in case of signatures based on RF and RR. However, we should also pay attention that having a bigger pool of features could also mean more risk of picking words that could potential increase the risk of misclassification. This is likely to have happened in case of BBC dataset in case of signatures based on RF where there is a drop in accuracy for about $\approx 7\%$. However, the signatures based on RR has increased a notch ($\approx 1\%$ to $\approx 2\%$) as expected on increasing the threshold.

4.4.2 Observation and Discussion

We have consolidated the results into two tables below. Table 4.1 highlights the results obtained from Sentence Corpus data set and Table 4.2 reflects the results obtained from BBC news dataset.

Tables 4.1 and 4.2 are structured in the same way. The leftmost column indicates the method or metric used to create the signatures namely, frequency, entropy or

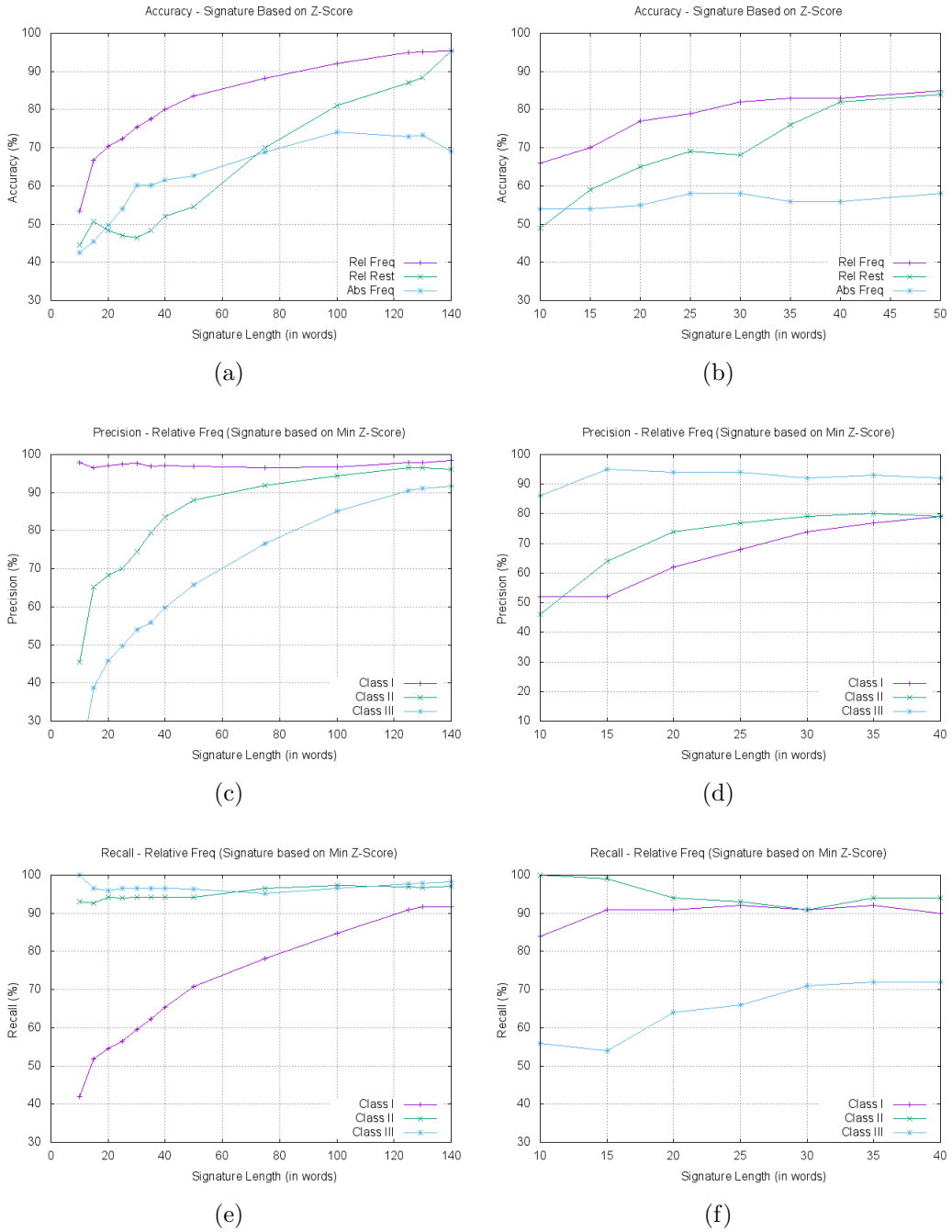

FIGURE 4.5: Comparison of accuracy, precision, recall in SC(left column) and BBC (right column) datasets using characteristic signatures based on minimum z-score (threshold $t = 75$)

Signatures based on	Property of feature	Accuracy(%) Averaged over 7 runs	S.D	Precision (class-wise)			Recall (class -wise)			t
				I	II	III	I	II	III	
Frequency	Absolute	93.62	1.51	92.29	92.86	95.71	92.76	98.06	90.77	50
	Frequency	93.62	1.51	92.29	92.86	95.71	92.76	98.06	90.34	75
	Relative	94.57	0.94	98.00	96.15	89.71	90.43	96.00	98.00	50
	Frequency	95.33	1.42	98.50	96.17	91.33	91.50	97.00	96.50	75
	Relative	92.71	2.66	94.14	89.57	94.43	89.43	98.43	92.57	50
	Rest	94.94	1.10	92.33	95.33	97.17	96.17	97.67	91.67	75
Entropy	Absolute	88.29	3.37	80.71	88.57	95.57	93.39	96.15	79.27	50
	Frequency	88.19	3.31	80.43	88.71	95.43	85.26	82.68	73.57	75
	Relative	87.95	1.79	84.29	92.86	86.71	86.29	92.43	85.43	50
	Frequency	89.29	2.72	87.86	92.57	87.43	82.74	68.32	64.69	75
	Relative	87.95	1.21	85.43	91.29	87.14	86.24	92.28	85.67	50
	Rest	88.86	2.23	83.29	89.86	92.86	82.54	69.18	71.18	75
Z-score	Absolute	73.00	4.31	46.23	79.71	93.29	96.11	90.24	58.10	50
	Frequency	70.00	6.24	48.71	61.29	97.43	96.02	96.10	53.51	75
	Relative	89.33	1.50	94.43	95.00	78.57	82.71	92.14	95.00	50
	Frequency	95.44	1.31	96.17	94.00	96.17	91.67	97.17	98.17	75
	Relative	73.14	0.62	79.86	93.86	45.71	72.23	77.95	97.29	50
	Rest	95.44	4.33	98.50	96.17	91.67	93.43	98.61	94.68	75

TABLE 4.1: Consolidated results from SC dataset

z-score. The second column is essentially the way the frequency of the features or words are captured through absolute frequency, relative frequency and relative rest. The third column denotes the average accuracy of the seven runs for each of the category formed by the combination of column first and second. The fourth column indicates the standard deviation in the computation of the average over seven runs. The following columns denotes the precision and recall values for classes I, II and III respectively. The last column in the table indicates the threshold used while creating the characteristic signatures. The general trend in all the experiments suggest that characteristic signatures based on relative frequency is more efficient than relative rest or absolute frequency in their ability to classify documents. It should further be noted that signatures created using frequency are more effective than signatures based on minimum entropy or minimum z-score when it comes to classifying documents. Further we also observed that absolute frequency also showed good ability to classify documents when used with

Signatures based on	Property of feature	Accuracy(%) Averaged over 7 runs	S.D	Precision (class-wise)			Recall (class -wise)			t
				I	II	III	I	II	III	
Frequency	Absolute Frequency	73.33	3.12	42	80	98	100	96	52	50
		73.33	3.12	42	80	98	100	91	58	75
	Relative Frequency	92.67	1.20	87	95	96	97	94	88	50
		83.67	1.50	79	80	92	90	94	72	75
	Relative Rest	86.15	1.17	65	95	98	98	85	80	50
		83.00	1.53	65	92	92	95	81	78	75
Entropy	Absolute Frequency	77.31	4.24	83	80	68	68	72	88	50
		80.00	8.30	80	77	83	81	65	74	75
	Relative Frequency	80.03	2.00	83	78	79	82	59	76	50
		81.89	2.26	80	68	95	61	63	58	75
	Relative Rest	77.56	4.35	67	70	97	93	99	59	50
		78.12	3.93	65	75	94	87	91	63	75
Z-score	Absolute Frequency	61.00	4.89	99	69	14	99	98	47	50
		60.10	4.44	99	68	15	100	94	47	75
	Relative Frequency	92.12	1.09	86	94	96	96	94	86	50
		84.33	1.74	81	80	92	90	94	72	75
	Relative Rest	80.67	0.70	60	83	99	99	96	66	50
		84.00	7.40	90	83	79	96	88	68	75

TABLE 4.2: Consolidated results from BBC dataset

signatures based on frequencies. Although in case of signatures based on Minimum entropy, absolute frequency was a notch better than relative frequency for sentence corpus dataset only. The general inference would support that characteristic signatures built on relative frequency tends to have better odds than absolute frequency and relative rest. This behavior could be attributed due to the process of normalizing frequency of each term in a document-term matrix before being used as signatures. This should remove terms that could have more occurrences in larger documents and neutralizes their ability to influence the characteristic signatures. Relative frequency is able to extract the features of a document that actually attributes to its class or category. However, relative rest apparently did not perform as expected. We assumed that the rest bit would further strengthen the ability to classify because it contains the weights of the non-significant terms in a document.

4.4.3 Comparison of Classification with SVM

We performed text classification using Support Vector Machine for both Sentence Corpus and BBC dataset. The SVM results are used as a benchmark to evaluate the performance obtained through characteristic signatures. The results obtained through SVM are shown in Table 4.3 (compares accuracy between SVM and characteristic signatures) and Table 4.4 depicts recall and precision values obtained using SVM. The approach to extract features in SVM is similar to what we have used during characteristic signatures. After filtering and removing stop words, we build a dictionary of features and transform the document in to a feature vector.

Dataset (3 Classes)	SVM Averaged over 7 runs		Characteristic Signature Analysis Accuracy(%)/ Deviation		
	Accuracy(%)	Deviation	Frequency	Min Entropy	Min Z-score
Sentence Corpus	96.43	1.77	95.33/1.42 94.57/0.94	89.29 /2.72	95.44/1.31
BBC	97.14	1.12	92 /1.20	81 /2.00	92/1.09

Table 4.3: Comparison of accuracy between SVM and characteristic signatures.

To evaluate the predictive accuracy of the model we have used linear SVM. One of the biggest advantages of SVM is it uses a subset of all the features to predict the test documents, which are also known as support vectors. We observed that the accuracy of document classification is quite decent with SVM. We ran document classification on the same samples of each dataset and calculated the average accuracy among all the test runs. The average accuracy in classification of Sentence Corpus and BBC are 96.43% and 97.14% respectively.

The above table shows comparison of the characteristic signatures against text classification using SVM. Signatures based on Frequency have in general showed better performance compared to signatures based on minimum entropy and minimum Z-score. In case of data set Sentence Corpus signatures based on Frequency and Minimum Z-score have yielded results very close to the benchmark when used with threshold value of 75. Again in case of BBC data collection, signatures based on Frequency and Z-score have yielded similar results. The main advantage of

Dataset	SVM Precision (%)			SVM Recall (%)		
	Class I	Class II	Class III	Class I	Class II	Class III
Sentence Corpus	97	98.14	94.86	96.43	97.86	95.57
BBC	97	97.14	97.71	96.14	98.00	97.71

(a)

Dataset	SVM Precision (%)			SVM Recall (%)		
	Class I	Class II	Class III	Class I	Class II	Class III
Sentence Corpus	98.50	96.17	93.33	91.50	97.50	96.50
BBC	87.15	95.00	95.89	97.15	94.00	98.00

(b)

TABLE 4.4: (a) Precision and Recall values for SVM
(b)Precision and Recall for Signature based on Frequency using relative frequency of words.

using our approach is that we can use the signatures at hand to classify text independent of the existence of number of classes. In other words, once we have characteristic signatures, we can build a model to identify documents from same classes without worrying about the total number of classes involved in a corpus. However, the number of classes present while performing classification would have significant influence in the performance of text classification using SVM.

4.4.4 Introduction of Unknown (4^{th}) Class in the Test Set

The motivation that drives the idea of document classification using characteristic signatures in the ability to form a representative definition of each class. In other words, to create a signature of the classes in training set, that would be sufficient enough to identify the existing classes even when an additional and likely unknown class is introduced in the test set. In order to establish this intention, we conducted few additional experiments and observed that characteristic signatures perform reasonably well when compared to SVM when an unknown or arbitrary class is introduced during prediction of the classes. For each dataset we created two scenarios with the idea to introduce a 4^{th} class exclusively in the test set with documents from label or class unrelated to the existing classes in the training set.

The first scenario had 4^{th} class comprising documents from one single class and the second scenario had documents from a mix of three separate classes which are unrelated to the classes in the training set. The 4^{th} class would have additional 100 test documents that is added to the existing test set. The model is trained using a similar approach that we adopted while studying the behavior of characteristic signatures, i.e. the classifier is learned based on the precious three known classes in the training set. Under scenario one the 4^{th} class for sentence corpus contained documents from entertainment label from BBC dataset. Similarly, the 4^{th} class in BBC comprised of randomly selected 100 test documents of computational biology (plos) label from sentence corpus. For the second scenario which should have the 4^{th} class as a mix of three separate labels, we randomly selected 100 documents altogether from three classes from sentence corpus and introduced it in BBC dataset as 4^{th} class and vice versa. It should be noted that we created seven samples of the 4^{th} class for both the scenario for each dataset. The intention is to allow the classifiers trainied on training set containing three classes or labels, to predict the test documents from the additional 4^{th} class which dont fit in the existing classes in the training set. Our aim is to study the performance of characteristic signatures and SVM in their ability to successfully classify the existing three classes and to eliminate the additional 4^{th} class. In order to implement this method we introduce an upper threshold on the KL-divergence measure that is used to determine the closeness or similarity of the test documents to the signatures of each class obtained from the training set. In order for a document in the test set to be classified in to one of the classes it should be less than the threshold fixed for the KL-divergence. Documents having a measure of KL-divergence greater than the threshold limit from any class signatures would be discarded. In this experiment we will observe whether characteristic signature is able to eliminate the documents from the additional 4^{th} class and still able to classify classes I, II and III. For the sake of a fair comparison we have chosen signatures that have proven to be competitive against document classification using SVM for both the datasets. Therefore, in case of Sentence Corpus dataset we decided to use signatures based on frequency using relative frequency and absolute frequency of words.

The value of KL-divergence threshold in this series of experiments vary from one another. The KL-divergence threshold is obtained using the distribution of min KL-divergence distance for each of the test document from the classes. We noticed that while using relative frequency of the words, the KL-divergence threshold is in the order of ≈ 6.1 to ≈ 7 for Sentence Corpus and BBC. However, the threshold is in order of ≈ 0.75 to ≈ 0.9 in case of absolute frequency of features or words. The below table depicts the results obtained from this test.

Dataset (4 Classes)	SVM Averaged over 7 runs		Characteristic signature using frequency	
	Accuracy(%)	Deviation	Absolute Frequency of words Accuracy(%) /Deviation	Relative Frequency of words Accuracy(%)/Deviation
Sentence Corpus	70.50	1.97	85.32 / 1.69	88.88 / 2.43

(a)

Dataset (4 Classes)	SVM Averaged over 7 runs						Characteristic signature based on frequency using relative frequency of words (Averaged over 7 runs)					
	Precision(%)			Recall(%)			Precision(%)			Recall(%)		
	I	II	III	I	II	III	I	II	III	I	II	III
Sentence Corpus	73.50	63.18	80.50	92.35	97.00	94.71	94.14	97.00	96.50	95.43	98.08	96.35

(b)

TABLE 4.5: Comparison of prediction of test documents between SVM and characteristic signatures after addition of class-4 in test collection on Sentence Corpus dataset. (a) Comparison of accuracy between SVM and Signatures based on Frequency using relative frequency of words and absolute frequency. (b) Comparison of Precision and Recall between SVM and Signature based on Frequency of relative frequency of words.

Table 4.5.1(a) clearly depicts that after introduction of class-4 (scenario 1) in to the Sentence Corpus test collection, the SVM model originally learned on 3 classes is unable to classify the 4 classes since SVM classifier has no knowledge about the additional class. The accuracy of SVM has dropped from $\approx 96\%$ to 70.50% and is $\approx 15\%$ less than the accuracy obtained using absolute frequency and $\approx 18\%$ lower than the accuracy obtained by relative frequency. We also observe that the

Precision for SVM has dropped significantly due to inability to classify the class-4 documents. However, due to the introduction of KL-divergence threshold the precision has been decent in case of characteristic signatures. Similar experiments are conducted on BBC dataset and results are shown in the table below.

Dataset (4 Classes)	SVM (Averaged over 7 runs)		Characteristic signature using frequency	
	Accuracy(%)	Deviation	Absolute Frequency of words Accuracy(%) /Deviation	Relative Frequency of words Accuracy(%)/Deviation
BBC	71.51	2.12	88.04 / 3.13	89.38 / 1.21

(a)

Dataset (4 Classes)	SVM (Averaged over 7 runs)						Characteristic signature using relative frequency of words (Averaged over 7 runs)					
							Z-Score					
	Precision(%)			Recall(%)			Precision(%)			Recall(%)		
Classes	I	II	III	I	II	III	I	II	III	I	II	III
BBC	73.20	63.50	80.00	92.05	97.35	94.84	93.05	94.50	87.08	92.62	97.00	90.21

(b)

TABLE 4.6: Comparison of prediction of test documents between SVM and Characteristic signatures after addition of class-4 in test collection on BBC dataset. (a) Comparison of accuracy between SVM and Signatures based on Frequency and Z-score using relative frequency of the words. (b) Comparison of Precision and Recall between SVM and Signature based on Z-score using relative frequency of words.

Table 4.6 shows results of addition of class-4 (scenario 1) in the BBC test collection and the performance of SVM and characteristic signatures. It is imperative that the accuracy of SVM has dropped significantly to 71.51% from $\approx$ 97% which is almost 26.50% less than the original accuracy obtained on 3 classes. On the other hand, signatures based on Z-scores and Frequency have yielded decent accuracy ($\approx$ 89% and $\approx$ 88% respectively) using relative frequency of words. Table 4.6(b) clearly depicts that the precision values for SVM had a significant drop whereas the precision values for signatures based on Z-score were significant. The results obtained from introducing 4^{th} class following scenario 2 has shown similar trend as found in scenario 1. Table 4.7 (a) depicts the results achieved for SVM and characteristic signatures on sentence corpus dataset. SVM managed to achieve

an average accuracy of 72.46% over seven runs with a standard deviation of 1.33. Since SVM classifier relies on small set of support vectors to classify documents the test documents, it is unable to predict the additional 4^{th} class from the existing classes I, II and III.

Dataset (4 Classes*)	SVM Averaged over 7 runs		Characteristic signature using frequency (Averaged over 7 runs)	
	Accuracy(%)	Deviation	Absolute Frequency of words Accuracy(%) /Deviation	Relative Frequency of words Accuracy(%)/Deviation
Sentence Corpus	72.46	1.33	88.21 / 2.23	88.50 / 3.10

(a)

Dataset (4 Classes*)	SVM Averaged over 7 runs						Characteristic signature based on frequency using relative frequency of words (Averaged over 7 runs)					
	Precision(%)			Recall(%)			Precision(%)			Recall(%)		
	I	II	III	I	II	III	I	II	III	I	II	III
Sentence Corpus	83.14	60.69	81.20	97.00	97.00	96.32	98.11	94.67	86.63	85.25	95.05	97.10

(b)

* 4^{th} class comprises of a mix of three separate classes different from training classes as mentioned in scenario 2.

TABLE 4.7: Comparison of prediction of test documents between SVM and Characteristic signatures after addition of class-4 following scenario 2 for Sentence Corpus dataset. (a) Comparison of accuracy between SVM and Signatures based on Frequency using relative frequency of words and absolute frequency. (b) Comparison of Precision and Recall between SVM and Signature based on Frequency of relative frequency of words.

We conducted this study with the second scenario with BBC new dataset (shown in table 4.8). In case of BBC dataset SVM attained an average accuracy of 72.90% over seven runs with a standard deviation of 0.68. On the other hand, signatures based on frequency using relative frequency of words reached an average accuracy of 87.70% over seven runs with a standard deviation of 0.68. This clearly is as an indication that the representative nature of the signature is helpful to identify and eliminate documents from labels which do not fit in the training set.

Data Set (4 Classes*)	SVM Averaged over 7 runs		Characteristic Signature (Averaged over 7 runs)	
			Frequency using relative frequency	Z-Score using relative frequency
	Accuracy(%)	Deviation	Accuracy(%)/ Deviation	Accuracy(%)/ Deviation
BBC	72.90	0.68	87.70/1.29	84.63 /1.50

(a)

Dataset (4 Classes*)	SVM Averaged over 7 runs						Characteristic signature based on frequency using relative frequency of words (Averaged over 7 runs)					
	Precision(%)			Recall(%)			Precision(%)			Recall(%)		
	I	II	III	I	II	III	I	II	III	I	II	III
BBC	90.42	86.57	54.43	95.54	97.50	97.85	90.17	95.50	94.83	95.20	94.20	90.80

(b)

* 4^{th} class comprises of a mix of three separate classes different from training classes as mentioned in scenario 2.

TABLE 4.8: Comparison of prediction of test documents between SVM and Characteristic signatures after addition of class-4 following scenario 2 for BBC dataset. (a) Comparison of accuracy between SVM and Signatures based on Frequency using relative frequency of words and absolute frequency. (b) Comparison of Precision and Recall between SVM and Signature based on Frequency of relative frequency of words.

The two scenarios included in this experiment was to strengthen the proposition that signatures can work relative well in situations where a random test document appeared which doesnt relate to the labels or classes of the training set. We showed that while adding the 4th class we made sure to create it using a single unrelated class as well as a mix of unrelated classes with respect to the training classes or labels. The results obtained explain that signatures are very much representative of their respective classes and are able to uniquely define each class. In other words, signatures contain information that helps to identify and define each class uniquely. SVM on the other hand uses a subset of data points or support vectors that is used to create hyperplanes that separate classes from one another based on training data set. Therefore, SVM is incompetent when it comes to handling the unknown classes introduced in the test set after training the model. SVM tries to fit the unknown class in to one the known 3 classes in the best possible manner. However, due to the characteristic nature of the signatures, they are able to fairly identify whether a document belongs to a particular class or not. The

KL-divergence threshold played an important role in the elimination of class-4 documents. This explains the high precision values exhibited by signatures. As we have observed in the previous sections that SVM is an effective tool to classify and has revealed high accuracy while predicting class I, II and III. The low accuracy of SVM in this experiment is solely attributed to the fact that documents from 4th class were classified in to one of three known classes.

4.4.5 Performance of signatures with overlapping classes

In the previous section we have illustrated how signatures are effective to identify unknown class labels in the test collection. Signatures have proven to be an effective technique to classify text documents when the class labels belong to distinct categories. In this section we will discuss the performance of signatures when the class labels are not so distinct and are similar in nature. Signatures behave like prototypes of a class and are useful to identify classes which are distinct. However, to illustrate their behavior for class labels which are similar and can have significant overlap, we have considered to use BBC dataset with class labels entertainment, politics and business. The intention was to have two classes which are inter-related and belongs to a domain which are related. In Table 4.9(a) a quick comparison is shown between SVM and signature based text classification where out of 3 classes two of the classes are similar and not so distinct. Due to the explicit nature of signatures, they are not expected to perform better if the features involved in two or more class labels have an significant underlying overlap. In such a scenario, signature based on frequency using absolute frequency of words yields an accuracy of $\approx 68\%$ and signatures based frequency using relative frequency of words achieved an accuracy of $\approx 88\%$. On the other hand, SVM performed better than signatures with an accuracy of $\approx 95\%$.

The closeness between two of the classes did not affect SVM in classification, however, signatures using absolute frequency of words were greatly affected and is reflected in its accuracy. However, signatures based on relative frequency of words achieved a decent accuracy which further reinforces the fact that relative

Data Set (3 Classes*)	SVM Averaged over 7 runs		Characteristic Signature (Averaged over 7 runs)	
			Frequency using absolute frequency	Frequency using relative frequency
	Accuracy(%)	Deviation	Accuracy(%)/ Deviation	Accuracy(%)/ Deviation
BBC	95.50	1.68	68.70/3.10	84.04 /1.27

(a)

Data Set (4 Classes)	SVM Averaged over 7 runs		Characteristic Signature (Averaged over 7 runs)	
			Frequency using absolute frequency	Frequency using relative frequency
	Accuracy(%)	Deviation	Accuracy(%)/ Deviation	Accuracy(%)/ Deviation
BBC	91.75	1.93	70.85/4.86	88.50 /2.06

(b)

* Using 2 overlapping class labels out of the 3 classes.

TABLE 4.9: (a) Classifying 3 classes with 2 overlapping classes in BBC dataset. (b) Classifying 4 classes in BBC dataset

frequency is an useful scheme to generate signatures. In this final section of our study we have also attempted to classify four classes using signature and compare the same with SVM as shown in Table 4.9(b). We have used signatures based on frequency using both absolute and relative frequency of words to classify 4 classes. Signatures built on absolute frequency and relative frequency achieved an accuracy 70.85% and 88.04% respectively. It is further interesting to note that relative frequency of words are consistently a better performer even when signatures using absolute frequency fails to perform. Therefore, we have tried to explore the ability of signatures on various settings.

Table 4.10 depicts a consolidated performance of signatures and SVM over the various analysis and observations conducted on BBC and Sentence Corpus dataset. The results are obtained from signatures based on frequency using relative frequency of words. The table clearly indicates the strength and weakness of our methods. Signatures are not as useful as SVM when the classes are not distinctive and have a reasonable overlap. However, signatures out perform SVM when it comes to preventing classification of labels that do not occur in the training set. The results obtained from signatures are encouraging but this also leaves us with

Classification (BBC)	Accuracy(%) / Std Dev	
	Signatures	SVM
3 classes	92.67 / 1.20	97.14 / 1.12
3 classes (2 non-distinct classes)	84.04 / 1.27	95.50 / 1.68
4 classes	88.50 / 2.06	91.75 / 1.93
Introduce 4th class in test set (scenario 1)	88.50 / 3.10	71.51 / 2.12
Introduce 4th class in test set (scenario 2)	87.70 / 1.29	72.90 / 0.68

(a)

Classification (Sentence Corpus)	Accuracy(%) / Std Dev	
	Signatures	SVM
3 classes	95.33 / 1.42	96.43 / 1.77
Introduce 4th class in test set (scenario 1)	88.88 / 2.43	70.50 / 1.95
Introduce 4th class in test set (scenario 2)	88.50 / 3.10	72.46 / 1.33

(b)

TABLE 4.10: (a) Classifying 3 classes with 2 overlapping classes in BBC dataset. (b) Classifying 4 classes in BBC dataset

further avenue to improve and enhance the performance and eventually scope for future work.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This work suggests that characteristic signature is an effective method of document classification. We proposed three approaches to develop characteristic signatures namely, frequency, entropy and z-score. Furthermore, we used three different properties of the features or words to create the characteristic signatures. The advantage of using signature is that signatures are inherently representative of their classes and thus, will remain same even if new classes are introduced to the corpus, i.e. our approach is able to classify documents in to respective classes irrespective of whether new classes are added to the corpus or not. We observed that our model was able to classify the existing three classes with decent efficiency even after a 4^{th} class was introduced. As discussed in section 4.4.4 document signatures could be very effective if there are one or more random classes being introduced in the test set after the model is trained on a training set with lesser number of labels or classes. Due to the representative nature of signatures, they are able to identify classes from training set efficiently compared to SVM. Looking into document classification ability, we observed that for every kind of signatures, relative frequency of the features seem to have better results while classifying documents into one of the classes. Specifically for signatures based on frequency, relative

frequency of the features or words has turned out to be the best metric in terms of accuracy for classifying documents. They have been consistent in their ability to classify documents in both the Sentence Corpus and BBC datasets. We can conclude that relative frequency of words has effectively more useful information about the identifying features and their weights to identify a document in to a class. Absolute frequency of words has also performed decently well, however, it is influenced by erratic length of the documents. Longer document are more prone to have words with higher frequency and thus influencing the weights of the features. Interestingly, signatures based on Z-score have performed better under certain categories of threshold and properties used for the features. The performance in document classification using characteristic signatures vary depending upon the length of the signatures used and the threshold used in the selection of the features. Our Algorithm is easily adaptable and by creating a prototype of this approach is a quick way to study the performance of various signatures of different lengths on a specific data collection.

5.2 Future Work

This study shows the potential of characteristic signatures for document classification which leaves a lot of scope for new ideas to be implemented and validated against existing benchmarks. Characteristic signatures could be built using other properties or features of the document collection in addition to words e.g. relations between various words or named entities. This model could be used or implemented in a generic recommendation system for documents and evaluate the performance of such a system and validate the same against a benchmark.

Bibliography

[1] D. Soergel. *Organizing information: Principles of data base and retrieval systems.* Orlando, FL, Academic Pres, 1985.

[2] F. W. Lancaster. *Indexing and abstracting in theory and practice.* London Facet Publishing, 3 edition, 2003.

[3] J. Aitchison. A classification as a source for thesaurus: The bibliographic classification of h. e. bliss as a source of thesaurus terms and structure. *Journal of Documentation*, 42(3):160–181, 1986.

[4] G. Salton. Automatic Information Organization and Retrieval. Technical report, McGraw Hill Text, 1968.

[5] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, pages 1–47, 2002.

[6] K. Mertsalov and M. McCreary. Document classification with support vector machines. *ACM Computing Surveys (CSUR)*, pages 1–47, 2009.

[7] M. Santini and M. Rosso. Testing a genre enabled application: A preliminary assessment. In *In Proc. of the 2nd BCS-IRSG Symposium onFuture Directions in Information Access*, pages 54–63, 2008.

[8] D. Greene and P. Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. *Proc. ICML*, 2006.

[9] A. Chambers. Statistical models for text classification: Applications and analysis. *University of California, Irvine*, 2013.

[10] C. Boulis and M. Ostendorf. Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. *Workshop on Feature Selection in Data Mining, in conjunction with SIAM conference on Data Mining*, 2005.

[11] C. Lee and G. G. Lee. Mmr-based feature selection for text categorization. *Proceedings of the Human Language Technologies. NAACL*, pages 5–8, 2008.

[12] Y. Yang and X. Liu. A re-examination of text categorization methods. *In Proceedings of the 22nd annual international conference on research and development in information retrieval (SIGIR)*, pages 42–49, 1999. ISSN 0916-8532.

[13] A. Y. Ng D. M. Blei and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, (3):993–1022, 2003.

[14] T. Joachims. A statistical learning model of text classification with support vector machines. In *Proceedings of the 24th ACM international conference on research and development in information retrieval (SIGIR)*, pages 128–136, 2001.

[15] T. Jebara. *Machine learning: Discriminative and Generative.* The Springer International Series in Engineering and Computer Science, 2004.

[16] D. J. Lipman and W.R. Pearson. Rapid and Sensitive Protein Similarity Searches. *American Association for the Advancement of Science*, (227):1435–1441, 1985.

[17] S. Christodoulakis. F. Ho and M. Theodoridou. The Multimedia Object Presentation Manager in MINOS: A Symmetric Approach. *Proc. ACM SIGMOD*, 1986.

[18] M. Warmuth J. Kivinen and P. Auer. The perceptron algorithm vs. winnow: Linear vs. logarithmic mistake bounds when few input variables are relevant. *In Conference on Computational Learning Theory*, 1995.

[19] T. Joachim. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Technical report, LS VIII Number 23, University of Dortmund*, 1997.

[20] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice.* Wiley, 2009.

[21] S. Vijayarani and A. Sakila. Template matching technique for searching words in document images. *International Journal on Cybernetics and Informatics (IJCI)*, (6), 2015.

[22] Q. Yu Y. Wang and W. Yu. An Improved Normalized Cross Correlation algorithm for SAR Image Registration. *IEEE IGARSS*, 2012.

[23] K. J. Mayer W. Krattenthaler and M. Zeiler. Point correlation: A reduced-cost template matching technique. *In: Proceedings of the first IEEE International Conference on Image Processing*, pages 208–212, 1994.

[24] J. Sivic. Efficient visual search of videos cast as text retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4):591–605, 2009.

[25] A. Rajaraman and J. D. Ullman. *Data Mining". Mining of Massive Datasets.* 2011.

[26] D. Pyle. *Data Preparation for Data Mining.* Morgan Kaufmann Publishers, Los Altos, California, 1999.

[27] M.F. Porter. An algorithm for suffix stripping. *Program 14.3*, pages 130–137, 1980.