Chapter - 4

CURSE OF DIMENSIONALITY

4.1 Introduction

In text categorization (TC), one of the biggest problems a classifier faces is the high dimensionality of the input space since every document is represented by terms in all documents in the whole corpus. Therefore the number of dimensions has to be scaled down substantially otherwise it would become (usually thousands or tens of thousands) a big nightmare for the learning algorithms. This is known as the curse of dimensionality, which means that the complexity of the problem increases exponentially with respect to the dimension of the problem, thus the learning itself incurs substantial computation. Therefore, prior to inductive learning, researchers often applies a pass of dimensionality reduction techniques which helps to reduce the dimensionality of term space, thereby improving the learning process.

Another important contribution of this chapter is to investigate the detailed analysis of feature discriminating power across the traditional methods and propose a novel feature selection method based on Dempster-Shafer theory of evidence. Thus, this chapter provides a partial answer to the question raised in this thesis with a theoretical explanation, i.e. “How can one propose a novel effective feature selection method by using the important prior information given by the training data set?”

This chapter is organized as follows. Section 4.2 describes the methods for representing text including the prominent vector space model for TC. Section 4.3 introduces various most common pr-processing tasks for text categorization. Section 4.4 gives an overview of various traditional feature selection proposed in the literature and gives an insight into each such methods. Section 4.5 reviews some of the term weighting borrowed from IR. In Section 4.6, it proposes a novel evident theoretic feature selection method based on the theory of evidence.
4.2 Text Representation

Most of the textual information stored in documents are unstructured by nature. Though documents are in machine readable format, it is generally not feasible to feed documents in its inherent form directly to learning algorithms in most cases. Therefore, prior to applying an inductive machine learning method to text categorization, the whole textual corpus must be transformed into a compact representation in order to be processed and categorized by the inductive learning classifiers. The most straight forward technique in IR to represent the document through the word it contains is vector space model (VSM). In VSM, each document is converted into a multi-dimensional vector in the term space, $\vec{d}_j = (w_{1j}, \ldots w_{Tj})$, where $|T|$ denotes the number of distinct terms. The contribution of a specific term to the document $d_j$ is represented by the value of $w_{kj}$ between (0, 1). Figure 4.1 shows an example of documents represented in VSM. In this three term dimensions, namely, Opec, Stake and Deal, there are three documents represented as three vectors in this 3-dimensional space. The number of dimensions of this term space is determined by the all informative words inferred from entire corpus. This model assume an implicit assumption that documents that are “close together" in term space are similar in semantic.

![Figure 4.1: Representation of documents in a vector space model](image-url)
Generally, there are two issues involved in text representation: a) What should a term be? How to select a relevant term? b) How to compute term weights? The first issue is the theme of Section 4.4. The second issue will be discussed in Section 4.5.

4.3 Pre-processing of Textual Documents

Text pre-processing is a most inevitable task in text categorization in which a text corpus is subjected to many feature engineering process such as removal of punctuation and special characters, common words and rare words. The idea behind this task is not only to reduce the unnecessary noise in the collection but also to improve the classification and computational efficiency of text classifier. Some of the commonly performed pre-processing steps are discussed below:

4.3.1 Word Distribution

As discussed in Section 2.4, word occurrences in natural language follows the Zipf distribution. It states that the frequency of any word occurrence is inversely proportional to its rank in the frequency table. Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc. Therefore it may be helpful in removing of such rare words which occurs two or fewer times as well as it may lead to reduce the feature space substantially. However, one must keep in mind that in a highly skewed datasets, there is a possibility that some potential feature may be removed by this process [For03].

4.3.2 Removal of Stop Words

It has been observed from prior research work that the term which occurs more frequently like the one in the below example is less likely to cater any sort of significance to the classifier. Thus it may be removed from the controlled dictionary to speedup processing and save space. These very frequent and uninformative words are known as stop words. Most of the time, these words act like a topic-neutral words such as articles, prepositions, conjunctions, etc (such as, “the”, “a”, “of” and so on). This is widely done by using a "stop word list" which is given in Appendix A.

However, this approach suffers from being a language specific and domain specific choice. Alternatively, a threshold on the number of documents in which the word occurs could be specified e.g. words that occur in more than half of the documents may be
considered as stop words [Fab02]. However, this choice may be harmful to highly-skewed datasets. In such datasets, one or two classes may include a large percentage of documents. Therefore, deleting stop words based on threshold may remove the discriminative features that distinguish these categories.

4.3.3 Stemming

In addition to the mundane pre-processing tasks, stemming is another significant step towards analysing and finding of morphological root of a word in the document collection such as discarding non-alphabetic characters and mark-up tags, case-folding, and stop words elimination. Stemmer analysis in text processing is the main perspective of the morphological concern with text classification.

One can form a group of words which are equivalent on the basis of similar semantic interpretation. For example, the occurrence of two distinct terms “credibility” and “credentials” in a document could be considered as equivalent since these two terms almost conveying the same semantic interpretation. Stemming could be understood with the objective of extracting the same morphological root word. As a result, the variation arises according to multiple occurrences of the same word in many grammatical forms could be eliminated.

With regard to this, a number of stemming algorithms, or stemmers, have been reported in the literature to map different morphological forms of terms to a common feature. Thus, it reduces the length of individual features by avoiding the original words rather documents are represented by stem. However application of stemming is quite controversial in TC. Though it has been widely used, but some reports suggest that it hurts effectiveness of the classifier [BM98]. However, the recent trend is to include this, since it decreases both the dimensionality of the feature space and the stochastic dependence between features, which results in a saving of storage space and processing time. Porter stemmer is the most common algorithm for stemming in English [Por80].

Silvatt and Ribeiroot conducted study on comparing the performance of combinations of feature engineering choices [SR03]. Their study concluded that elimination of stop word is a significant choice since it affected the performance considerably. They also showed that stemming also play a role on affecting the performance slightly when they removed some words which has literally no influence on the performance. Similarly, an extensive study
conducted by Song et al. concluded that elimination of stop word is neither helpful nor harmful to text categorization [SLY05]. Moreover, they also observed that stemming may harm the performance to some extent. However, they encouraged the adaption of stop word removal and stemming while considering the merits on reducing the term space significantly.

4.4 Features Selection

Generally constructing an inductive text classifier involves two key sub tasks, they are feature selection and classifier construction. In this Section, it discusses the methods to improve the performance of TC from the feature selection aspect. The issue here is that the feature can be at different level such as syllable, word, phrase, and other sophisticated semantic and/or syntactic representations by exploiting natural language processing knowledge. Each feature has unique importance in a text with respect to its semantic and context. Thus to influence the particular feature towards the semantic of the document, the researcher may use term weighting methods to assign appropriate weights to the features to improve the classification performance of TC.

One important goal of this Section is to study various feature selection and feature weighting methods for TC given the bag-of-words approach and also investigate the pros and cons of each feature selection methods. Although the most promising method for feature weighting is $tf.idf$ borrowed from information retrieval, recently there has been progress reported on improving feature discriminating power by exploiting information theory and probabilistic theory. Obviously, one may expect that the prior knowledge inferred from information theoretic methods with conditional probabilities of a term and category should play an important role in improving terms discriminating power for semantics of documents. However, many of them have not shown better result in terms of performance, some of them have shown remarkably better results in some cases. Moreover, the enhancement over these existing methods did not achieve the desired level that one might hope when using such rich information sources.

In Machine Learning, the researchers usually adapt a technique in order to curse the dimensionality of the term space is known as Feature selection in which a subset of the features selectively chosen form all available words in the whole corpus. Feature Selection techniques typically incorporate a search strategy for exploring the space of feature subsets, including methods for determining a suitable starting point and generating successive
candidate subsets, and an evaluation criterion to rate and compare the candidate sets, which serves to guide the search process. This evaluation schemes can be divided into two broad categories: \textit{filter} approach and \textit{wrapper} approach.

\textbf{4.4.1 Filter Methods}

Filter techniques attempt to remove irrelevant features from the feature set prior to the application of the learning algorithm. Initially, the data is analysed to identify those dimensions that are most relevant for describing its structure. The chosen feature subset is subsequently used to train the learning algorithm. Feedback regarding an algorithms performance is not required during the selection process, though it may be useful when attempting to gauge the effectiveness of the filter.

It is worth mentioning at this point that some classifications techniques perform implicit feature selection. For instance the process of building a decision tree will very often not select all the features for use in the tree. Features not used in the tree have no role then in classification. The key issue in the Filter strategy for feature selection is the criterion used to score the predictiveness of the features.

In recent years, \textit{Information Gain} (IG) has become perhaps the most popular criterion for feature selection. The Information Gain of a feature is a measure of the amount of information that a feature brings to the training set [PD07]. Information gain [YP97] is commonly used as a surrogate for approximating a conditional distribution for text classification. In information gain, class membership and the presence/absence of a particular term in a given category are seen as random variables; one computes how much information about the class membership is gained by knowing the presence/absence statistics. If the class membership is interpreted as a random variable $c_i$ with two values, positive ($c_i$) and negative ($\overline{c}_i$), and a word is likewise seen as a random variable $t$ with two values, present ($t_k$) and absent ($\overline{t}_k$).

For comparison purposes, this work will also consider Odds Ratio (OR) [YP97] which is an alternative filtering criterion. The basic idea of using odds ratio is to calculate the odds of a term occurring in the positive class (the category a term is related to) normalized by the odds of that term occurring in the negative class (the category a term is not related to). The odds ratio of a term $t_k$ for a category $c_i$ is defined using Table 4.1. Odds ratio is known to work well with the naïve Bayes text-classifier algorithm. For binary classification OR
calculates the ratio of the odds of a feature occurring in the class to the odds of the feature occurring in the non-class. Where a specific feature does not occur in a class, it can be assigned a small fixed value so that the OR can still be calculated. For feature selection, the features can be ranked according to their OR with high values indicating features that are very predictive of the class. The same can be done for the non-class to highlight features that are predictive of the non-class.

While IG is an effective strategy for feature selection it is criticised in the sense that features are considered in isolation so redundancies or dependencies are ignored. Two strongly correlated features may both have high IG scores but one may be redundant once the other is selected. More sophisticated Filter techniques that address these issues using Mutual Information to score groups of features have been researched by Novovićov’a et al. [NMP04] and have been shown to be more effective than these simple Filter techniques.

In Statistics, one often use \( \chi^2 \) test to test the independence between two random variables. This intuition is now adapted in text classification where \( \chi^2 \) [YP97] is used to measure the independence between a category and feature term. This \( \chi^2 \) measure of a term \( t_k \) for a category \( c_i \) is defined as shown in Table: 4.1. The \( \chi^2 (t_k, c_i) \) score denotes the weight of term \( t_k \) with respect to category \( c_i \). When a term is associated to more than one categories, then the score of that term is higher.

Bi-Normal Separation (BNS) was proposed by [For03] in order to select relevant and irrelevant features using inverse cumulative probability function. Forman observed that positive as well as negative feature play an important role in classification process. It is defined as:

\[
BNS(t_k, C_i) = \left| F^{-1}(p(t_k, C_i)) - F^{-1}(p(t_k, \overline{C_i})) \right| \tag{4.1}
\]

Where

- \( t_k \) - a term in a document vector
- \( C_i \) – a category \( i \)

\( F^{-1} \) - standard Normal distribution inverse cumulative probability function
Many of the metric listed in Table 4.1 have an implicit assumption that the discriminative feature for the given category $c_i$ are spread across the domain of both positive as well as negative instance of $c_i$ except few methods. This doesn’t mean that the intuitive principle behind every method is same. Most of the experiments conducted on the listed methods, conclude that the document frequency is well doing over others [YP97]. Informally, the conducted experiments seem to suggest that \{odds ratio, GSS\} > \{χ², information gain\} > \{mutual information\}, where the symbol > means “performs better than”. However, the outcome of these experiments are just indicative since any statement regarding the relative merits of these function requires comparative experiments performed in thoroughly controlled conditions and on a variety of different situations (e.g. different classifiers, different initial corpora, etc.). In addition to that, feature selection is inevitable step in text categorization since it is neither practically and computationally infeasible to use all available features, nor because of estimation of class labels when limited training dataset samples but with a large number of features.

Table 4.1. Traditional feature selection methods used in Text Categorization

<table>
<thead>
<tr>
<th>Function</th>
<th>Denoted by</th>
<th>Mathematical Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document Frequency</td>
<td>#(tk,ci)</td>
<td>$P(t_k/c_i)$</td>
</tr>
<tr>
<td>Information Gain</td>
<td>IG(tk,ci)</td>
<td>$P(t_k,c_i).\log\frac{P(t_k,c_i)}{P(t_k).P(c_i)} + P(\bar{t}_k,c_i).\log\frac{P(\bar{t}_k,c_i)}{P(\bar{t}_k).P(c_i)}.$</td>
</tr>
<tr>
<td>Mutual Information</td>
<td>MI(tk,ci)</td>
<td>$\log\frac{P(t_k,c_i)}{P(t_k).P(c_i)}$</td>
</tr>
<tr>
<td>Chi Square</td>
<td>$X^2(tk,ci)$</td>
<td>$g[P(t_k,c_i)P(\bar{t}_k,\bar{c}_i) - P(t_k,\bar{c}_i)P(\bar{t}_k,c_i)]^2 P(t_k).P(c_i).P(\bar{t}_k).P(\bar{c}_i)$</td>
</tr>
<tr>
<td>Correlation Coefficient</td>
<td>CC(tk,ci)</td>
<td>$\sqrt{g[P(t_k,c_i)P(\bar{t}_k,\bar{c}_i) - P(t_k,\bar{c}_i)P(\bar{t}_k,c_i)]} \sqrt{P(t_k).P(c_i).P(\bar{t}_k).P(\bar{c}_i)}$</td>
</tr>
<tr>
<td>Relevancy Score</td>
<td>RS(tk,ci)</td>
<td>$\log\frac{P(t_k/c_i) + d}{P(t_k/c_i) + d}$</td>
</tr>
<tr>
<td>Odd Ratio</td>
<td>OR(tk,ci)</td>
<td>$\frac{P(t_k/c_i).(1 - P(t_k/\bar{c}_i))}{(1 - P(t_k/c_i)).P(t_k/\bar{c}_i)}$</td>
</tr>
<tr>
<td>Simplified Chi-Square</td>
<td>sX²(tk,ci)</td>
<td>$P(t_k,c_i)P(\bar{t}_k,\bar{c}_i) - P(t_k,\bar{c}_i)P(\bar{t}_k,c_i)$</td>
</tr>
</tbody>
</table>
4.4.2. Wrapper Methods

Wrapper methods for feature selection make use of the learning algorithm itself to choose a set of relevant features. The wrapper conducts a search through the feature space, evaluating candidate feature subsets by estimating the predictive accuracy of the classifier built on that subset. The goal of the search is to find the subset that maximizes this criterion.

The obvious criticism of the Filter approach to feature selection is that the filter criterion is separate from the induction algorithm used in the classifier. This is overcome in the Wrapper approach by using the performance of the classifier to guide search in feature selection – the classifier is wrapped in the feature selection process. In this way, the merit of a feature subset is the generalisation accuracy it offers as estimated using cross-validation on the training data. If 10-fold cross validation is used then 10 classifiers will be built and tested for each feature subset evaluated – so the wrapper strategy is very computationally expensive. If there are \( p \) features under consideration then the search space is of size \( 2^p \) so it is an exponential search problem. For large values of \( p \) an exhaustive search is not practical because of the exponential nature of the search. The two most popular strategies are:

- **Forward Selection** which starts with no features selected, evaluates all the options with just one feature, selects the best of these and considers the options with that feature plus one other, etc.
- **Backward Elimination** starts with all features selected, considers the options with one feature deleted, selects the best of these and continues to eliminate features.

These strategies will terminate when adding (or deleting) a feature will not produce an improvement in classification accuracy as assessed by cross validation. Both of these are greedy search strategies and so are not guaranteed to discover the best feature subset. More sophisticated search strategies can be employed to better explore the search space; however, Reunanen [Reu03] cautions that more intensive search strategies are more likely to over-fit the training data.

4.5 Feature Weighting

In text categorization task, how much a term contributing towards semantic of the document is more important than the term type adapted in representing the document as not all term type are good at all situations. The selected features either by feature selection or
feature extraction receive the highest scores according to a criterion that measures the significance of the feature (term) for text categorization task. The criteria used to measure the significance “importance” plays an important role. One simple and effective criterion is the *document frequency* of a term; it means that only the features with highest number of occurrences in all documents are retained.

Salton [SB88] explore the options of the assigning relevant weights to the terms appropriately for IR tasks. Foremost, multiple occurrences of a particular term in one document closely represent the content of the document and can help to improve the recall measure. Next, term count alone may not have the discriminating power to select all the relevant documents from other irrelevant documents. Therefore, the *idf* factor has been proposed to help to improve the precision measure. In general, the two factors, *tf* and *idf*, are combined by a multiplication operation and are thought to improve both recall and precision measures. Finally, to take the effect of length of documents taken into account, a *cosine* normalization factor is incorporated to normalize the length of the documents. Other more sophisticated information theory criteria have been used in the literature, are discussed below.

### 4.5.1 Term Frequency

The most promising and widely used term weighting methods are listed in Table 3.1. They are a binary weight, a normal raw term frequency, a logarithm of term frequency and an inverse term frequency.

The most straightforward approach to text representation is the *binary* representation, which considers the presence or absence of particular term thus giving no important to term occurrences document (1 for present and 0 for absent). It is quite often used in certain machine learning algorithms such as Naive Bayes, decision tree, where a floating number format of term frequency might not be used or not be used without constraints. But the most familiar term frequency representation uses just the count of raw term (*tf*) in the document.

Moreover, different variants have been reported, such as log(1+*tf*) [BSAS94], where the logarithmic is performed to scale the effect of unfavourably high term frequency in one document. Inspired by the inverse document frequency, ITF (inverse term frequency) was proposed by [LK02]. These term frequency factors could be used as term weighting methods without other factors.
### Table 4.2. Term frequency component

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Binary weight equal to 1 for terms present in a vector</td>
</tr>
<tr>
<td>$T_f$</td>
<td>Raw term frequency (number of times a term occurs in a document)</td>
</tr>
<tr>
<td>$\log(1 + tf)$</td>
<td>Logarithm of the term frequency</td>
</tr>
<tr>
<td>$1 - \frac{r}{r + tf}$</td>
<td>Inverse term frequency (ITF), usually $r = 1$</td>
</tr>
</tbody>
</table>

#### 4.5.2 Document Frequency

Inverse document frequency is yet another interesting feature weighting method proposed [Jon72]. An intuitive idea of this method is to differentiate between relevant documents and irrelevant documents. It is more or less similar to the above said document frequency with the exception that instead counting term occurrences, it counts the documents in which term appears. It is computed as follows $\log(|T_r|/df)$, where $T_r$ is the total number of documents in the whole corpus and $df$ is the number of documents in which the particular term appears.

In a conventional probabilistic model, a term weight is computed as the proportion of relevant documents in which a term occurs divided by the proportion of non-relevant documents in which the term occurs, thus the computed term weight is based on the relevance properties of the documents. On the contrary to this fact, many IR task transformed this term weight into an inverse document frequency factor of the form $\log((|T_r| - df)/df)$ [WS81], because of the fact that lack of knowledge about term occurrences in both the relevant and non-relevant documents. This measure is also known as $idfprob$ (since it was from the probabilistic model). However, this measure is disguised in the form of feature selection called as *Odds Ratio* and has been widely used as in TC where it is computed from training dataset itself.

Some of the features weighting methods which are based on collection frequency components from IR perspective are listed in Table 4.2, which consist of a multiplier of 1 that ignores the collection frequency factor, a conventional inverse collection frequency factor ($idf$), and a probabilistic inverse collection frequency ($idf.prob$).
### Table 4.3. Document frequency component

<table>
<thead>
<tr>
<th>1.0</th>
<th>No change in weight; use original term frequency component</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log(</td>
<td>T_r</td>
</tr>
<tr>
<td>( \log((</td>
<td>T_r</td>
</tr>
</tbody>
</table>

#### 4.5.3 Product of Term Frequency and Inverse Document Frequency

The most widely used feature weighting method used in text categorization is \( tf.idf \) [HN04], where \( tf \) refers to term frequency of a term in a given document. \( idf \) is defined as the inverse document frequency, i.e., the ratio of the total number of documents present in a dataset to the number of documents in which a given term appears. A higher \( idf \) of a term indicates that the term appears in relatively few documents and may be more important during the process of text classification. \( tfidf \) is a commonly used technique for term weighing in the field of information retrieval and is also used in text classification. The \( tfidf \) of a term \( t_k \) in document \( d_i \) is defined using following equation 4.2:

\[
\text{tfidf}(t_k, d_i) = tf(t_k, d_i) \log \frac{|D|}{df(t_k)} \tag{4.2}
\]

Where
- \(|D|\) - total number of documents in a dataset
- \(tf(t_k, d_i)\) - term frequency of a term \( t_k \) in document \( d_i \)
- \(df(t_k)\) - the number of documents in which term \( t_k \) appears.

#### 4.5.3 Document Normalization

In order to reduce the effect of document length to the inductive learning classifier, the researchers often opted a technique called *cosine* normalization where term weighting range is limited to an interval \((0, 1)\). With regard to binary weight, it does not require any sort of normalization since the value can be either 0 or 1. Otherwise, if \( w_{kj} \) represents the weight of term \( t_k \) in document \( d_j \), the normalized term weight \( w_{kj} \) could be computed as
\[ w_{kj} = \frac{w_{kj}}{\sqrt{\sum_k w_{kj}^2}} \]  

Where \( w_{kj} \) – weight of a term \( t_k \) in a document \( d_j \)

### 4.6 Proposed Feature Selection Method based on TBM

Many researchers have pursued a number of combination techniques to combine multiple feature selection methods [MG98, MLA98]. The intuition in such technique is that each of the feature selection method validates features in different manner, so combining these methods together could help to extract possibly a better informative feature set which in turn improve the classifier effectiveness. This combination process should be carried out in controlled manner with objective of improving the effectiveness. It is obvious that the computation incurred in the combining operation is computationally quite expensive since it involves implementing a number of feature selection methods then merging the feature set they yields. The reported research works in this direction are discussed in the following paragraphs:

Rogati and Yang [RY02] have proposed a combining approach where normalization was done on each feature selection method then maximum value is chosen. Finally thersolding is done on combined list of features. They found an improvement in performance while combining c2 with either of DF or IG. Though the idea seems to be quite good but it could be improved up on producing discriminative features.

Del Castillo and Serrano [CS04] have developed an alternative approach using genetic algorithm in which different feature selection methods are combined to yield an optimal feature set. However this approach is not convincing since it is time consuming and feature selection process itself is quite complicated.

Doan and Horiguchi [DH04] have conducted a study on combining MI and DF. They first retrieved top informative features from using MI then these feature set is aggregated with top most informative features set chosen by DF. They found that this combined approach outperformed even the original MI feature set. The experiment conducted was based on combining the top 2000 features of the MI with either the top 100 or the top 200 features of the DF feature using the set union operation. Doan and Horiguchi observed that the combined list outperformed both a feature set of the top 2000 MI and a feature set that contains all the
terms in the training set. Though, it is understood that the union is a quite fast and simple combining method, but it is not so clear about the intuition behind the combination of MI feature set with a DF feature set with smaller size. Probably, this kind of act may lead to a bias to the MI list. On the other hand, combining the same number of features from both lists will not lead to any bias and thus may enhance the performance little better.

Though it is not fair enough to combine feature sets blindly without taking in to account that number of resultant features in the combined list, it would be fair that when evaluating such methods one should compare their performance with performance of the original set using same number of feature size. It is obvious that one may expect a reasonable improvement in the performance of the combined approach than the original set of features of same size. Therefore, to increases the performance of inductive learning classifier one might focus on combing multiple features set with reduced number of features instead of increasing the number of selected features on single feature selection method.

Nevertheless the computation involved in this combination operation is quite high, but reducing the dimensionality of the feature space is paramount important in TC; since it not only affect the storage space needed but also the slow down the classifier. Moreover one should not compromise on efficiency of combining operations attributes like fast, simple and effective. However while investigating the performance of this combination technique to infer pros and cons; one must consider different threshold values and different corpus.

Keeping this view in mind, this chapter proposes novel evident theoretic feature selection method for combining evidence underlying in each feature set so that resultant feature set perform better than the original ones.

Feature selection in text categorization is stated as follows: Given a set $X$ consisting of $n$ features $f_1, f_2, f_3, \ldots, f_n$, the problem of feature selection is to choose optimal subset $Y$ of $X$ ($Y \ll X$) which would bring the effectiveness for the system. To solve this problem, features can be filtered based on the criteria. For each feature, it computes the term score according to a criterion. Thus it has $t$ ways of representing documents with $t$ criteria.

Initially, the intuitive motivation behind the proposed approach is described and then provided a formal definition of the proposed method. In this approach, the existing feature selection criteria are considered as source of evidence. Each feature selection criteria such as information gain, odd ratio, chi-square predicts a set of features which are considered as an
independence item of evidence $E_i$ known as neighborhood. Thus each neighborhood consists of few hundred of features and these neighborhoods may overlap such that some features may fall in all neighbourhood thus playing an important role in representing the document. These neighbourhoods of evidences are combined to induce a mass function representing partial support by different neighbourhood. While combining evidences, this work takes into account individual feature weight (tf-idf), only the feature which has substantial relevancy in terms of weight is considered. For this purpose, the term weight $R$ is split into number of smaller intervals $R_1, R_2, \ldots, R_q$ to which a term weight may belong to. Feature selection problem is stated mathematically as follows:

This work takes the frame of discernment to be $\Omega$ that is the collection of all possible informative words derived from the training set. Let $t \in \Omega$ be the informative feature for which it seeks evidences from the existing metrics. For the sake of simplicity, this thesis considers only three neighbourhoods $E_1, E_2$ and $E_3$, to represent the set of features selected by IG, CHI and OR respectively. The frame of discernment along with overlapping neighborhoods of features of training set is shown schematically in Figure 4.2.

![Figure 4.2. An example of feature in frame of discernment](image)

Consider $E_i \in 2^\Omega$ and feature strength (tf-idf) $r \in R$. This work interested in the joint probability $P(E_i, r)$ —the probability that a randomly selected element $x$ of $\Omega$ belongs to $E_i$ and its feature strength falls in the interval $r$, i.e., $x \in E_i$ and $f(x) = r$. Since the knowledge about the distribution $p$ is unknown, it can be approximated $P(E_i, r)$ by applying the principle of indifference.
\[ P(E_i, r) = \frac{|E_i'|}{|\Omega|}. \]  \hspace{1cm} (4.4)

Where  \( E_i \) - a neighbourhood  \( i \)

\(|\Omega|\) - a frame of discernment

\( r \) – feature weight

\( E_i' = \{ x \in E_i : f(x) = r \} \)

Then a mass function  \( m[t] \) is induced for  \( t \) from the  \( h \) neighbourhoods, as a mapping  \( m[t] : 2^{\Omega} \to [0, 1] \) such that, for  \( X \in 2^{\Omega} \) and  \( r \in R \),

\[
m[t](X, r) = \begin{cases} \frac{P(X, r)}{K}, & \text{if } X = E_i \text{ for some } i \\ 0, & \text{otherwise} \end{cases} \]  \hspace{1cm} (4.5)

Here  \( K \) is a normalizing factor. It follows that  \( K = \sum_{i=1}^{h} \sum_{r \in R} P(E_i, r) \). Note that by  \( m[t](X, r) \) it means  \( m[t](X \cap \{ x \in \Omega : f(x) = r \}) \), which is similar to the interpretation of joint probability  \( P(X, r) \). Clearly  \( m[t] \) is a mass function. In particular

\[
\sum_{r \in R} \sum_{X \in 2^{\Omega}} m[t](X, r) = \sum_{r \in R} \sum_{i=1}^{h} m[t](E_i, r) = 1.
\]

**Decision Making based on Pignistic Probability**

This work proposes to choose a feature through marginal pignistic probability. For this it specify the joint pignistic probability as  \( \bar{BetP} : 2^{\Omega} \to [0,1] \) such that, for  \( X \in 2^{\Omega} \) and  \( r \in R \),

\[
\bar{BetP}(X, r) = \sum_{i=1}^{h} m[t](E_i, r) \times \frac{|X \cap E_i|}{|E_i|} \]  \hspace{1cm} (4.5)

Since  \( E_i \) is a collection of features which includes  \( t \in E_i \) which may belong to  \( E_i \) or not. One can understand  \( t \) as a singleton set, therefore  \( t \cap E_i' = \{ t \} \) is either 1 or 0 depends on its presence in  \( E_i \) or not and  \( |\{ t \}| = 1 \). Then it infer the following joint and marginal pignistic probabilities for  \( t \in \Omega \),
\[
\overline{BetP}(t,r) = \sum_{i=1}^{h} m[t](E_i, r) / |E_i|
\]

\[
\overline{BetP}(t) = \sum_{r \in R_j} \overline{BetP}(t,r)
\]

It derives a simple score for selecting discriminative feature is as follows:

\[ S = \arg \max_{j=1}^{m} \{ \overline{BetP}(t) \} \]

The derived feature selection rule is adapted to compute the score for each term in controlled dictionary. Then it selects top \( m \) features from the feature set by arranging terms with respect to the computed score.

**4.7 Summary**

In standard benchmark datasets used for text categorization, an enormous amount of noisy and irrelevant features hurt the performance of the text classifier invariably. This chapter presented a study on reducing size of the feature space and improving the learning process thru a process called as curse of dimensionality. It also analysed various traditional feature selection methods for text categorization in order to improve the classifier performance. Based on the analysis, a novel method of learning evidences from traditional methods for feature selection has been proposed. The theoretical investigation based on evidence theory indicated that the proposed method could improve the performance of text categorization in situations where the features of negative categories play a vital role.