CHAPTER 4
CLASSIFIER DESIGN FOR MULTICLASS CLASSIFICATION

Naive Bayes classifier is used for performing multi-class classification of URLs as it has less training time and memory requirement. However to support large scale data with high dimensional features, a novel feature weighting method for Naive Bayes classifier is proposed and the details are presented in this chapter. To realize the URL classifier as first level filter and to improve the classification performance of the Naive Bayes classifier, the method of combining a rejection framework is discussed in this chapter.

4.1 NAIVE BAYES CLASSIFIER

Naive Bayes (NB) is one of the popular methods of supervised classification and finds its application in many areas including Text Categorization and web page classification. The performance of Naive Bayes classifier is surprisingly well, even though it has unrealistic independent assumptions (Kim et al., 2006). As discussed in Section 2.3.1, in Text Categorization tasks, while representing a document in the vector space, the feature space dimensionality is determined by the number of unique words present in the training corpus. In multivariate Bernouli Naive Bayes model, the presence or absence of a word are considered, where as in multinomial Naive Bayes model, term frequency of a word is also considered while computing the word probability which in turn used to compute likelihood probability of a document using equation 2.11 and 2.9. In the proposed approach of URL classification, in addition to considering the term frequency, term goodness is also embedded into the computation
of likelihood probability as described in the following section. The feature vector representation followed in document classification poses memory constraint when large number of URLs need to be represented in a very high dimensional feature vector. To support large scale data for URL classification, in the proposed approach of Naive Bayes algorithm, the dictionary of unique terms are constructed offline by applying statistics based feature selection methods new feature weighting method is proposed for Naive Bayes Classifier. Without forming a feature vector for representing an URL, the weights of individual terms of an URL are calculated using the corresponding term frequency and term goodness and then embedded into the computation of likelihood probability of the URL. The proposed approach is detailed in the next section.

4.1.1 Naive Bayes Approach for URL Classification

In URL classification, the goal is to predict the class \( C_k \in C = C_1, C_2, C_3...C_m \) of an input URL \( U \). The instance \( U \) is characterized by the terms of URL. Naive Bayes classifier selects the class and predicts the class label \( C \) of \( U \) which has the maximum posterior probability using the following equation

\[
C = \arg \max_k P(C_k|U)
\]  

(4.1)

In an NB model, the posterior probability can be estimated based on Bayes rule and independence assumption using the following equation.

\[
P(C_k|U) = \frac{P(U|C_k)P(C_k)}{P(U)}
\]  

(4.2)

Here the denominator \( P(U) \) is independent of the class of URL and hence this posterior probability \( P(C_k|U) \) can be estimated using likelihood probability \( P(U|C_k) \) and prior probability \( P(C_k) \) ignoring \( P(U) \). This estimation need not be normalized to a distribution using \( P(U) \), since we are looking at only \( \arg \max \) \( P(C_k|U) \). The prior probability is calculated for every category according to the training
dataset. To obtain the prior probability of each category $P(C_k)$, number of URLs in category $C_k$ is divided by total number of URLs in the corresponding dataset.

In the proposed Naive Bayes algorithm for URL classification, the terms present in a single URL are considered to estimate the likelihood probability of the input URL $P(U|C_k)$. The term goodness value is embedded while computing this likelihood probability. To classify an URL in the test set using this approach, it is preprocessed by removing protocol information, word ‘www’ etc. as discussed in Section 3.1.3. And using delimiters such as dot, hyphen, underscore and other special characters, token features are derived. After removing the non-alphabetic characters, the URL is concatenated and n-grams ($n = 3$ to 8) features are derived. Along with the frequency of each feature (Tokens with n-grams, denoted as TNgut), the bag of terms for that URL is obtained. In this method, the weight of each individual term in the URL for a category $C_k$ is computed by multiplying its term goodness in the corresponding statistics dictionary $SDC_k$ by its term frequency. If it is not present in the dictionary, that term is assigned zero weight. The following equation is used to compute the weights of each term present in the given URL.

$$wgt(t_i) = g_i \ast f_i$$  \hspace{1cm} (4.3)

The weight of the given input URL $U$, is the sum of weights of all its terms, denoted by $wgt(U, C_k)$ and can be computed as given below.

$$wgt(U, C_k) = \sum_{i=1}^{n} wgt(t_i)$$  \hspace{1cm} (4.4)

The likelihood probability $P(U|C_k)$ is estimated using this weight and the term goodness of all the terms present in the corresponding statistics dictionary $SDC_k$ as shown below.

$$P(U|C_k) = \frac{wgt(U, C_k)}{SumGoodness(SDC_k)}$$  \hspace{1cm} (4.5)

Here $SumGoodness(SDC_k)$ is calculated by adding the term goodness($g_i$) of ev-
Every term \((t_i)\) in the dictionary \(SDC_k\). In this way, the term goodness is embedded into the computation of likelihood probability. The posterior probability of the given URL to belong to category \(C_k\) is computed by multiplying this likelihood probability and prior probability. In this way, the posterior probability has been calculated for every category and the class label of the given URL is predicted based on the category for which we have obtained maximum posterior probability as given by the equation 4.1. The procedure for URL classification using Naive Bayes classifier by utilizing Statistics based feature selection method is given in the Algorithm 3 for multi-class classification.

**Algorithm 3** Naive Bayes Algorithm with Statistics based Feature Selection

**Input:** A test URL \(U\)

**Output:** Predicted category label \(C\)

1. Preprocess \(U\) and extract bag of terms \(B_u\). Let \((t_i, f_i)\) denote term \(t_i\) with its frequency \(f_i\)
   \[B_u = \{(t_1, f_1), (t_2, f_2), \ldots, (t_n, f_n)\}\]

2. FOR every category \(C_k\)
   2.a) FOR each element \((t_i, f_i) \in B_u\)
       2.a.1) Obtain term goodness \(g_i\) from statistics dictionary \(SDC_k\)
       2.a.2) Calculate weight of \(t_i\)
       \[wgt(t_i) = g_i \times f_i\]
   ENDFOR
   2.b) Calculate weight of \(U\) in \(C_k\)
       \[wgt(U, C_k) = \sum_{i=1}^{n} wgt(t_i)\]
   2.c) Calculate likelihood probability
       \[P(U|C_k) = \frac{wgt(U,C_k)}{\text{SumGoodness}(SDC_k)}\]
   2.d) Estimate posterior probability
       \[P(C_k|U) = P(U|C_k)P(C_k)\]
   ENDFOR
3. Find the maximum and assign that label to \(U\)
   \[C = \arg\max_k P(C_k|U)\]
4.2 COMBINING REJECTION FRAMEWORK TO THE NAIVE BAYES CLASSIFIER

For URL based classifier, the generalization performance is more important. The URL is a tiny fraction of a web page and the minimal information present in URL alone is not sufficient to identify the category of a web page. Even though all the URLs may not have rich source of information to aid in classification, some URLs do contain clue about the web page. So machine learning algorithms can be made to learn the patterns from these type of URLs and classify them. But the classification decision is difficult for those URLs which do not provide any clue, so reject option can be added to the classifier. In this method, some URLs are rejected rather than classifying wrongly. And these unclassified URLs can be passed on to next stage for post processing by acquiring more information from its content. For example, if the URL to be classified contains IP address alone, then it can not be classified by extracting token and n-gram features. So additional information like page title, anchor text, or content must be used in the successive levels of the multi-level classifier after fetching the corresponding page. For the first level filter web page filter design, such costlier features need not be used and rejection option is less costlier approach. The performance of any classifier is measured by its accuracy, in other words its ability to recognize a pattern correctly. We use the term error/misrecognition/misclassification when a pattern of one class is classified as that of another class. When the confidence of a classifier is too low to make a correct decision for some inputs, we can convert the potential misclassifications into rejections by allowing the classifier to reject those inputs. In this way, we can improve the accuracy of the classifier and error rate can be minimized. A suitable trade-off between accuracy and rejection rate can be made while using these classifiers.

We propose a method that combines a rejection framework with Naive Bayes classifier to classify the URLs under multiclass scenario. Bayes decision rule assigns each URL U, to the class $C_k \in C$, for which the posterior probability
$P(C_k|U)$ is maximum as given in the Equation 4.1. Depending on the application, a trade-off between error rate and reject rate can be made. The best error-reject trade-off and the optimum rejection rule is suggested by Chow (1970). According to Chow, the optimum rule is to reject a pattern if the maximum of the posteriori probabilities is less than some threshold. A pattern is assigned to some class $C_k$, only if its posterior probability is the maximum among all the $m$ classes and also above the threshold. Note that what we have calculated as $P(C_k|U)$ is not the actual posterior probability and it is a calibrated probability measure. But it is sufficient for classification as we are interested in finding only $\arg\max P(C_k|U)$. The probability of URL $P(U)$ is independent of the category, so this denominator can be ignored to save computation. The estimated posterior probability value is a very small value. Even though it is sufficient to find the maximum, confidence score of the classifier can better be computed by normalizing these values to lie between 0 and 1, so that a suitable threshold can be chosen based on the confidence score rather than using the posterior probability value directly. So we estimate the Confidence Score of the classifier from that
Table 4.1: Illustrative Example: Why to use Confidence Score based Threshold?

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Posterior Probability</th>
<th>Confidence Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>5.33E-007</td>
<td>0.0108</td>
</tr>
<tr>
<td>C2</td>
<td>6.24E-007</td>
<td>0.01266</td>
</tr>
<tr>
<td>C3</td>
<td>3.44E-008</td>
<td>0.0006</td>
</tr>
<tr>
<td>C4</td>
<td>2.72E-008</td>
<td>0.0006</td>
</tr>
<tr>
<td>C5</td>
<td>1.55E-011</td>
<td>3.14E-007</td>
</tr>
<tr>
<td>C6</td>
<td>9.02E-010</td>
<td>1.83E005</td>
</tr>
<tr>
<td>C7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C8</td>
<td>7.37E-007</td>
<td>0.0149</td>
</tr>
<tr>
<td>C9</td>
<td>6.07E-007</td>
<td>0.0123</td>
</tr>
<tr>
<td>C10</td>
<td>9.83E-008</td>
<td>0.0020</td>
</tr>
<tr>
<td>C11</td>
<td>5.46E-007</td>
<td>0.0111</td>
</tr>
<tr>
<td>C12</td>
<td>6.84E-008</td>
<td>0.0014</td>
</tr>
<tr>
<td>C13</td>
<td>4.61E-005</td>
<td>0.9345</td>
</tr>
</tbody>
</table>

The following example illustrates the need for finding confidence score of the classifier and shows why it is better to use threshold based on the confidence score rather than directly using the posterior probability. The posterior probabilities calculated using the proposed method for the URL "http://www.golfclub.com/" is tabulated in Table 4.1 along with the computed confidence scores while considering 13 different categories.

Figure 4.1 illustrates, how the rejection framework is combined to the Naive Bayes classifier. In the proposed rejection framework with the Naive Bayes classifier for URL classification, an URL U is rejected based on the Confidence Score of the classifier, according to the following reject rule.

\[
\max_{k=1\ldots m} CS(C_k|U) < T
\]

where \( T \) is the rejection threshold, a constant between 0 and 1 (0 < T < 1) and
$CS(C_k|U)$ is the confidence score of class $C_k$ for the given URL $U$. To accept the URL $U$ for recognition and to classify it as belonging to class $C_k$, we have the following optimum recognition rule.

$$\max_{k=1..m} CS(C_k|U) \geq T$$

(4.8)

For the URL classifier with a rejection framework, we define the Rejection rate as follows:

$$RejectionRate = \frac{Number\ of\ Rejected\ URLs}{Total\ number\ of\ URLs}$$

(4.9)

According to minimum risk rule of Chow (1970), for binary classifier, if a rejection threshold is chosen below $1/2$, then reject rate is always zero. For a multi-class classifier with $m$ classes, we can define the Minimum Risk Rule as follows: when the rejection threshold is chosen below $1/m$, then the rejection rate is zero. So the reject/recognition rule is applied on an URL, by varying the values of rejection threshold that ranges from $1/m$ to 0.9 with the increments of 0.05. The rejected URLs from this first level classifier can be used as an input for the next level classifiers. In this dissertation, we have restricted our classifier design to be in first level alone with cheaper features extracted from URLs.

This chapter explained the Naive Bayes approach for URL Classification. Naive Bayes is the simplest classification technique and considerably performs well for data mining tasks. The performance of the Naive Bayes classifier using statistical feature selection methods have been studied. To use the proposed URL classifier as a first level filter, a rejection framework has been combined to the Naive Bayes classifier, that rejects some inputs when the confidence of a classifier is too low rather than misclassifying them.