CHAPTER 7

CLASSIFIER DESIGN FOR BINARY CLASSIFICATION

Using the Linear SVM weight based feature selection, the discriminating features are identified and URLs are classified under binary classification settings. For performing binary classification, Support Vector Machine is used. As the URL contains minimal information, a rejection framework is also combined with the SVM classifier so that the binary URL classifier can be used as a first level filter. The details are presented in this chapter.

7.1 BINARY CLASSIFICATION OF URLs USING SVM

In URL classification, the goal is to predict the class label of an URL as either belonging to the category $C_k$ or not. URLs from m categories are considered, the target category is denoted by $C_k$ and all the other categories are denoted by $\bar{C}_k$, in other words one-against-rest approach is followed for binary classification. To perform binary classification of URLs, training set composed of N URLs (observations) are available, each one belonging to one of two classes $C_k$ or $\bar{C}_k$. The binary classifier for $C_k$ determines whether the given URL belongs to the target category $C_k$ or not and predicts the class label as 1 or -1. The $i^{th}$ URL is defined by a feature vector $x_i \in R^d$ and a label $y_i \in \{ -1, 1 \}$ that indicates which class the URL belongs to. For performing binary classification, Support Vector Machine (SVM) with linear kernel is used.

Yuan et al. (2012) has given a comprehensive survey of advances in large-scale linear classification. Joachims (1998) has suggested Support Vector
Machine with linear kernel as a suitable method for text classification. The key properties of URL classification that makes it resemble text classification are: High dimensional input space, few irrelevant features, URL vectors are sparse. So SVM is a good choice of learning algorithm for URL classification problems.

SVM with linear kernel is used for URL classification as it supports high dimensional feature space and resistant to noise. For classifying URLs belonging to m categories, m binary classifiers are designed, one for each category C_k. For each category C_k, the universal dictionary D_k with discriminating features is considered and feature vector is formed in the space defined using this dictionary for training URLs of C_k. By training the SVM, the model is obtained for category C_k which is then used to classify the test URLs. The procedure for binary classification using SVM is given in Algorithm 5. As discussed in Sec-

Algorithm 5 Design of URL binary Classifier

**Input:** \( TrC_k \), the Training URLs of category \( C_k \)

**Output:** Predicted SVM output \( g(x) \) for the given test URL with the sign of output determining the class label

1. Preprocess each URL \( u \in TrC_k \) and extract bag of terms \( B_u \) for the all feature types \( F_r \).
   - 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 each element \( (t_i, f_i) \in B_u \)
   2.a) Obtain term goodness \( g_i \) from dictionary \( D_k \)
   2.a.2) Calculate weight of \( t_i \), if \( t_i \) is a discriminating feature i.e. \( t_i \subset D_k \)
      \[ wgt(t_i) = g_i \times f_i \]

3. Represent each URL of the form \( B_u \) as a vector in the space defined by \( D_k \) in the fixed feature dimension \( d_k \). Let \( x \) denote an URL that is represented as a vector in the Vector Space Model(VSM)

4. Train the SVM and obtain SVM classifier model \( SVMC_k \) for category \( C_k \)

Section 3.1.3, each URL in the training set is preprocessed by removing protocol information, word ’www’ etc. 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 every URL is obtained. Instead of using these set of terms as features, the discriminating dictionary learnt already (using Algorithm 4) for category $C_k$ is used to define the feature space of the input URL. By using this dictionary, the feature vector dimensionality is restricted and hence supports training large scale data with many URLs. In the existing approach of URL classification proposed by Baykan et al. (2011), $tf \times idf$ scoring method is used to weight the URL features. As $idf$ does not make use of rich class information available, linear SVM weights have been used in the proposed method. Depending on the discriminating capability of URL features, each feature is weighted by obtaining their corresponding term goodness (or) Linear SVM weights from the dictionary $D_k$ for category $C_k$. The weight for each feature type or term $t_i$ (Tokens, 3-grams, 4-grams, 5-grams, 6-grams, 7-grams and 8-grams) is calculated using the Equation given below.

$$wgt(t_i) = g_i \times f_i$$

where $g_i$ denotes the term goodness (i.e. linear SVM weights) of the feature (term) $t_i$ and $f_i$ denotes the frequency of feature $t_i$ in the given URL. If the extracted term $t_i$ is not a discriminating feature for the category $C_k$, the weight of that term becomes zero. In this way, every feature is weighted using $tf \times svmWgt$ and an URL is represented in the vector space, the dimensionality of this vector is determined by the dictionary size of $C_k$. By representing all the training URLs (that includes positive samples of $C_k$ and negative samples from remaining categories $C_{\bar{k}}$) with the respective positive and negative labels, SVM binary classifier for category $C_k$ is learnt. Using this SVM model ($SVMC_k$), (i.e. using the weight vector $w$ learnt for this category $C_k$ along with bias term) the new URLs in the test set are classified. In other words, using the Equation ??, the value of the discriminant function $g(x)$ for the given URL (represented as a vector $x$) is computed. Here $g(x)$ is the distance of $x$ from the optimal hyperplane and the sign of $g(x)$ determines the prediction label of the given URL. When the value of $g(x) \geq 0$, the given test URL is predicted as belonging to the category
\( C_k \), and \( \tilde{C}_k \) otherwise.

7.2 COMBINING REJECTION FRAMEWORK TO SVM CLASSIFIER

As URLs contain minimal information, a rejection framework can be added to the binary classifier and binary URL classifier can be used as a first level filter in the multi-stage or hierarchical classifiers. Even though SVM performs well for binary classification of URLs, combining a rejection option to the SVM can decrease its error rate further.

URLs contain minimal information and URL based classifier alone is not sufficient for all kind of applications that are very sensitive to classification errors. 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 or contains only IP address. To improve the reliability of the classifier and to avoid making unnecessary classification errors, a method to combine rejection framework with the URL based binary classifier is proposed in this dissertation. By this method, URL based binary classifier can be used as a first level filter in multi-stage / hierarchical classifiers. In this approach, the confidence of the URL based classifier is estimated based on which the input URLs are either classified or rejected. This approach helps to make quick and smart decisions on the fly and postpones the process of downloading the corresponding web page contents for extracting costlier feature to the successive levels only for rejected URLs. The existing rejection techniques for SVM classifier suggested in the literature by Zhang and Metaxas (2006) and Fumera and Roli (2002), either use a distance based threshold or embedded the rejection framework into the classifier design itself as a part of training which is a com-
plex technique. In this dissertation, a simple rejection methodology is proposed without affecting the design of the SVM classifier.

The optimum classification rule with reject option is defined by Chow (1970). A pattern \( x \) is rejected if the maximum of the posteriori probabilities are less than some threshold. In the proposed rejection framework with the SVM classifier for URL classification, an URL \( U \) is rejected based on the posterior probability of the classifier, according to the following reject rule.

\[
\max_{k=1,2} P(C_k|U) < T
\]

where \( T \) is the rejection threshold, a constant between 0 and 1 (0 < \( T < 1 \)) and \( P(C_k|U) \) is the posterior probability 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.

\[
\max_{k=1,2} P(C_k|U) \geq T
\]

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

\[
RejectionRate = \frac{\text{Number of Rejected URLs}}{\text{Total number of URLs}}
\]

Standard SVMs do not provide the posterior probability. But different ways of obtaining posterior probabilities for SVM classifier has been discussed by Vapnik (1995) and Platt (1999). The existing rejection techniques for SVM classifier suggested in the literature by Fumera and Roli (2002) and Mukherjee et al. (1999), either use a distance based threshold or embedded the rejection framework into the classifier design itself as a part of training which is a complex technique. In this paper, a simple rejection methodology is proposed without affecting the design of the SVM classifier. The prediction output of SVM classifier
is converted to a probability value which is used for taking rejection decision on
the URLs by following the method suggested by Vapnik and detailed by Flach
(2012).

The linear classifier classifies an example and produces a score $g(x) = w^Tx + b$. Due to geometric nature of linear classifiers, these scores can be used
to obtain the distance of $x$ from the decision boundary. The distance can be
obtained as given below.

$$d(x) = \frac{|g(x)|}{||w||} = \frac{w \cdot x + b}{||w||} = w' \cdot x + b'$$

where $w' = \frac{w}{||w||}$ rescaled to unit length and $b' = \frac{b}{||w||}$ the corresponding rescaled
intercept.

To obtain the probability estimate from the linear classifier outputting
distance score $d(x)$ for the observed point $x$, a mapping function known as S-
shaped logistic function is defined. The standard logistic function used to obtain
probability estimates is given below

$$d(x) \rightarrow \frac{1}{1 + e^{-\gamma(d(x) - d_0)}} \quad (7.5)$$

Here $\gamma$ is the scaling factor $d_0$ has the effect of moving decision boundary halfway
between mean of two classes. When the two classes are equally prevalent, class
means are equidistant from decision boundary and one unit of variance apart, the
mapping function reduces to the following equation.

$$d(x) \rightarrow \frac{1}{1 + e^{-d(x)}} \quad (7.6)$$

Thus the probability estimate for the URL $x$, with the given distance
score $d(x)$ can be obtained. The obtained probability is for positive prediction
when the $g(x)$ is positive and for negative prediction when the value of $g(x)$ is
negative. Optimum rejection rule suggested by Chow (1970) is applied to take
reject decisions on the URL pattern. To take a reject decision for an URL, this calibrated probability measure calculated using Equation 7.6 is used. According to Chow’s rule, for binary classifier, if a rejection threshold is chosen below 1/2, then reject rate is always zero. So rejection rule is applied on an URL, by varying the values of rejection threshold that ranges from 0.45 to 0.9 with the increments of 0.05. By conducting experiments, the impact of rejection threshold on rejection rate and error rate are analysed. The rejected URLs from this first level classifier can be used as an input for the next level classifiers for further processing which is beyond the scope of this dissertation. The process of converting the SVM distance score to posterior probability is illustrated below with examples. If the distance score of 5.199 is obtained from the SVM binary classifier for Games category URL ’http://www.gamespot.com/xbox/action/batmandarktomorrow/’, then this distance score is converted to a probability value using Equation 7.6. The probability of the given URL to belong to category ‘Games’ is 0.995. For various values of threshold ranging from 0.45 to 0.9, the rejection decision is taken. As this URL has the probability above the threshold 0.9, it’s predicted output is accepted and it is classified as ’Games’ by the SVM classifier.

The Arts category URL ’http://www.waopera.asn.au’ is classified as ’Arts’ by the Arts SVM classifier with the output score of 0.531, for which the computed posterior probability gives 0.62.

The prediction of the classifier is accepted, when the threshold is set to 0.45, 0.5, 0.55 or 0.6. This URL may be rejected for application that need very high accuracy, where the rejection threshold can be set to a value above 0.65.

The SVM classifier trained for ’Games’ category produces the score of -0.5649 for the URL ’http://www.olwydd.org’. When this distance score is converted to a probability, we obtain 0.637 as the probability for this negative prediction.

When the threshold is varied, this URL gets rejected when the rejection threshold is set to 0.6. Even though this URL belongs to the category
'Games', the classifier predicts it wrongly. As this URL contains less information, this wrong prediction can be avoided by rejecting the URL depending on the accuracy level required for the application need.

In this way, by combining the rejection framework to the binary classifier, it can be used as a first level filter in multi-stage/hierarchical classifiers. The URLs which get rejected in this stage based on URL features alone, can be processed later by extracting more costlier features like title of the web page, anchor text or the contents of the web page. Hence, the need for downloading the web page can be postponed to the successive levels of the classifier design.