CHAPTER 2

RELATED WORKS

In this Chapter, the problem of web page classification is presented along with the existing techniques suggested in the literature. The need for URL based web page classification systems is studied and the related techniques used in the literature for feature selection are analysed. The role of machine learning techniques for URL classification is presented along with a brief overview of effect of combining the rejection framework to the classifiers.

2.1 WEB PAGE CLASSIFICATION

Web page classification is a process of assigning a web page to one of the category labels. It can be divided into different types based on the content of web pages. Among the types such as subject classification, functional classification, sentiment classification etc. we address the problem of subject classification. It is the process of classifying the web page based on the subject or the topic it talks about.

According to the survey of Qi and Davison (2009), the existing automated classification systems classify web pages based on HTML content or by structure of the document or by generating summaries to decide the category of the web page under consideration. Qi and Davison (2006) categorized the web pages using not only the contents of it, but also with class information from the neighboring pages. They used class of neighboring pages to determine the topic of a web page. By considering ODP dataset with top 12 categories, they have shown that among the neighboring pages of a target page, sibling pages contribute more in providing clue for categorization task. Lim et al. (2005) proposed
a method for genre classification by combining multiple set of features. They used URL features, HTML tags, token information, most frequently used words and punctuation marks. They achieved a highest precision of 0.95 for research reports. Sun et al. (2002) classified the web pages by using context features and full text features. They extracted features from title, anchor text and hyperlinks in addition to full text of the web page. They evaluated their approach on WebKB dataset using SVM as their classifier. They reported an $F_1$ of 0.575 with their best combination of features. Craven and Slattery (2001) combined statistical text-learning method with relational rule learner and proposed an approach for classifying the hypertext. They used content and structure of the web pages. They evaluated their approach on WebKB dataset and achieved an $F_1$ measure of 0.53, 0.435, 0.275 and 0.542 for course, faculty, project and student categories for which the macro-average is 0.445. All the content based methods require the web page contents be downloaded for extracting features and hence waste bandwidth. Also it slows down the classification process.

The content based approaches are not suitable when the web page contains only images, so Zhang et al. (2006) analyzed the use of URLs for detecting pornographic web pages. They used N-gram representation for URLs and applied feature selection based on their own relevance measure. They combined content based approach with URL based features to detect the objectionable web pages. As the contents of a researcher homepage changes dynamically, Gollapalli et al. (2013) proposed a co-training method to classify the researcher homepages by combining the URL features with content features. They have used top unigrams and bigrams from URL strings, surface pattern and wordnet features. They have reported that, they are not able to extract URL features from 27% of URLs and combined the content features also. Kan (2004) categorized the web pages by using features extracted from URL text, title text, anchor text and page text. Also they showed that, it is possible to classify the web pages by using only URL features without using content and achieved an average $F_1$ measure of 0.338 on WebKB dataset.
As URL based classification system saves bandwidth and speeds up the classification process, it gains attraction. Kan and Thi (2005a) suggested another approach for classifying the web pages using only URL features and shown that classification speed can be improved when the features are derived from URLs alone. In this approach, the URLs were segmented into meaningful chunks and features such as URL component length, content, orthography, token sequence and precedence were considered. Then Maximum-entropy learning was applied on these features to classify the web pages and an $F_1$ measure of 0.525 was achieved on the WebKB dataset. On ODP dataset, Kan and Thi (2005b) have achieved the best mutliclass accuracy of 0.368.

Hernandez et al. (2012a) used the information present in URL tokens as features for classifying the web pages. The features calculated for every token not only depends on the corresponding token but also considers the sequence of all tokens in the URL up to that token’s position. The probability of token prefix is estimated and URL prototype is built using this information. Hernandez et al. (2012b) used these probabilistic token prefix features to build URL patterns that represent the different types of URLs in a site. Those patterns are used to classify pages according to their topic, without the need for downloading them. This research work focused only on the web pages from a certain site. Devi et al. (2007) classified web pages based on URLs by applying three classification algorithms viz., Naive Bayes, SVM and RBF network. They categorized the web pages of WebKB dataset considering only one category ’course’ and reported the overall success rate as 0.308, 0.380, 0.520 for the above three algorithms.

Baykan et al. (2009) classified the web pages into one of the 15 categories of ODP dataset by considering the URL features alone. They suggested different features for classifying URLs that are based on tokens and n-grams. In their dictionary based feature selection method, they compiled a token dictionary of indicator words by taking all the words in the first two levels of ODP hierarchy. They reported an $F_1$ of 0.23 for this dictionary based method. Baykan
et al. (2011) strengthened the construction of dictionary based method by applying some heuristics rules. They decided these rules by manually inspecting all the URLs in the ODP dataset. Applying these rules, they constructed different dictionaries with tokens, domains and all-grams and reported the performance in terms of $F_1$ as 0.46, 0.33 and 0.73 respectively. They have also used all-grams (n=4 to 8) as features for classifying URLs with SVM and Maximum Entropy Classifier. By this method, even though the performance of individual binary URL classifiers is reasonable, the multiclass performance is poor due to confusion among the categories. Singh et al. (2012) proposed an online incremental learning algorithm by framing the URL-based topic classification problem as a stateless Reinforcement Learning Problem. In their approach, the action space is comprised by URLs and immediate rewards are the on-topic web pages. They have also suggested an SVM based approach. To reduce the dictionary size and to solve the asymmetry problem, they have proposed a model in which incremental learning was performed by updating the features in which they considered less URLs for training and more number of URLs for testing by considering only 8 categories of ODP dataset. They have derived features from i) URLs alone ii) URL & anchor text (UA) iii) URL, title(anchot ext) & description and used three learning algorithms viz., SVM, Reinforcement Learning and Online Incremental Learning. With their URL based approach, they have obtained the $F_1$ measure of 0.71, 0.71 and 0.76 for the above three techniques. To address the large-scale topic classification, Kolcz et al. (2012) proposed a method by utilizing web graph information to precompute topical likelihood for known hosts. This method involves processing of massive amount of information even though it is done offline. In their approach of Graph enhanced URL classification problem, they considered a graph in which vertices correspond to URLs and the edges correspond to link between two URLs. Their objective is to label all the vertices of a graph accurately with the labelled set of URL vertices. By this complicated method, they have achieved only 1% improvement over the existing method suggested by Baykan et al. (2011) while considering 13 categories of ODP dataset.
Even though description is available in training data (ODP), getting this description and title requires downloading the web page at least partially. But it is not necessary for information filtering and focused crawling which will not serve the purpose. In this dissertation, a dictionary learning method is proposed to address the large-scale problem of web page classification by which discriminating URL features are learnt automatically and web graph information or description of the web page is not required for the proposed approach.

2.2 FEATURE SELECTION METHODS

Feature selection is the process of finding the relevant features discarding the irrelevant and redundant ones. According to the survey of Bolón-Canedo et al. (2013), two approaches of feature selection are individual evaluation and subset evaluation. In the first case, which is also known as feature ranking method, individual features are assessed based on their degree of relevance and assigned weights accordingly. In subset evaluation, each candidate subset is evaluated based on some evaluation criteria and the best one is chosen. Based on the relationship with the learning algorithm, the feature selection methods are further categorized into three types viz., filter method, wrapper method, and embedded method. In filter method, feature selection is done as a preprocessing step independent of the learning algorithm. In wrapper method, feature selection involves the learning algorithm and uses its prediction performance to assess the subset of features. The embedded methods perform feature selection in the process of training, and they are specific to a particular algorithm, like SVM-RFE. Among this, filter methods have lower computational cost, achieves good generalization ability and fast.

In text classification problems, the native feature space consists of unique terms that occur in documents, which is in the order of many thousands. So feature selection methods are applied to reduce the dimensionality of the fea-
ture space (Yang and Pedersen, 1997). The feature subset selection method for text classification is simple and works as follows: Some evaluation function is applied on a single feature, all the features are evaluated independently, a score is assigned to each of them, and features are sorted based on the assigned score. Then a predefined number of best features (top N features) are chosen and used for induction algorithm.

Various feature selection methods are suggested in the literature for text classification problems. As stated by Guyon and Elisseeff (2003), the potential benefits of feature selection include the following: facilitating data visualization and data understanding, for reducing the measurement costs and storage requirements, reducing training and utilization time. Guyon and Elisseeff (2003) have summarized various feature selection methods and its applications along with the advantages of using each method. Forman (2003) has made an extensive study of feature selection metrics for text classification problem by conducting experiments on the benchmark datasets. In their study, they have compared various evaluation measures such as accuracy, precision, recall, $F_1$ and proposed a new metric called Bi-Normal Separation which is found to outperform all the other performance measures. Bi et al. (2003) has proposed a method to rank the variables and to select features using SVM. In their study, they found that SVM weight based feature selection outperforms other feature selection techniques by conducting various experiments on benchmark data. Rakotomamonjy (2003) has also proposed a method based on SVM for selecting the relevant variables by ranking them. A comparative study of five different feature selection methods for text categorization are discussed in Yang and Pedersen (1997) that includes Document Frequency (DF), Information Gain (IG), Mutual Information (MI), $\chi^2$ test and Term Strength(TS). CHIR method is applied for text clustering and Li et al. (2008) shown that performance is better than $\chi^2$ method of feature selection. Janaki Meena et al. (2012) used the relevance measure $R_{w,c}$ of CHIR method as the heuristic value of a term in their enhanced Ant Colony Optimization algorithm to select features for text categorization.
The $\chi^2$ statistic is used as a feature selection method for text categorization by Yang and Pedersen (1997). The $\chi^2$ statistic measures the lack of independence between a term $t$ and category $c$ and it can be compared to $\chi^2$ distribution with one degree of freedom to judge the extremeness. The degree of dependence can be measured by comparing the observed co-occurrence frequencies with the frequencies expected when term $w$ and category $c$ are assumed to be independent and $\chi^2$ value can be computed. The $\chi^2$ based feature selection method chooses terms with the strong dependency to the categories by ranking the terms based on their $\chi^2$ value. $\chi^2$ value of a term is computed using the observed frequencies in the two-way contingency table of a term $t$ and a category $c$ with the expected frequencies. For this a 2x2 contingency table is formed for every term in the corpus containing $n$ documents of $m$ different categories. The expected frequency $E(i, j)$, where $i$ represents presence or absence of a term $w$ and $j$ represents whether the document belongs to category $c$ or not, can be calculated as:

$$E(i, j) = \frac{\sum_{a \in \{w, -w\}} O(a, j) \sum_{b \in \{c, -c\}} O(i, b)}{n}$$

(2.1)

The $\chi^2$ statistic for a term $w$ and with respect to category $c$ is defined as

$$\chi^2_{w,c} = \sum_{i \in \{w,-w\}} \sum_{j \in \{c,-c\}} \frac{(O(i, j) - E(i, j))^2}{E(i, j)}$$

(2.2)

The value of $\chi^2$ statistic is zero, if $w$ and $c$ are independent. If there is a dependency between $w$ and $c$, for a corpus with $m$ classes, the term-goodness of a term $w$ is defined as the average or maximum $\chi^2$ value. Here $P(c)$ denotes the probability of document to be in category $c$.

$$\chi^2_{avg}(w) = \sum_{j=1}^{m} P(c) \chi^2_{w,c}$$

(2.3)

or

$$\chi^2_{max}(w) = \max_j \{\chi^2_{w,c}\}$$

(2.4)
CHIR method is proposed in Li et al. (2008). The $\chi^2$ method selects the terms having strong dependency on the categories from the feature space without considering the type of dependency. So the terms having both positive and negative dependency are selected. The values of $\chi^2$ are affected by the sizes of the categories and it assigns more weight to the categories with bigger sizes, i.e. categories with more number of documents in it. So these issues are addressed in the CHIR method, which keeps features that have positive dependency for the categories. Also it is not affected by the size of the categories. In CHIR method, a relevancy measure named $R_{w,c}$ was introduced to evaluate the dependency of the term with category.

$$R_{w,c} = \frac{O(w,c)}{E(w,c)}$$  \hspace{1cm} (2.5)

If there is no dependency between a term $t$ and a category $c$ (i.e. if $\chi^2$ is not significant), then $R_{w,c}$ value is close to 1. The value of $R_{w,c}$ is greater than 1 if there is positive dependency and less than 1 for negative dependency. Combining $\chi^2$ value with this relevancy measure $R_{w,c}$ term-goodness of a term $w$ in a corpus with m classes is defined as:

$$r\chi^2(w) = \sum_{j=1}^{m} p(R_{w,c_j})\chi^2_{w,c_j}$$  \hspace{1cm} (2.6)

where $p(R_{w,c_j})$ is the weight of $\chi^2_{w,c_j}$ in the corpus in terms of $R_{w,c_j}$. It is defined as:

$$p(R_{w,c_j}) = \frac{R_{w,c_j}}{\sum_{j=1}^{m} R_{w,c_j}}$$  \hspace{1cm} (2.7)

with $R_{w,c_j} > 1$ When there is a negative dependency between the term and the category, $\chi^2_{w,c_j}$ value is not used for calculating term-goodness and that term is not chosen by CHIR algorithm. The term-goodness measure, $r\chi^2$, is the weighted sum of $\chi^2_{w,c_j}$ statistics when the term $w$ has positive dependency on the category $c_j$. The bigger value of $r\chi^2$ indicates that the term is more relevant for that category. In this way, $R_{w,c_j}$ measures the term-category dependency accurately. Li et al. (2008) has applied CHIR method for text clustering and they have shown
that performance is better than $\chi^2$ method of feature selection and two variants of $\chi^2$ method viz., Correlation Coefficient and SCHI. Janaki Meena et al. (2012) used the relevance measure $R_{w,c}$ of CHIR method as the heuristic value of a term in their enhanced Ant Colony Optimization algorithm to select features for text categorization.

The popular term weighting method $tf \times idf$ and its variants are widely used in Information Retrieval (IR) and for Supervised Learning tasks such as Text Categorization (TC). Baykan et al. (2011) followed this weighting method for URL classification also. But for categorization tasks, training data is available with class labels and this rich source of information can be utilized in weighting the features. For text categorization, different supervised term weighting methods are suggested in the literature (Debole and Sebastiani, 2004; Lan et al., 2009). By assigning higher weights for relevant terms, the performance of classification can be improved. Lan et al. (2009) suggested a measure named Relevance Frequency (RF) for Text Categorization and they have proposed a supervised term weighting method $tf \times rf$ by considering the distribution of relevant documents in the collection. The basic idea of $tf \times rf$ is that, if a high frequency term is more concentrated in the positive category than in the negative category, then it makes more contributions in selecting the positive samples than the negative samples. The notations $a$ and $c$ are used to denote the number of documents in the corpus in positive/negative category that contain the term. The number of documents that do not contain the term in the positive/negative category is denoted by $b$ and $d$. In $tf \times rf$ method, a term’s discriminating power is determined by the number of relevant documents that contain this term only, i.e., $a$ and $c$. They defined the ratio of $a$ to $c$ as the Relevance Frequency (RF) and replaced the $idf$ measure while combining it with $tf$ to weight a term. Their weighting method $tf \times rf$ is defined by the following equation

$$tf \times rf = \log \left(2 + \frac{a}{\max(1,c)} \right)$$

(2.8)
In general, the feature selection methods are applied to achieve two objectives: 1) to remove the redundant, noisy and irrelevant features from the feature space to optimize the classification performance. 2) to reduce the size of feature set in the data representation to optimize the use of computing resources like memory and processing time.

In this dissertation, the feature selection method for multiclass classification of URLs is proposed to meet the first objective of feature selection thereby improving the performance of Naive Bayes classifier. To calculate the term goodness of URL features, a variation of CHIR and Relevance Frequency measures have been used. The feature selection method for binary classification of URLs is proposed to meet the second objective as Support Vector Machines are robust to noise. For multiclass classification of URLs, statistics based feature selection is done and for binary classification linear SVM weight based approach is followed. Both the feature selection methods are like filter methods as the feature selection phase can be done off-line.

2.3 MACHINE LEARNING ALGORITHMS USED

Among the various machine learning algorithms applied for web page classification task, we have chosen two algorithms to perform URL based web page classification. Naive Bayes classifier is used for multiclass classification and Support Vector Machines are used for binary classification.

2.3.1 Naive Bayes Classifier

Naive Bayes classifiers are useful when a large training set is available and the attributes that describe the instances are conditionally independent given classification (Mitchell, 1997).

In document classification, the documents are represented as either
set-of-words or bag-of-words (Kim et al., 2006). In vector representation, each element corresponds to a word. The set-of-words representation (multivariate bernouli Naive Bayes) indicates the presence/absence of a word in the document and bag-of-words representation (multinomial Naive Bayes) considers the term frequency of the words into account. In both of these representations, the order in which the word appears in the document is not considered. In other words, the position of the word is ignored.

For text classification, the most commonly used method is Naive Bayes classifier with the bag-of-words document representation (Mitchell, 1997). For a given document \( d \) containing \( n \) words \( (w_1, w_2, ..., w_n) \), the probability that \( d \) belongs to class \( j \) can be estimated by

\[
P(C_j|d) = \frac{P(C_j)P(d|C_j)}{P(d)} \approx \frac{P(C_j)}{P(d)} \prod_{i=1}^{n} P(w_i|C_j)
\]

(2.9)

Using this likelihood probability along with prior probability, a new document can be classified. To classify a document among the \( c \) categories, the posterior probability for every category is calculated and the one that gives maximum probability is assigned as the category of the document. This can be calculated using the equation given below:

\[
\arg \max_{c_j \in C} P(c_j) \prod_{i=1}^{n} P(w_i|C_j)
\]

(2.10)

Using Laplace smoothing, word probability can be estimated in the following way:

\[
P(w_i|C_j) = \frac{N(w_i, C_j) + 1}{N(C_j) + T}
\]

(2.11)

where \( N(w_i, C_j) \) denotes the number of times word \( w_i \) appears in the training examples of category \( C_i \), \( N(C_j) \) denotes the total number of words in training set of \( C_j \) and \( T \) denotes the total number of unique words in the corpus.

For classification of URLs, Naive Bayes approach is proposed in this
thesis. To improve the performance of Naive Bayes classifier, the term goodness obtained from the feature selection method is embedded while computing the likelihood probability.

2.3.2 Support Vector Machine

Support Vector Machine (SVM) is a state-of-the-art classification method (Burges, 1998) and finds its application in many pattern recognition tasks such as text categorization (Joachims, 1998), web page classification (Baykan et al., 2011), gene selection (Guyon et al., 1996) etc. It is popular because of its high accuracy, ability to deal with high-dimensional data and flexibility in modelling diverse sources of data. The key concept of SVM is the dot product between two vectors, also referred to as an inner product or scalar product. A linear classifier is defined as $w^T x = \sum_i w_i x_i$. It is based on a linear discriminant function of the form

$$ f(x) = w^T x + b $$

(2.12)

where $w$ denotes the weight vector and $b$ denotes the bias. If we consider $b = 0$, then the set of points $x$ such that $w^T x = 0$ are all points that are perpendicular to $w$ and go through the origin - a line in two dimensions, a plane in three dimensions, and more generally, a hyperplane. The bias $b$ translates the hyperplane away from the origin. The hyperplane divides the space into two according to the sign of the discriminant function $f(x)$, the decision boundary of the classifier. It is the boundary between regions classified as positive and negative. To make non-linear SVM out of linear SVM, one should map the data from the input space $X$ to a feature space $F$ using a non-linear function $\phi$. In the space $F$, the discriminant function is

$$ f(x) = w^T \phi(x) + b $$

(2.13)

Explicitly computing non-linear features does not scale well with the number of input features. The inputs in d-dimensional space, after mapping to a feature space F, the dimensionality of feature space is quadratic in d. The effect of this
mapping results in

- A quadratic increase in memory usage for storing the features
- A quadratic increase in the time required to compute the discriminant function of the classifier
- Quadratic Complexity: Feasible for low dimensional data, but not acceptable for high dimensional data like gene expression
- The solution is **KERNEL** methods. It avoids explicit mapping of data to a high-dimensional feature

The weight vector can be expressed as a linear combination of the training examples

$$w = \sum_{i} \alpha_i x_i$$  \hspace{1cm} (2.14)

In input space,

$$f(x) = \sum_{i} \alpha_i x_i^T x + b$$  \hspace{1cm} (2.15)

In feature space $F$,

$$f(x) = \sum_{i} \alpha_i \Phi(x_i)^T \Phi(x) + b$$  \hspace{1cm} (2.16)

The representation in terms of the variables $\alpha_i$ is known as the dual representation of the decision boundary. Computing $\Phi(x_i)^T \Phi(x)$ is impractical for high dimensional feature space, and using kernel functions it can be computed efficiently

$$k(x, x') = \Phi(x)^T \Phi(x')$$  \hspace{1cm} (2.17)

The discriminant function in terms of kernel function

$$f(x) = \sum_{i} \alpha_i k(x, x_i) + b$$  \hspace{1cm} (2.18)

For the maximum margin classifier, the discriminant function maximizes the geometric margin $\frac{1}{||w||}$. That is, it minimizes

$$||w^2||$$  \hspace{1cm} (2.19)
This leads to the following constrained optimization problem

\[
\text{minimize} \quad \frac{1}{2}||w^2|| \quad \text{subject to} : y_i(w^T x_i + b) \geq 1 \quad i = 1..n
\] (2.20)

The constraints ensure that maximum margin classifier classifies each example correctly as data is linearly separable. In reality, for non-separable data, maximum margin can be achieved by allowing classifier to misclassify some training examples. So slack variables are introduced as a part of the constraint

\[
y_i(w^T x_i + b) \geq 1 - \xi_i
\] (2.21)

It is said to be margin error, when \(1 \geq \xi \geq 0\) and misclassification when \(\xi \geq 1\). Now another term is added in the objective function to penalize margin errors and misclassification.

\[
\text{minimize} \quad \frac{1}{2}||w^2|| + C\sum \xi_i \quad \text{subject to} : y_i(w^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0
\] (2.22)

Here \(C\), the constant that sets the relative importance of maximizing the margin and minimizing the amount of slack. This formulation is known as Soft Margin SVM introduced by Cartes and Vapnik. Using the method of Lagrange multipliers, we can obtain the dual formulation, which is expressed in terms of variables \(\alpha_i\). To find the model parameters (\(\alpha_i\) values) from the training examples, the following optimization problem is solved.

\[
\max_{\alpha} \left( \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \right)
\]

subject to \( \sum_{i=1}^{n} \alpha_i y_i = 0 \)

and \( \alpha_i \geq 0, \quad i = 1, 2, \ldots, n \)

The dual formulation leads to an expansion of the weight vector in terms of the
input examples.

\[ w = \sum_{i} y_i \alpha_i x_i \]  

(2.23)

The examples for which \( \alpha_i \geq 0 \) are those points that are on the margin, or within the margin, when soft margin SVM is used. These fraction of training examples are known as Support Vectors which provides support for decision boundary to be in a particular position in Euclidean space.

Traditional method for solving SVM optimization problem is in dual formulation. Recently it is shown that primal formulation can lead to efficient kernel-based learning [using Equation 2.22].

### 2.3.3 Combining Rejection Framework to the Classifier

Many classifiers compute scores on which their class predictions are based. Scoring classifier is a mapping \( s : X \rightarrow R^k \), i.e. a mapping from the instance space to a k-vector of real numbers. Scoring classifier outputs a vector \( s(x) = (s_1(x), s_2(x), ..., s_k(x)) \) rather than a single number. \( s_i(x) \) is the score assigned to class \( C_i \) for instance \( x \). Score indicates how likely it is that class label \( C_i \) applies. If we have two classes, it is sufficient to consider the score for only one class and we can use \( s(x) \) to denote the score of the positive class for instance \( x \). These scores are assigned by a classifier, and are not a property inherent to instances. Scores are not estimated from ‘true scores’ and the scoring classifier has to be learnt from examples in the form of instances \( x \) labelled with classes \( c(x) \). The task where we learn a function \( f \) from examples labelled with true function values \( \langle x, f(x) \rangle \) is called regression. The scoring classifier such as SVM uses the optimal hyperplane \( f(x) = 0 \) to separate positive and negative samples, whereas Rankers does not assume a particular score threshold for separating positives from negatives. A ranking is defined as a total order on a set of instances, possibly with ties. So ROC curve is used to obtain the threshold, based on which classification can be performed, thus turning the rankers into
classifiers. The scores produced by the classifier can be used as the confidence of the classifier, based on which a rejection option is suggested by Chow (1957). Chow has described an optimum rejection rule. The optimum rule is to reject a pattern if the maximum of posteriori probabilities is less than some threshold. For a classifier producing posterior probabilities, the following rule is applied to make reject decisions. \( \max_{k=1..m} P(Y|X) < T \), where \( P(Y|X) \) is the maximum of posterior probabilities among the \( m \) classes and \( T \) is the rejection threshold. For any fixed value of \( T \), the decision rule partitions the pattern space into two disjoint regions viz., acceptance region and rejection region. The pattern is rejected, if the classifier’s confidence is less than the specified threshold.

The research works focused on rejection framework in other applications are briefly described in this section followed by our proposed approach for URL classification problem. A classifier with reject option was introduced by Chow (1957) for character recognition problems, where the degree of ambiguity in the input patterns was examined and a decision of rejection is made when the pattern is too noisy and corrupted. Trapeznikov et al. (2013) applied rejection decision in each stage of a multi-stage classifier to increase the reliability of the automated system in medical diagnosis by adding different measurements and test results in every stage. In their approach, the initial stage classifier uses the features based on measurements that are easy to obtain and cheap. Libal (2013) has proposed an approach for pattern recognition of signals in which direct and multi-stage classifiers were used. The multi-stage method uses a multi-resolution representation of signal in wavelet transform. In their approach, at the first stage, the classifier is allowed to take decisions based on the coarse resolution of the signal, and more detailed resolutions are added in the subsequent stages of the multi-stage classifier. Maximum rejection classifier approach has been applied by Elad et al. (2001) to make face or non-face decision in an early stage of the classifier. Recently, for motion recognition in the domain of myoelectric control scheme, a confidence-based rejection method has been suggested by Scheme et al. (2013). They extended the idea of Rejection classifiers to Linear Discriminant Analysis
LDA) and shown that performance of LDA with Rejection(LDAR) is better than LDA for all the values of rejection threshold.

The URLs contain minimal information or in some cases no information at all about the category of the web page it points to. To gain an insight about the topic a web page, Baykan et al. (2011) used description of a web page also (called as snippets) along with URLs in training phase, which is available in the ODP dataset. They constructed a model by extracting features from both URLs and snippets and tried to classify the URLs by extracting URL features alone in the test phase. This method has not improved the result, as the terms (features) in the snippets are different from those terms in the URLs. And in real world situations, obtaining the title of the web page or description about the page requires at least partial downloading of the corresponding web page. So it may not be suitable for information filtering and focussed crawling applications.

Combining rejection framework to the URL classifier has been proposed in this thesis. By this approach, the URL classifier can be used as a first level filter in a multi-stage / hierarchical classifiers. We have used two classifiers, Naive Bayes classifier for performing multi-class classification and Support Vector Machine (SVM) for binary classification of URLs. In the Naive Bayes approach, by computing posterior probability and using Maximum A Posteriori (MAP) decision rule, the prediction of the given URL can be made and it can be classified into one of the k classes. As the posterior probability is a calibrated measure, the confidence scores need to be computed based on which rejection decision can be made.

SVM is a distance based classifier and classification decision can be made based on the distance score of the given input. As it would be difficult to choose a threshold based on distance score, we can use a threshold value between 0 and 1. So instead of using the distance score directly as a threshold, the probability estimate can be obtained from the distance score of SVM classifier using a logistic function.
The chapter gave an overview of basic research in web page classification, feature selection methods, machine learning algorithms used for web page classification. The different types of features used for web page classification has been discussed with a review of benchmark datasets used for research. The discussion of confidence based classification and its application areas are also presented.