ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

# SURVEY ON CLASSIFIER MODELS FOR TEXT CLASSSFICATION

**Archana Adiga*, Crystal Fiona Lobo, Naveen Chandavarkar, Dr. Balasubramani R**
* Department of Computer Science, NMAM Institute of Technology, Karnataka, India

**DOI:** 10.5281/zenodo.55305

**KEYWORDS:** KNN, Centroid, CenKNN, SVM, Naïve Bayes

## ABSTRACT
This paper gives a comparison of frequently used classifier models for text classification in the recent years. Text classification is the task of assigning pre-defined categories to free text documents. With the advent of technology we come across a large scale and high dimensional text corpus. It is a challenge to perform text classification on such corpus, in presence of imbalanced class distribution and noisy term features. There are a number of models for performing text classification with various degrees of success. We can simplify the task of text classification by using the following models: Support Vector Machine (SVM), k-nearest neighbor (KNN), Centroid (Rocchio), Naïve Bayes classifiers.

## INTRODUCTION
Classification is the task of assigning objects to one of the several predefined categories, a pervasive problem that encompasses many diverse applications. Classification is the task of learning a target function that maps each attribute set x to the one of the predefined class labels y. The target function is also known informally as classification model. For example: In a topic based text classification, documents are classified according to their class labels such as: scientific articles, news reports, movie reviews, and advertisements. Class labels can be defined on the way the text was created, the way it was edited, the register of language it uses, and the kind of audience to whom it is addressed. There are multiple sources for obtaining the text data like from the web, through newsgroups, bulletin boards, and broadcast or printed news which can have different formats, different preferred vocabularies and often significantly different writing styles.

Text classification is the primary requirement of text retrieval systems, which retrieves texts in response to a user query. It is an effective tool to organize the large volume of text data in electronic format such as email, webpages, blogs, scientific articles and news reports.

## LITERATURE SURVEY
Guansaog Pang. et. al[1] developed CenKNN: a scalable and effective text classifier which combines CentroidDR with K nearest neighbor classifier (KNN). It combines the best features of KNN and Centroid classifier. The idea behind this model is to use an effective and efficient class-centroid based dimension reduction method to substantially reduce dimensionality of documents, and then use the k-d tree structure to conduct a rapid K nearest neighbors search for KNN classification.

The K-Nearest-Neighbor (KNN) text classification algorithm is a popular instance based learning method. KNN is the most straightforward method to assign the majority class among the neighbors to the instance; while it is often more intuitive to assign higher weight to the nearer neighbors in the class assignment such as distance weighted voting methods. It is one of the most efficient classification methods. However, it is more expensive due to its lazy learning pattern, especially when classifying large-scale and high-dimensional document collections. This classification technique is sensitive to noise. KNN also suffers from the class imbalance problem due to majority voting classification scheme.

Centroid is a linear classifier. In Centroid, each class is represented by its centroid, and a test document is assigned to the class label of its closest centroid. Despite its simplicity, it outperforms many popular classifiers. But this model tends to be biased towards the majority class. Centroid can be plagued by modeling misfit problems when documents are not linearly separable.

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

In this paper, by combining KNN with efficient centroid-based text classification algorithm, a scalable and effective classifier model is proposed. This model combines the best feature of both the models and overcomes the shortcomings of both the model.

The proposed "Scalable and effective text classifier" system makes use of Vector Space Model (VSM) to represent the documents. Here, N set of training documents D= $\{(d_1, y_1), (d_2, y_2),\ldots..,(d_N, y_N)\}$, $y \in \{ C_1, C_2, ...., C_l\}$, a set of predefined classes. Each document can be represented by a term based vector $d_i = \{ t_1, t_2, ...., t_n\} \in R^n$ ,where weight vector of a document is calculated by the product of Term Frequency (TF) and Inverse Document Frequency (IDF).Here, n is the size of vocabulary (set of distinct terms) D.

In the proposed CenKNN, the high dimensional documents are projected into low dimensional documents using CentroidDR. CentroidDR computes the centroid of all the classes, and then maps the documents into the class-centroid-based space, using cosine similarity function. The class centroid can be represented by the formula:

$$\text{centroid}_j = \frac{1}{|C_j|} \sum_{d_i \in C_j} d_i \qquad (1)$$

Here, $\text{centroid}_j$ denotes the centroid of the class $C_j$ , j=1,2, ...., l. We then project documents using the formula:

$$u_i^{(j)} = \text{sim}(d_i, \text{centroid}_j) = \frac{(d_i, \text{centroid}_j)}{\|d_i\| \times \|\text{centroid}_j\|} \qquad (2)$$

Where s=1,2, ...., n; here s and j denotes the indices of dimensions for the original document di and projected data $u_i$ respectively. Details of CentroidDR are given below:

**Algorithm 1** (CentroidDR)

**Input:** Given a set of training documents D.

**Output:** The projected data $D^* = \{(u_1, y_1) , (u_2, y_2) , \ldots , (u_N, y_N )\}$, where $u_i \in R^l$ , $y_i \in \{C_1, C_2, \ldots, C_l\}$, for i = 1, 2, …, N.

1. Compute the centroid of each class via formula (1).
2. For each document, compute the similarities between the document and all of the centroids, and assign these similarity values to the dimension values of the projected data via formula (2).
3. Obtain the projected data $D^* = \{(u_1, y_1), (u_2, y_2), \ldots, (u_N, y_N )\}$. Each $u_i$ is projected from $d_i$. The coordinates of the original documents in the projected space are the similarities obtained in step 2, and their class labels are retained from their original documents.

Then k-d tree is used to store this projected data and search the K nearest neighbors to classify the documents. k-d trees are fast search trees that are widely used neighborhood search structures to speed up KNN classification. The proposed CenKNN algorithm, is as given in algorithm 2,

**Algorithm 2** (CenKNN)

**Input:** Given a set of training documents D and a test document $d_t$.

**Output:** The class label of document $d_t$.

1. Project the high dimensional documents D onto a class-centroid-based space via CentroidDR, and obtain the projected data $D^* = \{(u_1, y_1), (u_2, y_2), \ldots, (u_N, y_N )\}$.
2. Normalize the projected document vectors in $D^*$.
3. Build a k-d tree on the normalized data.
4. For the test document $d_t \in Rn$ , project it onto the class-centroid-based space, and normalize it, we then obtain $\hat{u}_t \in R^l$ .
5. Search for the K nearest neighbors of ût over the k-d tree.
6. Classify $\hat{u}_t$ based on the KNN decision rule as shown in (3).

$$y_t = \arg\max_{C_j} \sum_{\hat{u}_i \in KNN^{\hat{u}}_t} (1 - \text{dist}(\hat{u}_t, \hat{u}_i)) I(\hat{u}_i, C_j) \qquad (3)$$

Where $KNN^{\hat{u}}_t$denotes the set of K nearest neighbors of $\hat{u}_t$, dist ($\hat{u}_t$, $\hat{u}_i$) denotes the Euclidean distance between $\hat{u}_t$ and $\hat{u}_i$, and $I(\hat{u}_i, C_j)$ is the indicator function, which is 1 when $\hat{u}_i$ belongs to Cj otherwise 0.

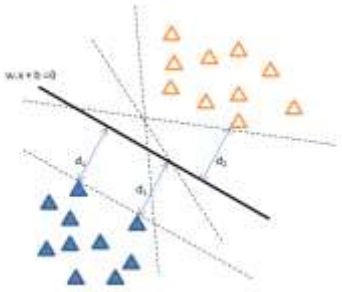
Lam Hong Lee et. al[2] developed Euclidean-SVM .In this type of text classification, the SVM approach is used in the training phase . It identifies the set of support vectors from each category. The Euclidean distance function is applied in the classification phase that computes the average distances between the testing data point and each

of the sets of support vectors from different categories. The Classification of different categories is decided based on the lowest average distance between its set of support vectors and the new data point.

The implementation of Structural Risk Management (SRM) principle determines the optimal separating hyper-plane; this guarantees high accuracy of the classifier. The equation of the hyper-plane that divides the data points in SVM is given in (4)

$$w.x + b = 0 \quad (4)$$

**Figure 1:**



***Optimal Separating hyper-plane***

According to the SVM methodology, there is always one optimal separating hyper-plane that lies half-way in between the maximal margin. The optimal separating hyper-plane is only determined by the closest data points of each category called as the Support Vectors (SVs). The maximal margin can be found by minimizing $½|w|^2$ as in (5)

$$\min \{½|w|^2\} \quad (5)$$

The partition of the non-linearly separable data points is done by implementing the kernel functions, by which the non-linearly separable data points transforms the data from its original coordinate space into a high dimensional feature space to separate the data points. The technique of the non-linear classification is similar to the linear classification, except that every dot product is replaced by a non-linear kernel function given in (6).

$$K(x_i, x_j) = (\phi(x_i), \phi(x_j)) \quad (6)$$

Mapping the data points into a high dimensional feature space using the kernel functions enhances the classification tasks as the SVM models separate the data points even with very complex boundaries.

The performance of the SVM classifier also depends on the soft margin parameter, C. The parameter C controls exchange between the margin and the size of slack variables. It creates a soft margin that allows some of the classification errors, especially when there are non-separable points in the classification. If the value of C is small, the number of training errors will increase due to under-fitting of the SV's and if the value of C is large, it will lead to over-fitting where high penalty for non-separable points occur and the classifier will behave like a hard margin SVM.

The SVM classification approach requires the selection of appropriate combination of parameter C and the kernel function. The optimization of the kernel function and parameters has to be incorporated into the training phase of the SVM to guarantee high accuracy of the classification task.

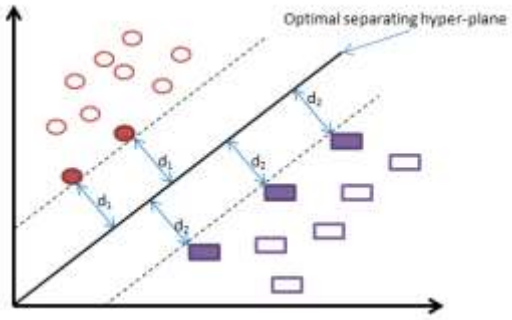
An enhanced classification framework for text classification is proposed by introducing the Euclidean distance measurement function to replace the optimal separating hyper-plane as the decision making function for the conventional SVM classification. This framework is known as Euclidean-SVM text classification framework.

Firstly, the SVM training algorithm reduces the training data points to the SV's by identifying and eliminating the training data points. Then in the classification phase, the Euclidean distance function is used to make the classification decision based on the average distance between the testing data point and each group of SV's from

different categories. The use of optimal separating hyper-plane is eliminated. The construction of the linear separating hyper-plane in high dimensional feature space is based on the SV's and the kernel function where the kernel function is incorporated to map the data points into a high dimensional feature space , so that the SV's are possibly separable by a linear separating hyper-plane . This affects the classification accuracy of the SVM.

Before proceeding to the training phase of the Euclidean-SVM text classification framework, a prepossessing approach is encountered wherein the text documents are transformed into a representable format for the SVM and the Euclidean-SVM format, which is in a numerical format. The pre-processing of the text documents is done by using the Bayesian vectorization technique. This transformation is necessary for the Euclidean-SVM text classification framework.

**Figure 2:**




***Vector space of the conventional SVM classifier with optimal separating hyper-plane***

There are two categories of training data points which represents the SV's of each category as shown in Figure 2. The optimal separating hyper-plane is constructed by maximizing the margin d1+ d2.The optimal separating hyper-plane is replaced by introducing the Euclidean distance function in making the decision for the classification task. After the SV's for each category are identified, they are mapped into the original vector space and the rest of the data points are discarded. In the classification phase, a new unlabeled data point is mapped into the vector space with the SV's and the average distance between the new data point and each SV's of different categories are computed using the Euclidean distance function. After computing the average distances, the new input data point will be selected with the category which has the lowest average distance between its SV's and the new data point . The algorithm of the Euclidean-SVM text classification framework is given below:

**Algorithm of the Euclidean-SVM text classification framework**

Pre-Processing Phase

1. Transform all the text documents (in both training set and testing set) into numerical format using the Bayesian Vectorization Technique.

Training Phase

1. Map all the training data points into the vector space of a SVM.
2. Identify and obtain the set of support vectors for each of the categories using SVM algorithm, and eliminate the rest of the training data points which are not identified as support vectors.
3. Map all the support vectors into the original vector space.

Classification Phase

1. Map the new unlabeled data point into the same original vector space with support vectors.
2. Use the Euclidean distance formula to calculate the average distances between the new data point and each of the sets of support vectors from different categories.
3. Identify the category which has the lowest average distance between its set of support vectors and the new data point.
4. Generate classification result for the new data point based on the identified category.

ISSN 2349-4506
Impact Factor: 2.545



With the combination of the SVM training algorithm and the Euclidean distance function; the impact of the kernel function and the parameter C on the classification accuracy of the conventional SVM can be minimized .We thus obtain an enhanced Euclidean-SVM classifier, while it is immune from the problem of determining the appropriate kernel functions and the parameter C.

Hiroshi Shimodaira[3] has compared two Naïve Bayes' models which are Bernoulli document model and Multinomial document model. Naïve Bayes classifier estimates the class-conditional probability by assuming that the attributes are conditionally independent for a class label y. Given a document D, the class is given by C. D can be classified as the class with the highest posterior probability P(C|D) , that can be re-expressed using Bayes' Theorem :

$$P(C|D) = \frac{P(D|C)\,P(C)}{P(D)} \propto P(D|C)\,P(C)\,. \tag{7}$$

A binary vector is used to represent a document in the Bernoulli documnet model. It is the best model for short documents. Given a vocabulary V containing a set of |V| words then the $t^{th}$ dimension of the document vector corresponds to word $w_t$ in the vocabulary. Consider the feature vector for the $i^{th}$ document $D_i$, $b_i$; then the $t^{th}$ element of $b_i$, written $b_{it}$ is either 0 or 1 representing the absence or presence of word $w_t$ in the $t^{th}$ document.

According to the naïve Bayes assumption, the probability of occurrence of each word is independent of the occurrences of the other words, then we can write the document likelihood $P(D_i|C)$ in terms of individuals word likelihoods $P(w_t|C)$ :

$$P(D_i|C) \sim P(\mathbf{b}_i|C) = \prod_{t=1}^{|V|} [b_{it}P(w_t|C) + (1 - b_{it})(1 - P(w_t|C))] \tag{8}$$

In the Multinomial document model, the document feature vectors capture the frequency of the words. It is superior to the Bernoulli's document model because it captures the frequency of words and not just the presence or absence .It is suitable for longer documents. If $P(w_t|C)$ be the probability of word $w_t$ occurring in class C that is estimated using the word frequency information from the document feature vectors.

According to the naïve Bayes assumption, the probability of occurrence of each word is independent of the occurrences of the other words, then we can write the document likelihood $P(D_i|C)$ as the multinomial distribution , where the numbers of draws corresponds to the length of the document, and the proportion of drawing item t is the probability of word type t occurring in a document of class C , $P(w_t|C)$.

$$P(D_i|C) \sim P(\mathbf{x}_i|C) = \frac{n_i!}{\prod_{t=1}^{|V|} x_{it}!} \prod_{t=1}^{|V|} P(w_t|C)^{x_{it}}$$
$$\propto \prod_{t=1}^{|V|} P(w_t|C)^{x_{it}}\,.$$

It can be used for both binary and multiclass classification problems.

## CONCLUSION

In CenKNN, the Vector Space Model used has some limitations. In this model long documents are poorly represented. The search keywords must exactly match the document terms. The order of the terms in the document is lost in the vector space representation. The k-d tree used in this model performs well on data of small dimensionality. The biggest limitation of the support vector approach lies in choice of the kernel. Another disadvantage is high algorithmic complexity and extensive memory requirements. The naïve Bayes algorithm affords fast, highly scalable model building and scoring. It scales linearly with the number of predictor. Thus, after the survey of various models our future work will be to design a naïve Bayes classifier model which is a combination of Bernoulli document model and Multinomial document model.

ISSN 2349-4506
Impact Factor: 2.545

Global Journal of Engineering Science and Research Management

**REFERENCES**

1. Guansong Pang , Huidong Jin , Shengyi Jiang, CenKNN: a scalable and effective text classifier, July 2014, Springer 2014
2. Lam Hong Lee , Chin HengWan , Rajprasad Rajkumar , Dino Isa, An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization August 2011, Springer Science+Business Media, LLC 2011
3. Hiroshi Shimodaira, Text Classification using Naive Bayes, 10 February 2015
4. S. M. Kamruzzaman, Farhana Haider, Ahmed Ryadh Hasan, Text Classification Using Data Mining ICTM 2005
5. http://www.scholarpedia.org/article/Text_categorization