ISSN 2349-4506
Impact Factor: 2.785

Global Journal of Engineering Science and Research Management

DATA MINING AND TEXT MINING: EFFICIENT TEXT CLASSIFICATION USING SVMS FOR LARGE DATASETS

Srikanth Bethu*, B Sankara Babu
* Asst. Prof, Department of Computer Science & Engineering Gokaraju Rangaraju Institute of Engineering and Technology, JNTU Hyderabad Hyderabad, India
Assoc. Prof, Department of Computer Science & Engineering Gokaraju Rangaraju Institute of Engineering and Technology, JNTU Hyderabad Hyderabad, India

DOI: 10.5281/zenodo.60657

KEYWORDS: Data mining, Dual problem, Learning algorithm, Primal problem, Support Vector Machines (SVM), Text mining.

ABSTRACT

The Text mining and Data mining supports different kinds of algorithms for classification of large data sets. The Text Categorization is traditionally done by using the Term Frequency and Inverse Document Frequency. This method does not satisfy elimination of unimportant words in the document. For reducing the error classifying of documents in wrong category, efficient classification algorithms are needed. Support Vector Machines (SVM) is used based on the large margin data sets for classification algorithms that give good generalization, compactness and performance. Support Vector Machines (SVM) provides low accuracy and to solve large data sets, it typically needs large number of support vectors. We introduce a new learning algorithm, which is comfortable to solve the dual problem, by adding the support vectors incrementally. It majorly involves a classification algorithm by solving the primal problem instead of the dual problem. By using this, we are able to reduce the resultant classifier complexity by comparing with the existing works. Experimental results done and produce comparable classification accuracy with existing works.

INTRODUCTION

Text classification [1] is the task of automatically classifying set of documents into categories from a predefined set. This task has several applications selective distribution of information to information consumers, spam filtering, and identification of document type. Automated text classification satisfies the need of manually organizing document. Text classification has gained importance because of the increased amount of documents over the years. Text documents needs to be categorized based on its contents.

The classification of the text is traditionally done by the term frequency and the inverse document frequency, this method has lot of problems. Term frequency has difficulty of classifying of documents in correct order due to occurrence of unimportant words in number of times and leads to wrong classification category. The text based on the “Tax” will explain the following example.

Example: 1.1 No filing of tax returns if salary is only income.

- “It is a relief for salaried class people in filling of tax and exempt them from filling tax returns who have source of income” said by Finance Minister.
- “Classes of persons” notification will be passed by government in requirement of producing income tax returns.
- The decision, on it will be effecting from August 1, 2012, will reduce compliance burden on initial tax payers, added by government.
- Since salaried taxpayers are not having other source of income will be excluded filing returns.
- "In last, the memorandum released by government says that, in case, if there is no source of income, there is no necessity of paying tax.

There are five paragraphs of the document. The five paragraphs are taken as input to system, and the word "tax" is taken as input for illustrating the computation of the distributional features. For computation, an array named "tax" is defined consisting of elements each corresponding to the paragraph.

Distributional Features

Count: The count [9] gives the number of occurrence of the particular word selected in the document. The calculation of count is done in two steps.

- An array "tax" is defined consisting of elements each corresponding to the paragraph. From the above paragraph example 1.1, if the word is present in the corresponding paragraph, each element of array is filled with the number of occurrences of the word, tax[c0, c1, c2, c3, c4] = tax[2,1,0,1,1].
- The elements of the array tax is summed the resulting value is count. Count is mathematically is defined as

$$\mathbf{{}_{i=0}\Sigma^{n-1}\ ci.}$$
$$\mathbf{Count\ is\ Count(tax,d) = {}_{i=0}\Sigma^{n-1}\ ci = 2+1+0+1+1=5.}$$

First appearance: The first appearance this measure gives where the word selected have first appeared in the document. The calculation of first appearance is done in two steps.

- An array "FirstApp" is defined consisting of elements each corresponding to the paragraph. Let's have reference from above paragraph, if the word tax have appeared in the paragraph then the paragraph number of the paragraph is entered in the element of the array, else the element of the array is filled with n-1.
- Minimum of the elements "FirstApp" array is taken as the first appearance. First appearance is mathematically is defined as

$$\mathbf{min_{i\varepsilon\{0.....(n-1)\}}\ {}^{ci>0}\ i : n}$$
$$\mathbf{FirstApp(tax,d) = min\{0,1,4,3,4\}=0.}$$

Compactness part number: The compactness part number measure gives how many parts of the document the word tax is present. The calculation of compactness part number is done in two steps.

- An array "ComPactPartNum" is defined consisting of elements each corresponding to the paragraph, if the word tax have appeared in the paragraph then number one is entered in the element of the array, else the element of the array is filled with 0.
- The sum of all the elements of the array "ComPactPartNum" is taken as the Compactness part number. Compactness part number is mathematically is defined as

$$\mathbf{{}_{i=0}\Sigma^{n-1}\ ci>0\ ,\ 1:0.}$$
$$\mathbf{ComPactPartNum(tax,d)=1+1+0+1+1=4.}$$

Last appearance: The last appearance this measure gives where the word selected have last in the document. The calculation of last appearance is done in two steps.

- An array "LastApp" is defined consisting of elements each corresponding to the paragraph. In the given example 2.1, if the word tax have appeared in the paragraph then number of that paragraph is entered, else -1 is entered in the corresponding element of the array.
- Maximum of the elements "LastApp" array is taken as the last appearance. Last appearance is mathematically is defined as

$$\mathbf{{}_{i=0}\Sigma^{n-1}\ ci>0\ ,\ i: -1.}$$
$$\mathbf{Last\ App(tax,d) = max\{0,1,-1,3,4\}=4.}$$

Compactness first and last distance: The compactness first and last distance measure gives the distance between a word's first and last appearance [9]. Compactness first and last distance is mathematically is defined as.

ComPactFLDist(tax,d)= LastApp(tax,d) - FirstApp(tax,d). ComPactFLDist(tax,d)=4-0=4.

Centriod:.The centriod measure will give the centre position of the word in the document. The calculation of centriod is done in two steps.

- The content of the array "tax" gives the occurrences the word "tax" in the corresponding paragraph. Each element of the array "tax" is multiplied by the corresponding paragraph number of the document. A constant variable Centroid(tax,d) is defined, to calculate the centroid.
- The obtained result of the first step is divided by the measure count calculated above.The result obtained is centriod.

  *i)/count(tax,d)

  Centriod is mathematically is defined as Centroid(tax,d)=(i=0 Σ n-1 ci.
  Centroid(tax,d)=(2*0+1*1+0*2+1*3+1*4)/5=
  8/5=1..

Compactness position variance: The compactness [7] position variance measure will gives the variance of the occurrence of the word tax from centroid. The calculation of compactness position variance is done in two steps.

- The content of the array "tax" gives the occurrences the word "tax" in the corresponding paragraph. Each element of the array "tax" is multiplied by the result of the subtraction of the corresponding number of the paragraph and centriod of the word "tax" . A variable "CompactPosVar(tax,d)" is defined.
- The obtained result of the first step is divided by the measure count calculated above. Compactness position variance is mathematically is defined as,

CompactPosVar(tax,d) = (i=0Σ $^{n-1}$ ci *|i-centroid(tax,d)|)/count(tax,d).

The set of objects from training data are inputs to these methods, the classes in which this objects called dependent variables and characteristics of objects are defined by set of variables known as independent variables. After a predictive model has begun, it is used to predict the class of the particular object in which class information is hidden known as priory. The main difference of supervised versus unsupervised (clustering) is knowing and having knowledge of classes and its different objects belonging to them.

SUPPORT VECTOR MACHINES (SVM) ALGORITHM

Support Vector Machine (SVM) can be defined as, it is a supervised machine learning algorithm which is used for classification and that completely satisfies regression challenges. SVM is mostly used in classification problems. In SVM algorithm, we can plot each data item as a point in n-dimensional space that is called 'n', which is number of features you have, with the value of each feature being the value of a particular coordinate. Then we are going to perform classification by detecting and finding the hyper-plane that differentiates the two classes very well in efficient manner. Support Vectors can be actively acts as the co-ordinates of individual observation.

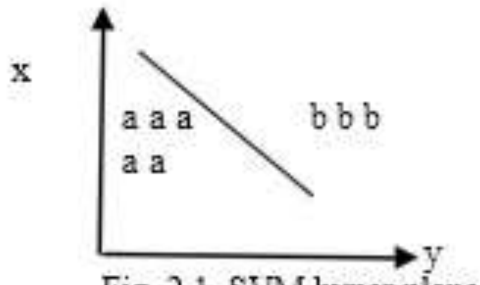
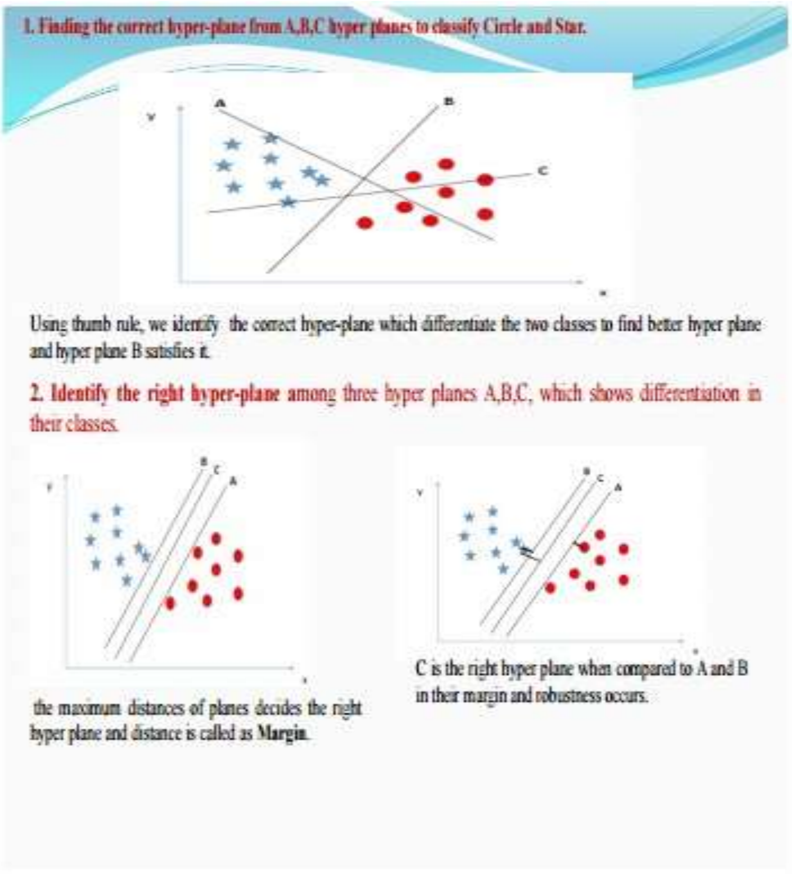


Fig .2.1. SVM hyper plane

Here a, b are called support vectors



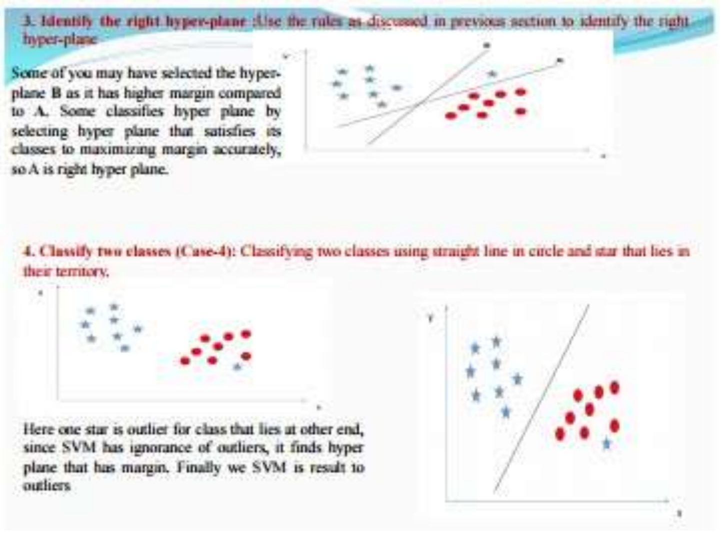
1. Finding the correct hyper-plane from A,B,C hyper planes to classify Circle and Star.

Using thumb rule, we identify the correct hyper-plane which differentiate the two classes to find better hyper plane and hyper plane B satisfies it.

2. Identify the right hyper-plane among three hyper planes A,B,C, which shows differentiation in their classes.

the maximum distances of planes decides the right hyper plane and distance is called as **Margin**.

C is the right hyper plane when compared to A and B in their margin and robustness occurs.



3. Identify the right hyper-plane :Use the rules as discussed in previous section to identify the right hyper-plane

Some of you may have selected the hyper-plane B as it has higher margin compared to A. Some classifies hyper plane by selecting hyper plane that satisfies its classes to maximizing margin accurately, so A is right hyper plane.

4. Classify two classes (Case-4): Classifying two classes using straight line in circle and star that lies in their territory.

Here one star is outlier for class that lies at other end, since SVM has ignorance of outliers, it finds hyper plane that has margin. Finally we SVM is result to outliers

Fig. 2.2. SVM different scenario cases

In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a p-1-dimensional hyper plane. The best choice to find right hyper plane is to represent its largest separation or margin in between two classes. So we can choose hyper plane which satisfies it from to its nearest data point, which has maximum side. This hyper plane is known as maximum – margin hyper plane and linear classifier is known as maximum – margin classifier.

The learning ability and computational complexity of training support vector machines are independent in dimension of the feature space, and reducing its complexity is needed to handle a large number of terms in practical applications of text classification. The dimension reduction method [4] is introduced to reduce the dimension of the document vectors. Experimental results shows that several dimensional reduction methods are designed particularly for clustered data, higher efficiency in which, both training and testing can be achieved without sacrificing prediction accuracy, even when the dimension of the input space is significantly reduced. The dimension reduction method has some limitations, important dimensions may be eliminated because of which the accuracy of the process may be affected. The complexity of the process may increase because of the lot of processing has to take place.

Related work: SVMs separable case – Soft margin classification

We define a Priori based on the construction of a decision rule to classify examples into one of two classes that based on a training set of examples. The Support Vector Machines (SVM) constructs a decision surface that bisects the two categories and maximizes the margin of separation between two classes of points. This can be used as a basis for classifying points of unknown class.

Suppose we have N training data points f(x1; y1); (x2; y2); : : : ; (xN; yN)g where xi 2 Rd and yi 2 f+1. The problem of finding a maximal margin separating hyper plane can be written as

- **Min w,b 1/2 w^T w subject to y i (w^T x i –b)**

This is a convex quadratic programming problem. Introducing Lagrange multipliers α and solving to get the Wolfe dual, we get:

- **Maximize α£ D≡ $\sum^N$i=1 α - ½ ∑i,j αiαjyiyjxi . xj subject to α ≥0, ∑I αi yi ≡ 0**

The Primal Problem solutions are given as

- **W= $\sum^N$ i=1 αi yi Xi**

To train the SVM, we search through the feasible region of the dual problem and maximize the objective function. Refer figure
2.1. The equations of the three hyper planes:

- **Optimal hyper plane H : y = w . x - b = 0**
- **Supporting hyper planes - parallel and equidistant to optimal hyperplane**

H1 : y = w . x - b = +1
H2: y = ((w. x) – b) = -1
H1

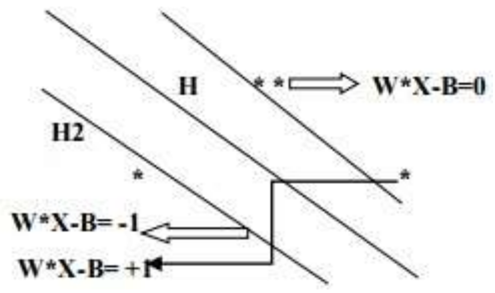


Fig. 2.3. Linear support vector machine example

Fig.2.3. explains about separation examples like star * and square boxes using Hyper planes. The classifications which occur with α > 0 are termed as support vectors, which actively participate in the defining of optimal separating hyper plane to remove other examples.

The classification of a new object x will be as follows

f(x) = sgn (w . x – b)
= sgn ((∑N i=1 α i x i y i (x i y i) . x – b
= sgn (∑ N i=1 α i y i ((x i . x) – b)

SVMs - Non-Separable case - Soft Margin Classification

As we know very large datasets invariably have noisy data points and there will be no linear separation in the input space. In this case, we will have training examples between the two supporting hyper planes. There will be a need of way to tolerate noise and outliers, and take into consideration the positions of more training points than just those closest to the boundary. We would like to reduce the constraints of the primal problem, by introducing non-negative slack variables (£i($1 \leq i \leq N$)) in the constraints.

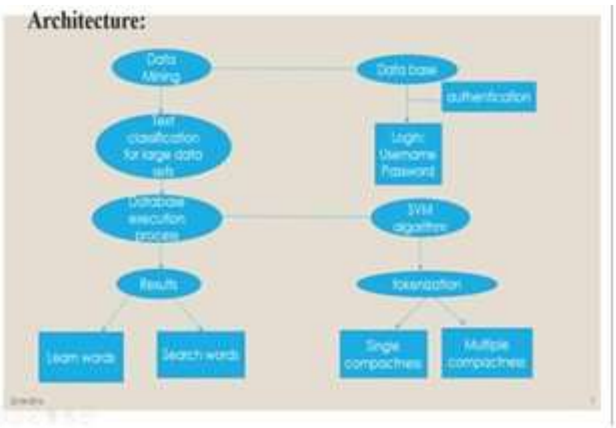
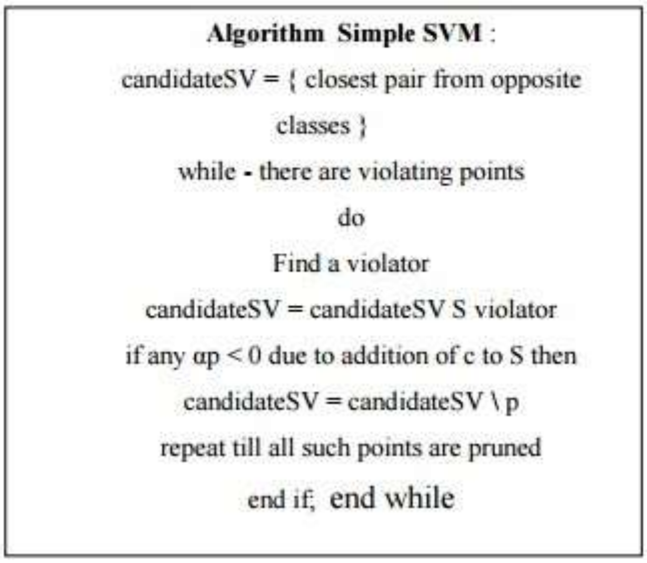


Fig. 2.4.Architecture showing execution of text classification using SVM algorithm

Here, the text classification is done through support vector machine algorithm (SVM). Authentication is connected to database, there the user can login by his/her username and password. Then, tokenization is done by selecting the single document or multiple documents. Further we get result by learn words in which where we can add and delete a word in a document. Later, coming to search words we can search the word whether it is present in the document or not.



```
Algorithm Simple SVM :
candidateSV = { closest pair from opposite
classes }
while - there are violating points
do
Find a violator
candidateSV = candidateSV S violator
if any αp < 0 due to addition of c to S then
candidateSV = candidateSV \ p
repeat till all such points are pruned
end if; end while
```

Text box 2.1: Sample SVM algorithm

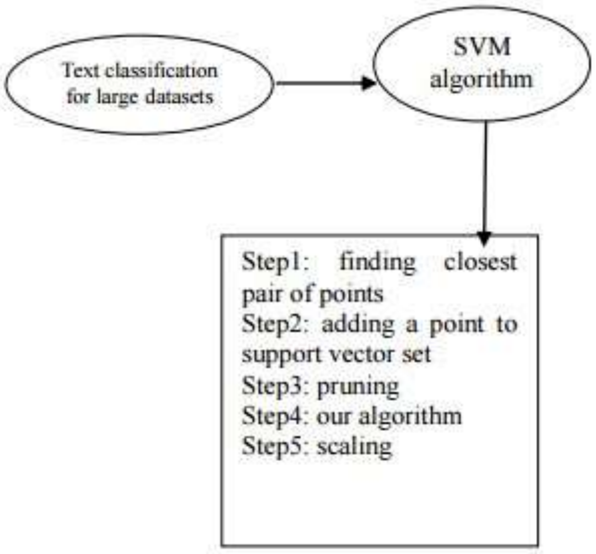

Text classification for large datasets
SVM algorithm
Step1: finding closest pair of points
Step2: adding a point to support vector set
Step3: pruning
Step4: our algorithm
Step5: scaling

Fig. 2.5. Text classification process

SVM algorithm generally has a Candidate Support Vector set, that initializes the Candidate set with the closest pair of points, from its opposite classes that is as same as like the SVM algorithm. After algorithm detects a violating point in the dataset and it immediately adds point to the Candidate set. The addition of the violating point as a Support Vector may be prevented by other candidate Support Vectors already present in the set can be happened. So we can simply deny and recover all such points from the Candidate set. Quadratic Penalty Formulation can be used to ensure the linear separability of the data points in the kernel space. To find the Closest Pair of Points, we observe that n=2 kernel computations are required to find closest pair of points in kernel space, where n indicated as the total number of data points. But, in case if we use a distance preserving kernel that is like the exponential kernel the nearest neighbours in the feature space are the same as the nearest neighbours in the kernel space, so that we need not perform any costly kernel evaluations for the initialization step. SVMs are defined as and can be a useful tool for insolvency analysis, for non-regularity case in the data. Suppose when the data are not regularly distributed or have an unknown distribution, it can help to evaluate information, i.e. financial ratios which should be transformed prior to entering the score of classical classification techniques.

EXPERIMENTAL RESULTS

The support vector machine algorithm is implemented by the traditional method that is using term frequency and the performance should be compared with the new developed algorithm of the support vector machine. The file consists of 30 documents, which need to be classified. The 15 documents are of institute for electrical and electronic engineers, related to computer science engineering. The remaining 15 documents are of conference papers of various topics like medical, civil engineering, etc.

Category set c= c1, c2	Expert judgment c1.c2
Classifier judgment	C1 TP =9 FP =9
	C2 FN= 5 TN = 10

Table 3.1: Contingency table for term frequency

Here C1 = Computer science documents. C2 = Non computer science documents.

TP derives true positive, FP derives false positive, FN derives false negative TN derives true negative.

The Table 3.1 shows contingency table for term frequency, values for the precision and recall are given below. Precision: It is fraction of retrieved documents that are relevant. P = T P / (T P+ F P). P = 9/(9+5) = 0.6. Recall : It is fraction of the documents that are relevant and are successfully retrieved. R = T P/ (T P+ F N). R =9/(9+5) = 0.64. The traditional method of classifying the documents by the term frequency gives the precision = 0.6 and recall = 0.64. Performance by the Support vector machine using compactness of the word. The 30 documents are taken as the example for the set of documents that need to be classified. The term frequency is the factor for the classification.

Category set C= C_1, C_2		Expert judgment	
		C_1	C_2
Classifier judgment	C_1	TP = 15	FP = 10
	C_2	FN = 0	TN = 12

Table 3.2: Contingency table for compactness

Here C1 = Computer science documents. C2 = Non computer science documents. Where TP defines true positive, FP defines false positive, FN defines false negative TN defines true negative.

The Table 3.2 shows contingency table for compactness, values for the precision and recall are given below. Precision: It is fraction of retrieved documents that are relevant. P = T P / (T P+ F P). P = 15/(15+10) = 0.6. Recall : It is fraction of the documents that are relevant and are successfully retrieved. R = T P / (T P+ F N). R = 15/(15+0) = 1. The traditional method of classifying the documents by the term frequency gives the precision = 0.6 and recall = 0.64. The Precision and recall value of compactness is calculated.

Fig 3.1 depicts the Support vector machine [5] classification result for distributional features. Corpus consists of 30 documents, 15 documents are of institute for electrical and electronic engineers, related to computer science engineering. The remaining 15 documents are of conference papers of various topics like medical, civil engineering, etc. The result is achieved and analyzed.

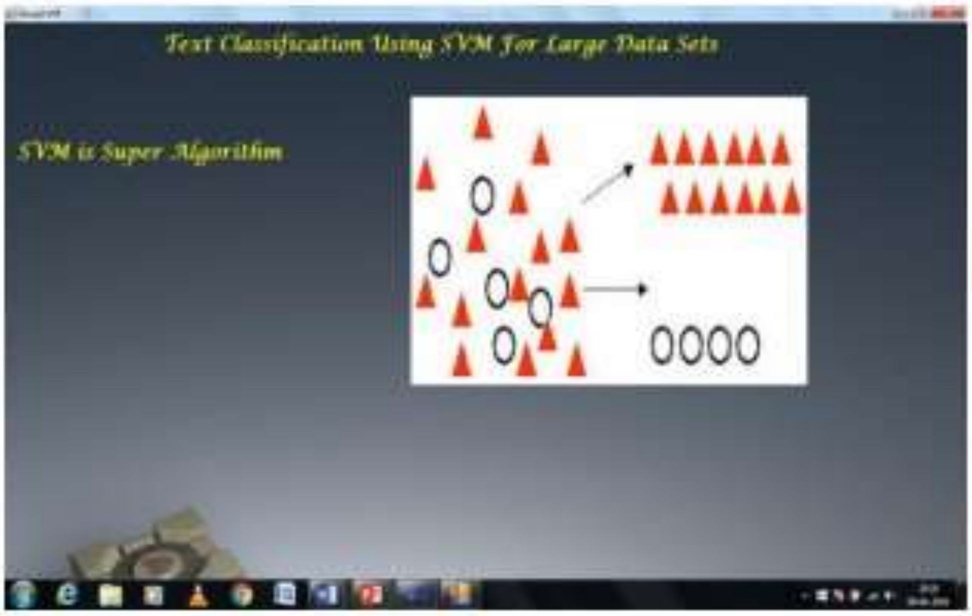


Fig. 3.1. Text classification process

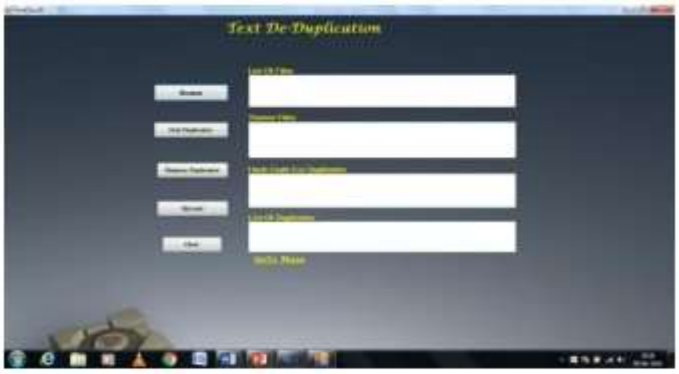


Fig.3.2. Text De-duplication process

The above fig. 3.1 and fig. 3.2 describes about the classification page where the user can browse and upload files. Using the classification page we can remove the duplicate files and store original files, it performs certain operations like find duplicates, remove duplicates, browse files, upload files, shows hash code with duplicates.

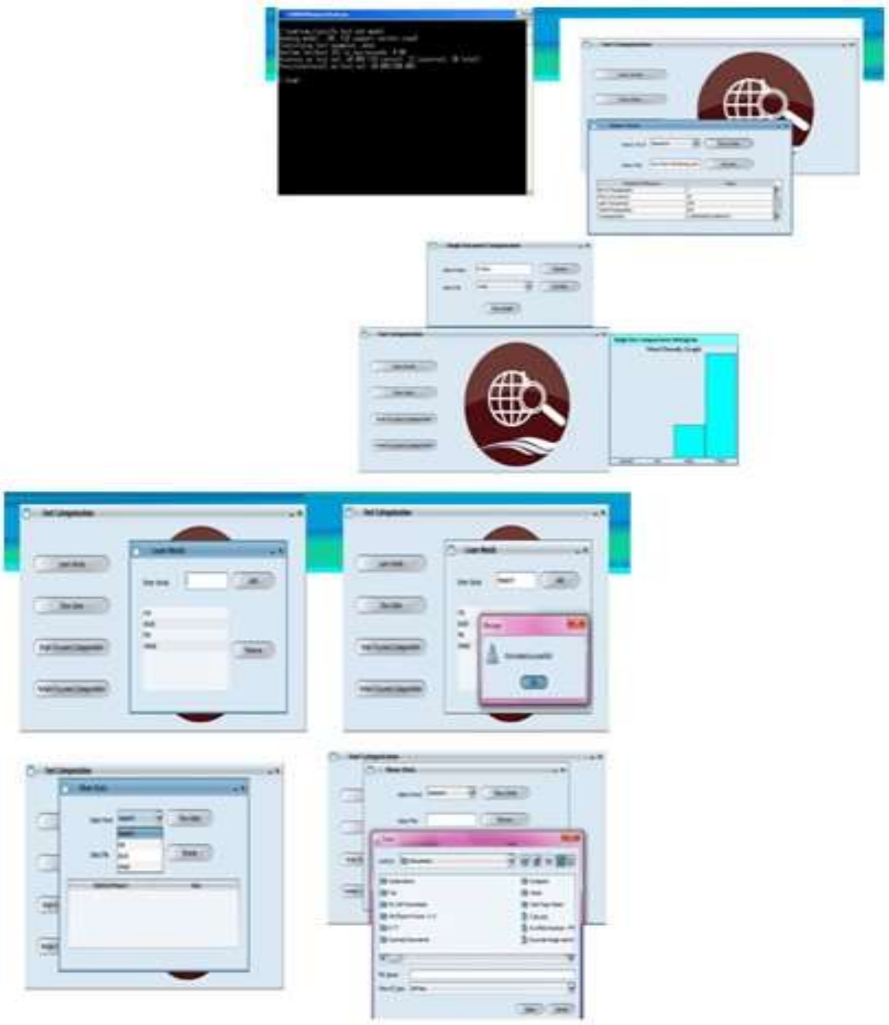


Fig.3.3. Checking the Text De-duplication classification The above fig. 3.3. Describes about running of the SVM on the system to perform classification of large data sets and puts them in a graph mode.

CONCLUSION

The algorithms that were developed for dual and primal problems, we can observe the achievement of less number of support vectors, and where the resultant classifier can classifies the unknown and unseen examples efficiently. We can also notice that, continuously solving the dual problem will give better solution and also to be observed is that, for all the cases, we have not worked with the complete (whole) dataset at one time at all. This analysis makes our algorithms scalable and efficient in terms of memory and there is no need of loading the full dataset in main memory

REFERENCES

1. B. Pfahringer and M. Sauban, "Text Categorization Using Document Profiling" Seventh European Conf Principles and Practice of Knowledge Discovery in Databases (PKDD '03), pp 411-422, 2003.
2. Chengqing Zong, and Chu-Ren Huang, Shoushan Li, Rui Xia, , "A Framework of Feature Selection Methods for Text Categorization" Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp 692–700, 2009.
3. F. Li and Y. Yang, "A Los Function Analysis for Classification Methods in Text Categorization," Proc. 20th Int'l Conf. Machine Learning (ICML '03), pp 472- 479, 2003.
4. Hyunsoo, Haesun Park, and Kim Peg Howland "Dimension Reduction in Text Classification with Support Vector Machines" Journal of Machine Learning Research vol 6, pp 1-17,(2005) .
5. Manu Konchady, "Text Mining Application Programming", Cengage Learning ,1st edition , pp 209 - 233, 2008.
6. X.-B. Xue and Z.-H. Zhou, "Distributional Features for Text Categorization" Proc. 17th European Conf. Machine Learning (ICML '06), pp 497-508, 2006.
7. Xiao-Bing Xue and Zhi-Hua Zhou "Distributional Features for Text Categorization" IEEE Transactons On Knowledge And Data Engineering, Vol. 21, No. 3, pp 428- 442, 2009.