A high performance Hybrid Algorithm for Text Classification

Prema Nedungadi, Haripriya Harikumar, Maneesha Ramesh
Amrita CREATE, Amrita University

***Abstract*—The high computational complexity of text classification is a significant problem with the growing surge in text data. An effective but computationally expensive classification is the k-nearest-neighbor (kNN) algorithm. Principal Component Analysis (PCA) has commonly been used as a preprocessing phase to reduce the dimensionality followed by kNN. However, though the dimensionality is reduced, the algorithm requires all the vectors in the projected space to perform the kNN. We propose a new hybrid algorithm that uses PCA & kNN but performs kNN with a small set of neighbors instead of the complete data vectors in the projected space, thus reducing the computational complexity. An added advantage in our method is that we are able to get effective classification using a relatively smaller number of principal components. New text for classification is projected into the lower dimensional space and kNN is performed only with the neighbors in each axis based on the principal that vectors that are closer in the original space are closer in the projected space and also along the projected components. Our findings with the standard benchmark dataset Reuters show that the proposed model significantly outperforms kNN and the standard PCA-kNN hybrid algorithms while maintaining similar classification accuracy.**

Keywords— Text classification; dimensionality reduction, PCA; kNN; Hybrid classifier; term weighting,

I. INTRODUCTION

There is an exponential growth of text information associated with the web technology and the internet. To gather useful information from it the text has to be categorized correctly and efficiently. Text Classification is the process of finding the correct class for each document, given a set of classes and a collection of text documents.

Content based classification is a type of classification in which the weight given to particular terms in a document determines the class to which the document is assigned. In the vector space model, documents are represented as vectors, where each entry corresponds to the weight of terms of the document. A popular unsupervised weighting scheme for text classification is the tf*idf (term frequency* inverse-document-frequency) [1, 2]. There are other effective supervised weighting schemes such as tf*rf, (term frequency * relevance frequency), TF-ICF (term frequency * inverse corpus frequency) [3] and so on. Effective Text Classification depends on both the right choice of the weighting scheme and the classification algorithm.

Applications of text categorization include descriptive type question answering system, search engines, automatic text scoring and recommendation systems.

There are two types of approaches to text classification, the rule based approach and machine learning based approach. In the rule based approach, the classification rules are defined manually and documents are classified based on rules. In the machine learning approach, the classification rules or equations are found automatically using sample labeled documents. This class of approaches has much higher recall but a slightly lower precision than rule based approaches but is more practical for big data. Therefore, machine learning based approaches are replacing the rule based approaches for text classification.

An effective but computationally expensive classification is the kNN text classification. PCA has been used as a pre-processing phase of KNN so as to reduce the dimensionality. However, the kNN computational cost, though reduced is still high as it uses every vector in the projected space.

This paper proposes a new hybrid text classification approach using principal component analysis to reduce the dimension, and applying kNN only for neighboring vectors in the principal components thus reducing the inputs for the kNN classifier. We show that our hybrid model is able to classify data with a high level of accuracy while using a smaller number of principal components thus significantly reducing the computational time.

II. RELATED WORK

Recently many different types of machine learning classification techniques [4,5] such as Naïve-Bayes, Rocchio, SVM, K-nearest neighbor, centroid-based classification [6,7] , Markov models [8], and hybrid [11,13,18,19] versions have been proposed.

A. Classification Algorithms

Naïve-Bayes [4, 5, 9, 10] is a probability based classification technique. It works based on the Principal of Bayes' theorem. This algorithm computes the posterior probability of the document belonging to different classes and assigns the document to the class with the highest posterior probability. But one of the major limitations of the classifier is that it performs very poorly when features are highly correlated, since it assumes class conditional independence.

978-1-4799-2259-14/$31.00©2014

Rocchio's algorithm [4, 5, 11] makes use of the centroid of a class to predict the class of a newly arrived document. It finds a prototype vector (centroid or mean value) for each class or category. Although it is faster and easier to implement, the classification accuracy is comparatively lower and influenced by outliers as we only consider the mean data for classification.

The idea of SVM [13, 14] is to find linear separators. It is an efficient technique for classification. But SVM kernel functions are used for solving non-linear separable problem and for multiclass problem, it is computationally expensive.

kNN [4,5] is a classification algorithm where objects are classified by voting several labeled training examples with their smallest distance from the object. This method performs well even in handling the classification tasks with multi-categorized documents but requires much more time for classifying objects with either a large number of training examples or with high dimensional data. In order to classify a new data, we need to loop over all the training examples to find its nearest neighbors. The need to store all training examples leads to more storage requirement. Online response is typically slower and it is difficult to find an optimal value of k. In spite of this, kNN is one of the most commonly used and accurate algorithms used for classification.

Principal component analysis (PCA) [15,16] is a mathematical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. It is useful for reducing the dimensionality of a high dimensional data.

In [17], the centroid-based classification method is used for the language identification problem. The high run-time performance of centroid-based classification is quite useful for on-line tasks like language identification. However, centroid based classification is affected by smaller variations in data and by outliers.

B. *Hybrid Algorithms*

A few hybrid versions of classification algorithms include an improved KNN algorithm for text categorization [18] builds the classification model by combining constrained one pass clustering algorithm and KNN text categorization. During the clustering processes, each cluster is represented as a cluster vector in accordance with the centroid vector for each cluster. But this is used for unsupervised learning.

A hybrid algorithm based on variable precision rough set is proposed to combine the strength of both kNN and Rocchio techniques and overcome their weaknesses [11]. Rocchio classifiers are used to classify most of the new documents effectively and efficiently. They considered binary text classification that assigns each document either to the positive class or to its complement negative class.

The SVM-kNN hybrid classification approach [13] has the objective to minimize the impact of parameter on classification accuracy. The SVs from different categories are used as the training data of the nearest neighbor classification algorithm in which the Euclidean distance function is used to calculate the average distance between the testing data point to each set of SVs of different categories. But SVM has some drawbacks when dealing with non-linear separable problem and multiclass problem.

Our goal is the take advantage of the strengths of the kNN algorithm but reduce the computational complexity. There have been models proposed that perform some pre-processing such as PCA before kNN.

An effective strategy to accelerate the standard kNN was proposed by projecting each category onto one principal component [19]. In this case, a new vector has to be projected into multiple principal components, one for each class thus increasing the computational time. They compute the principal component for each class separately and then use the first principal component from each PCA result.

Our model differs from [19] in that we perform only one PCA, and the number of principal components selected is independent of the categories. We show that we significantly reduce the classification time while maintaining the accuracy.

III. System Architecture

When a user is given a text data as input, the hybrid text classification algorithm performs the classification in an efficient way and predicts the class of the given data, shown in Figure 1.

PCA is used as a preprocessing phase as we are dealing with high dimensional data. kNN is used for predicting the label of the new data.

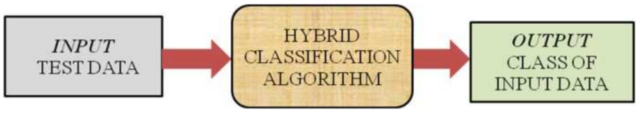

Figure 1. Overall Architecture: Hybrid classification algorithm

Algorithm 1 is our main algorithm for performing the hybrid text classification.

A. *Algorithm - PCACLASSIFIER ( )*

```
Input: {Training dataset with class, Test data}
Output: {Class of Test data}

1. Compute the principal components of the training data.
2. Project the train data along each principal component.
3. Project the test data along each principal component.
```

4. Perform binary search over each projected space and find the L nearest neighbors.
5. Compute the similarity between test data and its neighbours in the projected space.
6. Select the most similar *k* neighbors.
7. Predict the category of test data based on the selected neighbors

The input to this algorithm is the training data along with the class information and the test data. Dimensionality reduction for high dimensional data is performed using Principal Component Analysis (PCA). The dimensionally reduced data in the projected space is used for classification. We limit the kNN space by finding the L nearest neighbors along each principal component in the dimensionally reduced feature space. The data that is clustered in the original space will also be closer in the projected space and along each of the principal component axis

Figure 2 shows the projected space of the newly arrived data with its neighbors along the first principal component. There will be additional neighbors along the principal component that are not neighbors to the new data, and kNN will discard these.

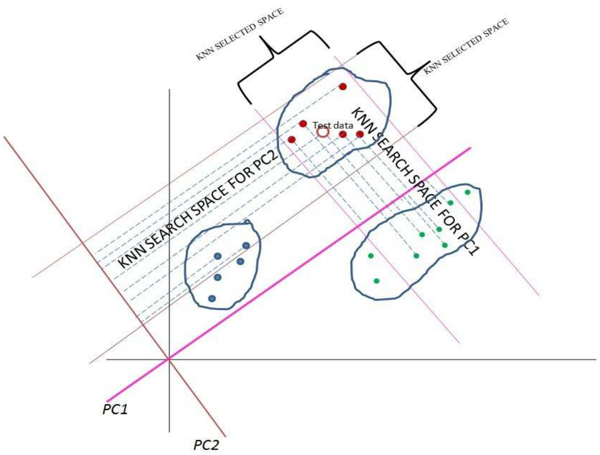

Figure 2. Reduced search space for kNN

Figure 3 depicts the training phase of the hybrid algorithm. The input set contains the training data along with its class information and the test data. PCA is used here to reduce the number of features of the training dataset.

The projected value along each principal component and its corresponding index are stored in a two dimensional array. The index is later used for retrieving the original data as well as the projected data and finding neighbors along the axis. As the array is sorted, the binary search finds the closest neighbors with a complexity of O (log n), where n is the size of the array.

With our model we not only reduce the dimensionality of the vectors with PCA, but also reduce the search space by limiting it to the data that corresponds to the close neighbors along the axis and the time to find the close neighbors using binary search.

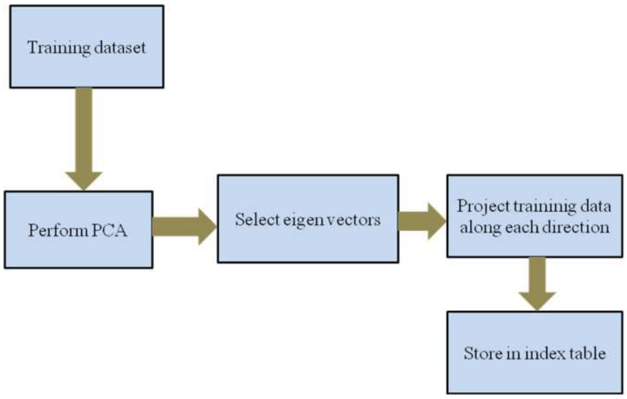

Figure 3. Training phase of Hybrid classification algorithm

Figure 4 shows the testing phase of our proposed model. When a new data arrives we project the newly arrived data along the principal components. Traditional kNN calculates similarity with the entire training set. In order to reduce the classification time of kNN we proposed projecting the vector along each component and selecting the L nearest neighbors along each principal component. These neighbors are the input dataset for kNN. So instead of searching in the entire training data and the original feature space the search space is reduced.

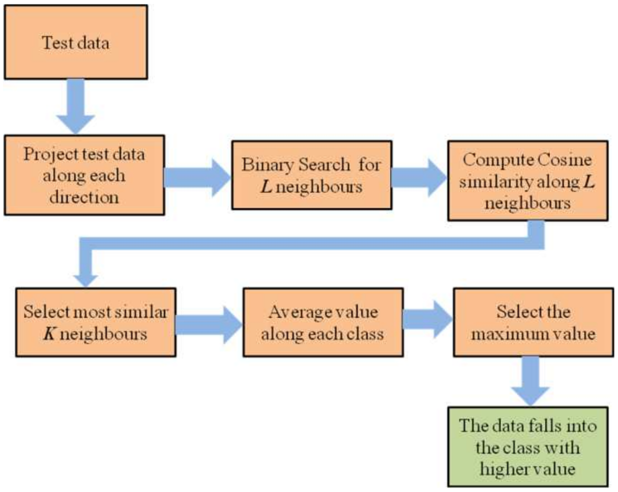
Test data
Project test data along each direction
Binary Search for *L* neighbours
Compute Cosine similarity along *L* neighbours
Select most similar *K* neighbours
Average value along each class
Select the maximum value
The data falls into the class with higher value

Figure 4. Testing phase of Hybrid classification algorithm

Once we find the closet set of neighbors along each principal component, kNN is performed with this limited set of vectors and the k nearest neighbors are selected. The prediction of the class of the newly arrived data is based on the most similar *k* neighbors in the selected subset.

We may perform kNN based on the original vectors or based on vectors in the projected vectors. kNN in the original space is slightly more expensive as the vectors are larger, but has higher accuracy.

The complexity of the main hybrid classifier algorithm is **$O(m*n) + O(m) + m_{test}*[\ O(p) + O(e*Lhalf) + O(e*k) + O(e) + O(t)] + 2*O(e)$,** where m is the number of instances, n the number of features and e the number of principal component selected, *Lhalf* is half of L(searching for L neighbors along each component), m_{test} is the number of test data, $p = a * L$, in which $a = e*b$, a and b are constants. t is the index of the class with maximum value. $O(t)$ is a constant except in situations where more than one class has the same maximum value.

From the computed time complexity we can say that our proposed algorithm depends to a large extent on the number of principal components (Eigen vectors) selected. The initial $O(m*n)$ is for reading the entire data and for the remaining computations the reduced feature space is used. Our proposed model has reduced the time and space complexity compared to the existing models.

IV. Experimental Evaluation

A. Datasets

We used both synthetic and real dataset Reuters dataset with a total of 6532 documents, 5741 features belonging to 52 categories.

B. Analysis by varying the number of instances on synthetic dataset

We tested the execution time of the Hybrid Classifier by varying the number of instances while keeping the number of features at1000, L as 7 and K as 5 Figure 5 shows that the graph is directly proportional to the number of instances.

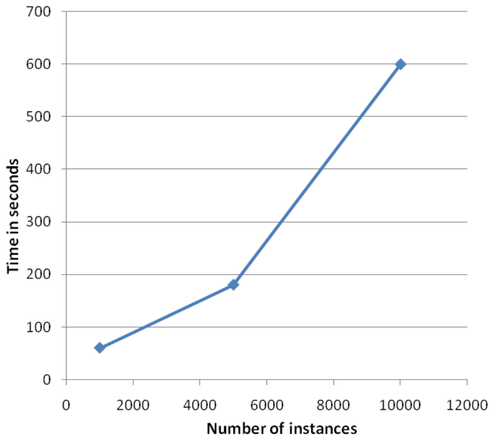

Figure 5. Varying number of instances and L=7, K=5 and features = 1000

The percentage of accuracy of classification as well as the time taken to predict test datasets on Reuters data is shown in Table I.

TABLE I. CLASSIFICATION ACCURACY AND TIME TAKEN BY 1000 REUTERS TEST DATA

K	Correctly classified (Accuracy in %)	Time in seconds
5	94.2	537
10	91.0	569
20	87.5	560
30	86.1	595
40	86.0	651
50	85.0	660

We used 1000 test data and 6532 training data with 5741 features. For the Reuters data, we achieve 94% accuracy when k=5 suggesting that this is an optimal value of *k* for this dataset.

TABLE II. CLASSIFICATION ACCURACY OF REUTERS DATA WHEN USING ORIGINAL AND PROJECTED DATA

K	Classification Accuracy (Accuracy in %)	
	Projected data	*Original data*
5	94.2	98.5
6	93.7	98.5
7	91.7	98.2
8	91.3	98.5
9	90.6	98.0
10	91.0	98.0

Next we compared the classification accuracy by performing the last step in our hybrid classifier with the projected data and the original data with varying the K values as shown in Table II. As expected, the accuracy is relatively higher when using the original data instead of the projected data for similarity computation. For the projected data, classification accuracy is better for smaller values of k while for the original data it was nearly the same.

TABLE III. TIME TAKEN FOR CLASSIFYING A REUTERS TEST DATA FOR VARIOUS APPROACHES IN SECONDS

K	Traditional kNN	PCA+kNN	PCA+Indexed kNN(Hybrid)
5	4613	44.519	20.0176
7	4674	44.767	20.336
9	4796	44.373	20.067

The time comparison using Reuters data with traditional kNN, PCA preprocessed kNN and our Hybrid kNN is shown in Table III. Our hybrid classifier outperformed the PCA+kNN approach In PCA preprocessed kNN we used PCA for dimensionality reduction.

Figure 6 shows the time taken for the hybrid kNN both for original as well as projected data and that the time for kNN in the projected space is less than in the original space.

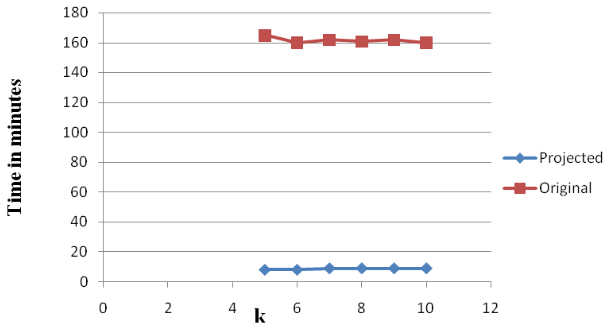

Figure 6. Computational Varying K and fixed L as 50 on Reuters data

Figure 7 shows the time comparison by varying L and performing kNN on the subset of the original and the projected data. The time taken for projected data is less compared to original. As L increases, the subspace for the kNN search increases and we see an increase in the time. As the dimensions in the original space is much larger, the time for kNN similarity comparisons increases at a faster rate for kNN performed the original date

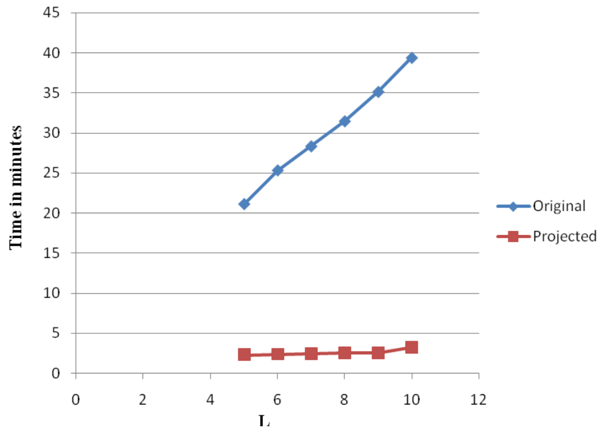

Figure 7. Varying L and fixed K as 5 on Reuters data

The classification accuracy of real as well as original data by varying the *L* value is shown in Table IV. As expected the accuracy is high in case of original data.

TABLE IV. CLASSIFICATION ACCURACY OF REUTERS DATA WITH ORIGINAL AND PROJECTED DATA BY VARYING L

L	**Classification Accuracy (Accuracy in %)**	
	Projected data	*Original data*
5	88.8	99.6
6	90.3	99.5
7	89.8	99.5
8	90.3	99.5
9	90.6	99.5
10	90.6	99.3

The classification accuracy and the time taken by PCA+kNN and our proposed Hybrid version is shown in Table V. This PCA+kNN is an approach in which PCA is used only as a preprocessing step for dimensionality reduction and then the traditional kNN is performed in the projected space for classification. We used a subset of the Reuters dataset for this comparison with a testing dataset of size 100 and a training dataset of size 6532. Our results are positive and show that a significant reduction in time and a small improvement in the classification accuracy.

TABLE V. TIME AND ACCURACY COMPARISON

K=5	**PCA+kNN**	**PCA+Indexed kNN(Hybrid)**
Time	29 minutes	1 minute
Accuracy	96%	97%

VI. CONCLUSION

The kNN classifier is a popular classifier and is highly effective. One of the major limitations of kNN is its classification time when dealing with high dimensional as well as big data.

In this paper, we proposed an efficient hybrid algorithm based on PCA to reduce the dimension and a novel way to find the neighbors along each principal component so as to restrict the kNN space and increase the efficiency. We use the projected space of the original data for classification and showed that with the standard Reuters dataset, our hybrid model is able to classify data with a high level of accuracy while using a small number of principal components and a smaller set of projected data for kNN thereby significantly reducing the computational time. Future work will include data analysis with various term weighting schemes and additional datasets to understand the effectiveness of these with our hybrid algorithm.

ACKNOWLEDGMENT

This work derives inspiration and direction from the Chancellor of Amrita University, Sri Mata Amritanandamayi Devi.

REFERENCES

[1] Charles Elkan, Deriving TF-IDF as a Fisher Kernel, Springer-Verlag Berlin Heidelberg , 2005

[2] Man Lan, Chew Lim Tan, Jian Su, and Yue Lu, Supervised and Traditional Term Weighting Methods for Automatic Text Categorization, IEEE transactions on pattern analysis and machine intelligence, vol. 31, no. 4, april 2009

[3] Reed, J.W., Jiao, Y., Potok, T.E., Klump, B.A., Elmore, M.T., Hurson, A.R., 2006. TF-ICF: A new term weighting scheme for clustering dynamic data streams. In: Proc. Internat. Conf. on Machine Learning Applications (ICMLA), Orlando, p. 258–263

[4] Pratiksha Y. Pawar and S. H. Gawande, *Member,* IACSIT, A Comparative Study on Different Types of Approaches to Text Categorization, International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012

[5] Charu C. Aggarwal , ChengXiang Zhai ,A Survey of Text Classification Algorithms, Chapter 6: MINING TEXT DATA

[6] Tan, S., 2008. An improved centroid classifier for text categorization. Expert Syst. Appl. 35 (1–2), 279–285.

[7] Kruengkrai, C., Srichaivattana, P., Sornlertlamvanich, V., Isahara, H., 2005. Language identification based on string kernels. In: Proc. 5th International Symposium on Communications and Information Technologies (ISCIT), Beijing, pp. 896–899.

[8] Xafopoulos, A., Kotropoulos, C., Almpanidis, G., Pitas, I., 2004. Language identification in web documents using discrete HMMs. Pattern Recognition 37 (3), 583–594.

[9] Wenyuan Dai, Gui-Rong Xue, Qiang Yang Yong Yu,Transferring Naive Classifiers for Text Classification, Association for the Advancement of Artificial Intelligence (www.aaai.org),2007

[10] S.L. Ting, W.H. Ip, Albert H.C. Tsang , Is Naïve a Good Classifier for Document Classification?, International Journal of Software Engineering and Its Applications Vol. 5, No. 3, July, 2011

[11] Duoqian Miao , Qiguo Duan, Hongyun Zhang, Na Jiao "Rough set based hybrid algorithm for text classification, Expert Systems with Applications 36 (2009) 9168–9174

[12] Yun-Qian Miao, Mohamed Kamel , Pairwise optimized Rocchio algorithm for Text Categorization, Elsevier, 2010

[13] Chin Heng Wan, Lam Hong Lee, Rajprasad Rajkumar, Dino Isa, A hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbour and support vector machine, Expert Systems with Applications 39 (2012) 11880–11888

[14] Christopher J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Kluwer Academic Publishers, Boston,1998

[15] I.T.Jollife, Principal Component Analysis, Second Edition,Springer, 2002.

[16] Lindsay I Smith, A tutorial on Principal Components Analysis, February 26, 2002

[17] Hidayet Takc, Tunga Gungor, A high performance centroid-based classification approach for language identification, Pattern Recognition Letters 33 (2012) 2077–2084

[18] Shengyi Jiang, Guansong Pang, Meiling Wu, Limin Kuang , An improved K-nearest-neighbour algorithm for text categorization, Expert Systems with Applications 39 (2012) 1503–1509

[19] Min DU, Xing-shu CHEN,Accelerated *k*-nearest neighbours algorithm based on principal component analysis for text categorization,Journal of Zhejiang University-SCIENCE C (Computers & Electronics) 2013-14