International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 3, May - 2012

Improved kNN Algorithm by Optimizing Cross-validation

Ms. Soniya S. Dadhania
M.E Computer Science and Engineering

Prof. J. S. Dhobi
Head of Computer Department,GEC,Modasa

Abstract

Nowadays web applications based on short text is increasing rapidly. Moreover, the classification algorithms which are applied to short text data are Support Vector Machines algorithm, k-Nearest Neighbors algorithm and Naive Bayes algorithm. kNN algorithm depends on the distance function and the value of k nearest neighbor. Traditional kNN algorithm can select best value of k using cross-validation but there is unnecessary processing of the dataset for all possible values of k. Proposed kNN algorithm is an optimized form of traditional kNN by reduceing the time and space for evaluating the algorithm. Experiments are performed in developer version of weka 3.7.5.Comparison of proposed kNN algorithm is done with traditional kNN algorithm, Support vector machine and Naïve Bayes algorithm. The proposed algorithm is more promising than the traditional kNN algorithm as time taken to process and space used for cross-validation in classification are reduced.

1. Introduction

The increasingly important role played by short texts in the modern means of Web communication and publishing, such as Twitter messages, blogs, chat massages, book and movie summaries, forum, news feeds, and customer reviews, opens new application avenues for text mining techniques but it also raises new scientific challenges. Although text classification and clustering are well established techniques, they are not successful in dealing with short and sparse data, because standard text similarity measures require substantial word co-occurrence or shared context.

Text classification is a learning task, where pre-defined category labels are assigned to documents based on the likelihood suggested by a training set of labeled documents. Text categorization methods proposed in the literature are difficult to compare. Datasets used in the experiments are rarely same in different studies. Even when they are the same, different studies usually use different portions of the datasets or they split the datasets as training and test sets differently. Moreover, classifications will be performed using Support Vector Machines, k-Nearest Neighbors and Naive Bayes. For the analysis and comparison of different results, precision, recall and F-measure are used.

KNN is a typical example of lazy learning. Lazy learning simply stores training data at training time and delays its learning until classification time. In contrast, eager learning generates an explicit model at training time. k-NN algorithm classifies a test document based on it k nearest neighbour. The training examples can be considered as vectors in a multidimensional feature space. The space is partitioned into regions by locations and labels of the training samples. A point in the space is assigned to a class in which most of the training points belong to that class within the k nearest training samples. Usually Euclidean distance or Cosine similarity is used. During the classification phase, the test sample (whose class needs to be identified) is also represented as a vector in the feature space. Distances or similarities from the test vector to all training vectors are computed and k nearest training samples is selected. There are a number of ways to classify the test vector to a specific class. The classical k-NN algorithm determines the class with the majority voters from its k-nearest neighbours [2].

2. Scope of improvement in kNN

Although kNN has been widely used for decades due to its simplicity, effectiveness, and robustness, it can be improved according to the application. Improvement can be done on following parameters.

(1) **Distance Function**: The distance function for measuring the difference or similarity between two instances is the standard Euclidean distance.

(2) **Selection of Value K**: The neighborhood size is artificially assigned as an input parameter.

(3) **Calculating Class Probability**: The class probability estimation is based on a simple voting.

3. Proposed kNN algorithm

Distance function

kNN algorithm depends on the distance function used for calculating the distance between input test object and objects in training set. To measure the distance of data in the kNN, the distance function is important The most commonly used function is the Euclidean distance function (Euclid), which measures two input vectors (one typically being from a stored instance, and the other being an input vector to be classified). One weakness of the Euclidean distance function is that if one of the input attributes has a relatively large range, then it can overpower the other attributes. Therefore, distances are often normalized by dividing the distance for each attribute by the range (i.e., maximum-minimum) of that attribute. The cosine similarity is commonly used in text classification [15].

In proposed algorithm cosine similarity function applied instead of Euclidian distance but the results are found similar both distance functions.

Cosine Similarity

Given two vectors of attributes, A and B, the cosine similarity, θ, is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

For text matching, the attribute vectors A and B are usually the term frequency vectors of the documents. The cosine similarity can be seen as a method of normalizing document length during comparison. In the case of information retrieval, the cosine similarity of two documents will range from 0 to 1, since the term frequencies (tf-idf weights) cannot be negative. The angle between two term frequency vectors cannot be greater than 90°.

Selection of value k

In the traditional kNN algorithm, the value of k is fixed beforehand. If k is too large, big classes will overwhelm small ones. On the other hand, if k is too small, the advantage of kNN algorithm, which could make use of many experts, will not be exhibited. To find best value of k suitable to the data set traditional kNN algorithm uses cross-validation. The CROSSVALIDATION specifies settings for performing V-fold cross-validation to determine the "best" number of neighbors.

- V-fold cross validation divides the data into V folds. Then, for a fixed k, it applies nearest neighbor analysis to make predictions on the vth fold (using the other V−1 folds as the training sample) and evaluates the error. This process is successively applied to all possible choices of v. At the end of V folds, the computed errors are averaged. The above steps are repeated for various values of k. The value achieving the lowest average error is selected as the optimal value for k.
- If multiple values of k are tied on the lowest average error, then the smallest k among those that are tied is selected.[17]

Cross-validation process of traditional kNN algorithm works efficiently when number of instances is small in data set i.e. when size of data set is small but as the size of data set increases it takes larger time to cross-validate for each value of k specified by user in terms of max value of k.

Example A: for better understanding of cross-validations in Traditional kNN:

For some data set DS1:
No. of attributes: 21
No. of instances: 12000
Maximum value of k: 10
Best value of k: 5
The cross-validation process will repeated for:
(Max k- 1) * No. of instances = (10-1)*12000=9 *12000 = 108000

This is large value and takes large time for processing.

To overcome problem of unnecessary iteration in cross-validation for finding best value of k new algorithm is proposed.

The CROSSVALIDATION in proposed kNN algorithm also specifies setting for performing V- fold cross-validation but for determining the "best" number of neighbors the process of cross-validation is not applied to all choice of v but stop when the best value is found. It is observed from the results of effect of value of k in kNN that before and after achieving best value of k accuracy of classification decreases. The cross-validation process starts from maximum value of k specified as input up to value of k = 1.

In proposed algorithm at each v-fold performance of the previous fold is compared if the performance is decreased at v-fold then value of k used in previous fold is selected as best value of k. Newly proposed kNN algorithm will reduce the number of iterations for finding out the best value of k. due to decrease in number of iteration time and space needed to find best k is also decreased.

Applying proposed kNN in Example A we get:
The cross-validation process will repeated for:
(Max k-1-best k) * No. of instances = (10-1-5)*12000 = 4*12000 = 48000

In proposed kNN algorithm the iteration of the loop for finding best value of k is reduced from 108000 to 48000.

4. Experimental results

WEKA 3.7.5 is used for performing experiments. Proposed algorithm is coded using Java. Datasets are taken from http://kavita-ganesan.com/opinosis-opinion-dataset and http://boston.lti.cs.cmu.edu/classes/95-865/HW/HW2/

P=precision, R=recall, F=F measure.

Comparing kNN, Naïve Bayes and SVM for short text classification

Data set 1: Review of notebook
No. of attributes: 7
No. of instances: 19
Cross-validation: 10 folds

Category No	k-NN			Naïve Bayes			SVM		
	P	R	F	P	R	F	P	R	F
1	1	1	1	1	1	1	1	1	1
2	0.5	1	0.66	0.55	1	0.71	0	0	0
3	1	1	1	0.83	1	0.9	0.66	0.8	0.72
4	1	1	1	1	0.75	0.85	1	0.75	0.85
5	0	0	0	0	0	0	0	0	0
Avg	0.68	0.79	0.72	0.66	0.75	0.68	0.51	0.5	0.50

Table 1: Comparison in Data set 1

Data set 2: Review of Swiss hotel
No. of attributes: 6
No. of instances: 18
Cross-validation: 10 folds

Category No	k-NN			Naïve Bayes			SVM		
	P	R	F	P	R	F	P	R	F
1	0.8	0.8	0.8	0.75	0.6	0.66	0.8	0.8	0.8
2	1	1	1	1	1	1	1	1	1
3	0.6	0.75	0.66	0.5	0.75	0.6	0.42	0.75	0.54
4	1	0.8	0.88	1	0.8	0.88	1	0.4	0.57
Av	0.85	0.83	0.84	0.82	0.78	0.79	0.88	0.72	0.72

Table 2 Comparison in Data set 2

Data set 3: Auto
No. of attributes: 21
No. of instances: 12000
Classifier: kNN
Cross-validation: 10 folds

Category No	k-NN			Naïve Bayes			SVM		
	P	R	F	P	R	F	P	R	F
1	0.99	0.99	0.99	0.99	0.99	0.99	1	0.99	0.99
2	0.99	0.99	0.99	0.99	1	0.99	0.99	1	0.99
Av	0.99	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99

Table 3 Comparison in Data set 3

Data set 4: Ford
No. of attributes: 11
No. of instances: 6000
Classifier: kNN
Cross-validation: 10 folds

Category No	k-NN			Naïve Bayes			SVM		
	P	R	F	P	R	F	P	R	F
1	0.87	0.87	0.87	0.68	0.96	0.80	0.73	0.94	0.82
2	0.87	0.87	0.87	0.93	0.56	0.70	0.92	0.65	0.76
Av	0.87	0.87	0.87	0.81	0.76	0.75	0.82	0.80	0.79

Table 4 Comparison in Data set 4

Average result of Data set -1

	Precision	Recall	F- measure
kNN	***0.688***	***0.792***	***0.722***
Naïve Bayes	0.664	0.75	0.689
SVM	0.514	0.5	0.503

Table 5 Average result of Data set 1

Average result of Data set -2

	Precision	Recall	F- measure
kNN	***0.856***	***0.833***	***0.84***
Naïve Bayes	0.819	0.778	0.788
SVM	0.817	0.722	0.724

Table 6 Average result of Data set 2

Average result of Data set -3

	Precision	Recall	F- measure
kNN	***0.998***	***0.998***	***0.998***
Naïve Bayes	0.997	0.997	0.997
SVM	0.995	0.995	0.995

Table 7 Average result of Data set 3

Average result of Data set -4

	Precision	Recall	F- measure
kNN	***0.878***	***0.878***	***0.877***
Naïve Bayes	0.814	0.764	0.754
SVM	0.829	0.802	0.798

Table 8 Average result of Data set 4

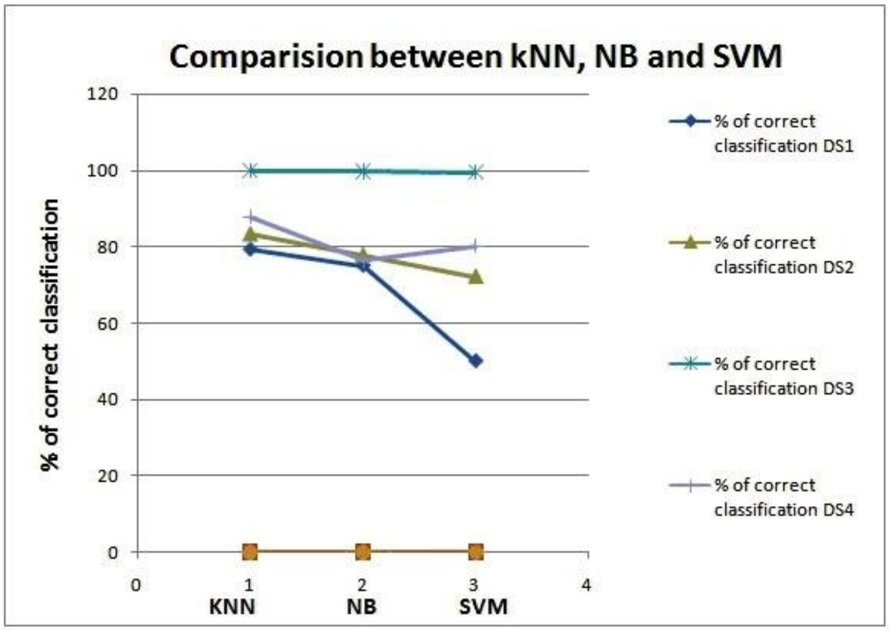

Figure 1 Copmarison between kNN, NB and SVM

Examining effect of k value in kNN

Data set: 1
Detail of data set: review of iPod
No. of attributes: 68
No. of instances: 19
Classifier: kNN
Cross-validation: 10 folds

Selected value of k	Correctly classified instances
1	63.16%
3	**84.21%**
5	78.94%
8	73.68%

Table 9 Effect of k value in Data set 1

Data set: 2
No. of attributes: 21
No. of instances: 12000
Classifier: kNN
Cross-validation: 10 folds

Selected value of k	Correctly classified instances
1	**99.81%**
3	99.79%
5	99.7%
7	99.67%

Table 10 Effect of k value in Data set 3

Data set: 3
No. of attributes: 21
No. of instances: 6000
Classifier: kNN
Cross-validation: 10 fold

Selected value of k	Correctly classified instances
1	87.33%
3	88.78%
5	89.23%
7	**89.41%**
9	89.3%
11	89.09%

Table 11 Effect of k value in Data set 3

Data set: 4
No. of attributes: 21
No. of instances: 1382
Classifier: kNN
Cross-validation: 10 fold

Selected value of k	Correctly classified instances
1	62.44%
3	66.20%
5	68.16%
7	68.23%
9	68.37%
11	69.17%
13	69.60%
15	69.97%
17	**70.47%**
19	69.75%
21	69.68%

Accuracy Increases

Best k

Accuracy Decreases

Table 12 Effect of k value in Data set 4

Form the above tables it can be concluded that if value of k is appropriate to the data then increase in efficiency of the kNN will be noticeable. Here in data set -1 value of k=3 is best value of k which gives 84.21% correctly classified instances. If value of k is less then or greater then best k value, it will affect the performance of kNN.
Also from observations it is clear that accuracy increases up to the best value of k and then after accuracy starts to decrease.

Comparison of traditional kNN and proposed kNN

To find best value of k suitable to the data set traditional kNN algorithm uses cross-validation.
The CROSSVALIDATION in proposed kNN algorithm also specifies setting for performing V- fold cross-validation but for determining the "best" number of neighbors the process of cross-validation is not applied to all choice of v but stop when the best value is found. It can be observed from the results of effect of value of k in kNN that before and after achieving best value of k accuracy of classification decreases. Newly proposed kNN algorithm will reduce the number of iterations for finding out the best value of k. Due to decrease in number of iteration time and space needed to find best k are also decreased.
Data sets used to examine the effect of value of k are used in comparison of traditional and proposed algorithm.

Data set	Maximum value of k	% of correct classification	No. of iteration in traditional kNN	No. of iteration in proposed kNN	Reduced no. of iteration
Data set - 1	10	84.21	9	7	2
Data set - 2	10	99.81	9	9	0
Data set - 3	10	89.41	9	4	5
Data set - 4	20	70.47	19	4	15

Table 13 Traditional kNN v/s Proposed kNN

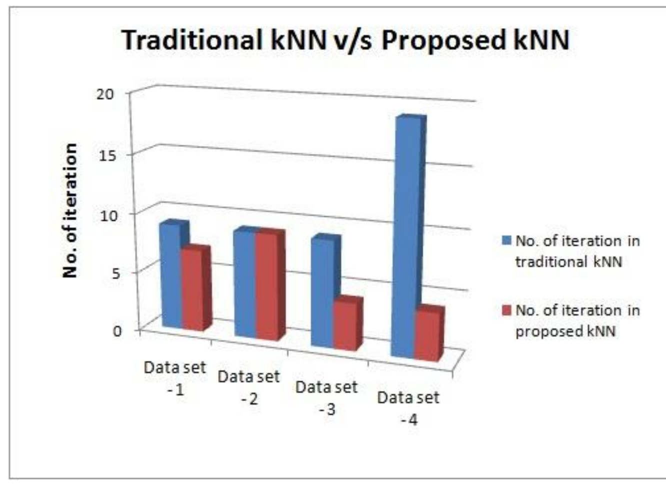

Figure 2 Traditional kNN v/s Proposed kNN

5. Conclusion

For short text classification kNN, Naïve Bayes and SVM algorithms can be used. From the results in section 4 it is concluded that kNN give better accuracy than other two algorithms. Also when kNN algorithm is used with attribute selection its accuracy for classification increases. kNN algorithm depends on the distance function and the value of k nearest neighbor, traditional kNN algorithm finds best value of k using cross-validation. But time and space used by it is larger due to unnecessary processing for each and every value of k from Maximum k to 1. In proposed kNN algorithm the unnecessary processing of cross-validation is reduced due to which time and space used for classification is also reduced. Table 13 shows reduced number of iteration in proposed algorithm. Proposed kNN is more promising than traditional kNN as larger dataset can be used for classification and time for evaluation and building model for large dataset is reduced.

6. References

[1] Arzucan ¨Ozg¨ur, Levent ¨ Ozg¨ur, and Tunga G¨ung¨or "Text Categorization with class-based and corpus-based keyword selection" *In Proceedings of the 20th international conference on Computer and Information Sciences,* **Publisher:** Springer-Verlag October,2005

[2] XindongWu, Vipin Kumar, J. Ross Quinlan , Joydeep Ghosh , Qiang Yang, Hiroshi Motoda , Geoffrey J. McLachlan, Angus Ng , Bing Liu , Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand and Dan Steinberg "Top 10 algorithms in data mining" *Journal Knowledge and Information Systems* Volume 14 Issue 1, December ,2007

[3] Xuan-Hieu Phan, Le-Minh Nguyen, Susumu Horiguchi "Learning to classify short and sparse text & web hidden topics from large-scale data collection" *In Proceedings of the 17th international conference on World Wide Web* ACM New York, NY, USA,2008

[4] Alfonso Marin "Comparison of Automatic Classifiers' Performances using Word-based Feature Extraction Techniques in an E-government setting" *University essay from KTH/Skolan for informations- och kommunikationsteknik (ICT)* January,2011

[5] Victoria Bobicev, Marina Sokolova "An Effective and Robust Method for Short Text Classification" *In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence* volume 3, July 2008

[6] Hitesh Sajnani, Sara Javanmardi, DavidW. McDonald, Cristina V. Lopes " Multi-Label Classification of short text: A Study on wikipedia Barnstars" *Paper from AAAI-11 workshop on Analyzing microtext* 2011

[7] Bengel, J, Gauch, S. Mittur, E. and Vijayaraghavan R."Chat room topic detection using classification." *In proceeding of 2nd Symposium on Intelligence and Security Informatics*,2004

[8] Yiming Yang. "A study of thresholding strategies for text categorization." *In proceeding of 24th International ACM SIGIR Conference* 2001

[9] Yang, Y. and Liu, X. "A re-examination of text categorization methods" *In proceedings of the 22nd annual international ACM SIGIR conference on Research and Information Retrieval* 1999

[10] Wei Wang, Sujian Li, Chen Wang "An Improved KNN Algorithm for Text Categorization" *In Proceedings of NTCIR-7 Workshop Meeting* 2008

[11] Ali-Mustafa Qamar, Eric Gaussier, Jean-Pierre Chevallet, Joo Hwee Lim, "Similarity Learning for Nearest Neighbor Classification" *ICDM '08 Proceedings of the 2008 Eighth IEEE International Conference on Data Mining* 2008

[12] Wen-tau Yih and Christopher Meek, "Improving Similarity Measures for Short Segments of Text" *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*[2007]

[13] Jasmine Kalathipparambil Sudhakaran, Ramaswamy Vasantha "A Mixed Method Approach for Efficient Component Retrieval from a Component Repository" *Journal of Software Engineering and Applications* ,2011

[14] Gautam Bhattacharya a, Koushik Ghosh b, Ananda S. Chowdhury "An affinity-based new local distance function and similarity measure for kNN algorithm" *Pattern Recognition Letters , Volume 33 Issue 3 Publisher: Elsevier Science Inc* ,2012

[15] Muhammed Miah **"**Improved k-NN Algorithm for Text Classification**"** *Department of Computer Science and Engineering University of Texas at Arlington, TX, USA*,2003

[16] *Z. Xie, W. Hsu, Z. Liu, & M. Lee, SNNB: "*A selective neighborhood based Naive Bayes for lazy learning", *Proc. 6th Pacific-Asia Conf. on KDD, Taipei, Taiwan,* 2002

[17] "cross validation subcommand in kNN", http://publib.boulder.ibm.com/infocenter/spssstat/v20r0m0/index.jsp