SV-kNNC: An Algorithm for Improving the Efficiency of k-Nearest Neighbor

Anantaporn Srisawat, Tanasanee Phienthrakul, and Boonserm Kijsirikul

Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University,
Bangkok 10330 Thailand
anantaporn.s@student.chula.ac.th, tanasanee@yahoo.com,
boonserm.k@chula.ac.th

Abstract. This paper proposes SV-kNNC, a new algorithm for k-Nearest Neighbor (kNN). This algorithm consists of three steps. First, Support Vector Machines (SVMs) are applied to select some important training data. Then, k-mean clustering is used to assign the weight to each training instance. Finally, unseen examples are classified by kNN. Fourteen datasets from the UCI repository were used to evaluate the performance of this algorithm. SV-kNNC is compared with conventional kNN and kNN with two instance reduction techniques: CNN and ENN. The results show that our algorithm provides the best performance, both predictive accuracy and classification time.

1 Introduction

In supervised learning, kNN is one of the most popular choices due to its simplicity [1]. The advantages of kNN include the ability to model complex target functions by a collection of less complex local approximation. Moreover, information presented in the training examples is never lost [2].

However, the main practical difficulties of kNN are the amounts and characteristics of training data. kNN methods are based on a distance function for all pairs of a new query instance and training data. If there are a lot of training data, kNN will take a lot of time in the classification process. In addition, the training data have strong influence on the target output. Noisy data may also reduce the performance of kNN. Hence, if unimportant data and noisy data are eliminated, both classification time and error should be reduced.

This paper proposes SV-kNNC that applies SVM [3] and k-mean clustering [4] to improve the efficiency of kNN. In this approach, SVM are used to reduce the training data of kNN for improving the classification time. Moreover, the accuracy of kNN can also be improved by using k-means clustering to assign weight for each training data.

2 SV-kNNC

In this section, we propose SV-kNNC that consists of three processes: instance selection process, weight assigning process, and classification process. The model of SV-kNNC is illustrated in Fig. 1.

Q. Yang and G. Webb (Eds.): PRICAI 2006, LNAI 4099, pp. 975 – 979, 2006.
© Springer-Verlag Berlin Heidelberg 2006

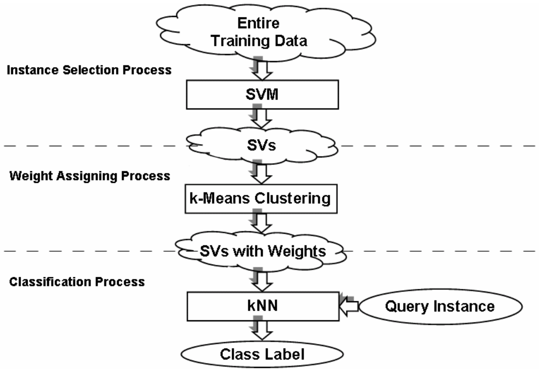


Fig. 1. SV-kNNC Model

2.1 Instance Selection

A main difficulty of kNN is the classification time that increases with the number of training data. If unimportant training data are discarded, the classification time can be reduced. Conventional data reduction methods such as CNN [5], IB3, DROP [6, 7], and ENN [8] gradually add or remove training data. Since these methods do not consider all training data at the same time, they may not yield the actual optimal training set.

We notice that the support vector machine uses only support vectors (SVs), examples that are closest to the hyperplane, to classify unseen data. Therefore, the obtained SVs are representative training data that should be sufficient to represent all training data.

For instance selection process, entire training data are fed into SVM and a set of SVs is produced as the output of SVM training. These SVs are then stored in memory. This process also can eliminate redundant instances of each class in the feature space.

2.2 Weight Assigning

While SVs are good representative data for each class in the feature space, some SVs may not be suitable as training data for kNN. This is because the distance between instances is changed when these SVs are retransformed to the input space. The class label of an instance may differ from its surrounding nearest neighbor instances. These instances may be the cause of misclassification.

To avoid this problem, re-checking the contribution of SVs to the classification in the input space before using them to classify a query instance is considered. Weight is then assigned to determine the contribution of each SV. The instance which has a higher weight is more credible, and thus it will have more impact on kNN classification.

There are 2 steps in weight assigning process. First, the k-means algorithm is used to partition all SVs into pre-defined *k* clusters. If there are more than one class label appearing in one cluster, the instances with the majority class will be more credible than the instances with the minority classes. After data are clustered, the class labels of all SVs are considered. In this step, each instance in a cluster is assigned weight according to the proportion of the instance class labels in that cluster. This weight of sample x_i, denoted as w_i, is shown in equation (1).

$$w_i = \frac{n(class(x_i))}{Total} \tag{1}$$

where $i = 1,\ldots,m$, for m is the number of all SVs. $n(class(x_i))$ is the number of samples in the cluster that have the same class with the sample x_i. $Total$ is the number of total samples in the cluster.

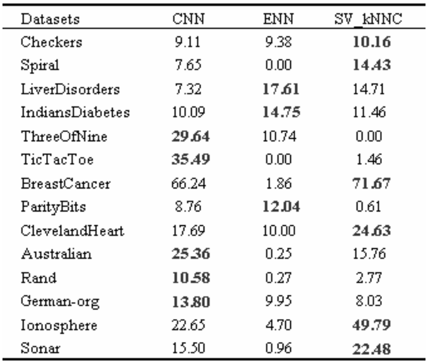
2.3 Classification

In classification process, unseen instances are classified by the kNN algorithm. Only weighted SVs, which are calculated in the previous process, are used as training data of kNN. When a new query instance is entered, kNN finds a set of *k* nearest instances from a set of SVs. Weights of SVs with the same class label are summed up, and the class label with the maximum weight is produced as the class label of the query instance.

3 Experiments and Results

In order to verify the performance of our approach, fourteen datasets from the UCI repository [9] are tested using 5-folds cross validation. Each dataset contains two classes. When SVM, CNN, and ENN were run, the training data were selected differently by these algorithms. Percentages of reduced data are showed in Table 1. We found that SV-kNNC has the highest ability on data reduction. It reduced data more than CNN and ENN on 6 datasets. Moreover, SV-kNNC reduced training examples up to 71.67% on the BreastCancer dataset.

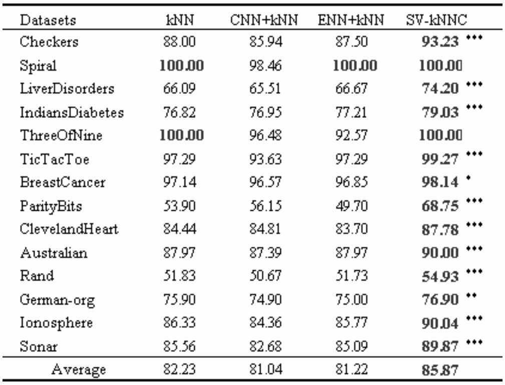
Table 1. Percentage of reduced data



Datasets	CNN	ENN	SV_kNNC
Checkers	9.11	9.38	**10.16**
Spiral	7.65	0.00	**14.43**
LiverDisorders	7.32	**17.61**	14.71
IndiansDiabetes	10.09	**14.75**	11.46
ThreeOfNine	**29.64**	10.74	0.00
TicTacToe	**35.49**	0.00	1.46
BreastCancer	66.24	1.86	**71.67**
ParityBits	8.76	**12.04**	0.61
ClevelandHeart	17.69	10.00	**24.63**
Australian	**25.36**	0.25	15.76
Rand	**10.58**	0.27	2.77
German-org	**13.80**	9.95	8.03
Ionosphere	22.65	4.70	**49.79**
Sonar	15.50	0.96	**22.48**

In Table 2, the accuracies of SV-kNNC are compared with the conventional kNN and kNN with two instance reduction techniques: CNN and ENN. These results show that the accuracies of SV-kNNC are statistically significantly higher than the conventional kNN with significance level better than 0.05 on all datasets except for datasets *Spiral* and *ThreeOfNine*. The SV-kNNC also provides the best accuracies on all datasets when compared with the other data reduction techniques, i.e. CNN and ENN.

Table 2. Comparison of accuracies on various algorithms



Datasets	kNN	CNN+kNN	ENN+kNN	SV-kNNC
Checkers	88.00	85.94	87.50	**93.23** ***
Spiral	**100.00**	98.46	**100.00**	**100.00**
LiverDisorders	66.09	65.51	66.67	**74.20** ***
IndiansDiabetes	76.82	76.95	77.21	**79.03** ***
ThreeOfNine	**100.00**	96.48	92.57	**100.00**
TicTacToe	97.29	93.63	97.29	**99.27** ***
BreastCancer	97.14	96.57	96.85	**98.14** *
ParityBits	53.90	56.15	49.70	**68.75** ***
ClevelandHeart	84.44	84.81	83.70	**87.78** ***
Australian	87.97	87.39	87.97	**90.00** ***
Rand	51.83	50.67	51.73	**54.93** ***
German-org	75.90	74.90	75.00	**76.90** **
Ionosphere	86.33	84.36	85.77	**90.04** ***
Sonar	85.56	82.68	85.09	**89.87** ***
Average	82.23	81.04	81.22	**85.87**

Statistical significance for the difference from kNN at level: * 0.05, ** 0.025, *** 0.01

From the experimental results, we found that SV-kNNC enhances the performance of kNN, both classification time and predictive accuracy. Furthermore, the data reduction ability of SV-kNNC is better than CNN and ENN. This is because CNN and ENN keep or remove an instance depending on its nearest instances. Besides, both CNN and ENN are especially sensitive to noise, and noisy instances may be retained. Hence, there are some instances that are still misclassified by CNN and ENN. On the other hand, SVM selects a set of good representative examples of each class by inspecting whole instances at the same time. Thus, some noisy and redundant data may be removed.

4 Conclusions

This paper proposes an approach for data reduction to enhance performance of kNN, called SV-kNNC. This algorithm is divided into three steps. First, SVs from SVM learning process are used to be the training examples. Then, these SVs are assigned weights by the ratio of each class label on a cluster from k-mean clustering. Finally, SVs with weights are used to classify the query instances by kNN classification process.

The experimental results showed that SV-kNNC has the ability to reduce data (more than 70% in some dataset). Thus, the classification time of SV-kNNC is less than kNN. In addition, the accuracy of SV-kNNC is better than the conventional kNN, and kNN with two instance reduction techniques (CNN and ENN) on the UCI benchmarks.

SV-kNNC achieves the best performance because training data are analyzed twice before classification process. First, SVM eliminates redundant data and selects the instances that are nearest to a decision surface in the feature space. Then, these SVs are re-checked and are assigned weights. Therefore, the obtained training data are more effective than the conventional techniques.

Acknowledgement

The authors acknowledge the financial support provided by the Thailand Research Fund and the Royal Golden Jubilee Ph.D. Program.

References

1. Cover, T., Hart, P.: Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory. 13 (1967) 21-27
2. Mitchell, T.M.: Machine Learning. The McGraw-Hill, New York (1997)
3. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, New York (1998)
4. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1 (1967) 281-297
5. Hart, P.E.: The Condensed Nearest Neighbor Rule. IEEE Transactions on Information Theory. 14 (1968) 515-516
6. Wilson, D.R., Martinez, T.R.: Reduction Techniques for Instance-Based Learning Algorithms. Machine Learning, 38 (2000) 257-286
7. Jankowski, N., Grochowski, M.: Comparison of Instance Selection Algorithms I. Algorithms Survey. ICAISC (2004) 598-603
8. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules using Edited Data. IEEE Transactions on Systems, Man, and Cybernetics. 2 (1972) 408-421
9. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998)