A New Parameter-Free Classification Algorithm Based on Nearest Neighbor Rule and *K*-means for Mobile Devices

TUNG-SHOU CHEN, CHIH-CHIANG LIN, YUNG-HSING CHIU*
Graduate School of Computer Science and Information Technology
National Taichung Institute of Technology
No. 129, Sec. 3, Sanmin Rd., Taichung 404
TAIWAN
http://csit.ntit.edu.tw

Abstract: - This paper proposes a parameter-free classifier which combines *K*-means with Nearest Neighbor Rule (NNR) - called Incremental Cluster-based Classification (ICC). The classifier is used in low power and capacity devices such as Personal Digital Assistant (PDA) and Smartphone. In the training phase, ICC employs *K*-means to group instances into several clusters, and then incrementally separates the cluster into two clusters until the cluster members belong to the same type within each cluster. Thus instances have uniform class label within each cluster. In the predicting phase, ICC adopts NNR to find a centroid which is the nearest neighbor of the unlabeled instance. Since the training data are substituted by the cluster centroids; memory and computation requirements are decreased. *K*-means and NNR are both simple and efficient methods. ICC is easy to redo and have efficient performance and is, hence, suitable for low capacity hardware. In this paper, the prediction accuracy of ICC is evaluated and compared with those of NNR and Support Vector Machine (SVM). Our experimental results show that the prediction accuracy of ICC is comparable to NNR. Although NNR is the easiest to use and redo, it is sensitive to noises and consumes time and memory for a large dataset. Despite the higher accuracy of LIBSVM, it is time-consuming to select an appropriate kernel function and related parameters. ICC is parameter-free, simple to operate and easy to implement. Mobile users can complete their work more conveniently and accurately.

Key-Words: - Classification, Parameter-Free, *K*-means, Nearest Neighbor Rule (NNR), Support Vector Machine (SVM), Mobile Devices

1 Introduction

Since wireless environment is becoming more and more convenient, mobile commerce or commercial transactions via mobile devices are rapidly developed in recent years. Analyzing and processing commercial and personal data using mobile devices is an important requirement for many application domains such as security and finance. These applications require the ability to analyze data quickly. Classification is a supervised learning technique. It offers the computer the ability to recognize objects. This technique is being extensively used in many fields, including Bioinformatics [1], Intrusion Detection System [2], Decision Support System [3] and more [4]. A classifier is always expected to be accurate, fast, simple to operate, easy to implement, robust and scalable [5].

Many useful tools have been employed for classification. Among them, Nearest Neighbor Rule (NNR) [6], which uses the nearest training instance to predict the class of the new instance, is parameter-free and easy to implement. As well, NNR is simple to operate and has high accuracy. However, it is sensitive to noises [7]. *K*-Nearest Neighbor Rule (*K*-NNR) [6] mitigates the adverse effects of noises by voting among *K* closest neighbors. It is difficult to decide a suitable value *K* for each situation [8]. Users may have to try many times and spend much time to find an appropriate value in different environments. Moreover, NNR and *K*-NNR are time and memory consuming in a large dataset [7]. They are unsuitable in mobile device which is low in power and capacity.

Support Vector Machine (SVM) is one of the powerful machine learning techniques, which was introduced in 1995 by Vapnik [9] and derived from the statistical learning theory. This technique maps the non-separable data into a higher-dimensional feature space and establishes a hyperplane, which maximizes the margin from the hyperplane to the closest training instance. Thus SVM can predict the unlabeled instance by the hyperplane accurately. Unfortunately, selecting a correct kernel function and appropriate correlated parameters for SVM is a complex problem [10]. Optimizing the margin of separation require powerful hardware, therefore SVM is unsuitable to be applied in mobile devices.

Moreover, SVM is not easy to be understood for those without solid background of mathematics or machine learning, leading to poor interpretation of the results and failure to make the best use of SVM.

In order to develop a useful classification algorithm for mobile devices such as the Personal Digital Assistant (PDA) and the Smartphone, we develop a new parameter-free classifier which combines *K*-means [11] with NNR, called Incremental Cluster-based Classification (ICC). In the training phase, ICC employs *K*-means to group the instances into *K* clusters, and then separate the cluster incrementally until the cluster members belonged to the same type within each cluster. Thus the instances have uniform class within each cluster and then the training data are substituted by the cluster centroids. In the predicting phase, ICC adopts the NNR to find the cluster centroids which are the nearest neighbor of each testing instance. Since the number of centroids is small, the performance of this method is effective. Moreover, *K*-means and NNR both are simple and efficient methods, hence ICC is easy to implement and suitable to be applied in mobile device.

The rest of this paper is organized as follows. Section 2 gives a brief overview of the proposed method. Section 3 discusses the comparison of experimental results between ICC, NNR and SVM. Finally, conclusions are drawn in Section 4.

2 Incremental Cluster-based Classification (ICC) Algorithm

The general classifier usually consists of two principal phases: training and predicting. In training phase, ICC adopts the concept of the literature [12] to partition the dataset into the clusters incrementally and then the generated cluster centroids are used to construct the training model. ICC determines the class of the unlabeled instance by the training model in the predicting phase. The two phases are described, separately, as follows.

2.1 ICC Training Phase

Let a set of training instances $X=\{x_1, x_2, x_3, ..., x_m\}$, and a training set $R=\{<x_1, t_1>, <x_2, t_2>, <x_3, t_3>, \ldots, <x_m, t_m>\}$, t_m is the class label of each instance; $d(x_i, x_j)$ represent the similarity between x_i and x_j, where $i, j=1,2,3, \ldots, m$.

The objective of this phase is to construct a training model which is developed as follows:

Step 0: Initially *K* is set to the class number in the whole training set.

Step 1: Employ *K*-means to group the instances into *K* clusters. Consequently, *K* clusters are represented by *K* centroids in this step.

Step 2: If the class label of whole cluster members is the same in a cluster, this cluster will not be separated. Otherwise, separate the cluster into two clusters by *K*-means.

Step 3: Repeat Step 1 and Step 2, until cluster members belonged to the same type within each cluster, respectively. Thus the cluster member type is uniform within each cluster and each cluster is associated to a class label that is its members.

Step 4: Set the label t_i to each centroid; let t_i to be the same as that of each cluster.

Step 5: Extract all cluster centroids.

This phase is a process of the incremental clustering number with the accuracy condition; hence the cluster members are held by the same class within each cluster. The cluster centroids represent whole training instance set *X*, and the training model is constructed. The number of centroids is small. Therefore memory and computation requirements are decreased and the predicting performance of ICC will be effective.

2.1 ICC Predicting Phase

In this phase, ICC employs NNR to predict a class label of new instances. Consider a new instance *y* and its class label *t* is unlabeled. Firstly; ICC calculates the similarity between *y* and each centroids c_i, and then finds the most similar one. Finally, ICC determines the class *t* of the new instance *y* as the class t_i of the most similar centroid.

3 Experimental Results

In this paper, the performance of ICC is evaluated and compared with NNR and LIBSVM [13] by using the published benchmark datasets. The experimental results are based on the calculated accuracy rate (see Table 1). In these experiments, the accuracy rate is defined as follows:

$$\textit{accuracy rate (\%)} = \textit{(the number of correct separation in X)} / \|R\| .$$

As shown in Table 1, the comparison is based on 5-fold and 10-fold cross validations and tested on five benchmark datasets. All of these datasets are found in UCI machine learning database [14]. The first is the Diabetes dataset which has 8 attributes and 768 instances. The second is the Ionosphere dataset consisting of 34 attributes and 351 instances. The

third is the WDBC dataset which has 30 attributes and the size of instances is 569. The fourth is the Breast Cancer dataset which contains 10 attributes and the size of instances is 683. The fifth is the Wine dataset with 13 attributes and 175 instances.

Table 1 The comparison of accuracy rate between ICC, NNR and LIBSVM

Datasets / v-fold		ICC	NNR	LIBSVM -Bad	LIBSVM -Best
Diabetes (%)	5	67.1	68.6	67.2	77.3
	10	66.4	70.0	68.2	77.2
Ionosphere (%)	5	84.3	83.4	65.8	92.6
	10	84.9	84	65.8	92.6
WDBC (%)	5	88.9	91.2	62.7	96.1
	10	89.3	91.1	62.7	96.5
Breast Cancer (%)	5	56.8	62.6	65.0	66.0
	10	60.0	61.5	65.0	66.6
Wine (%)	5	90.3	94.3	41.1	97.7
	10	91.2	94.8	41.1	98.3

As shown in Table 1, the prediction accuracy of ICC is comparable to that of NNR. Although NNR is the easiest to use and redo, it is time and memory consuming for a large dataset. ICC substitutes the whole training dataset by the cluster centroids. Hence the memory and computation requirements are decreased. For example, Wine and Ionosphere datasets are replaced with 14 and 97 cluster centroids, respectively.

LIBSVM-Best indicates the best experimental results of LIBSVM, which we spent a lot of time to find. In each dataset, the best result of LIBSVM was used to find the different kernel type including linear, polynomial and RBF kernel functions. We obtained the best results of LIBSVM by using the RBF kernel function, whose associated parameter gamma was between 0.0001 and 10, and the parameter C was from 0.1 to 10000.

The experimental results of LIBSVM with arbitrary kernel function and parameters are indicated by LIBSVM-Bad. The gray cells in Table 1 indicate that LIBSVM is sensitive to parameter setting. Thus Users have to select the appropriate kernel function and parameters by trials and errors when they want to have a better result. Although LIBSVM outperformed ICC with regards to overall prediction accuracy, it is time-consuming to select an appropriate kernel and related parameters. In contrast with SVM, ICC is parameter-free, simple to operate and easy to redo.

4 Conclusions

In this paper, we proposed a parameter-free classifier; the Incremental Cluster-based Classification (ICC), which combines *K*-means with NNR. Experimental results showed that ICC could classify the data accurately, and its prediction accuracy is comparable to NNR. Since memory and computation requirements are decreased by replacing the training dataset with clustering centroids, the ICC predicting performance is effective. Users use ICC without having to find the appropriate parameter by trials and errors, so ICC is easier to use than SVM which need to explicitly specify the kernel function and parameters. ICC consists of *K*-means and NNR which are both simple and efficient methods. Therefore, ICC could be migrated to mobile device and use to assist mobile users to process their data more conveniently and quickly.

References:

[1] Chen, T.S., Tu, B.J., and Juan, L.C., Using Data Mining to Detect and Classify the Large Cell Lung Cancer and Adenocarcinoma on Microarray, *Proceedings of International Conference on Informatics, Cybernetics, and Systems (ICICS)*, 2003, pp. 1519-1524.

[2] Chen, R.C., Chen, J., Chen, T.S., Hsieh, C.H., Chen, T.Y., and Wu, K.Y., Building an Intrusion Detection System Based on Support Vector Machine and Genetic Algorithm, *Lecture Notes in Computer Science (LNCS)*, 2005, pp. 409-414.

[3] Chen, R.C., Chen, T.S., Chien, Y.E., and Yang, Y.R., Novel Questionnaire-Responded Transaction Approach with SVM for Credit Card Fraud Detection, *Lecture Notes in Computer Science (LNCS)*, 2005, pp. 916-921.

[4] Han, J., Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.

[5] Dunham, M.H., *Data Ming: Introductory and Advanced Topics*, Prentice Hall, 2003.

[6] Cover, T. M., and Hart, P. E., Nearest Neighbor Pattern Classification, *IEEE Transactions on Information Theory*, 1967, pp. 21-27.

[7] Roiger, R.J., Geatz, M.W., *Data Mining: A Tutorial-Based Primer*, Addison Wesley, 2003.

[8] P. E. Hart, The condensed nearest neighbor rule, *IEEE Transactions on Information Theory*, Vol.14, No.3, 1968, pp.515-516.

[9] Vapnik, V.N., *The Nature of Statistical Learning Theory*, Springer, 1995.

[10] Zhao, X. M., Huang, D. S., Cheung, Y. M., Wang, H. Q., and Huang, X, A Novel Hybrid GA/SVM System for Protein Sequences Classification, *Lecture Notes in Computer Science (LNCS)*, 2004, pp. 11–16.

[11] MacQueen, J., Some Methods for Classification and Analysis of Multivariate Observations, *The Proceedings of Conference of Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, 1967, pp. 281–297.

[12] Chen, T.S., Tsai, T.H., Chen, Y.T., Lin, C.C., Chen, R.C., A combined *K*-means and hierarchical clustering method for improving the clustering efficiency of microarray, *Proceedings of International Conference on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2005, pp. 405–408.

[13] Chang, C.C. and Lin, C.J., *LIBSVM: A Library for Support Vector Machines*, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[14] Newman, D.J., Hettich, S., Blake, C.L., and Merz, C.J., *UCI Repository of Machine Learning Databases*, http://www.ics.uci.edu/~mlearn/ML Repository.html, 1998.