Enhanced KNNC Using Train Sample Clustering

Hamid Parvin[1(✉)], Ahad Zolfaghari[2], and Farhad Rad[2]

[1] Department of Computer Engineering, Mamasani Branch, Islamic Azad University, Mamasani, Iran
parvin@iust.ac.ir

[2] Department of Computer Engineering, Yasouj Branch, Islamic Azad University, Yasouj, Iran
rad@comp.iust.ac.ir

Abstract. In this paper, a new classification method based on k-Nearest Neighbor (kNN) lazy classifier is proposed. This method leverages the clustering concept to reduce the size of the training set in kNN classifier and also in order to enhance its performance in terms of time complexity. The new approach is called Modified Nearest Neighbor Classifier Based on Clustering (MNNCBC). Inspiring the traditional lazy k-NN algorithm, the main idea is to classify a test instance based on the tags of its k nearest neighbors. In MNNCBC, the training set is first grouped into a small number of partitions. By obtaining a number of partitions employing several runnings of a simple clustering algorithm, MNNCBC algorithm extracts a large number of clusters out of those partitions. Then, a label is assigned to the center of each cluster produced in the previous step. The assignment is determined with use of the majority vote mechanism between the class labels of the patterns in each cluster. MNNCBC algorithm iteratively inserts a cluster into a pool of the selected clusters that are considered as the training set of the final 1-NN classifier as long as the accuracy of 1-NN classifier over a set of patterns included the training set and the validation set improves. The selected set of the most accurate clusters are considered as the training set of proposed 1-NN classifier. After that, the class label of a new test sample is determined according to the class label of the nearest cluster center. While kNN lazy classifier is computationally expensive, MNNCBC classifier reduces its computational complexity by a multiplier of 1/k. So MNNCBC classifier is about k times faster than kNN classifier. MNNCBC is evaluated on some real datasets from UCI repository. Empirical results show that MNNCBC has an excellent improvement in terms of both accuracy and time complexity in comparison with kNN classifier.

Keywords: Edited nearest neighbor classifier · kNN · Combinatorial classification

1 Introduction

One of the most important goals of artificial intelligence is to design models with high recognition rates [30-33]. In pattern recognition, the input space is mapped into the high dimensional feature space, and in the feature space it is tried to determine the optimal hyperplane(s), so that the mapped function better approximates the main function for each unseen data. The mapping that is named classification is an interesting subject in machine learning and data mining communities with a lot of studies around it [16-18].

Despite the simplicity, k-Nearest Neighbor (kNN) classifier is one of the most fundamental classifiers. It is also the simplest classifier. When there is a little or no prior knowledge about the data distribution, kNN classifier could be automatically the first choice for a classification study.

In a lot of recent emerging applications, such as text categorization [21-22], multiple classifier systems (MCS) [23-24], intrusion detection field [25] (an intrusion detection problem is first converted to a text categorization problem; then it is treated as a text categorization problem), medical systems such as diagnosis of diabetes diseases [26], thyroid diseases [27] and myocardial infarction [28], and image classification [29] and etc., kNN classifier has been successfully applied and proves its effectiveness.

The kNN classifies a test sample x by assigning it the label most frequently represented among the k nearest samples; in other words, a decision is made by examining the labels on the k-nearest neighbors and taking a majority vote mechanism. kNN classifier was developed from the need to perform discriminant analysis when reliable parametric estimates of the probability densities are unknown or difficult to determine. In 1951, Fix and Hodges introduced a non-parametric method for pattern classification that is known the k-nearest neighbor rule [1] and [15]. Later in 1967, some of the formal properties of the k-nearest neighbor rule have been worked out; for instance it was shown that for k=1 and $n \to \infty$ the kNN classification error is bounded above by twice the Bayes error rate [2]. Once such formal properties of kNN classification were established, a long line of investigation ensued including new rejection approaches [3], refinements with respect to Bayes error rate [4], distance weighted approaches [5-6], soft computing [7] methods and fuzzy methods [8-9].

Some advantages of kNN include: the simplicity to use, the robustness to learn in a noisy training data (especially if it uses the inverse square of weighted distances as the "distance metric"), and finally the effectiveness in learning at a large scale training dataset. Although kNN has the mentioned advantages, it has some drawbacks such as: high computational cost (because it needs to compute distance of each query instance to all training samples); large memory consumption (in proportion with the size of training set); ineffectiveness in multidimensional datasets; sensitivity to well-setting of parameter k (number of effective nearest neighbors); sensitivity to the used distance metric; and finally ignoring a weighting mechanism for features [10].

The High computational cost of the nearest neighbor algorithm, in both space (storage of prototypes) and time (distance computation) has received a great deal of analysis. Suppose we have N labeled training samples in d dimensions, and seek to find the closest to a test point x (k = 1). In the most naive approaches we inspect each stored point in turn, calculate its Euclidean distance to x, retaining the identity only of the current closest one. Each distance calculation is O(d), and thus this search is O(dN^2) [10].