Optik 125 (2014) 2320–2326

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Kernel representation-based nearest neighbor classifier

Xiaozhao Fang[a], Yuwu Lu[a], Zhengming Li[a,b], Lei Yu[a,b], Yan Chen[a,*]

[a] Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
[b] Shenzhen Key Laboratory of Urban Planning and Decision-Making Simulation, Shenzhen, China

ARTICLE INFO

Article history:
Received 20 May 2013
Accepted 14 October 2013

Keywords:
Nearest neighbor classifier
Classification mechanism
Feature space
Kernel function
Nonlinear mapping

ABSTRACT

An improvement to the nearest neighbor classifier (INNC) has shown its excellent classification performance on some classification tasks. However, it is not very clearly known why INNC is able to obtain good performance and what the underlying classification mechanism is. Moreover, INNC cannot classify low-dimensional data well and some high-dimensional data in which sample vectors belonging to different class distribution but have the same vector direction. In order to solve these problems, this paper proposes a novel classification method, named kernel representation-based nearest neighbor classifier (KRNNC), which can not only remedy the drawback of INNC on low-dimensional data, but also obtain competitive classification results on high-dimensional data. We reveal the underlying classification mechanism of KRNNC in details, which can also be regarded as a theoretical supplement of INNC. We first implicitly map all samples into a kernel feature space by using a nonlinear mapping associated with a kernel function. Then, we represent a test sample as a linear combination of all training samples and use the representation ability to perform classification. From the way of classifying test samples, KRNNC can be regarded as the nonlinear extension of INNC. Extensive experimental studies on benchmark datasets and face image databases show the effectiveness of KRNNC.

1. Introduction

Nearest neighbor classifier (NNC) is a well-known but one of the oldest classifiers [1–8]. The performance of NNC depends crucially on the way of computing distance. The similarity between two patterns is usually measured by a distance function. In other words, when a query pattern comes, we firstly identify the nearest neighbor of the query pattern from all training patterns using a distance function. Then we classify the query pattern into the same class as the training pattern that has the minimum distance.

In recent years, there has been growing interest in finding a proper metric to determine the nearest neighbor of a sample. Goldberger et al. [9] proposed to learn Mahalanobis distance measure via a stochastic variant of the leave-one-out 1-NN score on training samples. Wang et al. [8] produced an extremely simple adaptive distance measurement that allocates different weights for each sample. In addition, Samet proposed the MaxNearestDist algorithm for finding K nearest neighbors [10]. Many other methods [11–14] were proposed for searching the nearest neighbor computationally efficiently. However, in these methods, the relationship among all training samples is ignored. In other words, they independently calculate distances between a test sample and each training sample and thus ignore the relationship among all training samples. This may be a reason that leads to an incorrect classification result. As shown in Fig. 1, when we use Euclidean distance, intra-class distance between (a) and (b) is small while inter-class distance between (a) (or (b)) and (e) is large. However, if an image is occluded, Euclidean distance might have "unexpected" intra-class and inter-class distances. For example, intra-class distance between (a) and (d) is larger than inter-class distance between (e) and (d). By using NNC, one can properly identify the similarity between ((a) or (b)) and (c) and consider them as images from the same class. However, (d) and (e) may be incorrectly regarded as images from the same class while they come from different classes.

Some other methods attempted to use other metrics to measure neighbor relation, such as kernel methods [15,16] and sparse representation (SR) [17] methods. Kernel methods use a nonlinear mapping to map all samples into a kernel feature space in which the measurement of neighbor relation is implemented. SR, a distinctive image classification method, has received great attention [18–21] and obtained promising performance on image recognition such as face recognition [22–26]. SR requires that a test sample is sparsely represented by a weighted sum of all training samples. Classification is performed by evaluating the representation ability on the test sample of each class and then the test sample is assigned to the class that has maximum representation ability.

* Corresponding author at: Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen, China.
E-mail addresses: xzhfang168@126.com (X. Fang), lizhengming2004@126.com (Y. Lu), luyuwu2008@163.com (Z. Li), leilayu@hitsz.edu.cn (L. Yu), jadechenyan@gmail.com (Y. Chen).

0030-4026/$ – see front matter © 2013 Elsevier GmbH. All rights reserved.
http://dx.doi.org/10.1016/j.ijleo.2013.10.074

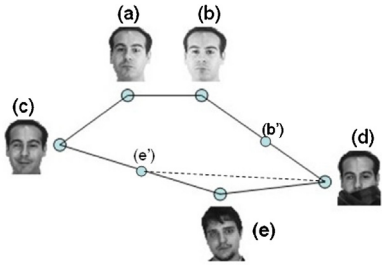

Fig. 1. The position relations among different samples.

'Sparse' implies that representation coefficients of some training samples are equal to zero and the sparsity of the representation coefficients can be measured by l_1-minimization [17]. In [27,28], two kernel sparse representation-based classifiers were presented to extend sparse representation-classifiers (SRC) to their nonlinear versions (KSR/KSRC). Experimental results showed that SRC and its nonlinear versions have better performance than NNC and nearest subspace (NS) [29] on some benchmark datasets. However, KSR and KSRC are very time-consuming because they need to solve l_1-minimization problems. Although some algorithms have been presented to speed up l_1-minimization problems [30], different acceleration algorithms have different performance on some given classification tasks [31]. Therefore, it is difficult for users to choose a proper method. In contrast, l_2-norm-based representation method, proposed by Zhang et al. [31], is considered as a powerful and attractive representation tool due to its excellent results in terms of accuracy and computational complexity. Moreover, l_2-norm-based representation algorithm can be regarded as a special distance metric method that estimates distances between a test sample and each training sample. l_2-norm-based nearest neighbor classifier (INNC) [32], as an improved version of NNC, has shown its excellent classification performance on high-dimensional data. From the way of classification, INNC takes into account the relationship among all training samples and calculates distances between a test sample and each training sample dependently. Therefore, INNC can obtain good performance on high-dimensional data.

However, representation-based methods, i.e., INNC and linear regression classification (LRC) [33], have quite limited classification ability when they classify low-dimensional data and some high-dimensional data in which sample vectors belonging to different class distribution but have the same vector direction. This is due to the following reasons.

When sample has very low dimensionality and the number of training samples of each class is greater than their dimensionality, a test sample can be accurately represented by an arbitrary class. That is to say, the deviation of each class from a test sample trends to zero. Taking LRC for example:

Suppose there are c classes and each class has n training samples (all samples are column vector). We use the *ith* class samples to represent a test sample, namely $y = X_i\beta_i (i = 1, 2, ...c)$, where $X_i = \left[x_1^i, \ x_2^i, ...x_n^i\right]$ $(x_j^i \in \Re^m, j = 1, 2, ...n)$ is the ith class training samples and y $(y \in \Re^m)$ is the test sample. When $m = n$, the solution of $y = X_i\beta_i$ is $\hat{\beta}_i = X_i^{-1}y$. Consequently, the deviation $D_i = ||y - X_i\hat{\beta}_i||^2 = ||y - X_iX_i^{-1}y||^2 = 0$, $i = 1, 2, ...c$. It seems that LRC might perform uncomfortably in classifying the test sample y.

For some high-dimensional data in which a test sample vector and some training samples vector have different class distribution but on the same vector direction, LRC, INNC, and even SRC may lose their classification ability. The main reason is that these data points with the same direction would overlap each other after implementing the normalization [28].

Kernel methods have been used widely as nonlinear transformation technologies, which was originally used to construct the nonlinear support vector machines (SVMs) [34]. A Mercer kernel implicitly uses a nonlinear mapping that maps samples in an input space into a high or even infinite dimensional kernel feature space. In the kernel feature space, we can implement linear processing, i.e., a linear classification, with respect to a nonlinear classification in the input space [28]. In other words, kernel methods can make samples from nonlinear separable in an input space into linear separable in a kernel feature space.

In this paper, we propose a novel method, named kernel representation-based nearest neighbor classifier (KRNNC), which has the advantages of both kernel methods and representation-based methods. KRNNC takes into account the relationship among all training samples and dependently calculates distances between a test sample and each training sample. Moreover, KRNNC overcomes the above two drawbacks. In other words, KRNNC can not only classify low-dimensional data well, but also obtain a competitive classification result on high-dimensional data. The main reasons are that (1) in a kernel feature space, the relation among samples, such as the vector direction relation, associated with an original space may be changed so that the separability among samples might be enlarged. (2) In a kernel feature space, the dimension of low-dimensional data associated with an original space become very high so that the deviation between each class sample and a test sample has discrimination. In the kernel feature space, we first assign a weight (representation coefficient) to each training sample by using representation ability of each training sample to represent a test sample. Then we calculate the distance between the test sample and the result of multiplying each training sample by their corresponding coefficient. We finally classify the test sample into the same class as the training sample that has the minimum distance. The proposed method uses a special distance metric to measure the similarity between each training sample and test sample. From the way of classifying a test sample, KRNNC can be regarded as the nonlinear extension of INNC. What's more, we discussed in detail the underlying classification mechanism of KRNNC, which can be regarded as a theoretical supplement of INNC. The classification mechanism is that the collaborative decision-making of all training samples plays the essential role for classifying a test sample into the correct class.

Fig. 1 visually shows position relations among different samples in a kernel feature space, a transform result of an original input space. When a test sample (d) comes, we represent (d) by using a linear combination of all training samples and then each training sample has a corresponding representation coefficient that represents the contribution of each training sample to represent (d). In other words, each training sample moves to an appropriate 'place' by multiplying each training sample by their corresponding coefficient, such as (b) and (e) move to (b') and (e'), respectively. It is very clear that the distance between (d) and (b') is smaller than that between (d) and (e'). KRNNC correctly classifies (d) into the same class as (b'). It can be seen that we use a nonlinear mapping to transform all samples into a kernel feature space in which test samples are correctly classified. The proposed method is simple and computationally efficient. Our experimental results show that KRNNC can achieve higher classification accuracies than other methods, especially facing with very low-dimensional data (see Table 2). The contributions of the proposed method are: first, it overcomes the drawbacks that representation-based methods, i.e., INNC and LRC, have quite limited classification ability when they classify low-dimensional data and some high-dimensional data in which sample vectors belonging to different class distribution but have the same vector direction. Second, it clearly presents a theoretical analysis of the classification mechanism of KRNNC, which can be regarded as a theoretical supplement of INNC presented in [32]. Third, it provides a very large number of experiments on