Neurocomputing 195 (2016) 137–142

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Self-representation nearest neighbor search for classification

Shichao Zhang [a,*], Debo Cheng [a], Ming Zong [a], Lianli Gao [b]

[a] Guangxi Key Lab of Multi-source Information Mining & Security, College of Computer Science and IT, Guangxi Normal University, Guilin, 541004, China
[b] University of Electronic Science and Technology of China, Chengdu, China

ARTICLE INFO

Article history:
Received 17 March 2015
Received in revised form
4 June 2015
Accepted 7 August 2015
Available online 5 February 2016

Keywords:
k nearest neighbors
Sparse coding
Self-representation
Decision tree
Reconstruction

ABSTRACT

This paper proposes a self-representation method for kNN (*k* nearest neighbors) classification. Specifically, this paper first designs a self-reconstruction method to reconstruct each data point by all the data, and the derived reconstruction coefficient is then used for calculating the *k* value for each training sample. Furthermore, a decision tree is built with the resulting *k* values for each data point to output labels of the training samples. With the built decision tree, the proposed method classifies test samples. Finally, the experimental results on real datasets showed the proposed method outperformed the state-of-the-art methods.

1. Introduction

Since the *k* nearest neighbor (kNN) method has been widely applied in various applications, such as pattern recognition, cancer diagnosis, and text classification, it has been selected as one of top-10 data mining algorithms [16]. The kNN method first identifies *k* nearest training samples for a test sample, and then predicts the test sample with the major class among the *k* nearest training samples [3]. Specifically, first, the kNN method uses Euclidean distance in Eq. (1) (or other metric learning methods) to measure the difference or similarity between training samples and test samples. The Euclidean distance $Dist(x_i, x_j)$ between two data points x_i and $x_j$is defined as follows.

$$Dist(x_i, x_j) = \sqrt{\sum_{i=1}^{n} (x_i - x_j)^2} \tag{1}$$

The kNN method then uses the major class of x's k nearest neighbors to estimate the unknown sample x:

$$c(x) = \arg\max_{c \in C} \sum_{i=1}^{k} \delta(c, c(y_i)) \tag{2}$$

where $y_1, y_2, ..., y_k$ are the k nearest neighbors of x, k is the number of the neighbors, c denotes the finite set of class labels and $\delta(c, c(y_i))$ is an indicator, and $\delta(c, c(y_i)) = 1$ if $c = c(y_i)$; $\delta(c, c(y_i)) = 0$ otherwise.

To improve the performance of conventional kNN method, different methods have been proposed for gaining a higher accuracy [2,12,15,17,20]. For example, some literatures [11,13,14] focused on constructing a reasonable distance function and setting appropriate k values to improve the classification accuracy. However, a few literature have been focused on the issue of setting different k values for different samples. That is, most of kNN algorithms used a fixed k for predicting each test sample [15,19]. This often leads to inaccurate predictions [21]. Recently, literatures on setting different k values for different samples have demonstrated the potential power for all kinds applications [2,26]. For example, Lall et al. mentioned that the appropriate k should assign $k = \sqrt{n}$ to the datasets with the sample size larger than 100 [7]. Mitra et al. thought that the k value should assign $k = \lceil \sqrt{N} \rceil$ to the test sample (N is the number of training samples and the symbol '[]' denotes the greatest integer function) [10]. Sahigara proposed to obtain the appropriate k value by Monte Carlo validation for QSAR analysis [12], called ADNN in the experimental part of this paper. However, ADNN needs to repeat several times and to smooth the parameter k, i.e., it is a time consuming process.

This paper focuses on setting optimal k values for different test samples with kNN method based on the correlation among the samples. We employ a reconstruction method to obtain such correlation instead of traditional methods, such as the Euclidean distance and similarity estimate [34,35]. Specifically, the proposed Self Representation kNN (SR-kNN) method first employs the least square loss function to reconstruct the training samples for gaining the correlation coefficient matrix. And then the correlation coefficient matrix is used to obtain the kth different most relevant training samples. Furthermore, the corresponding various k values

* Corresponding author. Tel.: +86 13707738539.
E-mail address: zhangsc@mailbox.gxnu.edu.cn (S. Zhang).

http://dx.doi.org/10.1016/j.neucom.2015.08.115
0925-2312/

of the training samples are taken as labels to build a decision tree. Consequently, the best value of alterable k is learned at training process and used by kNNC for all test samples. The experimental results on real medical datasets verify that the time consuming of our approach is approximate to the state-of-the-art kNN method, but less than traditional methods. We also indicate that the proposed method outperforms the state-of-the-art methods in terms of classification accuracy.

The rest of this paper is organized as follows. In Section 2, we briefly recall related work about the improved kNN method. We provide the detail of our approach in Section 3. The experiment analysis is presented in Section 4. Finally, Section 5 concludes this paper.

2. Related work

The kNN algorithm has been successful applied in missing value imputation, regression and classification [2,5,29,36]. For example, Zhang et al. proposed a Grey-Based distance measure for kNN iteration imputation algorithm to improve the imputation performance [21,28,27,30]. Burba et al. used some asymptotic properties to the kNN method for regression [1]. Goldberger et al. proposed a distance metric learning algorithm to improve kNN method for classification [4].

Recently, many literatures have been proposed to improve the kNN method on different aspects, such as distance metric and classification rule. For example, Li et al. proposed a random kNN criterion for significantly stable, robust and fast learning [9]. Yigit proposed an Artificial Bee Colony (ABC) algorithm to find the best distance weight for each test sample in kNN method [19]. Lan et al. described the Multi-Source k-Nearest Neighbor (MS-kNN) algorithm for protein function prediction [8]. Su improved the kNN classifier and employed a genetic algorithm for clustering [13]. Therefore, we propose to obtain a flexible k value for kNN algorithm and reduce the time consuming via creating a decision tree for the test samples.

Regarding the studies on the distance metric of kNN methods, Younes et al. considered the dependencies between labels with the kNN rule for multi-class classification problems [20]. Parameswaran et al. defined a large margin multi-tasks metric learning used in the kNN method [11]. Jiang et al. synthesized the improved distance function in the kNN method to improve the kNN drawbacks [6].

Regarding the study about the optimum k value of kNN methods, Xie et al. proposed a model to learn different k values for each data points, but this method needs high time consuming [17]. Taneja et al. used the leave-one-out cross-validation method to determine k value for each data point and such a method is also time consuming [14]. Xu et al. used a kinds of support vector machine to increase accuracy by taking the parameter k into account act an important role, and analyze its influence [18]. Cheng et al. proposed a novel data-driven method to obtain unfixed k values for kNN algorithm based on sparse learning [2].

3. Approach

In this section, we present the basic concepts and describe the proposed method. Finally, we proposed an optimization method for solving the resulting objective function.

3.1. Notation

In this paper, we denote matrices as boldface uppercase letters, vectors as boldface lowercase letters, and scalars as normal italic letters, respectively. Moreover, we denote the Frobenius norm, ℓ_2- norm of a matrix $\mathbf{X}$ and ℓ_1- norm of a matrix $\mathbf{X}$ as $\|\mathbf{X}\|_F=\sqrt{\sum_i\|\mathbf{x}^i\|_2^2}=\sqrt{\sum_j\|\mathbf{x}_j\|_2^2}$, $\|\mathbf{X}\|_2=\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}|x_{ij}|^2}$ and $\|\mathbf{X}\|_1=\sum_i\|\mathbf{X}^i\|_2$, respectively. We further denote the transpose operator, the trace operator, and the inverse of a matrix $\mathbf{X}$ as X^T, $Tr(X)$, and X^{-1}, respectively.

3.2. Self-representation for sparse reconstruction

Given training samples $X=[x_1,x_2,...,x_n]\in R^{n\times m}$, where n and m are the number of samples and features, respectively, we use a linear function to represent each sample $x_j\in R^m$ (where $j=1,\ldots,n$) as:

$$x_j=\sum_{i=1}^{n}z_ix_i+e_i \tag{3}$$

where $z_i\in R^m$ and $e_i\in R^m$, respectively, are the dictionary of x_i and the error term. By simple transformation following [22], we change Eq. (3) into its matrix form as:

$$\mathbf{X}=\mathbf{ZX}+\mathbf{E} \tag{4}$$

where $\mathbf{Z}=[z_{ij}]\in R^{n\times n}$, Eq. (4) is a sample of self-representation model, $\mathbf{E}$ is the error term.

In this work, we use the characteristics of self-representation of the training samples to reconstruct themselves for generating the representation matrix $\mathbf{Z}\in R^{n\times n}$. To do this, we define the following objective function:

$$\arg\min_{\mathbf{Z}}\|\mathbf{ZX}-\mathbf{X}\|_F^2 \tag{5}$$

where $\|\mathbf{ZX}-\mathbf{X}\|_F^2$ denotes the reconstruction error. The matrix $\mathbf{Z}$ reflects the importance of different samples to make the representation of the reconstruction error small. Due to Eq. (3) is a convex function, so we can easily get the solution: $\mathbf{Z}=(\mathbf{XX}^T)^{-1}\mathbf{XX}$. The term $(\mathbf{XX}^T)^{-1}$ is not always reversible in real application. As we all know, the ℓ_2- norm is used to avoid the irreversible problem. Thus we use the ℓ_2- norm regularization term to avoid this problem, and the objective function is rewrite as follows:

$$\arg\min_{\mathbf{Z}}\|\mathbf{ZX}-\mathbf{X}\|_F^2+\mu\|\mathbf{Z}\|_2^2 \tag{6}$$

where μ is a parameter of the ℓ_2- norm regularization term. We can get the solution of Eq. (6): $\mathbf{Z}=(\mathbf{XX}^T+\mu\mathbf{I})^{-1}\mathbf{XX}$, $\mathbf{I}$ denote an identity matrix. The representation matrix $\mathbf{Z}$ does not have the sparsity, and cannot reflect the number of the training samples with the other adjacent training samples, while the ℓ_0- norm regularization term can produce sparsity to represent $\mathbf{X}$ using as few entries of $\mathbf{X}$. Therefore, we use the ℓ_0-norm regularization term to instead of the ℓ_2- norm regularization term, this can be formally expressed as follows:

$$\arg\min_{\mathbf{Z}}\|\mathbf{ZX}-\mathbf{X}\|_F^2+\gamma\|\mathbf{Z}\|_0 \tag{7}$$

where the constant γ is a tuning parameter and $\|\mathbf{Z}\|_0$ is the pseudo ℓ_0- norm. Unfortunately, this criterion is not convex and the process of searching its solution is NP-hard. Recently, the ℓ_1- norm regularization term has been proved to lead to the sparse correlation coefficient matrix. Moreover, its optimal solution is a convex optimization issue. Therefore, we use the ℓ_1- norm regularization term to instead of the ℓ_0- norm regularization term for gaining an optimal sparse representation coefficient matrix $\mathbf{Z}$. Its objective function is rewritten as follows:

$$\arg\min_{\mathbf{Z}}\|\mathbf{ZX}-\mathbf{X}\|_F^2+\rho\|\mathbf{Z}\|_1 \tag{8}$$

where ρ is a tuning parameter of the ℓ_1- norm regularization