Pattern Recognition Letters 109 (2018) 44–54

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

A novel *k*NN algorithm with data-driven *k* parameter computation

Shichao Zhang[a], Debo Cheng[a,b,*], Zhenyun Deng[a], Ming Zong[c], Xuelian Deng[d]

[a] *Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, Guangxi, China*
[b] *Information Technology and Mathematical Sciences, University of South Australia, Adelaide, Australia*
[c] *Institute of Natural and Mathematical Sciences, Massey University, Auckland, New Zealand*
[d] *College of Public Health and Management, Guangxi University of Chinese Medicine, Nanning, Guangxi, China*

ARTICLE INFO

Article history:
Available online 4 October 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
kNN method
kNN prediction
Parameter computation

ABSTRACT

This paper studies an example-driven *k*-parameter computation that identifies different *k* values for different test samples in *k*NN prediction applications, such as classification, regression and missing data imputation. This is carried out with reconstructing a sparse coefficient matrix between test samples and training data. In the reconstruction process, an ℓ_1-norm regularization is employed to generate an element-wise sparsity coefficient matrix, and an LPP (Locality Preserving Projection) regularization is adopted to keep the local structures of data for achieving the efficiency. Further, with the learnt *k* value, *k*NN approach is applied to classification, regression and missing data imputation. We experimentally evaluate the proposed approach with 20 real datasets, and show that our algorithm is much better than previous *k*NN algorithms in terms of data mining tasks, such as classification, regression and missing value imputation.

1. Introduction

The *k*NN (*k* Nearest Neighbors) algorithm is a non-parametric, or an instance-based, or a lazy method, and has been regarded as one of the simplest method in data mining and machine learning [27,37,38]. The principle of *k*NN algorithm is that the most similar samples belonging to the same class have high probability. Generally, the *k*NN algorithm first finds *k* nearest neighbors of a query in training dataset, and then predicts the query with the major class in the *k* nearest neighbors. Therefore, it has recently been selected as one of top 10 algorithms in data mining [32].

As well known, *k*NN algorithm is often sensitive to the selection of the *k* value. Although efforts have been focused on this topic for a long time, setting *k* value is still very challengeable in *k*NN algorithm [40]. Lall and Sharama mentioned that setting a suitable *k* should satisfy $k = \sqrt{n}$ for training datasets with sample size larger than 100 [21]. Ghosh investigated a Bayesian method for guiding us well in selecting k mainly [11]. Mitra et al. thought it is without any theories to guarantee that $k = \left[\sqrt{n}\right]$ is suitable for each test sample. Liu et al. pointed out, it has been proved that a fixed *k* value is not suitable for many test samples in a given training dataset [22].

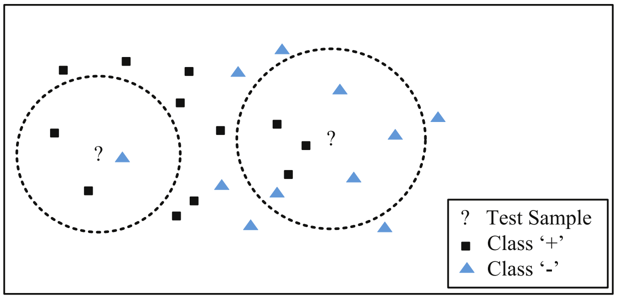

Fig. 1. Training examples for *k*NN classification.

We now illustrate the above limitations of *k*NN algorithm with a fixed *k* value in Figs. 1 and 2. Fig. 1 is an example of a binary classification task, where the classes training samples are marked as '+' and '-' respectively, and the labels of test samples are marked with the symbol '?'. Fig. 2 is an example of a missing data imputation, where the symbol '?' stands for data with missing values.

In Fig. 1, when setting *k*=5 for the *k*NN algorithm, there are two test samples that are predicted to '+' class according to the *k*NN rule. And the left test sample is incorrectly predicted. When setting *k*=1, the test samples are both incorrectly predicted. From the training examples, it is reasonable to take *k*=3 and *k*=7 for the left test sample and the right one, respectively.

[*] Corresponding author at: Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, Guangxi, China.

E-mail addresses: zhangsc@mailbox.gxnu.edu.cn (S. Zhang), cheng7294@foxmail.com (D. Cheng).

https://doi.org/10.1016/j.patrec.2017.09.036
0167-8655/© 2017 Elsevier B.V. All rights reserved.

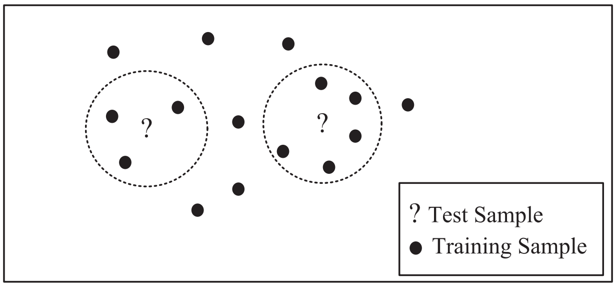

Fig. 2. Training examples for *k*NN regression/ missing value imputation.

For a similar scenario of missing value imputation in Fig. 2, the right and left test samples should be assigned different *k*, *i.e.*, k=3 and k=5 respectively. This scenario also indicates that different test samples should take different numbers of nearest neighbors in real *k*NN prediction applications. That is to say, setting a fixed constant for all test samples may often lead to low prediction rates in real classification applications.

Motivated by the above facts, this paper proposes a *k*-parameter computation for *k*NN approximate prediction based on Sparse learning, called S-*k*NN[1] [7]. The *k*-parameter computation can identify different *k* values for predicting different test samples with *k*NN algorithm. This is carried out by reconstructing a sparse coefficient matrix between test samples and training data [48,49]. With the matrix, an optimal *k* value can be obtained for each test sample one by one. In the reconstruction, a least square loss function is applied to achieve the minimal reconstruction error, and an norm regularization is utilized to result in the element-wise sparsity (*i.e.*, the sparse codes appear in the element of the coefficient matrix) for generating various *k* values for different test samples [46,47]. We also employ the Locality Preserving Projection (LPP) regularization to preserve the local structures of data during the reconstruction process, aiming to further improve the reconstruction performance [14,18]. The proposed S-*k*NN algorithm is experimentally evaluated against data mining tasks, such as classification, regression and missing value imputation. Comparing with previous *k*NN algorithms, the main contributions are as follows.

- Existing *k*NN approximate prediction algorithm is with a fixed *k* value for the whole problem space. The S-*k*NN algorithm identifies an optimal *k* value for each test sample, *i.e.*, the parameter *k* can be different for different test samples.
- Different from conventional Least Absolute Shrinkage and Selection Operator (LASSO) [20,29], our approach takes the local structures of samples into account.
- This paper proposes a novel optimization method to solve the designed objective function.

The remainder of the paper is organized as follows. Section 2 briefly reviews related *k*NN methods for classification, regression and missing value imputation. Section 3 is the main body of our S-*k*NN method. The proposed method is experimentally evaluated with real datasets in Section 4. Finally, this research is concluded in Section 5.

2. Related work

The study of *k*NN method has been a hot research topic in data mining and machine learning since the algorithm was proposed in 1967 [8]. In this section, we briefly review the applications of *k*NN algorithm in data mining tasks, such as classification, regression and missing value imputation.

2.1. Classification

KNN classification algorithm first selects *k* closest samples(*i.e.*, *k* nearest neighbors) for a test sample from all the training samples, and then predicts the test sample with a simple classifier, *e.g.*, majority classification rule. Liu et al. designed a new anomaly removal algorithm under the framework of *k*NN classification [22], which adopts mutual nearest neighbors whose advantage is that pseudo nearest neighbors can be identified instead of *k* nearest neighbors to determine the class labels of unknown samples. Weinberger et al. used semi-definite programming to learn a Mahanalobis distance metric for *k*NN classification, and adopted the target that *k* nearest neighbors always belong to the same class to optimize the measure metric, which samples from different classes are separated by a large margin [31]. Moreover, Goldberger et al. proposed a novel non-parametric *k*NN classification that learns a new quadratic distance metric and calls neighborhood component analysis (NCA) method [12]. This method focuses on the learned distance to be low-rank, so as to saving the storage and search costs. Jamshidi and Kaburlasos proposed an effective synergy of the Intervals' Number *k*-nearest neighbor classifier, and the gravitational search algorithm (GSA) for stochastic search and optimization [19]. Saini et al. presented an application of *k*-Nearest Neighbor (*k*NN) algorithm as a classifier for detection of QRS-complex in ECG [28]. This algorithm uses a digital band-pass filter to reduce the interference present in ECG false detection signal. For avoiding the influence of *k* value, Varmuza et al. used the repeated double cross validation method to search an optimum *k* for *k* nearest neighbor classification [30].

2.2. Regression

The *k*NN regression has been widely used and studied for many years in pattern recognition and data mining. In regression analysis, Burba et al. utilized kernel estimator based some asymptotic properties of the *k*NN to improving the performance of *k*NN regression [4]. Moreover, the purpose of their work utilized local adaptive bandwidth to study the non-parametric *k*NN algorithm. Ferraty and Vieu utilized the functional version of the Nadaraya–Watson kernel type estimator to construct the non-parametric characteristics of *k*NN algorithm for estimation, classification and discrimination on high dimensional data [10]. In the theory of *k*NN algorithm, Mack studied the L^2 convergence and the asymptotic distribution [23], and Devroye proved the strong consistency and the uniform convergence of *k*NN algorithm [9]. Hu et al. proposed a data-driven method for the battery capacity estimation, and used a non-linear kernel regression model based on the *k*NN to capture the dependency of the capacity on the features. This work also utilizes the adaptation of particle swarm optimizations to find the feature weights for the *k*NN regression model [17]. Goyal et al. took the interrelatedness of these metrics into account and statistically established the extent to improve the explanatory power of multiple linear regression. And then they conducted stepwise regression to identify influential metrics to avoid over fitting of data, and proposes suitability of *k*NN regression in the development of fault prediction model [13]. Cycle time of wafer lots for semiconductor fab was a critical task, therefore, Ni et al. combined the particle swarm optimization with a Gaussian mutation operator and a simulated weight of the features for *k*NN regression, and then used it to predict the cycle time of wafer fab [25]. Zhou proposed semi-supervised regression with co-training [41], which employed two *k*NN regressors with different distance metrics, each of which la-

[1] In this manuscript, we rewrote the parts (*i.e.*, Sections 1 and 4) and added the parts (*i.e.*, Sections 2.1, 2.2, 2.3, 3.3, and 3.4), compared to our former conference version.