Pattern Recognition Letters 110 (2018) 36–43

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Redefining nearest neighbor classification in high-dimensional settings

Julio López [a], Sebastián Maldonado [b,*]

[a] Facultad de Ingeniería y Ciencias, Universidad Diego Portales, Av. Ejército 441, Santiago, Chile
[b] Facultad de Ingeniería y Ciencias Aplicadas, Universidad de los Andes, Mons. Álvaro del Portillo 12455, Las Condes, Santiago, Chile

ARTICLE INFO

Article history:
Received 21 July 2017
Available online 27 March 2018

Keywords:
Nearest neighbor classification
High-dimensional datasets
Distance metrics

ABSTRACT

In this work, a novel nearest neighbor approach is presented. The main idea is to redefine the distance metric in order to include only a subset of relevant variables, assuming that they are of equal importance for the classification model. Three different distance measures are redefined: the traditional squared Euclidean, the Manhattan, and the Chebyshev. These modifications are designed to improve classification performance in high-dimensional applications, in which the concept of distance becomes blurry, i.e., all training points become uniformly distant from each other. Additionally, the inclusion of noisy variables leads to a loss of predictive performance if the main patterns are contained in just a few variables, since they are equally weighted. Experimental results on low- and high-dimensional datasets demonstrate the importance of these modifications, leading to superior average performance in terms of Area Under the Curve (AUC) compared with the traditional k nearest neighbor approach.

1. Introduction

The k Nearest Neighbor (k-NN) classifier [5] is a well-known pattern recognition method that has been used widely in several applications. Simplicity is its main virtue, allowing the classification of two or more patterns based on a quite simple rule: a test sample will belong to the class that the majority of its k nearest neighbors belongs to Han and Kamber [9].

Due to this simplicity, the k-NN method has several issues to deal with. Two main shortcomings, which are related to high-dimensional applications, are discussed in this paper. First, metrics such as the Euclidean distance may not be suitable in this context, since the concepts of distance and proximity are ill defined [10,20]. A second issue is feature relevancy: in contrast to methods such as logistic regression or Support Vector Machines, the feature importance cannot be derived with the original version of k-NN, and all variables are assumed to be equally important in obtaining the neighbors [9]. This fact can cause poor prediction if most variables are irrelevant, 'diluting' the patterns present in the relevant variables. Nowadays, there are several applications that have hundreds or even thousands of potentially redundant or irrelevant variables. In most cases, all the information is collected at once, and it is not clear which variables are relevant a priori. For such applications, models are required for helping us disentangling the signal from the noise.

In this work, these two issues are taken into account in order to design robust k-NN classifiers for both low- and high-dimensional settings. Three different distance metrics (Euclidean, Manhattan, and Chebyshev) are studied and modified in order to incorporate only a subset of the available information. Filter methods for feature selection are embedded in the definition of the distance metric, in order to encourage sparsity based on only the most relevant variables for the problem.

The remainder of this paper is organized as follows: previous work on k-NN is discussed in Section 2. The proposed framework for k-NN classification based on novel distance metrics is described in Section 3. In Section 4, experimental results using binary classification datasets are given. Finally, the main conclusions of this study and ideas for future developments are presented in Section 5.

2. k-NN classification

The k Nearest Neighbor method is arguably the simplest pattern recognition method for classification [9]. Given a fixed value for k, i.e., the number of neighbors, this nonparametric approach assigns the class label y^* to an unlabeled sample $\mathbf{x}^*$, which occurs most frequently in its neighborhood of k closest examples from the training set [5].

Formally, given two sets of m training tuples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, where $\mathbf{x}_i \in \Re^n$ and $y_i \in \{-1, +1\}$ are its respective class labels, and given mt test samples $\{\mathbf{x}_l^t\}_{l=1}^{mt} \subset \Re^n$, this method first computes the distance between an unlabeled sample $\mathbf{x}_l^t$ and all training examples $\mathbf{x}_i$, for $i = 1, \ldots, m$. Assuming a set $\mathcal{S}$ that contains all variables

* Corresponding author.
E-mail address: smaldonado@uandes.cl (S. Maldonado).

https://doi.org/10.1016/j.patrec.2018.03.023
0167-8655/

($|\mathcal{S}| = n$), the squared Euclidean distance is usually used for this task [9]:

$$d_{l_2}(\mathbf{x}_l^t, \mathbf{x}_i) = \sum_{j \in \mathcal{S}} (x_{l,j}^t - x_{i,j})^2. \quad (1)$$

Next, the k training observations that are closest to $\mathbf{x}_l^t$ are selected, i.e., the k elements with lowest $d_{l_2}(\mathbf{x}_l^t, \mathbf{x}_i)$ for all $i = 1, \ldots, m$. The label assigned to $\mathbf{x}_l^t$ is the most frequent one among these k elements.

Several improvements to the traditional k-NN algorithm have been developed in recent years. One research line involves using alternative distance measures for improving performance [17,26,27] or dealing with different types of data [4,6]. For example, a penalty dissimilarity measure was proposed in Datta et al. [6] in order to deal with missing information. Cost and Salzberg [4] proposed a weighted measure for handling symbolic features. Variations of the Minkowski distance have been used previously in domains such as anomaly [19] and intrusion detection for preventing network attacks [17].

An important aspect of this research is adapting the k-NN distance metric for dealing with high dimensionality. Few studies have been proposed in this direction. Hastie and Tibshirani [10] proposed a locally adaptive strategy to try to ameliorate this course of dimensionality in k-NN classification, and Pal et al. [20] proposed a dissimilarity measure based on mean absolute differences between inter-point distances. The latter strategy reduces the negative effects caused by a high dimensionality, such as the concentration of pairwise distances, thus improving predictive performance.

A related research line uses the k-NN principles for performing feature selection. Navot et al. [18] proposed a feature-weighted k-NN version for simultaneous regression and feature selection. This strategy was used to model cortical neural activity. Li et al. [13] developed an ensemble strategy based on various k-NN classifiers, which were constructed based on random subsets of variables. This approach, which resembles the reasoning behind random forest, can be used as feature ranking, and subsequently for performing backward feature elimination. Another feature selection method that uses the ideas behind random forest and k-NN was proposed by Park and Kim [21].

Other heuristics have been used for performing feature selection and k-NN classification. For example, Tahir et al. [23] proposed a hybrid approach based on a Tabu search for simultaneous feature weighting and classification. Lee et al. [12] used genetic algorithms for dealing with the issue of having various scales in the datasets. The authors proposed an efficient k-NN reference set editing strategy for maximizing accuracy, while reducing running times and memory resources.

Efficiency has also been a relevant topic in the k-NN literature. Beliakov and Li [1] proposed an efficient strategy for replacing the sort operation, by using order statistics and parallel computing via GPUs. Li et al. [14] developed a strategy for reducing the number amount of target samples to be considered by creating partial sets of the nearest neighbors.

3. Proposed strategy for nearest neighbor classification

Dimensionality reduction is quite an important topic in pattern recognition. A low-dimensional data representation reduces the risk of overfitting by constructing simple models with few parameters, yielding to a better predictive performance. It also provides a better understanding of the outcome of the model while reducing storage and acquisition costs [15]. In pattern recognition and image processing, dimensionality reduction is related with feature extraction, which corresponds to the process of constructing new features from the original dataset, aiming at reducing redundancy and identifying latent dimensions of the image that describe the data with sufficient accuracy.

Most methods are able to deal with noisy/irrelevant features by either removing them during the model training, or weighting them down when constructing a separating hyperplane. Decision trees fall in the first category, while methods such as logistic regression, SVM, or ANN on the second. In contrast, k-NN weights all variables equally, and it is usually outperformed by these alternative methods for this reason.

The main idea of the proposed approach is to adapt the classic k-NN method in order to deal with the two main issues pointed out in the introduction: the course of dimensionality faced by distance metrics such as the Euclidean norm, and the problem of having equal weights for all variables, which may lead to poor prediction if redundant or irrelevant variables, are included in the k-NN classification task.

Our contribution is twofold: first, we propose variations of the Minkowski distance that are more suitable under conditions of high-dimensionality, such as the Chebyshev distance or l_∞-norm. Additionally, we propose a modification of the Minkowski metric as the distance of two samples based only on a subset of the available variables, demonstrating that this modified Minkowski distance can, indeed, be considered as a distance measure.

Formally, the following distance metric is proposed for a given set of variables $\mathcal{U} \subset \mathcal{S}$, which is a subset of the full set of features $\mathcal{S}$, and two data objects $\mathbf{x}_k \in \Re^{|\mathcal{S}|}$ for $k = \{1, 2\}$:

$$d_{l_p,\mathcal{U}}(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|_{p,\mathcal{U}} = \left(\sum_{j \in \mathcal{U} \subset \mathcal{S}} |x_{1,j} - x_{2,j}|^p \right)^{1/p}, \quad (2)$$

for $p \geq 1$. This distance is designed to be used with $p \in \{1, 2, \infty\}$, i.e., the Manhattan, Euclidean, and Chebyshev distances, respectively. The proof that the proposed modified Minkowski distance satisfies the various properties required for being a distance measure is presented in Appendix A (see online supplementary material).

Next, the modified k-NN algorithm is proposed. The inputs of the model are the training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, the (unlabeled) test objects $\{\mathbf{x}_l^t\}_{l=1}^{mt}$, a predefined number of nearest neighbors k, a predefined number of selected attributes r, and the Minkowski distance parameter $p \in \{1, 2, \infty\}$. The output is the label vector for the test samples. The proposed strategy is formalized in Algorithm 1.

Algorithm 1 Modified k-NN method.

Input: Training tuples $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$; Test samples $\{\mathbf{x}_l^t\}_{l=1}^{mt}$; Number of nearest neighbors k; Feature ranking strategy FR; Number of selected attributes r; Distance parameter $p \in \{1, 2, \infty\}$.

Output: Test labels $\{\mathbf{y}_l^t\}_{i=1}^{mt}$.

1: $\mathbf{R} \leftarrow$ *Feature ranking resulting from using strategy FR on the training samples.*
2: $\mathcal{U} \leftarrow$ *subset of attributes corresponding to the r largest values of rank* $\mathbf{R}$.
3: **for** $l = 1, \ldots, mt$ **do**
4: Compute the distance between the sample $\mathbf{x}_l^t$ and all the training samples $\mathbf{x}_i$, $i = 1, \ldots, m$, using the modified Minkowski distance $d_{l_p,\mathcal{U}}(\mathbf{x}_l^t, \mathbf{x}_i)$.
5: $\mathcal{N}_l \leftarrow$ *Subset of the k nearest neighbors from the training set of the test sample* $\mathbf{x}_l^t$.
6: $\mathbf{y}_l^t \leftarrow$ *Label corresponding to the mode in* $\mathcal{N}_l$.
7: **end for**

The first step of the algorithm corresponds to the construction of the feature ranking $\mathbf{R}$. This ranking is constructed by sorting the variables according to its relevancy using, e.g. statistical measures.