Weighted Nearest Centroid Neighbourhood

Víctor Aceña(✉), Javier M. Moguerza, Isaac Martín de Diego, and Rubén R. Fernández

Rey Juan Carlos University, c/ Tulipán s/n, 28933 Móstoles, Spain
{victor.acena,javier.moguerza,isaac.martin,ruben.rodriguez}@urjc.es
http://www.datasciencelab.es

Abstract. A novel binary classifier based on nearest centroid neighbours is presented. The proposed method uses the well known idea behind the classic k-Nearest Neighbours (k-NN) algorithm: one point is similar to others that are close to it. The new proposal relies on an alternative way of computing neighbourhoods that is better suited to the distribution of data by considering that a more distant neighbour must have less influence than a closer one. The relative importance of any neighbour in a neighbourhood is estimated using the SoftMax function on the implicit distance. Experiments are carried out on both simulated and real data sets. The proposed method outperforms alternatives, providing a promising new research line.

Keywords: Nearest Neighbours · Classification · Nearest Centroid Neighbourhood · Parameter selection · Similarity measure

1 Introduction

The k-Nearest Neighbours (k-NN) algorithm is a basic method that assigns the most frequent class label among the k training points closest to the target point. The main idea of assigning a value based on the most similar points fits into the human thought of constant search for similarities between objects to compare them. Because of this and its computational simplicity, it is widely used in Machine Learning, Pattern Recognition or Data Mining algorithms.

The k-NN setting parameters are the number of neighbours and the distance function. However, some other factors should be considered in order to evaluate the performance of the method: the size and shape of the neighbourhood, determined by the way it is computed; the number of neighbours, caused by the choice of k; the similarity among neighbours, determined by the chosen distance; and finally, the relative importance of a neighbour, class, or feature.

In this paper, it is presented a classifier built by using the information from all these factors. The method is named wk-NCN, weighted k-Nearest Centroid Neighbourhood. A novel way of calculating neighbourhoods based on centroids is used. Thus, a method for calculating an adaptive neighbours to the data

H. Yin et al. (Eds.): IDEAL 2019, LNCS 11871, pp. 94–101, 2019.
https://doi.org/10.1007/978-3-030-33607-3_11

distribution is provided. Furthermore, the well-known SoftMax function, on the implicit distance, is used to estimate the proper weight of each neighbour.

The rest of the paper is structured as follows. Some of the related k-NN improvements are presented in Sect. 2. Section 3 addresses the fundamentals of the proposed *wk*-NCN classifier. The experiments and results for simulated and real data sets are detailed in Sect. 4. Section 5 concludes and provides future guidelines.

2 Related Work

The analysis and improvements of the k-NN method has been object of study since it was firstly proposed by Cover and Hart in 1967 [3]. In this section, a brief review on some of the variations of the original algorithm is presented, specially focused on the relevant factors regarding the performance of the algorithm: the size and shape of the neighbourhood, the distance metric, and the importance of a neighbour, class or feature.

One of the main issues of k-NN is the selection of the k parameter. That is, how to define the best number of neighbours for classification tasks. Many authors have proposed methods for finding the optimal k in different contexts. One of the most relevant and used contribution is the optimal k by Silverman [11]: $k = n^{4/(d+4)}$, where n is the sample size and d is the number of features in the data set. In the same way, Ghosh [6] uses Bayesian methods to estimate k as opposed to the classical cross-validation and likelihood cross-validation. Jaiswal, Bhadouria and Sahoo [9] present an automated parameter selector guided by *Cuckoo* search, in which k and the distance metric among four usual metrics, are optimised together by a meta-heuristic method.

Beyond a search for common distance metrics, many other flexible or adaptable metrics can be designed, such as the proposed by Hastie's and Tibshirani [7]. In order to create searching methods for adaptive parameters, Hulett, Hall and Qu [8] build an automatic selection of neighbouring instances as defined by a dynamic local region. Zhang et al. [12] propose the S-k-NN algorithm, which uses the reconstruction of the correlation between test samples and training samples in order to automatically calculate the optimal k for any test sample.

Adaptive methods based on building neighbourhoods dependent on the data distribution have been developed. For instance, Chaudhuri [2] proposes a new definition of neighbourhood to capture the idea that the neighbours should be as near to the target point and as symmetrically placed around it as possible. This method has been called Nearest Centroid Neighbourhood (NCN). Under this definition of neighbourhood García et al. [5] developed the kNCN model for regression tasks.

Finally, a remote neighbour should have less influence than a nearby neighbour on the decision for a test point. This idea has been theoretically proven by Samworth, Richard et al. [10] and developed by Biswas et al. [1].

The proposal presented in this paper is based on these two ideas: to build a data distributed based neighbourhood, and to differently weight the neighbours according to their distance to the point of interest. It is possible to obtain a

method that adapts to the data distribution by using the neighbourhood definition proposed in [2]. In addition, the effect of the nearest neighbours can be weighted in an effective way to create the weighted Nearest Neighbourhood method by using the SoftMax function.

3 Method

Let $X = \{(x_1, x_2, \ldots, x_n) \mid x_i \in \mathbb{R}^d\}$ be the set of training data points and $y = \{(y_1, y_2, \ldots, y_n) \mid y_i \in \{0, 1\}\}$ the set of its corresponding class labels. Let $N_k(S, p)$ be the k-neighbourhood of a point p in a data set S, that is, the k points in S nearest to p.

3.1 The k-NN and k-NCN classifiers

The main idea behind the k-NN classifier is quite simple: to estimate the probability of a class as the proportion of points in the neighbourhood of a point that belong to that class. It is formalised as follows. Given a training data set X, a positive integer k and a new point to be classified p, the k-NN classifier identifies the k neighbours in the training data set that are nearest to p: $N_k(X, p)$. Then, the conditional probability of class $j \in \{0, 1\}$ can be estimated using the following equation:

$$P(y = j|X = p) = \frac{1}{k} \sum_{i \in N_k(X,p)} I(y_i = j) \tag{1}$$

The k-NCN method (first presented in [2]) is also based on the neighbourhood of a point, dependent on a parameter k. However, the neighbourhood is based on centroids calculation. Thus, the set of k neighbours whose centroid is closest to the point p is selected. Following [2], the class label will be estimated as indicated in Algorithm 1. The input of Algorithm 1 are the training data, $X = \{x_1, x_2, \ldots, x_n\}$, the number of centroid neighbours k, the distance metric ($Dist$) to compute the similarity between points, and the target point p whose class will be estimated. The output is the radius that defines the neighbourhood around the target point p.

The intuition behind the calculation of k-NCN is to establish a symmetry within the neighbourhood with respect to a point p and to preserve the closeness of the points contained in such neighbourhood. This intuition is confirmed during the second iteration of the *while* loop in Algorithm 1, since the second neighbour is not the second nearest neighbour. Instead, the algorithm selects a point in a direction diametrically opposite to the first neighbour with respect to p. Thus, advantages over the k-NN method are achieved.

In low-density areas of the data set, the neighbourhood are quite bigger than the obtained using the classic nearest neighbourhood. In high-density areas the neighbourhood radius is roughly the same in both methods. These issues are shown in Fig. 1.

Once the neighbourhoods are computed, it is possible to estimate the conditional probability of class in the same way as in the k-NN method (Eq. 1).

Algorithm 1. Nearest centroid neighbours

Input: $(X, k, p, Dist)$
Output: r
$Q \leftarrow \emptyset$
$q_1 \leftarrow findNN(X, p)$ ▷ q_1 is the nearest neighbour
$Q \leftarrow q_1$
$X_{aux} \leftarrow X - \{q_1\}$
$r \leftarrow Dist(q_1, p)$
$j \leftarrow 1$
while $j \leq k$ **do**
  $j \leftarrow j + 1$
  $dist\_min \leftarrow \infty$
  for all $x_i \in X_{aux}$ **do**
    $M \leftarrow computeCentroid(Q \cup \{x_i\})$
    if $Dist(M, p) \leq dist\_min$ **then**
      $q_j \leftarrow x_i$
      $dist\_min \leftarrow Dist(M, p)$ ▷ update minimum distance
  $Q \leftarrow Q \cup \{q_j\}$
  $r \leftarrow max(r, Dist(q_j, p))$ ▷ update the radius when needed
  $X_{aux} \leftarrow X_{aux} - \{q_j\}$

3.2 *wk*-NCN classifier

The advantages of the k-NCN method over k-NN method could be a drawback in some situations. Notice that, when the neighbourhood is very wide, there could be points far from the target that contribute too much to the estimation of the class probability. One way to reduce this effect is to weight the contribution of each neighbour depending on the distance to the target. In this paper, it is proposed to incorporate this weighting by using the SoftMax function, defined as follows:

$$\sigma : \mathbb{R}^T \rightarrow [0,1]^T$$

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{t=1}^{T} e^{z_t}} \; j = 1, 2, \ldots, T$$

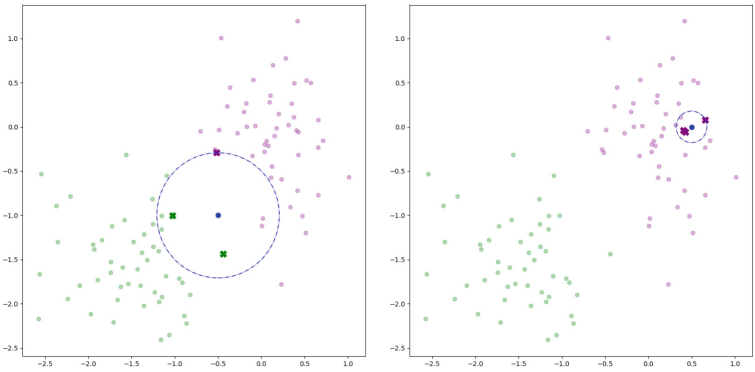
Thus, the estimation of the conditioned probability of a point is:

$$P(y = j|X = p) = \sum_{i \in N_k(X,p)} \frac{e^{-||x_i - p||_2}}{\sum_{i \in N_k(X,p)} e^{-||x_i - p||_2}} \cdot I(y_i = j) \tag{2}$$

where $||\cdot||_2$ defines the $L^2 - norm$. The *wk*-NCN method computes the nearest centroid neighbours as indicated in Algorithm 1, and estimates the probability of class using Eq. 2. Thus, the neighbourhood depends on the distribution of the data set and the neighbours are weighted based on their distance to the point of interest. The weights decrease exponentially with respect to the distance between each neighbour and the target point.

4 Experiments and Results

The performance of the proposed method is first evaluated on simulated data sets. Then, its relative performance regarding alternative methods is estimated on a battery of real data sets.



(a) Low density neighbourhood (b) High density neighbourhood

Fig. 1. Nearest Centroid neighbours examples. Comparison between high and low density areas. The figure shows the computed neighbourhoods by 3-NCN method of two example points. The two colours in the plot represents the category labels. Within each neighbourhood, the points marked are the 3 Nearest Centroid Neighbours, that define the neighbourhood.

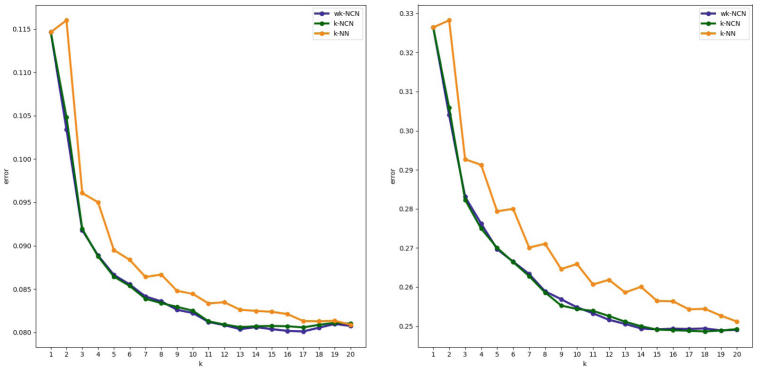
4.1 Simulated Data

Three different scenarios are considered: separated binary classes, overlapped binary classes, and highly overlapped binary classes. Thus, three binary classification data sets are generated using two bivariate normal distributions. In the first scenario, two separated classes are considered: $\mathcal{N}(\mu_i = 0, \sigma_i = 0.5)$ and $\mathcal{N}(\mu_i = -2, \sigma_i = 0.5)$, respectively. In the second scenario, two overlapped classes are considered: $\mathcal{N}(\mu_i = 0, \sigma_i = 0.5)$ and $\mathcal{N}(\mu_i = -1, \sigma_i = 0.5)$, respectively. In the third scenario, two highly overlapped classes are considered: $\mathcal{N}(\mu_i = 0, \sigma_i = 0.5)$ and $\mathcal{N}(\mu_i = -0.5, \sigma_i = 0.5)$, respectively.

With the settings, 150 samples of each class are generated for each example. In addition, 100 simulations are performed on each scenario. The methods wk-NCN, k-NCN and the classical k-NN are tested. In order to measure the error the leave one out evaluation method was considered. The average rate of misclassification over all simulations was calculated for each $k \in \{1, \ldots, 20\}$. As expected, in the scenario of separate classes the errors of the three methods are zero. But in the other two scenarios the methods wk-NCN and k-NCN outperform the k-NN for the same k-value as shown in Fig. 2. Given the normality of the generated data, there are no significant differences between the wk-NCN and k-NCN methods.

4.2 Real Data

In this experiment, the performance of the methods presented in the previous section is evaluated on real data sets obtained from the UCI repository [4]. A description of these data sets is detailed in Table 1. Each data set has been



(a) Overlapped data (b) Highly overlapped data

Fig. 2. Performance (proportion of misclassified instances) of the *wk*-NCN and alternative methods for simulated data sets.

divided into two sets: training set (70%) and test set (30%). Training sets are used to fit the models, and test sets are be used to evaluate the performance of the models.

The proposal, *wk*-NCN with k parameter selected by leave one out, is compared to six alternative methods: k-NCN and k-NN with k parameters selected by leave one out; k-NN where the parameter k is the optimal value proposed by Silverman, k-NN with k equals 1, that is 1-NN; Decision Tree and Random Forest models. Table 2 details the classification errors on the test sets for all the trained models.

Notice that the proposed *wk*-NCN method obtained better or equal results than the k-NCN method for all the considered data sets. Additionally, a model raking is built in order to achieve global summary of the results. For each data set, each model is scored according to its error in a bottom-up way. That is, the model with the smallest error has 1 point, the next one has 2 points, etc. Therefore, the lowest score is the best model. Figure 3 shows the average and

Table 1. Data sets summary

Data set	Samples	Features
Ionosphere	351	34
Mammographic	961	6
Congressional voting records	435	16
Breast cancer wisconsin (Diagnostic)	569	32
Banknote authentication	1372	5
Blood transfusion service center	748	5
Connectionist (Sonar, Mines vs. Rocks)	208	60

Table 2. Real data sets test errors (proportion of misclassified instances). Best results in bold for each data set.

Method	Test errors						
	Mammographic	Ionosphere	Congressional	Breast	Banknote	Blood	Connectionist
wk-NCN	**0.217**	0.151	0.100	0.041	**0.000**	0.213	**0.206**
k-NCN	0.229	0.151	0.100	0.053	**0.000**	0.227	**0.206**
k-NN (*k*-loo)	0.229	0.151	0.100	**0.029**	0.002	0.253	**0.206**
k-NN (*k*-Silverman)	0.225	0.151	0.086	0.035	0.015	**0.209**	**0.206**
1-NN	0.245	0.151	0.100	0.053	0.002	0.316	**0.206**
Decision tree	0.257	0.104	**0.014**	0.089	0.034	0.262	0.397
Random forest	0.233	**0.085**	**0.014**	0.029	0.005	0.284	0.333

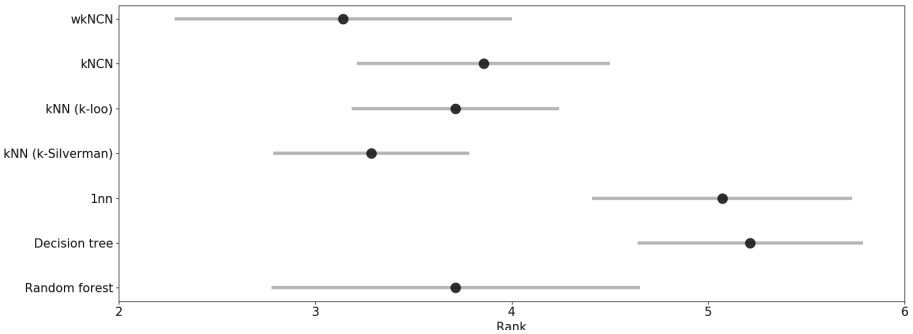


Fig. 3. Average rank and confidence interval for the proposed and alternative methods.

standard deviation for the rank of each method when all the data sets where considered. It is observed that the *wk*-NCN method obtains the best overall score, followed by the *k*-NN with *k* parameter selected by Silverman method. A hypothesis test to evaluate the differences among the methods is performed. The null hypothesis is that the alternative methods obtain lower error than the proposed one. The *wk*-NCN method statistically outperforms *k*-NCN, *k*-NN (k-loo), 1-NN (p-value < 0.01) and Decision Tree methods (p-value < 0.1).

5 Conclusions

This paper presents the weighted Nearest Centroid Neighbourhood method for binary classification. It is based on an alternative version of neighbourhood, that establish a symmetry within the neighbourhood with respect to the target point. Furthermore, the number of neighbours varies according to the distribution around the target point. The SoftMax function is used to weight the effect of each neighbour according to the implicit distance to the target.

The experimental results show that the *wk*-NCN outperforms some of the alternative methods. In particular, the proposed method equals or improves all

k-NCN experiments. This suggests that weighting the distances by the SoftMax function is effective. This opens the research line to look for other metrics to weight the effect of neighbours according to their distance.

Although wk-NCN is adaptable to the distribution of the data, this could be improved by considering two factors. First, a constant radius is generated for each neighbourhood. Therefore, an adaptive version of the neighbourhood should be explored without losing the useful symmetry structure provided by the method. Furthermore, the number of neighbours in the neighbourhood is inherently variable in this method. Therefore, when creating neighbourhoods, an adaptive k will be a natural variation.

Acknowledgements. Research supported by grant from the Spanish Ministry of Economy and Competitiveness, under the Retos-Colaboración program: SABERMED (Ref: RTC-2017-6253-1); Retos-Investigación program: MODAS-IN (Ref: RTI2018-094269-B-I00); and the support of NVIDIA Corporation with the donation of the Titan V GPU.

References

1. Biswas, N., Chakraborty, S., Mullick, S.S., Das, S.: A parameter independent fuzzy weighted k-nearest neighbor classifier. Pattern Recogn. Lett. **101**, 80–87 (2018)
2. Chaudhuri, B.: A new definition of neighborhood of a point in multi-dimensional space. Pattern Recogn. Lett. **17**(1), 11–17 (1996)
3. Cover, T.M., Hart, P.E., et al.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
4. Dua, D., Graff, C.: UCI Machine Learning Repository (2017). http://archive.ics.uci.edu/ml
5. García, V., Sánchez, J., Marqués, A., Martínez-Peláez, R.: A regression model based on the nearest centroid neighborhood. Pattern Anal. Appl. **21**(4), 941–951 (2018)
6. Ghosh, A.K.: On optimum choice of k in nearest neighbor classification. Comput. Stat. Data Anal. **50**(11), 3113–3123 (2006)
7. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification and regression. In: Advances in Neural Information Processing Systems, pp. 409–415 (1996)
8. Hulett, C., Hall, A., Qu, G.: Dynamic selection of k nearest neighbors in instance-based learning. In: 2012 IEEE 13th International Conference on Information Reuse & Integration (IRI), pp. 85–92. IEEE (2012)
9. Jaiswal, S., Bhadouria, S., Sahoo, A.: KNN model selection using modified cuckoo search algorithm. In: 2015 International Conference on Cognitive Computing and Information Processing (CCIP), pp. 1–5. IEEE (2015)
10. Samworth, R.J., et al.: Optimal weighted nearest neighbour classifiers. Ann. Stat. **40**(5), 2733–2763 (2012)
11. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Routledge, New York (2018)
12. Zhang, S., Cheng, D., Deng, Z., Zong, M., Deng, X.: A novel KNN algorithm with data-driven k parameter computation. Pattern Recogn. Lett. **109**, 44–54 (2018)