Instance Selection for the Nearest Neighbor Classifier: Connecting the Performance to the Underlying Data Structure

Vicente García[1], Josep Salvador Sánchez[2](✉), Alberto Ochoa-Ortiz[1], and Abraham López-Najera[1]

[1] Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, Chihuahua, Mexico
{vicente.jimenez,alberto.ochoa,abraham.najera}@uacj.mx
[2] Institute of New Imaging Technologies, Department of Computer Languages and Systems, Universitat Jaume I, Castelló de la Plana, Spain
sanchez@uji.es

Abstract. Instance selection is one of the most successful solutions to low noise tolerance of the nearest neighbor classifier. Many algorithms have been proposed in the literature, but further research in this area is still needed to complement the existing findings. Here we intend to go beyond a simple comparison of instance selection methods and correspondingly, we carry out a qualitative analysis of why some algorithms perform better than others under different conditions. In summary, this paper investigates the impact of instance selection on the underlying structure of a data set by analyzing the distribution of sample types, with the purpose of linking the performance of these methods to changes in the data structure.

Keywords: Nearest neighbor classifier · Instance selection · Editing · Sample types

1 Introduction

In supervised learning, one of the most popular models is the well-known and understood k-nearest neighbors (kNN) algorithm [6]. A query sample is assigned to the class represented by the majority of its k nearest neighbors in the training set (a collection of correctly classified instances). A particular case is when $k = 1$, in which a query sample is decided to belong to the class indicated by its closest neighbor. The 1NN classifier benefits from a number of advantages, which have made it useful and effective for many real-world problems: (i) it can be implemented easily due to its conceptual simplicity; (ii) it does not require any a priori probabilistic information relating to data; and (iii) the error rate for 1NN is at most twice the optimal Bayes error as the training set size tends to infinity.

A. Morales et al. (Eds.): IbPRIA 2019, LNCS 11867, pp. 249–256, 2019.
https://doi.org/10.1007/978-3-030-31332-6_22

However, the kNN algorithm suffers from some inherent weaknesses that may hinder its efficiency and effectiveness due to intensive computational requirements and extreme sensitivity to errors or noise in the training set. To tackle these issues, data reduction has been devised as a tool to obtain a reduced representation of the training set that can closely maintain the properties of the original data. Depending on the approach of the data reduction methods, these can be categorized into two groups [8]: *instance reduction* or *condensing* to lessen the size of the training set by removing redundant examples, and *instance selection* or *editing* to achieve a similar or even higher accuracy by eliminating erroneous and noisy samples. While condensing retains borderline samples (those that are close to the decision boundaries) and removes internal samples, the general strategy followed by editing is the opposite (i.e., the examples chosen to be removed are the borderline samples). In the case of editing, most studies have focused either on carrying out experimental comparisons over multiple data sets or on designing a meta-learning framework to choose the best performing algorithm through the analysis of some data characterization measures [5,12,14].

The aim of this paper is to contribute to further understanding of the effects of instance selection on the underlying structure of data sets and to explore for possible connections with the classification accuracy of 1NN. To this end, we will exploit local information to categorize the instances into four different groups (safe, borderline, rare, and outlier), compare their distribution in the original training set with that in the edited set, and relate the performance of some instance selection methods to changes in the data structure. Hopefully, this will allow to gain some insight into the reasons why the 1NN classification performance depends so heavily on the particular instance selection algorithm.

2 Categorization of Sample Types

In the context of learning from imbalanced data, several authors have proposed to distinguish between *safe* and *unsafe* samples according to their neighborhood [11,15,17]. The safe samples are placed in homogeneous regions with data from a single class and are sufficiently separated from instances belonging to any other classes, whereas the remaining samples are referred to as unsafe. The safe samples are correctly classified by most models, but the unsafe samples may make their learning especially difficult and more likely to be misclassified.

The common property of the unsafe samples is that they are located close to instances that belong to the opposite class. However, the unsafe samples can be further divided into three subtypes: *borderline*, *rare* and *outlier* [10,15]. The borderline samples are located closely to the decision boundary between classes. The rare samples form small data structures or clusters located far from the core of their class. Finally, the outliers are single samples that are surrounded by instances from the other class. In the present work, this categorization of samples types will not be used for class-imbalanced problems, but for the analysis of instance selection methods.

A straightforward method to identify each sample type consists of analyzing the local distribution of the data, which can be modeled either by computing their k-neighborhood or through a kernel function (this consists in setting a local area around the instance and estimating the number of neighbors and their class labels within it). It has been claimed that analyzing a local distribution of instances is more appropriate than using global approaches because the minority class is often formed by small sub-groups with difficult, nonlinear borders between the classes [15,17].

Algorithm 1. Identification of sample types for multi-class data

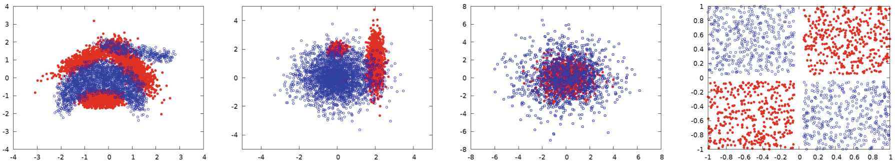
```
Input:
$S$ {Input data set}
$k$ {Neighborhood size}

Output:
$safe$ {Set of safe samples}
$borderline$ {Set of borderline samples}
$rare$ {Set of rare samples}
$outlier$ {Set of outlier samples}

for all $z_i \in S$ do
    $neighbors \leftarrow$ computeNeighbors($z_i, S - \{z_i\}, k$)
    $sameClass \leftarrow$ countSameClass($y_i, neighbors$)
    if $sameClass \geq \lfloor 0.8k \rfloor$ then
        $safe \leftarrow safe \cup \{z_i\}$
    else
        if $sameClass \geq \lfloor 0.5k \rfloor$ then
            $borderline \leftarrow borderline \cup \{z_i\}$
        else
            if $sameClass \geq \lfloor 0.2k \rfloor$ then
                $rare \leftarrow rare \cup \{z_i\}$
            else
                $outlier \leftarrow outlier \cup \{z_i\}$
            end if
        end if
    end if
end for
```

Suppose we have a data set, $S = \{z_i = (x_i, y_i)\}$, where $x_i \in X \subset \mathbb{R}^d$ is a vector of attributes describing the i-th instance and y_i is its class label. The type of a sample z_i is often decided by comparing the number of its k nearest neighbors that belong to the class of z_i with the number of neighbors of the opposite class. Following the procedure given in Algorithm 1, which is a generalization for multi-class data of the scheme proposed by Stefanowski and Wilk [20], a safe sample is characterized by having a neighborhood dominated by instances that belong to its same class, rare samples and outliers are mainly surrounded by instances from different classes, and the borderline samples are surrounded by instances both from their same class and also from a different class. Here we have used two functions: *computeNeighbors* to search for the k nearest neighbors of a sample z_i and store them in a vector named *neighbors*, and *countSameClass* to count how many of the k nearest neighbors belong to the class of z_i.

Most authors choose a fixed size $k = 5$ because smaller values may poorly distinguish the nature of instances and higher values would violate the assumption of local neighborhood. By using $k = 5$, an instance z_i will be defined as: (i) *safe* if at least 4 neighbors are from the class y_i; (ii) *borderline* if 2 or 3 neighbors belong to the class y_i; (iii) *rare* if only one neighbor belongs to the class y_i, and this has no more than one neighbor from its same class; and (iv) *outlier* if all its neighbors are from the opposite class.

3 Databases and Experimental Setting

Experiments were conducted on the artificial data sets depicted in Fig. 1, which are all two-dimensional and correspond to well-balanced binary classification problems. Using synthetic data allows to know their characteristics a priori and analyze the results in a fully controlled environment.



(a) Banana (b) Clouds (c) Gaussian (d) XOR

Fig. 1. Artificial data sets

The Banana data set is a non-linearly separable problem with 5,300 samples that belong to two banana shaped clusters [1]. The Clouds database has 5,000 samples where one class is the sum of three different normal distributions and the other class is a single normal distribution [3]. The Gaussian database consists of 5,000 instances where one class is represented by a multivariate normal distribution with zero mean and standard deviation equal to 1 and the other by a normal distribution with zero mean and standard deviation equal to 2 in all directions [3]. In the XOR database, a total of 1,600 random bivariate samples were generated following a uniform distribution in a square of length equal to 2, centered at zero (apart from a strip of width 0.1 along both axes); the samples were labeled at each quadrant to reproduce the well-known XOR problem, that is, the label of each point (x, y) was computed as $\text{sign}(x) \cdot \text{sign}(y)$ [2].

The stratified 10-fold cross-validation method was adopted for the experiments, thus preserving the prior class probabilities of a database and the statistical independence between the training and test blocks of each fold. The experiments were carried out as follows: (i) the training sets were preprocessed by various editing techniques, (ii) the 1NN classifier with the Euclidean distance was applied using each data set, and (iii) the proportion of each sample type in both the original sets and the filtered sets was recorded. Our hypothesis is that the analysis of the distribution of sample types in a data set may allow to explain the performance of each editing algorithm.

The instance selection techniques used in the experiments were: (1) all-kNN editing (aKNN) [21], (2) Wilson's editing (WE) [23], (3) editing with estimation of class probabilities and threshold (CPT) [22], (4) modified Wilson's editing (MWE) [9], (5) model class selection (MCS) [4], (6) Multiedit (MultiE) [7], (7) editing based on nearest centroid neighborhood (NCN) [19], (8) pattern by ordered projections (POP) [16], (9) editing based on relative neighborhood graph (RNG) [18], and (10) variable-kernel similarity metric (VSM) [13].

4 Results and Discussion

For each database, the graphs in Fig. 2 display to the accuracy rates of 1NN using both the original training set with no preprocessing (the dotted horizontal lines) and the collection of edited sets (vertical bars). The results for Banana, Clouds and XOR show that all instance selection methods had a very similar behavior: leaving aside the 10-VSM method, the 1NN classifier trained with the original training sets and with the edited sets performed equally well. The most interesting results were for the Gaussian database because most algorithms improved the performance achieved by 1NN using the original sets and even more important, some differences from one algorithm to another can be seen. However, the question here was why some methods performed better than others. Thus taking care of this objective, the next step in the experiments was to analyze the underlying structure of each data set and investigate any possible link between the distribution of sample types and the performance of instance selection methods.

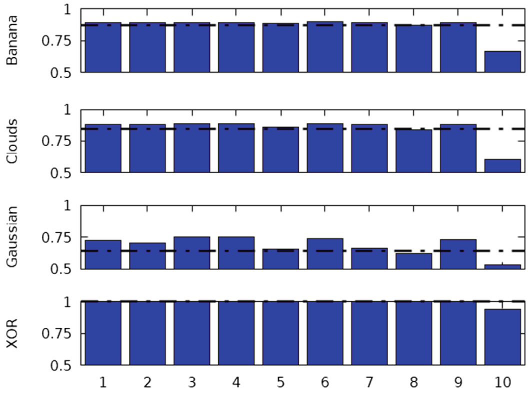
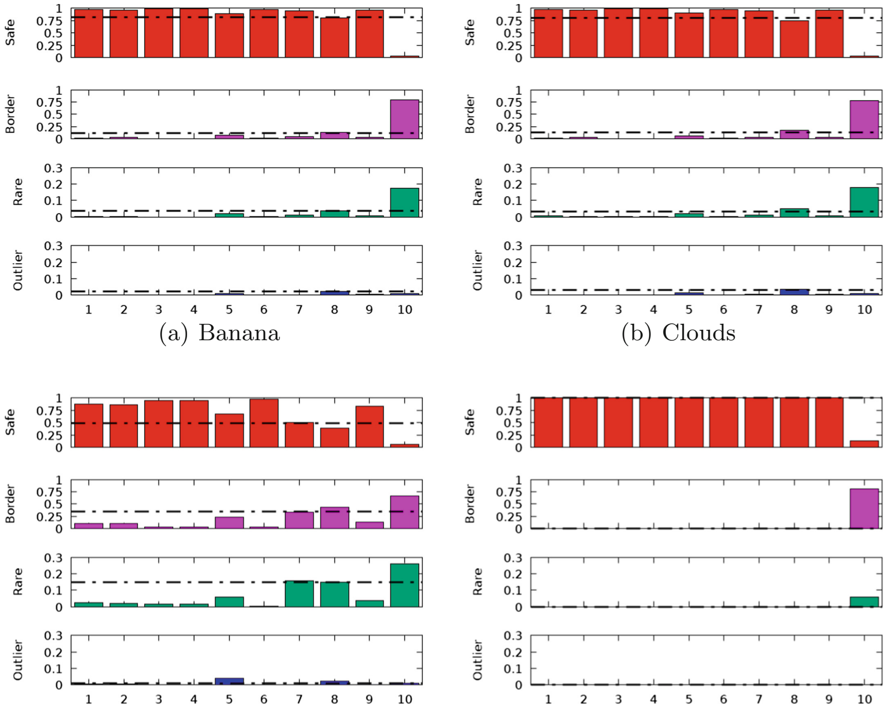


Fig. 2. Classification accuracy

The vertical bars in Fig. 3 represent the proportions of safe, borderline, rare and outlier samples in the sets after applying the editing algorithms. The dotted horizontal lines are for the proportions in the original sets, which should be interpreted as a reference value. A rapid comparison of the proportions of safe and unsafe samples in the original training sets reveal that the Gaussian database represents the most complex and interesting problem with very high class overlapping, and XOR corresponds to the easiest problem with linear class separability and no overlapping.



(a) Banana

(b) Clouds

(c) Gaussian

(d) XOR

Fig. 3. Proportion of sample types

Figure 3 shows that, as expected from the selection strategy of editing, most methods (8-POP and 10-VSM were the exception) led to an increase in the proportion of safe samples and a decrease in the proportion of unsafe samples (borderline, rare and outlier) compared to the reference values (original training sets). However, paying special attention to the Gaussian problem, one can note that there exist significant differences in the data structure resulting by the application of each algorithm; for instance, 3-CPT, 4-MWE and 6-MultiE were the methods with the highest increase of safe samples and also the highest decrease of unsafe samples, while the changes produced by 5-MCS and 7-NCN were negligible. This may explain the different behavior of 1NN trained by 3-CPT, 4-MWE or 6-MultiE and that trained by 5-MCS or 7-NCN as shown in Fig. 2. This observation reinforces our hypothesis and was supported by the fact that those algorithms with the largest positive changes in the underlying structure of data sets also achieved the highest accuracy rates and therefore, it seems possible to conclude that the analysis of the distribution of sample types can be a useful tool to explain the performance of editing methods.

5 Concluding Remarks

This paper has shown that the performance of instance selection methods can be understood by analyzing the underlying structure of data sets. To this end, one can use local information to categorize the samples into different groups (safe, borderline, rare, and outlier) and compare their distribution in the original training set with that in the edited set.

The experiments have revealed that the algorithms with the highest increase of safe samples and the highest decrease of unsafe samples correspond to those with the highest improvement in 1NN accuracy. Although this work can be further extended by incorporating other techniques and some real-life databases, we believe that these initial observations could be utilized to provide a qualitative discussion of the experimental results in papers where several procedures have to be compared each other.

Acknowledgment. This work has partially been supported by the Universitat Jaume I under grant UJI-B2018-49.

References

1. Alcalá-Fdez, J., et al.: KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J. Mult.-Valued Log. S. **17**, 255–287 (2011)
2. Barandela, R., Ferri, F.J., Sánchez, J.S.: Decision boundary preserving prototype selection for nearest neighbor classification. Int. J. Pattern Recogn. **19**(6), 787–806 (2005)
3. Blayo, E., et al.: Deliverable R3-B4-E Task B4: benchmarks, ESPRIT 6891. In: ELENA: Enhanced Learning for Evolutive Neural Architecture (1995)
4. Brodley, C.E.: Adressing the selective superiority problem: automatic algorithm/model class selection. In: Proceedings of the 10th International Machine Learning Conference, Amherst, MA, pp. 17–24 (1993)
5. Caises, Y., González, A., Leyva, E., Pérez, R.: Combining instance selection methods based on data characterization: an approach to increase their effectiveness. Inf. Sci. **181**(20), 4780–4798 (2011)
6. Dasarathy, B.V.: Nearest Neighbor (NN) Norms: Nn Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
7. Devijver, P.A.: On the editing rate of the MULTIEDIT algorithm. Pattern Recogn. Lett. **4**(1), 9–12 (1986)

8. García, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans. Pattern Anal. **34**(3), 417–435 (2012)
9. Hattori, K., Takahashi, M.: A new edited *k*-nearest neighbor rule in the pattern classification problem. Pattern Recogn. **33**, 521–528 (2000)
10. Krawczyk, B., Woźniak, M., Herrera, F.: Weighted one-class classification for different types of minority class examples in imbalanced data. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, Piscataway, NJ, pp. 337–344 (2014)
11. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the 14th International Conference on Machine Learning, Nashville, TN, pp. 179–186 (1997)
12. Leyva, E., González, A., Pérez, R.: A set of complexity measures designed for applying meta-learning to instance selection. IEEE Trans. Knowl. Data Eng. **27**(2), 354–367 (2015)
13. Lowe, D.G.: Similarity metric learning for a variable-kernel classifier. Neural Comput. **7**(1), 72–85 (1995)
14. Mollineda, R.A., Sánchez, J.S., Sotoca, J.M.: Data characterization for effective prototype selection. In: Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis, Estoril, Portugal, pp. 27–34 (2005)
15. Napierala, K., Stefanowski, J.: Types of minority class examples and their influence on learning classifiers from imbalanced data. J. Intell. Inf. Syst. **46**(3), 563–597 (2016)
16. Riquelme, J.C., Aguilar-Ruiz, J.S., Toro, M.: Finding representative patterns with ordered projections. Pattern Recogn. **36**, 1009–1018 (2003)
17. Sáez, J.A., Krawczyk, B., Woźniak, M.: Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. Pattern Recogn. **57**, 164–178 (2016)
18. Sánchez, J.S., Pla, F., Ferri, F.J.: Prototype selection for the nearest neighbor rule through proximity graphs. Pattern Recogn. Lett. **18**, 507–513 (1997)
19. Sánchez, J.S., Pla, F., Ferri, F.J.: Improving the *k*-NCN classification rule through heuristic modifications. Pattern Recogn. Lett. **19**(13), 1165–1170 (1998)
20. Stefanowski, J., Wilk, S.: Selective pre-processing of imbalanced data for improving classification performance. In: Proceedings of the 10th International Conference in Data Warehousing and Knowledge Discovery, Turin, Italy, pp. 283–292 (2008)
21. Tomek, I.: An experiment with the edited nearest-neighbor rule. IEEE Trans. Syst. Man Cybern. **6**(6), 448–452 (1976)
22. Vázquez, F., Sánchez, J.S., Pla, F.: A stochastic approach to Wilson's editing algorithm. In: Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis, Estoril, Portugal, pp. 35–42 (2005)
23. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE T. Syst. Man Cybern. **2**(3), 408–421 (1972)