Improving the Behavior of the Nearest Neighbor Classifier against Noisy Data with Feature Weighting Schemes

José A. Sáez[1], Joaquín Derrac[2], Julián Luengo[3], and Francisco Herrera[1]

[1] Department of Computer Science and Artificial Intelligence, University of Granada, CITIC-UGR, Granada, Spain, 18071
{smja,herrera}@decsai.ugr.es
[2] School of Computer Science & Informatics, Cardiff University, Cardiff CF24 3AA, United Kingdom
jderrac@decsai.ugr.es
[3] Department of Civil Engineering, LSI, University of Burgos, Burgos, Spain, 09006
jluengo@ubu.es

Abstract. The Nearest Neighbor rule is one of the most successful classifiers in machine learning but it is very sensitive to noisy data, which may cause its performance to deteriorate. This contribution proposes a new feature weighting classifier that tries to reduce the influence of noisy features. The computation of the weights is based on combining imputation methods and non-parametrical statistical tests. The results obtained show that our proposal can improve the performance of the Nearest Neighbor classifier dealing with different types of noisy data.

Keywords: noisy data, feature weighting, classification.

1 Introduction

The Nearest Neighbor (NN) classifier [4] uses the full training dataset to establish a classification rule, based on the most similar or nearest training instance to the query example. The most frequently used similarity function for the NN classifier is Euclidean distance [1] (see Equation 1, where X and Y are two instances and M is the number of features that describes them).

$$d(X,Y) = \sqrt{\sum_{i=0}^{M}(x_i - y_i)^2} \tag{1}$$

However, features containing enough noise may lead to erroneous similarities between the examples obtained and, therefore, to a deterioration in the performance of NN, which is known to be very sensitive to noisy data [10]. One way of overcoming this problem lies in modifying the similarity function, that is, the way in which the distances are computed. With this objective, Feature

M. Polycarpou et al. (Eds.): HAIS 2014, LNAI 8480, pp. 597–606, 2014.

Weighting methods [12], [9] try to improve the similarity function, by introducing a weight for each of the features (W_i, usually $W_i \in [0, 1]$). These methods, which are mostly based in the Euclidean distance, modify the way in which the distance measure is computed (Equation 2), increasing the relevance of those features with greater weights associated with them (near to 1.0).

$$d_w(X, Y) = \sqrt{\sum_{i=0}^{M} W_i \cdot (x_i - y_i)^2} \tag{2}$$

These weights W_i can be regarded as a measure of how useful a feature is with respect to the final classification task. The higher a weight is, the more influence the associated feature will have in the decision rule used to compute the classification of a given example. Therefore, an adequate scheme of weights could be used to diminish the worst features of the domain of the problem, which could be those containing the more harmful amount of noise to the classification task. Thus, the accuracy of the classifier could be greatly improved if a proper selection of weights is made.

This contribution proposes a novel approach for weighting features, based on the usage of imputation methods [3], [6], [5]. These are commonly employed to estimate those feature values in a dataset that are unknown, formally known as missing values (MV), using the rest of the data available. Therefore, imputation methods enable us to estimate a new distribution of the original dataset, in which the distribution of each feature is conditioned to the rest of the features or all the data. These conditioned distributions of each feature can be compared with the original ones in order to detect the relevance of each feature, depending on the accuracy of the estimation for that feature performed by the imputation method.

The Kolmogorov-Smirnov statistic [11] may then be used to evaluate the differences between the original distribution of the features and that of the imputed ones. It is thus possible to measure how well the values of each feature can be predicted using the rest of the data. This enables us to give less importance to those features with high changes between their original and estimated value distributions - these features that contain too much noise or the more harmful noise and therefore are not easily predictable using the rest of the data, which increases the effect of those features that are easily predictable, and which have therefore likely a less amount of noise.

The study is completed with an experimentation in which our proposal is compared with the NN classifier, considering 25 supervised classification problems taken from the Keel-Dataset repository [2], into which we will introduce different types and levels of noise.

The rest of this contribution is organized as follows. In Section 2 we describe our proposal. In Section 3 we present the experimental framework, and in Section 4 we analyze the results obtained. Finally, in Section 5 we enumerate some concluding remarks.

2 A Weighting Scheme to Reduce the Effect of Noisy Data

This section describes the weighting method proposed, which is based on three main steps, described in the following subsections. Section 2.1 is devoted to the first step (called *the imputation phase*), whereas Section 2.2 describes the second step (*the computation of the weights*). Finally, Section 2.3 characterizes the third step (*the classification model*).

2.1 Imputation of the Dataset

The first step consists of creating a whole new estimated dataset DS' from the original one DS. In order to do this, an imputation method is used. In this contribution we will consider the following imputation methods (although other imputation methods may be chosen):

1. **KNNI** [3]. Based on the k-NN algorithm, every time an MV is found in a current example, KNNI computes the k ($k = 10$ in our experimentation) nearest neighbors and their average value is imputed. KNNI also uses the Euclidean distance as a similarity function.
2. **CMC** [6]. This method replaces the MVs by the average of all the values of the corresponding feature considering only the examples with the same class as the example to be imputed.
3. **SVMI** [5]. This is an SVM regression-based algorithm developed to fill in MVs. It works by firstly selecting the examples in which there are no missing feature values. In the next step, the method sets one of the input features, some of the values of which are missing, as the decision feature, and the decision feature as the input feature. Finally, an SVM for regression is used to predict the new decision feature.

If the original dataset DS is composed of the features $f_1, f_2, \ldots, f_M$, the imputed dataset DS' will be formed by the features $f'_1, f'_2, \ldots, f'_M$ whose values are obtained by the imputation method.

The procedure to obtain DS' from DS is based on assuming iteratively that each feature value of each example of the dataset DS, that is, $e(f_i)$, is missing. Then, the imputation method IM is used to predict a new value for that feature value. The new dataset DS' is obtained by repeating this process for each feature value, until the whole dataset has been processed. Carrying out this process, it is possible to estimate a distribution of values for each feature, which is conditioned to the rest of the features or the totality of the data. The new dataset DS' will contain these conditioned distributions for each feature.

2.2 Computation of Weights Using the Kolmogorov-Smirnov Test

The next step consists of measuring which features are most changed after the application of the imputation method. Given the nature of the imputation techniques, some features are expected to remain unchanged (or to present only

small changes in their values' distribution) whereas other features may present a higher level of disruption when their imputed values are compared with the original ones. Thus, those features that are more difficult to predict with the rest of the features/data will contain the more harmful noise and therefore we will try to make them less important to the classification task. The Kolmogorov-Smirnov test [11] provides a way of measuring these changes. This test works by computing a statistic D_n, which can be regarded as a measure of how different two samples are.

The test is a nonparametric procedure for testing the equality of two continuous, one-dimensional probability distributions. It quantifies a distance between the empirical distribution functions of two samples. The null distribution of its statistic, D_n, is computed under the null hypothesis that the samples are drawn from the same distribution.

Given two samples, X and Y, and their empirical distribution functions F_X and F_Y

$$F_X(x) = \frac{1}{n}\sum_{i=1}^{n} I_{X_i \leq x}, \qquad F_Y(x) = \frac{1}{n}\sum_{i=1}^{n} I_{Y_i \leq x} \tag{3}$$

(where $I_{X_i \leq x}$ is the indicator function, equal to 1 if $X_i \leq x$ and equal to 0 otherwise) the Kolmogorov-Smirnov statistic is

$$D_n = \sup_x |F_X - F_Y| \tag{4}$$

In the approach of this contribution, the D_n statistic provides a valuable way of estimating the degree of change undergone by a feature through the imputation process. By computing the D_n statistic associated with the differences between both samples of the feature (original and imputed), it is possible to measure the greater degree of difference between the expected distribution of both samples. Hence, the greater D_n value obtained, the more different the imputed version of the feature distribution will be (when compared with the original one).

The D_n statistic can be easily transformed into a weight. Since $D_n \in [0, 1]$, features with a lower value of D_n (near to 0.0) it will have little influence on the computation of the similarity function of the NN rule, whereas features with a higher value of D_n (near to 1.0) will be the most influential when computing the distance between two examples. Defining the statistical D_n^i for the feature i as

$$D_n^i = \text{Kolmogorov-Smirnov}(e_{f_i}, e_{f_i'}) \tag{5}$$

(where e_{f_i} and $e_{f_i'}$ are the empirical distributions of the features $f_i \in \mathcal{A}$ and $f_i' \in \mathcal{A}'$ respectively, and $\mathcal{A}$ denotes the set of features of the original dataset DS and $\mathcal{A}'$ denotes the set of features imputed in DS'), then the weights $W_i \in [0, 1]$ computed for a feature $f_i \in \mathcal{A}$ are

$$W_i = (1 - D_n^i)/(\sum_{j=1}^{M} 1 - D_n^j) \tag{6}$$

Therefore, the Kolmogorov-Smirnov test is applied to measure the degree of difference between each attribute f_i and its estimated version f_i'; then, this difference is used to build the weight for the attribute f_i (see Equation 6).

2.3 Final Classification Model

The final classifier considers NN with the weighted Euclidean distance (Equation 2) and the weights computed throughout the Kolmogorov-Smirnov statistic (Equation 6). Since we will consider three different imputation methods (KNNI, CMC and SVMI), three different feature weighting classifiers will be created. Throughout the study, we will denote them as FW-KNNI, FW-CMC and FW-SVMI according to the imputation method used.

Considering weights computed from the D_n statistic, we aim to reduce the effect that changing features have on the computation of the distance. These features, with a larger associated D_n value, will be those easily estimated by the imputation method (whose sample distribution differs poorly if the original and imputed versions are compared). They are preferred since they will contain a less harmful noise, and are the key features describing the dataset.

By contrast, features with a small D_n value will be those whose sample distribution has been greatly changed after the application of the imputation method. Since these features are not easily estimated when the rest of the data is available (the imputation method cannot recover their values properly), they are not preferred in the final computation of the distance, and thus a lower weight is assigned to them.

3 Experimental Framework

Section 3.1 describes the base datasets employed and Section 3.2 shows the noise introduction processes. Finally, Section 3.3 describes the methodology followed to analyze the results.

3.1 Base Datasets

The experimentation considers 25 real-world datasets from the KEEL-Dataset repository [2]. They are described in Table 3, where #EXA refers to the number of examples, #FEA to the number of numeric features and #CLA to the number of classes. For datasets containing missing values (such as *bands* or *dermatology*), the examples with missing values were removed from the datasets before their usage and thus all the attribute values of the datasets considered are known. In this way, the percentage of missing values of each dataset does not influence the results or conclusions obtained. Therefore, the only missing values considered in this contribution are those assumed during the execution of our proposal.

Table 1. Datasets employed in the experimentation

dataset	#EXA	#FEA	#CLA	dataset	#EXA	#FEA	#CLA
banana	5300	2	2	pima	768	8	2
bands	365	19	2	satimage	6435	36	7
bupa	345	6	2	sonar	208	60	2
dermatology	358	34	6	tae	151	5	3
ecoli	336	7	8	texture	5500	40	11
heart	270	13	2	vowel	990	13	11
hepatitis	80	19	2	wdbc	569	30	2
ionosphere	351	33	2	wine	178	13	3
iris	150	4	3	wq-red	1599	11	11
led7digit	500	7	10	wq-white	4898	11	11
mov-libras	360	90	15	wisconsin	683	9	2
newthyroid	215	5	3	yeast	1484	8	10
phoneme	5404	5	2				

3.2 Introducing Noise into Datasets

In order to control the amount of noise in each dataset and to check how it affects the classifiers, noise is introduced into each dataset in a supervised manner. Two different noise schemes, which are proposed in the specialized literature [14], are used in order to introduce a noise level of $x\%$ into each dataset.

- **Random Class Noise**. It supposes that exactly $x\%$ of the examples are corrupted. The class labels of these examples are randomly changed by other one out of the M classes.
- **Random Attribute Noise**. $x\%$ of the values of each attribute in the dataset are corrupted. To corrupt an attribute A_i, approximately $x\%$ of the examples in the dataset are chosen, and their A_i value is assigned a random value from $\mathbb{D}_i$. A uniform distribution is used either for numerical or nominal attributes.

A collection of new noisy datasets are created from the aforementioned 25 base real-world datasets. Both types of noise are independently considered: class and attribute noise. For each type of noise, the noise levels $x = 10\%$ and $x = 30\%$ are studied. Thus, the results of our proposal will be compared with those of NN considering three different scenarios: with the 25 unaltered real-world datasets, with the 25 datasets with a 10% of noise level and with the 25 datasets with a 30% of noise level.

3.3 Methodology of Analysis

The performance estimation of each classifier on each dataset is obtained by means of 3 runs of a 10-fold *distribution optimally balanced stratified cross-validation* (DOB-SCV) [7], averaging its test accuracy results. The usage of this

partitioning reduces the negative effects of both prior probability and covariate shifts [8] when classifier performance is estimated with cross-validation schemes.

For the sake of brevity, only the averaged performance results are shown for each classification algorithms at each type and level of induced noise, but it must be taken into account that our conclusions are based on the proper statistical analysis, which considers all the results (not averaged). Thus, in order to properly analyze the results obtained, Wilcoxons signed rank statistical test [13] is used, as suggested in the literature. This is a non-parametric pairwise test that aims to detect significant differences between two sample means; that is, between the behavior of the two algorithms involved in each comparison (which is usually viewed as the the averaged test performance results for each dataset). For each type and noise level, our proposal and NN using the Euclidean distance will be compared using Wilcoxons test and the p-values associated with these comparisons will be obtained. The p-value represents the lowest level of significance of a hypothesis that results in a rejection and it allows one to know whether two algorithms are significantly different and the degree of this difference.

4 Analysis of Results

This section presents the analysis of the results obtained. Each table of results is divided into two different parts. On the left hand of the table the average accuracy results are found, whereas on the right hand of the table the associated Wilconson's test p-values resulting of the comparison of each one of our proposals with the NN method are shown.

Table 2 shows the test accuracy obtained by each classifier on base and class noise datasets.

Table 2. Results on base and class noise datasets and associated p-values

	Accuracy			p-values		
Method	Base	$x = 10\%$	$x = 30\%$	Base	$x = 10\%$	$x = 30\%$
NN	79.37	74.96	65.46	-	-	-
FW-CMC	81.98	77.38	67.46	0.1107	0.1107	0.0787
FW-KNNI	81.97	77.36	67.41	0.1447	0.0827	0.2699
FW-SVMI	81.94	77.30	67.19	0.0626	0.0647	0.4352

From this table, several remarks can be made:

– The performance results of each one of our proposals is better than those of the NN method with the base datasets and also with the class noise datasets (approximately, higher than a 2% in all the cases).

– As the table shows, every proposal obtains low p-values when they are compared with NN: with the base datasets and both levels of class noise in the case of FW-CMC and with the base datasets and the noise level $x = 10\%$ in the case of the methods FW-KNNI and FW-SVMI. Some of these comparisons are also significant at a level of significance 0.1. This shows that the application of our approach to feature weighting improves the performance of the NN classifier with datasets suffering from class noise (sometimes significantly), regardless of the specific imputation method chosen.

On the other hand, Table 3 shows the test accuracy obtained by each classifier on base and attribute noise datasets. The following points are observed from this table:

– Our methods also outperforms the performance of NN with the datasets with different levels of attribute noise (generally they are a 2% better with the base datasets, a 1% better with the noise level $x = 10\%$ and a 0.5% with the noise level $x = 30\%$).
– The Wilcoxon's test p-values are also low, showing and advantage of our three proposals, even though in the case of FW-SVMI against NN with the noise level of $x = 30\%$ the p-value obtained is slightly higher. However, very low p-values are obtained with the two noise levels for the methods FW-CMC and FW-KNNI; they are indeed significant considering a significance level of 0.1.

Table 3. Results on base and attribute noise datasets and associated p-values

	Accuracy			p-values		
Method	Base	$x = 10\%$	$x = 30\%$	Base	$x = 10\%$	$x = 30\%$
NN	79.37	71.69	58.40	-	-	-
FW-CMC	81.98	72.62	59.17	0.1107	0.0067	0.0002
FW-KNNI	81.97	72.58	58.87	0.1447	0.0483	0.0246
FW-SVMI	81.94	72.44	58.61	0.0626	0.1318	0.2414

From the results of Tables 2-3, it is possible to conclude that the proposals presented in this contribution are able to improve the performance of the NN classifier dealing with noisy data, and in some cases, in a significant way.

5 Conclusions

In this contribution we have proposed a new scheme for feature weighting developed to improve the performance of the NN classifier in presence of noisy data, in which the weights are computed by combining imputation methods and the

Kolmogorov-Smirnov statistic. We have assigned a lower weight to that features that were more affected by the presence of noise (those features whose original and imputed distribution of values were more different). In this way, we have reduced the importance of these features that contain the more harmful noise and therefore are not easily predictable using the rest of the data and increased the importance of of those features that are easily predictable, and which have therefore likely a less amount of noise.

The results obtained show that all our approaches enhance the performance of NN in the presence of noise. The statistical analysis performed confirms our conclusions, even though in some cases the differences found are not statistically significant.

Acknowledgment. Supported by the Projects TIN2011-28488, P10-TIC-06858 and P11-TIC-9704. J. A. Sáez holds an FPU grant from the Spanish Ministry of Education and Science.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6, 37–66 (1991)
2. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing 17(2-3) (2011)
3. Batista, G.E.A.P.A., Monard, M.C.: An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence 17(5-6), 519–533 (2003)
4. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13, 21–27 (1967)
5. Khosla, R., Howlett, R.J., Jain, L.C. (eds.): KES 2005. LNCS (LNAI), vol. 3683. Springer, Heidelberg (2005)
6. Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.): RSFDGrC 2005. LNCS (LNAI), vol. 3642. Springer, Heidelberg (2005)
7. Moreno-Torres, J.G., Sáez, J.A., Herrera, F.: Study on the Impact of Partition-Induced Dataset Shift on k-fold Cross-Validation. IEEE Transactions on Neural Networks and Learning Systems 23(8), 1304–1312 (2012)
8. Moreno-Torres, J.G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition 45(1), 521–530 (2012)
9. Paredes, R., Vidal, E.: Learning weighted metrics to minimize nearest-neighbor classification error. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(7), 1100–1110 (2006)
10. Sáez, J., Luengo, J., Herrera, F.: Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. Pattern Recognition 46(1), 355–364 (2013)

11. Smirnov, N.V.: Estimate of deviation between empirical distribution functions in two independent samples. Bulletin of Moscow University 2, 3–16 (1939) (in Russian)
12. Wettschereck, D., Aha, D.W., Mohri, T.: A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. Artificial Intelligence Review 11, 273–314 (1997)
13. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin 1(6), 80–83 (1945)
14. Zhu, X., Wu, X.: Class Noise vs. Attribute Noise: A Quantitative Study. Artificial Intelligence Review 22, 177–210 (2004)