An Experimental Study on Rank Methods for Prototype Selection

Jose J. Valero-Mas · Jorge Calvo-Zaragoza · Juan R. Rico-Juan · José M. Iñesta

Abstract Prototype Selection is one of the most popular approaches for addressing the low efficiency issue typically found in the well-known k-Nearest Neighbour classification rule. These techniques select a representative subset from an original collection of prototypes with the premise of maintaining the same classification accuracy. Most recently, rank methods have been proposed as an alternative to develop new selection strategies. Following a certain heuristic, these methods sort the elements of the initial collection according to their relevance and then select the best possible subset by means of a parameter representing the amount of data to maintain. Due to the relative novelty of these methods, their performance and competitiveness against other strategies is still unclear. This work performs an exhaustive experimental study of such methods for prototype selection. A representative collection of both classic and sophisticated algorithms are compared to the aforementioned techniques in a number of datasets, including different levels of induced noise. Results report the remarkable competitiveness of these rank methods as well as their excellent trade-off between prototype reduction and achieved accuracy.

Keywords k-Nearest Neighbour · Data reduction · Prototype Selection · Rank methods

Jose J. Valero-Mas · Jorge Calvo-Zaragoza · Juan R. Rico-Juan · José M. Iñesta
Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain
Tel.: +349-65-903772
Fax: +349-65-909326
E-mail: {jjvalero,jcalvo,juanra,inesta}@dlsi.ua.es

1 Introduction

The k-Nearest Neighbour (kNN) rule is one of the most common algorithms for supervised non-parametric classification [10], where statistical knowledge of the conditional density functions is not available. This rule hypothesises about a given input considering the most common label among its k-nearest prototypes of the training set. Its simplicity, straightforward implementation and an error bounded by twice the Bayes error [6] are important qualities that characterise this classifier. Nevertheless, one of the main problems of this technique is its low efficiency in both running time and memory usage, since it needs to store and query every single prototype of the training set in order to compute the distances required.

Data Reduction (DR) techniques, a special case of Data Preprocessing, are widely used in kNN classification as a means of overcoming the previously commented drawbacks. They aim at reducing the size of the training set while keeping the same classification accuracy as with the original data [14]. DR can be further divided into two common approaches [19]: Prototype Generation (PG) and Prototype Selection (PS). The former method creates new artificial data to replace the initial set while the latter one simply selects certain elements from that set. The work presented here focuses on PS techniques, which are less restrictive than PG as they do not require information about type of representation used for encoding the data [4].

Given the importance of PS methods, numerous approaches have been proposed throughout the years [13], typically divided into three main families: condensing strategies, which try to keep only the most relevant prototypes; editing strategies, which pursue a removal of the prototypes located in dubious zones; and hybrid methods, which look for a compromise between the two previous approaches.

Moreover, a new approach has been recently developed, in which the key question is not to select prototypes but to sort them by their importance for the classification accuracy. This new approach, referred as *rank methods*, is finally guided by a tuning parameter which specifies the amount of data to keep.

Due to the novelty of the aforementioned rank-based family, it is hitherto unclear its competitiveness against state-of-art methodologies. To plug that gap, the present work aims at providing a comprehensive experimental study on the performance of rank methods for PS. A representative series of algorithms is selected from the literature and compared against rank methods in a number of scenarios, which differ in their size and amount of mislabelled samples (to simulate a noisy condition), and taking into account different metrics of interest.

The rest of the paper is structured as follows: Section 2 introduces some of the most common approaches for Prototype Selection in kNN. Section 3 describes the rank methods to be assessed. Section 4 describes the experimental set-up as well as the evaluation methodology proposed. Results obtained are presented and analysed in Section 5. Finally, Section 6 outlines the general conclusions obtained as well as possible future work.

2 Background

The *Condensed Nearest Neighbour* [17] was one of the first techniques aimed at reducing the size of the training set for kNN classification. This method focuses on keeping those prototypes close to boundaries and removing the rest. The reduction starts with an empty set S, and every prototype of the initial training set is queried randomly. If the prototype is misclassified using the 1-NN rule and the set S, then the prototype is included in S. Otherwise, the prototype is discarded. At the end, the set S is returned as a representative reduced version of the initial training set. Note that the main assumption behind the method is that if a prototype is misclassified with a 1NN, it is probably close to the boundaries. Therefore, it should be maintained. Extensions to this technique include: *Reduced Nearest Neighbour* [16], which performs the condensing algorithm and then revisits each maintained prototype

to assure whether it is actually necessary for the classification; *Selective Nearest Neighbour* [23], which assures that the nearest neighbour of each prototype of the initial training set is in the condensed subset; and *Fast Condensing Nearest Neighbour* [1], which provides a fast, order-independent variant of the algorithm.

Following an inverse approach, the *Editing Nearest Neighbour* [28] was the first proposal to reduce the training set by removing outliers and noisy instances. It starts with a set S equal to the initial training set. The process applies the 1NN rule to each single prototype in S. If the element is misclassified (class predicted by kNN does not match with the prototype one), the element is removed from S. Otherwise, it is maintained. Common extensions to this technique are the *Repeated-Editing Nearest Neighbour* [25], which repeatedly applies editing until homogeneity is reached, and the *Multi-Editing Nearest Neighbour* [9], which repeatedly performs editing over distributed blocks of the training set.

Several algorithms have appeared as a combination of the two aforementioned general ideas, referred as hybrid approaches. Some good representatives of hybrid methods are: the *Multi-Editing Condensing Nearest Neighbour* [7], a combination of *Multi-Editing* and *Condensing* strategies; the *Decremental Reduction Optimization Procedure* [28], in which instances are ordered according to the distance to their nearest neighbours and then, starting from the furthest ones, those which do not affect the generalization accuracy are removed; and the *Iterative Case Filtering* [2], which bases its performance on the coverage and reachability premises to select the instances subset able to maximize the classification accuracy. In addition, Evolutionary Algorithms (EA) have also been adapted to perform PS [5]. For instance, the *Cross-generational elitist selection, Heterogeneous recombination and Cataclysmic mutation* search [11], whose name indicates the behaviour of its genetic operators, is considered one of the most successful applications of EA for this task.

Unfortunately, PS methods often carry an accuracy loss with respect to directly using the original training set. This is why PS has been hybridised with other paradigms such as Ensemble methods [15] or Feature Selection [8,26]. Nevertheless, we shall restrict ourselves to consider conventional PS for our experimental study.

3 Rank methods for Prototype Selection

This section introduces the gist of rank methods for PS as well as those strategies already proposed under this paradigm.

The main idea behind rank methods is that prototypes of the training set are not selected but ordered. Following some heuristics, prototypes are given a score that indicates its relevance with respect to classification accuracy. Eventually, prototypes are selected starting from the highest score until a certain criterion is accomplished.

A particular approach for rank methods is to follow a voting heuristic, *ie.* prototypes vote for the rest of the prototypes that help them to be correctly classified. After the voting process, the received score is normalised to produce *importance rate* so that the sum over these rates for all the prototypes of a given class is equal to 1. Then, the training set is sorted according to those values and the best candidates are selected until their accumulated score exceeds an external parameter $\alpha \in (0, 1]$ that allows the performance of the rank method to be tuned. Low values of this parameter will lead to a higher reduction of the size of the training set, while high values will remove only the most irrelevant prototypes. Although tuning parameters may be considered an inconvenient, in this case this is specially interesting because the parameter allows the user to enhance a particular objective (either reduction or accuracy) depending on the requirements of the system.

The experimental study carried out in this paper focuses on the two voting heuristics proposed so far, to our best knowledge: Farthest Neighbour (FN) and Nearest to Enemy (NE). Both strategies are based on the idea that a prototype can give one vote to another one, and the question is to decide which prototype it is given to. Although these strategies were previously published [22], we shall revisit below their main ideas for a better readability of the current paper.

For the sake of clarity, some notation is presented. We will use $d(\cdot, \cdot)$ to denote the distance between prototypes used for the kNN rule. Let $\zeta(p)$ denote the class label of prototype p. Let us call *friends of p* (f_p) the set of prototypes that share class label with p, *ie.* $f_p = \{p' : \zeta(p') = \zeta(p)\}$, and *enemies of p* ($e_p$) the rest of prototypes. As both strategies loop over each prototype of the training set, we will use a to denote the prototype issuing the vote. Then, we will use b as the nearest enemy of a: $\arg\min_{p \in e_p} d(a, p)$.

3.1 Farthest Neighbour voting

The *one vote to the Farthest Neighbour* (FN) strategy searches for a prototype c, which is the farthest friend of a but still closer than b. That is, a will give its vote to prototype c such that

$$c = \arg\max_p d(a, p) : p \in f_a \wedge d(a, p) < d(a, b)$$

The idea is to vote for a prototype that contributes to classify a correctly using the kNN rule as well as to reduce the density of prototypes over a definite area.

3.2 Nearest to Enemy voting

The *one vote to the Nearest to Enemy* (NE) strategy makes a vote for the *friend* that is the closest to b. This friend must also be within the area centred at a and radius $d(a, b)$. Formally, a will give its vote to prototype c such that

$$c = \arg\min_p d(p, b) : p \in f_a \wedge d(a, p) < d(a, b)$$

The idea is to try to avoid any misclassification produced by c using kNN rule in an area with prototypes of other classes.

The previously described strategies can be extended by letting b be the n-nearest enemy in order to reduce the influence of possible outliers. We will denote these strategies by n-FN and n-NE. For example, configuration $n = 2$ will use the second nearest enemy to voting prototype a, which has proved to be useful in practice [22].

4 Experimental setup

4.1 Datasets

Due to the experimental nature of the paper and in order to consistently assess the introduced strategies, a considerable number of datasets were considered. The performance of PS algorithms is highly related to the size of the dataset. That is why the use of small datasets[1] is common in works concerning DR. However, the use of such sets does not really make sense from our point of view: PS aims at speeding up kNN classification by reducing the information to be processed. In that sense, datasets not containing a large amount of prototypes can be processed relatively fast and, therefore, reduction would be useless in this scenario.

Our experiments were carried out with five corpora: the *NIST SPECIAL DATABASE 3* (NIST3) of the National Institute of Standards and Technology, from which a subset of the upper case characters was randomly selected; the *United States Postal Office* (USPS)

---

[1] Given that this number of elements is highly dependent on the memory and computation capabilities of the system considered, we shall restrict ourselves to the definition by Garcia et al. [13] in which this threshold is set to 2,000 prototypes.

handwritten digit dataset [18]; the *Handwritten Online Musical Symbol* (HOMUS) dataset [3]; and two additional corpora of the UCI (*Penbased* and *Letter*). For the two first cases, contour descriptions with Freeman Chain Codes (FCC) [12] were extracted and the Edit Distance (ED) [27] was used as dissimilarity measure. In the third case, and due to its good results in the baseline experimentation offered with these data, Dynamic Time Warping (DTW) [24] is used. Since datasets from the UCI may contain missing values in the samples, the Heterogeneous Value Difference Metric (HVDM) [29] is used for the last two datasets. Table 1 shows a summary of the main features of these datasets.

In spite of not being a common procedure, we will add synthetic noise to assess the robustness of the considered PS methods in this type of scenarios. The noise will be induced by swapping labels between pairs of prototypes randomly chosen. The noise rates (percentage of prototypes that change their label) considered were 0 %, 20 %, and 40 % since these are common values in this kind of experimentation [20].

4.2 Prototype Selection strategies

The main goal of the current work is to provide a comprehensive comparative experiment to evaluate the performance and competitiveness of rank methods as PS strategies. To cover the different families of approaches presented in Section 2, we shall consider the following particular strategies:

- **No selection**: all the prototypes of the initial training set (ALL).
- **Classical**: Condensing Nearest Neighbour (CNN), Editing Nearest Neighbour (ENN), Fast Condensing Nearest Neighbour (FCNN), Editing Condensing Nearest Neighbour (ECNN) and Editing Fast Condensing Nearest Neighbour (EFCNN).
- **Hybrid**: Iterative Case Filtering (ICF), Decremental Reduction Optimization Procedure (DROP).
- **Evolutionary**: Cross-generational elitist selection, Heterogeneous recombination and Cataclysmic mutation (CHC).
- **Rank**: 1-FN, 2-FN, 1-NE, 2-NE; each of them considering values of α within the range $(0, 1)$ with a granularity of 0.1. The extreme values have been discarded since $\alpha = 0$ would mean an empty set and $\alpha = 1$ is equivalent to ALL.

As mentioned above, a desirable feature of PS methods is the robustness against noise. In this sense, after the PS process, a low k value in the kNN rule should be enough, as hardly any noise would be expected to remain in the reduced set [21]. Thus, classification experiments with the kNN rule will only consider $k = 1$ for the set obtained when applying PS and $k = 1, 3, 5, 7$ for the ALL case in order to give some hint about the loss caused by induced noise during classification. Higher values of k for PS may lead to a misunderstanding in both results and discussion, since it would not be clear whether the noise is being handled by the PS method or by the kNN.

4.3 Performance measurement

In order to analyse the performance of PS strategies, we have taken into account the accuracy of the classification and the size of the selected set. While the former indicates the ability of the method to choose the most relevant prototypes, the latter one depicts its reduction skills.

Although these measures allow us to analyse the performance of each considered strategy, it is not possible to establish a comparison among the whole set of alternatives to determine which is the best one. The problem is that PS algorithms try to minimise the number of prototypes considered in the training set and, at the same time, they try to increase classification accuracy. Most often, these two goals are contradictory so improving one of them implies a deterioration of the other. From this point of view, PS-based classification can be seen as a Multi-objective Optimization Problem (MOP) in which two functions are meant to be optimised at the same time: minimisation of prototypes in the training set and maximisation of the classification success rate. The usual way of evaluating this kind of problems is by means of the *non-dominance* concept. One solution is said to dominate another if, and only if, it is better or equal in each goal function and, at least, strictly better in one of them. The set of non-dominated elements represents the different optimal solutions to the MOP. Each of them is usually referred to as Pareto optimal solution, being the whole set usually known as Pareto frontier.

Thus, the considered strategies will be compared by assuming a MOP scenario in which each of them is a 2-dimensional solution defined as $(acc, size)$ where acc is the accuracy obtained by the strategy and $size$ is the rate (%) of selected prototypes with respect to the original set. To analyse the results, the pair obtained by each scheme will be plotted in $2D$ point graphs where the non-dominated set of pairs will be enhanced. In the MOP framework, the strategies within this set can be considered the best without defining any order between them.

Name	Instances	Classes	Dissimilarity
USPS	9298	10	ED
NIST3	6500	26	ED
HOMUS	15200	32	DTW
Penbased	10992	10	HVDM
Letter	20000	26	HVDM

Table 1 Description of the datasets used in the experimentation.

Unfortunately, pursuing these two objectives at the same time prevents the performance of statistical methods to measure the actual significance about different results achieved. This may be done if one criterion was given more importance than the other. However, since that particular analysis depends on the requirements of each underlying classification task, we shall not consider that case. In addition, normalised accuracy and reduction rates could be combined into a single figure but it would not be a good indicator for comparing different approaches from the point of view of PS. For all above, we will perform a 4-fold cross validation over each dataset considered and our analysis will focus on overall average results.

5 Results

This section analyses the results obtained. These results can be checked in Table 2, which depicts the arithmetic mean of the accuracy and size figures obtained for the considered datasets. Bold values represent the non-dominated solutions, which can be graphically seen in Figs. 1 and 2 for the different induced noise cases considered.

Additionally, the appendix included in this paper breaks down the accuracy and size figures for each dataset and noise configuration. Nevertheless, the analysis is performed on the aforementioned global average figures.

Let us pay attention first to the case when no induced noise is considered. It can be observed that, when no information was discarded (ALL scheme), conventional kNN achieved some of the highest accuracy values for all k configurations. Note that increasing this k parameter did not have any noticeable effect. Given that the datasets considered have very little noise, ENN algorithm did not significantly reduce the size of the set (a reduction rate around 10 %), maintaining similar accuracies to those achieved by the conventional kNN strategies.

On the other side, the condensing family of algorithms (CNN and its extensions) showed some remarkable results: all of them achieved great reduction rates, especially ECNN and EFCNN, which simply required around a 10 % of the set size, and performed well in terms of accuracy (only around 3 % lower than the ALL configurations).

DROP3 also achieved high reduction rates (around 9 % of the maximum), but with a significant drop in accuracy when compared to the conventional kNN algorithm (decreased around 10 % with respect to the scores in the ALL cases). ICF, however, achieved neither a high reduction nor a remarkable accuracy.

The CHC evolutionary algorithm obtained one of the highest reduction rates, as it only required around a 3 % of the total amount of prototypes. The accuracy achieved, although lower than in most of the previous cases, was close to an 84 %, which is a good result given the high data reduction performed.

The NE and FN rank methods showed a very interesting behaviour. When considering their probability mass parameter $\alpha \leq 0.5$, the reduction figures obtained covered a similar range to the reductions obtained with the other strategies: for instance, 1-$NE_{0.20}$ achieved a similar reduction to CHC (around 3 % of the initial set size) or 1-$FN_{0.40}$ is comparable to FCNN (approximately, 20 % of the total amount of prototypes). As it can be seen, these configurations can produce an *aggressive* reduction in the set size, which is often paired with a substantial accuracy loss (e.g., 2-$NE_{0.10}$ which reduces the set to approximately 1 % of its size achieving an accuracy figure around 70 %). However, more *conservative* configurations such as when considering $a = 0.5$ achieved results quite close to the ALL case, with around a third or a fourth of the total number of prototypes.

When considering $\alpha > 0.5$, these methods progressively tend to the ALL case as they also include prototypes located at the lowest positions of the rank (*i.e.*, the ones with the least number of votes). This increase in the reduced set size (up to an 80 % of the complete set size when $\alpha = 0.9$) did not carry a remarkable accuracy improvement (less than a 3 % of improvement with respect to the $\alpha = 0.5$ cases). Nevertheless, it should be noted that the 1-$NE_{0.90}$ improved the accuracy of the ALL case with 80 % of the initial set size, possibly because the method discarded noisy instances present in the datasets.

Name	Noise 0 %		Noise 20 %		Noise 40 %	
	Acc	Size	Acc	Size	Acc	Size
ALL ($k = 1$)	93.4	100	76.3	100	63.3	100
ALL ($k = 3$)	93.5	100	86.3	100	75.7	100
ALL ($k = 5$)	93.4	100	90.9	100	86.1	100
ALL ($k = 7$)	93.1	100	**91.5**	**100**	**89.0**	**100**
ENN	92.3	93.3	**91.0**	**67.1**	**88.4**	**48.7**
CNN	90.3	18.0	67.8	57.3	55.7	72.6
FCNN	90.4	17.7	67.5	55.1	55.5	71.2
ECNN	**90.0**	**10.4**	87.6	9.0	84.4	8.5
EFCNN	90.1	10.5	**88.0**	**8.9**	**84.3**	**8.1**
DROP3	84.6	9.5	74.4	9.9	63.5	10.7
ICF	77.3	15.3	68.2	17.1	59.0	18.4
CHC	**84.4**	**3.1**	**71.5**	**2.6**	**60.2**	**2.3**
1-FN$_{0.10}$	80.8	3.6	83.1	4.2	**83.5**	**4.9**
1-FN$_{0.20}$	**86.2**	**8.3**	87.1	10.0	85.7	11.5
1-FN$_{0.30}$	88.5	14.2	88.3	16.8	81.7	19.3
1-FN$_{0.40}$	90.1	20.3	86.0	24.9	73.4	29.3
1-FN$_{0.50}$	91.3	28.3	80.4	34.9	68.1	39.3
1-FN$_{0.60}$	92.0	38.3	76.9	44.9	64.1	49.2
1-FN$_{0.70}$	92.6	48.4	75.2	54.9	62.9	59.2
1-FN$_{0.80}$	93.0	60.6	73.3	64.9	60.0	69.2
1-FN$_{0.90}$	93.4	80.1	74.0	80.1	59.8	80.5
1-NE$_{0.10}$	**71.7**	**1.3**	**81.6**	**3.3**	83.4	4.4
1-NE$_{0.20}$	79.9	3.3	**86.6**	**8.1**	86.0	10.7
1-NE$_{0.30}$	**85.3**	**6.4**	**88.6**	**14.4**	82.3	18.4
1-NE$_{0.40}$	89.1	10.7	86.9	22.4	75.0	28.4
1-NE$_{0.50}$	**91.3**	**17.3**	80.8	32.3	68.8	38.4
1-NE$_{0.60}$	**92.2**	**27.8**	76.7	42.2	64.5	48.3
1-NE$_{0.70}$	92.8	41.8	74.5	52.2	62.8	58.3
1-NE$_{0.80}$	**93.2**	**60.2**	72.7	62.6	60.0	68.4
1-NE$_{0.90}$	**93.5**	**80.1**	74.3	80.1	60.0	80.3
2-FN$_{0.10}$	80.4	3.6	**82.9**	**3.9**	83.4	4.3
2-FN$_{0.20}$	85.7	8.2	86.8	9.1	85.3	10.3
2-FN$_{0.30}$	88.2	14.0	87.8	15.7	84.6	17.3
2-FN$_{0.40}$	89.8	19.8	86.8	22.8	76.8	26.6
2-FN$_{0.50}$	90.9	27.5	80.3	32.7	68.1	36.7
2-FN$_{0.60}$	91.7	37.5	75.9	42.7	63.6	46.6
2-FN$_{0.70}$	92.4	47.9	73.3	52.7	61.2	56.6
2-FN$_{0.80}$	93.0	60.4	71.2	62.8	58.1	66.6
2-FN$_{0.90}$	93.4	80.1	73.5	80.1	59.0	80.1
2-NE$_{0.10}$	71.2	1.3	**80.5**	**2.7**	**82.9**	**3.6**
2-NE$_{0.20}$	79.6	3.3	**86.0**	**6.8**	**86.1**	**9.0**
2-NE$_{0.30}$	84.7	6.2	88.0	12.3	85.3	15.8
2-NE$_{0.40}$	88.7	10.4	88.3	19.3	77.7	25.0
2-NE$_{0.50}$	**91.0**	**16.4**	81.4	28.9	68.8	35.0
2-NE$_{0.60}$	**92.1**	**26.9**	75.8	38.8	63.8	44.9
2-NE$_{0.70}$	**92.7**	**41.2**	72.0	48.8	60.6	54.9
2-NE$_{0.80}$	93.2	60.1	70.9	60.7	57.5	64.9
2-NE$_{0.90}$	93.5	80.1	74.0	80.1	59.3	80.1

Table 2 Average figures of the results obtained with the datasets considered. Values in bold represent the non-dominated elements defining the Pareto frontier.

In summary, rank methods proved their capability of producing a good trade-off between reduction and classification accuracy in terms of their reduction parameter α. This way, the user is able to tune the reduction degree according to the requirements of the particular application, thus prioritising either accuracy or reduction depending on the particular requirements of the application.

The following lines present the analysis of the performance when noise is induced in the set. As results show qualitatively similar trends, remarks will not focus on a particular noise configuration but on the general behaviour.

The mislabelling noise in the samples dramatically changed the previous situation. Accuracy results for conventional kNN suffered an important drop as noise figures raised. Nevertheless, the use of different k values palliated this effect and improved the accuracy rates. Especially remarkable is the $k = 7$ case in which kNN scored the maximum classification rate compared to the other schemes in both noisy configurations considered.

ENN algorithms proved their robustness in these noisy environments, as their classification rates were always among the best results obtained. Moreover, the reduction rates achieved were higher than in the noiseless scenario, since the prototypes these approaches remove are the ones actually producing class overlapping.

Results with CNN and FCNN schemes depicted their sensitiveness to noise as they obtained some of the worst accuracies in these experiments. Due to the impossibility of discarding noisy elements, the reduction is not properly performed, leading to a situation in which there is neither an important size reduction nor a remarkable performance. Furthermore, the use of different k values did not upturn the accuracy results.

EFCNN and ECNN, on the contrary, were less affected than CNN and FCNN due to the introduction of the editing phase in the process. This improvement is quite noticeable as, while the latter approaches obtained accuracy rates of around 50 % and 60 % with a reduction rate between 50 % and 70 %, the former algorithms achieved precision rates over 80 % with roughly 10 % of the prototypes.

Hybrid algorithms DROP3 and ICF, just like the CNN and FCNN approaches, were not capable of coping with noisy situations either. Accuracy rates obtained were quite poor like, for instance, the case of the ICF method with a 40 % of synthetic noise was not able to reach a 60 % of accuracy. However, it must be pointed out that, despite achieving similar accuracy rates, hybrid algorithms still showed better reduction figures than the CNN and FCNN strategies. For example, for an induced noise rate of 40 %, CNN obtained

an accuracy of 55.7 % with 72.6 % of reduction while DROP3 achieved 63.5 % with only a 10.7 % of prototypes.

Results obtained with the CHC evolutionary scheme showed its relative sensitivity to noise. In these noisy scenarios, although it still depicted one of the highest reduction figures amongst the compared methods with rates around 2 %, its classification performance was significantly affected as no result was hardly higher than 70 %.

The NE and FN rank-based methods demonstrated to be interesting algorithms in the noiseless scenario: for low α values, the reduction rates achieved, together with the high accuracy scores obtained, are very competitive against other methods; at the same time, high α values achieved accuracy figures comparable to, or even higher than, the ALL case with just 20 % to 40 % of the initial amount of prototypes. Results in the proposed noisy situations reinforce these remarks for the former case: on average, none of these algorithms showed accuracy rates lower than 80 % while, at the same time, the number of distances computed did never exceed the 20 % of the maximum. It is also important to point out that, while ECNN and EFCNN schemes also showed a remarkable reduction rate with good accuracy figures, these approaches internally incorporate an editing process for tackling the noise in the data, whereas the rank methods depicted a clear robustness to these situations by themselves, as long as α remains low. Nevertheless, if α is increased, the accuracy of these methods noticeably lowers since the algorithm is forced to include all prototypes in the computed rank, which progressively leads to the ALL case. In such situation, the 1NN search is not able to cope with the noise, which results in the low accuracy figures obtained.

In addition to the commented results, we now tackle the PS-based classification from the point of view of a MOP problem.

Considering the case with no induced noise (Fig. 1), the solution portraying the maximum accuracy result with the least number of prototypes is the 1-NE$_{0.90}$, defining the right-hand end of the Patero frontier. The ALL and ENN configurations do not belong to this frontier as, although they achieved roughly the same accuracy as the previous method, they required a larger amount of prototypes. The rest of the solutions, in spite of exhibiting lower accuracy results, in some cases the loss was not much significant. Examples of this behaviour can be checked in the non-dominated algorithm EFCNN, which achieved accuracy results around 3 % lower than the maximum, computing roughly a fifth of the maximum number of distances, respectively. Regarding the proposed rank methods (in red), it can be

observed that in the non-dominated frontier, in the region of up to 20 % of the total of distances (the one in which most of the PS algorithms studied lie) there is a clear balance between them and the rest of the strategies. This proves the competitiveness of these methods with respect to other classic strategies. Additionally, rank methods also cover the region above the 20 % of distances since the probability mass α allows the selection of the amount of prototypes to maintain.

With respect to the datasets with induced noise (see Fig. 2), the first difference is that the ALL case (with $k = 7$) belongs to the Patero frontier for both noise figures considered. However, other schemes were equally capable of achieving the same accuracy with a lower computational cost. For instance, when considering the case of inducing 40 % of noise in the datasets, both the 7NN and ENN configurations achieved very similar accuracies but, while the former method requires the computation of all the distances, the latter requires less than a half of them.

Rank methods depicted remarkable compromises between accuracy and number of prototypes when considering low α values. An important number of configurations proved to be capable of dealing with these noise figures since they constituted part of the non-dominance frontier. For instance, in the 20 % of noise situation, the 1-NE$_{0.30}$ configuration only differed in a 3 % of accuracy with respect the maximum (given by 7NN) but computes roughly a 15 % of the total amount of distances. However, when setting α to a high value, accuracy was noticeably affected since the algorithms were forced to include noisy prototypes with fewer votes located at the lower parts of the rank. In this case, points moved away from the Pareto frontier, proving not to be interesting configurations for such amount of noise.

6 Conclusions

The k-Nearest Neighbour (kNN) rule is one of the most common, simple and effective classification algorithms in supervised learning. Prototype Selection (PS) algorithms have been used as a way of improving some kNN issues such as computational time, noisy instances removal or memory usage. Due to the importance of the task, there is a large and ever increasing number of approaches to perform PS. Among these approaches, rank methods have recently emerged as an interesting alternative. Based on a particular relevance criterion, these methods rank the prototypes for eventually performing a set reduction guided by a tuning parameter.

In our comprehensive experimentation, voting-based rank methods were compared to a representative set of PS algorithms in order to evaluate their performance. Our results reported a competitive performance against more complex proposals, achieving remarkable compromises between reduction rates and accuracy.

Furthermore, when configured to maintain a rather reduced amount of prototypes, these methods also showed a noteworthy robustness against noise without the requirement of preprocessing with an editing step, as other strategies do. Finally, especial interest lies in the fact that the considered rank methods are guided by a tuning parameter. Although this parameter may be seen as a drawback, it permits the user to enhance either size reduction or classification accuracy depending on the requirements of the system.

Due to the demonstrated competitiveness and straightforward implementation, rank methods constitute an interesting alternative to other classically considered PS strategies. As future work, it would be interesting to develop new heuristics or further extend the proposed ones.

Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants performed by any of the authors.

References

1. Angiulli, F.: Fast Nearest Neighbor Condensation for Large Data Sets Classification. IEEE Transactions on Knowledge and Data Engineering **19**(11), 1450–1464 (2007)
2. Brighton, H., Mellish, C.: On the Consistency of Information Filters for Lazy Learning Algorithms. In: J. Zytkow, J. Rauch (eds.) Principles of Data Mining and Knowledge Discovery, *Lecture Notes in Computer Science*, vol. 1704, pp. 283–288. Springer Berlin Heidelberg (1999)
3. Calvo-Zaragoza, J., Oncina, J.: Recognition of Pen-Based Music Notation: the HOMUS dataset. In: Proceedings of

Fig. 1 Results of the Prototype Selection processes with no induced noise, facing accuracy and size of the reduced set. Non-dominated elements defining the Pareto frontier are highlighted.

the 22nd International Conference on Pattern Recognition, pp. 3038–3043. Stockholm, Sweden (2014)

4. Calvo-Zaragoza, J., Valero-Mas, J.J., Rico-Juan, J.R.: Prototype generation on structural data using dissimilarity space representation: A case of study. In: Pattern Recognition and Image Analysis - 7th Iberian Conference, IbPRIA 2015, Santiago de Compostela, Spain, June 17-19, 2015, Proceedings, pp. 75–82 (2015)

5. Cano, J., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study. IEEE Transactions on Evolutionary Computation **7**(6), 561–575 (2003). DOI 10.1109/TEVC.2003.819265

6. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13**(1), 21–27 (1967)

7. Dasarathy, B.V., Sánchez, J.S., Townsend, S.: Nearest Neighbour Editing and Condensing Tools-Synergy Exploitation. Pattern Analysis and Applications pp. 19–30 (2000)

8. Derrac, J., Cornelis, C., García, S., Herrera, F.: Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection. Information Sciences **186**(1), 73–92 (2012). DOI 10.1016/j.ins.2011.09.027

9. Devijver, P.A., Kittler, J.: Pattern recognition: A statistical approach. Prentice Hall (1982)

10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2 edn. John Wiley & Sons, New York, NY (2001)

11. Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Proceedings of the First Workshop on Foundations of Genetic Algorithms, pp. 265–283. Bloomington Campus, Indiana, USA (1990)

12. Freeman, H.: On the encoding of arbitrary geometric configurations. IRE Transactions on Electronic Computers **EC-10**(2), 260–268 (1961). DOI 10.1109/TEC.1961.5219197

13. Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(3), 417–435 (2012). DOI 10.1109/TPAMI.2011.142

14. García, S., Luengo, J., Herrera, F.: Data Preprocessing in Data Mining, *Intelligent Systems Reference Library*, vol. 72. Springer, Cham, Switzerland (2015). DOI 10.1007/978-3-319-10247-4

15. García-Pedrajas, N., De Haro-García, A.: Boosting Instance Selection Algorithms. Knowledge-Based Systems **67**, 342–360 (2014). DOI 10.1016/j.knosys.2014.04.021

16. Gates, G.: The reduced nearest neighbor rule (corresp.). IEEE Transactions on Information Theory **18**(3), 431–433 (1972). DOI 10.1109/TIT.1972.1054809

17. Hart, P.: The condensed nearest neighbor rule (corresp.). IEEE Transactions on Information Theory **14**(3), 515–516 (1968)

18. Hull, J.: A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence **16**(5), 550–554 (1994). DOI 10.1109/34.291440

19. Nanni, L., Lumini, A.: Prototype reduction techniques: A comparison among different approaches. Expert Systems

(a) Noise 20 %

(b) Noise 40 %

Fig. 2 Results of the Prototype Selection processes with induced noise, facing accuracy and size of the reduced set. Non-dominated elements defining the Pareto frontiers are highlighted for each noise configuration.

with Applications **38**(9), 11,820–11,828 (2011). DOI 10.1016/j.eswa.2011.03.070

20. Natarajan, N., Dhillon, I., Ravikumar, P., Tewari, A.: Learning with noisy labels. In: Advances in Neural Information Processing Systems, pp. 1196–1204 (2013)

21. Pekalska, E., Duin, R.P., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition **39**(2), 189–208 (2006). DOI http://dx.doi.org/10.1016/j.patcog.2005.06.012. Part Special Issue: Complexity Reduction

22. Rico-Juan, J.R., Iñesta, J.M.: New rank methods for reducing the size of the training set using the nearest neighbor rule. Pattern Recognition Letters **33**(5), 654–660 (2012)

23. Ritter, G., Woodruff, H., Lowry, S., Isenhour, T.: An algorithm for a selective nearest neighbor decision rule (corresp.). IEEE Transactions on Information Theory **21**(6), 665–669 (2006). DOI 10.1109/TIT.1975.1055464

24. Sakoe, H., Chiba, S.: Readings in Speech Recognition. In: A. Waibel, K.F. Lee (eds.) Readings in Speech Recognition, chap. Dynamic Programming Algorithm Optimization for Spoken Word Recognition, pp. 159–165. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990)

25. Tomek, I.: An experiment with the edited nearest-neighbor rule. Systems, Man and Cybernetics, IEEE Transactions on **SMC-6**(6), 448–452 (1976). DOI 10.1109/TSMC.1976.4309523

26. Tsai, C.F., Eberle, W., Chu, C.Y.: Genetic algorithms in feature and instance selection. Knowledge-Based Systems **39**(0), 240–247 (2013). DOI 10.1016/j.knosys.2012.11.005

27. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. Journal of the Association for Computing Machinery **21**(1), 168–173 (1974). DOI 10.1145/321796.321811

28. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE Transactions on Systems, Man and Cybernetics **SMC-2**(3), 408–421 (1972). DOI 10.1109/TSMC.1972.4309137

29. Wilson, D.R., Martinez, T.R.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research pp. 1–34 (1997)

Appendix: Partial results obtained

This appendix breaks down the general results into the figures obtained by each single Prototype Selection algorithm and dataset studied. For a better understanding, each table corresponds to a different induced noise configuration of the three considered.

Name	HOMUS		Letter		NIST3		Penbased		USPS		Average	
	Acc	Size	Acc	Size	Acc	Size	Acc	Size	Acc	Size	Acc	Size
ALL ($k=1$)	88.6	100	95.8	100	91.2	100	99.4	100	91.9	100	93.4	100
ALL ($k=3$)	87.5	100	95.8	100	91.9	100	99.1	100	93.3	100	93.5	100
ALL ($k=5$)	86.5	100	95.8	100	92.3	100	99.3	100	93.3	100	93.4	100
ALL ($k=7$)	85.6	100	95.6	100	91.9	100	99.2	100	93.3	100	93.1	100
ENN	85.1	88.6	94.0	95.6	90.7	90.9	99.3	99.4	92.4	92.1	92.3	93.3
CNN	85.8	26.0	92.1	17.3	87.7	22.2	98.4	4.8	87.6	19.9	90.3	18.0
FCNN	85.7	24.6	92.3	17.8	87.8	21.1	98.4	4.9	87.6	20.2	90.4	17.7
ECNN	82.9	15.8	91.0	12.7	87.5	10.8	98.5	3.9	90.2	8.9	90.0	10.4
EFCNN	82.5	15.2	91.4	13.6	88.1	11.4	98.5	3.9	89.8	8.6	90.1	10.5
DROP3	77.3	15.2	84.4	12.6	82.3	9.5	94.0	3.4	85.2	6.7	84.6	9.5
ICF	57.9	18.8	83.0	24.8	82.4	16.1	88.4	9.0	75.0	7.8	77.3	15.3
CHC	71.9	3.2	81.1	7.1	83.9	3.0	95.7	1.2	89.2	0.8	84.4	3.1
1-FN$_{0.10}$	72.6	3.9	70.7	3.5	82.5	4.3	94.6	4.1	83.8	2.2	80.8	3.6
1-FN$_{0.20}$	77.7	8.9	81.4	8.3	86.2	9.4	97.2	9.2	88.4	5.5	86.2	8.3
1-FN$_{0.30}$	80.2	15.1	86.8	14.4	87.9	15.8	98.0	15.8	89.6	10.1	88.5	14.2
1-FN$_{0.40}$	83.5	21.9	89.3	18.6	88.8	22.4	98.8	22.4	90.0	16.2	90.1	20.3
1-FN$_{0.50}$	85.3	30.0	91.9	26.0	89.6	31.0	98.9	30.4	90.5	24.0	91.3	28.3
1-FN$_{0.60}$	86.9	40.0	93.3	36.0	90.3	41.0	99.1	40.4	90.6	34.1	92.0	38.3
1-FN$_{0.70}$	88.3	49.9	94.2	46.0	90.5	51.0	99.2	50.4	91.0	44.9	92.6	48.4
1-FN$_{0.80}$	89.2	60.8	95.0	60.1	90.4	61.2	99.2	60.5	91.2	60.2	93.0	60.6
1-FN$_{0.90}$	89.8	80.3	95.4	80.1	91.0	80.0	99.3	80.1	91.6	80.1	93.4	80.1
1-NE$_{0.10}$	66.6	1.7	54.9	1.2	74.7	2.1	77.7	0.3	84.7	1.1	71.7	1.3
1-NE$_{0.20}$	74.0	4.1	71.4	3.6	82.5	5.0	84.2	0.7	87.6	3.1	79.9	3.3
1-NE$_{0.30}$	78.0	7.6	81.0	7.4	86.9	9.2	90.7	1.6	89.9	6.3	85.3	6.4
1-NE$_{0.40}$	82.4	11.8	88.1	12.6	88.4	15.0	96.1	3.2	90.8	11.1	89.1	10.7
1-NE$_{0.50}$	85.0	18.5	91.7	19.7	89.6	22.4	98.8	7.4	91.4	18.3	91.3	17.3
1-NE$_{0.60}$	87.3	28.7	93.6	29.7	90.0	32.2	99.2	20.1	91.0	28.3	92.2	27.8
1-NE$_{0.70}$	88.8	43.0	94.5	41.6	90.1	42.9	99.3	40.0	91.1	41.3	92.8	41.8
1-NE$_{0.80}$	89.5	60.6	95.1	60.1	90.7	60.0	99.3	60.1	91.5	60.1	93.2	60.2
1-NE$_{0.90}$	89.9	80.3	95.5	80.1	91.0	80.0	99.3	80.1	91.7	80.1	93.5	80.1
2-FN$_{0.10}$	72.7	3.9	69.9	3.5	81.6	4.3	95.0	4.2	83.0	2.1	80.4	3.6
2-FN$_{0.20}$	77.2	8.8	81.3	8.3	86.1	9.4	97.1	9.3	87.0	5.1	85.7	8.2
2-FN$_{0.30}$	80.2	14.9	86.6	14.4	87.3	15.8	98.0	15.9	88.9	9.1	88.2	14.0
2-FN$_{0.40}$	83.3	21.5	89.0	18.2	88.3	22.4	98.7	22.5	89.6	14.4	89.8	19.8
2-FN$_{0.50}$	85.0	29.3	91.5	25.4	89.0	30.7	98.9	30.6	90.0	21.4	90.9	27.5
2-FN$_{0.60}$	86.6	39.3	93.1	35.3	89.7	40.7	99.1	40.5	90.1	31.7	91.7	37.5
2-FN$_{0.70}$	87.9	49.2	94.2	45.3	90.3	50.7	99.2	50.5	90.2	43.7	92.4	47.9
2-FN$_{0.80}$	89.1	60.4	95.2	60.1	90.5	60.9	99.2	60.6	90.9	60.1	93.0	60.4
2-FN$_{0.90}$	89.7	80.3	95.5	80.1	91.1	80.0	99.3	80.1	91.5	80.1	93.4	80.1
2-NE$_{0.10}$	66.6	1.5	54.3	1.3	74.3	2.1	79.3	0.4	81.7	1.1	71.2	1.3
2-NE$_{0.20}$	72.9	3.7	71.6	3.8	82.2	5.0	84.2	1.0	87.1	3.0	79.6	3.3
2-NE$_{0.30}$	77.8	6.9	80.5	7.5	86.2	9.1	89.8	1.9	89.1	5.8	84.7	6.2
2-NE$_{0.40}$	82.1	10.8	88.2	12.6	88.0	14.7	95.7	3.7	89.8	10.1	88.7	10.4
2-NE$_{0.50}$	84.7	17.2	92.0	19.3	89.2	21.7	98.7	7.6	90.5	16.4	91.0	16.4
2-NE$_{0.60}$	87.2	27.3	93.5	29.0	89.8	31.3	99.2	20.1	90.5	26.9	92.1	26.9
2-NE$_{0.70}$	88.9	42.0	94.6	41.1	90.1	42.0	99.3	40.0	90.9	40.7	92.7	41.2
2-NE$_{0.80}$	89.6	60.3	95.2	60.1	90.7	60.0	99.4	60.1	91.3	60.1	93.2	60.1
2-NE$_{0.90}$	89.8	80.3	95.4	80.1	91.2	80.0	99.4	80.1	91.7	80.1	93.5	80.1

Table 3 Results in terms of classification accuracy and set size reduction obtained by the different datasets considered when no noise is induced. The last two columns depict average results of the aforementioned figures of metric.

Name	HOMUS		Letter		NIST3		Penbased		USPS		Average	
	Acc	Size	Acc	Size	Acc	Size	Acc	Size	Acc	Size	Acc	Size
ALL ($k=1$)	72.9	100	79.0	100	75.1	100	83.3	100	71.4	100	76.3	100
ALL ($k=3$)	81.0	100	88.6	100	85.9	100	94.1	100	81.9	100	86.3	100
ALL ($k=5$)	85.2	100	93.5	100	91.2	100	98.4	100	86.4	100	90.9	100
ALL ($k=7$)	85.1	100	94.0	100	91.4	100	99.0	100	87.9	100	91.5	100
ENN	83.1	60.6	92.3	71.1	89.9	62.3	98.5	75.6	91.2	66.0	91.0	67.1
CNN	65.5	61.1	71.4	56.4	65.5	59.5	74.5	52.3	62.2	57.1	67.8	57.3
FCNN	64.4	57.7	71.8	55.0	65.7	55.9	74.0	49.4	61.7	57.6	67.5	55.1
ECNN	80.0	12.3	88.6	10.5	85.0	8.8	96.8	4.8	87.6	8.8	87.6	9.0
EFCNN	79.9	11.4	89.2	11.3	86.4	9.1	96.9	4.4	87.4	8.2	88.0	8.9
DROP3	64.1	15.2	78.0	12.7	69.0	9.5	88.2	5.3	72.9	6.7	74.4	9.9
ICF	48.2	18.8	81.1	28.4	68.1	16.1	87.3	14.3	56.2	7.8	68.2	17.1
CHC	61.1	3.2	67.3	4.8	70.8	3.0	86.7	0.9	71.5	0.9	71.5	2.6
1-FN$_{0.10}$	73.4	4.2	74.7	4.2	83.3	4.4	95.9	4.4	88.0	4.0	83.1	4.2
1-FN$_{0.20}$	77.8	9.9	84.3	10.1	86.9	10.0	97.5	10.2	88.9	9.9	87.1	10.0
1-FN$_{0.30}$	79.6	16.7	88.2	16.8	87.4	16.6	97.7	16.9	88.5	17.0	88.3	16.8
1-FN$_{0.40}$	79.3	24.8	84.8	25.2	84.5	24.9	96.2	24.2	85.3	25.4	86.0	24.9
1-FN$_{0.50}$	75.3	34.7	80.3	35.2	80.0	35.0	89.0	34.2	77.4	35.4	80.4	34.9
1-FN$_{0.60}$	73.5	44.7	77.9	45.2	77.2	44.8	85.0	44.2	70.8	45.4	76.9	44.9
1-FN$_{0.70}$	73.0	54.7	76.0	55.2	75.3	54.8	81.6	54.1	70.3	55.4	75.2	54.9
1-FN$_{0.80}$	71.3	64.7	74.6	65.2	73.6	64.8	79.0	64.2	68.0	65.4	73.3	64.9
1-FN$_{0.90}$	72.1	80.1	75.7	80.1	73.5	80.0	80.4	80.1	68.4	80.2	74.0	80.1
1-NE$_{0.10}$	72.6	3.3	70.7	3.2	82.3	3.2	94.4	3.0	87.8	3.6	81.6	3.3
1-NE$_{0.20}$	77.7	8.0	82.6	8.1	86.1	7.7	97.4	7.5	89.3	9.0	86.6	8.1
1-NE$_{0.30}$	80.0	14.3	87.9	14.6	87.8	13.8	98.2	13.4	89.0	16.0	88.6	14.4
1-NE$_{0.40}$	79.7	22.1	85.6	23.8	85.4	21.8	97.6	20.0	86.0	24.1	86.9	22.4
1-NE$_{0.50}$	75.4	32.0	80.7	33.7	80.1	31.9	89.5	29.7	78.4	34.1	80.8	32.3
1-NE$_{0.60}$	72.9	42.0	78.1	43.7	76.4	41.6	84.5	39.7	71.4	44.1	76.7	42.2
1-NE$_{0.70}$	71.5	52.0	75.9	53.8	74.2	51.6	80.3	49.7	70.4	54.1	74.5	52.2
1-NE$_{0.80}$	70.2	62.5	74.6	63.8	72.6	62.1	77.7	60.3	68.4	64.2	72.7	62.6
1-NE$_{0.90}$	72.1	80.2	76.1	80.1	73.9	80.0	80.9	80.1	68.8	80.2	74.3	80.1
2-FN$_{0.10}$	73.9	3.8	73.3	3.8	83.2	4.2	96.3	4.2	87.6	3.4	82.9	3.9
2-FN$_{0.20}$	78.1	8.9	83.0	9.1	86.7	9.5	97.6	9.5	88.7	8.4	86.8	9.1
2-FN$_{0.30}$	79.7	15.4	86.7	15.6	87.2	16.1	97.5	16.2	88.1	15.0	87.8	15.7
2-FN$_{0.40}$	79.7	22.8	87.9	22.0	84.9	23.5	96.2	23.2	85.4	22.7	86.8	22.8
2-FN$_{0.50}$	75.9	32.6	80.5	32.0	79.8	33.4	88.4	33.1	76.8	32.6	80.3	32.7
2-FN$_{0.60}$	73.3	42.6	76.2	42.0	75.7	43.4	84.1	43.1	70.1	42.6	75.9	42.7
2-FN$_{0.70}$	71.4	52.6	73.4	52.0	73.4	53.4	80.4	53.1	68.1	52.6	73.3	52.7
2-FN$_{0.80}$	69.7	62.6	70.9	62.0	72.3	63.4	77.3	63.1	65.8	62.6	71.2	62.8
2-FN$_{0.90}$	71.6	80.1	74.8	80.1	73.4	80.0	79.7	80.1	67.8	80.1	73.5	80.1
2-NE$_{0.10}$	72.4	2.6	68.0	2.7	80.9	2.8	94.0	2.5	87.3	2.9	80.5	2.7
2-NE$_{0.20}$	76.7	6.5	81.1	6.9	86.2	6.7	97.0	6.3	89.0	7.4	86.0	6.8
2-NE$_{0.30}$	79.6	11.8	86.7	12.7	87.8	12.1	97.7	11.3	88.3	13.7	88.0	12.3
2-NE$_{0.40}$	81.5	18.6	88.8	20.2	86.0	18.8	97.9	17.7	87.6	21.2	88.3	19.3
2-NE$_{0.50}$	75.7	28.2	81.6	30.0	80.1	28.6	90.0	26.5	79.4	31.0	81.4	28.9
2-NE$_{0.60}$	71.7	38.1	76.8	40.0	75.8	38.4	83.1	36.5	71.8	41.0	75.8	38.8
2-NE$_{0.70}$	69.0	48.1	73.0	50.0	72.1	48.4	76.9	46.5	68.8	51.0	72.0	48.8
2-NE$_{0.80}$	68.7	60.8	71.1	60.7	71.5	60.4	76.9	60.0	66.1	61.4	70.9	60.7
2-NE$_{0.90}$	72.1	80.3	75.2	80.1	73.8	80.0	80.3	80.1	68.5	80.1	74.0	80.1

Table 4 Results in terms of classification accuracy and set size reduction obtained by the different datasets considered with when using a 20 % of induced noise. The last two columns depict average results of the aforementioned figures of metric.

Name	HOMUS		Letter		NIST3		Penbased		USPS		Average	
	Acc	Size	Acc	Size	Acc	Size	Acc	Size	Acc	Size	Acc	Size
ALL ($k = 1$)	60.0	100	65.4	100	62.6	100	70.1	100	58.6	100	63.3	100
ALL ($k = 3$)	70.4	100	76.9	100	75.7	100	83.9	100	71.4	100	75.7	100
ALL ($k = 5$)	80.3	100	87.9	100	87.2	100	94.5	100	80.8	100	86.1	100
ALL ($k = 7$)	83.0	100	90.9	100	90.0	100	97.5	100	83.5	100	89.0	100
ENN	80.0	41.7	89.7	53.1	88.2	43.4	96.7	58.3	87.4	46.8	88.4	48.7
CNN	53.4	75.6	58.4	72.5	54.4	74.8	61.9	69.0	50.6	71.3	55.7	72.6
FCNN	53.0	73.3	58.4	71.8	53.8	71.7	61.7	67.6	50.4	71.5	55.5	71.2
ECNN	76.8	10.1	85.9	9.3	82.6	7.6	94.1	6.4	82.4	9.1	84.4	8.5
EFCNN	76.9	9.4	86.2	9.8	84.4	7.3	93.9	5.7	79.9	8.5	84.3	8.1
DROP3	54.2	15.2	69.7	13.9	56.8	9.5	77.8	8.4	58.9	6.7	63.5	10.7
ICF	39.0	18.8	73.2	29.9	56.3	16.1	80.0	19.2	46.7	7.8	59.0	18.4
CHC	48.2	3.2	56.8	3.5	55.7	3.0	80.7	0.9	59.5	0.8	60.2	2.3
1-FN$_{0.10}$	73.9	4.8	76.3	5.0	83.2	4.9	96.4	4.8	87.5	4.8	83.5	4.9
1-FN$_{0.20}$	77.2	11.4	84.3	11.6	85.0	11.2	96.0	11.4	85.9	11.7	85.7	11.5
1-FN$_{0.30}$	73.9	19.5	79.6	19.9	81.5	19.0	94.6	18.3	79.0	19.7	81.7	19.3
1-FN$_{0.40}$	69.7	29.1	70.0	30.6	72.7	29.3	82.9	28.0	71.6	29.4	73.4	29.3
1-FN$_{0.50}$	64.9	39.1	66.1	40.6	66.8	39.3	76.2	37.9	66.3	39.4	68.1	39.3
1-FN$_{0.60}$	60.7	49.1	63.7	50.6	63.3	49.0	72.1	47.9	60.7	49.4	64.1	49.2
1-FN$_{0.70}$	60.8	59.1	62.2	60.6	62.4	59.0	68.3	58.0	60.8	59.4	62.9	59.2
1-FN$_{0.80}$	58.0	69.1	60.7	70.6	58.9	69.0	65.2	67.9	57.1	69.4	60.0	69.2
1-FN$_{0.90}$	58.3	80.4	60.2	80.8	59.2	80.3	64.5	80.1	57.0	80.7	59.8	80.5
1-NE$_{0.10}$	73.0	4.3	76.0	4.5	84.0	4.2	96.2	4.3	87.8	4.6	83.4	4.4
1-NE$_{0.20}$	77.3	10.5	84.2	11.0	85.9	10.1	96.3	10.5	86.3	11.4	86.0	10.7
1-NE$_{0.30}$	74.9	18.4	79.3	19.3	82.1	17.7	95.5	17.2	79.6	19.2	82.3	18.4
1-NE$_{0.40}$	70.1	28.0	70.9	30.2	73.8	27.9	84.7	26.7	75.6	29.0	75.0	28.4
1-NE$_{0.50}$	65.0	38.0	67.1	40.3	67.8	38.0	76.8	36.7	67.4	39.0	68.8	38.4
1-NE$_{0.60}$	60.7	48.1	64.2	50.2	63.6	47.8	72.3	46.6	61.9	49.0	64.5	48.3
1-NE$_{0.70}$	60.6	58.1	62.5	60.2	62.3	57.8	67.5	56.7	61.3	59.0	62.8	58.3
1-NE$_{0.80}$	57.7	68.1	60.9	70.3	59.1	67.8	64.5	66.7	57.8	69.1	60.0	68.4
1-NE$_{0.90}$	58.4	80.3	60.3	80.6	59.4	80.1	64.7	80.1	57.3	80.6	60.0	80.3
2-FN$_{0.10}$	73.9	4.3	76.1	4.4	83.6	4.4	95.7	4.4	87.9	4.1	83.4	4.3
2-FN$_{0.20}$	77.0	10.1	83.2	10.6	84.8	10.3	95.5	10.5	86.1	10.1	85.3	10.3
2-FN$_{0.30}$	76.5	17.2	85.0	17.5	84.0	17.2	94.8	17.2	82.8	17.3	84.6	17.3
2-FN$_{0.40}$	72.0	26.3	74.3	27.4	75.7	27.0	84.5	26.1	77.5	26.5	76.8	26.6
2-FN$_{0.50}$	64.1	36.3	67.6	37.4	67.4	37.2	75.6	36.1	65.9	36.4	68.1	36.7
2-FN$_{0.60}$	60.6	46.3	63.3	47.4	63.0	46.9	70.1	46.1	61.2	46.5	63.6	46.6
2-FN$_{0.70}$	59.0	56.3	60.4	57.4	61.4	56.9	65.9	56.1	59.3	56.4	61.2	56.6
2-FN$_{0.80}$	56.3	66.3	57.9	67.4	58.2	66.9	62.7	66.1	55.3	66.5	58.1	66.6
2-FN$_{0.90}$	57.6	80.0	59.0	80.1	58.6	80.0	63.9	80.1	55.8	80.2	59.0	80.1
2-NE$_{0.10}$	73.8	3.4	74.3	3.8	82.7	3.5	95.6	3.7	88.0	3.8	82.9	3.6
2-NE$_{0.20}$	77.2	8.6	83.5	9.5	86.2	8.5	97.1	8.9	86.7	9.5	86.1	9.0
2-NE$_{0.30}$	77.3	15.3	85.3	16.3	84.9	15.1	95.1	15.5	83.8	16.7	85.3	15.8
2-NE$_{0.40}$	72.7	24.3	75.1	26.8	76.0	24.4	87.2	23.7	77.7	25.6	77.7	25.0
2-NE$_{0.50}$	64.5	34.3	68.3	36.8	68.5	34.6	76.5	33.7	66.2	35.6	68.8	35.0
2-NE$_{0.60}$	60.4	44.3	63.5	46.8	63.5	44.3	70.4	43.6	61.4	45.6	63.8	44.9
2-NE$_{0.70}$	58.1	54.3	60.6	56.8	60.2	54.3	64.7	53.6	59.3	55.6	60.6	54.9
2-NE$_{0.80}$	55.2	64.3	57.9	66.8	57.2	64.3	61.2	63.7	55.9	65.6	57.5	64.9
2-NE$_{0.90}$	57.6	80.1	59.1	80.1	59.4	80.0	64.3	80.1	56.3	80.2	59.3	80.1

Table 5 Results in terms of classification accuracy and set size reduction obtained by the different datasets considered with when using a 40 % of induced noise. The last two columns depict average results of the aforementioned figures of metric.