

# Selecting promising classes from generated data for an efficient multi-class Nearest Neighbor classification

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

Available online: <http://dx.doi.org/10.1007/s00500-016-2176-0>

**Abstract** The Nearest Neighbor rule is one of the most considered algorithms for supervised learning because of its simplicity and fair performance in most cases. However, this technique has a number of disadvantages, being the low computational efficiency the most prominent one. This paper presents a strategy to overcome this obstacle in multi-class classification tasks. This strategy proposes the use of Prototype Reduction algorithms that are capable of generating a new training set from the original one to try to gather the same information with fewer samples. Over this reduced set, it is estimated which classes are the closest ones to the input sample. These classes are referred to as *promising classes*. Eventually, classification is performed using the original training set using the Nearest Neighbor rule but restricted to the promising classes. Our experiments with several datasets and significance tests show that a similar classification accuracy can be obtained compared to using the original training set, with a significantly higher efficiency.

**Keywords** Nearest Neighbor classification · Prototype Reduction · Promising classes

## 1 Introduction

The Nearest Neighbor rule (NN) is the most representative instance-based method for supervised classification. Most of its popularity in classification tasks comes from its conceptual simplicity and straightforward implementation.

This method just requires to work over a metric space, *i.e.*, that in which a distance between two instances can be defined. More precisely, given an input  $x$ , the NN rule assigns to  $x$  the label of the nearest prototype of the training set. An interesting theoretical property of this rule is that its probability of error is bounded above by twice the Bayes error rate [8]. An additional advantage of this classifier is that it is well suited to problems facing *multi-class classifications*, that is, those in which the set of possible labels contains more than two elements [3]. In this sense, unlike other algorithms such as Support Vector Machines, which have to choose some kind of strategy to adapt to this scenario [15], the NN rule does not have to make any adjustment since it is naturally multi-class.

In turn, a NN classifier needs to examine all the training data each time a new element has to be classified. As a consequence, it does not only depict considerable memory requirements in order to store all these data, which in some cases might be a very large number of elements, but also shows a low computational efficiency as all training information must be checked at each classification query [19]. If we assume that the number of instances per class does not change, because there is a minimum number of prototypes needed to

---

This work has been supported by the Vicerrectorado de Investigación, Desarrollo e Innovación de la Universidad de Alicante through the FPU programme (UAFPU2014–5883), the Spanish Ministerio de Educación, Cultura y Deporte through a FPU Fellowship (Ref. AP2012–0939) and the Spanish Ministerio de Economía y Competitividad through Project TIMuL (No. TIN2013-48152-C2-1-R, supported by UE FEDER funds).

---

Jorge Calvo-Zaragoza · Jose J. Valero-Mas · Juan R. Rico-Juan  
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: {jcalvo,jjvalero,juanra}@dlsi.ua.es

achieve a fair accuracy, these problems become particularly uncomfortable. The complexity, both in time and space, grows linearly with the number of classes of the task.

One possibility to improve the time efficiency of NN when dealing with a multi-class problem is narrowing the search space to a smaller set of classes. Given an input, if the  $c$  most promising classes could be known, the NN rule restricted only to those classes would significantly reduce the computational cost of the process. This idea was preliminary studied in [6], when a set of promising classes was used to alleviate the accuracy drop produced by Prototype Selection methods.

It is obvious that both the efficiency and the effectiveness of this process would be closely related to the problem of finding which classes are the most promising ones. Very often, however, these two criteria involve a mutual worsening: finding a fast way to retrieve the most promising classes might lead to a poor accuracy, whereas using a more comprehensive approach might involve greater computational cost.

That is why we seek for a method that aims at optimizing both issues at the same time. To this end, the training set is used as a seed to create a smaller set that is able to represent broadly the same knowledge with far fewer data. In classifying time, this reduced set will be used to predict which  $c$  classes are the most promising following neighboring criteria. Note that this is a fast process since the set can be reduced in advance. Once this  $c$  classes are known, the original training set is used with the NN rule to make the final decision. The key step in this process is how to create this reduced set, for which we use data generation algorithms.

Data generation algorithms can be broadly divided into two groups according to their main goal. First, there exist oversampling algorithms whose main intention is to generate data to fill the search space where no information is available. The best representative of this kind of algorithm is the SMOTE family [7, 14] — designed to work with imbalanced sets — which consist in generating data of the minority class to balance the training set.

On the other hand, data generation algorithms can be seen as a subset of Prototype Reduction algorithms. It comprises the set of techniques aimed at obtaining a reduced set which, if provided to the system, would produce the same output as the original one [13]. In those cases, generation methods, usually called Prototype Generation (PG), are devoted to creating a new set of labeled prototypes that replace the initial training set. Obviously, this new set is expected to be smaller than the original one since the decision boundaries can be defined more efficiently. A particular case of these

techniques is referred to as Prototype Selection (PS), which just select the most representative instances of the set instead of generating new ones. The idea behind PS methods is that an optimal set can be composed by samples of the original training set. For a more comprehensive understanding of the difference between these two approaches, reader is referred to the work of Nanni and Lumini [20].

In this paper it is proposed an approach that uses the training set to generate a new reduced set from which to retrieve fast and accurately the most promising classes. These classes are used to classify the input instance with the original training set. The intention is to obtain a similar accuracy than that obtained with a conventional NN applied directly on the whole set, but with a much greater efficiency.

The rest of the paper is structured as follows: Section 2 describes the intrinsics of our approach; Section 3 presents the experimental setup and the results obtained; and Section 4 concludes the present work, draws the main conclusions and introduces some ideas for future work.

## 2 Selecting promising classes from generated data for an efficient NN classification

Let  $T$  be a training set which consists of pairs  $\{(x_i, y_i) \mid x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_{i=1}^{|T|}$  drawn from an unknown function  $f : \mathcal{X} \rightarrow \mathcal{Y}$ . Typically,  $\mathcal{X}$  is a feature space and  $\mathcal{Y}$  is a discrete set of *labels* or *classes*. The main goal in supervised classification is to approximate this function  $f$  from labeled data.

Given an input  $x \in \mathcal{X}$ , the NN rule hypothesizes about  $f(x)$  by choosing the label of the nearest prototype to  $x$  from  $T$ , based on a dissimilarity function  $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+ \cup \{0\}$ . This algorithm is usually described as a *lazy* learner which, in opposition to *eager* learners, does not build a classification model out of the training data.

Despite the popularity of the NN in classification tasks, this method suffers from several drawbacks, one of which clearly limits its application [2]: as an instance-based classifier, it depicts a rather poor efficiency since many distance computations are repeated each time an input has to be classified due to the aforementioned lack of model. This paper proposes a strategy to alleviate this high computational complexity in tasks involving multi-class classification.

Our strategy is based on three main steps. Given an input  $x \in \mathcal{X}$  to be classified:

1. Estimate a subset of the possible labels  $\mathcal{C} \subset \mathcal{Y}$ , which correspond to those that are more likely to be

the actual label of  $x$ . This subset is called *promising classes*.

2. Consider a subset of the original training set, restricted to those samples that represent any of the promising classes. That is,  $T_C = \{(x_i, y_i) \in T | x_i \in \mathcal{X}, y_i \in \mathcal{C}\}$ .
3. Classify input  $x$  following the NN rule using  $T_C$  as training set.

Clearly, the interesting (and challenging) part of this strategy is to find the so-called promising classes, without significantly increasing the computation needed. Assuming that this stage always selected the actual label of query  $x$  within the set of promising classes, the classifier would have no accuracy loss whereas time savings would indeed be relevant. Alas, this condition seems hard to fulfill in practice and, therefore, appropriate strategies must be considered.

Our proposal is to make use of Prototype Reduction algorithms, which are devoted to generating a reduced training set out of the original one by either generating new samples appropriately located in the space (PG) or selecting the most representative ones (PS). Over this set, the  $c$  nearest classes to the input  $x$  are proposed as promising classes.

Let  $R$  be the reduced set resulting from the reduction method. Given an input  $x$ , the set of the  $c$  (parameter to be fixed) most promising classes  $\mathcal{C}_c \subset \mathcal{Y}$  contains the first  $c$  labels that appear if we query the prototypes of the set  $R$  in ascendant order to the distance to  $x$ .

If the reduced set  $R$  is compact enough (high reduction but keeping the most relevant knowledge), this search should be fast and accurate. Moreover, our next step would also serve to alleviate the accuracy drop that these algorithms might cause, as has been reported in other works [12, 24]. Note that obtaining this reduced set has to be computed just once, and it can be done before classification time so that its computation would not increase the complexity of the classification itself.

## 2.1 Prototype Reduction methods

This section presents the Prototype Reduction techniques that have been considered for this work. In order to get a more comprehensive experimentation, both PG and PS methods have been considered.

### 2.1.1 Prototype Generation

Let us consider the following PG algorithms for the task of building this reduced artificial set:

- Reduction by Space Partitioning is a set of heuristics proposed by Sánchez [23]. Among the number

of strategies considered, the third one (RSP3) consists in dividing the space until a number of class-homogeneous subsets are obtained; a prototype is then generated from the centroid of each subset.

- Evolutionary Nearest Prototype Classifier (ENPC) algorithm was proposed by Fernández and Isasi [11]. It performs an evolutionary search using a set of prototypes that can improve their local quality by means of genetic operators.
- Decaestecker [9] proposed a method that made use of gradient descent and simulated annealing, usually called Mean Squared Error (MSE). The name of the algorithm comes from the use of a mean squared error as cost function for the stochastic search.

The choice of these algorithms has been decided in order to cover the different types of algorithms: MSE as a classical method, ENPC as evolutionary search and RSP3 as heuristic approach.

### 2.1.2 Prototype Selection

A set of representative PS techniques has also been considered in this work:

- Fast Condensing Nearest Neighbor (FCNN) was proposed by Angiulli [1]. It computes a fast, order-independent condensing strategy based on seeking the centroids of each label.
- The Nearest to Enemy (NE) rule [22] gives a probability mass value to each prototype following a voting heuristic based on neighboring criteria. Prototypes are selected according to a parameter (fixed to 0.3 in our case) that indicates the probability mass desired for each class in the reduced set.
- Incremental Reduction Optimization Procedure 3 (DROP3) [25] applies an initial noise filtering step so as to eliminate the dependency on the order of presentation of the instances; after that, these instances are ordered according to the distance to their nearest neighbors and then, starting from the farthest ones, instances which do not affect the generalization accuracy are removed.
- The Iterative Case Filtering Algorithm (ICF) algorithm [4] bases its performance on the coverage and reachability premises to select the subset of instances able to maximize the classification accuracy following the NN rule.

As in the previous case, the choice is expected to cover typical searching methodologies of PS.

Name	Instances	Classes	Features
NIST3	6500	26	520
MNIST	9000	10	650
USPS	9298	10	680
MPEG-7	1400	70	210
HOMUS	15200	32	1760

**Table 1** Description of the datasets used in the experimentation.

### 3 Experimentation

The main goal of our experiments is to test whether the benefits of our proposal are fulfilled: on one hand, whether or not the efficiency classification is improved with respect to the conventional NN classifier; on the other hand, it is important to assess if our process carries any accuracy loss. Additionally, it is interesting to check how accurate the procedure to get the most promising classes is, in order to know if misclassifications are produced due to this search or due to the NN classification itself.

#### 3.1 Corpora

Five different datasets of isolated symbols have been considered: the National Institute of Standards and Technology DATABASE 3 (NIST3), from which a subset of the upper case characters was randomly selected, a subset of the Mixed National Institute of Standards and Technology dataset (MNIST) [18] of handwritten digits, the United States Postal Office handwritten digits dataset (USPS) [16], the MPEG-7 shape silhouette dataset [17], and the Handwritten Online Musical Symbol (HOMUS) dataset [5]. In terms of class representation, these datasets can be considered as being balanced. Note that we focus on datasets with many class labels because our approach is useless otherwise.

Given the difference in the nature of the corpora, we are going to follow an approach that allows us to map them all onto a similar feature-based representation. One of the best representations studied by Pekalska and Duin [21] is EdiCon, which uses the training data itself for building a feature vector from each instance. Once feature vectors are obtained, the Euclidean distance is used as dissimilarity function wherever it is needed.

Table 1 summarizes the details of the datasets considered for this work.

A 5-fold cross-validation process has been applied to each dataset to provide more robust figures with respect to the variance of the training data.

#### 3.2 Results

Three metrics of interest are considered: as a theoretical efficiency measure, the ratio of distances that is needed in the whole process, with respect to the distances computed by the conventional NN rule (Distances); the accuracy of the classification process (Accuracy); finally, we include a metric for assessing the goodness of the most promising classes, calculated as the ratio of times the actual class of the input element to be classified is among this set. This latter figure represents a bound on the accuracy that can be obtained considering only the set of promising classes (Bound).

Table 2 shows results obtained by the different schemes, where the number of promising classes is 2 or 3. The *ALL* row represents the results obtained by considering the initial training set. It can be used for comparison purposes because its efficiency and accuracy is the same that would be obtained with the conventional NN rule. We now present the average results obtained in all datasets to analyze the general trend in the results. Statistical analysis will be presented in the next section.

An initial remark to begin with is that, although the accuracy of conventional NN classification cannot be outperformed by the other schemes, the differences observed are not very high, especially when the number of promising classes is fixed to 3.

Reduction configurations achieve a remarkable decrease of the number of distances computed in all cases considered. Specifically, ENPC shows an interesting behavior when dealing with this task: its accuracy is just 0.3 % and 0.1 % away from the ALL scheme for  $c = 2$  and  $c = 3$ , respectively, while it also depicts one of the highest reduction rates (24.7 % and 29.9 % of the total number of distances computed). FCNN results are also of great interest but less pronounced in this case. In spite of showing very similar figures, FCNN is worse or equal than ENPC in all metrics considered.

Regarding other algorithms, it can be checked that their performances might be competitive against ENPC or FCNN, but just in one criterion. On the one hand, MSE, NE or DROP3 show similar reduction capabilities, but with lower accuracy figures; on the other hand, RSP3 maintains the classification figures at the cost of computing a larger amount of distances (close to a 15 %). In general, ICF depicts the worst behavior: its reduction is one of the less pronounced without achieving competitive classification results.

Moreover, the classification accuracy seems to be strongly related to the bound obtained: the higher the bound, the higher the accuracy. However, it should be noted that the bound of the reduction schemes is always

Algorithm	Distances		Accuracy		Bound	
	$c = 2$	$c = 3$	$c = 2$	$c = 3$	$c = 2$	$c = 3$
ALL	100	100	88	88	94.5	96.5
MSE	25.2	30.6	86.3	86.8	91.1	93.7
ENPC	24.7	29.9	87.7	87.9	93.6	96.2
RSP3	40.9	45.1	87.2	87.5	92.8	95.4
FCNN	34.9	39.3	87.5	87.9	93.4	95.9
NE	20.2	25.5	83.9	85.3	88.2	91.6
DROP3	25.1	30.2	84.3	86.1	89.1	93.2
ICF	34.4	39.3	84.5	86.5	89.1	93.4

**Table 2** Average results obtained considering a 5-fold cross-validation in each dataset. Normalized distances results (%) of the different algorithms are obtained referring to the ALL method with the same  $c$  value. Accuracy of the final classification and the maximum accuracy that could be obtained with the promising classes are also included.

higher than the accuracy of the NN classifier. This implies that the procedure to select the promising classes is quite accurate, and the accuracy drop is caused by the last classification step.

### 3.2.1 Statistical analysis

Previous section presented the average results obtained in our experiments. It allowed us to analyze the general trend of our proposal. However, as commented above, these figures must not be used to draw strong conclusions.

In this section, statistical tests are used for comparing the results objectively [10]. Specifically, Wilcoxon rank-sum tests shall be used in this work, which allow comparing results obtained by pairs.

Our main intention is to check whether our approach is fulfilling the objective of keeping the accuracy of a NN classifier with a significant reduction of its computational cost. To this end, Table 3 shows the results of this test in the three metrics considered above, comparing our different schemes against the ALL configuration (equal to NN classifier). The significance of  $p$  has been established to 0.05. The  $c$  value considered for ALL is the same that the one depicted in the corresponding row.

Not surprisingly, every reduction scheme entails a significantly lower number of distances computed than a NN classifier, so it is more interesting to focus on the accuracy figures. It is clear that no configuration is able to significantly improve that metric. Nevertheless, it can be seen that some of them are able to depict a significantly equal performance. Specifically, ENPC and RSP3 with both 2 and 3 promising classes, and FCNN only when 3 promising classes are considered. Therefore, our initial premise is fulfilled in these cases: it is

Algorithm	c	ALL		
		Distances	Accuracy	Bound
MSE	2	✓	✗	✗
	3	✓	✗	✗
ENPC	2	✓	=	✗
	3	✓	=	✗
RSP3	2	✓	=	✗
	3	✓	=	=
FCNN	2	✓	✗	✗
	3	✓	=	✗
NE	2	✓	✗	✗
	3	✓	✗	✗
DROP3	2	✓	✗	✗
	3	✓	✗	✗
ICF	2	✓	✗	✗
	3	✓	✗	✗

**Table 3** Results obtained for the statistical significance tests comparing our approach against using the original training set. For each comparison, distances computed, accuracy and bound are assessed. Symbols ✓, ✗ and = state that results achieved by elements in the rows significantly improve, decrease or do not differ respectively to the results by the elements in the columns. Significance has been set to  $p < 0.05$ .

possible to achieve a similar accuracy of that obtained with the conventional NN rule with a higher efficiency. The rest of configurations cannot be objectively compared because they improve accuracy at the cost of losing classification accuracy. With respect to the bound, only RSP3 with  $c = 3$  is able to retrieve the same information than using the original set for the task of retrieving the most promising classes.

We shall now perform pairwise comparisons among the different configurations. In order to provide a compact interpretation, and having already concluded that some algorithms are not competitive, we restrict our-

selves to the ones improving the NN rule in some sense. In this regard we have selected ENPC, RSP3 and FCNN schemes with both  $c = 2$  and  $c = 3$ . Table 4 shows the results of such comparison, considering the three metrics of interest. Note that the table is inversely diagonal, since the value in the cell  $(i, j)$  is the opposite of the value in  $(j, i)$ .

For the sake of compactness, let us use  $SCH_c$  to denote the scheme  $SCH$  when considering  $c$  promising classes.

The first thing to emphasize is that ENPC algorithm is clearly the best scheme taking into account efficiency and efficacy of the classification: ENPC<sub>2</sub> significantly reduces the number of distances computed with respect to any other configuration, while maintaining — or even improving (against FCNN<sub>2</sub>)— the accuracy of the classification; ENPC<sub>3</sub> either improves the efficiency (against RSP3 and FCNN<sub>3</sub>) or the accuracy (against FCNN<sub>2</sub>). It is also important to highlight its goodness by taking into account that ENPC is capable of showing a similar or better performance in some metrics under less favorable conditions. For instance, ENPC<sub>2</sub> achieves a similar accuracy than RSP<sub>3</sub> and FCNN<sub>3</sub>, whereas ENPC<sub>3</sub> needs less distances than RSP<sub>3</sub>.

On the other hand, RSP3 and FCNN are reported to be very similar in a general sense: when considering the same  $c$ , they are always significantly equal in all metrics considered. However, FCNN<sub>3</sub> can boast about having the same efficiency than RSP<sub>2</sub>, despite having to seek within a larger number of classes.

With respect to the bound, it seems that it is more relevant the number of promising classes than the reduction method chosen, because schemes with  $c = 3$  obtain a significantly better bound than any other scheme with  $c = 2$ . The only exception is ENPC<sub>2</sub> being equal to RSP<sub>3</sub>. In addition, the configurations with the same number of promising classes are always significantly equal in this metric.

## 4 Conclusions

This paper proposes a procedure that aims at improving the efficiency of the NN rule without decreasing its classification. Due to the lack of a model, this rule has to compute a distance between the input query and every sample of the training set, which usually entails a high computational cost.

This procedure is designed to deal with tasks involving multi-class classification. Our main hypothesis assumes that it is possible to get a reliable estimation of the most probable classes that could represent the input query (promising classes), and restrict the NN

search to the samples of those classes in the training set. This would reduce the time complexity of the classification without compromising accuracy.

Our approach uses Prototype Reduction algorithms, which are expected to maintain the most relevant information from the training set but with far fewer samples, for the task of making a fast and accurate proposal of promising classes. To this end, we selected a representative set of techniques of this kind.

Our results, supported by the use of several datasets and statistical significant tests, report that it is possible to obtain a significantly similar accuracy of that using the conventional NN rule but computing a lower number of distances. In this regard, ENPC reduction algorithm has shown a remarkable ability to cope with this task. It is able to achieve, on average, just a 0.3 % or 0.1 % lower accuracy with around 24 % and 30 % of total number of distances needed by the conventional NN rule, when the set of promising classes is restricted to 2 or 3, respectively.

Another interesting outcome of our results is that the percentage of times that any of the promising classes is the actual label of the input is noticeably higher than the accuracy achieved by the classifier. Therefore, the most interesting line for future work is to try to fill this gap by providing a way to be more accurate in the classification when the set of promising classes is estimated.

## Compliance with Ethical Standards

**Funding:** Work supported by the Spanish Ministerio de Educación, Cultura y Deporte through FPU fellowship (AP2012-0939), Vicerrectorado de Investigación, Desarrollo e Innovación de la Universidad de Alicante through the FPU programme (UAFPU2014-5883) and the Spanish Ministerio de Economía y Competitividad Project TIMuL (No. TIN2013-48152-C2-1-R supported by UE FEDER funds).

**Conflict of Interest:** Authors declare that they have no conflict of interest.

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

## References

1. Angiulli, F.: Fast Nearest Neighbor Condensation for Large Data Sets Classification. *IEEE T. Knowl. Data En.* **19**(11), 1450–1464 (2007)
2. Bhatia, N., Vandana: Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085* (2010)

Algorithm	c	ENPC						RSP3						FCNN					
		c = 2			c = 3			c = 2			c = 3			c = 2			c = 3		
		D	A	B	D	A	B	D	A	B	D	A	B	D	A	B	D	A	B
ENPC	2	–	–	–	✓	=	✗	✓	=	=	✓	=	=	✓	✓	=	✓	=	✗
	3	✗	=	✓	–	–	–	✓	=	✓	✓	=	=	=	✓	✓	✓	✓	=
RSP3	2	✗	=	=	✗	=	✗	–	–	–	✓	=	✗	=	=	=	=	=	✗
	3	✗	=	=	✗	=	=	✗	=	✓	–	–	–	✗	=	✓	=	=	=
FCNN	2	✗	✗	=	=	✗	✗	=	=	=	✓	=	✗	–	–	–	✓	=	✗
	3	✗	=	✓	✗	=	=	=	=	✓	=	=	=	✗	=	✓	–	–	–

**Table 4** Results obtained for the statistical significance tests comparing the best configurations of our approach. For each comparison, distances computed (D), accuracy (A) and bound (B) are assessed. Symbols ✓, ✗ and = state that results achieved by elements in the rows significantly improve, decrease or do not differ respectively to the results by the elements in the columns. Symbol – is used for the case in which a particular method is compared to itself. Significance has been set to  $p < 0.05$ .

- Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
- Brighton, H., Mellish, C.: On the Consistency of Information Filters for Lazy Learning Algorithms. In: J.M. Żytkow, J. Rauch (eds.) Principles of Data Mining and Knowledge Discovery, *Lecture Notes in Computer Science*, vol. 1704, pp. 283–288. Springer Berlin Heidelberg (1999)
- Calvo-Zaragoza, J., Oncina, J.: Recognition of Pen-Based Music Notation: the HOMUS dataset. In: Proceedings of the 22nd International Conference on Pattern Recognition, ICPR, pp. 3038–3043 (2014)
- Calvo-Zaragoza, J., Valero-Mas, J.J., Rico-Juan, J.R.: Improving kNN multi-label classification in Prototype Selection scenarios using class proposals. *Pattern Recognition* **48**(5), 1608–1622 (2015)
- Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* pp. 321–357 (2002)
- Cover, T., Hart, P.: Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on* **13**(1), 21–27 (1967). DOI 10.1109/TIT.1967.1053964
- Decaestecker, C.: Finding prototypes for nearest neighbour classification by means of gradient descent and deterministic annealing. *Pattern Recognition* **30**(2), 281–288 (1997)
- Demsar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* **7**, 1–30 (2006)
- Fernández, F., Isasi, P.: Evolutionary Design of Nearest Prototype Classifiers. *J. Heuristics* **10**(4), 431–454 (2004)
- Garcia, S., Derrac, J., Cano, J.R., 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)
- García, S., Luengo, J., Herrera, F.: Data Preprocessing in Data Mining. Springer (2015)
- Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Advances in intelligent computing, pp. 878–887. Springer (2005)
- Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on* **13**(2), 415–425 (2002)
- Hull, J.: A database for handwritten text recognition research. *IEEE T. Pattern Anal.* **16**(5), 550–554 (1994)
- Latecki, L.J., Lakämper, R., Eckhardt, U.: Shape descriptors for non-rigid shapes with a single closed contour. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 424–429 (2000)
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. In: Intelligent Signal Processing, pp. 306–351. IEEE Press (2001)
- Mitchell, T.M.: Machine Learning. McGraw-Hill, Inc. (1997)
- Nanni, L., Lumini, A.: Prototype reduction techniques: A comparison among different approaches. *Expert Syst. Appl.* **38**(9), 11,820–11,828 (2011). DOI 10.1016/j.eswa.2011.03.070
- Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence). World Scientific Publishing Co., Inc. (2005)
- 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)
- Sánchez, J.: High training set size reduction by space partitioning and prototype abstraction. *Pattern Recogn.* **37**(7), 1561 – 1564 (2004)
- Triguero, I., Derrac, J., García, S., Herrera, F.: A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification. *IEEE T. Syst. Man Cy. C* **42**(1), 86–100 (2012)
- Wilson, D.R., Martinez, T.R.: Instance pruning techniques. In: Proceedings of the Fourteenth International Conference on Machine Learning, ICML ’97, pp. 403–411. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997)