Soft Computing (2021) 25:15403–15415
https://doi.org/10.1007/s00500-021-06178-2

APPLICATION OF SOFT COMPUTING

Prototype generation in the string space via approximate median for data reduction in nearest neighbor classification

Francisco J. Castellanos[1] · **Jose J. Valero-Mas**[1] · **Jorge Calvo-Zaragoza**[1]

Accepted: 20 August 2021 / Published online: 2 September 2021

Abstract
The k-nearest neighbor (kNN) rule is one of the best-known distance-based classifiers, and is usually associated with high performance and versatility as it requires only the definition of a dissimilarity measure. Nevertheless, kNN is also coupled with low-efficiency levels since, for each new query, the algorithm must carry out an exhaustive search of the training data, and this drawback is much more relevant when considering complex structural representations, such as graphs, trees or strings, owing to the cost of the dissimilarity metrics. This issue has generally been tackled through the use of data reduction (DR) techniques, which reduce the size of the reference set, but the complexity of structural data has historically limited their application in the aforementioned scenarios. A DR algorithm denominated as reduction through homogeneous clusters (RHC) has recently been adapted to string representations but as obtaining the exact median value of a set of string data is known to be computationally difficult, its authors resorted to computing the set-median value. Under the premise that a more exact median value may be beneficial in this context, we, therefore, present a new adaptation of the RHC algorithm for string data, in which an approximate median computation is carried out. The results obtained show significant improvements when compared to those of the set-median version of the algorithm, in terms of both classification performance and reduction rates.

Keywords String Space · Data Reduction · k-Nearest Neighbor · Prototype Generation

1 Introduction

In the pattern recognition (PR) field, the objective of supervised classification algorithms is to label unknown prototypes[1] according to a finite set of categories by considering the knowledge automatically gathered from a reference corpus of labeled data. These algorithms have been extensively used in a wide variety of tasks, such as image classification (Ciregan et al. 2012), speech recognition (Abdel-Hamid et al. 2012), written text or music recognition (Plamondon and Srihari 2000; Calvo-Zaragoza et al. 2018), or audio analysis (McVicar et al. 2014), among many others.

One of the main elements in the learning process involved in these algorithms is the type of data representation considered, and two main families are typically distinguished in PR literature (Duda et al. 2012): *statistical* representations, for which data is encoded as a collection of numerical features, and *structural* codifications, for which data is represented as symbolic structures such as graphs, trees, or strings. Although the superior representation capabilities of structural codifications generally achieve higher classification rates than statistical codifications, their applicability is considerably limited owing to the reduced range of algorithms capable of dealing with them (Bunke and Riesen 2012).

Distance-based classifiers constitute one of the rare PR families of algorithms that are capable of processing structural data, since they require only the definition of a dissimilarity measure in order to deal with the actual data (Mitchell 1997). There are, in this respect, some algorithms that are

[1] We have followed the example of previous works in this field as regards the terminology employed, and have used the term *prototype* as a synonym of *sample* or *instance*; *i.e.*, an input element from the classification domain.

✉ Francisco J. Castellanos
fcastellanos@dlsi.ua.es

Jose J. Valero-Mas
jjvalero@dlsi.ua.es

Jorge Calvo-Zaragoza
jcalvo@dlsi.ua.es

[1] Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

able to compute the dissimilarity among structural representations such as graphs (Gao et al. 2010), trees (Bille 2005) or strings (Levenshtein 1966), thus enabling the application of these classifiers. One of the best-known classifiers of this nature is the k-nearest neighbor (kNN) rule (Cover and Hart 1967), which is based on the computation of the distance between the element to be classified and a set of annotated elements, with the k closest elements being those that determine the category to which it is assigned. Nevertheless, despite its high performance, as reported in literature, kNN is also considered a low-efficiency algorithm, since all the elements in the reference corpus must be queried each time a new element needs to be classified (Yang et al. 2019). This issue is of particular importance in the context of structural data owing to the large amount of time consumed in the computation of dissimilarities.

One potential solution associated with the aforementioned issue is the data reduction (DR) paradigm (García et al. 2015). The objective of this family of methods is to reduce the size of the reference corpus so as to improve the time efficiency of the process without, as far as is possible, compromising the performance of the classification. Note that, given its relevance, toolkits such as the KEEL toolkit have been developed in order to allow the direct application of this type of preprocessing stages (Alcalá-Fdez et al. 2009).

DR techniques are generally grouped in two main strategies (Nanni and Lumini 2011): (i) Prototype Selection (PS), which includes methods that replace sets of same-class elements with a single representative prototype belonging to the same set, and (ii) Prototype Generation (PG), which simplifies groups of same-class elements by deriving a new artificial prototype from the elements in the set. In general, PG achieves higher performance rates with respect to PS, but its application is severely restricted by the data representation considered; while the necessary operations are relatively straightforward for statistical representations, this task is considerably more challenging as regards structural representations, in which these operations may not even be mathematically defined.

One possible solution by which to tackle this limitation is shown in the work by Calvo-Zaragoza et al. (2017b). In this case, the initial structural data is mapped onto a statistical representation by means of the so-called dissimilarity space (DS) (Duin and Pekalska 2012) in order to then apply PG in the new representation space. While the results obtained showed the competitiveness of PG together with DS, the mapping process itself entails a loss of the representation capabilities and, eventually, of the classification rate. Avoiding this mapping and addressing DR methods directly in the original data would, therefore, maintain the advantages of the structural representation and consequently enhance the classification scores.

In this line, a recent work by Valero-Mas and Castellanos (2020) proposed the implementation of the well-known PG method reduction through homogeneous clusters (RHC) (Ougiaroglou and Evangelidis 2016) for kNN-based classification, in which data is represented as strings. This approach is based on recursively dividing the initial corpus into homogeneous clusters in order to then replace each of them with a representative prototype generated as the median element of the cluster. However, as the computation of the exact median element from a set of string data is known in the literature as an NP-complete problem (Calvo-Zaragoza et al. 2017a), the work resorted to the use of the set-median value of each cluster, i.e., selecting that median string which minimizes the sum of the distances to the remaining elements in the set.

Despite the simplification of the complexity of data provided by DR techniques such as RHC, it should be noted that they are usually aligned with lower-performance figures. However, it is of crucial importance to maintain the performance at the same time as the reduction process is carried out. In this work, we, therefore, further explore the capabilities of PG when applied to string data. More precisely, and taking the work of Valero-Mas and Castellanos (2020) as a starting point, we expand the capabilities of RHC as regards string data by introducing the computation of the approximate median value, rather than resorting to the set-median strategy. An approximation to the exact median value should, intuitively speaking, improve the performance of the reduction process. Please note that as mentioned above, the computation of the exact median value from a set of string data continues to be an open question in the PR community and, we, therefore, address an approximated version of it.

The remainder of this work is structured as follows: Sect. 2 presents the data reduction field for efficient kNN search, while Sect. 3 contextualizes the issue of the median string calculation. Section 4 presents the adaptation of the RHC algorithm to the string space on the basis of the approximate median, and Sect. 5 explains the evaluation methodology proposed. Section 6 shows and analyzes the results obtained, and finally, Sect. 7 provides the conclusions to this work and shows future work to be addressed.

2 Background in data reduction

As a representative example of lazy learning, kNN is usually associated with low-efficiency performances, since each new element to be classified implies exhaustively consulting the entire reference corpus. Three main approaches with which to deal with this issue can be found in the relevant literature (Rico-Juan et al. 2019):

- *Fast similarity search* this mainly considers the use of indexes and pivot elements in order to speed up the search process.
- *Approximate search* rather than performing exhaustive searches in order to find the exact element, this strategy accelerates the process through approximate searches.
- *Data reduction* (DR) this uses simplification techniques to reduce the size of the reference corpus, obtaining a reduced version of it, ideally without impacting on the performance of the scheme.

This work focuses on the DR family of methods, which tackles the aforementioned performance issue in kNN by proposing policies with which to reduce the size of the reference corpus in order to compute fewer distances. This reduction is typically carried out as a preprocessing stage, and the additional computation it implies that does not, therefore, increase the actual temporal cost of the classification task. As stated above, two particular strategies can be found in DR literature (Nanni and Lumini 2011): PS, which finds groups of similar same-class elements and replaces them with an existent element belonging to the same group, and PG, which replaces sets of same-class samples with new artificial elements with approximate features to the elements to be replaced. PG approaches generally achieve sharper reduction rates than the PS approaches, but their applicability is considerably limited owing to the difficulty of dealing with structural domains.

With regard to the PG challenge, four main categories of the mechanism used for data generation purposes can be found in the relevant literature (Triguero et al. 2012):

- *Class relabeling* considering the premise that there may be errors as regards labeling, those elements whose class does not match their respective nearest elements are relabeled.
- *Centroid-based* when there are similar elements in terms of proximity, these mechanisms find same-class groups and compute their centroids, which are the artificial elements generated that will replace the entire groups. The result is a reduced dataset whose elements are the centroids of the similar-element groups in the original dataset.
- *Positioning adjustment* those elements whose label does not match with their nearest elements are considered noise, and are, therefore, reallocated by modifying their features in order to make them close to their same-class elements.
- *Space splitting or Space partitioning* following the premise that same-class elements are usually close together in the feature space, these techniques subdivide the space into different regions so as to enclose same-class original elements in order to then replace them by generating new artificial elements. Note that the elements generated may not be derived from the original ones, since they represent regions and not groups of elements.

Note that class-relabeling and positioning-adjustment strategies are not purely reduction methods, but rather prototype generation methods, signifying that, within DR contexts, they are usually combined with other methods that effectively reduce the reference set. Moreover, the use of certain methods implies the alteration of certain elements. While approaches that work with statistical data can be computed in an almost straightforward manner, the operations involved become an important issue in structural spaces, thus affecting the applicability of several methods. However, this problem must be solved, given that structural spaces are suitable representations in classification tasks, as proven in several works (Calvo-Zaragoza et al. 2016; Riesen and Schmidt 2019). Reduction processes in this domain would make it possible to improve the efficiency of the classification methods while barely sacrificing performance, and are consequently of significant interest in this research field.

With regard to space partitioning, one of the most recent methods is the so-called reduction through homogeneous clusters (RHC) proposed by Ougiaroglou and Evangelidis (Ougiaroglou and Evangelidis 2016) for statistical representations. RHC tackles the reduction process in a two-stage fashion: *space partitioning*, which splits the space into homogeneous-class regions—typically by using the k-means algorithm (Duda et al. 2012)—and *prototype generation*, in which a new element is created through the use of a feature-wise median operation.

This RHC method was recently adapted to string-based representations of Valero-Mas and Castellanos (2020). In this adaptation, the *prototype generation* stage of the RHC algorithm required the computation of the median value of a set of strings. Given that this computation is an NP-hard problem (Calvo-Zaragoza et al. 2017a), the median computation was tackled by considering a set-median strategy instead.

In this paper, we present an extension to the aforementioned work, which replaces the set-median strategy with an iterative method that can be used to approximate the median value of the string data set. This signifies that new artificial elements are now created rather than simply being selected in the *prototype generation* stage of the RHC algorithm. Our hypothesis is that the computation of the (approximate) median string should improve the performance of the reduction process with respect to the use of the set median, given that the former is more representative of a set of strings.

3 Median string computation

Let $(\mathcal{X}, d)$ represent a metric space, where $\mathcal{X}$ constitutes a set of prototypes codified as strings and $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_0^+$ is a dissimilarity measure. For a subset of data $\mathcal{S} \subseteq \mathcal{X}$, the median string p constitutes the element that minimizes the sum of distances to all the prototypes in $\mathcal{S}$ (Nicolas and Rivals 2005). This can be mathematically formulated as:

$$\arg\min_{p \in \mathcal{X}} \sum_{s \in \mathcal{S}} d(p, s) \tag{1}$$

Note that no restriction is imposed on element p but that of belonging to set $\mathcal{X}$. Nevertheless, in some cases, p is additionally constrained to being one of the elements of the subset $\mathcal{S}$, and is, therefore, known as the set-median string (Kohonen 1985).

In spite of its conceptual simplicity, the computation of the median value in the string domain still constitutes an open research question owing to the fact that it is an NP-complete problem (Calvo-Zaragoza et al. 2017a). This signifies that while works such as that of Kruskal (1983) propose strategies for the exact median calculation of this median value in the string domain, its applicability is severely conditioned by its extremely low efficiency.

As a consequence of the high time requirements of the string median computation, some works have addressed the calculation of an approximate version of the median string so as to consider its use in practical applications in which time efficiency is a crucial aspect. One of the strategies most frequently addressed in this context is that of building the approximate median from an initial string to which successive modifications are applied. For example, the work of Fischer and Zell (2000) or that of Hinarejos (2003) addressed the median computation in the string space through the use of the aforementioned strategy, but in spite of the high quality of the calculated median, the high temporal cost makes them unsuitable for practical scenarios.

In order to reduce the computational efforts of the strategies commented on previously, Abreu and Rico-Juan (2014) proposed an algorithm with which to obtain the median element of a set of string prototypes more efficiently than its predecessors had done, while maintaining the performance to a great extent. The objective of this algorithm is to obtain he string that minimizes the dissimilarity between the computed median and all the prototypes belonging to the original set of data strings. It begins with an approximation of it by means of the set-median string. The method then successively modifies the approximate median sample using the most repeated transformations that have been computed for all the initial set. These transformations are typically modeled by means of the edit distance (Levenshtein 1966), a metric that indicates the different transformations that should be applied to a string that is to be converted into another string. These transformations can be only three possible operations: insertion, deletion, and substitution. Using a similar principle as a basis, the work by Mirabal et al. (2019) also considers the different transformations reported by the edit distance and proposes an innovative ranking strategy with which to establish the correct order in which they must be applied. Other research efforts can be found in Kaysar and Khan (2020), in which an approximate version of the median string is computed using Markov chains for DNA motif classification, or in Chakraborty et al. (2021), which performs this approximation by resorting to a probabilistic framework based on the so-called Ulam metric.

In this work, we decided to use the algorithm of Abreu and Rico-Juan (2014) owing to its reported efficiency. Note that while this algorithm may not achieve the most adequate approximation of the exact median string, it stands out as an ideal strategy for real-world scenarios thanks to its remarkable efficiency.

Finally, for the sake of clarity, Table 1 shows a representative summary of the aforementioned literature survey.

4 Method

Of all the different space partitioning techniques for DR, the reduction through homogeneous clusters (RHC) algorithm proposed by Ougiaroglou and Evangelidis (2016) stands out as one of the most recent state-of-the-art PG methods with which to optimize the efficiency of the kNN classifier. Please recall that RHC consists of two stages: (i) that concerning *space splitting* for the creation of class-homogeneous sets of data, and (ii) that concerning *prototype generation* for the derivation of a new element from the elements in each set.

In spite of its good performance, the original conception of the RHC algorithm is restricted to statistical data representations; while calculations such as the mean or median values can be straightforwardly computed for statistical data, these operations are not easily applicable to structural codifications. In this respect, the work of Valero-Mas and Castellanos (2020) is the first attempt to adapt the RHC algorithm to the case of string data by considering a set-median strategy. The method presented in this paper extends the aforementioned work to a new point at which new prototypes may be generated through the calculation of an approximation of the exact median string by means of a state-of-the-art approximation algorithm. This extension is introduced as follows.

Let $\mathcal{T}$ be the initial training set with annotated elements, with $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{T}|}$, where x_i stands for the i-th element within a structural space $\mathcal{X}$, and y_i is the class to which it belongs, and $\mathcal{Y}$ is the set of possible categories. Let $\zeta(\cdot)$ also be a function that retrieves the class of a given element from

Table 1 Summary of approaches found in the literature related to the median string computation

References	Principle described or proposed
Kohonen (1985)	Set-median algorithm with which to select the element within the set of data that minimizes the distance to the rest of elements
Fischer and Zell (2000)	Algorithm with which to obtain an approximate version of the median string by means of an iterative process in which multiple operations on the candidate element are performed simultaneously in each step
Hinarejos (2003)	Perturbation-based algorithm with which to obtain an approximate median value of a set of strings by performing successive modifications from a given initial one
Nicolas and Rivals (2005)	This describes the median string computation as a minimization problem of the sum of distances.
Abreu and Rico-Juan (2014)	Method that initially computes the set-median string and then performs successive transformations (based on the edit distance) among the candidate and the elements within the set of data with the aim of minimizing a global cumulative dissimilarity score
Mirabal et al. (2019)	Proposal based on the use of the edit distance in order to estimate the transformation required and a new ranking proposal that establishes their order
Kaysar and Khan (2020)	Approximate median string computation based on Markov chains for DNA motif classification
Chakraborty et al. (2021)	Median string approximation algorithm based on the use of the Ulam median probabilistic model

$\mathcal{X}$, i.e., $\zeta(x_i) = y_i \in \mathcal{Y}$. Moreover, let $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_0^+$ denote a dissimilarity function in space $\mathcal{X}$, which, in the case of string data, we assume to be the well-known edit distance (Levenshtein 1966). With all of the above, Algorithm 1 shows the algorithm employed to reduce $\mathcal{T}$ through the use of the RHC approach.

Note that Algorithm 1 lacks the definition of two operators, and more precisely, the *initial-median*$(\cdot)$ and *cluster-median*$(\cdot)$ operators in lines 5 and 12, respectively. These operators, which are responsible for the retrieval of the median value of a set of strings, constitute the main difference with respect to the proposal of Valero-Mas and Castellanos (2020); while in the referenced work both operators were constrained to a set-median calculus, in this proposal these operators may retrieve either the approximate or the set-median values. Note also that these operators are now independent from each other, which provides the possibility of selecting different median estimations for each of the processes. This yields four possible variations of the RHC algorithm—considering either the set median or the approximate median of each of the *initial-median*$(\cdot)$ and *cluster-median*$(\cdot)$ processes. This will be commented on and examined later in Sect. 5.

In order to approximate the median value of a set of strings, we consider the strategy proposed by Abreu and Rico-Juan (2014) . In conceptual terms, this algorithm starts from the calculation of the set-median value of the set of strings in question in order to then apply a series of transformations with which to obtain the approximated median strings. With regard to the previous notation, it is possible to define *set-median*$(\cdot)$ as the function that retrieves the set-median value of a set of strings using Eq. (2).

$$\text{set-median}(\mathcal{P}) = \arg\min_{p \in \mathcal{P}} \sum_{p' \in \mathcal{P}} d\left(p, p'\right) \tag{2}$$

Some additional definitions are required in order to eventually introduce the strategy of Abreu and Rico-Juan (2014) so as to calculate the approximate median. Let $\mathcal{P}$ be a set of same-class prototypes. Moreover, let *dissimilarity-transformations* $\left(p, p'\right)$ be used to denote the function that retrieves the list of transformations, considering the aforementioned edit distance required to convert the string $p \in \mathcal{P}$ into another string $p' \in \mathcal{P}$. Let *histogram*$(\cdot)$ be the function employed to obtain the histogram from a given list of elements and *transform*(p, h) be the function that applies a given transformation h to string prototype p. With all of the above, the approximate median algorithm can be described as shown in Algorithm 2.

Note that the algorithm uses a preliminary approximation by employing the set-median strategy. As it progresses, this initial median is adjusted in order to minimize the dissimilarity between the median string and all the prototypes involved in this operation—those belonging to a particular cluster. These adjustments are performed according to the most representative transformations that should be applied in all the prototypes involved so as to convert them into the current median. In those cases in which it is not necessary to perform any transformations, the algorithm returns directly the current value without making any more adjustments. This usually occurs when a cluster contains either one element or repeated ones. However, if there are any transformations to be carried out, the algorithm first obtains a global histogram for the transformations of all prototypes and then retrieves only the most frequently repeated transformation from them. If the application of this selected transformation to the current median achieves improvements in terms of reducing the sum of distances to the other prototypes in the group, the median string is then replaced with the new candidate and the algorithm is repeated. In this case, the algorithm will fin-

Algorithm 1 Reduction through use of Homogeneous Clusters, extracted and adapted from Valero-Mas and Castellanos (2020).

function RHC($\mathcal{T}$) ▷ Initial set $\mathcal{T} = t_1 \dots t_{|\mathcal{T}|}$
  $\mathcal{R}, \mathcal{C} \leftarrow \emptyset$
  for each $y \in \mathcal{Y}$ **do** ▷ Clustering
    $\mathcal{V} \leftarrow \{t_i \in \mathcal{T} \quad : \quad \zeta(t_i) = y\}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup$ *initial-median* $(\mathcal{V})$
  end for
  for each $c \in \mathcal{C}$ **do**
    $\mathcal{S} \leftarrow \left\{t_i \in \mathcal{T} \quad : \quad c = \arg\min_{c' \in \mathcal{C}} \; d\left(t_i, c'\right)\right\}$
    if $|\{\zeta(t_i) \quad : \quad t_i \in \mathcal{S}\}| > 1$ **then**
      $\mathcal{R} \leftarrow \mathcal{R} \cup$ RHC $(\mathcal{S})$ ▷ Non-homogeneous
    else
      $\mathcal{R} \leftarrow \mathcal{R} \cup$ *cluster-median* $(\mathcal{S})$ ▷ Homogeneous
    end if
  end for
  return $\mathcal{R}$ ▷ Reduced version of $\mathcal{T}$
end function

Algorithm 2 Approximate-median of Abreu and Rico-Juan (2014).

function *transformations*$(s, \mathcal{Z})$
  $E \leftarrow \emptyset$ ▷ Transformations
  for each $z \in \mathcal{Z}$ **do**
    $E \leftarrow E \parallel$ *dissimilarity-transformations* (z, s)
  end for
  return E
end function

function *approx-median*$(\mathcal{P})$ ▷ $\mathcal{P} = p_1 \dots p_{|\mathcal{P}|}$
  $m \leftarrow$ *set-median* $(\mathcal{P})$ ▷ Initial set-median computation
  loop forever
    $D \leftarrow$ *transformations* $(m, \mathcal{P})$ ▷ Edit Distance transformations
    if $|D| = 0$ **then**
      return m
    end if
    $H \leftarrow$ *histogram* $(\mathcal{D})$
    $h \leftarrow \arg\max(H)$ ▷ Most repeated transformation
    $m' \leftarrow$ *transform* (m, h) ▷ Median candidate
    $d_m \leftarrow \sum_{\forall p \in \mathcal{P}} \; d(m, p)$
    $d_{m'} \leftarrow \sum_{\forall p \in \mathcal{P}} \; d\left(m', p\right)$
    if $d_{m'} < d_m$ **then**
      $m \leftarrow m'$ ▷ Median has been improved
    else
      return m ▷ Median has not been improved
    end if
  end loop forever
end function

ish when no improvements to the overall distance value are possible.

5 Experimental setup

In this section, we present the experimental setup considered for the assessment of our proposal, which is shown graphically in Fig. 1. We should remind the reader that the experiments must evaluate the efficiency of the kNN classifier after the RHC has been applied directly to the string domain. Note that in these experiments, we have fixed a value of $k = 1$ for the classifier.

As stated in Sect. 4, the RHC requires the configuration of two stand-alone processes in order to estimate the median value from a set of strings. Given that we are considering two different methods with which to estimate this value (the approximate and the set-median approaches), we shall analyze the four resulting combinations, henceforth denominated as SET, SET- APPROX, APPROX- SET and APPROX. The relation between these four situations with the type of median calculation is shown in Table 2. Finally, note that SET stands for the baseline with a pure set-median strategy inherited

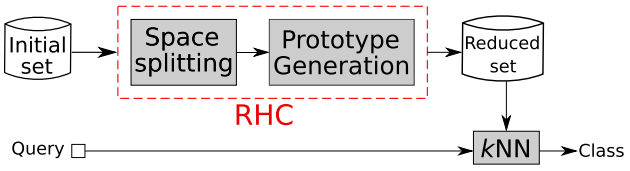


Fig. 1 Graphical description of the experimental scheme proposed. The initial training data is reduced using the RHC algorithm for DR, after which a given query is classified using the kNN rule with $k = 1$

from Valero-Mas and Castellanos (2020), while the rest correspond to our different proposals.

5.1 Corpora

For the experimentation, we have considered three different datasets with isolated handwritten digits and characters, details of which are provided in Table 3:

- NIST: Special database of National Institute of Standards and Technology with images of 28×28 pixels of handwritten characters (Wilkinson 1992).
- MNIST: Modified National Institute of Standards and Technology with a collection of handwritten digits provided by LeCun et al. (1998) as images of 28×28 pixels.
- USPS: The United States Postal Service dataset with a collection of handwritten digits images of 16×16 pixels (Hull 1994).

Table 2 Set of combinations resulting from the two aforementioned potential operations in Algorithm 1: *initial-median*$(\cdot)$ and *cluster-median*$(\cdot)$

Denotation	Operation	
	initial-median$(\cdot)$	*cluster-median*$(\cdot)$
State of the art		
SET	*set-median*$(\cdot)$	*set-median*$(\cdot)$
Our method		
SET- APPROX	*set-median*$(\cdot)$	*approx-median*$(\cdot)$
APPROX- SET	*approx-median*$(\cdot)$	*set-median*$(\cdot)$
APPROX	*approx-median*$(\cdot)$	*approx-median*$(\cdot)$

Table 3 Description of the datasets considered in this work. The table shows the total number of instances, the number of categories or classes and the size of the images with isolated handwritten elements

Dataset	Instances	Classes	Image size (px)
NIST	5,200	26	28×28
MNIST	10,000	10	28×28
USPS	8,684	10	16×16

Since the corpora considered contain data as images depicting isolated symbols, for the purpose of our research, we have considered the algorithm proposed by Rico-Juan and Micó (2003), in which the contour of the symbols is extracted and later encoded as Freeman Chain Code (Freeman 1961). Note that we acknowledge that this classification scheme is not the optimum, as more recent developments such as those based on convolutional neural networks achieve better results in terms of classification performance. Nevertheless, the data in this work constitutes a showcase with which to prove our hypothesis of applying PG to a string-based collection of data, and we do not claim that the classification performance constitutes any state-of-the-art result.

5.2 Metrics

Since this work tackles the efficiency issue typically associated with the kNN rule, the evaluation requires the use of two different aspects: on the one hand, it is necessary to assess the efficiency figures achieved after the reduction process and, on the other, it is necessary to verify the classification performance achieved by the scheme.

As occurred with the performance, in order to avoid any bias toward any particular class in the case of a certain data imbalance, we have considered the use of the *F-measure* (F_1). In a two-class classification problem, F_1 is described as

$$\mathrm{F}_1 = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{3}$$

where TP represents *True Positives* or the correctly classified elements, FP stands for *False Positives* or type I errors and FN indicates *False Negatives* or type II errors.

As given in Table 3, the corpora considered contain non-binary class distributions. We, therefore, considered the macro-averaged F_1 score, a multi-class metric computed as the average of the F_1 score of each class, and formulated as

$$\mathrm{F}_1^M = \frac{1}{|\mathcal{Y}|} \cdot \sum_{i=1}^{|\mathcal{Y}|} \mathrm{F}_1^{(i)}, \tag{4}$$

where $\mathcal{Y}$ is the set of possible categories and $\mathrm{F}_1^{(i)}$ is the F_1 score for the class $y_i \in \mathcal{Y}$.

In order to carry out a further analysis of the results, we also examined the Precision (P) and Recall (R) figures of merit, whose harmonic mean is the F_1 and which are defined as

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad \mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

Analogously to the F_1 case, since the corpora used for the experiments depict a non-binary class distribution, we con-

sidered the use of macro-averaged Precision (P^M) and Recall (R^M) metrics, which are computed as the as the average class-wise Precision and Recall figures, respectively.

In order to evaluate the reduction capabilities of the proposals, we have studied the resulting set sizes of the training data in each case. Note that computation time was discarded owing to its variability depending on the load of the computing system.

Given that the objective of DR methods is to simultaneously optimize two contradictory goals—set size reduction and classification performance—some approaches will retrieve sharper reduction figures at the expense of a decrease in the classification rate, while others will simple have the opposite behavior. It is not, therefore, possible to retrieve a global optimum.

DR can consequently be addressed as a multi-objective optimization problem (MOP), in which the two objectives to be optimized are the aforementioned reduction capabilities and classification performance. The possible solutions in this framework are usually retrieved by employing the concept of non-dominance: one solution is said to dominate another if it is better or equal in each goal function, and at least strictly better in one of them. The resulting elements, which are typically known as non-dominated elements, constitute the so-called Pareto frontier in which all elements are optimal solutions of the problem and there is no hierarchy among them. In Sect. 6, we shall, therefore study the aforementioned frontier so as to determine the most representative case studies in terms of efficiency and performance within the cases considered in this work.

Finally, we decided to use an additional metric that relates the performance and degree of reduction: the so-called *estimated profit per prototype* (Valero-Mas et al. 2016). This measure is defined as the ratio between the classification rate and the number of distances computed or, in this context, the number of elements in the training set. Keeping this in mind, in this case, we are not relying on the classification accuracy as the efficiency figure, but have instead adapted the original definition of this metric in order to use the F_1 score rather than the classification accuracy.

6 Results

This section presents the results of the experimentation and analyzes the figures obtained with the proposed extensions of the RHC algorithm to the baseline model considered.

Table 4 shows the results obtained for each corpus in terms of F_1 and training-set size for all the configurations considered, along with the case in which no reduction method is applied. The figures obtained after averaging out the individual corpus results are also provided, and are additionally shown graphically in Fig. 2.

An initial remark that can be made is that, as expected, the classification performance achieves its maximum when no data reduction is applied. However, since the corpora are composed of string elements, the distance-based algorithms have an extremely poor efficiency owing to the fact that they are a type of structural representation. This highlights the importance of designing a data reduction strategy with which to reduce the complexity of that computation.

Of all the reduction strategies considered, the SET strategy corresponds to the baseline whose figures are, in terms of performance and size, the principal ones to be compared to our approach. In this respect, please note that, while the approximate median versions—APPROX, SET- APPROX, and APPROX- SET—have slight losses in performance, they are compensated by the degree of reduction achieved by the RHC algorithm. With regard to the size of the reduced dataset, it should be noted that APPROX and APPROX- SET monopolize the best results, attaining an average of 18.6% of size with respect to the non-reduced case, while the SET baseline case reduces it to only 19.4%. Note that SET- APPROX also surpasses the SET case as regards reduction, but the margin is simply too small to be relevant with 19.3%.

In addition to the metrics analyzed above, we also show the results in terms of P^M and R^M in Table 5. As can be observed, both the Precision and the Recall metrics are highly correlated with the F_1^M figures. As expected, the case without any reduction process, labeled as ALL, attained the highest performance values for all the corpora considered, given that the entire reference partitions are used for each data collection. However, as noted previously, the reader should recall that this situation is the least efficient owing to the high complexity of the dissimilarity computation among string data.

Focusing on the reduction processes, the state-of-the-art process—SET—underwent a drop in the results, as occurred with those obtained with the F_1^M metric. However, our method SET- APPROX clearly surpasses these results for all corpora and figures of merit, thus representing an improvement when compared to the SET method. The following is of particular note: for the different corpora, it achieves an average of $P^M = 88.4\%$ and $R^M = 87.8\%$, which is approximately 1% higher than the reference SET strategy. The other proposed methods—APPROX- SET and APPROX—also obtain competitive figures, but very similar to those obtained by the SET process in terms of P^M and R^M. However, note that, although these strategies do not attain the best performance results, the degree of reduction is higher than that of the SET approach, thus reducing the complexity of the search and classification process. All of the above makes it possible to conclude that there is a very similar pattern to that shown in Table 4.

In terms of non-dominance, it is important to mention that the Pareto frontier is defined by the SET- APPROX and APPROX approaches, which are two of the proposals of this work, together with the ALL case. In this respect, the SET

Table 4 Results in terms of F_1 and training-set size with standard deviation σ with respect to the original corpus size. Each row indicates the results for all the situations considered, as defined in Table 2

Case study	Metric (%)	Corpora			
		NIST	MNIST	USPS	Avg.
State of the art					
ALL	F_1^M	**89.2 ± 1.4**	**94.1 ± 0.6**	**90.2 ± 0.4**	**91.2 ± 0.8**
	Size	**100**	**100**	**100**	**100**
SET	F_1^M	84.2 ± 2.0	90.7 ± 0.6	85.1 ± 1.0	86.7 ± 1.2
	Size	22.6 ± 0.3	15.0 ± 0.1	20.6 ± 0.4	19.4 ± 0.3
Our method					
SET- APPROX	F_1^M	**85.5 ± 1.3**	**91.8 ± 0.7**	**86.3 ± 1.1**	**87.9 ± 0.9**
	Size	**22.6 ± 0.3**	**14.5 ± 1.0**	**20.6 ± 0.4**	**19.3 ± 0.6**
APPROX- SET	F_1^M	83.7 ± 1.4	90.1 ± 0.4	**84.0 ± 0.7**	85.9 ± 0.8
	Size	22.2 ± 0.3	14.1 ± 0.1	**19.5 ± 0.3**	18.6 ± 0.2
APPROX	F_1^M	**84.0 ± 1.6**	**90.6 ± 0.5**	**84.6 ± 0.9**	**86.4 ± 1.0**
	Size	**22.1 ± 0.4**	**14.1 ± 0.4**	**19.6 ± 0.5**	**18.6 ± 0.4**

The values in bold type show the non-dominated elements for each corpus

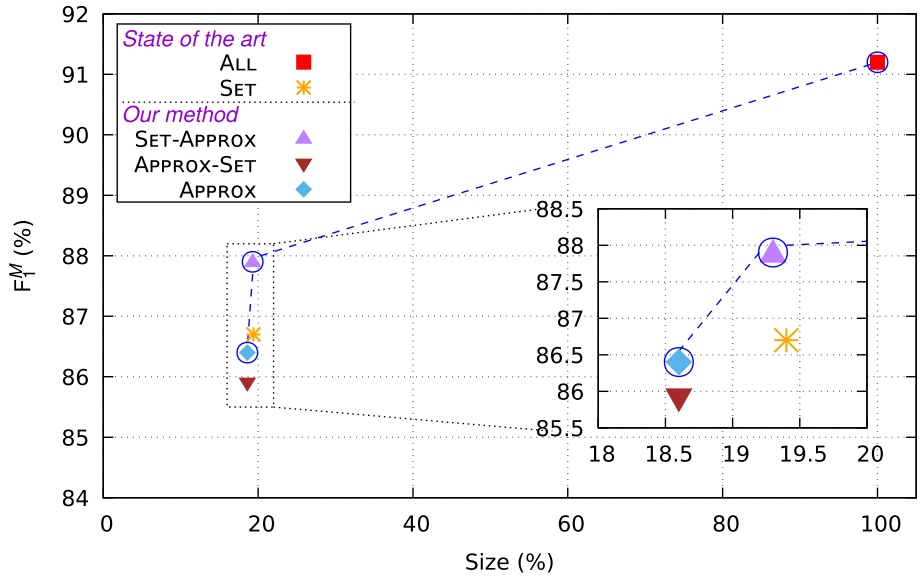


Fig. 2 Average results in terms of resulting set size and F_1^M represented as a graph. The blue dashed line represents the Pareto frontier, while the circle points represent the non-dominated elements of the curve

approach, which constitutes the baseline of this work, is improved by these proposals. This can be clearly observed in Fig. 2, in which the SET case is located behind the Pareto frontier.

The loss of performance in the approximate median versions appears to be related to the high degree of reduction achieved in the corpora. Nevertheless, the advantages in terms of reduction could be representative in terms of efficiency. In order to complement the results above it is, therefore, necessary to compare the performance-size ratio, i.e., the contribution made by each element in the final dataset to the performance computation or *estimated profit per prototype*, which is depicted in Table 6.

The analysis of the results obtained in terms of the *estimated profit per prototype* shows that precisely the approximate-based methods increase this ratio, thus confirming the positive overall performance of the proposals in this work when compared to both the set-median and non-reduction cases.

In short, APPROX and SET- APPROX have proven to be particularly relevant as non-dominated approaches, achieving great reduction levels without notably compromising the performance with respect to the baseline considered, i.e., the set-median RHC method. Although SET- APPROX does not attain such a high reduction coefficient as the other approximate methods, note that it still improves the degree of reduction in SET, in addition to achieving a better performance with all corpora. Although Table 6 shows that this method obtains slightly lower efficiency figures within the approximate-based versions, we can observe that it sacrifices the reduction factor for an improvement in performance, surpassing the baseline results in all cases. APPROX- SET also

Table 5 Results in terms of P^M and R^M with standard deviation σ with respect to the original corpus size

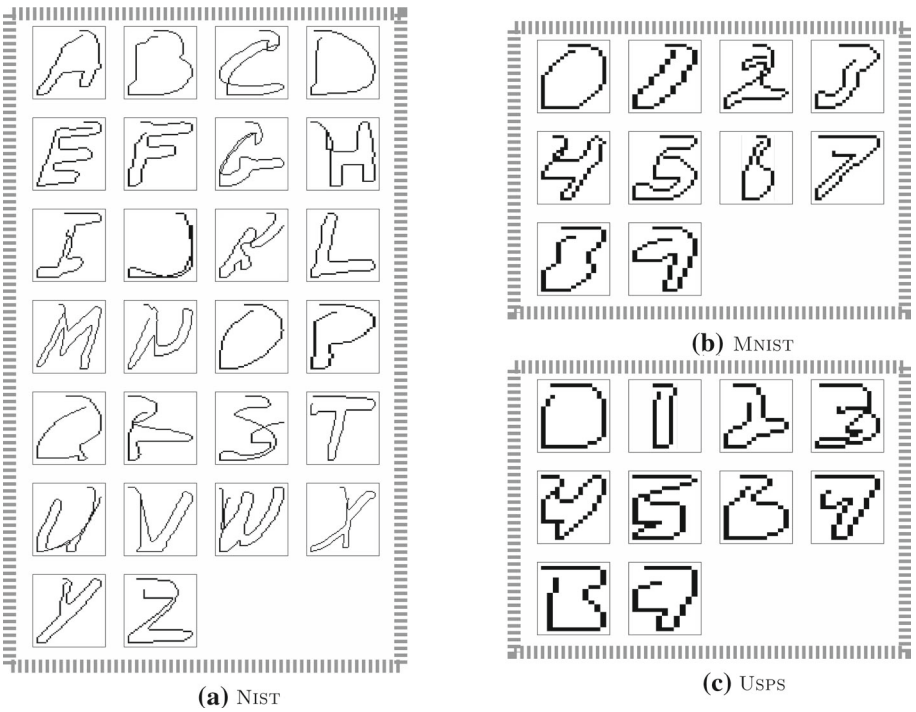
Case study	Metric (%)	Corpora			
		NIST	MNIST	USPS	Avg.
State of the art					
ALL	P^M	90.0 ± 1.1	94.2 ± 0.6	90.7 ± 0.5	91.6 ± 0.7
	R^M	89.2 ± 1.4	94.1 ± 0.6	90.0 ± 0.3	91.1 ± 0.8
SET	P^M	85.2 ± 1.2	90.7 ± 0.6	85.6 ± 1.1	87.2 ± 0.7
	R^M	84.2 ± 2.0	90.7 ± 0.6	84.9 ± 0.9	86.6 ± 1.1
Our method					
SET- APPROX	P^M	86.4 ± 1.1	91.9 ± 0.7	86.8 ± 1.2	88.4 ± 1.0
	R^M	85.4 ± 1.3	91.9 ± 0.7	86.1 ± 1.1	87.8 ± 1.0
APPROX- SET	P^M	84.8 ± 1.0	90.2 ± 0.4	84.6 ± 0.6	86.5 ± 0.7
	R^M	83.6 ± 1.2	90.2 ± 0.4	84.0 ± 0.8	85.9 ± 0.8
APPROX	P^M	84.8 ± 1.5	90.7 ± 0.5	85.0 ± 1.0	86.9 ± 1.0
	R^M	83.9 ± 1.6	90.6 ± 0.5	84.3 ± 0.7	86.3 ± 0.9

Each row indicates the results for all the situations considered, as defined in Table 2

Table 6 Average results in terms of the *estimated profit per prototype* metric and standard deviation σ

Case study		NIST	MNIST	USPS	Avg.
State of the art					
	ALL	0.021 ± 0.000	0.012 ± 0.000	0.013 ± 0.000	0.015 ± 0.000
	SET	0.089 ± 0.002	0.076 ± 0.001	0.059 ± 0.001	0.075 ± 0.001
Our method					
	SET- APPROX	**0.091 ± 0.001**	0.079 ± 0.006	0.060 ± 0.001	0.077 ± 0.003
	APPROX- SET	**0.091 ± 0.002**	**0.080 ± 0.001**	**0.062 ± 0.001**	0.077 ± 0.001
	APPROX	**0.091 ± 0.002**	**0.080 ± 0.003**	**0.062 ± 0.002**	**0.078 ± 0.002**

The figures have been augmented by two orders of magnitude in order to facilitate the analysis of the results
The values in bold type show the best results for each corpus and the average results



(a) NIST **(b)** MNIST **(c)** USPS

Fig. 3 Examples of elements generated for each class of the corpora considered

makes it possible to obtain great efficiency and performance, but its results are masked by the other approaches considered when focusing on reduction capabilities. Nevertheless, it improves the baseline method in terms of efficiency and the reduction factor.

All of the above allows us to conclude that the approximate-based methods—SET- APPROX, APPROX- SET and APPROX—make it possible to surpass the efficiency of the set-median RHC adaptation, with SET- APPROX being the best as regards performance and APPROX being the best as regards efficiency.

To complement the results, Fig. 3 provides a graphic example of the strings generated with the approximated-median approach considered for each class of the corpora used in the experimentation. It can be appreciated that the elements generated are visually similar to the class to which they belong. Furthermore, an example of the retrieval of an approximated element from a class-homogeneous set of data using the algorithm considered is shown in Fig. 4. Note that the element generated has been slightly modified with respect to the set-median element, thus minimizing the dissimilarity value between the new artificial element and all the elements involved. Note also that if this algorithm is used in RHC, the four elements in the example would be replaced with the new element generated in order to reduce the size of the corpus considered.

To conclude this analysis, a statistical significance test has been performed on these results. More precisely, we have considered the Wilcoxon rank-sum test (Demšar 2006) in order to exhaustively compare all the situations considered in a pair-wise fashion, i.e., the results obtained with all the reduction cases, along with the case in which no reduction is performed. We also considered the *estimated profit per prototype* to be a figure of merit for this analysis, as it correctly summarizes both the classification rate and the set size reduction in a single value. Since the idea is to obtain a general remark, this analysis is not of any of the particular corpora considered. The individual results obtained for each fold, corpus, and reduction scenario, therefore, constitute each of the samples of the distributions to be compared. Table 7 presents the results obtained when considering a significance level of p-value < 0.01.

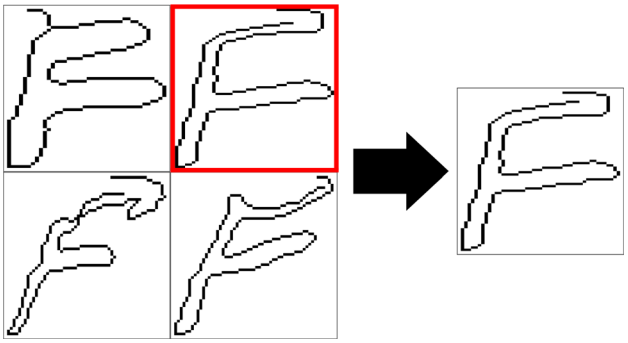


Fig. 4 Example of approximate generation from a cluster of elements belonging to the 'F' class in the NIST corpus. In the example, the algorithm begins with the set-median element—shown in red—and the string sequence is slightly modified in order to minimize the dissimilarity with all the elements involved

The results of this statistical analysis show that all the solutions proposed in this work—SET- APPROX, APPROX- SET and APPROX—achieve significantly better results than both baselines considered—the SET strategy. However, note that for the p-value considered, these proposed methods are not significantly different from each other. Finally, as expected, all reduction scenarios significantly improve the ALL case, as no type of size optimization is carried out in this scenario.

As a final note, the different experiments carried out support the initial premise that an approximate median string value is better suited to the RHC reduction strategy than the reference set-median proposed in the literature. In this regard, and as commented on throughout the manuscript, while the retrieval of the exact median in the string space might further improve the results in terms of the reduction in and representativeness of the data, this process constitutes an

Table 7 Results according to the Wilcoxon rank-sum test when comparing the different scenarios considered under the average *estimated profit per prototype* figure of merit

Case study	*State-of-the-art*		*Our method*		
	ALL	SET	SET- APPROX	APPROX- SET	APPROX
State of the art					
ALL	–	✗	✗	✗	✗
SET	✓	–	✗	✗	✗
Our method					
SET- APPROX	✓	✓	–	=	=
APPROX- SET	✓	✓	=	–	=
APPROX	✓	✓	=	=	–

Symbol ✓ shows that the row method is a significant improvement over the column case. Symbol ✗ highlights that the result achieved by the approach in the row is significantly lower than that in the column. Finally, = is used to highlight that results obtained by the row and the column methods are relatively similar. A significance level of $p < 0.01$ has been considered for the analysis

NP-complete problem (Calvo-Zaragoza et al. 2017a), and it is not, therefore possible to consider it in a practical scenario.

7 Conclusion

While the *k*NN rule constitutes one of the best-known distance-based classifiers, it is generally associated with low efficiency figures when tackling scenarios in which there are large amounts of data and computationally expensive dissimilarity metrics. This work addresses this issue when considering structural data representations and, more precisely, string codifications. Taking the work by Valero-Mas and Castellanos (2020) as a reference, since it proposes an adaptation to the string domain of the data reduction (DR) algorithm reduction through homogeneous clusters (RHC) originally devised for feature-based representations, we further explore several strategies in an attempt to further improve the results reported. More precisely, our hypothesis is that replacing clusters of same-class elements with the exact median would be more appropriate that using the set-median strategy proposed in the reference work.

Nevertheless, since the calculation of the exact median of a collection of string data is deemed an NP-hard problem (Calvo-Zaragoza et al. 2017a), it would appear necessary to consider alternative mechanisms that represent an intermediate solution between the existing set-median computation and the unfeasible exact computation. With that premise in mind, in this paper, we present a new adaptation of the RHC algorithm for DR when considering string data, which relies on the computation of an approximated version of the median string of the distribution as a possible solution to the constraints mentioned.

This proposal has been assessed by carrying out a set of experiments comprising several string data corpora, figures of merit, and RHC configurations. The results obtained corroborate the hypothesis that the use of an approximate median strategy makes it possible to obtain a more representative and reduced set of data, which leads to an improvement in terms of both size reduction and classification performance when compared to the baseline case. On average, we observed that our proposal achieves a maximum *F-measure* of 87.9%, whereas that attained by the state-of-the-art method is 86.7%. Although this figure could be considered as a slight improvement as regards performance, it must be pointed out that there is a narrow margin of improvement given that the exhaustive *k*NN approach achieves a base classification rate of 91.2%. An important reduction in the original corpora will also be noted, given that, in the best-case scenario, 18.6% of the original set size is used, whereas the reference set-median RHC method obtains a figure of 19.4%. Again, while this improvement may be considered as reduced, it must be noted that the high complexity of the Euclidean distance used by the *k*NN classifier for string data justifies the use of our proposal.

All of the above reasons allow us to conclude that the use of the approximate median operator rather than the set-median algorithm is beneficial for problems involving the *k*NN classifier in string data scenarios. As mentioned previously, our proposal not only provides an improvement regarding the scheme efficiency (a greater reduction capacity), but also provides superior effectiveness figures (a higher performance) when compared to those of the state of the art.

Future work will consider the adaptation of other DR feature-based algorithms to the string space. Moreover, a study of other alternative methods with which to retrieve an approximate median value from a set of string data could provide relevant insights into the relevance of this stage. Finally, we consider that this work could also be extended in order to consider other structural codifications such as trees or graphs.

Author Contributions F.J.C., J.J.V.-M. and J.C.-Z. made equally contributions as regards the conception of the work, the experimental work, the data analysis, and writing the paper.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This research work was partially funded by "Programa I+D+i de la Generalitat Valenciana" through grants ACIF/2019/042 and APOSTD/2020/256, the Spanish Ministry through HISPAMUS project TIN2017-86576-R, partially funded by the EU, and the University of Alicante through project GRE19-04.

Availability of data and material Data are available from the authors upon request.

Declarations

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

Code availability Not applicable

Ethical approval This paper contains no cases of studies with human participants performed by any of the authors.

References

Abdel-Hamid O, Mohamed AR, Jiang H, Penn G (2012) Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In: 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4277–4280

Abreu J, Rico-Juan JR (2014) A new iterative algorithm for computing a quality approximate median of strings based on edit operations. Pattern Recogn Lett 36:74–80

Alcalá-Fdez J, Sánchez L, Garcia S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM et al (2009) Keel: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput 13(3):307–318

Bille P (2005) A survey on tree edit distance and related problems. Theoret Comput Sci 337(1–3):217–239

Bunke H, Riesen K (2012) Towards the unification of structural and statistical pattern recognition. Pattern Recogn Lett 33(7):811–825

Calvo-Zaragoza J, Rizo D, Iñesta JM (2016) Two (note) heads are better than one: pen-based multimodal interaction with music scores. In: Proceedings of the 17th international society for music information retrieval conference (ISMIR). New York City, pp 509–514

Calvo-Zaragoza J, Oncina J, de la Higuera C (2017a) Computing the expected edit distance from a string to a probabilistic finite-state automaton. Int J Found Comput Sci 28(05):603–621

Calvo-Zaragoza J, Valero-Mas JJ, Rico-Juan JR (2017b) Prototype generation on structural data using dissimilarity space representation. Neural Comput Appl 28(9):2415–2424

Calvo-Zaragoza J, Castellanos FJ, Vigliensoni G, Fujinaga I (2018) Deep neural networks for document processing of music score images. Appl Sci 8(5):654

Chakraborty D, Das D, Krauthgamer R (2021) Approximating the median under the ulam metric. In: Proceedings of the 2021 ACM-SIAM symposium on discrete algorithms (SODA). SIAM, pp 761–775

Ciregan D, Meier U, Schmidhuber J (2012) Multi-column deep neural networks for image classification. In: Computer vision and pattern recognition (CVPR), 2012 IEEE conference on, IEEE. pp 3642–3649

Cover TM, Hart PE (1967) Nearest neighbor pattern classification. Inf Theory IEEE Trans 13(1):21–27

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Duda RO, Hart PE, Stork DG (2012) Pattern classification. Wiley, New Jersey

Duin RP, Pekalska E (2012) The dissimilarity space:bridging structural and statistical pattern recognition. Pattern Recogn Lett 33(7):826–832

Fischer I, Zell A (2000) String averages and self-organizing maps for strings. In: Proceedings of the second ICSC symposium on neural computation (NC'2000)

Freeman H (1961) On the encoding of arbitrary geometric configurations. IRE Trans Electron Comput 2:260–268

Gao X, Xiao B, Tao D, Li X (2010) A survey of graph edit distance. Pattern Anal Appl 13(1):113–129

García S, Luengo J, Herrera F (2015) Data preprocessing in data mining. In: Intelligent systems reference library

Hinarejos CDM (2003) La cadena media y su aplicación en reconocimiento de formas. PhD thesis, Universitat Politècnica de València

Hull JJ (1994) A database for handwritten text recognition research. IEEE Trans Pattern Anal Mach Intell 16(5):550–554

Kaysar MS, Khan MI (2020) A modified median string algorithm for gene regulatory motif classification. Symmetry 12(8):1363

Kohonen T (1985) Median strings. Pattern Recogn Lett 3(5):309–313. https://doi.org/10.1016/0167-8655(85)90061-3

Kruskal JB (1983) An overview of sequence comparison: time warps, string edits, and macromolecules. SIAM Rev 25(2):201–237

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl 10(8):707–710

McVicar M, Santos-Rodríguez R, Ni Y, De Bie T (2014) Automatic chord estimation from audio: a review of the state of the art. IEEE/ACM Trans Audio Speech Lang. Process. (TASLP) 22(2):556–575

Mirabal P, Abreu J, Seco D (2019) Assessing the best edit in perturbation-based iterative refinement algorithms to compute the median string. Pattern Recogn Lett 120:104–111

Mitchell TM (1997) Machine learning. McGraw-Hill, New York

Nanni L, Lumini A (2011) Prototype reduction techniques: a comparison among different approaches. Expert Syst Appl 38(9):11820–11828

Nicolas F, Rivals E (2005) Hardness results for the center and median string problems under the weighted and unweighted edit distances. J Discrete Algorithms 3(2–4):390–415

Ougiaroglou S, Evangelidis G (2016) Rhc: a non-parametric cluster-based data reduction for efficient *k*-nn classification. IEEE Trans Pattern Anal Appl 19(1):93–109

Plamondon R, Srihari SN (2000) Online and off-line handwriting recognition: a comprehensive survey. IEEE Trans Pattern Anal Mach Intell 22(1):63–84

Rico-Juan JR, Micó L (2003) Comparison of AESA and LAESA search algorithms using string and tree edit distances. Pattern Recogn Lett 24(9):1427–1436

Rico-Juan JR, Valero-Mas JJ, Calvo-Zaragoza J (2019) Extensions to rank-based prototype selection in k-nearest neighbour classification. Appl Soft Comput 85:105803. https://doi.org/10.1016/j.asoc.2019.105803

Riesen K, Schmidt R (2019) Online signature verification based on string edit distance. Int J Doc Anal Recogn 22(1):41–54

Triguero I, Derrac J, Garcia S, Herrera F (2012) A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Trans Syst Man Cybern Part C Appl Rev 42(1):86–100

Valero-Mas JJ, Castellanos FJ (2020) Data reduction in the string space for efficient knn classification through space partitioning. Appl Sci 10(10):3356

Valero-Mas JJ, Calvo-Zaragoza J, Rico-Juan JR (2016) On the suitability of prototype selection methods for knn classification with distributed data. Neurocomputing 203:150–160

Wilkinson RA (1992) The first census optical character recognition system conference, vol 4912. US Department of Commerce, National Institute of Standards and Technology

Yang L, Zhu QS, Jinlong H, Wu Q, Cheng D, Hong X (2019) Constraint nearest neighbor for instance reduction. Soft Comput. https://doi.org/10.1007/s00500-019-03865-z