Article

Data Reduction in the String Space for Efficient kNN Classification Through Space Partitioning

Jose J. Valero-Mas [1],*,† and Francisco J. Castellanos [2],†

[1] Independent Scientist, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain
[2] Pattern Recognition and Artificial Intelligence Group, Department of Software and Computing Systems, University of Alicante, 03690 Alicante, Spain; fcastellanos@dlsi.ua.es
* Correspondence: jjvalero@dlsi.ua.es
† These authors contributed equally to this work.

Abstract: Within the Pattern Recognition field, two representations are generally considered for encoding the data: *statistical* codifications, which describe elements as feature vectors, and *structural* representations, which encode elements as high-level symbolic data structures such as strings, trees or graphs. While the vast majority of classifiers are capable of addressing statistical spaces, only some particular methods are suitable for structural representations. The *k*NN classifier constitutes one of the scarce examples of algorithms capable of tackling both statistical and structural spaces. This method is based on the computation of the dissimilarity between all the samples of the set, which is the main reason for its high versatility, but in turn, for its low efficiency as well. Prototype Generation is one of the possibilities for palliating this issue. These mechanisms generate a reduced version of the initial dataset by performing data transformation and aggregation processes on the initial collection. Nevertheless, these generation processes are quite dependent on the data representation considered, being not generally well defined for structural data. In this work we present the adaptation of the generation-based reduction algorithm Reduction through Homogeneous Clusters to the case of string data. This algorithm performs the reduction by partitioning the space into class-homogeneous clusters for then generating a representative prototype as the median value of each group. Thus, the main issue to tackle is the retrieval of the median element of a set of strings. Our comprehensive experimentation comparatively assesses the performance of this algorithm in both the statistical and the string-based spaces. Results prove the relevance of our approach by showing a competitive compromise between classification rate and data reduction.

Keywords: string space; data reduction; *k*-Nearest neighbor; prototype generation

1. Introduction

In Pattern Recognition (PR), supervised classification is defined as the task of predicting the label of a given element out of a discrete set of categories based on the knowledge extracted from other labeled samples [1]. This discipline is largely applied in a wide variety of disciplines such as optical text or music recognition [2,3], audio analysis [4], speech recognition [5] and image categorization [6], among many others.

One of the crucial points in classification tasks is the representation considered for encoding the data. In this regard, two paradigms are typically differentiated in the literature [7]: on the one hand, the so-called *statistical representations* represent the data as vectors of numerical descriptors which describe each element based on a collection of features; on the other hand, *structural representations* consider powerful and flexible high-level symbolic data structures for representing the data, such as strings, trees or graphs. Thus, statistical representations show the clear advantage of being addressable by most classification algorithms while structural codifications generally exhibit superior accuracy

rates but are only handled by a reduced set of algorithms [8], mainly distance-based classifiers as, for instance, the *k*-Nearest Neighbor rule or Support-Vector Machines (SVM) since they only require the definition of a dissimilarity measure among the data points [9]. For instance, works such as the one by Riesen and Schmidt [10] for signature verification or the contribution by Calvo-Zaragoza et al. [11] for handwritten music symbols recognition state the adequacy of structural approaches, and more precisely string codifications, over other data representations.

As one of the most well known distance-based classifiers, the *k*-Nearest Neighbor (*k*NN) algorithm [12] is widely used in PR due to its conceptual simplicity and theoretical low errors properties [13]. This classifier assigns the most common label among the *k* closest elements to the input query obtained by performing pairwise dissimilarities to all the elements in the training set without deriving a classification model (lazy learning). Thus, unlike other distance-based classifiers as SVM, the *k*NN algorithm is usually related to low efficiency figures since the entire training data must be queried for classifying a new element [14]. Furthermore, this issue is especially noticeable in the structural space since dissimilarity metrics are generally quite complex and time-consuming. Therefore, distance-based classifiers like the *k*NN rule require strategies to reduce the complexity and cost of their computations, and particularly in structural domains such as string data.

Data Reduction (DR) is one of the main proposals for overcoming the efficiency issues inherent to *k*NN [15]. This family of methods aims at obtaining a reduced set of the original training data so that the time consumption is severely reduced while the classification rate is, ideally, not affected. While several approaches have been proposed in the literature, two particular strategies stand out in the literature [16]: (i) Prototype Selection (PS) methods, which reduce the initial data by selecting a subset out of it; and (ii) Prototype Generation (PG) methods, which generate a new set of data by means of performing some transformations on the initial one. In general, PG methods obtain higher reduction figures than PS, but their applicability is severely limited by the data representation considered since the transformations required are not as straightforward to define as in statistical codifications.

The difficulties associated with structural representations have hindered the development of PG methods as possible DR strategies for tackling the *k*NN efficiency issue in the structural space. Hence, most research efforts related to PG have been devoted to statistical representations. A relevant algorithm in this topic is Reduction through Homogeneous Clusters (RHC) by Ougiaroglou and Evangelidis [17], which reduces the initial set of data by obtaining same-class clusters of prototypes for then generating a new single prototype as the median value of the ones in the cluster. As most PG techniques, RHC was designed for statistical representations, being thus unsuitable for its application to structural data.

In this work, we aim at further studying the possibilities of PG as a DR strategy in structural representations due to their aforementioned relevance in the PR field. More precisely, we present an adaptation of the state-of-the-art RHC method by Ougiaroglou and Evangelidis [17] to the string space. As aforementioned, this algorithm replaces same-class subsets of prototypes by new elements generated by estimating their median value. Thus, the main issue to tackle is the actual retrieval of the median value of a group of strings, which in our case we resort to the set median as the calculus of the exact median string constitutes an NP-hard problem [18]. Additionally, in order to compare the performance of RHC strategy in both statistical and structural spaces, we make use of the Dissimilarity Space (DS) technique [19] to map the initial strings representation onto a feature-based codification so that additional conclusions can be gathered.

The rest of the work is structured as follows: Section 2 introduces the general background of the work; Section 3 presents the adaptation of the RHC algorithm to the string space; Section 4 explains the evaluation methodology proposed; Section 5 shows and discusses the results obtained; finally, Section 6 concludes the work and proposed future work to be addressed.

2. Background in Prototype Generation for Efficient Nearest Neighbour Classification

The lack of efficiency constitutes one of the main issues in the *k*NN rule as it relies on comprehensively consulting the entire training set for every query. In this regard, several strategies have been posed to palliate this drawback, which are generally divided into three categories [20]: (i) Fast Similarity Search, which proposes the creation of search indexes for a fast set consulting; (ii) Approximate Search, which works on the premise of retrieving sufficiently similar prototypes to a given query in the training set instead of exhaustively searching for the exact ones; and (iii) Data Reduction, which seeks for a reduced version of the training set without significantly altering its classification rate. In this work we focus on the latter family of methods, and more precisely on the so-called Prototype Generation strategies.

Prototype Generation (PG) stands for the particular type of DR processes which generate an alternative training set by applying certain transformations on the original training data [15]. The premise behind these methods is that, given a certain data corpus to be reduced, the most adequate elements to properly summarize it may not be among the existing prototypes, but they could be generated using some type of data aggregations. Thus, the main issue here is the definition of the generation policy.

According to Triguero et al. [21], PG strategies are broadly divided into four categories depending on the mechanism considered to obtain the reduced set:

- Class relabeling: This family of mechanisms considers that certain elements may be mislabelled due to tagging errors or noise in the data, being thus necessary to modify their categories by following certain criteria. Note that, while generalization accuracy is generally improved, no reduction is achieved.
- Centroid-based: These techniques divide the training data into different subsets, mainly resorting to proximity criteria, for then obtaining their centroids, which constitute the generated prototypes.
- Position adjustment: Methods belonging to this case modify the training set altering the features of the prototypes to reallocate them with the aim of improving the success rate. Given that this adjustment does not report any size reduction, these methods are generally paired with an initial reduction process.
- Space partitioning: This strategy divides the space into different regions for then generating one or more representative elements from each of them. While each of these partitions may contain one or more prototypes, the generated elements are not necessarily derived from them since the actual premise is to somehow represent the space partition independently of the data.

Note that some of the presented PG mechanisms imply operations on the data itself as, for instance, retrieving the centroid of a set of instances or modifying them in some sense. While such processes are relatively straightforward to apply in the case of feature-based data, in structural representations this point arises as an important issue. This fact limits the application of PG in classification tasks involving these latter data representations. Nevertheless, since structural representations have been proved as being the most suitable choice for some particular classification tasks [10,11], the issue of adapting such reduction strategies for this data codifications arises a relevant research problem.

In this paper, we present the adaptation of a space partitioning PG method to the case of string data, more precisely the Reduction through Homogeneous Clusters. This approach performs the reduction by recursively applying a clustering process on the initial data until a set class-homogeneous groups are retrieved for then generating a representative element out of each cluster as the median value of the elements in it. Thus, in our structural version of the RHC, this process implies obtaining of the median value of a set of strings. Nevertheless, given that the estimation of the median string constitutes an intricate problem by itself due to its high complexity [22], in this case we resort to the use of the set median as a means of retrieving the median value of a string data distribution.

3. Reduction Through Homogeneous Clusters in the String Space

The Reduction through Homogeneous Clusters algorithm proposed by Ougiaroglou and Evangelidis [17] stands out as one of the most recent proposals for PG based on a space partitioning premise for feature data. This method basically works in two different phases:

1. The partitioning phase in which the space is divided into a set of regions comprising one or more prototypes each one. These regions are obtained by means of applying a clustering process to the initial data until reaching a class homogeneity, i.e., the prototypes associated to a region share the same class. This process is typically carried out using a k-means process [1].
2. For each of those class-homogeneous clusters, a new single is derived as a combination of the prototypes associated to the cluster. Typically, this process is conducted by a feature-wise mean or median operation.

As commented, one of the main contributions of this work is the adaptation of the RHC algorithm to the case of structural representations, and more precisely to string data, which is now presented. Note that several considerations must be taken into account due to the particularity of the data representation, since the original algorithm is only designed for processing statistical data.

Let us denote initial set of instances $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{T}|}$ where x_i represents the i-th prototype from a given structural space $\mathcal{X}$ and y_i stands for its associated label belonging to the set of possible categories $\mathcal{Y}$. Let us also denote $\zeta(\cdot)$ as the function that retrieves the class associated to instance x_i, i.e., $\zeta(x_i) = y_i \in \mathcal{Y}$. Finally, consider $d(\cdot)$ as a dissimilarity measure in space $\mathcal{X}$. The string-based RHC algorithm retrieves a reduced version $\mathcal{R}$ out of the initial set $\mathcal{T}$ using Algorithm 1.

Algorithm 1 Reduction through Homogeneous Clusters.

1: **function** RHC($\mathcal{T}$)   ▷ Initial set $\mathcal{T} = t_1 \dots t_{|\mathcal{T}|}$
2:     $\mathcal{R}, \mathcal{C} \leftarrow \varnothing$
3:     **for each** $y \in \mathcal{Y}$ **do**
4:        $\mathcal{V} \leftarrow \{t_i \in \mathcal{T} \;\; : \;\; \zeta(t_i) = y\}$ ▷ Class-homogeneous grouping
5:        $\mathcal{C} \leftarrow \mathcal{C} \cup \text{set-median}(\mathcal{V})$
6:     **end for**
7:     **for each** $c \in \mathcal{C}$ **do**
8:        $\mathcal{S} \leftarrow \{t_i \in \mathcal{T} \;\; : \;\; c = \arg\min_{c' \in C} \; d(t_i, c')\}$ ▷ $\mathcal{S}$: Set of prototypes in cluster c
9:        **if** $|\{\zeta(t_i) \;\; : \;\; t_i \in \mathcal{S}\}| > 1$ **then**   ▷ Cluster homogeneity as set cardinality
10:          $\mathcal{R} \leftarrow \mathcal{R} \cup \text{RHC}(\mathcal{S})$ ▷ Non-homogeneous cluster
11:        **else**
12:          $\mathcal{R} \leftarrow \mathcal{R} \cup \text{set-median}(\mathcal{S})$ ▷ Homogeneous cluster
13:        **end if**
14:     **end for**
15:     **return** $\mathcal{R}$ ▷ Reduced set $\mathcal{R} = r_1 \dots r_{|\mathcal{R}|}$
16: **end function**

The main consideration in this design is the actual computation of the median string. As it has been introduced, the retrieval of the exact median value of a set of strings is known to be a NP-hard problem [18]. Thus, in this case we consider the set-median operation due to its lower complexity. This process is the one denoted as set-median$(\cdot)$ in Algorithm 1 and is obtained following Equation (1).

$$\text{set-median}(\mathcal{S}) = \arg\min_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}} d(s, s') \tag{1}$$

where $d(\cdot)$ denotes the dissimilarity measure previously defined and $\mathcal{S}$ is the set of prototypes contained in a particular cluster.

4. Experimentation

Figure 1 shows the experimental scheme conceived for this work. The idea is to comparatively assess the performance of the RHC strategy in both the string and feature-based spaces. For that the initial string data undergo two different processes: on the one hand, the data is directly processed in the string space with our extended RHC proposal; on the other hand, the string data is mapped onto a feature-based representation by means of the Dissimilarity Space methodology for then being processed by the original RHC. The comparison between the results obtained in both representation spaces provides the main conclusions of the work.
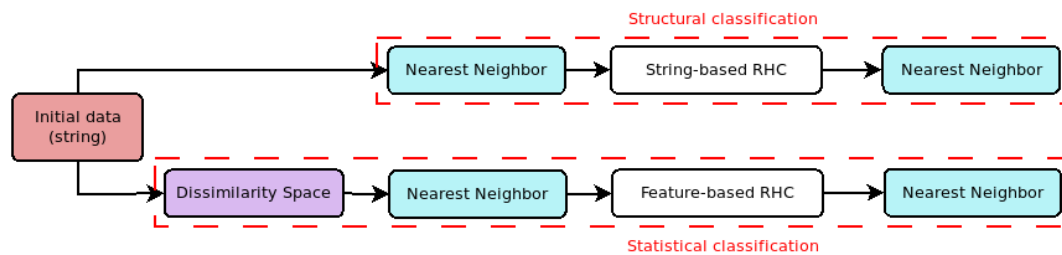

Figure 1. Description of the experimental setup considered. Dissimilarity Space is used for obtaining the feature-based representation of the initial string data. In both string and feature-based representations, classification is performed both before and after the RHC Data Reduction process.

As of dissimilarity metrics used in this experimentation, we have considered the well-known Edit Distance [23] for the string space. This measure defines the distance between two sequences of characters as the minimum number of modifications (insertions, deletions, or substitutions) required to transform one string into the other. Regarding the feature-based representation we have resorted to the Euclidan distance. In all the classification stages we have fixed the parameter $k = 1$ for the kNN classifier. Note that since each cluster obtained by the RHC algorithm is represented by a single element, there is no point in using other values for this parameter.

The rest of the section introduces the Dissimilarity Space process considered for mapping the string data to a statistical representation as well as the corpora and performance metrics used for the evaluation.

4.1. Dissimilarity Space Mapping

Dissimilarity Space stands for the strategy of mapping a given structural representation onto a feature-based one by computing pairwise dissimilarities between the elements of the dataset [19]. By performing this process a new dataset is derived in which the new features constitute the actual dissimilarity values to a subset of the data usually known as pivots. This process enables the use of certain data transformations which are unfeasible, or at least not well defined, in the structural space but it usually entails a drop in terms of classification accuracy due to the loss of representation capabilities.

Mathematically, let again $\mathcal{T}$ be a labeled set of prototypes such that $\mathcal{T} = \{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_{i=1}^{|\mathcal{T}|}$, where $\mathcal{X}$ and $\mathcal{Y}$ denote a structural space and a set of discrete classes, respectively. In order to map the prototypes of $\mathcal{T}$ onto a feature space $\mathcal{F}$, Dissimilarity Space methods seek for a subset $\mathcal{R}$ out of the training set ($R \subseteq T$) by following a certain policy. The elements of $\mathcal{R}$, which constitute the aforementioned pivots, are noted as r_i with $1 \leq i \leq |\mathcal{R}|$. Then, a prototype $x \in \mathcal{X}$ can be represented in $\mathcal{F}$ as a set of features $(v_1, v_2, v_3, ..., v_{|\mathcal{R}|})$ computed as $v_i = d(x, r_i)$, where $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ represents a dissimilarity function in space $\mathcal{X}$. This way, an $|\mathcal{R}|$-dimensional vector is obtained for each prototype in the initial space.

Among the existing policies for selecting the mapping pivots, in our experimentation we consider the use of the RandomC strategy [24]. This approach selects a random subset of prototypes from each class for the mapping process. The number of prototypes selected from each class is exactly c (tuning

parameter), being thus the total number of pivots $|R| = c|Y|$. In our experiments different values of parameter c are considered to compare its influence in the overall performance.

4.2. Corpora

Regarding the different corpora for the experimentation, we considered three different datasets of handwritten characters publicly available as images: the United States Postal Service (USPS) dataset of handwritten digits [25] whose images have a resolution of 16×16 pixels; the NIST SPECIAL DATABASE (NIST) of handwritten characters [26] in images of 28×28 pixels; and the MNIST collection of isolated handwritten digits [27] which is also distributed as images of 28×28 pixels.

In order to extract a string-based representation, all these images have undergone a process of contour extraction as the one described in [28] which is later encoded as a Freeman Chain Code [29]. Note that in no case we are claiming that this representation is the most suitable for these datasets but it allowed us to perform the considered experimentation. Figure 2 shows an example of this codification process.
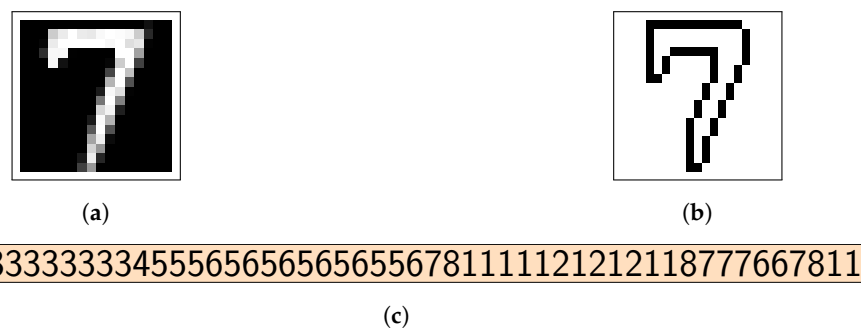

(a) (b)

33333333333455565656565656556781111121212118777667811

(c)

Figure 2. Example of a contour and string codification of a sample from the USPS corpus representing the digit 7. (**a**) Original image; (**b**) Extracted contour; (**c**) String representation using Freeman Chain Codes.

So as to obtain the feature-based representations we have resorted to the RandomC strategy previously introduced. Several values were tested to tune the c parameter of the number of pivots per class taking as a reference value the performance of the Nearest Neighbor classifier in the target space. Nevertheless, as no significant differences were observed, we considered the value of $c = 5$. This particular value constitutes the lowest one tested in our tuning experimentation and implies the least number of features in the statistical space.

As of data partitions, the sets have been distributed in five different folds maintaining the same class distribution as the complete set. It must be mentioned that, for a fair comparison between the two representation spaces, the different folds in both structural and statistical representation contain the exact same instances with the sole different of the encoding type considered.

Finally, Table 1 summarizes the details of the different data collections considered providing a description about the length of the string samples obtained after the aforementioned encoding processes from the images as well as the number of features after the RandomC stage.

Table 1. Description of the data collections considered in terms of number of instances, classes, and sample sizes in both the initial string and the Dissimilarity Space (feature-based) one using RandomC. *Min*, *Med*, and *Max* stand for the minimum, median, and maximum datum length while *Q1* and *Q3* denote the first and third quartiles, respectively.

Dataset	Instances	Classes	String Length					Feature Space Size
			Min	Q1	Med	Q3	Max	
NIST	5200	26	11	153	190	232	674	130
MNIST	10,000	10	15	53	59	69	115	50
USPS	8684	10	13	42	46	51	78	50

4.3. Performance Measurement

In order to assess and compare the different situations proposed in the experimentation, we evaluate both the classification rate achieved by each strategy as well as the resulting size of the dataset. Note that, in the context of a DR task, the former concept relates to the goodness of the reduction algorithm to extract a representative set of prototypes while the latter conceptis associated with the reduction capabilities of the strategy.

As of classification performance metric, we have considered the use of the *F-measure* (F_1) to avoid any bias towards any particular class in the case of a certain data imbalance. In a two-class classification task this metric is defined as a function of the successes and misclassifications of the algorithm as

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{2}$$

where TP, FP, and FN stand for *True Positives* or correctly classified elements, *False Positives* or type I errors, and *False Negatives* or type II errors, respectively.

It must be noted that, due to the non-binary nature of our datasets, we consider the use of the macro-averaged F_1 score which extends the definition of the binary F_1 to multiclass scenarios. This measure is defined as the average of the F_1 scores obtained for each class, that is:

$$F_1^M = \frac{1}{|\mathcal{Y}|} \cdot \sum_{i=1}^{|\mathcal{Y}|} F_1^{(i)} \tag{3}$$

where $\mathcal{Y}$ represents the set of classes in the task and $F_1^{(i)}$ the value of the F_1 metric for class $y_i \in \mathcal{Y}$.

Reduction capabilities have been assessed comparing the resulting set sizes of the training data in the different situations proposed. Computation time was discarded as evaluation metric due to its variability depending on the load of the computing system.

As commented, DR methods seek for simultaneously optimizing two contradictory goals, set size reduction and classification performance. Thus, it is not possible to retrieve a global optimum: some approaches will retrieve sharper reduction figures at the expense of a decrease of the classification rate while other will just show the opposite behavior.

In this sense DR can be addressed as a Multi-objective Optimization Problem (MOP) in which the two objectives to be optimised are the aforementioned reduction capabilities and classification performance. The possible solutions under this framework are usually retrieved resorting to 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, typically known as non-dominated elements, constitute the so-called Pareto frontier in which all elements are optimal solutions of the problem without any hierarchy among them.

Finally, in order to provide a single value which relates both the performance and reduction capabilities of the strategies considered, we also consider the *estimated profit per prototype* measure defined as the ratio between the classification accuracy and the total number of distances computed [30]. It must be mentioned that, for its use in this work, this metric was slightly adapted from its original definition by considering the F_1 instead of the classification accuracy as well as the resulting set size instead of the number of distances computed.

5. Results

This section presents the results obtained following the experimental methodology considered. Table 2 provides the figures achieved for each dataset in terms of the F_1 and resulting set size. Each value constitutes the average of the five folds considered in the cross-validation scheme. Note that, as previously mentioned, the idea is not to outperform the classification rates of this particular corpora but to proof the validity of the DR proposal for the string space.

Table 2. Results in terms of F_1 as classification rate and resulting set size for the different corpora considered. ALL and RHC stand for the classification results obtained with the initial set and the reduced one, respectively. Set size percentages are obtained referring to the ALL case for each corpus and data representation. Standard deviation values are not reported as they are lower or equal than 0.01. Bold values highlight the non-dominated elements for each dataset.

Dataset	Classification Performance (%)				Set Size (%)			
	String-Based		Feature-Based		String-Based		Feature-Based	
	ALL	RHC	ALL	RHC	ALL	RHC	ALL	RHC
NIST	**89.2**	**84.2**	82.5	77.2	**100**	**22.6**	100	27.2
MNIST	**94.1**	**90.7**	92.4	**88.6**	**100**	**15.0**	100	**10.3**
USPS	**90.2**	85.1	89.5	**84.7**	**100**	20.6	100	**12.8**

An initial remark to begin with is that the best classification rate for each dataset is obtained when addressing the case of the string-based data representation with no prototype reduction policy. This is actually the expected behavior as there is no information loss due to any reduction and/or mapping process. Also note that these cases report the lowest efficiency in the entire experimentation since not only the entire training set is used for the classification but also the actual computation of the Edit Distance for each pair of prototypes exhibits a remarkable time consumption by itself.

Once the RandomC procedure is applied, there is a clear drop in the classification rates compared to its counterpart in the string space. As it can be checked, the NIST dataset suffers the sharper drop going from a value of $F_1 = 89.2\%$ to $F_1 = 82.5\%$, roughly a 7%; USPS is the least affected one going from $F_1 = 90.2\%$ to $F_1 = 89.5\%$, just almost a 1% performance decrease; MNIST depicts an intermediate decrease figure of about 2% as it drops from $F_1 = 94.1\%$ to $F_1 = 92.4\%$. As in the previous case, given that no size optimization procedure is applied, the entire training set is again used for the classification task. Nevertheless, since this space considers the Euclidean distance, the process is faster than the string-based situation. n.

Having already introduced the cases related to the exhaustive search, we now discuss the results involving the RHC implementations. Considering the string space, once the proposed reduction method is applied, there is a decrease in the set sizes which are paired with a drop in the classification performance. More precisely, for all corpora the reduction figures are roughly between the 75% and 90% of the initial set size with performance drop close to the 5% with respect to the non-reduced case. This fact shows that this adaptation of the RHC algorithm is capable of properly dealing with the task given the remarkable reduction achieved with a very limited performance drop.

Focusing now on the equivalent situation in the statistical space, several points can be commented. As it may be checked, the performance decrease between the non-reduced and reduced feature-based cases is relatively similar to the same comparison in the string space (a classification rate drop between 3% and 5%). Note that, while these performance decreases refer to the dataset once it has been mapped onto the feature space, it must be reminded that this process entails a performance drop itself which can be avoiding if working in the original string space.

As of the reduction achieved by the RHC in this feature space, it is shown that the figures obtained are sharper than in the string representation both for the MNIST and the USPS sets with reduction figures close to 90%. Nevertheless, the reduction achieved by this algorithm in NIST set for string codifications is not as sharp as in the feature-based data.

Regarding the analysis in terms of the non-dominance criterion, the first point to comment is that the cases of non-reduced classification in the string space belong to the Pareto frontier for all corpora. This is a expected result given that they achieve the best classification performance at the expense of exhibiting the lowest efficiency among the cases considered. However, note that its statistical counterpart does not contain any element among the non-dominated elements. This point makes sense

since the non-reduced feature-based elements suffer a certain performance drop due to the mapping process which is not paired with any reduction in the set size. In spite of this, the cases based on RHC in statistical representation improve the reduction figures with respect to the structural space, hence obtaining non-dominated results for both MNIST and USPS feature-based corpora. Oppositely, the RHC method on string space has obtained better size-reduction level in NIST dataset, turning it into another non-dominated result.

Note that, depending on the corpus considered, the RHC-based non-dominating solutions which define the frontier are obtained from either the string space or the feature one, if not both. Concretely, the non-dominated solution in the NIST dataset is depicted by the case of the RHC in the string space whereas for the USPS corpus it is the RHC in the statistical one. The only case in which the RHC contributes to define the frontier in both spaces is with the MNIST dataset.

As a last analysis on the results, Table 3 presents the figures obtained in terms of the estimated profit per prototype metric. As the values retrieved are generally quite low, for an easier comparison all results have been scaled up two orders of magnitude.

Table 3. Results in term of the *estimated profit per prototype* metric. All values have been scaled up two orders of magnitude for simplifying the comparison. Bold values represent the best overall profit for each corpus. All results report deviation values lower than 0.01.

Dataset	String-Based		Feature-Based	
	ALL	RHC	ALL	RHC
NIST	0.021	**0.089**	0.020	0.068
MNIST	0.012	0.076	0.012	**0.110**
USPS	0.013	0.059	0.013	**0.096**

The first point that can be observed is that all reduced cases achieve better profit figures than the non-reduced ones. This fact depicts that, while both set size and classification rate decrease with the reduction process, the accuracy drop is less accused than the size one, which is the desired behavior.

Note that, while the best profit figures for both the MNIST and USPS datasets are obtained in the feature space, in the case of the NIST corpus this optimum is obtained in the string encoding. This fact of obtaining better results directly in the structural space without mapping the data onto a statistical space justifies the research efforts towards the development of reduction strategies in the native structural representation.

Finally, in order to statistically assess the figures obtained, a significance analysis has been performed on these results. More precisely we have considered the Wilcoxon rank-sum test [31] for comparing the results obtained before and after applying the RHC reduction algorithm for each representation space separately with a significance level of p-value < 0.05. As a figure of merit we have considered the estimated profit per prototype since it properly summarizes both performance and reduction capabilities of the strategy considered in one single value. Furthermore, since the idea is comparing the non-reduced and reduced scenarios, we are not particularizing in any of the corpora. Thus, the individual results obtained for each fold, corpus, and reduction scenario constitute each of the samples of the distributions to compare. The resulting data distribution is graphically shown in Figure 3.
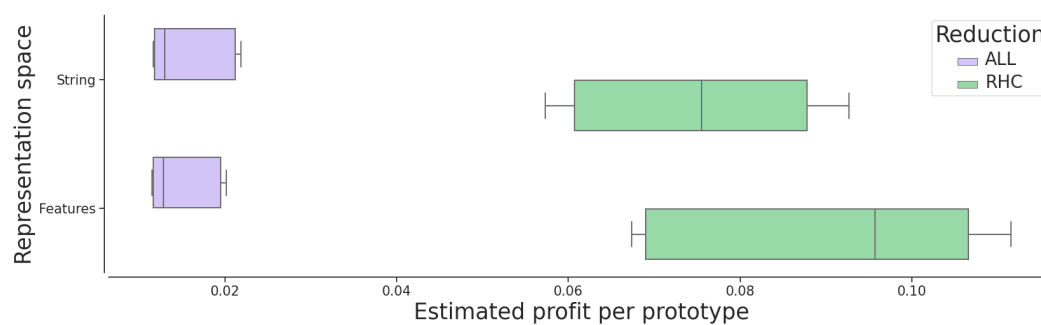
Figure 3. Boxplot graph of the data distributions considered the statistical analysis. Data is grouped according to the representation space considered and the reduction strategy performed (note that ALL stands for the case in which no reduction is performed). All values have been scaled up two orders of magnitude for an easier understanding of the plot.

The results of this statistical test state that, for both string and feature-based spaces, the profit per prototype figures obtained for the case in which no reduction is performed are significantly lower than the results obtained once the RHC method is applied. This fact is somehow graphically confirmed by Figure 3 as the RHC data distribution consistently achieves better profit figures than the non-reduced cases without any overlapping areas between them.

6. Conclusions and Future Work

Prototype Generation methods stand as one of the possible Data Reduction strategies for improving the efficiency of the *k*-Nearest Neighbor classifier. These methods perform transformations and combinations of the prototypes in the initial set of data with the aim of reducing its size and improving, if possible, the success rate of the classifier.

Despite its usefulness in the Pattern Recognition field, these methods are constrained by the representation space considered. Hence, while the aforementioned transformation and combination operations are straightforward to apply in the context of a statistical or feature-based space, their definition in the case of structural data requires some additional considerations.

In this work we present the adaptation of a state-of-the-art Prototype Generation strategy to the case of string data, more precisely the Reduction through Homogeneous Clusters method. This algorithm addresses the reduction process by recursively applying a clustering process on the initial data until a set of class-homogeneous groups is retrieved for then generating a single element from each cluster as the median value of the prototypes in it. In our case this problem has been tackled with the retrieval of the set median value the string data at issue.

The experimentation carried out compares the performance of the original with the proposed algorithm for the same data in both representation spaces using a Dissimilarity Space mapping process. Results obtained show that the relevance of the proposal as it avoids the representation gap of the mapping process as well as showing the best compromise between classification rate and data reduction.

In light of the results obtained, a first point which should be considered is the possibility of approximating the actual median string for the prototype generation stage instead of considering the set median value. In principle, the results presented in this work should constitute a lower bound in terms of performance. Thus, more sophisticated policies for the retrieval of the median string should report better representation capabilities and, hence, higher classification rates.

As noted, the retrieval of the median value is totally dependent on the distance considered. In this regard, the use of alternative string-based distance metrics as, for instance, the Cosine similarity, could report additional conclusions.

Another aspect to address is the adaptation of other generation-based Data Reduction strategies to the string space. While this work proves the validity and interest of adapting a feature-based reduction

method to this particular representation space, other strategies may obtain sharper reduction rates and with higher classification figures, thus being a promising path to explore.

In addition, extending these developments to other structural-based codifications as, for instance, tree structures or graphs also constitutes an important research challenge due to the complexity of the generation operations in these representations,

Finally, a last point to consider as future work is the application of these reductions string-reduction strategies as a preprocessing stage other dissimilarity-based algorithms as, for instance, Support Vector Machines or Nearest Class Mean rule, among others.

Author Contributions: J.J.V.-M. and F.J.C. equally contributed in the conception of the work, experimental, data analysis, and paper writing. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
2. Plamondon, R.; Srihari, S.N. Online and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 63–84. [CrossRef]
3. Calvo-Zaragoza, J.; Castellanos, F.J.; Vigliensoni, G.; Fujinaga, I. Deep neural networks for document processing of music score images. *Appl. Sci.* **2018**, *8*, 654. [CrossRef]
4. McVicar, M.; Santos-Rodríguez, R.; Ni, Y.; De Bie, T. Automatic chord estimation from audio: A review of the state of the art. *IEEE/ACM Trans. Audio Speech Lang. Process. (TASLP)* **2014**, *22*, 556–575. [CrossRef]
5. Abdel-Hamid, O.; Mohamed, A.; Jiang, H.; Penn, G. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In Proceedings of the 2012 IEEE international conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 4277–4280.
6. Ciregan, D.; Meier, U.; Schmidhuber, J. Multi-column deep neural networks for image classification. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3642–3649.
7. Calvo-Zaragoza, J.; Valero-Mas, J.J.; Rico-Juan, J.R. Prototype generation on structural data using dissimilarity space representation. *Neural Comput. Appl.* **2017**, *28*, 2415–2424. [CrossRef]
8. Bunke, H.; Riesen, K. Towards the unification of structural and statistical pattern recognition. *Pattern Recognit. Lett.* **2012**, *33*, 811–825. [CrossRef]
9. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 1997.
10. Riesen, K.; Schmidt, R. Online Signature Verification Based on String Edit Distance. *Int. J. Doc. Anal. Recognit.* **2019**, *22*, 41—54. [CrossRef]
11. Calvo-Zaragoza, J.; Rizo, D.; Iñesta, J.M. 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, NY, USA, 7–11 August 2016; pp. 509–514.
12. Cover, T.M.; Hart, P.E. Nearest neighbor pattern classification. *Inf. Theory IEEE Trans.* **1967**, *13*, 21–27. [CrossRef]
13. Calvo-Zaragoza, J.; Valero-Mas, J.J.; Rico-Juan, J.R. Improving kNN multi-label classification in Prototype Selection scenarios using class proposals. *Pattern Recognit.* **2015**, *48*, 1608–1622. [CrossRef]
14. Garcia, S.; Derrac, J.; Cano, J.; Herrera, F. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 417–435. [CrossRef]
15. García, S.; Luengo, J.; Herrera, F. *Data Preprocessing in Data Mining*; Intelligent Systems Reference Library: Berlin, Germany, 2015.
16. Nanni, L.; Lumini, A. Prototype reduction techniques: A comparison among different approaches. *Expert Syst. Appl.* **2011**, *38*, 11820–11828. [CrossRef]

17. Ougiaroglou, S.; Evangelidis, G. RHC: A non-parametric cluster-based data reduction for efficient *k*-NN classification. *IEEE Transcations Pattern Anal. Appl.* **2016**, *19*, 93–109. [CrossRef]

18. Calvo-Zaragoza, J.; Oncina, J.; de la Higuera, C. Computing the expected edit distance from a string to a probabilistic finite-state automaton. *Int. J. Found. Comput. Sci.* **2017**, *28*, 603–621. [CrossRef]

19. Duin, R.P.; Pękalska, E. The dissimilarity space: Bridging structural and statistical pattern recognition. *Pattern Recognit. Lett.* **2012**, *33*, 826–832. [CrossRef]

20. Rico-Juan, J.R.; Valero-Mas, J.J.; Calvo-Zaragoza, J. Extensions to rank-based prototype selection in k-Nearest Neighbour classification. *Appl. Soft Comput.* **2019**, *85*, 105803. [CrossRef]

21. Triguero, I.; Derrac, J.; Garcia, S.; Herrera, F. A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* **2012**, *42*, 86–100. [CrossRef]

22. Abreu, J.; Rico-Juan, J.R. A new iterative algorithm for computing a quality approximate median of strings based on edit operations. *Pattern Recognit. Lett.* **2014**, *36*, 74–80. [CrossRef]

23. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.

24. Duin, R.P.; Pękalska, E. *Dissimilarity Representation For Pattern Recognition, The: Foundations And Applications*; World Scientific: London, UK, 2005; Volume 64.

25. Hull, J.J. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 550–554. [CrossRef]

26. Wilkinson, R.A. *The First Census Optical Character Recognition System Conference*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 1992; Volume 4912.

27. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]

28. Rico-Juan, J.R.; Micó, L. Comparison of AESA and LAESA search algorithms using string and tree edit distances. *Pattern Recognit. Lett.* **2003**, *24*, 1427–1436. [CrossRef]

29. Freeman, H. On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Comput.* **1961**, *EC-10*, 260–268. [CrossRef]

30. Valero-Mas, J.J.; Calvo-Zaragoza, J.; Rico-Juan, J.R. On the suitability of Prototype Selection methods for kNN classification with distributed data. *Neurocomputing* **2016**, *203*, 150–160. [CrossRef]

31. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.