Neural Computing and Applications manuscript No.
(will be inserted by the editor)

Prototype Generation on Structural Data using Dissimilarity Space Representation

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

Received: date / Accepted: date

Abstract Data Reduction techniques play a key role in instance-based classification to lower the amount of data to be processed. Among the different existing approaches, Prototype Selection (PS) and Prototype Generation (PG) are the most representative ones. These two families differ in the way the reduced set is obtained from the initial one: while the former aims at selecting the most representative elements from the set, the latter creates new data out of it. Although PG is considered to delimit more efficiently decision boundaries, the operations required are not so well defined in scenarios involving structural data such as strings, trees or graphs. This work studies the possibility of using Dissimilarity Space (DS) methods as an intermediate process for mapping the initial structural representation to a statistical one, thereby allowing the use of PG methods. A comparative experiment over string data is carried out in which our proposal is faced to PS methods on the original space. Results show that the proposed strategy is able to achieve significantly similar results to PS in the initial space, thus standing as a clear alternative to the classic approach, with some additional advantages derived from the DS representation.

Keywords kNN classification · Prototype Generation · Structural Pattern Recognition · Dissimilarity Space

1 Introduction

In the Pattern Recognition (PR) field, two fundamental approaches can be found depending on the model used for representing the data [12]: a first one, usually known as structural or syntactical, in which data is represented as symbolic data structures such as strings, trees or graphs; and a second one, known as statistical or feature representation, in which the representation is based on numerical feature vectors that are expected to sufficiently describe the actual input.

The election of one of these approaches has some noticeable implications and consequences: structural methods offer a wide range of powerful and flexible high-level representations, but only few PR algorithms and techniques are capable of processing them; statistical methods, in spite of being less flexible in terms of representation, depict a larger collection of PR techniques [5].

Independently of whether we use a structural or a feature representation, instance-based PR methods, for which the k-Nearest Neighbor rule (kNN) is the most representative, may be applied for classification tasks. Generally, these methods just require to work over a metric space, *i.e.*, that in which a distance between two

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

Jorge Calvo-Zaragoza
Department of Software and Computing Systems, University of Alicante
Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain
Tel.: +349-65-903772
E-mail: jcalvo@dlsi.ua.es

Jose J. Valero-Mas
Department of Software and Computing Systems, University of Alicante
Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

Juan R. Rico-Juan
Department of Software and Computing Systems, University of Alicante
Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

points can be defined. Instead of obtaining a set of classification rules out of the available information, they need to examine all the training data each time a new element has to be classified. As a consequence, they not only depict considerable memory requirements in order to store all these data, which in some cases might be a very large number of elements, but also show a low computational efficiency as all training information must be checked at each classification task [28].

Data Reduction techniques, a particular subfamily of Data Preprocessing methods, try to solve these limitations by means of selecting a representative subset of the training data [19]. Two common approaches for performing this task are Prototype Generation (PG) and Prototype Selection (PS). Both families of methods focus on reducing the size of the training set for lowering the computational requirements while maintaining, as far as possible, the classification accuracy. The former family creates new artificial data to replace the initial set while the latter one simply selects certain elements from that set.

It must be pointed out that the two aforementioned DR paradigms do not show the same dependency on the data representation used. PS algorithms have been widely used in both structural and feature representations as the elements are not transformed but simply selected. On the other hand, PG methods require to modify or create data in order to intelligently place new elements and, while this process can be easily performed in feature representations, it becomes remarkably difficult for structured data, at least in terms of developing a generic strategy for any type of data structure (e.g., strings, trees, or graphs).

In this paper we study the possibility of applying PG methods to structured representations by means of using Dissimilarity Space (DS) methods so as to solve the aforementioned obstacle. By using DS techniques, the initial structural representation can be mapped onto a feature-based one, thereby allowing the use of statistical PG techniques not available in the original space. Our intention is to assess whether this approach deserves further consideration when faced against the classical choice of applying PS in the initial structural space.

This paper expands the initial idea proposed in the work of Calvo-Zaragoza et al. [8] by providing a more far-reaching experimentation, in which a broader number of DS methods is considered. Stronger statements about the performance of the proposal are drawn, supported by a comprehensive evaluation in terms of number of datasets and statistical significance tests.

The rest of the paper is structured as it follows: Section 2 introduces the task of Data Reduction; Section 3 explains the idea of Dissimilarity Space and its application to our case; Section 4 describes the evaluation methodology proposed; Section 5 shows and thoroughly analyzes the results obtained; finally, Section 6 explains the general conclusions obtained and discusses possible future work.

2 Background on Data Reduction

Among the different stages which comprise the so-called Knowledge Discovery in Databases, Data Preprocessing is the set of tasks devoted to provide the information to the Data Mining system in the suitable amount, structure and format [25]. Data Reduction (DR), which constitutes one of these possible tasks, aims at obtaining a reduced set with respect to the original data which, if provided to the system, would produce the same output as the original data [19].

DR techniques are widely used in kNN classification as a means of overcoming its previously commented drawbacks, being the two most common approaches Prototype Generation (PG) and Prototype Selection (PS) [29]. Both methods focus on obtaining a smaller training set for lowering the computational requirements and removing ambiguous instances while keeping, if not increasing, the classification accuracy.

PS methods try to select the most profitable subset of the original training set. The idea is to reduce its size to lower the computational cost and remove noisy instances which might confuse the classifier. Typically, three main families can be considered based on the objective pursued during the process:

- **Condensing**: The idea followed by these methods is to keep only the most representative prototypes of each class and reduce as much as possible the data set. While accuracy on training set is usually maintained, generalization accuracy is lowered.
- **Editing**: These approaches focus on eliminating instances which produce some class overlapping, typical situation of elements located close to the decision boundaries or noisy data. Data reduction rate is lower than in the previous case but generalization accuracy tends to be higher.
- **Hybrid**: These algorithms look for a compromise between the two previous approaches, which means seeking the smallest data set while improving, or at least maintaining, the generalization accuracy of the former set.

Given its importance, many different approaches have been proposed throughout the years to carry out this task. The reader may check the work of Garcia et al. [18] for an extensive introduction to this topic as well as a comprehensive experimental comparison for the different methods proposed. Since trying to maintain the same accuracy as with the initial training set is difficult to fulfill in practical scenarios, much research has been recently devoted to enhance this process through the combination with other techniques. Some of these include Feature Selection [35], Ensemble methods [20] or modifications to the kNN rule [7].

On the other hand, PG methods are devoted to creating a new set of labeled prototypes that replace the initial training set. Under the DR paradigm, this new set is expected to be smaller than the original one since the decision boundaries can be defined more efficiently. Depending on the focus where placing the new prototypes, three main families of strategies can be found:

- **Centroid-based**: subsets of prototypes of the initial training set are grouped taking into account proximity, labeling and representation criteria. Then, the centroid of this subset is generated as a new prototype for the final set.
- **Position adjustment**: from an initial subset of the training set, selected following any strategy (for instance, a PS method), prototypes are moved around their neighborhoods following a particular heuristic. The objective is to find the location in which they can be more profitable for classification purposes.
- **Space partitioning**: the idea is to divide the input space into regions of interest. Then, representatives of each space are generated. Variations in space division and generation within each one provide the different methods of this family.

Reader is referred to the work of Triguero et al. [34] to find a further extension to this introduction to PG methods.

Under the same conditions, PG is expected to perform better than PS since the former can be seen as a generalization of the latter. Nevertheless, while PS only needs information about similarity or proximity between different prototypes, for which one can use the same dissimilarity function considered for the kNN rule, PG needs information about the representation space. Indeed, the PG family represents a more restrictive option than the simple selection of prototypes because it is hard to be used under structural spaces. In these cases, it is difficult to develop generic operations such as '*move a prototype towards a specific direction*' or '*find the centroid of a subset of prototypes*'. Thus, generating new prototypes in structural data is not a trivial matter.

Given the theoretical advantages of PG over PS methods, finding strategies to generate prototypes on structural data would be of great interest. In this work, it is proposed a method that fills this gap. It consists of a two-stage algorithm which first maps the structural data onto features vectors, after which common PG techniques can easily work. To perform this mapping, we resort here to the so-called Dissimilarity Space representation. Next section details our proposal.

3 Prototype Generation over Structural Data using Dissimilarity Space Representation

Current PG algorithms assume that data is defined over a vector space. Thus, it is feasible to perform geometric operations to find new points of interest in which new labeled prototypes can be generated. The intention is to maintain the accuracy of the kNN classifier with fewer prototypes than in the original training set.

Nevertheless, when working over a structural space, it is just known a distance function that allows knowing the proximity between two points of the space (this is also referred as metric space). In that case, PG algorithms are not able to generalize the geometric operations utilized in the vector space. Serve as an example the median operation: its computation is easy for n-dimensional points whereas it becomes NP-complete when points are strings [22]. Some examples of works addressing related issues include the work of Abreu and Rico-Juan [1], in which the median of a string set is approximated using edit operations, or Ferrer and Bunke [16], in which an iterative algorithm for the computation of the median operation on graphs is exposed. Nevertheless, all of them take advantage of the knowledge of the specific structural data to create these new prototypes. Therefore, generalization to other structural representations cannot be assumed.

We propose a new strategy as a possible solution to the problem stated above. The process itself follows a simple procedure which consists in mapping data onto a new vector, or feature, space. This process, known as *embedding*, has been extensively studied for decades [23, 4]. Once data is represented as feature vectors, conventional prototype generation strategies may be used.

In this work we are going to restrict ourselves to a particular family of embedding algorithms known as Dissimilarity Space (DS) representation [13]. Broadly, DS representations are obtained by computing pairwise dissimilarities between the elements of a representation set, which actually constitutes a subset of the initial structural training data selected following a given criterion.

The choice of using DS instead of other techniques is justified by some reasons directly related to the actual object of study:

1. It only requires a distance or dissimilarity function between prototypes. Taking into account that this work focuses on DR techniques for improving kNN classification — which also needs this function —, the requirement is assumed to be effortless.
2. The intention of the work is to measure the performance of PG on the new space. Therefore, it is preferable that results are more related to the PG technique instead of the quality of the embedding method. That is why it is considered a simple method (but with a strong background) rather than a more complex one.

During experimentation, the classification results obtained after applying a set of PG techniques to the DS representation will be compared to the results obtained when using PS techniques in the initial structural space so as to check whether our approach can be useful in these situations. On the other hand, below we introduce the DS transformation and the particular strategies considered.

3.1 Dissimilarity Space transformation

Let $\mathcal{X}$ denote a structural space in which a dissimilarity function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ is defined. Let Y represent the set of labels or classes of our classification task. Let T be a labeled set of prototypes such that $T = \{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in Y\}_{i=1}^{|T|}$.

In order to map the prototypes of T onto a feature space $\mathcal{F}$, DS-based methods seek for a subset R out of the training set ($R \subseteq T$). The elements of R, usually known as *pivots*, are noted as r_i with $1 \leq i \leq |R|$. Then, a prototype $x \in \mathcal{X}$ can be represented in $\mathcal{F}$ as a set of features $(v_1, v_2, v_3, \ldots, v_{|R|})$ such that $v_i = d(x, r_i)$. This way, an $|R|$-dimensional real-valued vector can be obtained for each point in the space $\mathcal{X}$. Different heuristics were proposed in the work of Pekalska et al. [30] for the selection of pivots, some of which have been considered for our work and are briefly described below.

3.1.1 RandomC

The RandomC strategy selects a random subset of prototypes, in which the number of prototypes of each class is exactly c (tuning parameter), that is, $|R| = c|Y|$. In order to compare the influence of parameter c in the feature representation, some different values will be considered at experimentation stage.

3.1.2 kCenters

This strategy performs a k-medoids clustering process on every class considered. The initialization is performed as proposed in the work of Arthur and Vassilvitskii [3] (k-means++). The different centroids obtained after the process are included in R, *i.e.*, $|R| = k|Y|$. As happened in the previous case, the value k may alter the representation of the new space so some tuning will be considered during the experimentation.

3.2 EditCon

The main idea behind EditCon is to select the most representative prototypes of the training set to be used as pivots. To this end, this technique applies two PS algorithms to the initial training set: as a first step, an Editing process [37] is used to remove noisy information; then, a Condensing process [21] is performed so as to keep only the informative elements. No parameters are considered in this case.

4 Experimentation

Figure 1 shows the implemented set-up for performing the experimentation. As it can be checked, out of the initial structural elements, a feature representation is obtained using a particular DS method. DR techniques are then applied to both data representations but, while PS methods are applied to structural and feature representations, PG is only performed on the latter. Finally, the Nearest Neighbor (NN) algorithm, parameterized with k=1, is used for the classification.

For these experiments, different configurations of the c and k parameters of the RandomC and kCenters, respectively, have been tested. The values considered have been 5, 10 and 15 prototypes per class.

We shall now describe the different datasets, Data Reduction strategies studied and the performance metrics considered for this study.

4.1 Datasets

Five different datasets of isolated symbols have been considered: the National Institute of Standards and Technology DATABASE 3 (NIST3), from which a subset of the upper case characters was randomly selected, the Mixed National Institute of Standards and Technology dataset (MNIST) [27] of handwritten digits, the United States Postal Office handwritten digits dataset (USPS) [24], the MPEG-7 shape silhouette dataset [26], and the Handwritten Online Musical Symbol (HOMUS) dataset [6]. In terms of class representation, these datasets can be considered as being totally balanced. Freeman

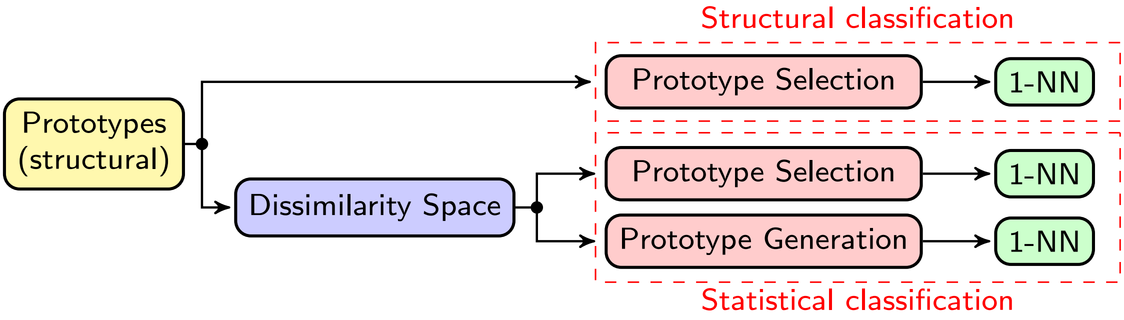

Fig. 1 Experimental set-up tested. DS is used for mapping structural data into a feature-based space. PS is applied to both structural and feature data while PG is only performed on the latter. 1-NN is used for the classification in all cases.

Name	Instances	Classes
NIST3	6500	26
MNIST	70000	10
USPS	9298	10
MPEG-7	1400	70
HOMUS	15200	32

Table 1 Description of the datasets used in the experimentation.

Chain Codes [17] have been considered as contour descriptors. Since this structural data is represented with strings, the well-known the Edit distance [36] is considered as dissimilarity. Once data is mapped onto feature vectors, the Euclidean distance is used.

A 5-fold cross-validation process has been applied for each dataset to examine the variance to the training data.

Reader is referred to Table 1 to find more details about the composition of the datasets.

4.2 Data Reduction strategies

A representative set of DR algorithms covering a wide range of selection variants was used for the experimentation. However, in order to perform a fair comparison between the two DR strategies, we are only showing the results for the PS algorithms retrieving similar size reductions to the PG algorithms. These techniques are briefly introduced in the following lines.

4.2.1 Prototype Selection (PS) algorithms

- Fast Condensing Nearest Neighbor (FCNN) [2]: computes a fast, order-independent condensing strategy based on seeking the centroids of each label.
- Farther Neighbor (FN) [31]: gives a probability mass value to each prototype following a voting heuristic based on neighborhood. Prototypes are selected according to a parameter (fixed to 0.3 in our case) that indicates the probability mass desired for each class in the reduced set.
- Cross-generational elitist selection, Heterogeneous recombination and Cataclysmic mutation algorithm (CHC) [14]: evolutionary algorithm commonly used as a representative of Genetic Algorithms in PS. The configuration of this algorithm has been the same as in [9].

This subset of techniques is expected to cover three typical searching methodologies of PS: FCNN as condensing, FN as heuristic approach and CHC as evolutionary search.

4.2.2 Prototype Generation (PG) algorithms

- Reduction by Space Partitioning 3 (RSP3) [32]: divides the space until a number of class-homogeneous subsets are obtained; a prototype is then generated from the centroid of each subset.
- Evolutionary Nearest Prototype Classifier (ENPC) [15]: performs an evolutionary search using a set of prototypes that can improve their local quality by means of genetic operators.
- Mean Squared Error (MSE) [10]: generates new prototypes using gradient descent and simulated annealing. Mean squared error is used as cost function.

The parameters of these algorithms have been established following the work of Triguero et al. [34]. As in the previous case, we try to consider a representative set of generation techniques: MSE as a classical method, ENPC as evolutionary search and RSP3 as heuristic approach.

4.3 Performance measurement

In order to assess the results, we have considered as metrics of interest the classification accuracy of the reduced set as well as its size. While the former indicates the ability of the DR method to choose the most relevant prototypes, the latter one depicts its reduction skills.

For these figures of merit we show the results obtained when averaging the scores for each dataset, which allows to understand the general performance of each scenario at a glance. Nevertheless, in order to perform a rigorous comparison among the strategies, a significance test has been performed facing accuracy and set size figures.

It must be considered that, although these measures are suitable to evaluate the performance of each single strategy, it is not possible to establish a clear comparison among the whole set of alternatives to determine the best one. DR algorithms aim at minimizing the number of prototypes considered in the training set while, at the same time, increasing the classification accuracy. Most often, these two goals are contradictory so improving one of them implies a deterioration of the other. From this point of view, classification in DR scenarios can be seen as a Multi-objective Optimization Problem (MOP) in which two functions have to be simultaneously optimized: reduction of the training set and maximization of the classification success rate. Usually, the evaluation of this kind of problems is carried out in terms of the *non-dominance* concept. One solution is said to dominate another if, and only if, it is better or equal in each goal function and, at least, strictly better in one of them. The set of non-dominated elements represents the different optimal solutions to the MOP. Each of them is usually referred to as Pareto optimal solution, being the whole set usually known as Pareto frontier.

Finally, classification time is also considered in this study to assess the influence of the type of data representation in these terms.

5 Results

Average results in terms of classification accuracy and set size obtained on the different datasets are presented in Table 2. Additionally, Table 3 shows the corresponding average classification times. Normalization (in %) is done with respect to the whole dataset. ALL refers to results obtained when using the whole training set (no DR algorithm is applied). Furthermore, Table 4 shows the average number of attributes obtained in each dataset when applying the different DS processes to the initial structural space.

For a better understanding, Figure 2 shows graphically the results in a 2D representation where accuracy and size are confronted. Non-dominant elements representing the Pareto frontier are highlighted.

A first initial remark is that, on average, the DS process implies a reduction in classification accuracy. For a given algorithm, when comparing the accuracy results obtained in the initial space with any of the corresponding DS cases, there is a decrease in these figures. For instance, when considering the ALL case, average classification accuracy goes from 90.8 % in the initial space to figures around 88 % in the different DS spaces considered, which is around a 3 % decrease in accuracy simply because of the mapping stage.

For both structural and feature-based representations, PS techniques depict a decrease in the classification accuracy when compared to the ALL case. This effect is a consequence of the reduction in the set size. In the DS space, however, PG achieves slightly better classification results with similar reduction rates than the PS algorithms, somehow showing the superior robustness of these methods. As an example, for RandomC(5), ENPC achieves an accuracy of 85.6 % with a set size of 15 % whereas 1-FN$_{0.3}$ roughly gets to a classification rate of 80.7 % with a 16.5 % of the initial set.

Nevertheless, the main outcome out of the results obtained by the PG algorithms is that the scores obtained in the DS space are quite similar to the ones obtained by PS schemes in the initial structural space. Although this point shall be later thoroughly assessed through statistical tests, these figures may allow us to qualitatively see the proposed strategy as a clear competitor of PS in structural data.

In terms of classification times, results show DS strategies as much faster than structural ones (several orders of magnitude) due to the complexity reduction achieved by using Euclidean distance instead of Edit distance.

Regarding the considered DS strategies, it can be checked that the results are not remarkably affected by the DS algorithm considered as neither accuracy values nor sizes show dramatic changes among them. In the same sense, parameters of RandomC and kCenters do not seem to have a remarkable influence either as figures obtained by the different configurations are very similar.

When considering the non-dominance criterion, we can see that most elements defining the Pareto frontier are PS configurations in the structural space, more precisely CHC, FCNN and the ALL configuration (see Fig. 2). When mapping to the statistical space, CHC extends the frontier as, despite its accuracy loss, it achieves remarkable reduction rates. Concerning our proposal of PG in the DS space, we can see that the different configurations fill some areas of the space where the rest of

			PS						PG					
	ALL		FCNN		1–FN$_{0.3}$		CHC		RSP3		ENPC		MSE	
	Acc	**Size**	**Acc**	**Size**	**Acc**	**Size**	**Acc**	**Size**	**Acc**	**Size**	**Acc**	**Size**	**Acc**	**Size**
No DS	90.8	100	87.9	22.0	84.6	13.9	81.0	3.6	-	-	-	-	-	-
RandomC(5)	87.4	100	84.1	26.8	80.7	16.5	75.5	3.6	85.8	31.5	85.6	15.0	83.3	14.3
RandomC(10)	88.0	100	84.5	26.2	81.3	16.5	75.8	3.3	86.2	31.0	86.0	14.4	85.5	14.4
RandomC(15)	88.2	100	84.9	26.0	81.6	16.5	76.6	3.3	86.7	31.3	86.1	14.3	84.1	14.4
kCenters(5)	87.9	100	84.1	26.4	81.2	16.5	76.6	3.4	86.2	30.2	85.9	14.8	83.8	14.3
kCenters(10)	88.0	100	84.5	26.1	81.1	16.5	76.7	3.6	86.6	30.9	86.2	14.4	84.2	14.4
kCenters(15)	88.3	100	85.1	25.7	81.4	16.5	77.0	3.5	86.7	30.7	86.3	14.1	84.2	14.4
EditCon	88.0	100	84.9	25.9	81.5	16.6	75.8	3.7	86.5	32.6	86.2	14.5	83.7	14.4

Table 2 Results obtained with the different DS algorithms configurations considered. Figures shown represent the average of the results obtained for each single dataset. No DS depicts results obtained in the initial structural space. Selection and generation techniques are regarded as PS and PG respectively. ALL stands for the case in which no selection or generation is performed. Normalization (%) is performed with respect to ALL case of each dataset separately.

		PS			PG		
	ALL	FCNN	1–FN$_{0.3}$	CHC	RSP3	ENPC	MSE
No DS	877.3	221.3	136.7	50.7	-	-	-
RandomC(5)	3.15	0.91	0.56	0.13	1.09	0.46	0.13
RandomC(10)	5.07	1.5	0.96	0.28	1.71	0.72	0.14
RandomC(15)	6.72	2.03	1.36	0.47	2.21	0.92	0.18
kCenters(5)	3.17	0.90	0.56	0.13	1.06	0.47	0.08
kCenters(10)	5.05	1.48	0.96	0.29	1.68	0.71	0.14
kCenters(15)	6.73	2.01	1.36	0.48	2.18	0.9	0.18
EditCon	20.75	7.2	5.39	2.92	6.53	2.48	0.39

Table 3 Average classification time (in seconds) for the different DS algorithms configurations considered. Figures shown represent the obtained when processing each single dataset. No DS depicts results obtained in the initial structural space. Selection and generation techniques are regarded as PS and PG respectively. ALL stands for the case in which no selection or generation is performed.

	RandomC(5)	RandomC(10)	RandomC(15)	kCenters(5)	kCenters(10)	kCenters(15)	EditCon
NIST3	130	260	390	130	260	390	520
MNIST	50	100	150	50	100	150	650
USPS	50	100	150	50	100	150	680
MPEG-7	350	700	1050	350	700	1050	210
HOMUS	160	320	480	160	320	480	1760
Average	148	296	444	148	296	444	764

Table 4 Number of features in the dissimilarity space for each DS algorithm and dataset.

the considered approaches do not have a relevant presence. It is also interesting to point out the presence of ENPC as part of the non-dominant elements set, thus remarking the interest of the strategy proposed in the paper.

Finally, some remarks can be done attending to the information in Table 4 regarding the number of attributes in the feature space for the datasets considered together with the general performance information in Table 2. As it can be seen, the election of a particular DS method implies a great difference in the number of attributes. For instance, RandomC(5) supposes a third of the number of attributes in RandomC(15) and around a seventh of the ones retrieved by the EditCon algorithm. Nevertheless, accuracy results (cf. Table 2) do not report a clear difference in the results. As an example, in the ALL situation, kCenters(5) and EditCon report a very similar average accuracy (around 80 %)

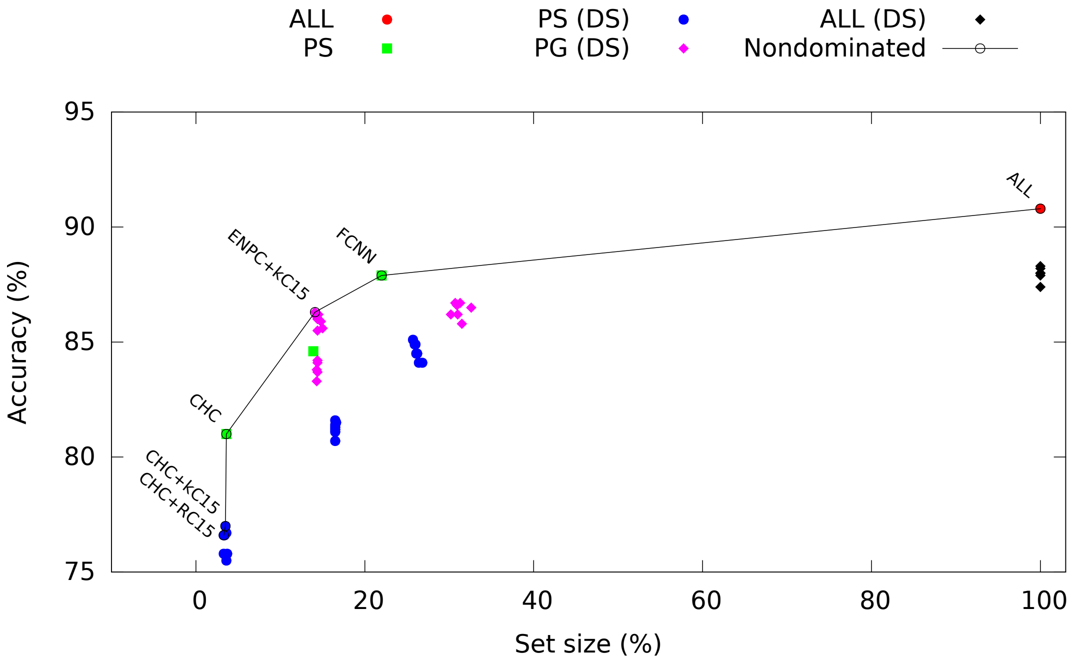

Fig. 2 Average results of the different configurations considered, facing accuracy and size of the reduced set. Non-dominated elements defining the Pareto frontier are highlighted.

but with a great difference in terms of number of attributes.

5.1 Statistical significance

As aforementioned, in order to statistically estimate the competitiveness of the proposed strategy, a Wilcoxon rank-sum test [11] has been performed. As we aim at assessing the competitiveness of using PG in DS spaces against PS in the initial space, accuracy and set size figures shall be compared. Table 5 shows the results of this test when considering a significance $p < 0.05$.

We note that PG strategies are not competitive in accuracy against the ALL case in the structural space as they achieve significantly lower classification rates. In terms of reduction, as expected, all PG strategies significantly outperform the ALL case, as the latter does not perform any kind of reduction.

When compared to the PS algorithms in the structural space, it can be checked that RSP3 does not achieve a remarkable reduction rate as set sizes are significantly higher than the ones in the initial space. However, regarding classification rate, RSP3 stands as a clear competitive algorithm as results are never significantly worse than the ones by the PS strategies.

The evolutionary algorithm ENPC achieves noticeable reduction rates as, except when compared to CHC, figures are significantly similar to, or even better than, the considered PS strategies. Classification rates are, in general, similar to the ones in PS except for the CHC algorithm, in which ENPC always shows a significant improvement, and some particular cases of FCNN in which ENPC shows a significant decrease.

MSE shows the poorest performance of the considered algorithms with respect to accuracy. This can be clearly seen when compared to the FCNN or the CHC cases in which the results of the tests are significantly worse than the ones of the other PG strategies. Although this poor performance could be due to a sharp reduction rate, this is not the case. For instance, if we check the ENPC and MSE cases with RandomC(10) against FCNN we can see that, while the former achieves accuracy results similar to the PS algorithm with a significantly lower set size, MSE shows worse classification results than the PS strategy with a similar set size.

5.2 Discussion

Experiments show that the performance of PG in the feature-based space seems to be somehow bounded by the DS mapping process: the PG configurations considered are capable of retrieving classification rates similar to the ones achieved when not performing data reduction in this new space; however, these figures are still far from the ones achieved in the original space without any reduction either. While this could be a particularity of a precise DS method, our experiments show that

PG	DS method	ALL		PS					
				FCNN		1-FN$_{0.3}$		CHC	
		Acc	**Size**	**Acc**	**Size**	**Acc**	**Size**	**Acc**	**Size**
RSP3	RandomC(5)	✗	✓	=	✗	=	✗	✓	✗
	RandomC(10)	✗	✓	=	✗	✓	✗	✓	✗
	RandomC(15)	✗	✓	=	✗	✓	✗	✓	✗
	kCenters(5)	✗	✓	=	✗	✓	✗	✓	✗
	kCenters(10)	✗	✓	=	✗	✓	✗	✓	✗
	kCenters(15)	✗	✓	=	✗	✓	✗	✓	✗
	EditCon	✗	✓	=	✗	✓	✗	✓	✗
ENPC	RandomC(5)	✗	✓	✗	=	=	=	✓	✗
	RandomC(10)	✗	✓	=	✓	=	=	✓	✗
	RandomC(15)	✗	✓	=	✓	=	=	✓	✗
	kCenters(5)	✗	✓	✗	=	=	=	✓	✗
	kCenters(10)	✗	✓	=	✓	=	=	✓	✗
	kCenters(15)	✗	✓	=	✓	=	=	✓	✗
	EditCon	✗	✓	✗	=	=	=	✓	✗
MSE	RandomC(5)	✗	✓	✗	=	=	=	=	✗
	RandomC(10)	✗	✓	✗	=	=	=	=	✗
	RandomC(15)	✗	✓	✗	=	=	=	✓	✗
	kCenters(5)	✗	✓	✗	=	=	=	✓	✗
	kCenters(10)	✗	✓	✗	=	=	=	✓	✗
	kCenters(15)	✗	✓	✗	=	=	=	✓	✗
	EditCon	✗	✓	✗	=	=	=	=	✗

Table 5 Results obtained for the statistical significance tests comparing PG in the DS space with PS in the structural one. For each comparison, accuracy and set size are assessed. Symbols ✓, ✗ and = state that results achieved by elements in the rows significantly improve, decrease or do not differ respectively to the results by the elements in the columns. Significance has been set to $p < 0.05$.

this effect is inherent to the mapping process itself. A possibility to consider to palliate this effect would be the use of more robust embedding algorithms.

Taking this limitation into account, we can see the proposed strategy of PG in the DS space as very competitive when compared to PS in the initial space: considering the performance limitation due to the space mapping, and except for the case in which we compare MSE with FCNN, accuracy results achieved by PG are similar or even better than the ones by PS. This proves that PG algorithms can cope with the aforementioned drop.

Regarding the reduction capabilities, the proposed scheme achieves similar figures to the ones obtained by PS in the initial space: except when considering RSP3, which does not achieve great reduction figures, or when comparing to CHC, which performs the sharpest reduction, sizes do not significantly differ in the comparison.

In general, we can see that the proposed strategy of mapping the initial structural representation to a statistical one for then performing PG is able to achieve classification and reduction rates significantly similar to the ones obtained by PS in the initial space. This fact clearly questions the usefulness of the method as it does not improve over the results obtained in the classical scenario. However, if we consider computational cost for the classification, we can see that the proposed strategy stands as a very interesting alternative as it achieves statistically similar results in significantly shorter (several orders of magnitude) time lapses (see Table 3) than the structural representations. Additionally, if speed is the major concern, the proposed DS mapping with PG still stands as an interesting approach since a remarkable amount of fast-search algorithms have been proposed for feature-based space, in contrast to fast searching in metric spaces [33].

6 Conclusions

Prototype Generation techniques for Data Reduction in instance-based classification aim at creating new data out of the elements of a given set so as to lower memory requirements while precisely defining the decision boundaries. Although these methods are commonly used in statistical Pattern Recognition, they turn out to be quite challenging for structural data as the merging operations required cannot be as clearly defined as in the former approach. It has been proposed the use of Dissimilarity Space representations, which allow us to map structural data representations onto feature ones, to benefit from the advantages Prototype Generation methods depict.

The experimentation performed shows some important outcomes. In our results, PG approaches applied to structural data using DS representation are capable of competing with PS methods in the original space even though the mapping process implies information losses. Nevertheless, when compared to the figures obtained in the non-reduced structural space, PG methods depict lower accuracy results. Finally, classification using DS representations has been proved as a faster option than the one performed in the structural space as costly distance functions like Edit distance are replaced by low-dimensional Euclidean distance. This evinces the proposed approach as an interesting trade-off option between precision and time consumption.

Given the accuracy drop observed in the Dissimilarity Space mapping process, more sophisticated methods should be considered to check whether that loss could be somehow avoided. Additionally, experimentation could be extended including other Prototype Generation algorithms not considered in the present study.

References

1. Abreu, J., Rico-Juan, J.R.: A New Iterative Algorithm for Computing a Quality Approximated Median of Strings based on Edit Operations. Pattern Recogn. Lett. **36**(0), 74–80 (2014)
2. Angiulli, F.: Fast Nearest Neighbor Condensation for Large Data Sets Classification. IEEE T. Knowl. Data En. **19**(11), 1450–1464 (2007)
3. Arthur, D., Vassilvitskii, S.: K-means++: The Advantages of Careful Seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, pp. 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2007)
4. Borzeshi, E.Z., Piccardi, M., Riesen, K., Bunke, H.: Discriminative prototype selection methods for graph embedding. Pattern Recognition **46**(6), 1648–1657 (2013)
5. Bunke, H., Riesen, K.: Towards the unification of structural and statistical pattern recognition. Pattern Recogn. Lett. **33**(7), 811–825 (2012)
6. Calvo-Zaragoza, J., Oncina, J.: Recognition of Pen-Based Music Notation: the HOMUS dataset. In: Proceedings of the 22nd International Conference on Pattern Recognition, ICPR, pp. 3038–3043 (2014)
7. Calvo-Zaragoza, J., Valero-Mas, J.J., Rico-Juan, J.R.: Improving kNN multi-label classification in Prototype Selection scenarios using class proposals. Pattern Recognition **48**(5), 1608–1622 (2015)
8. Calvo-Zaragoza, J., Valero-Mas, J.J., Rico-Juan, J.R.: Prototype Generation on Structural Data using Dissimilarity Space Representation: A Case of Study. In: Paredes, Roberto and Cardoso, Jaime S. and Pardo, Xosé M. (ed.) 7th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), pp. 72–82. Springer, Santiago de Compostela, Spain (2015)
9. Cano, J.R., Herrera, F., Lozano, M.: On the Combination of Evolutionary Algorithms and Stratified Strategies for Training Set Selection in Data Mining. Appl. Soft Comput. **6**(3), 323–332 (2006)
10. Decaestecker, C.: Finding prototypes for nearest neighbour classification by means of gradient descent and deterministic annealing. Pattern Recogn. **30**(2), 281–288 (1997)
11. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**, 1–30 (2006)
12. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons (2001)
13. Duin, R.P.W., Pekalska, E.: The dissimilarity space: Bridging structural and statistical pattern recognition. Pattern Recognition Letters **33**(7), 826–832 (2012)
14. Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In: Proceedings of the First Workshop on Foundations of Genetic Algorithms, pp. 265–283. Indiana, USA (1990)
15. Fernández, F., Isasi, P.: Evolutionary Design of Nearest Prototype Classifiers. J. Heuristics **10**(4), 431–454 (2004)
16. Ferrer, M., Bunke, H.: An Iterative Algorithm for Approximate Median Graph Computation. In: Pattern Recognition (ICPR), 20th International Conference on, pp. 1562–1565 (2010)
17. Freeman, H.: On the encoding of arbitrary geometric configurations. Electronic Computers, IRE Transactions on **EC-10**(2), 260–268 (1961)
18. Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. IEEE Trans. Pattern Anal. Mach. Intell. **34**(3), 417–435 (2012)
19. García, S., Luengo, J., Herrera, F.: Data Preprocessing in Data Mining. Springer (2015)
20. García-Pedrajas, N., De Haro-García, A.: Boosting Instance Selection Algorithms. Knowl.-Based Syst. **67**(0), 342–360 (2014)
21. Hart, P.: The condensed nearest neighbor rule (corresp.). IEEE T. Inform. Theory **14**(3), 515–516 (1968)
22. de la Higuera, C., Casacuberta, F.: Topology of Strings: Median String is NP-Complete. Theor. Comput. Sci. **230**(1-2), 39–48 (2000)
23. Hjaltason, G., Samet, H.: Properties of embedding methods for similarity searching in metric spaces. Pattern Analysis and Machine Intelligence, IEEE Transactions on **25**(5), 530–549 (2003)
24. Hull, J.: A database for handwritten text recognition research. IEEE T. Pattern Anal. **16**(5), 550–554 (1994)
25. Kotsiantis, S.B., Kanellopoulos, D., Pintelas, P.E.: International Journal of Computer, Electrical, Automation, Control and Information Engineering **1**(12), 4091–4096 (2007)
26. Latecki, L.J., Lakämper, R., Eckhardt, U.: Shape descriptors for non-rigid shapes with a single closed contour. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 424–429 (2000)
27. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. In: Intelligent Signal Processing, pp. 306–351. IEEE Press (2001)
28. Mitchell, T.M.: Machine Learning. McGraw-Hill, Inc. (1997)
29. Nanni, L., Lumini, A.: Prototype reduction techniques: A comparison among different approaches. Expert Syst. Appl. **38**(9), 11,820–11,828 (2011). DOI 10.1016/j.eswa.2011.03.070
30. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence). World Scientific Publishing Co., Inc. (2005)

31. Rico-Juan, J.R., Iñesta, J.M.: New rank methods for reducing the size of the training set using the nearest neighbor rule. Pattern Recogn. Lett. **33**(5), 654–660 (2012)
32. Sánchez, J.: High training set size reduction by space partitioning and prototype abstraction. Pattern Recogn. **37**(7), 1561 – 1564 (2004)
33. Serrano, A., Micó, L., Oncina, J.: Which Fast Nearest Neighbour Search Algorithm to Use? In: J.M. Sanches, L. Micó, J.S. Cardoso (eds.) 6th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA). Funchal, Madeira, Portugal
34. Triguero, I., Derrac, J., García, S., Herrera, F.: A Taxonomy and Experimental Study on Prototype Generation for Nearest Neighbor Classification. IEEE T. Syst. Man Cy. C **42**(1), 86–100 (2012)
35. Tsai, C.F., Eberle, W., Chu, C.Y.: Genetic algorithms in feature and instance selection. Knowl.-Based Syst. **39**(0), 240–247 (2013)
36. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. J. ACM **21**(1), 168–173 (1974)
37. Wilson, D.L.: Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. IEEE T. Syst. Man Cyb. **2**(3), 408–421 (1972)