Prototype Generation on Structural Data using Dissimilarity Space Representation: A Case of Study

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

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante, Alicante, Spain
{jcalvo,jjvalero,juanra}@dlsi.ua.es

Abstract. Data Reduction techniques are commonly applied in instance-based classification tasks to lower the amount of data to be processed. Prototype Selection (PS) and Prototype Generation (PG) constitute the most representative approaches. These two families differ in the way of obtaining the reduced set out of 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 better delimit decision boundaries, operations required are not so well defined in scenarios involving structural data such as strings, trees or graphs. This work proposes a case of study with the use of the common RandomC algorithm for mapping the initial structural data to a Dissimilarity Space (DS) representation, 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 PG combined with RandomC mapping achieves a very competitive performance, although the obtained accuracy seems to be bounded by the representativity of the DS method.

1 Introduction

In the Pattern Recognition (PR) field, two fundamental approaches can be found depending on the model used for representing the data [6]: 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 methods or feature representations, in which the representation is based on numerical feature vectors.

The election of one of these approaches has some noticeable 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; feature methods, although less flexible in terms of representation, depict a larger collection of PR techniques for classification tasks [3].

Independently of whether we use a structural or a feature representation, instance-based PR methods, as for instance the k-Nearest Neighbor rule (kNN), may be applied. These methods, instead of obtaining a set of classification rules out of the available information, need to examine all the training data each time a new element has to be classified. As a consequence, these methods not only depict considerable memory requirements in order to store all the 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 [14].

Data Reduction (DR) techniques, a particular subfamily of the Data Preprocessing (DP) methods, try to solve these limitations by means of selecting a representative subset of the training data [11]. Two of the most common approaches for performing this task are Prototype Generation (PG) and Prototype Selection (PS) [13]. Both methods focus on reducing the size of the initial training set for lowering the computational requirements and removing noisy instances while keeping, if not increasing, the classification accuracy. The former method creates new artificial data to replace the initial set while the latter one simply selects certain elements from that set. Moreover, PG methods usually show a superior reduction and accuracy than the PS ones.

It must be pointed out that the two aforementioned DR techniques show a high 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 do require to modify the data in order to create new elements and, while this process can be easily performed in feature representations, it becomes remarkably difficult for structured data.

In this paper we shall study the possibility of applying PG methods to structured representations by means of using Dissimilarity Space (DS) methods so as to solve the aforementioned difficulty. DS techniques map structured data to feature representations, thereby allowing the use of statistical PR techniques not available for structured representations.

The rest of the paper is structured as it follows: Section 2 introduces the issue of Prototype Generation in structured data as well as the proposed approach; Section 3 presents the experimentation scheme implemented, the different data sets used and the results obtained; finally, Section 4 shows the general remarks obtained out of the study and proposes some possible future work.

2 Prototype Generation for Structural Data

The use of PS in instance-based classification assumes that the decision boundaries can be perfectly delimited by a subset of the prototypes in the training set, which may not always be true. PG methods, on the other hand, create new prototypes merging or evolving, when necessary, elements of the initial set to better define the decision boundaries [19].

As aforementioned, generating new prototypes in structural data is not a trivial matter. Some examples of works addressing this issue are [1] in which the median of a strings set is calculated using edit operations or [9] in which an iterative algorithms 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 representation cannot be assumed.

A possible solution to this issue is to use DS methods to map the structural data into a feature representation where these merging operations can be easily applied. 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.

In this paper we focus on the particular DS representation called RandomC, which shall be explained in the following section, for a proof-of-concept experience. The election of this method is motivated by its large application in the DS field as well as by the fact that the sampling process is performed equally for all clases in the set. Only one DS method is considered as the main aim of the paper resides in the DR techniques.

Using this method, the classification results obtained after applying a set of PG techniques to the feature representation will be compared to the results obtained when using PS techniques in both the initial structural space and the feature one so as to check whether PG can be useful in these situations.

2.1 Dissimilarity Space transformation: RandomC

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 $\mathcal{F}$, we resort to RandomC algorithm [15]. This algorithm selects a random subset of prototypes $R \subseteq T$, in which the number of prototypes of each class is exactly c (tuning parameter), that is, $|R| = c|Y|$. The elements of R are noted 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, a $|R|$-dimensional real-valued vector can be obtained for each point in the space $\mathcal{X}$.

In order to compare the influence of parameter c in the feature representation, some different values will be considered at experimentation stage.

3 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 the DS algorithm RandomC. In this experimental scheme, we fix its c parameter to retrieve 5, 10 or 15 prototypes per class.

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. The techniques used are listed in the following section.

In terms of the data used, two different isolated symbol datasets have been selected: the NIST SPECIAL DATABASE 3 (NIST3) of the National Institute of Standards and Technology, from which a subset of the upper case characters was randomly chosen (26 classes, 6500 images) and the United States Postal Office (USPS) handwritten digit dataset [12] (10 classes, 9298 samples). In both cases, contour descriptions with Freeman Chain Codes [10] are extracted.

Nearest Neighbor (NN) algorithm, parameterized with k=1, is used for the classification. Edit distance is considered as the dissimilarity measure for structural data whereas Euclidean distance is applied in the DS representation.

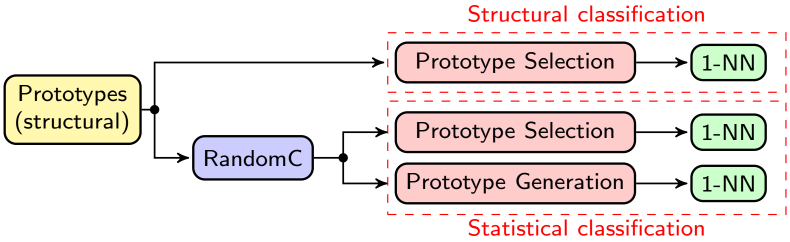

Fig. 1. Experimental set-up tested. RandomC is used as DS method. PS is applied to both structural and feature data while PG is only performed on the latter. 1-NN is used for the classification in all cases.

3.1 Data Reduction strategies

A representative set of PS 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 DR techniques are now briefly introduced:

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) [16]: 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) [7]: evolutionary algorithm commonly used as a representative of Genetic Algorithms in PS. The configuration of this algorithm has been the same as in [4].

Prototype Generation (PG) algorithms

- Reduction by Space Partitioning 3 (RSP3) [18]: divides the whole 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) [8]: performs an evolutionary search using a set of prototypes that can improve their local quality by means of genetic operators.
- Mean Squared Error (MSE) [5]: 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 [19].

3.2 Results

Results obtained for the different datasets proposed can be found in Tables 1 and 2 respectively. The performance metrics considered are the accuracy of the classification task and the size of the reduced set obtained (in % with respect to the whole dataset). ALL refers to results obtained when using the whole training set (no DR algorithm is applied).

		RandomC (5)		RandomC (10)		RandomC (15)		No DS	
Type	**Algorithm**	**Acc**	**Size**	**Acc**	**Size**	**Acc**	**Size**	**Acc**	**Size**
-	ALL	86.1	100.0	86.7	100.0	87.0	100.0	91.0	100.0
PS	FCNN	82.0	30.9	82.0	30.4	82.3	29.9	87.8	21.1
	1-FN(0.3)	79.7	16.8	81.5	16.9	81.2	16.9	87.9	15.8
	CHC	74.5	3.0	73.1	3.2	73.1	3.2	83.9	3.0
PG	RSP3	86.2	38.6	86.2	39.4	86.1	38.4	-	-
	ENPC	84.7	18.6	84.7	17.7	84.8	15.7	-	-
	MSE	82.8	7.0	83.2	5.6	83.1	5.7	-	-

Table 1. Results obtained with the NIST dataset for the proposed DS algorithm RandomC configurations. 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. Size normalization (%) is obtained with respect to the ALL case.

A first initial remark is that, in general, the DS process carries a reduction in the classification accuracy. For a given algorithm, when comparing the No DS accuracy results with any of the corresponding RandomC cases, there is a significant decrease in these figures. In the NIST set this effect is more noticeable: for instance, in the 1-FN(0.3) case of the NIST dataset, roughly a 10 % is missed between the original space and the RandomC (10) one. In the USPS, however, this effect is less accused, getting even to the point that the results obtained for

		RandomC (5)		RandomC (10)		RandomC (15)		No DS	
Type	**Algorithm**	**Acc**	**Size**	**Acc**	**Size**	**Acc**	**Size**	**Acc**	**Size**
-	ALL	91.3	100.0	91.7	100.0	91.7	100.0	91.8	100.0
	FCNN	87.7	21.1	88.4	20.1	88.2	20.1	87.6	20.2
PS	1-FN(0.3)	89.3	15.0	89.8	15.0	89.6	15.0	89.6	10.1
	CHC	87.4	0.9	88.0	0.9	88.4	0.9	89.2	0.8
	RSP3	91.4	18.7	91.7	17.7	91.7	18.3	-	-
PG	ENPC	89.8	5.7	90.7	5.0	90.6	5.5	-	-
	MSE	88.9	1.1	89.8	1.2	89.4	1.2	-	-

Table 2. Results obtained with the USPS dataset for the proposed DS algorithm RandomC configurations. 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. Size normalization (%) is obtained with respect to the ALL case.

the PS algorithms in the feature representation are similar, and in some cases better, than the corresponding ones in the initial space.

A second important point to comment is that for both structural and feature 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. PG, on the other hand, is capable of both achieving a remarkable size reduction, just as PS, and retrieving classification results close to the ALL case, showing the robustness of these methods. Especially interesting is the RSP3 algorithm which gets the same accuracy as the corresponding ALL scenario with barely a third of the initial set size.

Nevertheless, the main outcome out of the results obtained by the PG algorithms is that the scores they obtain in the feature domain are similar, when not higher, than the ones obtained by PS schemes in the initial structural space. This proves the proposed strategy as a clear competitor of PS in structural data, especially considering the simplicity of the RandomC DS algorithm employed.

Computation times were also measured for the classification schemes. Results proved DS strategies as much faster than structural ones because of the complexity reduction achieved by using Euclidean distance instead of Edit distance. For instance, classification times for the ALL case in the structural space for NIST and USPS span for 1127 and 216 seconds respectively while, for the RandomC (15) case, these tasks are accomplished in 102 and 6 seconds respectively.

Experiments show that the performance of PG seems to be limited by the results achieved by the ALL case in the feature space, then limiting the application of these techniques in situations where performance is a must. However, as the maximum achievable score is given by the ALL case, not PG algorithm but the DS technique are actually limiting the performance. In sight of this, performance might be boosted with the use of more robust DS algorithms.

Finally, the different c values for RandomC do not seem to have a remarkable effect on the results. For a given PS or PG technique, neither accuracies nor sizes do significantly change as this parameter is varied. As a consequence, low c values may be considered.

4 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 the 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 into feature ones, so as to benefit from the advantages Prototype Generation methods depict.

The experimentation performed shows some important outcomes. PG approaches applied to structural data using a DS representation are capable of competing with PS methods in the original space even though the mapping process implies information losses. However, PG methods are not capable of achieving accuracies reached in the non-reduced structural space as the mapping process does always carry a decrease in the overall performance, thus bounding the maximum achievable accuracy in the target space. 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.

Furthermore, work developed here opens several avenues for future work and extensions. For instance, a more comprehensive experimental setup could be addressed with datasets of other complex structures (such as trees or graphs), including a larger set of PS and PG algorithms. Moreover, experimentation with more advanced methods to map structured data into feature vectors, such those reported in [15,17], would be of great interest since accuracy achieved in our results seems to be bounded by the DS method applied. Finally, the inclusion of artificial noise in the data as well as the use of different parameters for the Nearest Neighbor classifier could be considered so as to assess the robustness of the system in adverse scenarios.

Acknowledgements

This work was partially supported by the Spanish Ministerio de Educación, Cultura y Deporte through a FPU fellowship (AP2012–0939), the Spanish Ministerio de Economía y Competitividad through Project TIMuL (No. TIN2013-48152-C2-1-R supported by EU FEDER funds), Consejería de Educación de la Comunidad Valenciana through project PROMETEO/2012/017 and Vicerrectorado de Investigación, Desarrollo e Innovación de la Universidad de Alicante through FPU program (UAFPU2014–5883).

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. Bunke, H., Riesen, K.: Towards the unification of structural and statistical pattern recognition. Pattern Recogn. Lett. 33(7), 811–825 (2012)
4. 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)
5. Decaestecker, C.: Finding prototypes for nearest neighbour classification by means of gradient descent and deterministic annealing. Pattern Recogn. 30(2), 281–288 (1997)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons (2001)
7. 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)
8. Fernández, F., Isasi, P.: Evolutionary Design of Nearest Prototype Classifiers. J. Heuristics 10(4), 431–454 (2004)
9. Ferrer, M., Bunke, H.: An Iterative Algorithm for Approximate Median Graph Computation. In: Pattern Recognition (ICPR), 20th International Conference on. pp. 1562–1565 (2010)
10. Freeman, H.: On the encoding of arbitrary geometric configurations. Electronic Computers, IRE Transactions on EC-10(2), 260–268 (1961)
11. García, S., Luengo, J., Herrera, F.: Data Preprocessing in Data Mining. Springer (2015)
12. Hull, J.: A database for handwritten text recognition research. IEEE T. Pattern Anal. 16(5), 550–554 (1994)
13. Li, Y., Huang, J., Zhang, W., Zhang, X.: New prototype selection rule integrated condensing with editing process for the nearest neighbor rules. In: IEEE International Conference on Industrial Technology, ICIT. pp. 950–954 (2005)
14. Mitchell, T.M.: Machine Learning. McGraw-Hill, Inc. (1997)
15. 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)
16. 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)
17. Riesen, K., Neuhaus, M., Bunke, H.: Graph Embedding in Vector Spaces by Means of Prototype Selection. In: Graph-Based Representations in Pattern Recognition, pp. 383–393. Springer Berlin Heidelberg (2007)
18. Sánchez, J.: High training set size reduction by space partitioning and prototype abstraction. Pattern Recogn. 37(7), 1561 – 1564 (2004)
19. 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)