From: Proceedings of the Eleventh International FLAIRS Conference. Copyright © 1998, AAAI (www.aaai.org). All rights reserved.

Prototype Selection from Homogeneous Subsets by a Monte Carlo Sampling

Marc Sebban
Equipe RAPID
Reconnaissance, Apprentissage et Perception Intelligente à partir de Données
UFR Sciences, Département de Mathématiques et Informatique,
Université des Antilles et de la Guyane, 97159 Pointe-à-Pitre (France)
marc.sebban@univ-ag.fr

Abstract

In order to reduce computational and storage costs of learning methods, we present a prototype selection algorithm. This approach uses information contained in a connected neighborhood graph. It determines the number of homogeneous subsets in the R^p space, and uses it to fix the number of prototypes in advance. Once this number is determined, we identify prototypes applying a stratified Monte Carlo sampling algorithm. We present an application of our algorithm on a simulated example, comparing results with those obtained with other methods.

Introduction

Selection of relevant prototype subsets has interested numerous searchers in pattern recognition for a long time. For example, non parametric classification methods such as *k-nearest-neighbors* (Hart 1968), *Parzen's windows* (Parzen 1962) or more generally methods based on geometrical models (Sebban 1996), (Preparata and Shamos 1985), have the reputation to have high computational and storage costs (Jain 1997), (Watson 1981), (Devijver 1982). Actually, the belonging class determination of a new instance often requires distance calculations with all points stored in memory. Nevertheless, the simplicity of these approaches encourages searchers in pattern recognition to build strategies to reduce the size of the learning sample, keeping classification accuracy (Hart 1968), (Gates 1972), (Ichino and Sklansky 1985) and (Skalak 1994).

Intuitively, we think that a small number of prototypes can have comparable performances (and perhaps higher) to those obtained with a whole sample. We justify this idea with two reasons :

1. Some noises or repetitions in data could be deleted,
2. Each prototype can be viewed as a supplementary degree of freedom. If we reduce the number of prototypes, we can sometimes avoid overfitting situations.

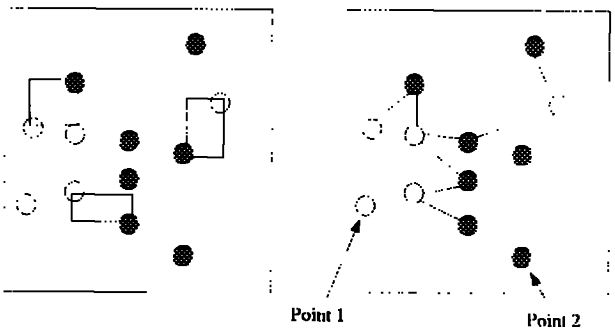

Figure 1: Point pruning by the rectangle method : 2 points are linked by an edge if the rectangle covering them does not contain any point ; points 1 and 2 are then deleted because the are not linked to any point of the set.

To reduce storage costs, some approaches use algorithms selecting misclassified instances such as *the condensed nearest neighbors* (Hart 1968) which allows to find a *consistent subset, i.e.* by correctly classifying all the remaining points in the sample set. In (Gates 1972), the author proposes the *reduced nearest neighbor rule* which improves the previous algorithm finding the *minimal consistent subset* if this one belongs to the Hart's consistent subset.

Among the other available algorithms, Ichino and Sklansky, in (Ichino and Sklansky 1985), propose to take into account **empty** rectangles linking points belonging to different classes to select prototypes (figure 1).

In (Skalak 1994), the author suggests two different prototype selection algorithms : the first one is a Monte Carlo sampling algorithm ; the second one applies random mutation hill climbing, where *fitness function* is the classification success rate on the learning sample. Our approach presented in this article is derived from the first one. This is the reason why we briefly review in the next section the useful aspects of the Skalak's selection algorithm. This method being limited to simple problems where classes of patterns are easily separable, we propose in section 3 an extension of the principle which determines in advance the number of prototypes from a connected neighborhood graph. In order to show the interest of our approach, we apply the algorithm on an example in section 4.

The Monte Carlo Sampling Algorithm

The principle of the Monte Carlo method is based on repeated stochastic trials (n). This way to proceed allows to approximate the solution of a given problem. In (Skalak 1994), the author suggests the following algorithm to determine prototypes. The number of prototypes (m) is fixed in advance as being the number of classes to learn.

1. Select n random samples, each sample with replacement, of m instances from the training set.
2. For each sample, compute its classification accuracy on the training set using a 1-nearest neighbor algorithm (Cover and Hart 1967).
3. Select the sample with the highest classification accuracy on the training set.
4. Classify the test set using as prototypes the sample with the highest classification accuracy on the training set.

This approach is simple to apply, but to fix m as being the number of classes is restrictive. Actually, it is limited to simple problems where classes of patterns are easy to separate. We can imagine some problems where the m classes are mixed, and where m prototypes are not sufficient. In the next section, we propose an improvement of this algorithm, searching for the number of prototypes in advance, and applying afterwards a stratified Monte Carlo sampling.

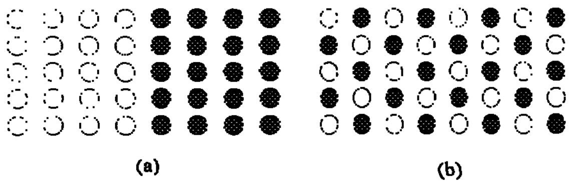

Figure 2: Examples of geometrical structures : on the one hand, the first example (a) shows a simple problem dealing with 2 classes (black and white). These classes are represented by two main structures of points belonging to the same class ; on the other hand, the second example (b) shows mixed classes, with numerous geometrical structures.

Homogeneous Subsets Extraction and Prototypes Selection

We think that performances of a learning algorithm, whatever the algorithm may be, depends necessarily on geometrical structures of classes to learn (figure 2). Thus, we propose to characterize these structures in R^p, called ***homogeneous subsets***, from the construction of a connected neighborhood graph.

Definition 1 *: A graph G is composed of a set of vertices noted Σ linked by a set of edges noted A ; they are thus the couple (Σ, A).*

Definition 2 *: A graph is considered connected if, for any couple of points $\{a, b\} \in \Sigma^2$, there exists a series of edges joining a to b.*

We automatically determine the number of prototypes applying the following algorithm :

1. Construction of the minimum spanning tree. This neighborhood graph is connected and contains the nearest neighbor of each point (figure 3).

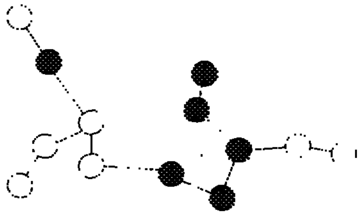

Figure 3: Minimum Spanning Tree (MST) : in this graph the edge length sum is minimum. From n points, the MST has always n-1 edges.

2. Construction of ***homogeneous subsets***, deleting edges connecting points which belong to different classes (*white* and *black*). Thus, a homogeneous subset is a connected sub-graph of the Minimum Spanning Tree (figure 4).

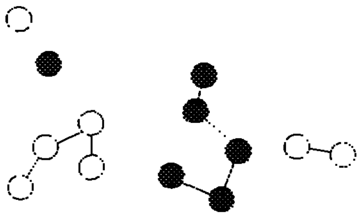

Figure 4: Homogeneous subsets

The number of homogeneous subsets seems to characterize the complexity to learn classes of patterns. **Thus, the more this number is high, the more the number of prototypes must be high**. Actually, when the problem is simple to learn (figure 2.a), a small number of prototypes is sufficient to characterize all the learning set. On the contrary, when the problem is complex (figure 2.b), the number of prototypes converges on the number of learning points. Thus, we decide to fix the number of prototypes in advance as being the number of homogeneous subsets. Afterwards, we search for the best instance of each homogeneous subset to identify its prototype. To do that, we apply a stratified Monte Carlo sampling.

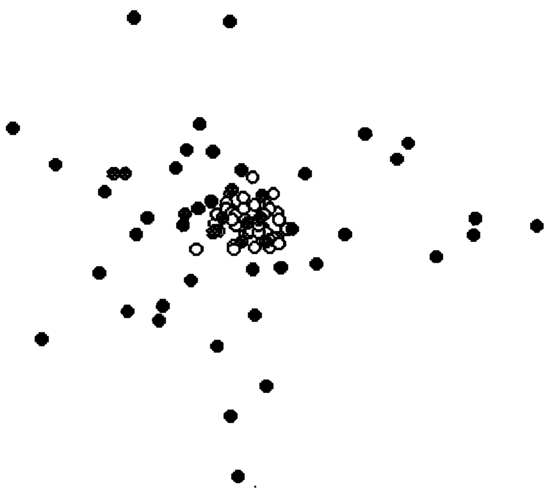

Figure 5: Example with two classes and 100 points

Example

Presentation

We apply in this section our prototype selection algorithm. Our problem is a simulation, with two classes C_1 and C_2 (figure 5), where :

1. C_1 contains 50 black points ; $\forall Xi \in C_1, Xi \equiv N(\mu,\sigma_1)$
2. C_2 contains 50 white points ; $\forall Xi \in C_2, Xi \equiv N(\mu,\sigma_2)$, where $\sigma_1 > \sigma_2$.

With these parameters we wanted to avoid two extreme situations :

1. the 2 classes are linearly separable. In this case (too simple) 2 prototypes are sufficient.
2. the 2 classes are totally mixed. In this case (too complex) the number of prototypes (m) converges on the number of points (n).

The validation set is composed of 100 new cases, equally chosen among the two classes C_1 and C_2.

	m=32	m=2	whole set
Success Rate	71%	55%	73%

Table 1: Results obtained on a validation set : the first column corresponds to the success rate obtained with our procedure ; the second corresponds to the Skalak's method, and the third one to the success rate obtained without reduction of the learning set

Results

In order to show the interest of our approach we want to verify that the reduction of the learning set does not reduce significantly the recognition performances of the model built on this new learning sample. We must verify that the success rate obtained with the subset is not significantly weaker than the one obtained with the whole sample.

Applying our algorithm of homogeneous subset extraction, we obtain $m = 32$ structures (figure 6). Results are presented in the table 1, with $n = 100$ trials.

This experiment shows that our algorithm allows to obtain a large reduction in storage, and results are close to those obtained with the whole learning set. The application of a frequency comparison test does not show a significant difference between the two rates (71% vs 73%). On the contrary, we bring to the fore with this example the limits of the Skalak's algorithm. where the number of prototypes is fixed in advance as the number of classes.

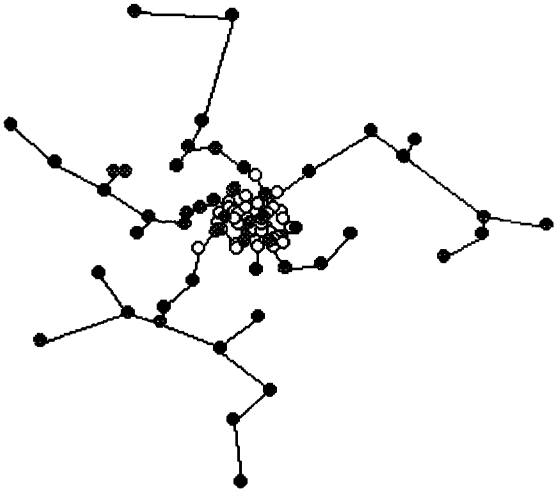

Figure 6: Minimum Spanning tree built on the training sample ; edges linking points which belong to different classes are deleted ; then, we obtain 32 homogeneous subsets.

Conclusion

We think that the reduction of storage costs of a learning set is closely related to the mixing of classes. In this article we have proposed to characterize this mixing by searching for geometrical structures (called *homogeneous subsets*) that link points of the same class. The smaller the number of structures is, the more the reduction of storage costs is possible. The main interest of our approach is to establish *a priori* the reduction rate of the learning set. Once the number m of prototypes is *a priori* fixed, numerous methods to select them are then available : *stochastic approaches, genetic algorithms, etc.* We have used a Monte Carlo sampling algorithm to compare our approach with other works. Nevertheless, we think that the *1-nearest neighbor* rule is not always adapted to classes with different standard deviation. Then, we are working on new labeling rules which take into account other neighborhood structures : *Gabriel structure, Delaunay polyhedrons, Lunes,* etc. (Preparata and Shamos 1985). These decision rules may allow to extract better prototypes from homogeneous subsets, using not only the 1-nearest neighbor to label a new unknown point but also the neighbors of neighbors.

References

COVER, T.M. AND HART, P.E. 1967. Nearest neighbor pattern classification. *IEEE Trans. Inform. Theory* 21-27.

DEVIJVER P. 1982. Selection of prototypes for nearest neighbor classification. In Proceedings of International Conference on Advances in Information Sciences and Technology.

GATES, G.W. 1972. The reduced Nearest Neighbor Rule. *IEEE Trans. Inform. Theory* 431-433.

HART, P.E. 1968. The condensed nearest neighbor rule. *IEEE Trans. Inform. Theory* 515-516.

ICHINO, M. AND SKLANSKY, J 1985. The relative neighborhood graph for mixed feature variables, *Pattern recognition, ISSN 0031-3203, USA, DA* (18):161-167.

JAIN, A. 1987. *Advances in statistical pattern recognition.* Springer-Verlag.

PARZEN, E. 1962. On estimation of a probability density function and mode. *Ann. Math. Stat* 33:1065-1076.

PREPARATA, F.P. AND SHAMOS, M.I. 1985. *Pattern recognition and scene analysis.* Springer-Verlag.

SEBBAN, M. 1996. Modèles théoriques en Reconnaissance de Formes et Architecture hybride pour Machine Perceptive. Ph.d Thesis., Lyon 1 University.

SKALAK, D.B. 1994. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In Proceedings of 11th International Conference on Machine Learning, Morgan Kaufmann, 293-301.

WATSON, D. 1981. Computing the n-dimensional Delaunay tesselation with application to voronoï polytopes, *The computer journal,* (24):167-172.