International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. 537-545
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Use of Neighborhood and Stratification Approaches to Speed up Instance Selection Algorithms

Frederic Ros[1], Rachid Harba[1], Marco Pintore[2] and Nadege Piclin[2]
[1] Institut Prisme Orleans University,
Orleans cedex 2, France
frederic.ros@univ-orleans.fr,rachid.harba@univ-orleans.fr

[2] Biochemics Consulting, Parc du Moulin,
Orleans Cedex 2, France
marco.pintore@biochemichs-consulting.com,nadege.piclin@biochemics-consulting.com

CORRESPONDING AUTHOR: Frederic ROS

***Abstract*: this paper investigates a method for instance selection in the context of supervised classification adapted to large databases. Based on the scale up concept, the method reduces the time required to perform the selection procedure by enabling the application of known condensation instance techniques to only small data sets instead of the whole set. The novelty of our approach relies in the way of hybridizing neighborhood and stratification approaches. The key idea is to consider instances found out for a given stratum to generate sub populations for the other strata representing critical regions of the feature space. Experiments performed with various data sets revealed the effectiveness and applicability of the proposed approach.**

***Keywords*:** supervised classification, instance selection, clustering algorithm, k-nearest neighbors.

I. Introduction

One of the primary, explicit challenges of the knowledge discovery and data mining community is the development of inductive learning algorithms that scale up to large data sets. The need is to turn towards more effective methodologies of data analysis which are capable of discovering the relevant information in reasonable scales of time and without parameter tuning. In the context of supervised classification, a database can be viewed as a training set, which is a series of patterns. Each training pattern is described by a set of features (from images, sounds, molecules...) and a class label specifying one of the possible categories. Comprehensibility and visualization are crucial issues for many applications in expert domain. At certain level, accurate models are not necessarily useful or interesting and other measures such as simplicity and novelty are also important. Rather than being interested in minute and quite often irrelevant details, the focal point is to reveal "phenomenal" at the level of some meaningful and easily comprehensible chunks of information that can be the object of fruitful interaction between experts.
Before any further algorithm, it would then be well worth running a data reduction technique to deepen and improve the expert approach. The reduction deals with two complementary objectives. The first objective addresses feature selection [1] where the primary purpose is to design a more compact classifier with as little performance degradation as possible. The second objective aims at generating a minimal consistent set, i.e., a minimal set whose classification accuracy is as close as possible to those obtained using all training instances. In this case, we talk about condensation or instance selection algorithms [2]-[4]. From a database represented by a matrix M_{mn} (m being the pattern number and n the feature one), the idea is therefore to reduce M in both dimensions. Most of approaches consist of managing this double reduction problem sequentially. A condensation approach is applied to patterns represented in a reduced feature space or a feature selection approach is applied to a reduced set of patterns selected from the original feature space. This paper deals with condensation. A condensation problem can be set as follows: Let $Z = z_1, \ldots, z_p$ be a set of samples described by a set of features $X = x_1, \ldots, x_f$. Each item, $z_j \in R^f$, is labeled, $L = 1, \ldots, l$ being the set of available labels. Given C_{1nn} a nearest neighbor classifier, the optimization problem consists in finding the smallest subset S_Z such that the classification accuracy of C_{1nn} over Z is maximal. The ultimate objective is to clearly find the smallest set of instances that enables the classifier to achieve nearly similar or better classification accuracy compared with the original set. In any cases, this smallest set of instances enables to deduce training sets without irrelevant samples on the basis of well-classified patterns. While remaining a challenging issue, condensation methods have been extensively studied in the literature by several exploratory techniques [5]-[9] including evolutionary algorithms [10]-[12]. These methods are general built upon the well-known k-NN methods [13]-[16] and usually seek to select representative instances, which could be border and/or central points. The reader interested can refer to [17] for a more theoretical review. Several methods give satisfying results and, among them, the DROP family ones [4] are the most popular today within the pattern recognition community. Their aim is discarding the non-critical instances. Starting from the original set, they remove the instances step by step, in an ordered and decremental way. An item is removed if at least as many of its

well classified neighbors can be correctly classified without it. Today, available methodologies and algorithms are sufficiently mature to handle a majority of problems with small sizes. In return, it is well-known that, for a problem of non-trivial size, the optimal solution of an instance selection problem is computationally intractable due to the resulting exponential search space. Tuning parameters is not anymore possible for large databases. The available algorithms mostly lead to suboptimal solutions and are incapable of providing acceptable solutions in an appropriate time. The speed of convergence has not been taken into account in the algorithm. For this reason, we consider another criterion T_{sz} called tractability that defines the necessary time to obtain S_Z. Different solutions have been investigated as the design of faster algorithms [18]-[20], the use of a relational representation or simply adopt the idea to take a random sample for running the data mining algorithm on it.

Several authors have also suggested adaptive or dynamic sampling approaches [19]. Random sampling remains difficult to use due to the difficulty of determining an appropriate sample size, and performances depend on it.

Through this idea, there is a danger of introducing a new bias into the learning, and determining an adequate sample size is a critical issue since theoretical results give impractical sizes.

If various solutions to handle the cases of non-trivial size can be imagined and combined, they seem to be not enough efficient for large databases.

More recently, several studies have proposed the use of scalability by approaches based on data partitioning [21]-[24]. The latter involves breaking the data set into subsets, learning from one or more of these subsets, and possibly combining the results. Data partitioning can be roughly done through "clustering" and "stratification" family approaches that in different views operate one "segmentation".

The stratification reduces the original data set size, splitting it into strata where the selection will be applied. The tractability is logically better, but the approach however requires application of an instance algorithm to all strata and this can be still time consuming. Olvera-Lopez and all propose a new fast instance selection method [23] for large datasets, based on clustering, which selects border prototypes and some interior prototypes. They propose to divide the training set in regions in order to find prototypes into small regions instead of finding them over the whole training set, which is very expensive. This approach appears to be an interesting direction but the authors do not explain how they handle the clustering problematic itself (cluster number, convergence…). An alternative approach consists of applying the divide-and-conquer principle [24] for scaling up instance selection algorithms, the idea being to apply in a recursive manner an instance selection algorithm to the selected instances of each subset regrouped in a new training set. This approach enables to seriously reduce the instance number, but does not address the tractability issue. Despite some successes of recent approaches, there is therefore a place for an improved method for selecting instance data.

The object of this paper is investigating a novel hybrid algorithm for instance selection with the objective to concentrate our efforts on the tractability aspect. The hybridization is structured in such a way that the classifier tractability and efficiency are optimized.

This algorithm manages the presence of minority classes. As a recall, minority classes are those that have few examples with respect to other classes. Our approach hybrids cluster and known instance selection techniques. The key idea is to consider instances found out for a given stratum in order to generate sub populations for the other strata. Instances firstly generated are automatically clustered allowing the populations representing the other strata to be fuzzy partitioned. A condensation algorithm applied to the sub populations obtained via a 1-nn procedure allows the other instances to be delivered.

This paper is organized as follows. Section 2 presents the hybrid approach to selecting instances and section 3 is dedicated to experimental results. Finally, Section 4 reports some concluding remarks and presents direction for further research.

II. Proposed approach

Our approach is based on the standard assumption to divide the initial population in strata. The idea consists of combining the use of neighborhood and condensation techniques to reduce the time required to perform the selection. The strata are firstly generated by including the problematic of minority classes. Then, the idea of the method is to apply a condensation algorithm C_s only to one stratum in order to obtain a set of preliminary instances (See Figure 1). This set is then automatically clustered via a proprietary algorithm providing several regions of interest. According to the clusters, interesting (influencing) patterns from the different strata are identified for each region to form new subsets Sr_i. Instances are therefore generated directly from Sr_i or subsets of Sr_i depending on the cardinality of each subset. The instances generated from the different regions are put together to constitute the final dataset.

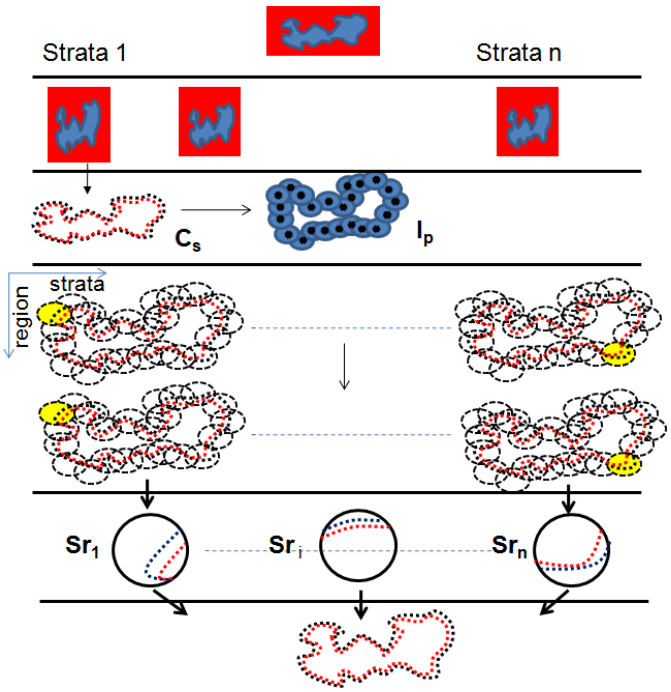

Figure 1. I_p is generated via C_s from one stratum leading to n_c clusters. Influencing patterns (Sr_i) are then determined via 1-nn procedure. Instances are then generated.

The process gains in efficiency as instances are found from small pattern sets instead of over the whole strata population. Our method is applicable to any condensation algorithm, this

last one being a parameter. The original idea of using clustering or neighborhood approach to perform instance selection is not new, neither the one to divide the initial training set in strata. Our method hybridizes these different ideas and the novelty relies in the way the hybridization is performed. The stratification is double and selective.

A. Data set partitioning in strata

The goal is to subdivide the training set into strata regarding the scalability aspect by making the searching process tractable and taking into account the problem of minority classes. The number of patterns for each category is generally different. Suppose known a minimal size in each stratum for each category in presence. We can deduce the number of strata c by selecting the category representing the maximum number of patterns. We propose to subdivide the training set T ($T = \cup_i T_i \; i \in [1,c]$) by considering each category independently. Then, for each category i, it is possible to partition T_i in n_i subgroups or strata, each of them having a cardinality $|T_{ij}| \geq = |T_i|/n_i \;\; \forall \; j \in [1, n_i]$. Each stratum S_p is composed of T_{ik} $i \in [1,c]$ and $k \in [1, n_i]$,

$$T_i = \cup_j T_{ij} \; j \in [1, n_i] \qquad (1)$$

The idea is to design strata by selecting the minor classes more frequently in order to make the class values uniformly distributed. Algorithm 1 consists of selecting the category having the largest number of subgroups, from which the stratum number is deduced. Then the algorithm consists of sequentially joining the available subgroups of the other categories. According to this process, some categories are not naturally present in some strata. Then, the completion is done for each stratum by randomly selecting a subgroup related to the categories not already included (Figure 2).

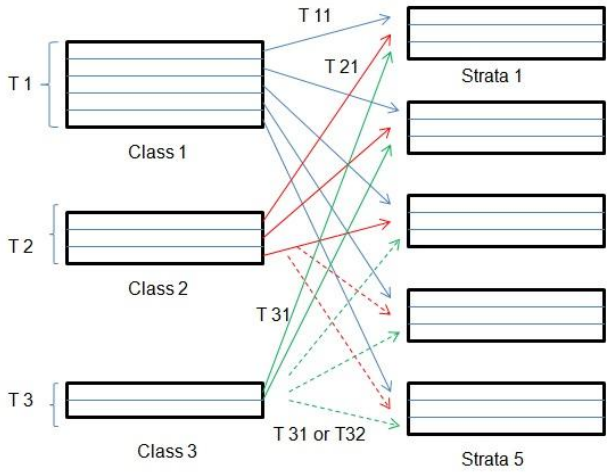

Figure 2. Example of stratum construction. As n_1=5, n_2=3, n_3=2 there is 5 strata. In stratum 3 to 5, some subgroups from class 2 and 3 are present at least two times.

Except for the category having the largest number of subgroups, some subgroups of the other categories are present in different strata.

Algorithm 1. Strata construction
Let n_1; n_2; . . . n_c be the number of subgroups for each category, c the number of categories in presence and S_i the data set of stratum i.

1. Find s / n_s = max (n_i) $i \in [1,c]$
2. For i=1 to n_s $S_i = T_{si}$ $i \in [1, n_s]$ (initialization of each stratum with class s)
3. For k=1 to c, For i=1 to n_k,
   $S_i = S_i \cup_q T_{kq}$, q ∈ [1, n_k]
4. For i=1 to n_s, for k=1 to c
   $S_i = S_i$ if class k is present else
   $S_i = S_i \cup T_{kq}$, q ∈ [1, n_k] is a random number

Regarding the stratified strategy, initial training data set T is divided into *N* sets S_i strata of approximately equal size, S_1, S_2, . . . , S_{ns}, with a revision of class distribution within each subset.

$$T \subseteq (\cup S_i) \; i \in [1, n_s] \qquad (2)$$

Different approaches to manage the problem of minor categories can be found in [25]-[26].

B. Data set partitioning in regions of interest via preliminary instances

The goal is to select interesting patterns in each stratum from which instances can be generated. This set is fuzzy partitioned into subsets that identify regions of the feature space. The idea underlying our method is that instances determined for a given stratum can serve as references to select interesting patterns in the other strata. The central point is to know how to find these patterns. Let be a condensation algorithm C_S that generates a set of instances from a given training set. C_S is then applied to a stratum j giving a set of preliminary instances Is_j also called I_p. The aim of condensation methods is to remove those instances that do not affect the decision boundary. Therefore I_p is generally very small compared to initial training set and gives information on the shape of the decision boundary.
It can be advantageously used to generate instances for the other stratum. The goal is to make the overall process as most tractable as possible. Then, our strategy consists of sub dividing I_p in n_c clusters, each of them covering a part of the decision boundary. Consider stratum 1 as the starting stratum to generate I_p.

$$I_p = I_{S1} = \cup I_{p_j} \; j \in [1, n_c] \text{ and } I_p \subseteq S_1 \qquad (3)$$

Each cluster has to be minimally consistent and the number the higher, the goal being to preserve efficiency while promoting the procedure computation. Many cluster approaches have been proposed and tested over the years have led to the common understanding that no universally “best” method exists. This understanding may be “natural,” but an extensive comparison has yet to be made among different validity measures [27].
In our case, the idea is to have each cluster containing at least a minimum number of elements in the neighborhood of a given

volume, to avoid null or very small clusters that are undesirable.
It should be underlined that even if the shape of the decision boundaries is complex, $|I_p|$ is very small in most of cases. This enables a very fast computation and also allows to applying the algorithm without any manual tuning.
We have developed a specific algorithm QUC (Quick Unsupervised Clustering) [28] particularly efficient to select critical classification prototypes from a database. An improved version of this algorithm QUC_1 is applied to I_p in order to determine n_c clusters.
Then, it is possible to define around each cluster an influence region R_i of the searching space. Each region R_i overlaps with its neighbors to avoid any intersection between the cluster and the category boundaries. It should be mentioned that the centers may not be representative of all the members, the distance between them being too large. More generally, clustering methods generally lack a precise control of the geometric size of each cluster. Each region is then identified by a subset of preliminary instances instead of a virtual cluster center.

QUC_1 algorithm:
The general outline of QUC_1 algorithm is illustrated in Figure 3 and can be depicted as follows:

Let $p_1; p_2; \ldots p_n$ be the set I_p of preliminary prototypes, k_r the number of nearest neighbors desired, and H_i the hyper sphere related to p_i.

1. Select the point p_s from I_p having the highest local density: initialize it as the first cluster ($I_s = p_s$) and update c_p with p_s.
2. Deactivate all points p_j from I_p belonging to H_i. The set consisting of the remaining points is renamed as I_{pa}.
3. Select the point p_i from I_{pa} the farest to c_p.
4. Select among the k_r nearest neighbors of p_i (including the inactive points) the point p_s presenting the highest local density.
5. Update c_p and I_s
6. Repeat Step 3 to 5 until I_{pa} becomes a null set.

The approach consists of a dual distance principle and neighborhood concept, which aims at finding prototypes iteratively to cover the pattern space by respecting some rules. Each prototype p_i defines an influence region determined by both its k_r nearest neighbors and a volume v_i automatically calculated on the basis of information delivered by I_p. Each new prototype has to be simultaneously far from a given prototype c_p while presenting a potential in classification. c_p is the prototype presenting the minimum distance with the prototypes already selected. Each prototype p_i defines an influence region determined by both its k_r nearest neighbors and a volume v_i automatically calculated on the basis of information delivered by I_p .

$$v_i = v = \tau * \sum_{i=1}^{|Ip|} d_{1nn}(I_{p_i}) \qquad (4)$$

where $d_{1nn}(x)$ is the one nearest neighbor distance of pattern x and the I_p elements except x, $\tau \in [0,1]$. A pattern $x \in H_i$ if it lies within a disc of radius centered at p_i or if it belongs to its k nearest neighbors, the value being fixed regarding the minimal consistency aimed. According to only the nearest consideration, regions of higher probability density are covered by smaller discs, and sparser regions are covered by larger discs. Consequently, more points are selected from the regions having higher density. By including the volume v_i, this assumption is attenuated and allows to moderate the leader number. It should be underlined that even if the shape of the decision boundaries is complex most of cases $|I_p|$ is very small. This enables a very fast computation and also allows to applying the algorithm without any manual tuning.

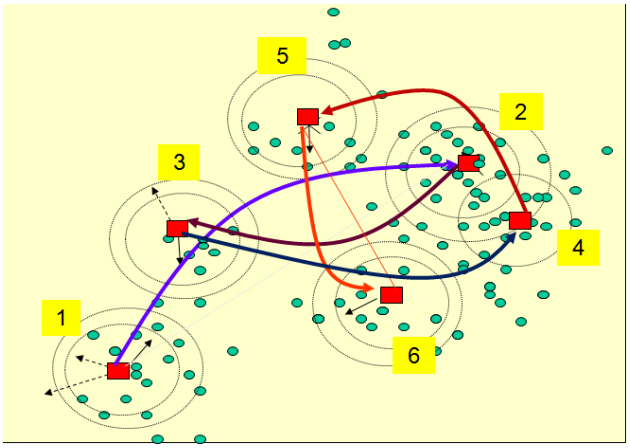

Figure 3. QUC_1 illustration: the process enables to have each cluster containing at least a minimum number of elements in the neighborhood of a given volume. Possible outliers are naturally discarded through this procedure.

It has to be underlined that k_r (1.. $|I_p|$) can be fixed *a priori* or by considering the ratio between the number of clusters and the maximum desired value c_{max}. If a cluster C_i is mainly composed of instances of the same category, it is associated to its nearest cluster containing an instance of another category. The selection is then done on the basis of the distance between two clusters C_i and C_j denoted as $d_c(C_i, C_j)$:

$$d_c(C_i, C_j) = \min_{\substack{\forall y_m \in C_i \text{ and } \forall y_n \in C_j \\ \text{and cat}(y_m) \neq \text{cat}(y_n)}} d(y_m, y_n) \qquad (5)$$

where $d(y_m, y_n)$ is the Euclidean distance between y_m and y_n while cat (y) identifies the category of y.
C_i is reinforced by adding some instances of C_j of different categories. The first condition for an instance $y \in C_j$ to be recruited is to be a nearest neighbor of one instance z of C_i and from a different category. The second condition for y is that z is its nearest neighbor among the z category. Some possible redundancies are not handled in this version.
The process is illustrated with an academic problem (Figures 4-7) of 10000 patterns that does not present any classification difficulty when managed by neighborhood approaches. It consists of four well separated rectangular clusters, two for each category shown (Figure 4). Clustering results are illustrated through different elementary stratum size (from Figures 5 to 7). The Figure 8 depicts the "crisp" segmentation obtained with an elementary size of 50 patterns per category without operating the selection (α=100%). The 26 delivered instances gives 7 clusters (k_r = 3) on the basis of 190 instances (seven are discarded). The computation time necessary to perform the segmentation is composed of the C_s time for 100

patterns, the clustering of 19 patterns and the calculations of 1900 distances (190 for each stratum).

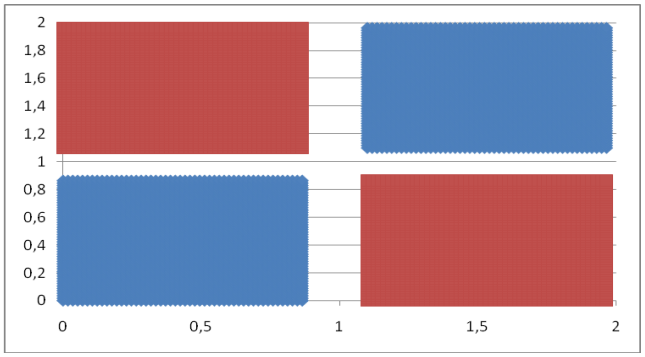

Figure 4. Basic and academic example including two categories, each of them represented by two rectangles comprising 10000 points.

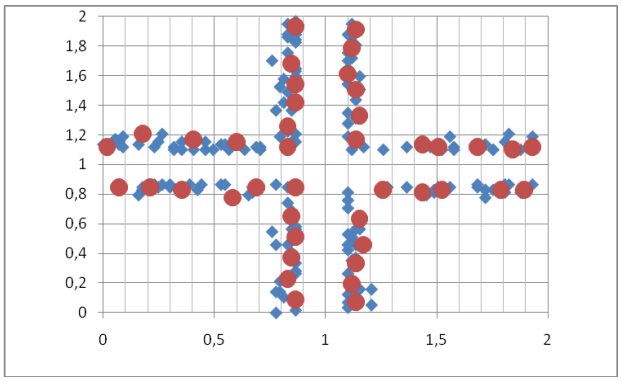

Figure 5. A circle point in the picture is an instance that identifies one cluster. It is linked to its nearest neighbors. This configuration has been obtained from one stratum of 800 patterns (400 for each category).

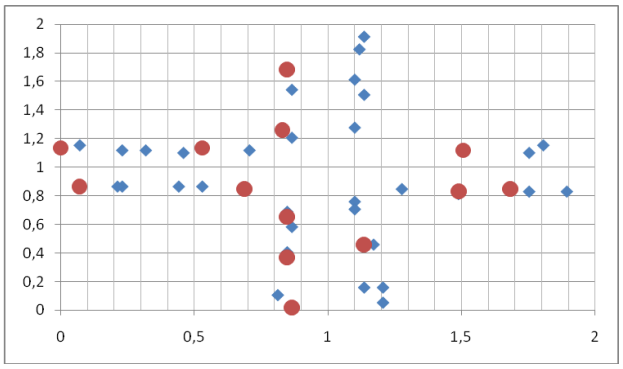

Figure 6. Configuration with 200 patterns per stratum

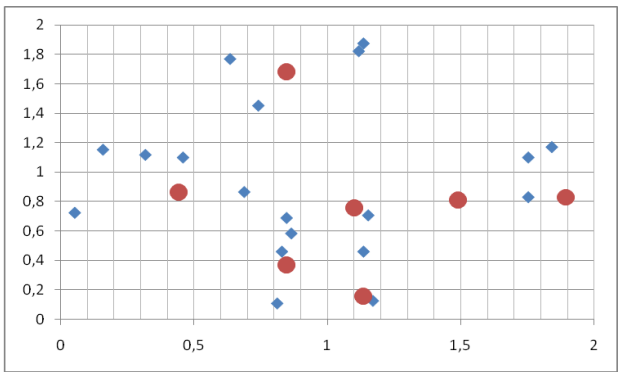

Figure 7. 26 different instances recruited from one stratum of 100 patterns have given seven clusters.

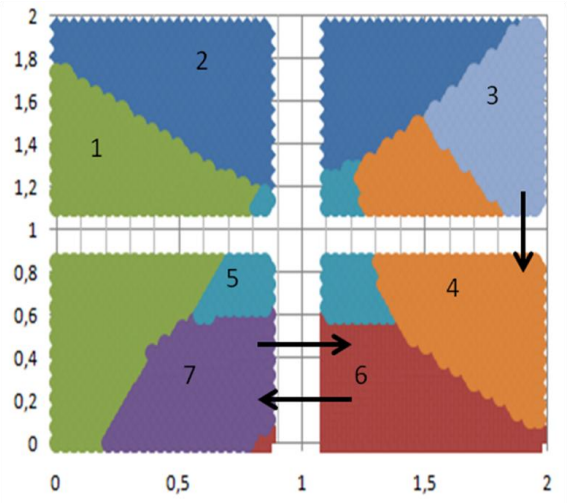

Figure 8. Database "segmentation" with seven clusters with α=100%. C_3, C_6 and C_7 are reinforced with instances from respectively C_4, C_7 and C_6.

C. Instance determination in each region

Let the cluster be represented by vectors z_1, z_2, …, z_β in p-dimensional Euclidean space where β is the number of clusters. Each cluster z_i is attached to a set of instances z_{ij} (j=1,…, β_i) where β_i is the number of instances for the cluster i including its center and $|I_p|=\sum_{i=1}^{\beta}\beta_i$.

A classical 1-nn procedure is used to determine the members of each cluster on the basis of the attached instances. Patterns to be clustered are denoted by x_k (k=1,…, n), n being the number of total patterns. A pattern x is defined as an influencing pattern for the active cluster C_a if it stands closer that another member from another cluster. It therefore belongs to its influence zone if one of the instances attached to this cluster is its nearest neighbor. A pattern y is attached to C_a if:

$$\underset{z_m \in C_a}{d(y,z_m)} < \min_{\substack{\forall z_n \in C_i \\ i=1 \text{ to } \beta}} (d(y,z_n)) \qquad (6)$$

but it is discarded if is not one of the k_i nearest neighbors. k_i is estimated as the ratio between the patterns concerned by the decision boundaries (α*n) and the number of reference instances (where α identifies the overall ratio of influencing patterns for determining the decision boundaries).

This "crisp" approach is a good base but not sufficient as some patterns close to the decision boundaries can belong to several clusters. To take into account this point, a pattern x can be assigned to different clusters when the distances are similar. The matrix $U = (\mu_{ik})$ (i=1,…,β, k=1,…,n) is then introduced to identify the degree of belonging of patterns x_k to cluster i:

$$\sum_{i=1}^{\beta}\mu_{ik} = 1, \forall k; \mu_{jk} > 0, \forall j, k \qquad (7)$$

This degree is calculated on the basis of d_p, the distance from a pattern y to the cluster i:

$$d_p(y, C_i) = \min_{\forall z_n \in C_i} \mathrm{d}(y, z_n) \qquad (8)$$

The goal of this fuzzy partition in regions is to have a small overlap avoiding that the cutting between regions falls at the borders between categories. A simple threshold rule can be applied on the basis of μ_{ik}.

Then, for each stratum, it is possible to associate a subset of interesting training patterns related to each influence region. Each pattern is both identified by its membership of one specific stratum and of one or several regions of the feature space.
A region is merged with another region if the number of members is too small. At the opposite, a region is split into sub-regions if it is too big. This can happen when the cluster number is small or in the case of heterogeneous category borders. The objective is to apply condensation algorithms to pattern set having a cardinality that does not exceed a pre defined value p_s. This region "segmentation" is obtained by randomly extracting k_p members from each stratum and from the active region to form a prototype set S_p of (n_s* k_p) cardinality. S_p is reinforced with the preliminary reference instances and clustered to produce a reduced set of prototypes R_p that identifies the sub-regions. Finally, a 1-nn procedure is applied to determine the sub-region members.

D. Algorithm review

The main operations of the standard process are the following ones. Firstly, it is necessary to apply a condensation algorithm C_s on one of size s to determine reference instances. Later, instead of processing a condensation algorithm C_s on a population containing n patterns, the approach consists of applying β times the algorithm C_s on small subsets.
Most of popular condensation algorithms applied to a set of n patterns presents a complexity of $0(n^2)$. Each operation presents then a complexity of $0(n_i^2)$, where n_i is the size of the subset i identified by the instances attached to cluster i. If the subsets have different sizes, it is however possible to estimate the average size and therefore deduce the complexity. There are (α*(γ*n)) patterns distributed among β clusters giving a size of (α* (γ*n))/β, where γ (γ≥1) is a coefficient identifying the fuzzy part of the process. The recruitment of interesting patterns requires distance calculation and ordering in each stratum. The number of calculations is therefore $|I_p|$*n_s*s.

III. Experimental results

Various benchmarks are therefore considered to validate the concept of the double optimization mechanism. As stated above, feature and instance selection are not independent and have to be considered globally. Our goal here is to focus on the instance part. Then, the selected data sets involve patterns represented by a few number of features suitable for applying k nearest approaches. The features are either the original ones or issued from a previous selection step. In our experiments, we have split off the data sets into training and test sets by applying a randomly partitioning. The condensation algorithm C_s selected to the stratification process is $DROP_4$ [4]. It is based on two main parameters: the k_f value for the filtering procedure and the k_{nn} value defining the k nearest neighbors considered in the processing. To simplify the tests, k_f, k_{nn}, k_p, k_r, p_s, c_{max}, α were respectively fixed to 1, 3, 4, 2, 300, 20, 60% for all the tests. Partition in strata was done to have 100 patterns of each category in each stratum.
The aim is to provide evidence that the performance of our stratification instance selection can be an interesting alternative to other approaches based on clustering. We therefore focus on the clustering approach presented in [23], which is conceptually the closest to ours. In this approach, the original training set is divided into regions via a clustering algorithm and prototypes are selected in each region. Like to the original paper, the c-means algorithm has been selected to perform the clustering. Different parameters have been tested in order to provide a fair comparison between the two approaches.

A. Data sets used

Nine data sets with well-known decision boundaries (except for one set) have been selected. Some of them have been downloaded from the UCI repository [29], some others are artificial, and the last one comes from the field of chemometrics. Focusing on applications dealing with databases of several thousand patterns, datasets including 5000 patterns have been considered for preliminary tests. Some random Gaussian noise was added to each dimension to obtain such data sets.
Concerning the artificial datasets, two category classification problems respectively in 1-D (named linear 1-D) and 2-D space (named square) with linear boundaries have been considered. We have also considered two class classification problems where each class follows a Gaussian distribution in the 8-D space. The first class is represented by a multivariate normal distribution with zero mean and standard deviation equal to 1 in each dimension; for the second class, the mean is also 0, but the standard deviation is 2 for each input variable. Furthermore, the 2-D spiral classification set, subject of many benchmarks, has been selected for its interesting complexity. The data points for the two classes C_k (k= 1, 2) are organized in spiral around each other for a same number of patterns in each class. For the data set called "chem." (5000 patterns, 166 features, 4 classes) coming from the chemometric field, most compounds were derived from analyses of the chemicals in a fathead minnow acute toxicity database [30]. To focus on the instance aspects only, relevant features have been selected before via our home made Genetic Algorithm [31].

B. Tests

Different cluster numbers (from 5 to 12) have been considered for the evaluation. Each algorithm was run independently with a different seed for the random number generator. The average of the results obtained by each algorithm in all data sets evaluated is shown in Tables 2 (our approach) and 3 (clustering approach done with 5 and 12 clusters). These tables are grouped in columns and, for each one, accuracy in training set, accuracy in test data, the number of final instances and the tractability are indicated.

For both data sets, our algorithm has clearly a better tractability than the clustering approach whatever the number of clusters. The difference depends on the complexity of the

boundaries, the density distribution of each population, and the number of clusters.

	Sample number	*Feature number*	*Class number*
Linear 1-D	5000	1	2
Square	5000	2	2
Gauss8D	5000	8	2
Concentric	5000	2	2
Clouds	5000	2	2
Spiral	5000	2	2
Phoneme	5000	3	4
Cancer	5000	4	4
Chem	5000	4	4

Table 1. Data sets used for our experiments

The biggest difference is obtained with the spiral problem presenting a ratio of more than ten. The difference between the methods is less relevant when the number of clusters increases. In the first case, the number of regions is directly driven by the process itself that combines the number of preliminary instances found for one stratum, c_{max} and p_s. In the second one, there is no indication about the appropriate number of clusters. Delivering relevant information without evaluating the complexity of the decision boundaries before is not possible.
Concerning the number of instances found by the approaches, it is difficult to distinguish relevant differences between the methods.
Most of time, the ratio between the results is less than 10%, and there is no clear advantage for a given algorithm. This can be easily explained; our approach is based on the same idea to apply a condensation algorithm to subsets of the original set. The difference relies only on the way to determine the subsets.
Concerning the classification performances, the clustering approach gives slightly better results for most databases. For the chemometric database, around 2% of difference for the training set, when applying the clustering approach, is found. This is probably due to some imperfections in the division procedure. We should investigate in this way to improve this point.
Concerning the performances, we have to mention the poor results for the square database obtained with the clustering approach. With 5 clusters, only about 85% of classification has been obtained whereas 92% is obtained with 12. If a simple modification in the native algorithm can solve this issue, the results bring however two remarks: the number of clusters remains problematic, and clustering approaches are not the most suitable for classification purposes. They remain unsupervised techniques and do not include the classification boundaries.

Finally, we would like to point out that our proposal in this first version appears to be competitive with other models, especially regarding the tractability criterion. On the basis on these preliminary tests (that need, of course, to be completed with larger databases), the scaling up problem is well managed.

	Train (%)	*Test (%)*	T_{sz}*(s)*	*Instance number*
Linear 1-D	99,8	99,7	0,06	17
Square	98,7	99,8	0,8	67
Gauss 8-D	86,1	85,9	0,34	280
Concentric	97,8	98,1	0,2	145
Clouds	83,6	83,8	0,9	211
Spiral	95,9	96,1	0,9	421
Phoneme	86,2	85,3	1	45
Cancer	99,8	99,9	0,11	9
Chem	81	81,1	1,7	472

Table 2. Results for the 9 data sets with our approach

	Train (%)	*Test (%)*	T_{sz}*(s)*	*Instance number*
Linear 1-D	99-98	99-99	0,17-0,27	5-9
Square	85-92	84-92	3,6-2,1	16-26
Gauss 8-D	87-88	88-89	11,7-4,6	288-301
Concentric	98-97	99-98	0,8-0,8	107-112
Clouds	85-84	85-85	4,3-2,9	392-388
Spiral	95-95	94-94	17,9-6,8	356-353
Phoneme	88-88	86-87	5,2-3	279-287
Cancer	99-99	99-100	0,2-0,3	12-11
Chem	83-84	82-82	10,2-5,2	486-473

Table 3. Results for the 9 data sets with the clustering approach

C. Discussion and perspectives

In a discovery approach we do not have to excessively worry about the classification accuracy. The essential is to avoid and change the result drastically and, especially, deliver understandable and viewable information that allows a better expert exploration support. The tractability is therefore critical. The method has been developed in this direction while keeping the idea of clustering that appears mandatory.
The dataset used in this work are not really large and complex (regarding the shape of the decision boundaries) except for the "Chem" database. For most classification problems, patterns that are concerned by the decision boundaries present a very little ratio of the whole population, say less than 10%. A value of $\alpha = 60\%$ was fixed in our test to compensate the fact that the same *a priori* small size was used for all the datasets.
The results obtained through this preliminary version on various databases are very encouraging and show that the concept is worth to be deeper investigated.
Some optimizations are therefore in progress to make the progress more automatic and reliable for managing very large databases and face different complexities. They concern the way the strata are generated and the recruitment of interesting patterns in each region from which the condensation algorithms are applied. These two points are linked and then have to be managed jointly. The more the preliminary instances are close to the true boundaries, the smaller sets can

be obtained leading naturally to better computation processing time. Similarly, the higher the cluster is, the smaller the pattern sets are. Accuracy related to the preliminary instances has however a cost; the tradeoff between accuracy and compactness has to be managed.

For the first point, it has been demonstrated that there is a serious difficulty to define the adequate size of a random set to work on it instead of the whole data set. This is a critical point but, by now, poor attention has been paid to the relevance of the simple stratum information, and the problem, in the recent scalability approaches [21]-[24], is solved by recommending *a priori* sizes. It is however possible, by some elementary statistical considerations, to define a minimal size of the strata from which the elementary distributions for each category are minimally represented. If this point has not been directly treated, there are different studies on the field that offer some interesting ingredients [32], [33]. If each stratum is minimally represented, it is then possible to cumulate several strata to obtain a "meta" stratum that fits to the boundaries complexity where the level of accuracy is better controlled.

For the second point, there is clearly a place to optimize the size of influencing patterns from which the condensation algorithms can be applied. The distribution Y of the nearest distances of the reference population can be processed and easily modeled via an exponential distribution from which the mean E(Y) and standard deviation $\sigma(Y)$ can be computed. Larger values of E(Y) indicate sparse organization and conversely small values indicate dense organization. This information combined with more local considerations gives relevant criteria to implement a more selective process in the recruitment of influencing patterns.

We should also underline that others combinative alternatives to apply condensation algorithms to small sets can be investigated and are in process. Each of them has naturally some strengths and weaknesses. They can be alternatively applied regarding the problem to be faced.

IV. Conclusion

In this study, a novel hybrid algorithm for instance selection in the context of supervised classification was investigated. Based on scalability concepts, it generates an efficient instance set in a few selection steps and manages the presence of minority classes. The scalability concerns the division of the algorithm in strata, but also the division of strata in regions which considerably reduces the time required to perform the selection. The novelty of our approach relies on the way to apply condensation and clustering algorithms to only small sets of patterns. Experiments performed with various data sets revealed the effectiveness and applicability of the proposed approach. As proved by the results, this algorithm is likely to give satisfactory results, within a reasonable time, when dealing with non trivial-size data sets.

References

[1] Yang J., Olafsson S. "Optimization-based feature selection with adaptive instance sampling" *Computers & Operations Research* 33, pp 3088–3106, 2006.

[2] Aha D, Kibler D, Albert M.K. "Instance-based learning algorithms" *Journal of Machine Learning* 6, pp 37–66, 1991.

[3] Swonger CW Sample set condensation for a condensed nearest neighbour decision rule for pattern recognition. In:Watanabe S (ed) Academic, Orlando, pp 511–519, 1972.

[4] Wilson D.R., Martinez T.R. "Reduction techniques for instance-based learning algorithms" *Journal of Machine Learning* 38(3), pp 257–286; 2000.

[5] Kim SW, Oommen BJ. "A brief taxonomy and ranking of creative prototype reduction schemes" *Pattern Anal Appl* 6, pp 232–244, 2003.

[6] Kim SW, Oommen BJ. 'Enhancing Prototype reduction schemes with recursion: a method applicable for Large data sets" *IEEE Trans Syst Man Cybern* 34(3) Part B, 2003.

[7] Grochowski M., Jankowski N. "Comparison of instance selection algorithms I. Algorithms survey," *in Proceedings of the Seventh International Conference on Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science* 3070, Springer, pp 580–585, 2004.

[8] Pabitra M, , C.A. Murthy, and Sankar K. Pal. "Density-Based Multiscale Data Condensation" *IEEE Trans On Pattern Ana and Mach. Int*, 24(6), pp 734 - 747, 2002.

[9] Wang, J., Neskovic, P., Cooper, L.N. "Improving nearest neighbor rule with a simple adaptive distance measure" *Patt. Rec. Lett.* 28, pp 207–213, 2007.

[10] Cano J.R, Herrera F., Lozano. "Using evolutionary algorithms as instance selection for data reduction in kdd: an experimental study" *IEEE Trans Evol. Comput*. 7(6):193–208, 2003.

[11] Ros F., Guillaume S. *An efficient nearest classifier, Book Chapter of Hybrid Evolutionary Systems, Studies in Computational Intelligence* 75, Springer Verlag, 2007.

[12] Kuncheva LI. "Fitness functions in editing k-NN reference set by genetic algorithms" *Pattern Recognition 30(6)*, pp 1041–1049, 1997.

[13] Hart PE. "The condensed nearest neighbor rule" *IEEE Trans Inf Theory* 16, pp 515–516, 1968.

[14] Gates GW. "The reduced nearest neighbor rule" *IEEE Trans Inf Theory* 18(3), pp 431–433, 1972.

[15] Dasarathy BV. "Minimal consistent subset (MCS) identification for optimal nearest neighbor decision systems design" *IEEE Trans SystMan Cybern* 24, pp 511–517, 1994.

[16] Anil K. Ghosh, Probal Chaudhuri, and C.A. Murthy. "On Visualization and Aggregation of Nearest Neighbor Classifiers" *IEEE Trans On Pattern Ana and Mach. Int* 27(6), pp. 1592-1602, 2005.

[17] Brighton H, Mellish C. "Advances in instance selection for instance-based learning algorithms" *Data Min Knowl Discov* 6, pp 153–172, 2002.

[18] Jaekyung Yang, Sigurdur Olafsson. "Optimization-based feature selection with adaptive instance sampling" *Journal of Computers & Operations Research* 33, pp 3088–3106, 2006.

[19] Domingo C, Gavalda R,Watanabe R. "Adaptive sampling methods for scaling up knowledge discovery algorithms" *Journal of Knowledge Discovery and Data Mining* 6(2), pp 131–52, 2002.

[20] Zhu X, Wu X. "Scalable representative instance selection and ranking" *In Proceedings of the 18th international conference on patter recognition (ICPR'06), vol 3. IEEE Computer Society*, pp 352–355, 2006.

[21] Herrera J.R. Cano, F. Herrera, M. Lozano. "Stratification for scaling up evolutionary prototype selection" *Pattern Recognition Lett.* 26 (7), pp 953–963, 2005.

[22] José Ruiz-Shulcloper and Walter G.Kropatsh "Progress in Pattern Recognition, Image Analysis and Applications" *Proceeding of the 13th Iberoamerican Congress on Pattern Recognition*, pp 9-12, 2008.

[23] Arturo Olvera-Lopez J., Carrasco-Ochoa A., Martınez-Trinidad J. "A new fast prototype selection method based on clustering" *Journal of Pattern Anal Applic*, pp 131-141, 2009.

[24] Haro-García A, García-Pedrajas D.:"A divide-and-conquer recursive approach for scaling up instance selection algorithms" *Data Min Knowl Disc* 18, pp 392–418, 2009.

[25] Jose-Ramon Cano a, Salvador Garcia b, Francisco Herrera. "Subgroup discover in large size data sets preprocessed using stratified instance selection for increasing the presence of minority classes", *Pattern Recognition Letters* 29, pp 2156–2164, 2008.

[26] Show-Jane Yen, Yue-Shi Lee "Cluster-based under-sampling approaches for imbalanced data distributions" *Expert Systems with Applications* 36, pp 5718–5727, 2009.

[27] Wataru Hashimoto, Tetsuya Nakamura, and Sadaaki Miyamoto "Comparison and Evaluation of Different Cluster Validity Measures Including Their Kernelization" *Journal of Advanced Computational Intelligence and Intelligent Informatics* 13 (3), 2009.

[28] Ros F, Pintore M., Chretien J.R. "Automatic Design of Growing Radial Basis Function Neural Networks using Supervised Clustering and neighborhood concepts" *Journal of Chemometry and Intelligent Laboratory Systems* 87 (2), pp 231-240, 2007.

[29] P.M. Murphy and D.W.Aha,"UCI Repository for Machine Learning Databases," technical report, Dept. of Information and Computer Scienc, Univ. of California, Irvine, Calif, 1994.

[30] Pintore M, Piclin N, Benfenati E, Gini G, Chretien JR "Predicting toxicity against the fathead minnow by Adaptive Fuzzy Partition", *QSAR & Combinatorial Science* 22, pp 210-219, 2003.

[31] Ros F, Guillaume S. Pintore M., Chretien J.R. "Hybrid Genetic Algorithm for Dual Selection" *Journal of Pattern Analysis and application* (1), pp 179-198, 2007.

[32] Wai Lama, Chi-Kin Keunga, Charles X. Lingb "Learning good prototypes for classi#cation using filtering and abstraction of instances" *Pattern Recognition* 35, pp 1491–1506, 2002.

[33] Shi L, Olafsson S. "Nested partitions method for global optimization" *Journal of Operations Research* 48, pp 390–407, 2002.

Author Biographies

Frederic Ros has an engineering degree in Microelectronics and Automatic, a Master in Robotics from Montpellier University and a Ph.D. degree from ENGREF (Ecole Nationale du Genie Rural des Eaux et Forets) Paris. He began his career in 1991 as a research scientist working on the field of image analysis for robotics and artificial systems from CEMAGREF (Centre National d'Ingénieurie en Agriculture) where pioneer applications combining neural networks, statistics and vision were developed. He managed the vision activity in GEMALTO during 14 years which is the world leader in the smart card industry. He was particularly involved in applied developments (related to data analysis, fuzzy logic and neural networks) with the aim of providing adaptive and self-tuning systems corresponding to the growing complexity of industrial processes and especially multi-disciplinary interactions. He has been an associate researcher at PRISME laboratory and head an innovation park for 4 years. He has co-authored over 70 conference and journal papers and made several reviews in this field.

Rachid Harba received the Agregation in electrical engineering from ENS Cachan, Cachan, France in 1983. He received the PhD degree in electrical engineering from INPG Grenoble, France in 1985. Since 1987, he has joined the Laboratory of Electronics, Signals, Images (LESI), Orléans, France, as an Associate Professor and teached at Polytech'Orleans engineering school. In 1997, he became a full Professor and took the head of the LESI. He is now a first class Professor at PRISME laboratory, University of Orleans. He is interested in signal and image processing applied to biomedical domains, material science and industrial applications. He is the author or coauthor of about 100 papers in Journals and conferences.

Marco Pintore is at present CEO of EtnaLead srl, a SME focused on the design of alternatives solutions for the pharmaceutical and cosmetic domains, after achieving a long experience in managing BioChemics Consulting SAS, a SME providing bioactivity prediction and virtual screening engineering services (pharmacy, cosmetics and environment). Marco has a large expertise, besides management, in the prediction of molecular bioactivity by data mining and 3D molecular modeling techniques, and he has been involved in about 40 publications in peer reviewed scientific journals. Marco has also participated as scientific leader in several projects funded by the European Commission, within the Fifth and Sixth Framework Programme, and was coordinator of several national funded R&D projects.

Nadège is biochemist, with a PhD in ChemoInformatics and BioInformatics from the University of Orléans. She has been a member of BioChemics team since its creation. She is Data Mining Manager, and is in charge of the realisation of the engineering services proposed by BioChemics. She also manages the scientifical aspects of several research projects. She is the author or coauthor of 20 papers in Journals and conferences.