Local-set based-on instance selection approach for autonomous object modelling

Carbonera, Joel Luis; Olszewska, Joanna Isabelle

Published in:
International Journal of Advanced Computer Science and Applications

Published: 31/12/2019

Document Version
Peer reviewed version

Local-Set Based-On Instance Selection Approach for Autonomous Object Modelling

Joel Luis Carbonera
IBM Research, Rio de Janeiro, Brazil
Email: joelc@br.ibm.com

Joanna Isabelle Olszewska
University of West Scotland, UK
Email: joanna.olszewska@ieee.org

***Abstract*—With the increasing presence of robotic agents in our daily life, computationally efficient modelling of real-world objects by autonomous systems is of prime importance for enabling these artificial agents to automatically and effectively perform tasks such as visual object recognition. For this purpose, we introduce a novel, machine-learning approach for instance selection called Approach for Selection of Border Instances (ASBI). This method adopts the notion of local sets to select the most representative instances at the boundaries of the classes, in order to reduce the set of training instances and, consequently, to reduce the computational resources that are necessary to perform the learning process of real-world objects by the artificial agents. Our new algorithm was validated on 27 standard datasets and applied on 2 challenging object-modelling datasets to test the automated object recognition task. ASBI performances were compared to those of 6 state-of-art algorithms, considering 3 standard metrics, namely, accuracy, reduction, and effectiveness. All the obtained results show that the proposed method is promising for the autonomous recognition task, while presenting the best trade-off between the classification accuracy and the data size reduction.**

***Keywords*—Machine Learning; Instance Selection; Autonomous Systems; Object Modelling; Visual Object Recognition; Computer Vision; Machine Vision.**

I. Introduction

Instance selection (IS) is a machine-learning, pre-processing task that consists in choosing a subset of instances among the total available data, in a way that the *subset* can support the machine learning task *with a low loss of performance* [1], [2]. Thus, every IS strategy faces a *trade-off* between the *reduction rate* of the dataset and the resulting *classification accuracy* [3], [4].

In Machine Learning, instance selection can be applied to reduce the data into a manageable subset, leading to a reduction of the computational resources (in terms of time and space) necessary to perform the learning process [5], [6], [7]. Besides that, instance selection techniques can be used to improve the learned models through the deletion of useless, redundant, erroneous, or noisy instances [5], [8].

In this paper, we propose a new instance selection algorithm called ASBI (*Approach for Selection of Border Instances*) that applies the notion of *local set* [7] to guide the instance selection process. Hence, the proposed ASBI algorithm aims to preserve the most relevant instances at the *boundaries* of the data classes. Indeed, border instances provide relevant information to support discrimination between classes [7].

Moreover, we aim to use this method in Robotics for tasks such as autonomous object modelling, since object modelling for autonomous agents requires a reduced set of training instances to cope with the *in-situ* computational constraints [9], [10]. Furthermore, effective object modelling coupled with machine-vision-based recognition algorithms [11], [12] could lead to efficient autonomous object recognition, which is very challenging for robots and robot ecologies [13], [14].

The contribution of this paper is thus twofold and consists of (i) the new instance selection algorithm ASBI[1] and of (ii) the application of instance selection algorithms, in particular ASBI, to the automated object modelling for autonomous agents.

Thence, the ASBI algorithm was evaluated, on one hand, on a generic classification task and, on the other hand, on a specific visual recognition task. Its performance was then compared with the performance of 6 well-established algorithms such as the *Edited Nearest Neighbour* (ENN) algorithm [15], the *Decremental Reduction Optimization Procedure* (DROP1-DROP5) algorithm [16], the *Iterative Case Filtering* (ICF) algorithm [17], the *Local Set-based Smoother* (LSSm) algorithm [7], the *Local Set Border Selector* (LSBo) algorithm [7], and the *Local Density-based Instance Selection* (LDIS) algorithm [18]. For the classification task, it was evaluated on 27 standard datasets, considering the SVM classifier [19]. For the visual recognition task, it was evaluated on 2 challenging and well-known datasets, using the approach proposed in [12]. The results show that ASBI provides the best trade-off between accuracy and reduction, in comparison with the other state-of-the-art algorithms.

The remaining part of the paper is organized as follows. Section II describes some state-of-the art methods in the field of instance selection. Section III presents the notation used throughout this paper, while our approach is explained in Section IV. The experimental evaluation is presented in Section V, and conclusions are drawn up in Section VI.

II. Related Works

In this section, we discuss some important instance reduction methods. In this discussion, we consider T as the original set of instances in the training set and S, with $S \subseteq T$, as the reduced set of instances, resulting from the instance selection process.

[1]The source code of our ASBI algorithm can be found at https://www.researchgate.net/publication/317788063_ASBI_Approach_for_selection_of_border_instances

The *Condensed Nearest Neighbour* (CNN) algorithm, which was introduced by [20], randomly selects one instance that belongs to each class from T and puts them in S. Then, each instance $\in T$ is classified using only the instances $\in S$. If an instance is misclassified, it is added to S, in order to ensure that it will be classified correctly. This process is repeated until there are no instances in T that are misclassified. CNN is a popular method. However, CNN can assign *noisy* instances to S, causing negative effects in the classification accuracy. CNN is also dependent on instance order in the set T. The time complexity of CNN is $O(|T|^2)$, where $|T|$ is the size of the training set.

The *Reduced Nearest Neighbour* (RNN) algorithm [21] assigns all instances in T to S at first. Then, it removes each instance from S until further removal causes no other instances in T to be misclassified by the remaining instances in S. RNN is less sensitive to noise than CNN and produces subsets S that are smaller than the subsets produced by CNN. The main drawback of RNN is its cubic time complexity.

The *Generalized Condensed Nearest Neighbour* (GCNN) algorithm was proposed by [3]. Considering, on one hand, $d_N(x)$ as the distance between x and its nearest neighbour and, on the other hand, $d_E(x)$ as the distance between x and its nearest enemy (i.e. the instance of a class that is different from the class of x), x is included by GCNN in S if $d_N(x) - d_E(x) > \rho$, where ρ is an arbitrary threshold. In general, GCNN produces sets S that are smaller than the sets produced by CNN. However, determining the value of ρ can be a challenge.

The *Edited Nearest Neighbour* (ENN) algorithm [15] assigns all training instances to S at first. Then, each instance in S is removed if it does not agree with the label of the majority of its k nearest neighbours. This strategy is effective for improving the classification accuracy of the learned models, because it removes noisy and outlier instances. However, since it keeps internal instances, it cannot reduce the dataset as much as other reduction algorithms. The literature provides some extensions of this method, such as [22].

In [16], the 5 presented algorithms (i.e. DROP1-DROP5) are based on the (*Decremental Reduction Optimization Procedure* (DROP) approach. These algorithms assume that each instance x has k nearest neighbours ($k \in \mathbb{N}$), and those instances which have x as one of their k nearest neighbours are called the *associates* of x. Among the proposed algorithms, DROP3 has the best trade-off between the reduction of the dataset and the accuracy of the classification. As an initial step, it applies a noise-filter algorithm such as ENN. Then, it removes an instance x if its associates in the original training set can be correctly classified without x. The main drawback of DROP3 is its high time complexity.

The *Iterative Case Filtering algorithm* (ICF) algorithm [17] is based on the notions of *Coverage set* and *Reachable set*. The coverage set of an instance x is the set of instances in T whose distance from x is less than the distance between x and its nearest enemy. It is worth noting that this notion is analogous to the notion of *local set*, which we adopt in our algorithm. On the other hand, the reachable set of an instance x is the set of instances in T that have x in their respective coverage sets. In this ICF method, a given instance x is removed from S if $|Reachable(x)| > |Coverage(x)|$, i.e. when the number of the other instances that can classify x correctly is greater than the number of instances that x can classify correctly.

The *Local Density-based Instance Selection* (LDIS) algorithm [18] selects the instances with the highest local density in their neighbourhoods. LDIS searches for representative instances only among the instances of each class (separately). For this reason, it is not necessary to perform a *global search* in the whole data set. This simple algorithm is able to produce representative subsets of data, resulting in high accuracies for classification tasks.

In [7], the authors adopted the notion of *local sets* (LS) to design 3 complementary methods for instance selection. In this context, the local set of a given instance x is the set of instances contained in the largest hypersphere centered on x such that it does not contain instances from any other class. The first algorithm called *Local Set-based Smoother* (LSSm) was proposed to remove instances harmful for the classification accuracy, i.e. instances that misclassify more instances than those that they classify correctly. It uses two notions, namely, *usefulness* and *harmfulness* to guide the process. Hence, the usefulness $u(x)$ of a given instance x is the number of instances having x among the members of their local sets, whereas the harmfulness $h(x)$ is the number of instances having x as the nearest enemy. For each instance x in T, the algorithm includes x in S if $u(x) \geq h(x)$. Since the primary goal of LSSm is to remove harmful instances, its reduction rate is lower than most of the instance selection algorithms. The second algorithm called *Local Set-based Centroids Selector* (LSCo) firstly applies LSSm to remove noise and then applies LS-clustering [23] to identify clusters in T [24]. The algorithm keeps in S only the centroids of the resulting clusters [25], [26], [27]. Finally, the *Local Set Border Selector* (LSBo) algorithm uses at first LSSm to remove noise, and then, it computes the local set of every instance $\in T$. Next, the instances in T are sorted in the ascending order of the cardinality of their local sets. In the last step, LSBo verifies for each instance $x \in T$ if any member of its local set is contained in S, thus ensuring the proper classification of x. If this is not the case, x is included in S to ensure its correct classification. The time complexity of the 3 approaches is $O(|T|^2)$. Among these 3 algorithms, LSBo provides the best balance between reduction and accuracy.

Other instance selection approaches can be found in surveys such as [28], [2].

III. NOTATIONS

In this section, we introduce the notations adopted from our previous papers, e.g. [18], [29], [30], [31], [32], [4], and used throughout this paper:

- $T = \{x_1, x_2, ..., x_n\}$ is the non-empty set of n instances (or data objects) representing the original dataset to be reduced in the instance selection process.
- Each $x_i \in T$ is an $m - tuple$, such that $x_i = (x_{i1}, x_{i2}, ..., x_{im})$, where x_{ij} represents the value of the j-th feature of the instance x_i, for $1 \leq j \leq m$.
- $L = \{l_1, l_2, ..., l_p\}$ is the set of p class labels that are used to classify the instances in T, where each $l_i \in L$ represents a given class label.

- $l\colon T \rightarrow L$ is a function that maps a given instance $x_i \in T$ to its corresponding class label $l_j \in L$.
- $c\colon L \rightarrow 2^T$ is a function that maps a given class label $l_j \in L$ to a given set C, such that $C \subseteq T$, which represents the set of instances in T whose class is l_j. It is worth noting that $T = \bigcup_{l \in L} c(l)$. In this notation, 2^T represents the *powerset* of T, which is the set of all subsets of T, including the empty set and T itself.
- $d\colon T \times T \rightarrow \mathbb{R}$ is a *distance function* (or dissimilarity function) which maps two instances to a real number representing the distance (or dissimilarity) between them.
- $S = \{x_1, x_2, ..., x_q\}$ is a set of q instances such as $S \subseteq T$. It represents the reduced set of instances resulting from the instance selection process.

IV. ASBI Algorithm

The proposed approach called ASBI (*Approach for Selection of Border Instances*) has been designed to reduce data in order to keep only a small number of data instances which are representative and which could be used in tasks such as autonomous object modelling and recognition. Indeed, ASBI preserves only the most representative instances at the boundaries of each class. It assumes that the instances at the boundaries can represent sufficient information to distinguish between classes and to support the classification of novel instances in their respective classes. This selection criterion is also adopted by other instance selection approaches, such as [7], but our approach improves the one proposed in [7] by adopting additional constraints for selecting the minimal set of border instances.

Hence, the ASBI algorithm adopts also the notion of *local set* (LS) [17], as follows:

Definition 1. The *local set* of a given instance i, with $i \in T$, is represented by $LS(i)$ and is the set of all instances contained in the bigger hypersphere centered at i, in a way that only instances whose class is $c(i)$ are included in the hypersphere. Let's consider $x \in T$ to be the nearest instance of i such as $c(i) \neq c(x)$, then $CL(i) = \{y | d(i,y) < d(i,x)\}$.

The notion of *local set* is represented in Fig. 1. This example considers 2 classes (i.e. the white circles and the black circles). In this scenario, the local set of the instance A is $LS(A) = \{A, B, C, D, E, F, G\}$. It is worth noting that the instances I, J, K, L, and M cannot be included due to the constraints imposed by the instance R to the size of the hypersphere (represented by a dashed circle). A bigger hypersphere would include the instance R, and $c(R) \neq c(A)$. The notion of local set allows thus the definition of the notion of *internality*.

Definition 2. The *internality* of an individual i, with $i \in T$, is represented by $int(i)$ and is the cardinality of $LS(i)$, i.e. $int(i) = |LS(i)|$.

Closer to the border of its class is an individual, lower is its distance to instances of other classes and, therefore, lower is the cardinality of its local set. Thus, the internality of an instance can be used to estimate how close this instance is to the border of its class.

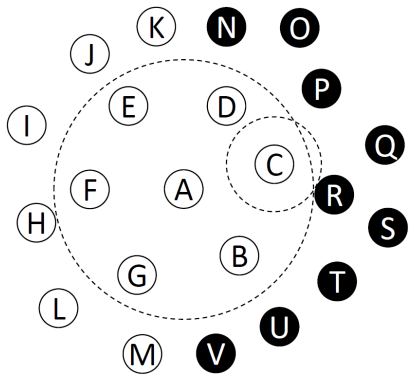

Fig. 1. Illustration of the notions of *internality* and *degree of potential noise*. The internality of A is $int(A) = |CL(A)| = |\{A, B, C, D, E, F, G\}| = 7$, while the *degree of potential noise* is $DPN(R) = |\{A, C\}| = 2$.

The notion of internality is illustrated in Fig. 1. In this case, the internality of A is given by $int(A) = |CL(A)| = |\{A, B, C, D, E, F, G\}| = 7$. On the other hand, the internality of C is given by $int(C) = |CL(C)| = |\{C\}| = 1$.

The ASBI algorithm adopts also the notion of *degree of potential noise* (DPN) of an instance.

Definition 3. The *degree of potential noise* of an instance i, with $i \in T$, is represented by $DPN(i)$ and is the number of instances in T which are classified in classes that are different from $c(i)$, where i is the closest instance in another class.

Figure 1 illustrates also the notion of degree of potential noise. In this example, $DPN(R) = 2$, since there are 2 instances (namely, A and C) in T that have R as the closest instance of a different class. It is worth noting that higher is the $DPN(i)$, higher is the possibility of the instance i being harmful for the classification of novel instances of its class.

Besides that, our approach adopts the notions of *coherence* and *selectivity*.

Definition 4. The *coherence* of an instance i, with $i \in T$, is represented by $coherence(i) = |\{x | x \in T \wedge i \in LS(x)\}|$ and is basically the number of instances in T whose local set includes i. It measures the degree of coherence of an individual with the information abstracted by its class.

Definition 5. The *selectivity* of an instance $i \in T$ is defined as:

$$selectivity(i) = \frac{coherence(i)}{internality(i)}. \quad (1)$$

In this context, higher is the $selectivity(i)$, higher is the priority of the individual i being considered as a candidate in the process of instance selection.

According to the *selectivity* definition, higher is the *coherence* and lower is the *internality* of an instance i, higher is the selectivity of i. Thus, this heuristic assigns higher values for *border instances*, since they have a higher value of *internality*. Moreover, among these border instances, this heuristic prioritizes those that are *more coherent* with their classes.

Algorithm 1: ASBI (Approach for Selection of Border Instances)

Input: A set T of instances.
Output: A set S of instances, such that $S \subseteq T$.
begin
  $T^{-} \leftarrow$ Removing noisy instances from T (using the algorithm LSSm);
  $candidates \leftarrow \emptyset$;
  foreach $A \in T^{-}$ **do**
    if $int(A) > DPN(A)$ **then**
      $candidates \leftarrow candidates \cup \{A\}$;
  Sorting the $candidates$ set, in a descending order, according to the *selectivity* of each candidate;
  $S \leftarrow \emptyset$;
  foreach $A \in candidates$ **do**
    if $LS(A) \cap S = \emptyset$ **then**
      $S \leftarrow S \cup \{A\}$;
  return S;

Considering all these notions, Algorithm 1 formalizes the *Approach for Selection of Border Instances* (ASBI). Our algorithm takes as input a set T of instances. Firstly, the algorithm applies the LSSm algorithm [7] to remove noisy instances from T. The outcome of this step is the set T^{-}. Secondly, the algorithm includes in the set $candidates$ each instance $A \in T^{-}$ whose *internality* is higher than its *degree of potential noise*. If A does not meet this requirement, that means that A is considered potentially harmful for classifying new instances and, for this reason, A is ignored. This can be viewed as an additional step of candidate filtering, which selects only instances that are more typical of their classes and that have lower potential of harming the classification. In the next step, the algorithm sorts the $candidates$ set in a descending order, according to the *selectivity* of each instance. Next, the algorithm initializes the set S as an empty set. This set is used for storing the instances in T^{-} that will be selected. In the final step, for each instance $A \in candidates$, the algorithm verifies if there are some instances in $LS(A)$ already included in S, i.e. the algorithm checks if the intersection between S and $LS(A)$) is not empty. If the intersection is empty, the instance A is included in S to ensure that A and other instances similar to A will be correctly classified. At the end, the algorithm returns the set S as the set of selected instances.

It is worth noting that ASBI algorithm is different from LSBo in two main aspects. On one hand, ASBI performs an additional step of noise removal, where instances whose *DPN* are greater or equal to their *internality* are removed. On the other hand, ASBI uses the notion of *selectivity* to prioritize the instances that are evaluated first. According to this latter criterion, the ASBI algorithm evaluates first the instances with higher coherence and lower internality. Hence, our approach prioritizes the selection of instances that are near to the borders of their classes, i.e. instances with low internality, but that can represent more information about their neighbours, since they have a high coherence.

The most expensive steps of the ASBI algorithm are the phase of noise removal and the process of building the local sets of each instance. The time complexity of the LSSm algorithm is $O(|T|^2)$. The time complexity for building the local sets of every instance in T^{-} is $O(|T^{-}|^2)$. Consequently, the time complexity of the ASBI algorithm is also $O(|T|^2)$. Thus, the time complexity of ASBI is equivalent to the time complexity of other well-known instance selection algorithms such as [17], [7].

V. Experiments

In order to evaluate our algorithm, we carried out two types of experiments. Hence, in Section V-A, we tested our ASBI algorithm on a generic classification task, considering 27 well-known datasets, while in Section V-B, we applied our ASBI algorithm to the autonomous object recognition task, using 2 challenging databases. For both experiments, the performance of our approach was compared with the ones of 6 state-of-art instance-selection (IS) algorithms, namely, LDIS, LSBo, DROP3, ICF, ENN, and LSSm.

In all experiments, we set $k = 3$ for DROP3, ENN, ICF, and LDIS, and we adopted the distance function $d\colon T \times T \rightarrow \mathbb{R}$, defined as follows:

$$d(x, y) = \sum_{j=1}^{m} \theta_j(x, y), \tag{2}$$

with

$$\theta_j(x, y) = \begin{cases} \alpha(x_j, y_j), & \text{if } j \text{ is a categorical feature,} \\ |x_j - y_j|, & \text{if } j \text{ is a numerical feature,} \end{cases} \tag{3}$$

where

$$\alpha(x_j, y_j) = \begin{cases} 1, & \text{if } x_j \neq y_j, \\ 0, & \text{if } x_j = y_j. \end{cases} \tag{4}$$

To evaluate the performance of the instance selection algorithms, we considered 3 standard metrics, namely, *accuracy*, *reduction* and *effectiveness* [7], [18].

The first 2 metrics are defined as follows:

$$accuracy = \frac{Success(Test)}{|Test|} \tag{5}$$

and

$$reduction = \frac{|T| - |S|}{|T|}, \tag{6}$$

where $Test$ is a given set of instances that are selected to be tested in a classification task, and $Success(Test)$ is the number of instances in $Test$ correctly classified in the classification task [7].

The third metric called *effectiveness* is then defined as follows:

$$effectiveness = accuracy \times reduction. \tag{7}$$

Indeed, the effectiveness is a measure of the degree to which an instance selection algorithm is successful in producing a small set of instances that allows a high classification accuracy of new instances [18].

TABLE I. DETAILS OF THE DATASETS USED IN THE CLASSIFICATION TASK.

Data set	Instances	Attributes	textbfClasses
Audiology	226	70	24
Breast cancer	286	10	2
Cardiotocography	2126	21	10
Cars	1728	6	4
Dermatology	358	35	6
Diabetes	768	9	2
E. Coli	336	8	8
Glass	214	10	7
Heart statelog	270	14	2
Ionosphere	351	35	2
Iris	150	5	3
Landsat	4435	37	6
Letter	20000	17	26
Lung cancer	32	57	3
Lymph	148	19	4
Mushroom	8124	23	2
Optdigits	5620	65	10
Page-blocks	5473	11	5
Parkinsons	195	23	2
Promoters	106	58	2
Segment	2310	20	7
Soybean	683	36	19
Spambase	4601	58	2
Splice	3190	61	3
Voting	435	17	2
Wine	178	14	3
Zoo	101	18	7

A. Experiment 1

In the first set of experiments, we run IS algorithms, such as ASBI, LDIS, LSBo, DROP3, ICF, ENN, and LSSm, to compare their performance in context of a classification task in 27 well-known, distinct datasets obtained from the UCI Machine Learning Repository[2], i.e. *audiology*, *breast cancer*, *cardiotocography*, *cars*, *dermatology*, *diabetes*, *e. coli*, *glass*, *heart statelog*, *ionosphere*, *iris*, *landsat*, *letter*, *lung cancer*, *lymph*, *mushroom*, *optdigits*, *page-blocks*, *parkinsons*, *genetic promoters*, *segment*, *soybean* (which combines the large soybean dataset and its test dataset), *spambase*, *splice junction gene sequences*, *voting*, *wine*, and *zoo*. Details of the data sets that were used are presented in Table I.

To evaluate the classification *accuracy* of new instances in each respective dataset, we adopted a SVM (*support vector machine*) classifier [33]. Following [34], we adopted the WEKA[3] 3.8 implementation of SVM that uses the sequential minimal optimization algorithm [35] for training the classifier, with the standard parametrization: $c = 1.0$, $tolerance Parameter = 0.001$, $epsilon = 1.0E-12$, using a polynomial kernel and a multinomial logistic regression model with a ridge estimator as calibrator.

Besides that, the accuracy, reduction and effectiveness were evaluated in a *n-fold cross-validation* scheme [18], [32], where $n = 10$. Thus, the dataset is at first randomly partitioned in 10 equally sized subsamples. From these subsamples, a single subsample is selected as test data ($Test$), and the union of the remaining 9 subsamples is considered as the *initial training set* (ITS). Next, an instance selection algorithm is applied to reduce the ITS and thus to produce the *reduced training set* (RTS). At this point, the *reduction* of the dataset can be measured. Finally, the RTS is used as the training set for the classifier, to classify the instances in $Test$. At this point, the accuracy achieved by the SVM classifier can be measured using RTS as the training set. This process is repeated 10 times, with each subsample used once as $Test$. The 10 values of accuracy and reduction are averaged to produce the *average accuracy* (AA) and *average reduction* (AR), respectively [29]. The *average effectiveness* is calculated by considering AA and AR. Tables II-IV report the *average reduction*, the *average accuracy* and the *average effectiveness*, respectively, achieved in each combination of dataset and instance selection algorithm, using the SVM classifier. The best results for each dataset are marked in bold typeface.

[2]http://archive.ics.uci.edu/ml/

[3]http://www.cs.waikato.ac.nz/ml/index.html

TABLE II. COMPARISON OF THE *reduction* ACHIEVED BY EACH ALGORITHM FOR EACH DATASET IN THE CLASSIFICATION TASK.

Algorithm	ASBI	LDIS	LSBo	DROP3	ICF	ENN	LSSm	Average
Audiology	0.73	**0.75**	0.60	0.70	**0.75**	0.36	0.08	0.57
Breast cancer	0.84	**0.88**	0.73	0.77	0.85	0.30	0.13	0.64
Cardiotocography	0.80	**0.86**	0.70	0.70	0.71	0.31	0.13	0.60
Cars	0.85	0.84	0.74	**0.87**	0.81	0.18	0.12	0.63
Dermatology	0.81	**0.87**	0.73	0.64	0.70	0.15	0.13	0.58
Diabetes	0.84	**0.91**	0.75	0.77	0.86	0.30	0.13	0.65
E. Coli	0.89	**0.90**	0.82	0.72	0.86	0.16	0.09	0.64
Glass	0.82	**0.90**	0.72	0.75	0.69	0.32	0.14	0.62
Heart statelog	0.81	**0.92**	0.69	0.73	0.78	0.30	0.14	0.62
Ionosphere	0.89	0.90	0.86	0.80	**0.92**	0.11	0.04	0.65
Iris	**0.95**	0.89	0.93	0.71	0.59	0.04	0.06	0.60
landsat	0.91	**0.92**	0.88	0.72	0.90	0.10	0.05	0.64
Letter	**0.87**	0.83	0.83	0.68	0.81	0.05	0.04	0.59
Lung cancer	0.73	**0.84**	0.56	0.76	0.70	0.58	0.20	0.63
Lymph	0.82	**0.90**	0.74	0.71	0.81	0.20	0.11	0.61
Mushroom	**0.99**	0.87	**0.99**	0.86	0.94	0.00	0.00	0.67
Optdigits	**0.93**	0.91	0.90	0.71	0.92	0.02	0.02	0.63
Pageblocks	**0.97**	0.86	0.96	0.71	0.95	0.04	0.03	0.65
Parkinsons	**0.90**	0.80	0.86	0.71	0.75	0.15	0.11	0.61
Promoters	0.73	**0.83**	0.60	0.60	0.67	0.19	0.05	0.53
Segment	**0.93**	0.83	0.91	0.69	0.83	0.04	0.04	0.61
Soybean	**0.87**	0.78	0.83	0.69	0.57	0.09	0.05	0.55
Spambase	**0.86**	0.83	0.81	0.74	0.80	0.16	0.08	0.61
Splice	0.70	**0.81**	0.59	0.65	0.75	0.23	0.05	0.54
Voting	**0.91**	0.79	0.88	0.79	0.93	0.08	0.04	0.63
Wine	0.85	**0.87**	0.78	0.72	0.79	0.23	0.10	0.62
Zoo	**0.89**	0.62	0.88	0.65	0.30	0.06	0.06	0.50
Average	**0.83**	0.82	0.77	0.70	0.75	0.16	0.08	0.59

TABLE III. COMPARISON OF THE *accuracy* ACHIEVED BY EACH ALGORITHM FOR EACH DATASET IN THE CLASSIFICATION TASK, WITH THE SVM CLASSIFIER.

Algorithm	ASBI	LDIS	LSBo	DROP3	ICF	ENN	LSSm	Average
Audiology	0.71	0.55	0.77	0.65	0.69	0.68	**0.81**	0.70
Breast cancer	0.69	0.65	0.66	0.71	0.66	**0.72**	0.71	0.69
Cardiotocography	0.63	0.62	0.62	0.64	0.65	0.66	**0.67**	0.64
Cars	0.85	0.84	**0.91**	0.86	**0.91**	0.90	**0.91**	0.88
Dermatology	0.95	0.96	0.96	0.96	0.96	0.96	**0.97**	0.96
Diabetes	0.76	0.74	0.75	0.76	0.76	**0.77**	**0.77**	0.76
E. Coli	0.74	0.80	0.78	0.81	0.77	0.81	**0.82**	0.79
Glass	0.46	0.49	0.45	0.52	0.50	0.53	**0.55**	0.50
Heart statelog	0.81	0.79	0.82	0.81	0.81	**0.84**	**0.84**	0.82
Ionosphere	0.82	0.81	0.53	0.80	0.78	0.87	**0.89**	0.79
Iris	0.53	0.82	0.47	0.91	0.69	**0.96**	**0.96**	0.76
landsat	0.85	0.84	0.85	0.86	0.85	**0.87**	**0.87**	0.86
Letter	0.73	0.75	0.74	0.80	0.75	**0.84**	**0.84**	0.78
Lung cancer	**0.45**	0.32	0.40	0.33	0.38	0.38	0.40	0.38
Lymph	0.81	0.80	**0.84**	0.79	0.81	0.80	**0.84**	0.81
Mushroom	0.96	**1.00**	0.97	1.00	0.96	**1.00**	**1.00**	0.98
Optdigits	0.97	0.96	0.98	0.98	0.97	**0.99**	**0.99**	0.98
Pageblocks	0.93	0.93	0.92	0.93	0.93	**0.94**	**0.94**	0.93
Parkinsons	0.84	0.83	0.82	0.84	0.82	**0.87**	0.86	0.84
Promoters	0.85	0.76	0.86	0.85	0.79	**0.87**	0.86	0.83
Segment	0.85	0.89	0.81	0.91	0.90	**0.93**	0.92	0.89
Soybean	0.85	0.86	0.85	0.91	0.92	0.93	**0.94**	0.89
Spambase	0.89	0.88	0.89	**0.90**	**0.90**	0.89	**0.90**	0.89
Splice	0.93	0.91	0.92	**0.94**	0.93	**0.94**	**0.94**	0.93
Voting	**0.95**	0.93	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	0.95
Wine	0.94	0.94	**0.97**	0.96	0.94	0.96	**0.97**	0.95
Zoo	0.89	**0.95**	0.91	0.90	0.92	0.92	0.94	0.92
Average	0.78	0.78	0.76	0.80	0.78	**0.82**	**0.82**	0.79

TABLE IV. COMPARISON OF THE *effectiveness* ACHIEVED BY EACH ALGORITHM FOR EACH DATASET IN THE CLASSIFICATION TASK, WITH THE SVM CLASSIFIER.

Algorithm	ASBI	LDIS	LSBo	DROP3	ICF	ENN	LSSm	Average
Audiology	0.51	0.41	0.46	0.46	**0.52**	0.25	0.06	0.38
Breast cancer	**0.58**	0.57	0.48	0.55	0.56	0.21	0.10	0.44
Cardiotocography	0.50	**0.53**	0.43	0.45	0.46	0.21	0.08	0.38
Cars	0.72	0.71	0.68	**0.75**	0.74	0.16	0.11	0.55
Dermatology	0.77	**0.84**	0.70	0.62	0.67	0.15	0.12	0.55
Diabetes	0.64	**0.67**	0.57	0.59	0.65	0.23	0.10	0.49
E. Coli	0.66	**0.72**	0.64	0.58	0.66	0.13	0.08	0.50
Glass	0.38	**0.44**	0.32	0.39	0.34	0.17	0.08	0.30
Heart statelog	0.66	**0.73**	0.57	0.59	0.63	0.26	0.12	0.51
Ionosphere	**0.73**	**0.73**	0.45	0.65	0.72	0.10	0.04	0.49
Iris	0.51	**0.73**	0.44	0.64	0.40	0.04	0.06	0.40
landsat	**0.77**	**0.77**	0.75	0.62	**0.77**	0.08	0.04	0.54
Letter	**0.63**	0.62	0.61	0.54	0.60	0.04	0.03	0.44
Lung cancer	**0.33**	0.27	0.23	0.25	0.27	0.22	0.08	0.24
Lymph	0.67	**0.73**	0.62	0.56	0.65	0.16	0.09	0.50
Mushroom	0.95	0.87	**0.96**	0.86	0.91	0.00	0.00	0.65
Optdigits	**0.90**	0.88	0.88	0.70	0.89	0.02	0.02	0.61
Pageblocks	**0.91**	0.81	0.89	0.66	0.89	0.04	0.03	0.60
Parkinsons	**0.76**	0.66	0.71	0.59	0.62	0.13	0.10	0.51
Promoters	0.62	**0.63**	0.52	0.50	0.53	0.17	0.05	0.43
Segment	**0.80**	0.73	0.74	0.63	0.74	0.03	0.03	0.53
Soybean	**0.74**	0.67	0.71	0.62	0.53	0.08	0.05	0.49
Spambase	**0.77**	0.74	0.72	0.67	0.72	0.14	0.07	0.55
Splice	0.66	**0.74**	0.54	0.61	0.70	0.22	0.05	0.50
Voting	0.87	0.73	0.84	0.75	**0.88**	0.07	0.04	0.60
Wine	0.80	**0.82**	0.75	0.69	0.74	0.22	0.09	0.59
Zoo	0.79	0.59	**0.80**	0.58	0.28	0.05	0.05	0.45
Average	**0.67**	0.66	0.61	0.58	0.61	0.12	0.06	0.48

Table II shows that ASBI achieves the highest *reduction* in several datasets and has the highest average reduction rate. Table III shows that ENN and LSSm achieve the highest *accuracy* in most of the datasets, but they do not provide high reduction rates, since they were designed for removing noisy instances. On the other hand, Tables II-III show that in cases where the achieved accuracy by ASBI is lower than the accuracy provided by other algorithms, this is compensated by a high reduction. Table IV shows that, in several datasets, ASBI has the highest *effectiveness* as well as the highest average effectiveness. We can also observe that ASBI provides the highest reduction rates and the best trade-off between both accuracy and reduction (represented by the effectiveness).

It is also worth noting that ASBI does not have any free parameter that should be provided by the user. This could be an advantage, especially for autonomous systems.

B. Experiment 2

This set of experiments consists in applying instance selection algorithms to autonomous object modelling in context of the autonomous agent's visual object recognition task. For this purpose, we used 2 challenging, online-available databases[4] that are called *CMU10_3D* and *CMU_KO8*, respectively, and that contain instances of different visual objects [36].

In particular, *CMU10_3D* has a training dataset with 250 images with 2D ground truth of 10 classes of *grocery* objects and a testing dataset with 50 images per object class, while *CMU_KO8* dataset is split into 200 training images with 2D ground truth of 8 classes of *household* items and 800 images, with 100 instances per object class.

In order to evaluate the instance selection algorithms performance when applied to autonomous object modelling and recognition, each respective training visual data is first pre-processed using a modified version of visual object model, which was proposed by [12], in order to extract attributes such as object's name, height, width, centroid, perimeter, and area. Then, IS algorithms are run separately on this resulting training dataset in order to reduce the number of instances required to automatically build the corresponding, effective object model for each object class. At this stage, the autonomous object modelling is done. Next, the machine-vision algorithm called template matching algorithm, which was introduced in [12], can be applied on the entire testing dataset in order to perform the autonomous object recognition task, using separately each of the reduced object models based on the respective IS algorithms.

[4]http://www.cs.cmu.edu/∼ehsiao/datasests.html

TABLE V. COMPARISON OF THE *reduction* ACHIEVED BY EACH ALGORITHM FOR EACH DATASET IN THE OBJECT RECOGNITION TASK.

Algorithm	ASBI	LDIS	LSBo	DROP3	ICF	ENN	LSSm	Average
CMU10 3D	0.85	**0.88**	0.73	0.75	0.53	0.30	0.14	0.60
CMU KO8	**0.89**	**0.89**	0.84	0.76	0.65	0.18	0.11	0.61
Average	0.87	**0.88**	0.78	0.76	0.59	0.24	0.12	0.61

TABLE VI. COMPARISON OF THE *accuracy* ACHIEVED BY EACH ALGORITHM FOR EACH DATASET IN THE OBJECT RECOGNITION TASK.

Algorithm	ASBI	LDIS	LSBo	DROP3	ICF	ENN	LSSm	Average
CMU10 3D	**0.72**	0.69	0.69	0.71	0.66	0.65	0.65	0.68
CMU KO8	**0.74**	0.72	0.70	0.69	0.67	0.66	0.66	0.69
Average	**0.73**	0.71	0.69	0.70	0.67	0.66	0.65	0.69

TABLE VII. COMPARISON OF THE *effectiveness* ACHIEVED BY EACH ALGORITHM FOR EACH DATASET IN THE OBJECT RECOGNITION TASK.

Algorithm	ASBI	LDIS	LSBo	DROP3	ICF	ENN	LSSm	Average
CMU10 3D	**0.62**	0.61	0.50	0.53	0.35	0.20	0.09	0.41
CMU KO8	**0.65**	0.64	0.58	0.52	0.44	0.12	0.07	0.43
Average	**0.64**	0.63	0.54	0.53	0.39	0.16	0.08	0.42

Results of this second type of experiments are reported in Tables V-VII, where the *average accuracy* (AA), *average reduction* (AR), and *average effectiveness* (AE) are computed respectively as the average of the obtained scores for each class, measure, dataset and IS algorithm; the best results for each dataset being marked in bold typeface.

From Tables V-VII, we can observe that the ASBI algorithm provides the best results in terms of visual object modelling, since ASBI achieves the highest scores for the average accuracy compared to the other IS algorithms. Furthermore, ASBI does not give the most reduced set of instances in every case, but achieves the best average effectiveness, i.e. the best trade-off between accuracy and reduction.

VI. CONCLUSIONS

In this paper, we have proposed a new algorithm for instance selection, which enhances the notion of local set and which is well suited for the autonomous systems' object modelling. Indeed, our algorithm called ASBI (*Approach for Selection of Border Instances*) selects only the most representative instances at the borders of their classes. This approach was successfully tested on 29 well-known datasets. ASBI shows the best average *effectiveness*, i.e. the best trade-off between *accuracy* and *reduction*, when compared to 6 state-of-art algorithms. The experiments also suggest that the strategies adopted by ASBI are useful to build compact object models for the automated, visual object recognition task.

ACKNOWLEDGMENT

The authors would like to thank IBM Research Brazil for the support of this work.

REFERENCES

[1] J. A. Olvera-Lopez, J. A. Carrasco-Ochoa, J. F. Martinez-Trinidad, and J. Kittler. A review of instance selection methods. *Artificial Intelligence Review*, 34:133–143, 2010.

[2] S. Garcia, J. Luengo, and F. Herrera. *Data Preprocessing in Data Mining*. Springer, 2015.

[3] C.-H. Chou, B-H. Kuo, and F. Chang. The generalized condensed nearest neighbor rule as a data reduction method. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, volume 2, pages 556–559, 2006.

[4] J. L. Carbonera and M. Abel. An efficient prototype selection algorithm based on dense spatial partitions. In *Proceedings of Artificial Intelligence and Soft Computing (ICAISC)*, pages 288–300, 2018.

[5] H. Liu and H. Motoda. On issues of instance selection. *Data Mining and Knowledge Discovery*, 6(2):115–130, 2002.

[6] W.-C. Lin, C.-F. Tsai, S.-W. Ke, C.-W. Hung, and W. Eberlem. Learning to detect representative data for large scale instance selection. *Journal of Systems and Software*, 106:1–8, 2015.

[7] E. Leyva, A. Gonzalez, and R. Perez. Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective. *Pattern Recognition*, 48(4):1523–1537, 2015.

[8] A. Zakeri and A. Hokmabadi. Efficient feature selection method using real-valued grasshopper optimization algorithm. *Expert Systems with Applications*, 119:61–72, 2019.

[9] H. Liu, Y. Liu, L. Huang, F. Sun, and D. Guo. Discovery of topical object in image collections. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1886–1892, 2015.

[10] T. Faulhammer, R. E. Ambru, C. Burbridge, M. Zillich, J. Folkesson, N. Hawes, P. Jensfelt, and M. Vincze. Autonomous learning of object models on a mobile robot. *IEEE Robotics and Automation Letters*, 2(1):26–33, 2017.

[11] J. I. Olszewska. Active contour based optical character recognition for automated scene understanding. *Neurocomputing*, 161:65–71, 2015.

[12] J. I. Olszewska. "Where Is My Cup?" – Fully Automatic Detection and Recognition of Textureless Objects in Real-World Images. In *Proceedings of IAPR International Conference on Computer Analysis of Images and Patterns (CAIP). LNCS 9256, Part I*, pages 501–512, 2015.

[13] J. Calzado, A. Lindsay, C. Chen, G. Samuels, and J. I. Olszewska. SAMI: Interactive, multi-sense robot architecture. In *Proceedings of the IEEE International Conference on Intelligent Engineering Systems*, pages 317–322, 2018.

[14] J. I. Olszewska. Designing transparent and autonomous intelligent vision systems. In *Proceedings of International Conference on Agents and Artificial Intelligence*, pages 850–856, 2019.

[15] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-2(3):408–421, 1972.

[16] D. R. Wilson and A. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.

[17] H. Brighton and C. Mellish. Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery*, 6(2):153–172, 2002.

[18] J. L. Carbonera and M. Abel. A density-based approach for instance selection. In *Proceedings of IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 768–774, 2015.

[19] M. Quinn and J. I. Olszewska. British sign language recognition in the wild based on multi-class svm. In *Proceedings of Federated Conference on Computer Science and Information*, pages 81–86, 2019.

[20] P. E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14:515–516, 1968.

[21] G. W. Gates. Reduced nearest neighbor rule. *IEEE Transactions on Information Theory*, 18(3):431–433, 1972.

[22] I. Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6:448–452, 1976.

[23] Y. Caises, A. Gonzalez, E. Leyva, and R. Perez. Combining instance selection methods based on data characterization: An approach to increase their effectiveness. *Information Sciences*, 181(20):4780–4798, 2011.

[24] L. Jing, M. K. Ng, and J. Z. Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):1026–1041, 2007.

[25] Z. He, X. Xu, and S. Deng. Attribute value weighting in k-modes clustering. *Expert Systems with Applications*, 38(12):15365–15369, 2011.

[26] L. Bai, J. Liang, C. Dang, and F. Cao. A cluster centers initialization method for clustering categorical data. *Expert Systems with Applications*, 39(9):8022–8029, 2012.

[27] S. Khan and A. Ahmad. Cluster center initialization algorithm for k-modes clustering. *Expert Systems with Applications*, 40(18):7444–7456, 2013.

[28] J. Hamidzadeh, R. Monsefi, and H. S. Yazdi. Irahc: Instance reduction algorithm using hyperrectangle clustering. *Pattern Recognition*, 48(5):1878–1889, 2015.

[29] J. L. Carbonera and M. Abel. Efficient prototype selection supported by subspace partitions. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 921–928, 2017.

[30] J. L. Carbonera. An efficient approach for instance selection. In *Proceedings of the International Conference on Big Data Analytics and Knowledge Discovery (DaWaK)*, pages 228–243, 2017.

[31] J. L. Carbonera and M. Abel. An efficient prototype selection algorithm based on spatial abstraction. In *Proceedings of the International Conference on Big Data Analytics and Knowledge Discovery (DaWaK)*, pages 177–192, 2018.

[32] J. L. Carbonera and M. Abel. Efficient instance selection based on spatial abstraction. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 286–292, 2018.

[33] M. F. Akay. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 36(2):3240–3247, 2009.

[34] I. M. Anwar, K. M. Salama, and A. M. Abdelbar. Instance selection with ant colony optimization. *Procedia Computer Science*, 53:248–256, 2015.

[35] J. C. Platt. 12 fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods*, pages 185–208, 1999.

[36] E. Hsiao and M. Hebert. Gradient Networks: Explicit shape matching without extracting edges. In *Proceedings of AAAI International Conference on Artificial Intelligence*, pages 417–423, 2013.