Mathematical and Computational Applications

MDPI

Article

Windowing as a Sub-Sampling Method for Distributed Data Mining

David Martínez-Galicia [1,*], Alejandro Guerra-Hernández [1], Nicandro Cruz-Ramírez [1], Xavier Limón [2] and Francisco Grimaldo [3]

[1] Centro de Investigación en Inteligencia Artificial, Universidad Veracruzana, Sebastián Camacho No 5, Xalapa, Veracruz, México 91000, Mexico; aguerra@uv.mx (A.G.-H.); ncruz@uv.mx (N.C.-R.)
[2] Facultad de Estadística e Informática, Universidad Veracruzana, Av. Xalapa s/n, Xalapa, Veracruz, México 91000, Mexico; hlimon@uv.mx
[3] Departament d'Informàtica, Universitat de València, Avinguda de la Universitat, s/n, Burjassot-València, 46100 València, Spain; francisco.grimaldo@uv.es
* Correspondence: davidgalicia@outlook.es

Received: 31 May 2020; Accepted: 29 June 2020; Published: 30 June 2020

Abstract: Windowing is a sub-sampling method, originally proposed to cope with large datasets when inducing decision trees with the ID3 and C4.5 algorithms. The method exhibits a strong negative correlation between the accuracy of the learned models and the number of examples used to induce them, i.e., the higher the accuracy of the obtained model, the fewer examples used to induce it. This paper contributes to a better understanding of this behavior in order to promote windowing as a sub-sampling method for Distributed Data Mining. For this, the generalization of the behavior of windowing beyond decision trees is established, by corroborating the observed negative correlation when adopting inductive algorithms of different nature. Then, focusing on decision trees, the windows (samples) and the obtained models are analyzed in terms of Minimum Description Length (MDL), Area Under the ROC Curve (AUC), Kulllback–Leibler divergence, and the similitude metric Sim1; and compared to those obtained when using traditional methods: random, balanced, and stratified samplings. It is shown that the aggressive sampling performed by windowing, up to 3% of the original dataset, induces models that are significantly more accurate than those obtained from the traditional sampling methods, among which only the balanced sampling is comparable in terms of AUC. Although the considered informational properties did not correlate with the obtained accuracy, they provide clues about the behavior of windowing and suggest further experiments to enhance such understanding and the performance of the method, i.e., studying the evolution of the windows over time.

Keywords: sub-sampling; windowing; distributed data mining

1. Introduction

Windowing is a sub-sampling method that enabled the decision tree inductive algorithms ID3 [1–3] and C4.5 [4,5] to cope with large datasets, i.e., those whose size precludes loading them in memory. Algorithm 1 defines the method: First, a window is created by extracting a small random sample of the available examples in the full dataset. The main step consists of inducing a model with that window and of testing it on the remaining examples, such that all misclassified examples are moved to the window. This step iterates until a stop condition is reached, e.g., all the available examples are correctly classified or a desired level of accuracy is reached.

Algorithm 1 Windowing.

Require: *Examples* {The original training set}
Ensure: *Model* {The induced model}
1: $Window \leftarrow sample(Examples)$
2: $Examples \leftarrow Examples - Window$
3: **repeat**
4: $stopCond \leftarrow true$
5: $model \leftarrow induce(Window)$
6: **for** $example \in Examples$ **do**
7: **if** $classify(model, example) \neq class(example)$ **then**
8: $Window \leftarrow Window \cup \{example\}$
9: $Examples \leftarrow Examples - \{example\}$
10: $stopCond \leftarrow false$
11: **end if**
12: **end for**
13: **until** *stopCond*
14: **return** *model*

Despite Wirth and Catlett [6] publishing an early critic about the computational cost of windowing and its inability to deal with noisy domains, Fürnkranz [7] argues that this method still offers three advantages: a) it copes well with memory limitations, reducing considerably the number of examples required to induce a model of acceptable accuracy; b) it offers an efficiency gain by reducing the time of convergence, specially when using a separate-and-conquer inductive algorithm, as FOIL [8], instead of the divide-and-conquer algorithms such as ID3 and C4.5., and; c) it offers an accuracy gain, specially in noiseless datasets, possibly explained by the fact that learning from a subset of examples may often result in a less over-fitting theory.

Even when the lack of memory is not usually an issue nowadays, similar concerns arise when mining big and/or distributed data, i.e., the impossibility or inconvenience of using all the available examples to induce models. Windowing has been used as the core of a set of strategies for Distributed Data Mining (DDM) [9] obtaining good accuracy results, consistent with the expected achievable accuracy and number of examples required by the method. On the contrary, efficiency suffers for large datasets as the cost of testing the models in the remaining examples is not negligible (i.e., the for loop in Algorithm 1, line 6), although it can be alleviated by using GPUs [10]. More relevant for this paper is the fact that these Windowing-based strategies based on J48, the Weka [11] implementation of C4.5, show a strong correlation (-0.8175845) between the accuracy of the learned decision trees and the number of examples used to induce them, i.e., the higher the accuracy obtained, the fewer the number of examples used to induce the model. The windows in this method can be seen as samples and reducing the size of the training sets, even up to a 95% of the available training data, still enables accuracy values above 95%.

These promising results encourage the adoption of windowing as a sub-sampling method for Distributed Data Mining. However, they suggest some issues that must be solved for such adoption. The first one is the generalization of windowing beyond decision trees. Does windowing behave similarly when using different models and inductive algorithms? The first contribution of this paper is to corroborate the correlation between accuracy and the size of the window, i.e., the number of examples used to induce the model, when using inductive algorithms of different nature, showing that the advantages of windowing as a sub-sampling method can be generalized beyond decision trees. The second issue is the need of a deeper understanding of the behavior of windowing. How is that such a big reduction in the number of training examples, maintains acceptable levels of accuracy? This is particularly interesting as we have pointed out that high levels of accuracy correlate with smaller windows. The second contribution of the paper is thus to approach such a question in terms of the informational properties of both the windows and the models obtained by the method. These properties do not unfortunately correlate with the obtained accuracy of windowing and suggest the

study of the evolution of the windows over as future work. Finally, a comparison with traditional methods as random, stratified, and balanced samplings, provides a better understanding of windowing and evaluates its adoption as an alternative sampling method. Under equal conditions, i.e., same original full dataset and size of the sample, windowing shows to be significantly more accurate than the traditional samplings and comparable to balanced sampling in terms of AUC. The paper is organized as follows: Section 2 introduces the adopted materials and methods; Section 3 presents the obtained results; and Section 4 discusses conclusions and future work.

2. Materials and Methods

This section describes the implementation of windowing used in this work, as included in JaCa-DDM; the datasets used in experimentation; and the experiments themselves.

2.1. Windowing in JaCa-DDM

Because of our interest in Distributed Data Mining settings, JaCa-DDM (https://github.com/xl666/jaca-ddm) was adopted to run our experiments. This tool [9] defines a set of windowing-based strategies using J48, the Weka [11] implementation of C4.5, as inductive algorithm. Among them, the Counter strategy is the most similar to the original formulation of windowing, with the exception of:

1. The dataset may be distributed in different sites, instead of the traditional approach based on a single dataset in a single site.
2. The loop for collecting the misclassified examples to be added to the window is performed by a set of agents using copies of the model distributed among the available sites, in a round-robin fashion.
3. The initial window is a stratified sample, instead of a random one.
4. An auto-adjustable stop criteria is combined with a configurable maximum number of iterations.

The configuration of the strategy (Table 1) used for all the experiments reported in this paper, is adopted from the literature [10].

Table 1. Configuration of the counter strategy. Adopted from Limón *et al.* [10].

Parameter	Value
Classifier	J48
Pruning	True
Number of nodes	8
Maximum number of rounds	15
Initial percentage for the window	0.20
Validation percentage for the test	0.25
Change step of accuracy every round	0.35

2.2. Datsets

Table 2 lists the datasets selected from the UCI [12] and MOA [13] repositories to conduct our experiments. They vary in the number of instances, attributes, and class' values; as well as in the type of the attributes. Some of them are affected by missing values. The literature [10] reports experiments on larger datasets, up to 4.8×10^6 instances, exploiting GPUs. However, datasets with higher dimensions are problematic, e.g., imdb-D with 1002 attributes does not converge using the Counter strategy.

Table 2. Datasets, adopted from UCI and MOA.

Dataset	Instances	Attributes	Attribute type	Missing values	Classes
Adult	48842	15	Mixed	Yes	2
Australian	690	15	Mixed	No	2
Breast	683	10	Numeric	No	2
Diabetes	768	9	Mixed	No	2
Ecoli	336	8	Numeric	No	8
German	1000	21	Mixed	No	2
Hypothyroid	3772	30	Mixed	Yes	4
Kr-vs-kp	3196	37	Numeric	No	2
Letter	20000	17	Mixed	No	26
Mushroom	8124	23	Nominal	Yes	2
Poker-lsn	829201	11	Mixed	No	10
Segment	2310	20	Numeric	No	7
Sick	3772	30	Mixed	Yes	2
Splice	3190	61	Nominal	No	3
Waveform5000	5000	41	Numeric	No	3

2.3. Experiments

Two experiments were designed to cope with the issues approached by this work, i.e., the generalization of windowing beyond decision trees; a deeper understanding of its behavior in informational terms; and the comparison with traditional sampling methods. All of them were executed on a Intel Core i5-8300H at 2.3GHz, up to 3.9GHz with 8Gb DDR4. 8 distributed sites were simulated on this machine. JaCa-DDM also allows the adoption of real distributed sites over a network, but the aspects of windowing we study here, are not affected by simulating distribution.

2.3.1. On the Generalization of windowing

The first experiment seeks to corroborate the correlation between the accuracy of the learned model and the amount of instances used to induce the model. It attempts to provide practical evidence about the generalization of windowing. For this, different Weka classifiers are adopted that replace J48. JaCa-DDM allows easy replacement and configuration of the new classifier artifacts of the system, namely:

Naive Bayes. A probabilistic classifier based on Bayes' theorem with a strong assumption of independence among attributes [14].

jRip. An inductive rule learner based on RIPPER that builds a set of rules while minimizing the amount of error [15].

Multilayer-perceptron. A multi-layer perceptron trained by backpropagation with sigmoid nodes except for numeric classes, in which case the output nodes become unthresholded linear units [16].

SMO. An implementation of John Platt's sequential minimal optimization algorithm for training a support vector classifier [17].

All classifiers are induced by running a 10-fold stratified cross-validation on each dataset, then observing the average accuracy of the obtained models and the average percentage of the original dataset used to induce the model, i.e., 100% means the full original dataset was used to create the window.

2.3.2. On the Properties of Samples and Models Obtained by Windowing

The second experiment pursues a deeper understanding of the informational properties of the computed models, as well as those of the samples obtained by Windowing, i.e., the final windows. For this, given the positive results of the first experiment, we focus exclusively on decision trees (J48), for

which different metrics to evaluate performance, complexity and data compression are well known. They include:

- The model accuracy defined as the percentage of correctly classified instances.

$$\frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

  where TP, TN, FP and FN respectively stand for the true positive, true negative, false positive, and false negative classifications using the test data.
- The metric AUC defined as the probability of a random instance to be correctly classified [18].

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{2}$$

  Even though this measure was conceived for binary classification problems. Foster Provost [19] proposes an implementation for multi-class problems based in the weighted average of AUC metrics for every class using a one-against-all approach, and the weight for every AUC is calculated as the class' appearance frequency in the data $p(c_i)$.

$$AUC_{total} = \sum_{c_i \in C} AUC(c_i) \cdot p(c_i) \tag{3}$$

- The MDL principle states that the best model to infer from a dataset is the one which minimizes the sum of the length of the model $L(H)$, and the length of the data when encoded using the theory as a predictor for the data $L(D|H)$ [20].

$$MDL = L(H) + L(D|H) \tag{4}$$

  For decision trees, Quinlan [21] proposes the next definition:

  1. The number of bits needed to encode a tree is:

$$L(H) = n_{nodes} * (1 + ln(n_{attributes})) + n_{leaves}(1 + ln(n_{c1asses})) \tag{5}$$

     where n_{nodes}, $n_{attributes}$, n_{leaves} and $n_{c1asses}$ stand for the number of nodes, attributes, leaves and classes. This encoding uses a recursive top-down, depth-first procedure, where a tree which is not a leaf is encoded by a sequence of 1, the attribute code at his root, and the respective encodings of the subtrees. If a tree or subtree is a leaf, its enconding is a sequence of 0, and the class code.
  2. The number of bits needed to encode the data using the decision tree is:

$$L(D|H) = \sum_{l \in Leaves} log_2(b + 1) + log_2\left(\binom{n}{k}\right) \tag{6}$$

     where n is the number of instances, k is the number of positives instances for binary classification and b is a known a priori upper bound on k, typically $b = n$. For non-binary classification, Quinlan proposes a iterative approach where exceptions are sorted by their frequency, and then codified with the previous formula.

- The Kullback–Leibler divergence (D_{KL}) [22] is defined as:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) log_2 \left(\frac{P(x)}{Q(x)} \right) \quad (7)$$

where P and Q are probability distributions for the full dataset and the window, both are defined on the same probability space X, and x represents a class in the distribution. Instead of using a model to represent a conditional distribution of variables, as usual, we focus on the class distribution, computed as the marginal probability. Values closer to zero reflect higher similarity.

- Sim_1 [23] is a similarity measure between datasets defined as:

$$sim_1(D_i, D_j) = \frac{|Item(D_i) \cap Item(D_j)|}{|Item(D_i) \cup Item(D_j)|} \quad (8)$$

where D_i is the window and D_j is the full dataset; and $Item(D)$ denotes the set of pairs attribute-value occurring in D. Values closer to one reflect higher similarity.

These metrics are used to compare the sample (the window) and the model computed by windowing, against those obtained as follows, once a random sample of the original data set is reserved as test set:

- Without sampling, using all the available data to induce the model.
- By Random sampling, where any instance has the same selection probability [24].
- By Stratified random sampling, where the instances are subdivided by their class into subgroups, the number of selected instances per subgroup is defined as the division of the sample size by the number of instances [24].
- By Balanced random sampling, as stratified random sampling, the instances are subdivided by their class into subgroups, but the number of selected instances per subgroup is defined as the division of the sample size by the number of subgroups, this allows the same number of instances per class [24].

Ten repetitions of 10-fold stratified cross-validation are run on each dataset. For a fair comparison, all the samples have the size of the window being compared. Statistical validity of the results is established following the method proposed by Demšar [25]. This approach enables the comparison of multiple algorithms on multiple data sets. It is based on the use of the Friedman test with a corresponding post-hoc test. Let R_i^j be the rank of the j^{th} of k algorithms on the i^{th} of N data sets. The Friedman test [26,27] compares the average ranks of algorithms, $R_j = \frac{1}{N} \sum_i R_i^j$. Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks R_j should be equal, the Friedman statistic:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (9)$$

is distributed according to χ_F^2 with $k-1$ degrees of freedom, when N and k are big enough ($N > 10$ and $k > 5$). For a smaller number of algorithms and data sets, exact critical values have been computed [28]. Iman and Davenport [29] showed that Friedman's χ_F^2 is undesirably conservative and derived an adjusted statistic:

$$F_f = \frac{(N-1) \times \chi_F^2}{N \times (k-1) - \chi_F^2} \quad (10)$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom. If the null hypothesis of similar performances is rejected, then the Nemenyi post-hoc test is realized

for pairwise comparisons. The performance of two classifiers is significantly different if their corresponding average ranks differ by at least the critical difference:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{11}$$

where critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$.

For the comparison of multiple classifiers, the results of the post-hoc tests can be visually represented with a simple critical distance diagram. This type of visualization will be described in the Statistical Tests in Section 3.

3. Results

Results are organized accordingly to the following issues:

- Generalization of the behavior of windowing, i.e., high accuracy correlating with fewer training examples used to induce the model, when other inductive algorithms, apart of J48, are adopted.
- Informational properties of the samples obtained by different methods, based on the Kullback–Leibler divergence and the attribute-value similitude.
- Properties of the models induced with the samples, in terms of their size, complexity, and data compression, which supplies information about their data fitting capacity.
- Predictive performance of the induced models in terms of accuracy and the AUC.
- Statistical tests about significant gains produced by windowing using the former metrics.

3.1. Windowing Generalization

Figure 1 shows a strong negative correlation between the number of training instances used to induce the models, expressed as a percentage with respect to the totality of available examples, and the accuracy of the induced model. Such correlation exists, independently of the adopted inductive algorithm. These results are consistent with the behavior of windowing when using J48, as reported in the literature [9] and corroborates that under windowing, in general, the models with higher accuracy use less examples to be induced.

However, accuracy is affected by the adopted inductive algorithm, e.g., Hypothyroid is approached very well by jRip (99.23 ± 0.48 of accuracy) requiring few examples (5% of the full dataset); while Multilayer-Perceptron is not quite successful in this case (92.26 ± 2.75 of accuracy) requiring more examples (24%). This behavior is also observed between SMO and jRip for Waveform5000. These observations motivated analyzing the properties of the samples and induced models, as described in the following subsections. Table 3 shows the accuracy results in detail and Table 4 shows the number of examples used to induce the models, best results are highlighted in gray. Appendix A shows the accuracy values for models without using windowing under a 10-fold cross-validation. Windowing accuracies are comparable to those obtained without using windowing. Table 7 also corroborate this this for the J48 classifier.

Table 3. Average windowing accuracy under a 10-fold cross validation (na = not available).

	J48		NB		jRip		MP		SMO	
Adult	86.17 ±	0.55	84.54 ±	0.62	na		na		na	
Australian	85.21 ±	4.77	85.79 ±	4.25	85.94 ±	3.93	81.74 ±	6.31	85.80 ±	4.77
Breast	94.42 ±	3.97	97.21 ±	2.34	95.31 ±	2.75	95.45 ±	3.14	96.33 ±	3.12
Diabetes	73.03 ±	3.99	76.03 ±	4.33	71.74 ±	7.67	72.12 ±	4.00	76.04 ±	3.51
Ecoli	82.72 ±	6.81	83.93 ±	7.00	81.22 ±	6.63	82.12 ±	7.49	84.53 ±	4.11
German	71.10 ±	5.40	75.20 ±	2.82	70.20 ±	3.85	69.60 ±	4.84	75.80 ±	3.12
Hypothyroid	99.46 ±	0.17	95.36 ±	0.99	99.23 ±	0.48	92.26 ±	2.75	94.30 ±	0.53
Kr-vs-kp	99.15 ±	0.66	96.65 ±	0.84	98.46 ±	0.95	98.72 ±	0.54	96.62 ±	0.75
Letter	85.79 ±	1.24	69.28 ±	1.26	85.31 ±	1.06	na		na	
Mushroom	100.00 ±	0.00	99.80 ±	0.16	100.00 ±	0.00	100.00 ±	0.00	100.0 ±	0.00
Poker-lsn	99.75 ±	0.07	60.02 ±	0.42	na		na		na	
Segment	96.53 ±	1.47	84.24 ±	1.91	95.54 ±	1.55	96.10 ±	1.15	92.42 ±	1.87
Sick	98.64 ±	0.53	96.34 ±	1.44	97.93 ±	0.95	96.32 ±	1.04	96.71 ±	0.77
Splice	94.04 ±	0.79	95.32 ±	1.07	92.75 ±	2.11	na		92.41 ±	1.34
Waveform5000	73.06 ±	2.55	82.36 ±	1.64	77.02 ±	1.59	na		85.94 ±	1.32

Table 4. Average size of the final window (the sample) under a 10-fold cross validation, in terms of the percentage of the full dataset used for induction (na = not available).

	J48		NB		jRip		MP		SMO	
Adult	0.30 ±	0.01	0.21 ±	0.00	na		na		na	
Australian	0.31 ±	0.02	0.25 ±	0.01	0.33 ±	0.02	0.39 ±	0.04	0.27 ±	0.01
Breast	0.17 ±	0.01	0.06 ±	0.00	0.14 ±	0.01	0.11 ±	0.01	0.09 ±	0.01
Diabetes	0.54 ±	0.05	0.40 ±	0.02	0.52 ±	0.04	0.48 ±	0.03	0.42 ±	0.02
Ecoli	0.38 ±	0.03	0.27 ±	0.01	0.40 ±	0.03	0.31 ±	0.03	0.29 ±	0.02
German	0.56 ±	0.04	0.43 ±	0.01	0.59 ±	0.02	0.58 ±	0.02	0.47 ±	0.02
Hypothyroid	0.05 ±	0.00	0.12 ±	0.01	0.05 ±	0.00	0.24 ±	0.01	0.12 ±	0.01
Kr-vs-kp	0.08 ±	0.01	0.16 ±	0.01	0.13 ±	0.00	0.08 ±	0.00	0.12 ±	0.00
Letter	0.35 ±	0.02	0.38 ±	0.00	0.39 ±	0.01	na		na	
Mushroom	0.03 ±	0.00	0.04 ±	0.00	0.03 ±	0.00	0.02 ±	0.00	0.02 ±	0.00
Poker-lsn	0.05 ±	0.00	0.59 ±	0.00	na		na		na	
Segment	0.16 ±	0.01	0.22 ±	0.01	0.19 ±	0.01	0.14 ±	0.01	0.18 ±	0.00
Sick	0.07 ±	0.00	0.10 ±	0.01	0.08 ±	0.00	0.11 ±	0.01	0.10 ±	0.00
Splice	0.26 ±	0.01	0.11 ±	0.00	0.25 ±	0.01	na		0.19 ±	0.00
Waveform5000	0.59 ±	0.02	0.22 ±	0.01	0.52 ±	0.00	na		0.26 ±	0.01

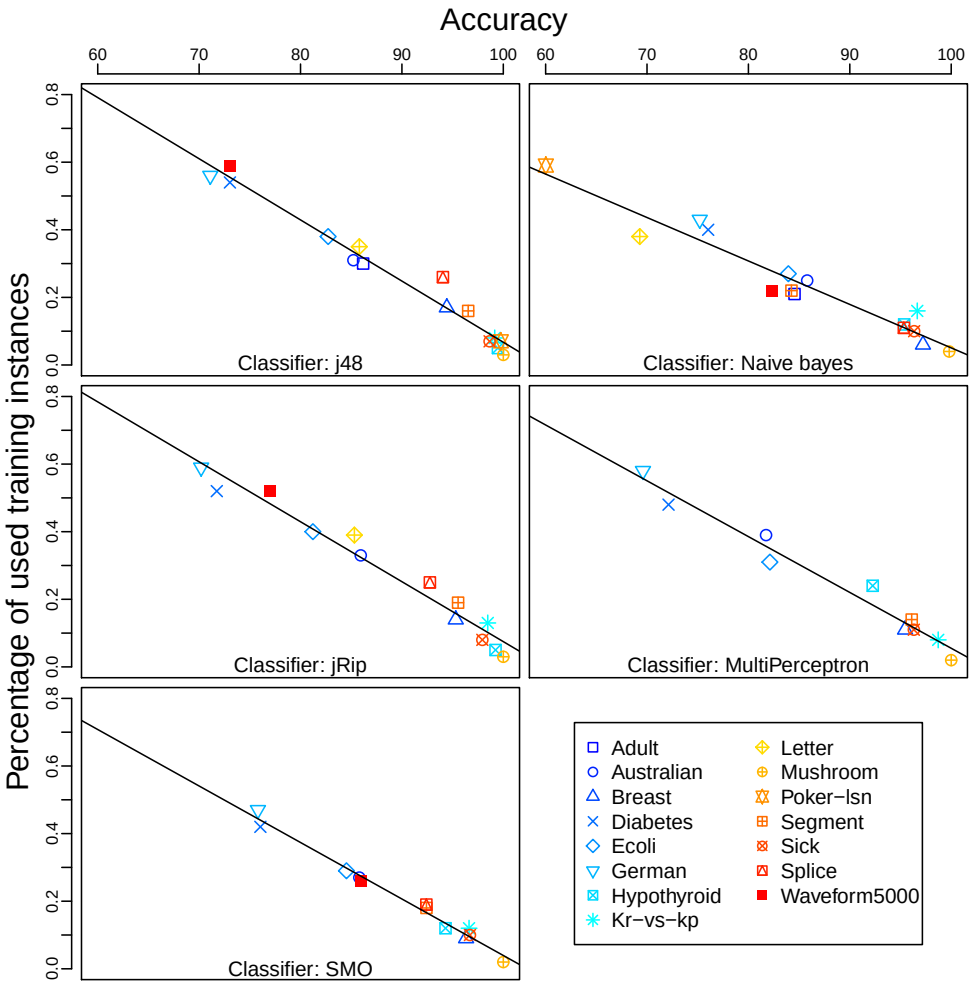

Figure 1. Correlation between accuracy and percentage of used training examples when windowing. J48 $= -0.98$, NB $= -0.96$, jRip $= -0.98$, MP $= -0.98$, and SMO $= -0.99$. In general, the models with higher accuracy use less examples to be induced.

Large datasets such as as Adult, Letter, Poker-Lsn, Splice, and Waveform5000 did not finish on reasonable time when using jRip, Multilayer-Perceptron and SMO, with and without windowing. In such cases, results are reported as not available (na). This might be solved by running the experiments in a real cluster of 8 nodes, instead of simulating the sites in a single machine, as done here, but it is not relevant for the purposes of this work. In the following results, Poker-lsn dataset was excluded because the cross-validations runs do not finish on a reasonable time, this might be solved with more computational power. The results were kept this way because they illustrate that some classifiers exhibit a computational cost which precludes convergence.

3.2. Samples Properties

For each dataset considered in this work, Table 5 shows some properties of the samples obtained by the following methods: windowing, as described before; the Full-Dataset under a 10-folds cross-validation (90% of all available data); and the random, stratified, and balanced samplings. Properties include the size of the sample in terms of the number of instances; the standard deviation of the class distribution (*St.Dv.C.D.*); and two measures of similarity between the samples and the original dataset: The Kullback–Leibler divergence and the metric sim_1. With the exception of Full-Dataset, the size of the rest of the samples is determined by the windowing method and its autostop method. For the sake of fairness, windowing is executed first and the size of the sample obtained in this way is

adopted for the rest of the sampling methods. Reductions in the size of the training set are as big as 97% of the available data (Hypothyroid).

According to Kullback–Leibler Divergence, windowing is the method that skews more the original class distribution in non-balanced datasets. It is also observed that the class distribution on the windows is more balanced, and its effectiveness probably depends on the number of available examples for the minority classes. For instance, Full-Dataset shows an unbalanced class distribution ($St.Dv.C.D. = 0.449$) in Hypothyroid, while windowing got a coefficient of 0.293. Windowing can not completely balance the number of examples per class since the percentage of the available examples for the minority classes are around of 5%. The random sampling, the Full-Dataset, and the stratified sampling do not tend to modify the class distribution. However, it does not seem to be a correlation between this coefficient and the obtained accuracy.

Full-Dataset is, without surprise, the sample that gathers more attribute/values pairs from the original data, since it uses 90% of the available data. It is included in the results exclusively for comparison with the rest of the sampling methods. Table 5 also show that windowing tends to collect more information content in most of the datasets compared with all the sampling, this is probably result of the heuristic nature of windowing. There are some datasets, like Breast and German, where all the techniques have one as the measured value of $Sim1$. Unfortunately, as in the previous case, this notion of similarity neither seems to correlate with the observed accuracy, for instance, as mentioned, for Breast and German all the sampling methods gathers all the original pairs attribute-value ($Sim_1 = 1.0$), but while the accuracy obtained for Breast is around 95%, when using German it is around 71%. In concordance with these results, the window for Breast uses 17% of the available examples, while German uses 64% (Table 5).

Table 5. Samples properties.

Dataset	Method	Instances		St. Dv. C.D.		KL Div		Sim1	
Adult	Windowing	14502.840 ±	574.266	0.083 ±	0.004	0.128 ±	0.004	0.386 ±	0.012
Adult	Full-Dataset	43957.800 ±	0.402	0.369 ±	0.000	0.000 ±	0.000	0.935 ±	0.001
Adult	Random-sampling	14502.840 ±	574.266	0.374 ±	0.049	0.005 ±	0.005	0.418 ±	0.013
Adult	Stratified-sampling	14502.840 ±	574.266	0.369 ±	0.000	0.000 ±	0.000	0.418 ±	0.013
Adult	Balanced-sampling	14502.840 ±	574.266	0.000 ±	0.000	0.206 ±	0.000	0.400 ±	0.013
Australian	Windowing	215.440 ±	14.363	0.031 ±	0.020	0.017 ±	0.008	0.999 ±	0.006
Australian	Full-Dataset	621.000 ±	0.000	0.078 ±	0.001	0.000 ±	0.000	0.999 ±	0.005
Australian	Random-sampling	215.440 ±	14.363	0.080 ±	0.047	0.004 ±	0.005	0.986 ±	0.016
Australian	Stratified-sampling	215.440 ±	14.363	0.078 ±	0.004	0.000 ±	0.000	0.986 ±	0.016
Australian	Balanced-sampling	215.440 ±	14.363	0.001 ±	0.002	0.009 ±	0.000	0.987 ±	0.016
Breast	Windowing	109.210 ±	14.732	0.043 ±	0.030	0.086 ±	0.031	1.000 ±	0.000
Breast	Full-Dataset	614.700 ±	0.461	0.212 ±	0.000	0.000 ±	0.000	1.000 ±	0.000
Breast	Random-sampling	109.210 ±	14.732	0.224 ±	0.107	0.019 ±	0.017	1.000 ±	0.000
Breast	Stratified-sampling	109.210 ±	14.732	0.215 ±	0.007	0.000 ±	0.000	1.000 ±	0.000
Breast	Balanced-sampling	109.210 ±	14.732	0.003 ±	0.003	0.066 ±	0.003	1.000 ±	0.000
Diabetes	Windowing	436.260 ±	27.768	0.087 ±	0.022	0.025 ±	0.009	0.751 ±	0.028
Diabetes	Full-Dataset	691.200 ±	0.402	0.213 ±	0.001	0.000 ±	0.000	0.954 ±	0.004
Diabetes	Random-sampling	436.260 ±	27.768	0.214 ±	0.021	0.001 ±	0.001	0.763 ±	0.028
Diabetes	Stratified-sampling	436.260 ±	27.768	0.215 ±	0.002	0.000 ±	0.000	0.766 ±	0.028
Diabetes	Balanced-sampling	436.260 ±	27.768	0.001 ±	0.001	0.067 ±	0.001	0.770 ±	0.028
Ecoli	Windowing	126.640 ±	8.579	0.109 ±	0.005	0.182 ±	0.055	0.761 ±	0.026
Ecoli	Full-Dataset	302.400 ±	0.492	0.145 ±	0.000	0.001 ±	0.001	0.979 ±	0.006
Ecoli	Random-sampling	126.640 ±	8.579	0.147 ±	0.010	0.007 ±	0.010	0.763 ±	0.025
Ecoli	Stratified-sampling	126.640 ±	8.579	0.154 ±	0.004	0.013 ±	0.003	0.758 ±	0.027
Ecoli	Balanced-sampling	126.640 ±	8.579	0.099 ±	0.004	0.113 ±	0.028	0.781 ±	0.028
German	Windowing	584.750 ±	25.308	0.119 ±	0.012	0.041 ±	0.006	1.000 ±	0.000
German	Full-Dataset	900.000 ±	0.000	0.283 ±	0.000	0.000 ±	0.000	1.000 ±	0.000
German	Random-sampling	584.750 ±	25.308	0.284 ±	0.022	0.001 ±	0.001	1.000 ±	0.000
German	Stratified-sampling	584.750 ±	25.308	0.283 ±	0.001	0.000 ±	0.000	1.000 ±	0.000
German	Balanced-sampling	584.750 ±	25.308	0.055 ±	0.022	0.079 ±	0.015	1.000 ±	0.000

Continued on next page

Dataset	Method	Instances		St. Dv. C.D.		KL Div		Sim1	
Hypothyroid	Windowing	151.680 ±	9.619	0.293 ±	0.017	0.262 ±	0.047	0.428 ±	0.017
Hypothyroid	Full-Dataset	3394.800 ±	0.402	0.449 ±	0.000	0.000 ±	0.000	0.979 ±	0.005
Hypothyroid	Random-sampling	151.680 ±	9.619	0.580 ±	0.149	0.212 ±	0.103	0.387 ±	0.020
Hypothyroid	Stratified-sampling	151.680 ±	9.619	0.516 ±	0.007	0.000 ±	0.001	0.387 ±	0.013
Hypothyroid	Balanced-sampling	151.680 ±	9.619	0.191 ±	0.004	0.668 ±	0.023	0.435 ±	0.016
Kr-vs-kp	Windowing	242.550 ±	18.425	0.050 ±	0.036	0.010 ±	0.012	0.998 ±	0.004
Kr-vs-kp	Full-Dataset	2876.400 ±	0.492	0.031 ±	0.000	0.000 ±	0.000	0.999 ±	0.004
Kr-vs-kp	Random-sampling	242.550 ±	18.425	0.221 ±	0.130	0.106 ±	0.099	0.975 ±	0.013
Kr-vs-kp	Stratified-sampling	242.550 ±	18.425	0.032 ±	0.003	0.000 ±	0.000	0.977 ±	0.009
Kr-vs-kp	Balanced-sampling	242.550 ±	18.425	0.001 ±	0.001	0.001 ±	0.000	0.977 ±	0.008
Letter	Windowing	7390.450 ±	491.435	0.008 ±	0.000	0.037 ±	0.002	0.989 ±	0.006
Letter	Full-Dataset	18000.000 ±	0.000	0.001 ±	0.000	0.000 ±	0.000	0.999 ±	0.002
Letter	Random-sampling	7390.450 ±	491.435	0.007 ±	0.001	0.022 ±	0.009	0.983 ±	0.008
Letter	Stratified-sampling	7390.450 ±	491.435	0.000 ±	0.000	0.000 ±	0.000	0.985 ±	0.007
Letter	Balanced-sampling	7390.450 ±	491.435	0.001 ±	0.000	0.001 ±	0.000	0.984 ±	0.006
Mushroom	Windowing	219.490 ±	16.871	0.043 ±	0.033	0.004 ±	0.005	0.968 ±	0.021
Mushroom	Full-Dataset	7311.600 ±	0.492	0.025 ±	0.000	0.000 ±	0.000	1.000 ±	0.000
Mushroom	Random-sampling	219.490 ±	16.871	0.504 ±	0.244	2.083 ±	1.852	0.833 ±	0.072
Mushroom	Stratified-sampling	219.490 ±	16.871	0.026 ±	0.004	0.000 ±	0.000	0.903 ±	0.032
Mushroom	Balanced-sampling	219.490 ±	16.871	0.002 ±	0.002	0.001 ±	0.000	0.902 ±	0.033
Segment	Windowing	371.280 ±	27.458	0.104 ±	0.008	0.390 ±	0.076	0.279 ±	0.015
Segment	Full-Dataset	2079.000 ±	0.000	0.000 ±	0.000	0.000 ±	0.000	0.938 ±	0.003
Segment	Random-sampling	371.280 ±	27.458	0.050 ±	0.007	0.105 ±	0.144	0.310 ±	0.019
Segment	Stratified-sampling	371.280 ±	27.458	0.002 ±	0.001	0.000 ±	0.000	0.315 ±	0.018
Segment	Balanced-sampling	371.280 ±	27.458	0.002 ±	0.001	0.000 ±	0.000	0.315 ±	0.018
Sick	Windowing	264.600 ±	17.420	0.305 ±	0.028	0.233 ±	0.032	0.565 ±	0.019
Sick	Full-Dataset	3394.800 ±	0.402	0.621 ±	0.000	0.000 ±	0.000	0.979 ±	0.005
Sick	Random-sampling	264.600 ±	17.420	0.623 ±	0.066	0.015 ±	0.014	0.483 ±	0.018
Sick	Stratified-sampling	264.600 ±	17.420	0.623 ±	0.002	0.000 ±	0.000	0.483 ±	0.014
Sick	Balanced-sampling	264.600 ±	17.420	0.002 ±	0.001	0.665 ±	0.002	0.495 ±	0.014
Splice	Windowing	835.300 ±	29.689	0.072 ±	0.011	0.036 ±	0.009	0.969 ±	0.043
Splice	Full-Dataset	2871.000 ±	0.000	0.169 ±	0.047	0.000 ±	0.000	0.987 ±	0.034
Splice	Random-sampling	835.300 ±	29.689	0.161 ±	0.000	0.014 ±	0.013	0.890 ±	0.060
Splice	Stratified-sampling	835.300 ±	29.689	0.161 ±	0.001	0.000 ±	0.000	0.862 ±	0.036
Splice	Balanced-sampling	835.300 ±	29.689	0.001 ±	0.001	0.104 ±	0.001	0.871 ±	0.046
Waveform-5000	Windowing	3263.590 ±	330.000	0.006 ±	0.004	0.000 ±	0.000	0.940 ±	0.018
Waveform-5000	Full-Dataset	4500.000 ±	0.000	0.004 ±	0.000	0.000 ±	0.000	0.983 ±	0.001
Waveform-5000	Random-sampling	3263.590 ±	330.000	0.018 ±	0.010	0.002 ±	0.002	0.932 ±	0.019
Waveform-5000	Stratified-sampling	3263.590 ±	330.000	0.004 ±	0.000	0.000 ±	0.000	0.932 ±	0.019
Waveform-5000	Balanced-sampling	3263.590 ±	330.000	0.000 ±	0.000	0.000 ±	0.000	0.932 ±	0.019

3.3. Model Complexity and Data Compression

Table 6 shows the results for the MDL, calculated using the test dataset. Respecting the number of bits required to encode a tree ($L(H)$), Windowing and Full-Dataset tend to induce more complex models, i.e, trees with more nodes. This is probably because windowing favors the search for more difficult patterns in the set of available instances, which require more complex models to be expressed. Respecting the number of bits required to encode the test data, given the induced decision tree, ($L(D|H)$) a better compression is achieved using windowing and Full-Dataset than when using the traditional samplings. Big differences in data compression using windowing are exhibit in datasets like Mushroom, Segment, and Waveform-5000. One possible explanation for this is that instances gathered by sampling techniques do not capture the data nature because of their random selection and the small number of instances in the sample.

The sum of the former metrics, the MDL, reports bigger models in most of the datasets when using windowing and Full-Dataset. This result does not represent an advantage, but properties such as the predictive performance also play an important role in model selection.

Table 6. Model complexity and test data compression.

Dataset	Method	L(H)		L(D | H)		MDL	
Adult	Windowing	1361.599 ±	465.850	2366.019 ±	59.709	3727.618 ±	483.653
Adult	Full-Dataset	2077.010 ±	282.565	2374.002 ±	49.985	4451.012 ±	270.561
Adult	Random-sampling	1009.386 ±	276.429	2420.278 ±	56.458	3429.664 ±	264.703
Adult	Stratified-sampling	1031.172 ±	181.155	2410.870 ±	49.932	3442.042 ±	186.437
Adult	Balanced-sampling	1351.736 ±	265.668	2423.024 ±	44.271	3774.759 ±	274.906
Australian	Windowing	77.299 ±	29.067	41.284 ±	6.849	118.582 ±	30.088
Australian	Full-Dataset	66.820 ±	16.934	41.044 ±	6.711	107.864 ±	17.430
Australian	Random-sampling	45.151 ±	18.592	41.820 ±	6.916	86.971 ±	19.120
Australian	Stratified-sampling	50.313 ±	22.016	41.836 ±	6.776	92.149 ±	21.220
Australian	Balanced-sampling	44.603 ±	22.878	42.327 ±	6.764	86.929 ±	22.830
Breast	Windowing	46.541 ±	13.199	25.904 ±	4.584	72.445 ±	12.435
Breast	Full-Dataset	58.757 ±	7.942	25.338 ±	5.280	84.095 ±	8.195
Breast	Random-sampling	22.301 ±	6.555	29.008 ±	7.229	51.309 ±	7.316
Breast	Stratified-sampling	23.991 ±	6.915	28.631 ±	6.720	52.622 ±	8.350
Breast	Balanced-sampling	22.767 ±	7.801	28.191 ±	5.710	50.959 ±	8.137
Diabetes	Windowing	59.000 ±	37.207	65.437 ±	5.227	124.437 ±	37.477
Diabetes	Full-Dataset	126.620 ±	46.019	64.383 ±	5.161	191.003 ±	45.988
Diabetes	Random-sampling	95.960 ±	38.989	65.674 ±	4.884	161.634 ±	39.119
Diabetes	Stratified-sampling	94.940 ±	39.261	64.354 ±	5.965	159.294 ±	39.505
Diabetes	Balanced-sampling	104.840 ±	36.621	65.263 ±	5.003	170.103 ±	36.829
Ecoli	Windowing	99.328 ±	23.152	29.959 ±	7.767	129.287 ±	23.257
Ecoli	Full-Dataset	144.454 ±	19.804	27.648 ±	6.460	172.102 ±	18.623
Ecoli	Random-sampling	69.348 ±	16.853	33.969 ±	9.853	103.317 ±	15.614
Ecoli	Stratified-sampling	65.678 ±	16.214	34.174 ±	10.710	99.852 ±	16.457
Ecoli	Balanced-sampling	83.869 ±	20.904	30.357 ±	7.087	114.226 ±	20.376
German	Windowing	315.252 ±	60.182	82.866 ±	5.220	398.118 ±	60.077
German	Full-Dataset	287.566 ±	54.049	83.857 ±	5.339	371.423 ±	53.413
German	Random-sampling	211.627 ±	51.692	83.245 ±	5.156	294.871 ±	51.783
German	Stratified-sampling	212.684 ±	54.545	83.006 ±	5.125	295.689 ±	53.830
German	Balanced-sampling	238.184 ±	51.813	84.412 ±	5.352	322.596 ±	51.356
Hypothyroid	Windowing	84.812 ±	19.108	28.291 ±	6.449	113.102 ±	20.727
Hypothyroid	Full-Dataset	122.317 ±	10.791	27.105 ±	6.877	149.422 ±	10.562
Hypothyroid	Random-sampling	15.667 ±	15.278	189.232 ±	110.454	204.899 ±	96.402
Hypothyroid	Stratified-sampling	30.645 ±	6.465	67.493 ±	22.683	98.138 ±	22.336
Hypothyroid	Balanced-sampling	45.353 ±	10.448	61.502 ±	18.798	106.854 ±	18.199
Kr-vs-kp	Windowing	198.034 ±	14.570	69.919 ±	4.871	267.953 ±	14.944
Kr-vs-kp	Full-Dataset	219.807 ±	16.870	69.345 ±	4.277	289.152 ±	17.014
Kr-vs-kp	Random-sampling	64.438 ±	18.816	98.961 ±	21.032	163.399 ±	21.636
Kr-vs-kp	Stratified-sampling	72.664 ±	18.341	92.724 ±	15.119	165.388 ±	15.947
Kr-vs-kp	Balanced-sampling	73.848 ±	18.721	91.842 ±	14.262	165.690 ±	15.840
Letter	Windowing	11862.644 ±	473.112	1248.697 ±	64.017	13111.341 ±	453.031
Letter	Full-Dataset	12431.372 ±	180.896	1165.793 ±	38.869	13597.165 ±	182.617
Letter	Random-sampling	7020.909 ±	385.222	1473.635 ±	81.356	8494.544 ±	358.576
Letter	Stratified-sampling	7102.767 ±	358.000	1461.702 ±	80.161	8564.469 ±	328.131
Letter	Balanced-sampling	7126.843 ±	381.507	1449.106 ±	76.567	8575.949 ±	354.232
Mushroom	Windowing	79.249 ±	7.033	76.881 ±	4.163	156.130 ±	7.189
Mushroom	Full-Dataset	77.237 ±	0.600	79.510 ±	1.744	156.747 ±	1.810
Mushroom	Random-sampling	18.228 ±	19.552	461.838 ±	353.124	480.066 ±	337.153
Mushroom	Stratified-sampling	31.126 ±	14.101	114.606 ±	23.525	145.732 ±	20.201
Mushroom	Balanced-sampling	31.879 ±	15.063	113.501 ±	22.427	145.380 ±	17.422
Segment	Windowing	348.723 ±	34.369	81.656 ±	10.719	430.379 ±	33.528
Segment	Full-Dataset	365.928 ±	22.569	79.045 ±	9.609	444.973 ±	22.295
Segment	Random-sampling	142.987 ±	22.538	135.754 ±	31.843	278.741 ±	31.578
Segment	Stratified-sampling	142.715 ±	18.438	126.640 ±	24.516	269.356 ±	26.762
Segment	Balanced-sampling	141.267 ±	17.852	127.325 ±	23.254	268.591 ±	26.010
Sick	Windowing	170.530 ±	26.600	50.476 ±	8.212	221.005 ±	26.977
Sick	Full-Dataset	182.701 ±	22.491	42.346 ±	7.910	225.047 ±	20.038
Sick	Random-sampling	21.786 ±	16.605	80.715 ±	38.277	102.501 ±	24.810
Sick	Stratified-sampling	31.126 ±	6.768	55.199 ±	13.736	86.325 ±	15.387

Continued on next page

Dataset	Method	L(H)		L(D|H)		MDL	
Sick	Balanced-sampling	57.996 ±	17.446	60.045 ±	9.531	118.040 ±	18.444
Splice	Windowing	725.951 ±	53.364	181.187 ±	11.871	907.139 ±	53.195
Splice	Full-Dataset	745.146 ±	51.142	179.689 ±	11.014	924.834 ±	52.532
Splice	Random-sampling	425.144 ±	52.153	187.097 ±	21.631	612.240 ±	47.209
Splice	Stratified-sampling	443.339 ±	51.337	188.061 ±	19.286	631.400 ±	48.312
Splice	Balanced-sampling	419.763 ±	41.676	188.473 ±	20.593	608.236 ±	40.687
Waveform-5000	Windowing	2418.668 ±	215.760	363.799 ±	56.499	2782.467 ±	224.433
Waveform-5000	Full-Dataset	2615.956 ±	94.305	415.810 ±	20.601	3031.766 ±	92.381
Waveform-5000	Random-sampling	1957.647 ±	203.398	413.447 ±	24.548	2371.094 ±	202.636
Waveform-5000	Stratified-sampling	1957.202 ±	199.174	417.104 ±	26.348	2374.306 ±	196.151
Waveform-5000	Balanced-sampling	1966.554 ±	193.650	417.152 ±	28.133	2383.706 ±	190.987

3.4. Predictive Performance

Table 7 shows the predictive performance in terms of accuracy and the AUC. Even though the random, stratified and balanced samplings usually induce simpler models, the decision trees do not seem to be more general than their windowing and Full-Dataset counterparts. In other words, the predictive ability of decision trees induced with the traditional samplings are, most of the time, lower than the models induced using windowing and Full-Dataset. Models induced with windowing have the same accuracy as those obtained by Full-Dataset and, sometimes, they even show a higher accuracy, e.g., waveform-500. In terms of AUC, windowing and Full-Dataset were the best samples, but the balanced sampling is pretty close to their performance.

Table 7. Predictive performance.

Dataset	Method	Test Acc		Test AUC	
Adult	Windowing	86.355 ±	0.889	78.227 ±	1.161
Adult	Full-Dataset	86.074 ±	0.390	77.080 ±	0.823
Adult	Random-sampling	85.516 ±	0.423	76.131 ±	2.021
Adult	Stratified-sampling	85.677 ±	0.401	76.680 ±	0.885
Adult	Balanced-sampling	80.489 ±	0.722	81.956 ±	0.580
Australian	Windowing	85.710 ±	4.355	85.471 ±	4.411
Australian	Full-Dataset	86.536 ±	3.969	86.239 ±	4.041
Australian	Random-sampling	85.101 ±	4.375	84.849 ±	4.517
Australian	Stratified-sampling	85.391 ±	4.164	85.142 ±	4.266
Australian	Balanced-sampling	85.536 ±	3.925	85.584 ±	3.854
Breast	Windowing	94.829 ±	2.804	94.368 ±	3.117
Breast	Full-Dataset	95.533 ±	2.674	95.058 ±	2.830
Breast	Random-sampling	92.696 ±	3.821	91.687 ±	4.739
Breast	Stratified-sampling	92.783 ±	3.485	91.956 ±	3.982
Breast	Balanced-sampling	92.433 ±	3.558	92.301 ±	3.627
Diabetes	Windowing	74.161 ±	4.864	70.041 ±	5.654
Diabetes	Full-Dataset	74.756 ±	4.661	71.211 ±	5.027
Diabetes	Random-sampling	72.280 ±	4.520	68.602 ±	5.403
Diabetes	Stratified-sampling	73.222 ±	5.113	70.254 ±	5.721
Diabetes	Balanced-sampling	71.018 ±	5.222	71.726 ±	4.937
Ecoli	Windowing	82.777 ±	6.353	88.848 ±	4.134
Ecoli	Full-Dataset	82.822 ±	5.467	88.873 ±	3.567
Ecoli	Random-sampling	80.059 ±	6.268	86.924 ±	4.218
Ecoli	Stratified-sampling	79.586 ±	6.227	86.721 ±	4.113
Ecoli	Balanced-sampling	79.405 ±	6.360	86.981 ±	4.034
German	Windowing	71.660 ±	4.608	63.119 ±	5.518
German	Full-Dataset	71.300 ±	3.765	62.605 ±	4.388
German	Random-sampling	71.800 ±	3.782	62.867 ±	4.408
German	Stratified-sampling	71.640 ±	3.799	62.857 ±	4.546
German	Balanced-sampling	67.820 ±	4.448	66.833 ±	4.014
Hypothyroid	Windowing	99.483 ±	0.346	98.880 ±	1.204
Hypothyroid	Full-Dataset	99.528 ±	0.353	98.871 ±	1.259
Hypothyroid	Random-sampling	94.340 ±	2.524	70.634 ±	23.378
Hypothyroid	Stratified-sampling	96.877 ±	1.652	94.594 ±	4.769

Continued on next page

Dataset	Method	Test Acc		Test AUC	
Hypothyroid	Balanced-sampling	96.236 ±	1.831	97.598 ±	1.421
Kr-vs-kp	Windowing	99.302 ±	0.583	99.294 ±	0.594
Kr-vs-kp	Full-Dataset	99.415 ±	0.433	99.412 ±	0.433
Kr-vs-kp	Random-sampling	94.171 ±	2.959	94.139 ±	3.061
Kr-vs-kp	Stratified-sampling	94.956 ±	1.766	94.956 ±	1.802
Kr-vs-kp	Balanced-sampling	94.984 ±	1.727	94.996 ±	1.756
Letter	Windowing	87.161 ±	2.074	93.324 ±	1.078
Letter	Full-Dataset	87.943 ±	0.720	93.731 ±	0.375
Letter	Random-sampling	82.216 ±	1.006	90.753 ±	0.523
Letter	Stratified-sampling	82.376 ±	1.148	90.836 ±	0.597
Letter	Balanced-sampling	82.430 ±	1.160	90.864 ±	0.603
Mushroom	Windowing	100.000 ±	0.000	100.000 ±	0.000
Mushroom	Full-Dataset	100.000 ±	0.000	100.000 ±	0.000
Mushroom	Random-sampling	73.746 ±	23.610	73.625 ±	23.684
Mushroom	Stratified-sampling	98.367 ±	0.813	98.312 ±	0.831
Mushroom	Balanced-sampling	98.424 ±	0.819	98.376 ±	0.831
Segment	Windowing	96.329 ±	1.655	97.859 ±	0.965
Segment	Full-Dataset	96.710 ±	1.335	98.081 ±	0.779
Segment	Random-sampling	90.719 ±	3.181	94.586 ±	1.855
Segment	Stratified-sampling	91.515 ±	2.074	95.051 ±	1.210
Segment	Balanced-sampling	91.455 ±	1.984	95.015 ±	1.157
Sick	Windowing	98.688 ±	0.640	93.667 ±	3.370
Sick	Full-Dataset	98.741 ±	0.523	93.662 ±	3.323
Sick	Random-sampling	96.193 ±	1.887	75.662 ±	19.843
Sick	Stratified-sampling	97.301 ±	1.051	86.908 ±	6.166
Sick	Balanced-sampling	94.785 ±	1.855	94.812 ±	2.641
Splice	Windowing	94.132 ±	1.682	95.626 ±	1.344
Splice	Full-Dataset	94.216 ±	1.474	95.723 ±	1.125
Splice	Random-sampling	89.997 ±	2.226	92.370 ±	1.951
Splice	Stratified-sampling	90.339 ±	1.973	92.757 ±	1.572
Splice	Balanced-sampling	89.846 ±	2.199	92.902 ±	1.570
Waveform-5000	Windowing	83.802 ±	9.864	87.848 ±	7.402
Waveform-5000	Full-Dataset	75.202 ±	1.989	81.396 ±	1.493
Waveform-5000	Random-sampling	75.046 ±	2.159	81.279 ±	1.619
Waveform-5000	Stratified-sampling	75.252 ±	1.981	81.431 ±	1.487
Waveform-5000	Balanced-sampling	75.514 ±	2.143	81.628 ±	1.609

3.5. Statistical Tests

The figures in this section visualize the results of the post-hoc Nemenyi test for the metrics previously shown in Tables 5, 6 and 7. This compact, information-dense visualization, called as Critical Difference diagram, consists on a main axis where the average rank of each methods is plotted along with a line that represents the Critical Difference (CD). Methods separated by a distance shorter than the CD are statistically indistinguishable, i.e., the evidence is not sufficient to conclude whether they have a similar performance and are connected by a black line. In contrast, methods separated by a distance larger than the CD have a statistically significant difference in performance. The best performing methods are those with lower rank values shown on the left of the figure.

Figure 2 shows the results for the number of bits required to encode the induced models ($L(H)$) presented in Table 6. The groups of connected algorithms are not significantly different. In this case, the complexity of the models induced using windowing does not show significant differences with the complexity of the models induced using the Full-Dataset or balanced sampling.

Figure 3 shows the results in terms of data compression given the decision tree ($L(D|H)$). If the compressibility provided by the models is verified on a stratified sample of unseen data, windowing and Full-Dataset tend to compress significantly better compared to traditional sampling methods. However, windowing tends to generate more complex models probably because its heuristic behavior enables the seek for more difficult patterns in the data.

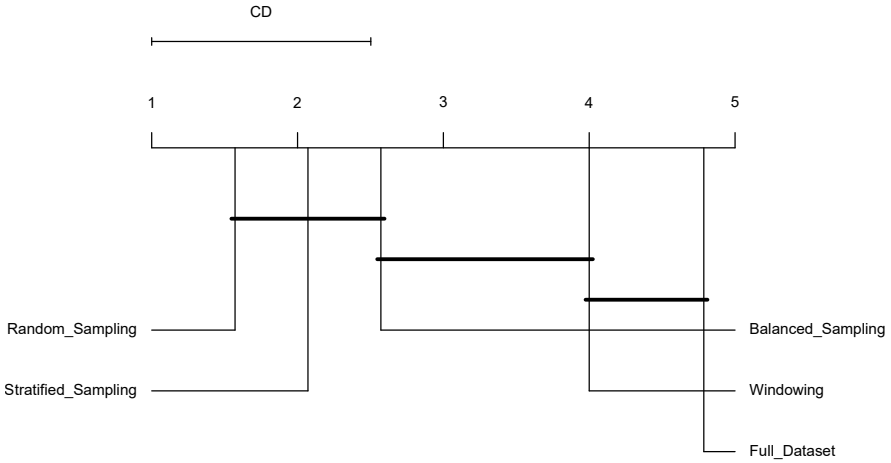

Figure 2. Demšar test regarding the required bits to encode trees, $L(H)$.

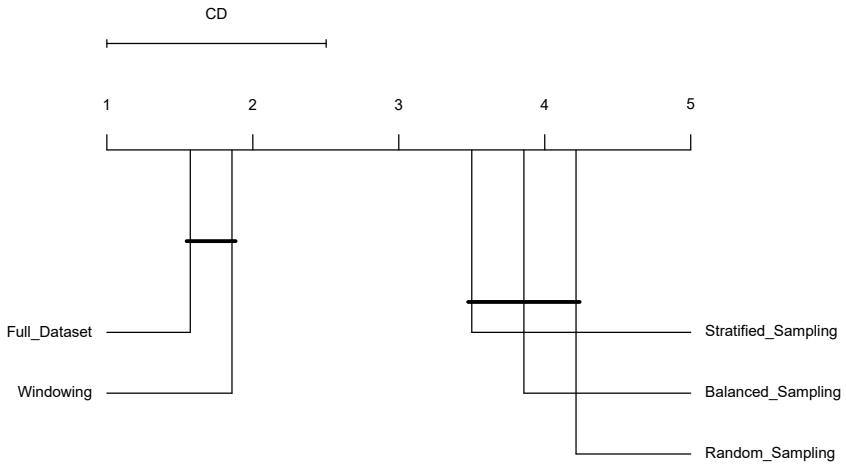

Figure 3. Demšar test regarding the required bits to encode the test data given the decision tree, $L(D|H)$.

Figure 4 shows the results in terms of MDL in the test set. Windowing and Full-Dataset do not show significant differences, nor they are statistically different to the traditional sampling methods. That is, that the induced decision trees generally need the same number of bits to be represented.

Figure 5 shows the results for accuracy. Windowing performs very well, being almost as accurate as Full-Dataset without significant differences. Both methods are strictly better than the random, balanced, and stratified samplings. When considering the AUC in Figure 6, results are very similar but the balanced sampling does not show significant differences with windowing and the Full-Dataset. Recall that both, windowing and balanced sampling, tend to balance the class distribution of the instances.

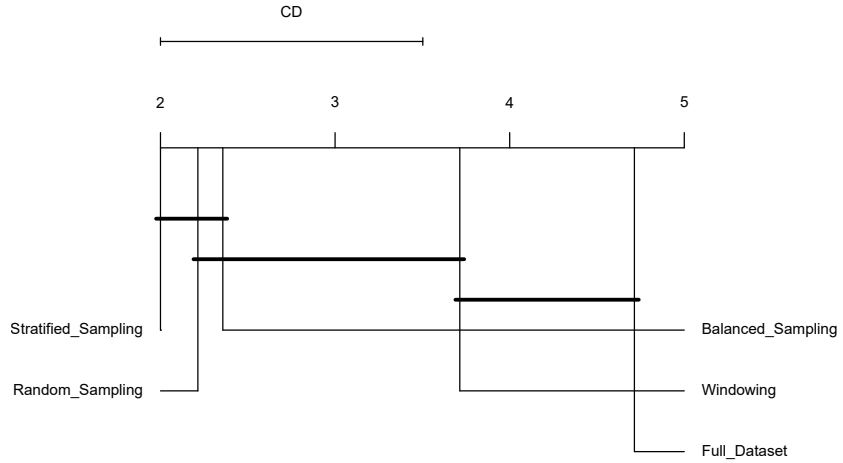

Figure 4. Demšar test regarding the MDL computed on the test dataset.

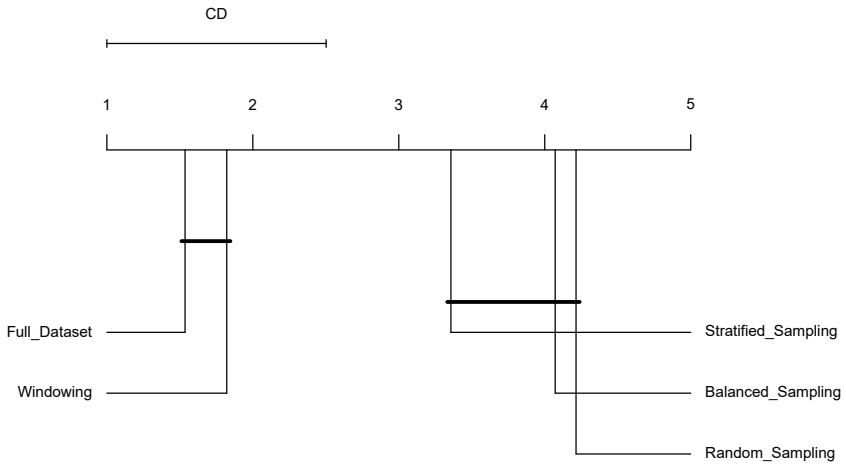

Figure 5. Demšar test regarding the accuracy over the test dataset.

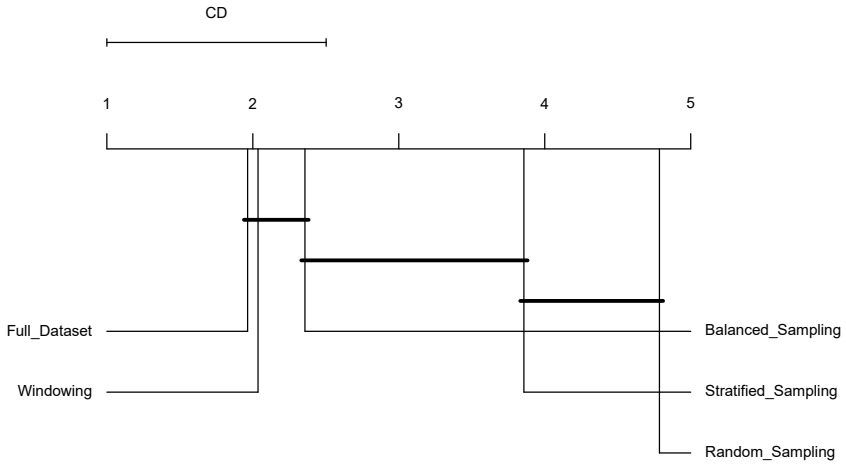

Figure 6. Demšar test regarding the AUC over the test dataset.

In terms of class distribution (Figure 7), windowing is known to be the method that tends to skew the distribution the most, given that the counter examples added to the window in each iteration of this algorithm belong most probably to the current minority class. As expected, the balanced and the

random sampling methods also skew the class distribution showing no significant differences with windowing. According to the percentage of attribute-value pairs given by Sim_1 (Figure 8), windowing and the traditional sampling methods cannot obtain the full set of attribute-value pairs included in the original dataset. Despite this, windowing is still very competent when it comes to prediction.

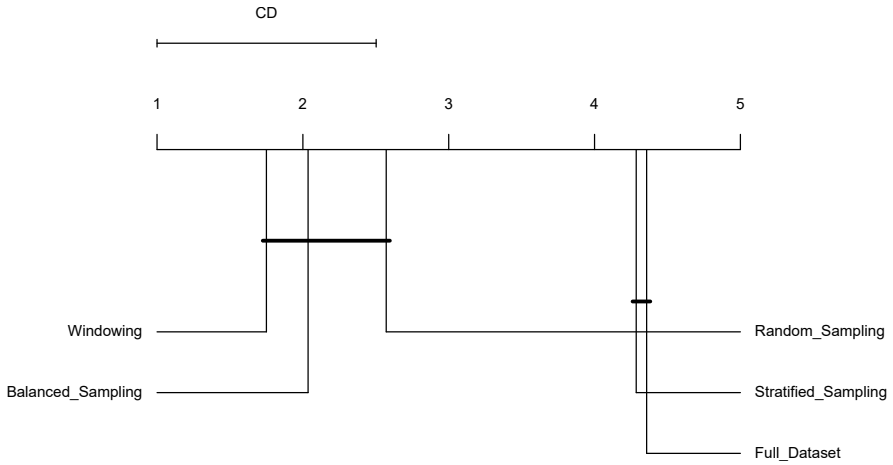

Figure 7. Demšar test regarding the Kullback–Leibler Divergence.

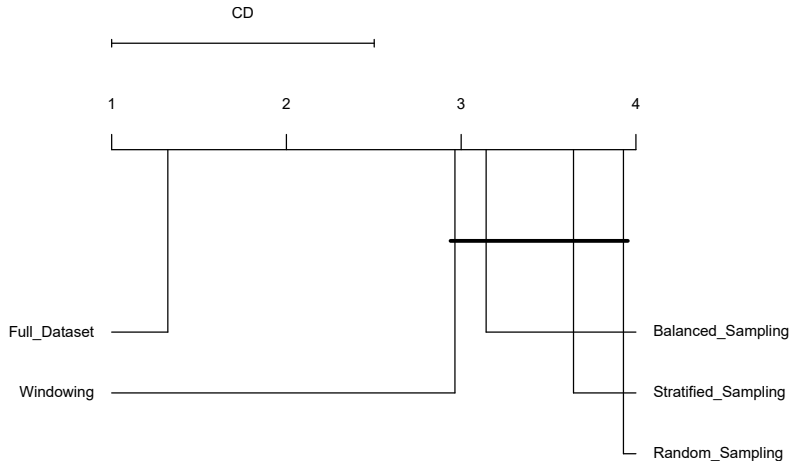

Figure 8. Demšar test regarding Sim1.

4. Conclusions

The generalization of the behavior of windowing beyond decision trees and the J48 algorithm has been corroborated. Independently of the inductive method used with windowing, high accuracies correlate with aggressive samplings up to 3% of the original datasets. This result motivates the study of the properties of the samples and models proposed in this work. Unfortunately, the Kullback–Leibler divergence and sim_1 do not seem to correlate with accuracy, although the first one is indicative of the balancing effect performed by windowing. MDL provided useful information in the sense that, although all methods generate models of similar complexity, it is important to identify which component of the MDL is more relevant in each case. For example, less complex decision trees, as those induced by random, balanced and stratified samplings, are more general but less accurate. In contrast, decision trees with better data compression, such as those induced using windowing and Full-Dataset, tend to be larger but more accurate. The key factor that makes the difference is the significant reduction of instances for induction. Recall that determining the size of the samples is

done automatically in windowing, based on the auto-stop condition of this method. When using traditional sampling methods the size must be figured out by the user of the technique. To the best of our knowledge, this is the first comparative study of windowing in this respect. This work suggests future lines of research on windowing, including:

1. Adopting metrics for detecting relevant, noisy, and redundant instances to enhance the quality and size of the obtained samples, in order to improve the performance of the obtained models. Maillo *et al.* [30] review multiple metrics to describe redundancy, complexity, and density of a problem and also propose two data big metrics. These kind of metrics may be helpful to select instances that provides quality information.
2. Studying the evolution of windows over time can offer more insights about the behavior of windowing. The main difficulty here is adapting some of the used metrics, e.g., MDL, to be used with models that are not decision trees.
3. Dealing with datasets of higher dimensions. Melgoza-Gutiérrez *et al.* [31] propose an agent & artifacts-based method to distribute vertical partitions of datasets and deal with the growing time complexity when datasets have a high number of attributes. It is expected that the achieved understanding on windowing contributes to combine these approaches.
4. Applying windowing to real problems. Limón *et al.* [10] applies windowing to the segmentation of colposcopic images presenting possible precancerous cervical lesions. Windowing is exploited here to distribute the computational cost of processing a dataset of 1.4×10^6 instances and 30 attributes. The exploitation of windowing to cope with learning problems of distributed nature is to be explored.

Author Contributions: Conceptualization, D.M.-G. and A.G.-H.; methodology, D.M.-G., A.G.-H. and N.C.-R.; software, A.G.-H., X.L. and D.M.-G.; validation, A.G.-H., N.C.-R., X.L. and F.G.; formal analysis, D.M.-G. and A.G.-H.; investigation, A.G.-H. and D.M.-G.; resources, X.L.; writing—original draft preparation, D.M.-G.; writing—review and editing, A.G.-H., N.C.-R., X.L. and F.G.; visualization, D.M.-G.; project administration, A.G.-H. All authors have read and agree to the published version of the manuscript.

Funding: The first author was funded by a scholarship from Consejo Nacional de Ciencia y Tecnología (CONACYT), Mexico, CVU:895160. The last author was supported by project RTI2018-095820-B-I00 (MCIU/AEI/FEDER, UE).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Results of Accuracy without Using Windowing

Table A1. Average accuracy without using windowing under a 10-fold cross validation (na = not available).

	j48	NB	jRip	MP	SMO
Adult	85.98 ± 0.28	83.24 ± 0.19	84.65 ± 0.16	na	na
Australian	87.10 ± 0.65	85.45 ± 1.57	84.44 ± 1.78	83.10 ± 1.28	86.71 ± 1.43
Breast	96.16 ± 0.38	97.84 ± 0.51	95.03 ± 0.89	96.84 ± 0.77	96.67 ± 0.40
Credit-g	73.59 ± 2.11	75.59 ± 1.04	73.45 ± 1.96	73.10 ± 0.72	76.66 ± 2.87
Diabetes	72.95 ± 0.77	75.83 ± 1.17	78.27 ± 1.81	74.51 ± 1.46	78.02 ± 1.79
Ecoli	84.44 ± 1.32	83.5 ± 1.64	82.25 ± 3.11	83.69 ± 1.44	83.93 ± 1.31
German	73.89 ± 1.59	76.94 ± 2.29	70.06 ± 0.90	70.26 ± 0.96	74.55 ± 1.76
Hypothyroid	99.48 ± 0.20	95.72 ± 0.68	99.60 ± 0.15	94.38 ± 0.25	94.01 ± 0.48
Kr-vs-kp	99.31 ± 0.06	87.68 ± 0.43	99.37 ± 0.29	99.06 ± 0.13	96.67 ± 0.37
Letter	87.81 ± 0.10	64.33 ± 0.28	86.34 ± 0.22	na	na
Mushroom	100.0 ± 0.00	95.9 ± 0.32	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
Poker-lsn	99.79 ± 0.00	59.33 ± 0.03	na	na	na
Segment	96.02 ± 0.29	79.95 ± 0.69	95.25 ± 0.52	95.61 ± 0.91	92.97 ± 0.36
Sick	98.88 ± 0.29	93.13 ± 0.43	98.19 ± 0.22	95.81 ± 0.45	93.70 ± 0.56
Splice	93.81 ± 0.39	95.05 ± 0.36	94.19 ± 0.27	na	93.46 ± 0.48
Waveform5000	75.58 ± 0.37	80.25 ± 0.33	79.54 ± 0.37	na	86.81 ± 0.21

References

1. Quinlan, J.R. Induction over large data bases. Technical Report STAN-CS-79-739, Computer Science Department, School of Humanities and Sciences, Stanford University, Stanford, CA, USA, 1979.
2. Quinlan, J.R. Learning efficient classification procedures and their application to chess en games. In *Machine Learning*; Michalski, R.S., Carbonell, J.G., Mitchell, T.M., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1983; Volume I, Chapter 15, pp. 463–482. doi:10.1016/B978-0-08-051054-5.50019-4.
3. Quinlan, J.R. Induction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–106.
4. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: San Mateo, CA, USA, 1993; Volume 1.
5. Quinlan, J.R. Improved Use of Continuous Attributes in C4.5. *J. Artif. Intell. Res.* **1996**, *4*, 77–90.
6. Wirth, J.; Catlett, J. Experiments on the Costs and Benefits of Windowing in ID3. In Proceedings of the Fifth International Conference on Machine Learning, Ann Arbor, MI, USA, 12–14 June 1988; Laird, J.E., Ed.; Morgan Kaufmann: San Mateo, CA, USA, 1988; pp. 87–99.
7. Fürnkranz, J. Integrative windowing. *J. Artif. Intell. Res.* **1998**, *8*, 129–164.
8. Quinlan, J.R. Learning Logical Definitions from Relations. *Mach. Learn.* **1990**, *5*, 239–266.
9. Limón, X.; Guerra-Hernández, A.; Cruz-Ramírez, N.; Grimaldo, F. Modeling and implementing distributed data mining strategies in JaCa-DDM. *Knowl. Inf. Syst.* **2019**, *60*, 99–143. doi:10.1007/s10115-018-1222-x.
10. Limón, X.; Guerra-Hernández, A.; Cruz-Ramírez, N.; Acosta-Mesa, H.G.; Grimaldo, F. A Windowing Strategy for Distributed Data Mining Optimized through GPUs. *Pattern Recognit. Lett.* **2017**, *93*, 23–30. doi:10.1016/j.patrec.2016.11.006.
11. Witten, I.H.; Frank, E.; Hall, M.A. *Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann Publishers: Burlington, MA, USA, 2011.
12. Dua, D.; Graff, C. UCI Machine Learning Repository, 2017. Available online: https://archive.ics.uci.edu/ml/index.php (accessed on 29 June 2020).
13. Bifet, A.; Holmes, G.; Kirkby, R.; Pfahringer, B. MOA: Massive Online Analysis. *J. Mach. Learn. Res.* **2010**, *11*, 1601–1604.
14. John, G.H.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–20 August 1995; Morgan Kaufmann: San Mateo, CA, USA, 1995; pp. 338–345.
15. Cohen, W.W. Fast Effective Rule Induction. In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; Morgan Kaufmann: San Francisco, CA, USA, 1995; pp. 115–123.

16. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J., Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*; MIT Press: Cambridge, MA, USA, 1986; pp. 318–362.
17. Platt, J. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In *Advances in Kernel Methods: Support Vector Learning*; Schoelkopf, B., Burges, C., Smola, A., Eds.; MIT Press: Cambridge, MA, USA, 1998.
18. Sokolova, M.; Lapalme, G. A Systematic Analysis of Performance Measures for Classification Tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. doi:10.1016/j.ipm.2009.03.002.
19. Provost, F.; Domingos, P. Well-Trained PETs: Improving Probability Estimation Trees (2000). Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.309 (accessed on 29 June 2020).
20. Rissanen, J. Stochastic Complexity and Modeling. *Ann. Stat.* **1986**, *14*, 1080–1100. doi:10.1214/aos/1176350051.
21. Quinlan, J.R.; Rivest, R.L. Inferring decision trees using the minimum description length principle. *Inf. Comput.* **1989**, *80*, 227–248.
22. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86.
23. Zhang, S.; Zhang, C.; Wu, X. *Knowledge Discovery in Multiple Databases*; Springer-Verlag London, Limited: London, UK, 2004.
24. Ros, F.; Guillaume, S. *Sampling Techniques for Supervised or Unsupervised Tasks*; Springer: Cham, Switzerland, 2019. doi:10.1007/978-3-030-29349-9.
25. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
26. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701.
27. Friedman, M. A Comparison of Alternative Tests of Significance for the Problem of *m* Rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. doi:10.1214/aoms/1177731944.
28. Zar, J.H. *Biostatistical Analysis (5th Edition)*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2007.
29. Iman, R.L.; Davenport, J.M. Approximations of the critical region of the fbietkan statistic. *Commun. Stat. Theory Methods* **1980**, *9*, 571–595. doi:10.1080/03610928008827904.
30. Maillo, J.; Triguero, I.; Herrera, F. Redundancy and Complexity Metrics for Big Data Classification: Towards Smart Data. *IEEE Access* **2020**, *8*, 87918–87928.
31. Melgoza-Gutiérrez, J.; Guerra-Hernández, A.; Cruz-Ramírez, N. Collaborative Data Mining on a BDI Multi-Agent System over Vertically Partitioned Data. In Proceedings of the 13th Mexican International Conference on Artificial Intelligence, Tuxtla Gutiérrez, Mexico, 16–22 November 2014; Gelbukh, A., Castro-Espinoza, F., Galicia-Haro, S.N., Eds.; IEEE Computer Society: Los Alamitos, CA, USA, 2014; pp. 215–220.