Towards 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] Universidad Veracruzana, Centro de Investigación en Inteligencia Artificial, Sebastián Camacho No 5, Xalapa, Ver., México 91000.
davidgalicia@outlook.es, {aguerra,ncruz}@uv.mx
[2] Universidad Veracruzana, Facultad de Estadística e Informática, Av. Xalapa s/n, Xalapa, Ver., Mxico 91000
hlimon@uv.mx
[3] Universitat de València, Departament d'Informàtica, Avinguda de la Universitat, s/n, Burjassot-València, España 46100
francisco.grimaldo@uv.es

Abstract. Windowing is a sub-sampling method that enables the induction of decision trees with large datasets. Using a small sample of the available training examples, the method can achieve levels of accuracy comparable or better than those obtained using the full available dataset. More relevant is the fact that Windowing-based strategies for Distributed Data Mining (DDM) have shown a correlation between the accuracy of the learned decision tree and the number of examples used to learn it, i.e., the higher the accuracy, the fewer examples used to induce the model. This paper corroborates that this behavior is also observed when adopting inductive algorithms of a different nature than C4.5 or ID3, the algorithms usually adopted when windowing, contributing to the use of Windowing as a general sub-sampling method for DDM. The paper also contributes exploring some metrics to the validation of the obtained sub-samples of examples.

Keywords: Sub-sampling · Windowing · Distributed Data Mining

1 Introduction

Windowing is a sub-sampling method that enabled the decision tree inductive algorithms ID3 [9–11] and C4.5 [12, 13] 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 the window and 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.

```
function WINDOWING(Examples)
    Window ← sample(Examples)
    Examples ← Examples − Window
    repeat
        stopCond ← true
        model ← induce(Window)
        for example ∈ Examples do
            if classify(model, example) ≠ class(example) then
                Window ← Window ∪ {example}
                Examples ← Examples − {example}
                stopCond ← false
    until stopCond
    return model
```

It has been argued [3] that the method offers three advantages: It copes well with memory limitations, reducing considerably the number of examples required to induce a model of acceptable accuracy. 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. 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.

Although the lack of memory doest not use to be an issue nowadays, similar concerns arise when mining big and/or distributed data. Windowing has been used as the core of a set of strategies for Distributed Data Mining (DDM) [6], obtaining consistent results with respect to the achievable accuracy and the number of examples required by the method. On the contrary, efficiency suffered for large datasets as the cost of testing the models in the remaining examples is not negligible. However, this is alleviated by using GPUs [5]. More relevant for this paper is the fact that the Windowing-based strategies shows 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. Reductions are as big as the 90% of the available training data.

The objective of this work is to corroborate if such a correlation is observed when using inductive algorithms of different nature, so that the advantages of windowing as a sub-sampling method could be generalized beyond decision trees. For this, the paper is organized as follows: Section 2 introduces the adopted methodology; Section 3 presents the obtained results; and Section 4 discusses conclusions and future work. A preliminary contribution of the paper is the study of some metrics to try to validate the obtained windows and to understand the way such sub-sampling works so efficiently in some cases.

2 Methodology

Because of our interest in distributed settings, JaCa-DDM [4] was adopted to run experiments. This tool [6] defines a set of Windowing-based strategies using J48, the Weka [14] implementation of C4.5, as inductive algorithm. Among them, Counter is the most similar to the original formulation of Windowing, excepting that: i) the dataset can be distributed in different sites, and ii) an auto-adjustable stop criteria with a established maximum number of iterations (10) is adopted. The parameters of the strategy, e.g., the maximum number of round, is adopted from the literature. The same configuration is used for all the experiments. The Counter strategy is tested on the datasets shown in Table 1, selected from the UCI [2] and MOA [1] repositories. 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.

Dataset	**Instances**	**Attribs**	**Types**	**Missing**	**Class**
Adult	48842	15	Mixed	Yes	2
Australian	690	15	Mixed	No	2
Breast	683	10	Numeric	No	2
Credit-g	1000	21	Mixed	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

Table 1. Datasets, adopted from UCI and MOA.

Apart from J48, the Counter strategy will be tested using the Weka implementations of Naive Bayes, jRip, Multi-Perceptron, and SMO as inductive algorithms. A 10-fold stratified cross-validation is run on each dataset, observing the average accuracy of the obtained models and the average percentage of original dataset used to induce the model, i.e., 100% means the full original dataset was used. All experiments 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.

[4] https://github.com/xl666/jaca-ddm

In order to understand the performed sub-sampling, the following measures were used to compare the obtained window and the original dataset:

- The Kullback-Leibler divergence (D_{KL}) [4] is defined as:

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

  where $P(x)$ is the full dataset class distribution and $Q(x)$ the window class 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 [15] is a similarity measure between datasets defined as:

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

  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.
- *Red* [7] measures redundancy in a dataset in terms of conditional population entropy (CPE), defined as:

$$CPE = -\sum_{i=1}^{n_c} p(c_i) \sum_{a=1}^{n_a} \sum_{v=1}^{n_{v_a}} p(x_{a,v}|c_i) log_2 p(x_{a,v}|c_i)$$

  where n_c is the number of classes, n_a is the number of attributes, and n_{v_a} is the number of values for the attribute a. c_i stands for the $i-th$ class and $x_{a,v}$ represents the $v-th$ value of attribute a. CPE can be normalized [3] in such a way that values closer to cero reflect lower redundancy:

$$Red = 1 - \frac{CPE}{\sum_{a=1}^{n_a} log_2 n_{v_a}}$$

3 Results

Figure 1 shows a strong negative correlation between the percentage of training instances used to induce the models and their accuracies, independently of the adopted inductive algorithm. This reproduces the results for J48 reported in literature [6] and corroborates that under Windowing, in general, the models with higher accuracy required less examples to be induced. However, accuracy is affected by the adopted inductive algorithm, e.g., Poker-lsn is approached very well by J48 (99.75 $\pm$ 0.07 of accuracy) requiring few examples (5% of the full dataset); while Naive Bayes is not quite successful in this case (60.02 $\pm$ 0.42 of accuracy) requiring more examples (59%). This behavior is also observed between jRip and MultiPerceptron for Hypothyroid; and between SMO and jRip for Waveform5000.

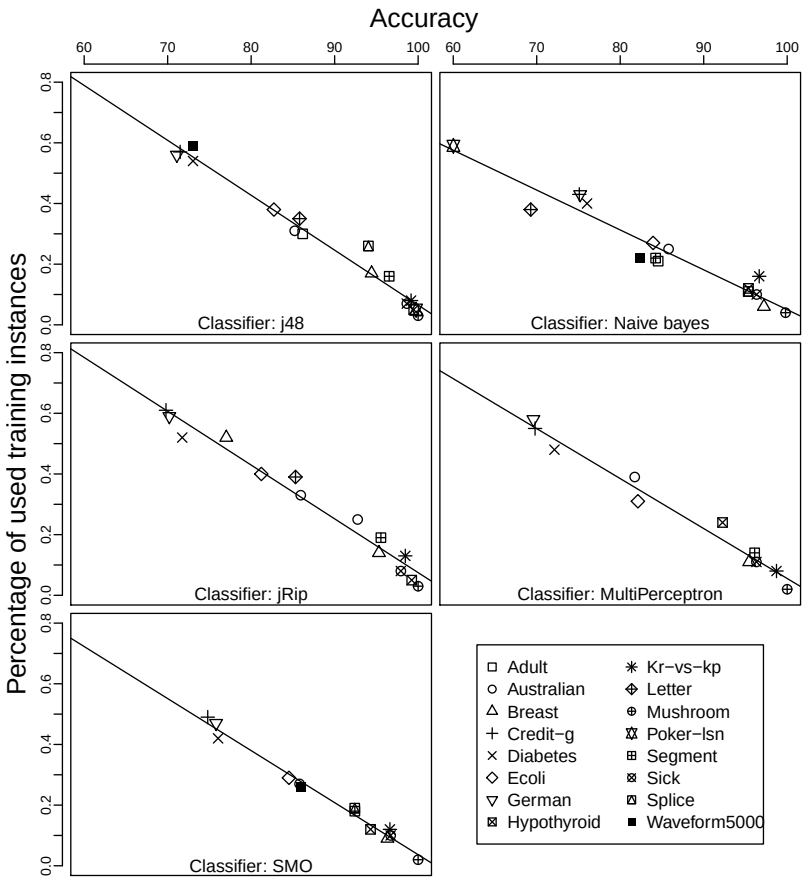

Fig. 1. Correlation between accuracy and percentage of used training examples. J48 = -0.98, NB = -0.96, jRip = -0.98, MP = -0.98 and SMO = -0.99.

Table 2 shows the accuracy results in detail while Table 3 show the number of used examples results, in terms of the percentage of the full dataset used for each inductive algorithm. Although not shown because of the available space, accuracies are comparable to those obtained without using Windowing, i.e., using the 100% of the available data to induce the models. Big datasets, as Adult, Letter, Poker-Isn, Splice, and Waveform5000 did not finish on reasonable time when using jRip, MultiPerceptron 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.

	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
Credit-g	71.50 ± 5.81	75.10 ± 2.60	69.80 ± 3.71	69.80 ± 5.63	74.80 ± 5.98
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 2. Accuracies obtained from 10-fold cross validation (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
Credit-g	0.57 ± 0.03	0.43 ± 0.01	0.61 ± 0.01	0.55 ± 0.04	0.49 ± 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

Table 3. Percentage of the full dataset used for induction (na = not available).

The Kullback-Leibler divergence coefficient between the windows and the full datasets was close to cero in all cases ($D_{KL} < 0.25$), evidencing that the class distribution of the windows is very similar to that observed in the full datasets. However it does not seem to be a correlation between this coefficient and the obtained accuracy, e.g., Mushroom has cero as divergence coefficient and 100%

of accuracy, but Waveform5000 has similar divergence but considerable lower accuracies.

Table 4 shows the results for sim_1, suggesting that the windows for Australian, Breast, German, Letter, Kr-vs-Kp, and Poker-lsn conserve all the values for their attributes observed in the full datasets; while Adult and Segment have problems achieving this. As in the previous case, this notion of similarity neither seems to correlate with the observed accuracies, e.g., Segment.

	j48	**NB**	**jRip**	**MP**	**SMO**
Adult	0.39±0.01	0.29±0.00	na	na	na
Australian	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Breast	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Credit-g	0.63±0.03	0.51±0.01	0.69 ± 0.01	0.63 ± 0.04	0.58 ± 0.01
Diabetes	0.73±0.04	0.63±0.02	0.72 ± 0.03	0.69 ± 0.02	0.64 ± 0.01
Ecoli	0.77±0.03	0.65±0.02	0.78 ± 0.02	0.69 ± 0.04	0.65 ± 0.03
German	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00
Hypothyroid	0.45±0.01	1.00±0.01	0.48 ± 0.01	0.68 ± 0.01	0.59 ± 0.01
Kr-vs-kp	1.00±0.01	0.97±0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
Letter	0.99±0.01	0.99±0.01	0.98 ± 0.00	na	na
Mushroom	0.97±0.02	0.99±0.01	0.98 ± 0.00	0.97 ± 0.01	0.97 ± 0.01
Poker-lsn	1.00±0.00	1.00±0.00	na	na	na
Segment	0.28±0.01	0.32±0.01	0.31 ± 0.01	0.25 ± 0.01	0.28 ± 0.00
Sick	0.57±0.02	0.58±0.01	0.59 ± 0.01	0.60 ± 0.02	0.60 ± 0.01
Splice	0.97±0.04	0.96±0.05	0.97 ± 0.03	na	0.96 ± 0.04
Waveform5000	0.93±0.01	0.71±0.01	0.90 ± 0.00	na	0.76 ± 0.01

Table 4. Table of similarity measure sim_1 using the 10-folds cross-validation windows.

Red shows consistently the same values for the windows and the full datasets, meaning that both of them have very similar leves of redundancy. Given the nature of Windowing this can be a little bit surprising, since the window is expected to be less redundant than the full dataset because it does not include examples already covered by the induced models. But *Red* measures the information value given the information about the class values, an intrinsic property of the data set; while the redundancy reduction expected by Windowing is a property of a dataset given a classifier. This behavior of *Red*, reported in literature [3], suggests that a different measure for redundancy in the sense of Windowing should be adopted.

4 Conclusions and future work

The correlation between the accuracy of the models obtained by Windowing and the number of examples used for this task was corroborated, independently of the adoptd inductive algorithm, i.e., high accurate models require fewer examples

to be learned. The metrics suggest that the windows have a class distribution very similar to the full datasets, as well as the same items (attribute-value pairs). They also have very similar intrinsic redundancy. Unfortunately, such similarites are not enough to explain the success of the technque since they do not correlate with the obtained accuracy of the models.

A metric reflecting the notion of redundancy in terms of the set of covered examples seems necessary to quantify the efficiency of Windowing a sub-sampling method. Also, the effect of noise in these metrics has to be considered explicitly, as well as its effect on accuracy. Observing the evolution of the windows through the whole process seems pertinent to the full understanding of Windowing.

References

1. Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: massive online analysis. *J. Mach. Learn. Res.*, 11:1601–1604, 2010.
2. Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
3. Johannes Fürnkranz. Integrative windowing. *Journal of Artificial Intelligence Research*, 8:129–164, 1998.
4. Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
5. Xavier Limón, Alejandro Guerra-Hernández, Nicandro Cruz-Ramírez, Héctor Gabriel Acosta-Mesa, and Francisco Grimaldo. A windowing strategy for distributed data mining optimized through GPUs. *Pattern Recognition Letters*, 93(Suplement C):23–30, July 2017.
6. Xavier Limón, Alejandro Guerra-Hernández, Nicandro Cruz-Ramírez, and Francisco Grimaldo. Modeling and implementing distributed data mining strategies in JaCa-DDM. *Knowledge and Information Systems*, 60(1):99–143, 2019.
7. Martin Møller. Supervised learning on large redundant training sets. *International Journal of Neural Systems*, 4(1):15–25, 1993.
8. J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.
9. John Ross Quinlan. Induction over large data bases. Technical Report STAN-CS-79-739, Computer Science Department, School of Humanities and Sciences, Stanford University, Stanford, CA, USA, May 1979.
10. John Ross Quinlan. Learning efficient classification procedures and their application to chess en games. In Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, editors, *Machine Learning*, volume I, chapter 15, pages 463 – 482. Morgan Kaufmann, San Francisco (CA), 1983.
11. John Ross Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
12. John Ross Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, San Mateo, CA., USA, 1993.
13. John Ross Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.
14. Ian H Witten, Eibe Frank, and Mark A Hall. *Data Mining: Practical Machine Learning Toools and Techniques*. Morgan Kaufmann Publishers, Burlington, MA., USA, 2011.
15. Shichao Zhang, Chengqi Zhang, and Xindong Wu. *Knowledge Discovery in Multiple Databases*. Advanced Information and Knowledge Processing. Springer-Verlag London, Limited, London, UK, 2004.