# Clustering Data Streams Using a Forgetful Neural Model

Douglas O. Cardoso, Felipe França
Universidade Federal do Rio de Janeiro
PESC-COPPE
Rio de Janeiro, RJ, Brazil
{douglascardoso, felipe}@cos.ufrj.br

João Gama
Universidade do Porto
LIAAD-INESC TEC
Oporto, Portugal
jgama@fep.up.pt

## ABSTRACT

To cluster a data stream is a more challenging task than its regular batch version, having stricter performance constraints. In this paper an approach to this problem is presented, based on WiSARD, a memory-based artificial neural network (ANN) model. This model functioning was reviewed and improved, in order to adapt it to this task. The experimental results obtained support the use of this system for the analysis of data streams in an informative way.

## CCS Concepts

**•Information systems → Clustering; Data stream mining; •Computing methodologies → Neural networks;**

## Keywords

Clustering; Data Streams; Neural Networks; WiSARD

## 1. INTRODUCTION

Weightless ANN models [1] are one of the first machine learning tools, which powered some of the most primitive pattern recognition systems [2]. Despite working straightforwardly, some of these models can effectively handle some of the most recent learning challenges [4, 5]. Moreover, this simplicity eases the creation of variations of these learners without compromising their operation. This was explored through this research, resulting in the successful development of a data stream clustering [6] system based on WiSARD [2], one of these models.

## 2. DATA STREAM CLUSTERING

Let a data stream $S = (s_1, s_2, s_3, \dots)$ be an unbounded sequence of inputs. Each input $s_i = (\vec{x}_i, t_i)$ is a pair consisting of a vector $\vec{x}_i \in \mathbb{R}^n$ representing an observation, and its respective arriving time $t_i \in \mathbb{R}$. As a processing constraint, the stream item $s_i$ is assumed to be available just until the item $s_{i+1}$ becomes known. During this research it was considered that $t_i = i$, but this is not required by this proposal [9]. In this work, the sliding window model [8] is used to define how observations become obsolete. This model considers just the $\omega$ most recent observations equally important to define the up-to-date knowledge, where $\omega$ is a model parameter. Moreover, under this model, the defined clusters should be somehow similar to those generated by a conventional clustering algorithm from these $\omega$ most recent observations.

A widespread technique to cluster data streams is to continuously process the raw input into an intermediate layer, which provides the information to high-level clustering, realized on demand. These two parts are called the data abstraction step and the clustering step [8]. The intermediate layer is usually a big collection of tiny groups, called micro-clusters. Algorithm 1 describes the general form of the data abstraction step. The clustering step, in turn, is a conventional clustering task which uses the micro-clusters as input.

```
for all $\vec{x}_i$, the observations from the stream do
    Discard information obtained from observation $\vec{x}_{i-\omega}$
    Discard micro-clusters which are no longer useful
    Find the micro-cluster $mc_j$ which better encloses $\vec{x}_i$
    if $mc_i$ is close enough to $\vec{x}_i$ then
        Update $mc_j$ definition using $\vec{x}_i$
    else
        Start a new micro-cluster based on $\vec{x}_i$
```

Algorithm 1: A generic data abstraction procedure.

## 3. NEURAL NETWORKS AND WISARD

Biological neurons receive stimuli through its dendrites, which are organized as a tree. These stimuli are combined during tree traversal until the neuron soma, where a response for such inputs is generated. This response is forwarded trough the axon to other neurons by synapses.

In the most popular mathematical abstraction of biological neurons [7], the synapses are represented by edges, connecting the nodes of the neural network. Various ANN models rely on the modification of weights of its edges by the superposition of the effects of the observations in the training sample. Weightless ANNs [1] are memory-based alternatives to weights-based ones. All links of these networks have no weight. Their memorizing nodes are responsible for learning.

The biological inspiration of these memory nodes comes from the fact that the input signals (excitatory or inhibitory) of biological neurons are combined by the dendritic tree before reaching the soma, prompting an output signal. This is

*SAC 2016,April 04-08, 2016, Pisa, Italy*
©2016 ACM. ISBN 978-1-4503-3739-7/16/04…$15.00
DOI: http://dx.doi.org/10.1145/2851613.2851889

similar to accessing a dictionary of bit values using a binary key. The most basic memory neurons operate this way.

The WiSARD is a weightless ANN model originally designed for classification. To realize a class prediction, it provides for each class a value in the interval $[0,1]$, representing how well the provided observation matches the acquired knowledge regarding that class. Each class score is computed from a structure called discriminator, which is responsible for storing the knowledge regarding a class. How a discriminator learns about its respective class is described in algorithm 2. Mind some notation introduced here: $\Delta_{\dot{y}}$ is the discriminator of class $\dot{y}$; $\Delta_{\dot{y},j}$ is the $j^{th}$ node of $\Delta_{\dot{y}}$; $\delta$ is the number of nodes which compose each discriminator. Consider as a randomly defined mapping $\text{addressing}(\vec{x}) = (a_1\ a_2\ \cdots\ a_\delta)$, $a_i \in \{0,1,\ldots,2^\beta - 1\}$. Moreover, $\delta$ and $\beta$ are model parameters. After training, a WiSARD instance can rate the matching between any known class $\dot{y}$ and an observation $\vec{x}$ as shown in eq. (1a). At last, classification happens according to eq. (1b).

```
for all $\Delta_{\dot{y},j}$, the network nodes do
    $\Delta_{\dot{y},j} \leftarrow \varnothing$        ▷ Initially, nodes are empty sets
for all pairs $(\vec{x}_i, y_i)$, the training sample do
    for all addresses $a_j$ in addressing$(\vec{x}_i)$ do
        $\Delta_{y_i,j} \leftarrow \Delta_{y_i,j} \cup \{a_j\}$
```

Algorithm 2: A description of WiSARD training procedure.

$$\text{matching}(\dot{y},\vec{x}) = \frac{1}{\delta}\sum_i \left[\text{addressing}_i(\vec{x}) \in \Delta_{\dot{y},i}\right]^{1} \quad ; \qquad (1a)$$

$$\hat{y} = \underset{\dot{y}}{\text{argmax}}\ \text{matching}(\dot{y},\vec{x}) \quad . \qquad (1b)$$

# 4. WISARD FOR CLUSTERING STREAMS

Targeting data stream clustering, it is proposed here to use WiSARD discriminators as micro-cluster representatives. Since clustering data is unlabeled, a discriminator will not "absorb" observations of some class, but those it matches well enough, better than other discriminators. This way, some parts of algorithm 1 would be translated as follows: line 4 is a search for the best matching discriminator in the same mold of eq. (1b); line 5 is similar to compare a matching rate to a threshold; line 6 is a regular absorption of $\vec{x}_i$ by the discriminator of $mc_j$; line 8 prompts the addition of a new discriminator to WiSARD, which absorbs $\vec{x}_i$.

This idea is supported by some interesting facts: a discriminator is natively an incremental learner; there is no restriction to adding or removing discriminators, since they exist independently; WiSARD provides a richer feedback than just a distance to a decision boundary as discriminative classifiers, what enables decisions beyond class prediction [4]; a discriminator is more informative than data abstraction units of other approaches to data stream clustering, which depict a data sample based on its mean and variance.

## 4.1 Knowledge Forgetting

The panorama provided so far is very positive with respect to this WiSARD adaptation. However, lines 2 and 3 of algorithm 1 remain unclear: the first regards cancelling the influence of an observation on aggregated knowledge so to keep it up-to-date; the second one concerns the proper ending of the life cycle of micro-clusters when they become useless.

The cancellation of the influence of an outdated observation on knowledge comes down to deleting the addresses obtained from it which were stored in the nodes of a discriminator. This can be accomplished keeping a reference to every stored address in a Least Recently Used cache, so that every time an address is stored, its reference is updated as the most recently used. Using this structure increases the cost of absorption, but it expressively simplifies dumping expired addresses: it is enough to delete the least recent cache entry until it becomes reference to a non-expired element.

Now, consider that the knowledge a discriminator possess is expanded when new addresses are added to the discriminator nodes, and it is contracted by the removal of expired addresses. Suppose that some time after creation, a discriminator $\Delta_k$ becomes "empty" (i.e., $\forall j, \Delta_{k,j} = \varnothing$), because all its addresses expired. From then on this discriminator will be unable absorb other observations, since it can not match any of their addresses (i.e., $\forall \vec{x}_i, \text{matching}(k,\vec{x}_i) = 0$). Thus, it can be discarded as it is no longer useful.

## 4.2 Clusters Imbalance

Because of the way WiSARD learns, it is strongly susceptible to have its effectiveness harmed when dealing with class imbalance. Although there is no information about classes for clustering, imbalance is still possible: in some instant during stream processing, two discriminators can have a very different number of observations associated to themselves [5]. To consider such difference during clustering, an alternative definition of matching comes in handy. Thus, it is proposed here to define the cardinality of a discriminator and a normalized matching rate as in eqs. (2a) and (2b), respectively

$$|\Delta_k| = \left(\prod_j |\Delta_{k,j}|\right) \quad ; \qquad (2a)$$

$$\text{matching}^{+}(k,\vec{x}) = \frac{\text{matching}(k,\vec{x})}{(|\Delta_k|)^{1/\delta}} \quad . \qquad (2b)$$

## 4.3 Algorithmic Description

To conclude the description of the proposed WiSARD-based system to cluster data streams, its functioning is detailed in algorithm 3, in the same format of algorithm 1.

# 5. EXPERIMENTAL EVALUATION

The effectiveness of the developed WiSARD-based system was assessed through a collection of experiments, comparing the quality of the $k$ micro-clusters which are maintained at a given time instant $t$ with those a conventional $k$-means algorithm generates using the current observations as input.

Two real-world data sets were used to evaluate the clustering performance. The Forest Cover Type (FCT) data set [3] is a popularly used data stream clustering benchmark. For the characterization of the data set as a stream, the observations were sorted with respect to their 'elevation' [9]. The original 10% data split of the Network Intrusion Detection (NID) data set [2] was also employed, with the observations in the original order. Just the numeric attributes of both data sets were considered.

[1] Iverson bracket: $[L] = 1$ if the logical expression $L$ is true; otherwise, $[L] = 0$.

[2] http://kdd.ics.uci.edu/databases/kddcup99/

| Data Set | Window Size ($\omega$) | Entropy | # Micro-clusters | Homogeneity | | |
|---|---|---|---|---|---|---|
| | | | | WiSARD | K-means | Difference |
| FCT | 400 | 0.681 | 06.34$_{(0.478)}$ | 0.079$_{(0.005)}$ | 0.098$_{(0.005)}$ | -0.019$_{(0.037)}$ |
| FCT | 2000 | 0.624 | 27.65$_{(0.858)}$ | 0.140$_{(0.004)}$ | 0.153$_{(0.003)}$ | -0.013$_{(0.024)}$ |
| FCT | 10000 | 0.605 | 59.74$_{(0.753)}$ | 0.181$_{(0.002)}$ | 0.205$_{(0.002)}$ | -0.023$_{(0.024)}$ |
| NID | 400 | 0.086 | 01.23$_{(0.077)}$ | 0.967$_{(0.025)}$ | 0.954$_{(0.035)}$ | +0.013$_{(0.160)}$ |
| NID | 2000 | 0.070 | 01.99$_{(0.110)}$ | 0.955$_{(0.024)}$ | 0.897$_{(0.012)}$ | +0.059$_{(0.220)}$ |
| NID | 10000 | 0.139 | 06.37$_{(0.120)}$ | 0.954$_{(0.021)}$ | 0.887$_{(0.016)}$ | +0.068$_{(0.186)}$ |

Table 1: Test definitions and results. Top performances are highlighted. Standard deviation of averages shown as subscripts.

```
for all $(\vec{x}, t)$, the stream elements do
    while min LRU $\leq t - \omega$ do
        $k, j, a_j = \underset{k,j,a_j}{\text{argmin}}\ \text{LRU}_{k,j,a_j}$
        $\Delta_{k,j} \leftarrow \Delta_{k,j} \setminus \{a_j\}$
        Delete $\text{LRU}_{k,j,a_j}$
    Delete all $\Delta_k$ for which $|\Delta_k| = 0$
    $k = \underset{k}{\text{argmax}}\ \text{matching}^{+}(k, \vec{x})$
    if $\text{matching}^{+}(k, \vec{x}) > \alpha$ then        ▷ $\alpha$ is a parameter
        for all addresses $a_j$ in addressing($\vec{x}_i$) do
            $\Delta_{k,j} \leftarrow \Delta_{k,j} \cup \{a_j\}$
            $\text{LRU}_{k,j,a_j} \leftarrow t$
    else
        Let $\Delta_*$ be a new discriminator
        for all addresses $a_j$ in addressing($\vec{x}_i$) do
            $\Delta_{*,j} \leftarrow \Delta_{*,j} \cup \{a_j\}$
            $\text{LRU}_{*,j,a_j} \leftarrow t$
```

Algorithm 3: WiSARD-based data abstraction procedure.

Each test, represented by one row of table 1, was repeated 10 times. For all runs, at each 10 thousand observations, it was realized a quality assessment of the current clustering. One of the classes of NID is by far the most frequent, what leads to the low entropy level of NID compared to that of FCT. This fact reflects on the number of micro-clusters maintained during stream processing. With respect to homogeneity, the WiSARD-based system performs similarly to K-means, despite the fact that the first not only learns incrementally but also discards useless information, while the last works as a regular batch learner.

It can be noticed that K-means was the superior alternative for the FCT data set, while WiSARD was in this condition regarding NID. This may be explained by the fact that NID is more stable than FCT as a stream. Consequently, to learn based on some previous established structure appears to be more beneficial for NID, while learning from scratch goes better for FCT.

## 6. CONCLUSION

Although there is a rich variety of works on data streams clustering, there is still room for improvements. The existing alternatives to the problem in question discard expired data on an estimate basis. However, the novel conception of micro-clusters, based on discriminators, allowed to overcome this condition. An intended continuation of this work is the definition of a distance-like measure for WiSARD discriminators, targeting to perform the clustering step based on their pairwise similarities. Currently, the throughput performance of the proposed system is still being analyzed and improved, based on redesigning the structure of WiSARD, and this already provides promising preliminary results.

## 7. ACKNOWLEDGMENTS

Douglas O. Cardoso thanks CNPq and CAPES process 99999.005992/2014-01 financial support.

João Gama thanks European Commission through the project MAESTRA (Grant number ICT-2013-612944).

## 8. REFERENCES

[1] I. Aleksander, M. D. Gregorio, F. M. G. França, P. M. V. Lima, and H. Morton. A brief introduction to weightless neural systems. In *17th European Symposium on Artificial Neural Networks, Proceedings*, 2009.

[2] I. Aleksander, W. Thomas, and P. Bowden. WiSARD, a radical step forward in image recognition. *Sensor Review*, 4(3):120–124, 1984.

[3] J. A. Blackard and D. J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

[4] D. O. Cardoso, F. M. G. França, and J. Gama. A bounded neural network for open set recognition. In *2015 International Joint Conference on Neural Networks*, 2015.

[5] D. O. Cardoso, M. D. Gregorio, P. M. V. Lima, J. Gama, and F. M. G. França. A weightless neural network-based approach for stream data clustering. In *Intelligent Data Engineering and Automated Learning - 13th International Conference, Proceedings*, 2012.

[6] J. Gama. *Knowledge Discovery from Data Streams*. Chapman and Hall / CRC Data Mining and Knowledge Discovery Series. CRC Press, 2010.

[7] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[8] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. Carvalho, and J. Gama. Data stream clustering: A survey. *ACM Comput. Surv.*, 46(1):13:1–13:31, July 2013.

[9] I. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with drifting streaming data. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(1):27–39, Jan 2014.