Issues on Evaluation Stream Learning Algorithms

João Gama, Raquel Sebastião, Pedro Rodrigues
LIAAD-INESC Porto, University of Porto, Portugal

KDD, Paris 2009

Data Streams

Continuous flow of data generated at **high-speed** in **dynamic**, **time-changing** environments.
The usual approaches for *querying*, *clustering* and *prediction* use **batch procedures** cannot cope with this streaming setting.

Most of the Machine Learning algorithms assume:

- Instances are independent and generated at random according to some probability distribution $\mathcal{D}$.
- It is required that $\mathcal{D}$ is stationary

In Practice: *finite* training sets, *static* models.

Data Streams

We need to maintain **decision models** in **real time**.
Decision Models must be capable of:

- **incorporating** new information at the speed data arrives;
- **detecting** changes and **adapting** the decision models to the most recent information.
- **forgetting** outdated information;

Unbounded training sets, dynamic models.
Examples are not *iid*.
How to evaluate decision models that evolve over time?

Survey of Evaluation Methods

Work	Evaluation Method	Memory Management	Data Sources	Examples Train	Examples Test	Learning Curves	Drift
VFDT	holdout	Yes	Artif.	1M	50k	Yes	No
	holdout	Yes	real	4M	267k	Yes	No
CVFDT	holdout	Yes	Artif.	1M	Yes	Yes	Yes
VFDTc	holdout	No	Artif.	1M	250k	Yes	No
UFFT	holdout	No	Artif.	1.5M	250k	Yes	Yes
FACIL	holdout	Yes	Artif.	1M	100k	Yes	Yes
MOA	holdout	Yes	Artif.	1G		Yes	No
ANB	Prequential	No	Artif.			Yes	Yes

Evaluation Experiments Design

You cannot touch the same water twice.

Cross Validation and variants does not apply.

Two alternatives:

- Holdout if data is stationary;
- Predictive Sequential (prequential).

What if the distribution is non-stationary ?

- The *Predictive Sequential* approach.
  A. Dawid, *Statistical theory: the Prequential Approach*, 1984
  - For each example:
    - First: make a prediction
    - Second: compute the loss, whenever the target is available.

Prequential Metrics

- Accumulated sum of a loss function:
  $S = \sum_{i=1}^{n} L(y_i, \hat{y}_i)$ or
  $S_t = L(y_t, \hat{y}_t) + S_{t-1}$
- Mean loss: $M = 1/n \times S$

1. A learning curve for a sequence of points;
2. Pessimist estimator of accuracy;
3. Problematic to apply with algorithms with large testing time.

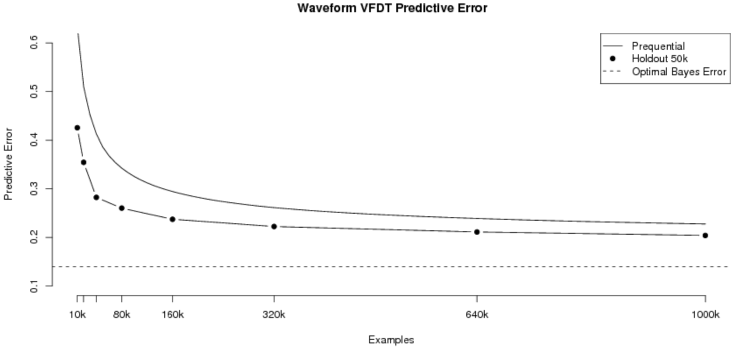
Prequential Evaluation

In this paper:

- Prequential Estimates:
  - using sliding windows: $S = \sum_{i=t}^{t+k} L(y_i, \hat{y}_i)$
  - fading factors: $S_t = L(y_t, \hat{y}_t) + \alpha \times S_{t-1}$
- Comparing two classifiers;
  - Monitoring relative performance (Q statistic) using fading factors;
  - McNemar test using fading factors;
- Dealing with change;
  - Page-Hinkley test using fading factors.

Prequential versus Holdout

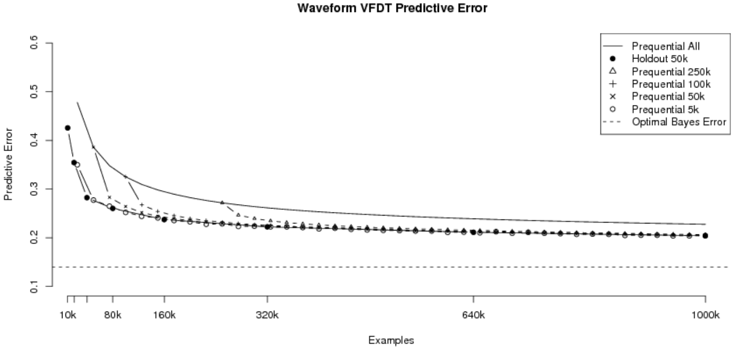
Prequential is a pessimistic estimator.



Prequential (sliding window) versus Holdout

$$S = \sum_{i=t}^{t+k} L(y_i, \hat{y}_i)$$

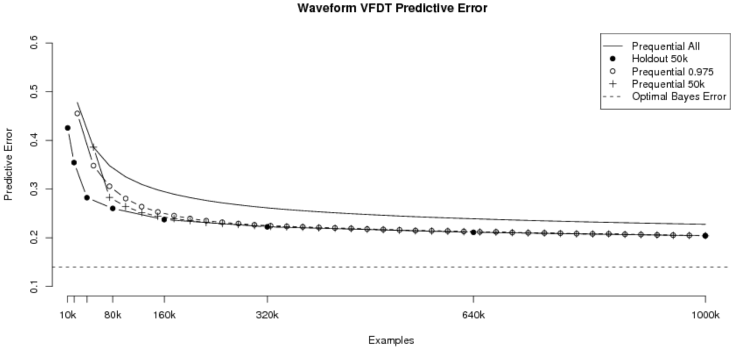
Prequential over a sliding window converges to the holdout estimator.



Prequential (fading factor) versus Holdout

$$S_t = L(y_t, \hat{y}_t) + \alpha \times S_{t-1}$$

Prequential using fading factors converges to the holdout estimator.



Accumulated Loss using Fading Factors

- The fading factor is multiplicative, corresponding to an exponential forgetting.
- At time-stamp t the weight of example $t - k$ is α^k.
- Fading factors are fast and memoryless.

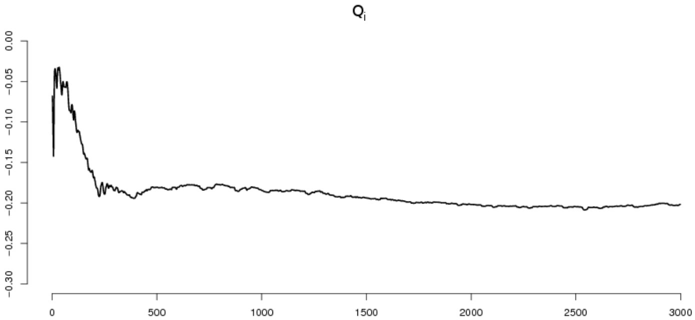
This is a strong advantage over sliding-windows that require to maintain in memory all the observations inside the window.

Comparing Two Classifiers

- Let S_i^A and S_i^B be the sequences of the prequential accumulated loss for each algorithm.
- A useful statistic that can be used with almost any loss function, is: $Q_i(A, B) = log(\frac{S_i^A}{S_i^B})$.
- The signal of Q_i is informative about the relative performance of both models, while its value shows the strength of the differences.

Accumulated Loss

Q_i reflects the overall tendency but exhibit long term influences and is not able to fast capture when a model is in a recovering phase.



Accumulated Loss over sliding windows

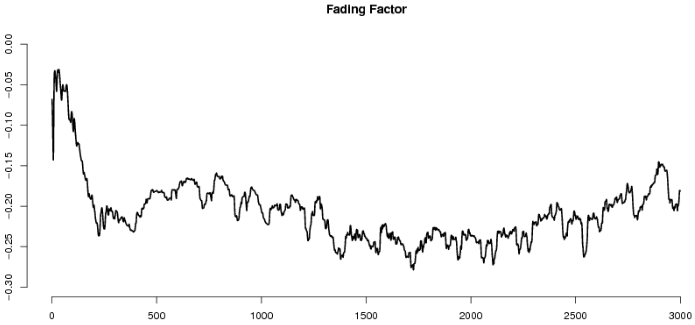
Q_i reflects the overall tendency but:

- exhibit long term influences and
- is not able to fast capture when a model is in a recovering phase.

Sliding windows is an alternative, with the known problems of deciding the window-size,

Accumulated Loss using Fading Factors

$$Q_i^{\alpha}(A,B) = log\left(\frac{L_i(A) + \alpha \times S_{i-1}^{A}}{L_i(B) + \alpha \times S_{i-1}^{B}}\right).$$



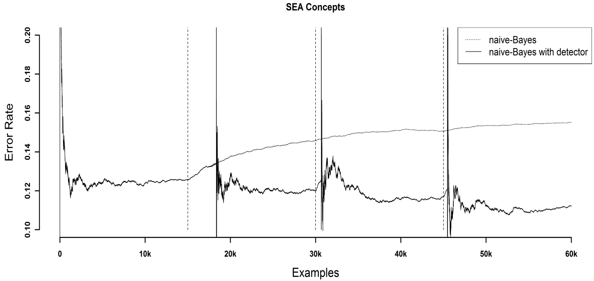
Signed McNemar Test for Comparative Assessment

- The McNemar test is one of most used tests for the 0-1 loss function;
- We need to compute two numbers:
  - $n_{0,1}$ denotes the number of examples misclassified by A and not by B;
  - $n_{1,0}$ denotes the number of examples misclassified by B and not by A;
- Both can be updated on the fly,
- Test statistic :

$$sign(n_{0,1} - n_{1,0}) \times \frac{(n_{0,1} - n_{1,0})^2}{n_{0,1} + n_{1,0}}$$

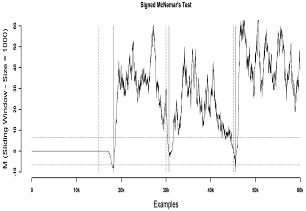
For a confidence level of 0.99, the null hypothesis is rejected if the statistic is greater than 6.635.

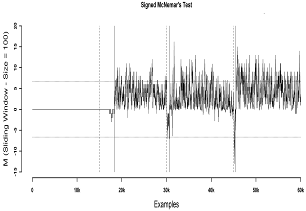
Signed McNemar Test



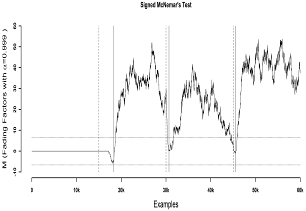
Signed McNemar Test

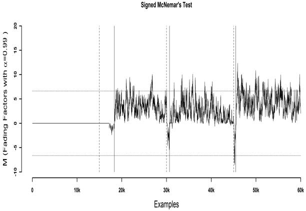
Sliding Windows: 1000, 100





Fading Factors: 99.9%, 99%





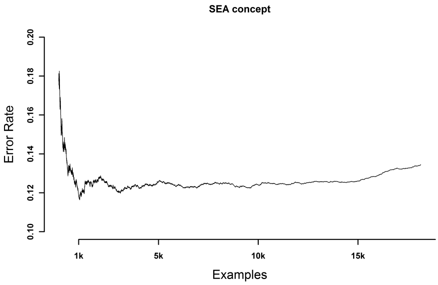
Concept drift and The Page-Hinckley Test

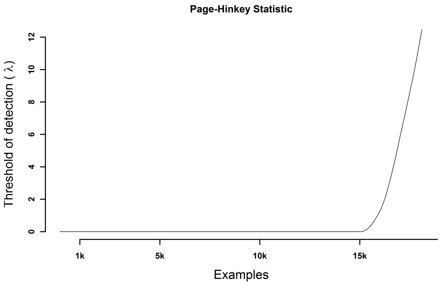
- The PH test is a sequential adaptation of the detection of an abrupt change in the average of a Gaussian signal.
- It considers a cumulative variable m_T, defined as the cumulated difference between the observed values and their mean till the current moment:

$$m_{t+1} = \sum_{1}^{t} (x_t - \bar{x}_t - \delta)$$

- The minimum value of this variable is also computed with the following formula: $M_T = min(m_t, t = 1 \dots T)$.
- The test monitors the difference between M_T and m_T: $PH_T = m_T - M_T$.
- When this difference is greater than a given threshold (λ) we alarm a change.

Illustrative Evaluation – Drift

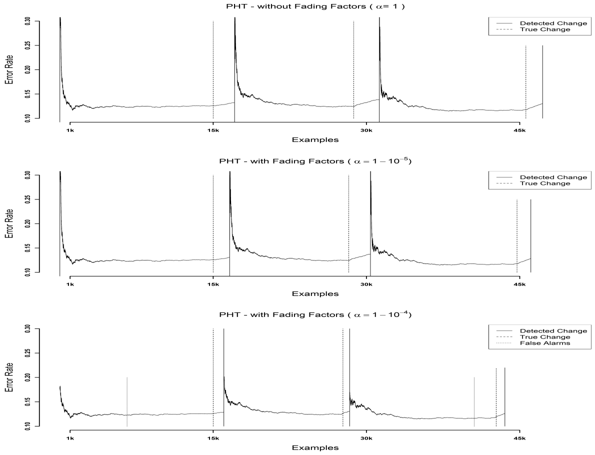




The left figure: the prequential error of a classifier with a change at point 15k.

The right figure: evolution of the Page-Hinckley test statistic.

Fading Factors and Delay Time



The evolution of the error rate and the delay times in drift detection using the Page-Hinckley test and different *fading-factors*.

Fading Factors and Delay Time

$$m_T = \alpha \times m_{T-1} + (x_t - \hat{x}_T - \delta)$$

	Fading Factors $(1-\alpha)$					
Drifts	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	0
1st drift	1045 (1)	1609	2039	2089	2094	2095
2nd drift	654 (0)	2129	2464	2507	2511	1640
3rd drift	856 (1)	1357	1609	1637	2511	1641

Table: Delay times in drift scenarios using different *fading factors*. We observe false alarms only for $1-\alpha = 10^{-4}$. The number of false alarms is indicated in parenthesis.

Lessons Learned I

- Prequential Error Estimates converges to holdout estimate:
  - Computed over a sliding window;
  - Computed using fading factors.
    - Fading factors are a faster and memory less approach, that do not require to store in memory all the errors in the window.
- Comparing Classifiers
  - The Q statistic using fading factors;
  - Hypothesis test using fading factors;
- Time-changing environments
  - The use of fading factors in drift detection achieve faster detection rates, maintaining the capacity of being resilient to false alarms when there are no drifts.

Lessons Learned II

Predictive sequential statistics to assess performance of algorithms in time-changing data streams.

Learning as a process and monitor the evolution of the learning process itself.

- Assess the performance of learning algorithms in dynamic environments;
- Compare algorithms and variants;
- Assess the evolution of performance in time-changing environments.