Streaming Machine Learning (SML)

Alessio Bernardo
09-11-2022

About me

Alessio Bernardo

alessiobernardo.github.io

Ph.D. Student in Data Science:

- Politecnico di Milano
- Research on Streaming Machine Learning

M.Sc. & B.Sc. Computer Engineering:

- Politecnico di Milano

Part I

Introduction

Credits

- Albert Bifet DATA STREAM MINING 2020-2021 course at Telecom Paris
- Alessio Bernardo & Emanuele Della Valle

It’s a streaming world!

It’s a streaming world …

Off-shore oil operations

Smart Cities

Power turbine

Social networks

Generate data streams!

E. Della Valle, S. Ceri, F. van Harmelen, D. Fensel **It's a Streaming World! Reasoning upon Rapidly Changing Information.** IEEE Intelligent Systems 24(6): 83-89 (2009)

… looking for reactive answers …

When a sensor on a drill in an oil-rig indicates that it is about to get stuck, how long can I keep drilling?

Where am I likely going to run into a traffic jam during my commute tonight?

Which electricity-producing turbine has sensor readings similar to any turbine that subsequently had a critical failure?

Who is driving the discussion about the top 10 emerging topics ?

Require continuous processing and reactive answer

… and conflicting requirements

A system able to answer those queries must be able to

- handle **volume**
- handle **velocity**
- handle **variety**
- cope with **incompleteness**
- cope with **noise**
- provide **reactive answers**
- support **fine-grained access**
- integrate **complex domain models**
- offer **high-level languages**

Stream Reasoning

- Research question
  - is it possible to **make sense in real time of multiple**, **heterogeneous**, **gigantic** and inevitably **noisy** and **incomplete data streams** in order **to support** the **decision processes** of extremely large numbers of concurrent users?

Emanuele Della Valle: On Stream Reasoning. PhD thesis, Vrije Universiteit Amsterdam, 2015. Available online at http://dare.ubvu.vu.nl/handle/1871/53293 .

Big Data Trend

Big Data Trend

Source: @LoriLewis and @OfficiallyChadd

Big Data Trend

***Data is growing**, and the rate of growth is accelerating. The sum of data generated by **2025** is set to accelerate exponentially to **175 zettabytes**, an **order of magnitude** bigger than the **storage** production **capability**.*

Dave Mosley,
CEO of SEAGATE TECHNOLOGY

Big Data Trend

***Innovation** is **not** driven **by trends**, but **by the need** to create **more value** under constraints. This exponential inflation will thus require **analysing** almost **30%** of global data in **real-time**.*

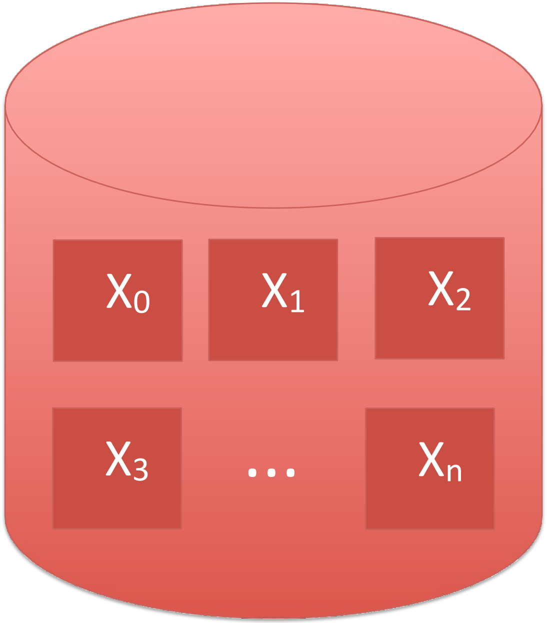
Dave Mosley,
CEO of SEAGATE TECHNOLOGY

Batch vs Data Stream

Batch

Random access to data

No restrictions on memory/time for training



Well defined training phase

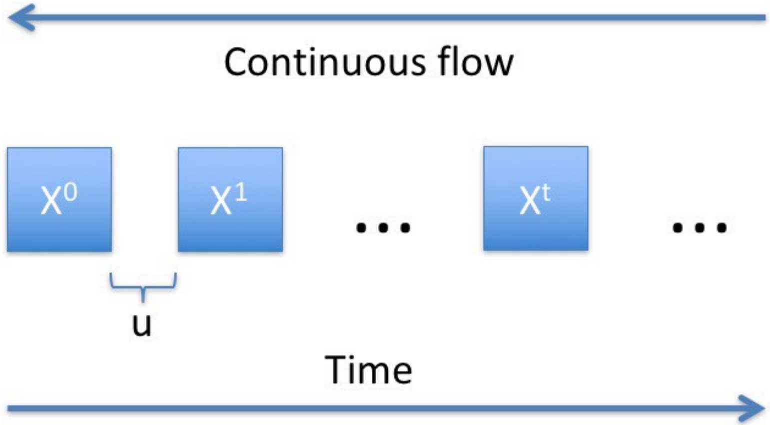
Access to all labeled data used for training

Data Stream

Continuous flow of data generated at **high-speed** in **dynamic**, **time-changing** environments.

Sequential access only

Evolving data

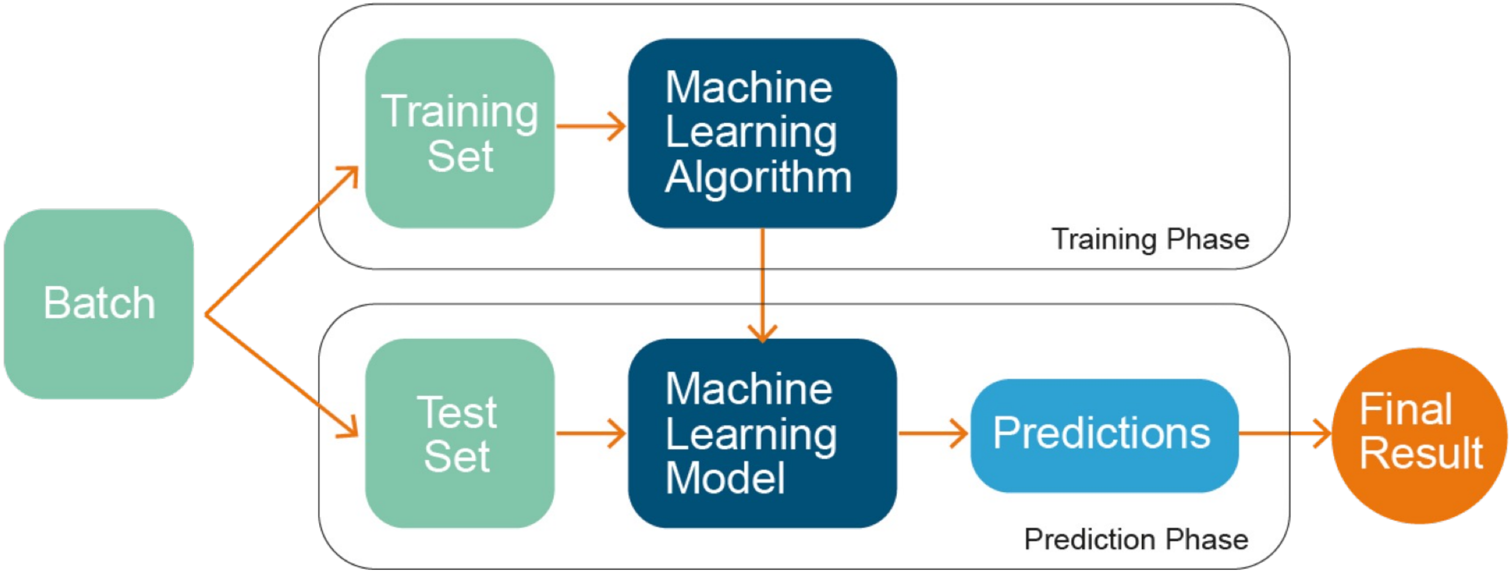


Strict time/memory requirements

Characteristics of data seen so far

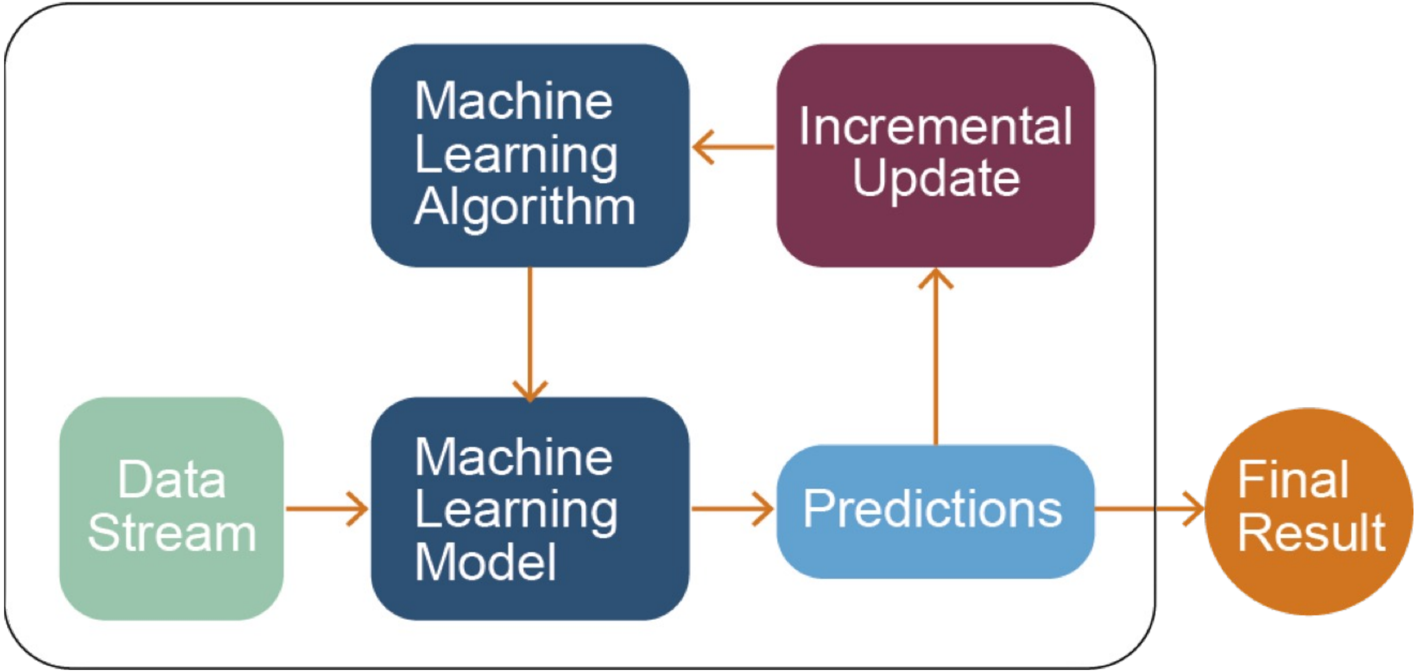
ML vs SML

ML Models

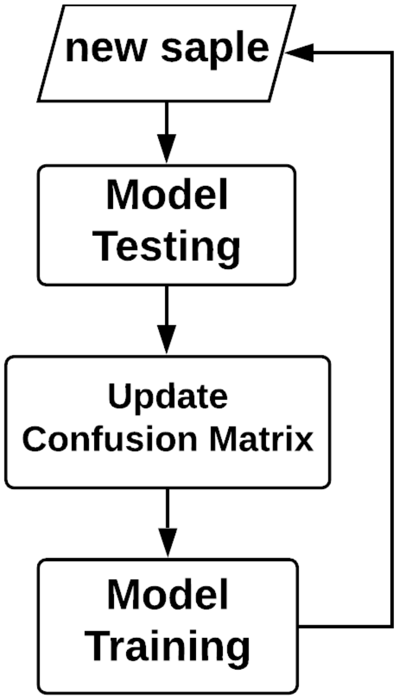


Batch
Training Set
Machine Learning Algorithm
Training Phase
Test Set
Machine Learning Model
Predictions
Final Result
Prediction Phase

SML Models

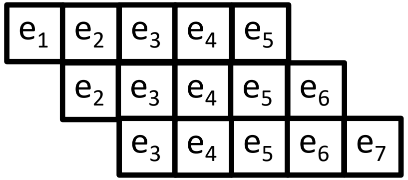

Machine Learning Algorithm
Incremental Update
Data Stream
Machine Learning Model
Predictions
Final Result

Prequential Evaluation



Estimate prequential error (PE):

- **Sliding window of size *w***



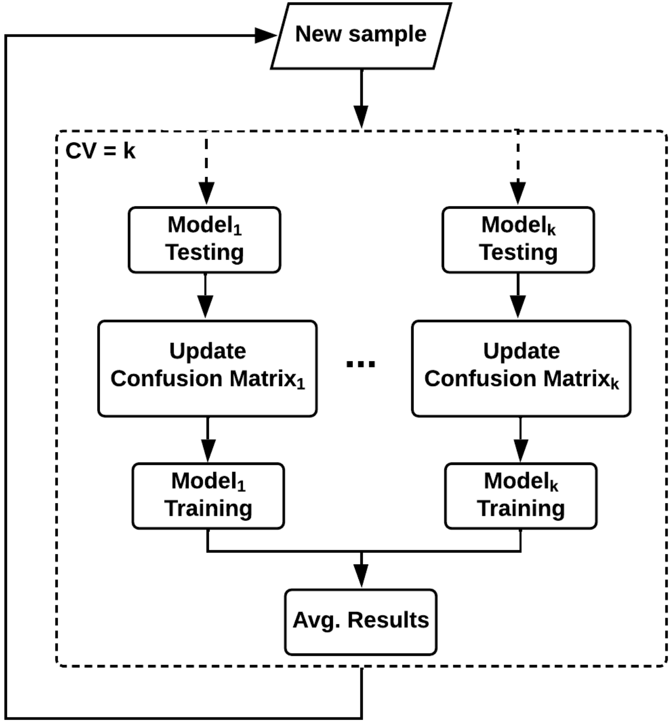
$$PE_i = \frac{1}{w} \sum_{k=i-w+1}^{w} e_k$$

- **Fading factor**

$$PE_i = \frac{\sum_{k=1}^{i} a^{i-k} * e_k}{\sum_{k=1}^{i} a^{i-k}} \qquad \text{with } 0 \ll \alpha \leq 1$$

Gama, J., Sebastião, R. and Rodrigues, P.P.: **Issues in evaluation of stream learning algorithms.** In ACM KDD, 2009.

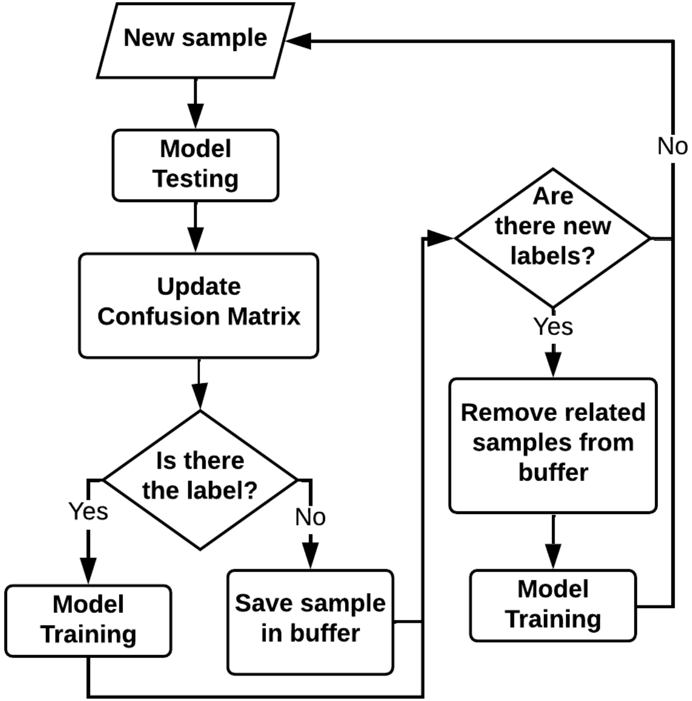
Prequential Evaluation – Cross Validation



- **K-fold distributed cross-validation:**
  each sample is used for testing in one classifier selected randomly, and used for training and testing all the others
- **K-fold distributed split-validation:**
  each sample is used for training in one classifier selected randomly, and for testing in all the classifiers
- **K-fold distributed bootstrap-validation:**
  each sample is used for training in approximately $^2/_3$ of the classifiers, with a separate weight in each classifier, and for testing in all the classifiers

Bifet, A., et al: **Efficient Online Evaluation of Big Data Stream Classifiers.** In ACM SIGKDD, 2015.

Prequential Evaluation – Delayed



- In real environments, can happen that the label arrives **delayed** w.r.t. the features
- Test the model with the features and wait for the label to train it

Gomes, HM., et al: **Adaptive random forests for evolving data stream classification.** In Machine Learning, 2017.

Evaluation metric – Kappa statistic

$$k = \frac{p - p_{rand}}{1 - p_{rand}}$$

where p is the accuracy of the classifier under consideration and p_{rand} is the accuracy of the Random classifier.

- If the classifier is perfectly correct, then $k = 1$.
- If the classifier achieves the same accuracy as the Random classifier, then $k = 0$.

I. Žliobaitè et al. **Evaluation methods and decision theory for classification of streaming data with temporal dependence.** In Machine Learning, 2015.

Evaluation metric – Kappa-Temporal statistic

$$k = \frac{p - p_{per}}{1 - p_{per}}$$

where p is the accuracy of the classifier under consideration and p_{per} is the accuracy of the Persistent classifier.

- If the classifier is perfectly correct, then $k = 1$.
- If the classifier achieves the same accuracy as the Persistent classifier, then $k = 0$.
- If the classifier performs worse then the Persistent classifier, then $k < 0$.

I. Žliobaitè et al. **Evaluation methods and decision theory for classification of streaming data with temporal dependence.** In Machine Learning, 2015.

SML Models

- Incorporate data on the fly
- Unbounded training sets
- Resource efficient
- Dynamic models

Benefits

- One sample at a time
- Incremental models
- Time and Memory management

Challenges

- Non-stationarity (Concept drift)
- Class imbalance
- Hyper-parameter Tuning

EXERCISE1: From batch to stream learning