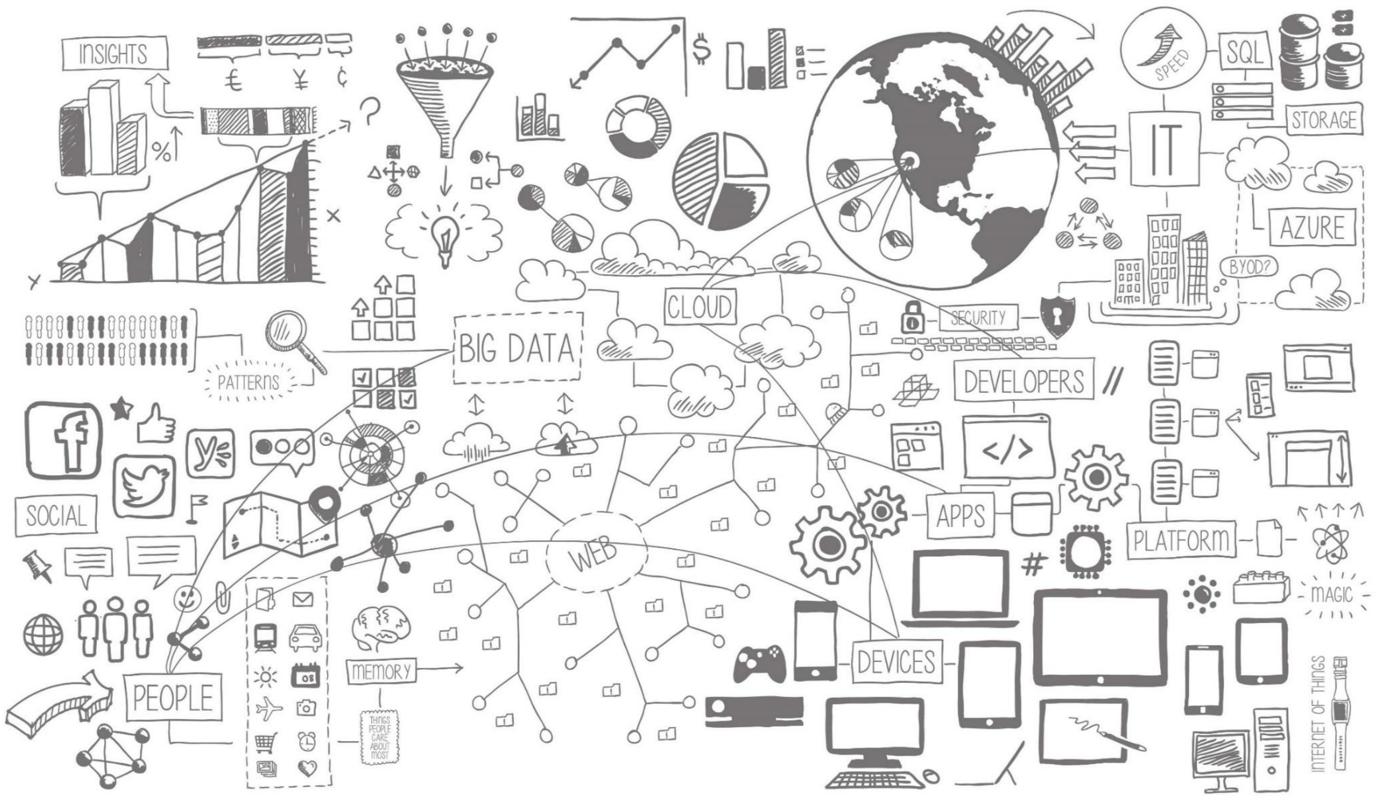
Statistical Machine Learning on

Evolving Non-stationary Data Streams

Outline

- Beyond i.i.d
- Stream Processing
- Incremental Learning
- Incremental Learning - Case Study
- Conclusion



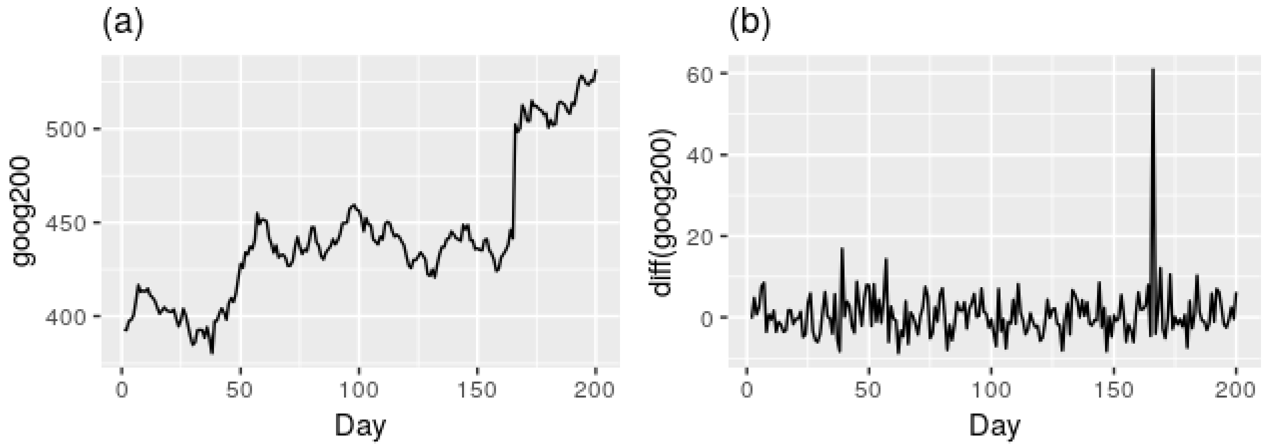
Beyond i.i.d

Beyond i.i.d

- **Traditional machine learning and data mining** assume the current observed data and the future data are **independent and identicically distributed (i.i.d)**.
- **Data samples**, in the **past and current time** do **not affect** the probability for **future** ones.

Beyond i.i.d

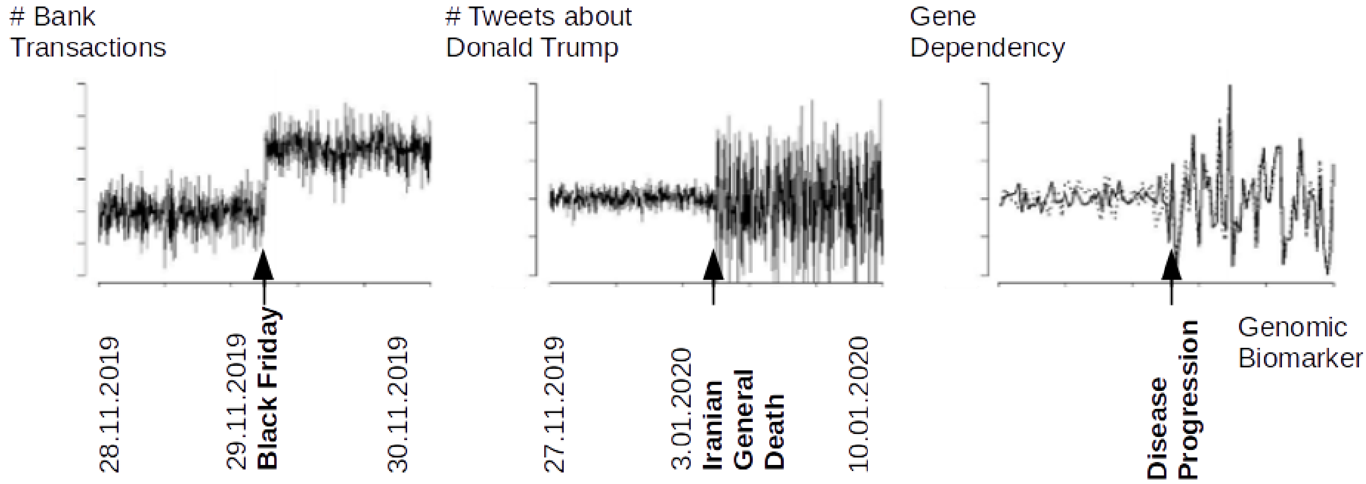
- Data samples arrive **continuously**, **online** through unlimited **streams** often at **high speed** [Sayed-Mouchaweh, 2016].
- The **process** generating these data streams **may evolve over time** (i.e. nonstationarity).



Beyond i.i.d

In order to deal with evolving data streams, the **model learnt from the streaming data** must capture up-to-date **trends** and **transient patterns** in the stream.

Types of change in regression

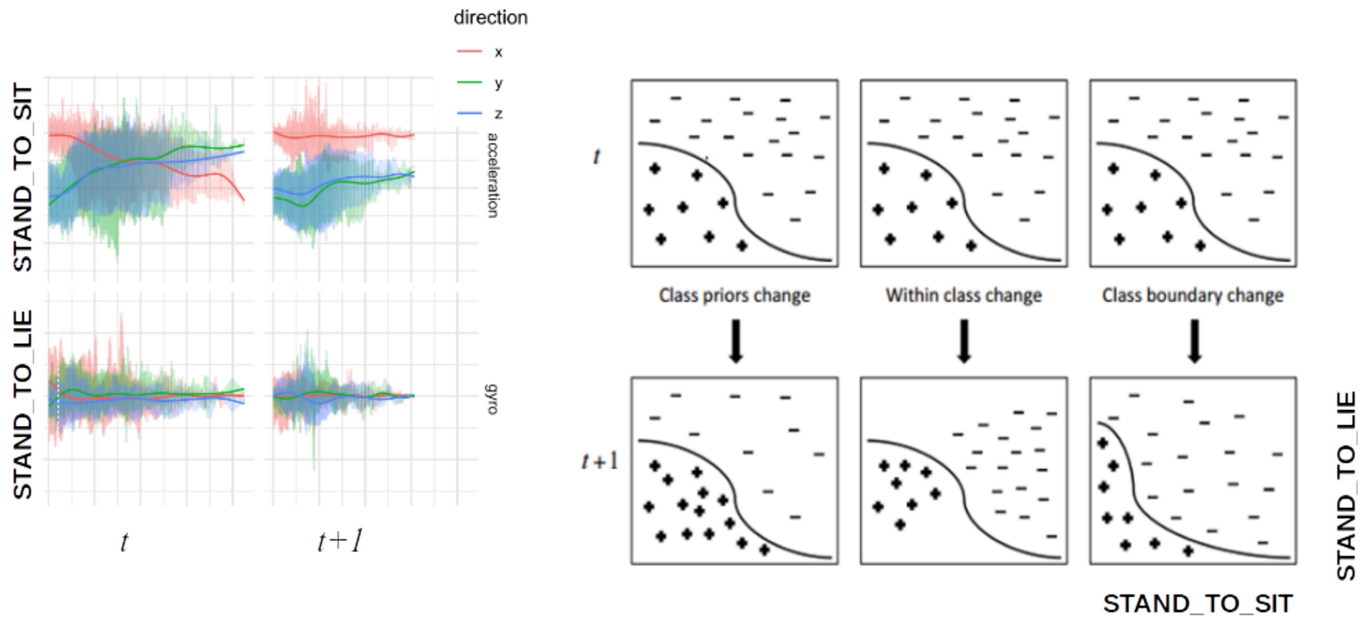


Beyond i.i.d

Updating the model by incorporating new examples, we must also **eliminate the effects of outdated examples** representing outdated concepts through **one-pass**.

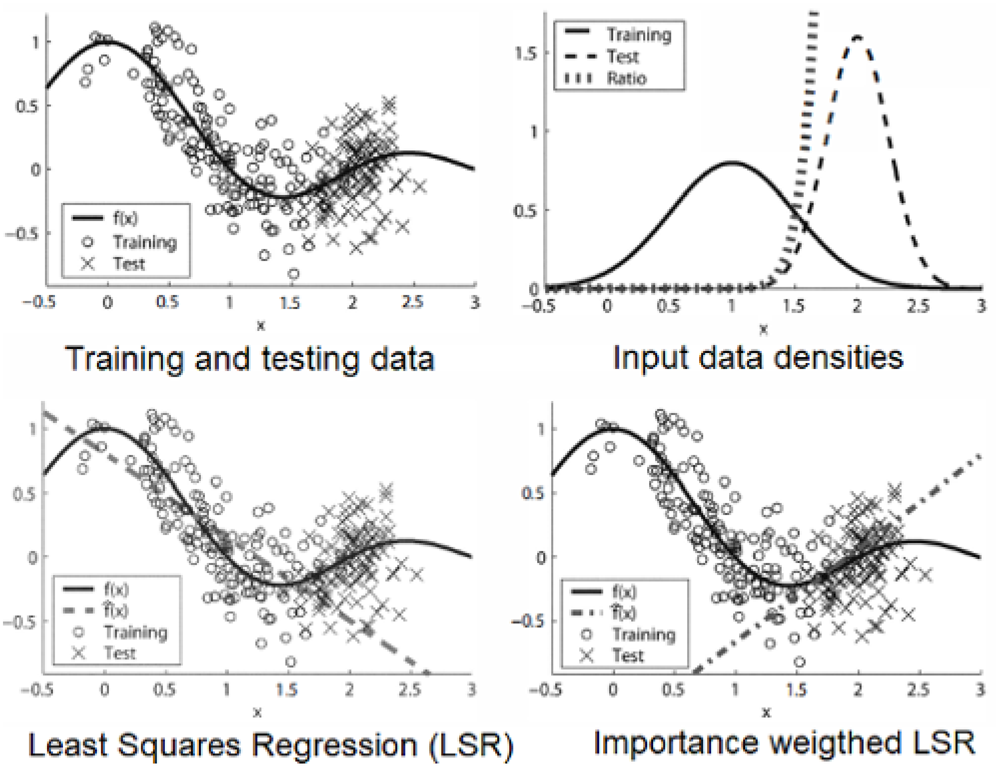
Typical changes in classification

[Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set (http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions)](http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions)



Beyond i.i.d

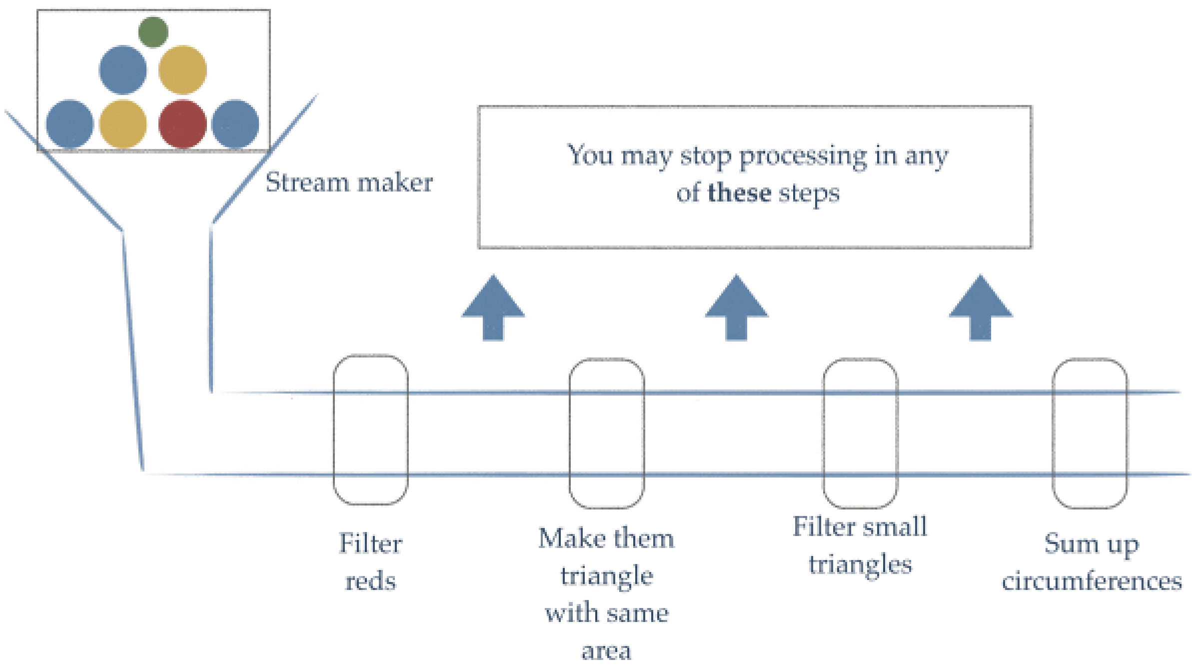
When **training and test data** follow **different probability distributions**, but the **conditional distributions of output** values given input points (i.e. **target function**) are **unchanged**, we face **covariate shift** [Sugiyama et al., 2012].



Stream Processing

Stream Processing

Given a sequence of data (**a stream**), a series of operations (functions) is applied to each element in the stream, in a declarative way, we specify **what we want to achieve and not how** [Bifet, 2010].



Stream Processing

Data **Stream Processing** is more challenging than **batch processing** [Bifet, 2010]:

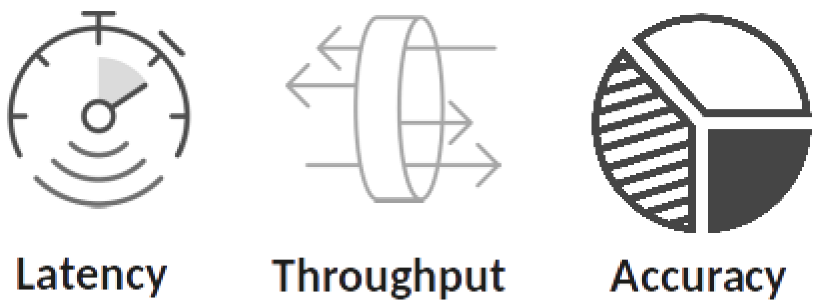
- The amount of data is **extremely large**, potentially infinite - **impossible to store**
- Only a **small summary** can be computed and stored, and the rest is discarded - unfeasible to go over it
- The **speed of arrival is high**, so that each datum has to be processed in **real time**, and then discarded
- The **distribution generating the items** can **change over time**
- **Data from the past** may become **irrelevant (or even harmful)** for the current summary

Incremental Learning

Incremental Learning

Unlike **conventional machine learning**, the **data flow** targeted by **incremental learning** becomes available **continuously** over time and needs to be **processed in a single pass**.

The inherent **challenges** here are:



Incremental Learning - Case Study

Principal Component Analysis (PCA)

PCA

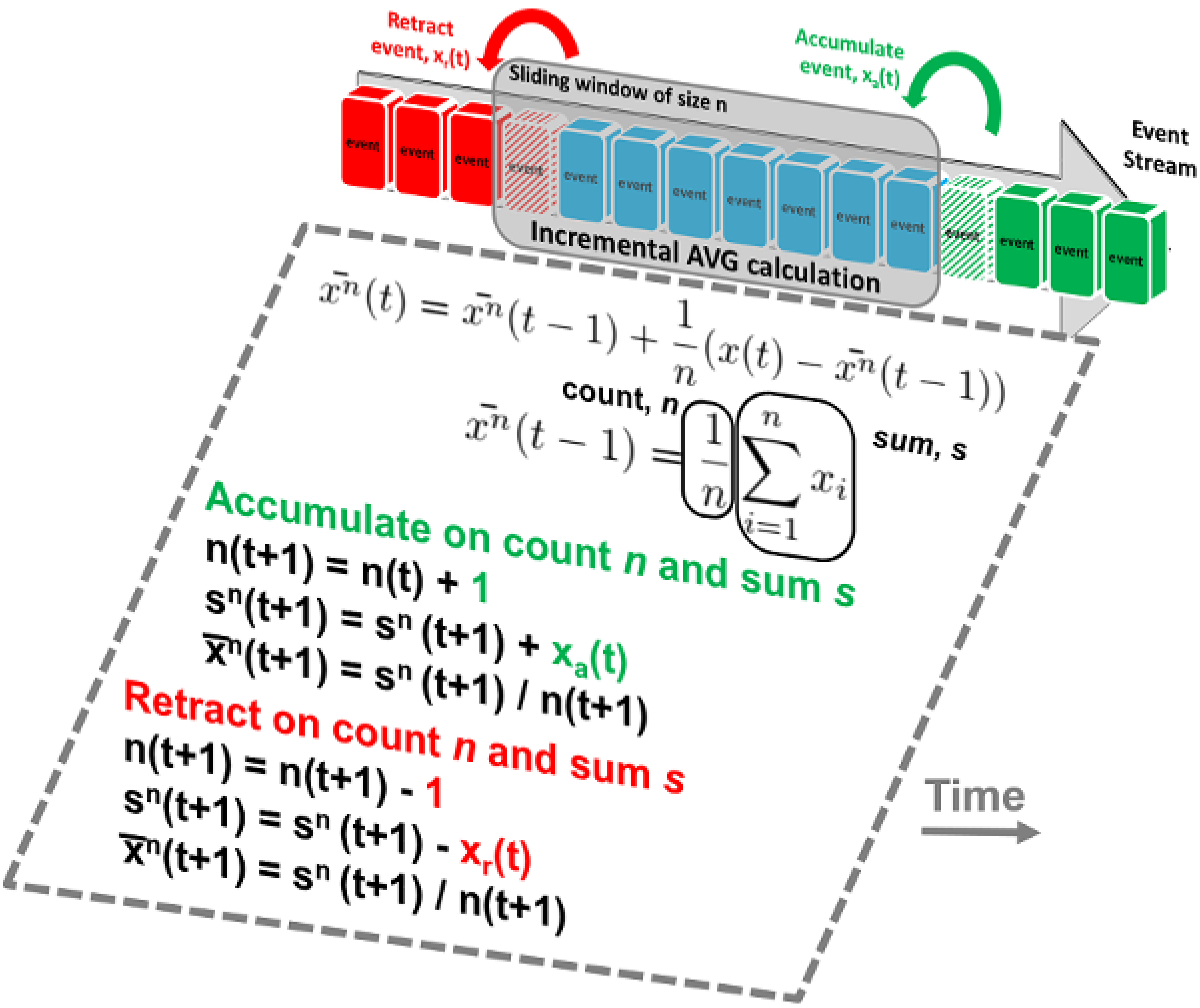
Data: Stream of n-dimensional data vectors $x(t)$
Result: Eigen features w_p and λ_p
Compute the mean feature vector $\mu = \frac{1}{n}\sum_k^n x_k(t)$;
Compute the covariance matrix $C = E\{x - \mu\}\{x - \mu\}^T$;
Compute eigenvalues λ_i and eigenvectors z_i of C;
while *Estimate high-value eigenvectors* **do**
  Sort eigenvalues λ_i in decreasing order;
  Choose a threshold θ :
  Choose the first p dominant λ_i to satisfy $(\sum_i^p \lambda_i)(\sum_i^n \lambda_i)^{-1} \geq \theta$;
  Select eigenvectors w_p corresponding to λ_p;
end
Extract principal components from x, $P = V^T x$ where V is the matrix of principal components

Tackle the **inherent problems in traditional PCA** impeding it to allow it to learn incrementally:

- Calculation of the **mean and other descriptive statistics** as the data is available
- **Sorting the dominant eigenvalues** in the rank update of the QR decomposition
- Calculating the **covariance matrix**

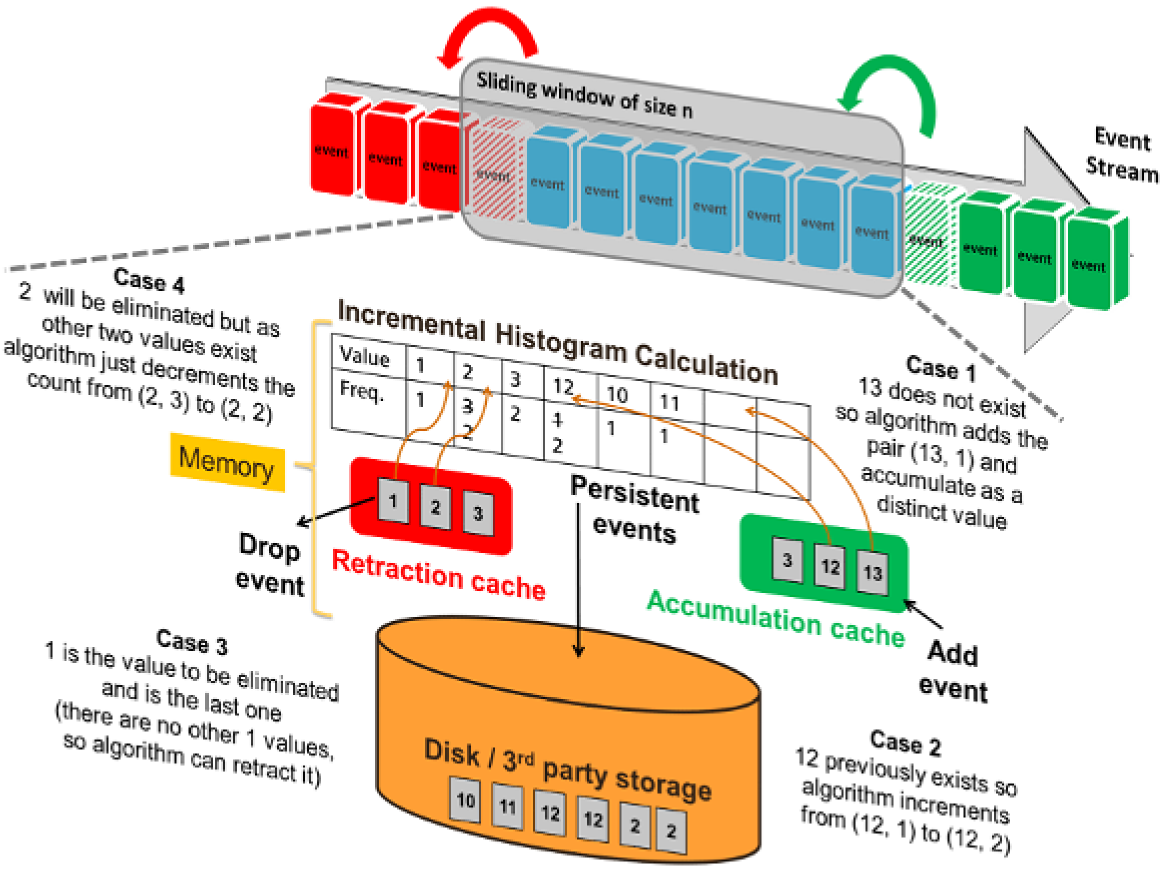
Towards incremental PCA

Incremental **calculation of the mean and other descriptive statistics** on the datastream.



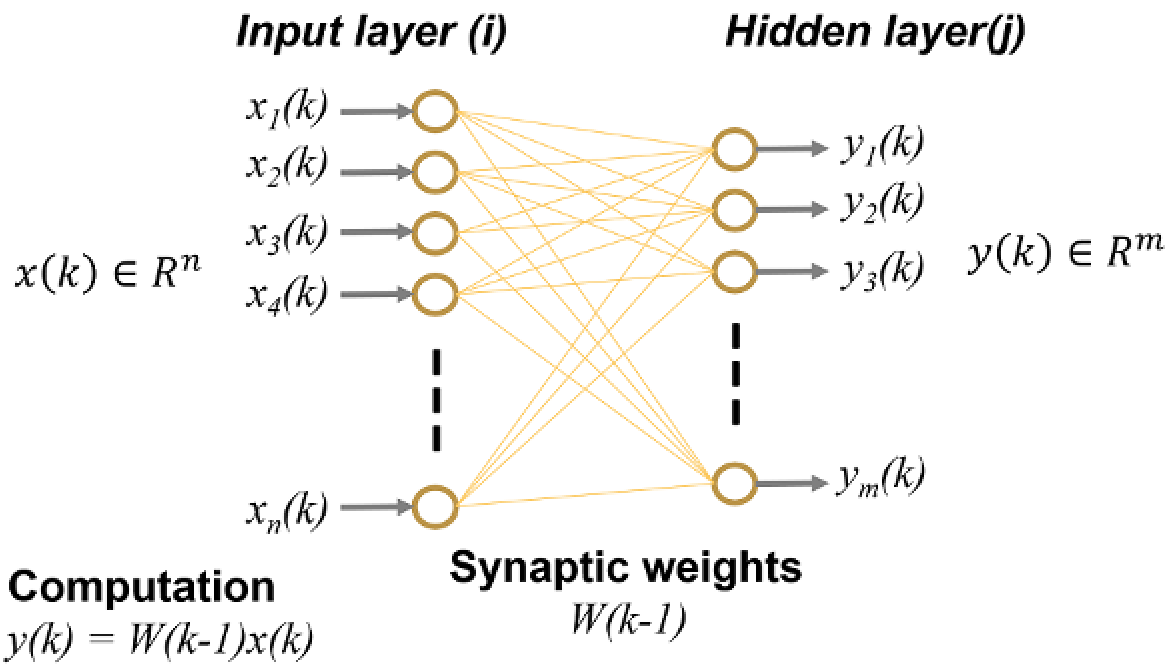
Towards incremental PCA

Incremental **updates depending on counts (i.e. histogram)**, which contain sorted eigenvalues.

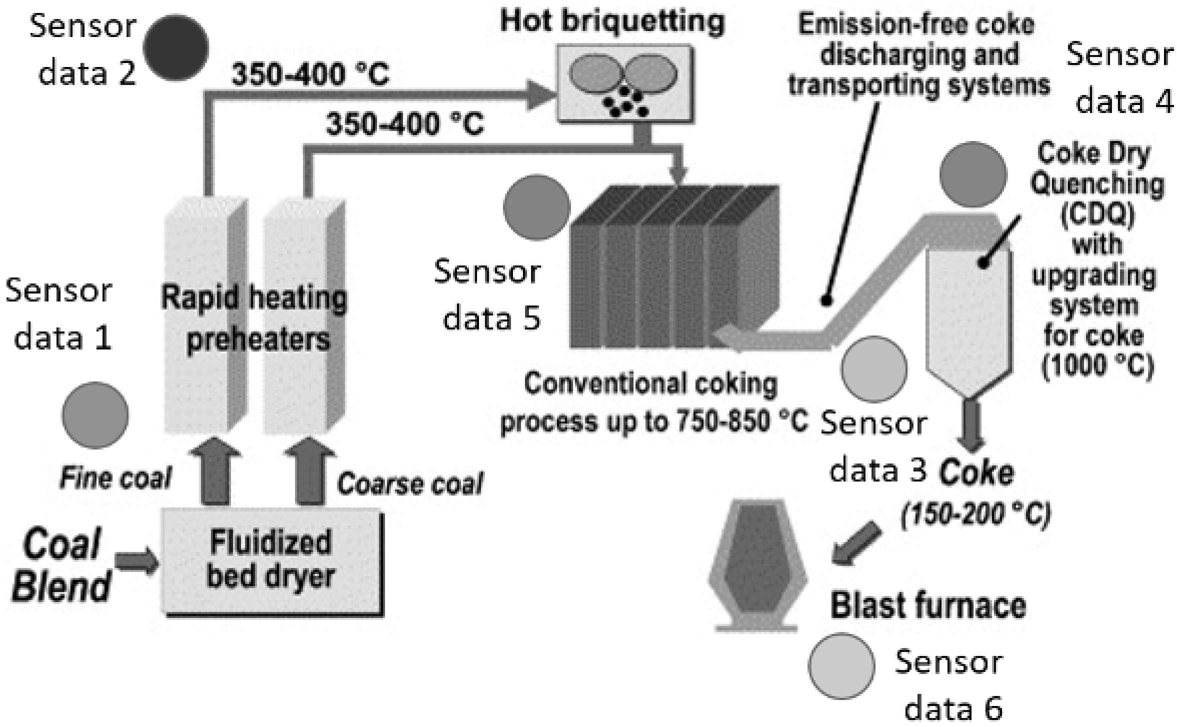


Towards incremental PCA

Incremental **estimation of the covariance matrix**, as **neural synaptic weights** converge to the eigenvectors (unique set of **optimal weights** and **uncorrelated outputs**) [Axenie et al., 2019].



Real-world application



Multi-class classification task (fault identification in predictive maintenance [Axenie et al., ECML PKDD 2019])

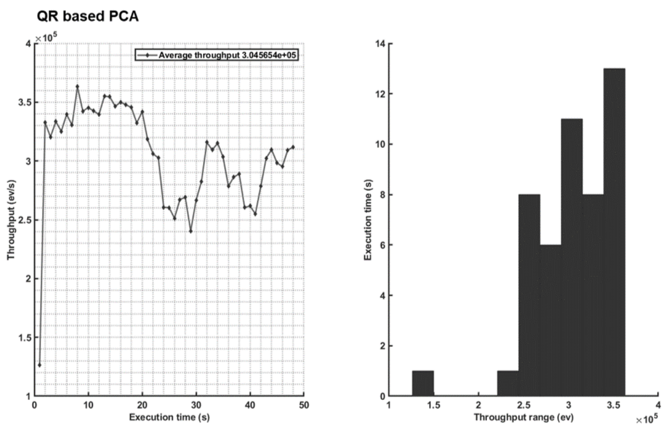
Goal: identify faults in the production line by querying the eigenvalues and eigenvectors to extract the normal and faulty operation configuration prior to a multi-class classifier.

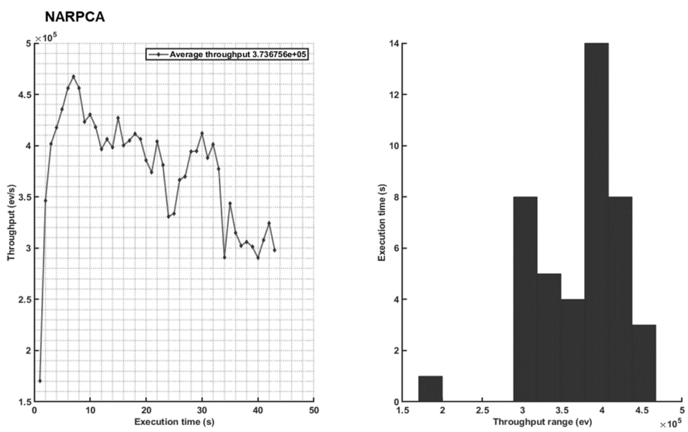
Real-world application

The datastream:

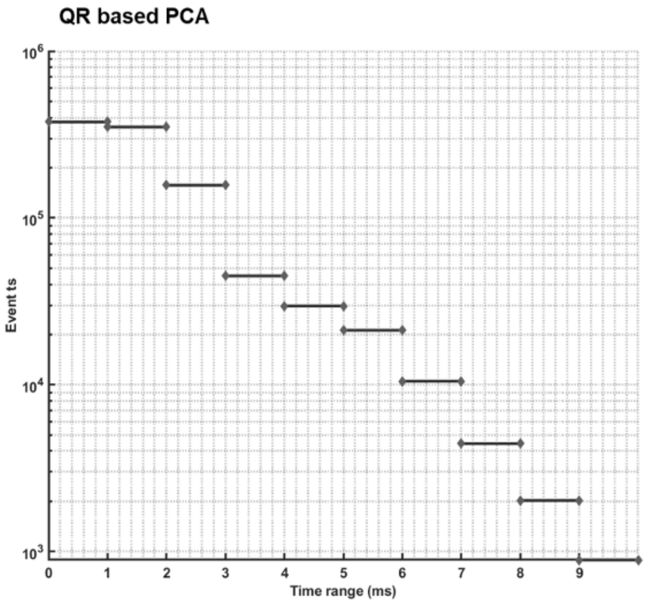
- 2M incoming events at 40 kHz.
- eigenvalues of the input X are close to the class labels (i.e. $1, 2, \ldots, d$)
- eigenvectors are close to the canonical basis of R^d, where d is the number of principal components to extract class number for the multi-class classification task (i.e. 10 classes, 9 faults and 1 normal).

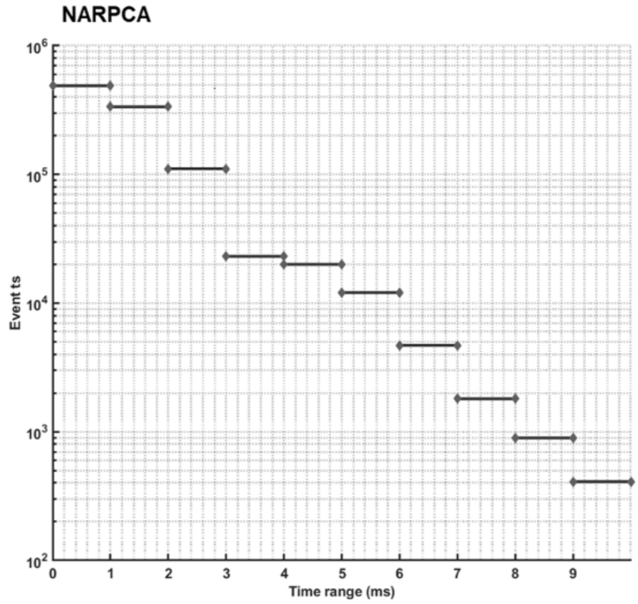
Performance analysis: Throughput





Performance analysis: Latency





Performance analysis: Overall

PCA Model	NARPCA	QRPCA
Avg. Latency (ms)	0.96664	1.19262
Avg. Throughput(ev/s)	373675	304565

Performance analysis: Accuracy

Eigenvalue	Eigenvalue estimate	Eigenvector variance
1	0.994071965	0.033719458
2	1.99658601	0.023661145
3	3.00600192	0.013884741
4	4.00420688	0.025106736
5	5.04173253	0.022354039
6	5.95475267	0.007637369
7	6.88985141	0.011129644
8	7.87972194	0.015864081
9	8.90795326	0.007244545
10	10.0642228	0.014663302

Conclusions

Conclusions

- Incremental learning on data streams is a challenging problem
- Data size, data speed and the underlying changes in the data properties yield new learning models
- Model accuracy, processing latency and processing throughput are usually a trade-off

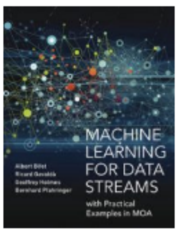
References

[Bifet, 2010] Albert Bifet - Adaptive Stream Mining Pattern Learning and Mining from Evolving Data Streams, IOS Press, 2010.

[Sugiyama et al., 2012] Masashi Sugiyama, Motoaki Kawanabe - Machine Learning in Non-Stationary Environments Introduction to Covariate Shift Adaptation-The MIT Press, 2012.

[Sayed-Mouchaweh, 2016] Moamar Sayed-Mouchaweh - Learning from Data Streams in Dynamic Environments-Springer International Publishing, 2016.

[Axenie et al., 2019] C. Axenie, Radu Tudoran, Stefano Bortoli, Mohamad Al Hajj Hassan, Alexander Wieder, Goetz Brasche, SPICE: Streaming PCA fault Identification and Classification Engine in Predictive Maintenance, IoT Stream Workshop, European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2019).



MOA is the most popular open source framework for data stream mining, with a very active growing community (blog). It includes a collection of machine learning algorithms (classification, regression, clustering, outlier detection, concept drift detection and recommender systems) and tools for evaluation. Related to the WEKA project, MOA is also written in Java, while scaling to more demanding problems. A new book on MOA has been published at MIT Press.

Lecture notebook download

In [ ]: