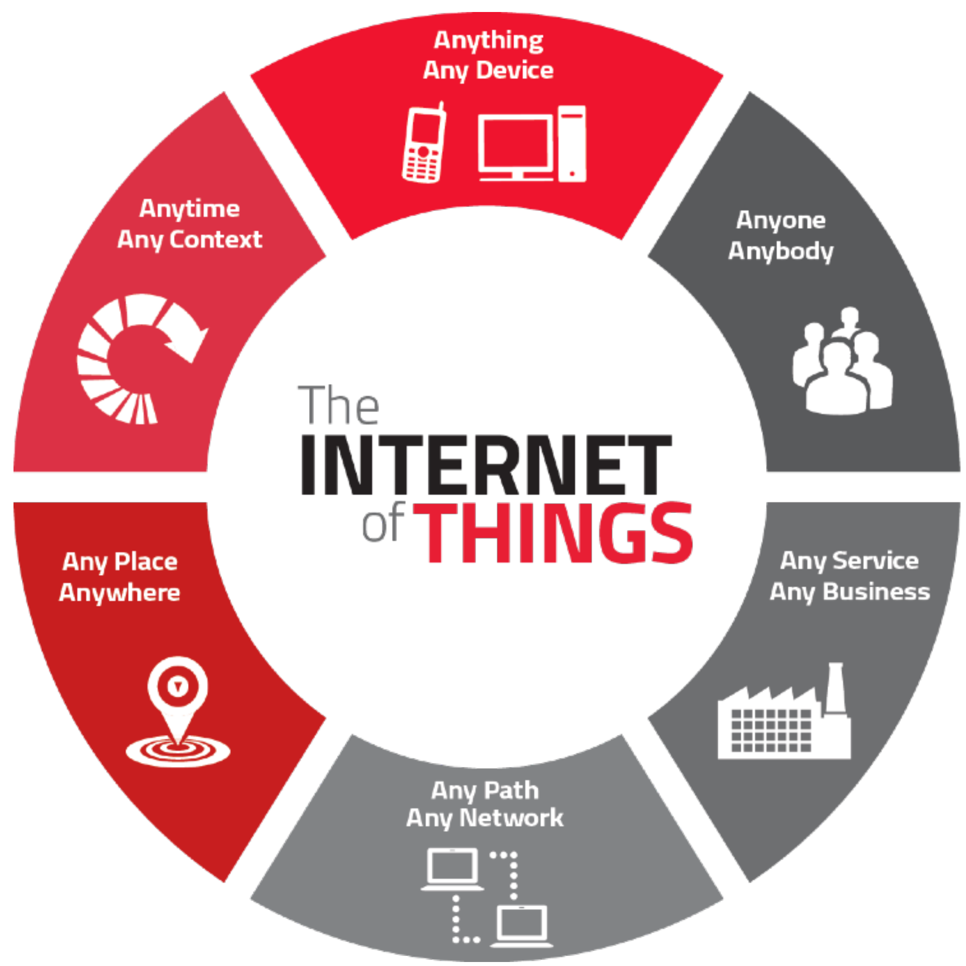
Massive Online Analytics (MOA) **for the** **Internet of Things** (IoT)

Albert Bifet (@abifet)

SystemX, 14 September 2017

IoT Setting

INTERNET OF THINGS

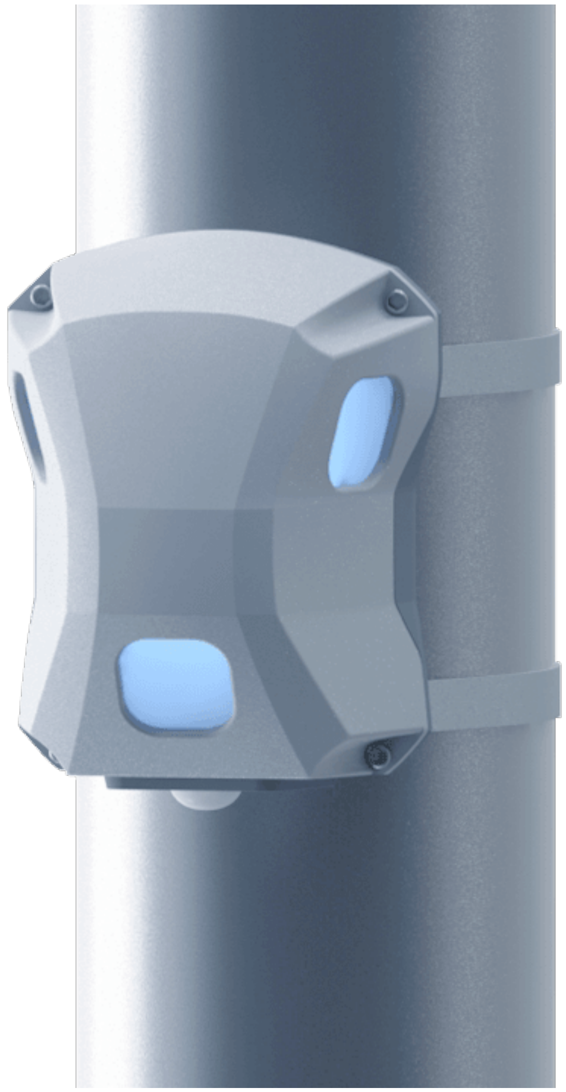


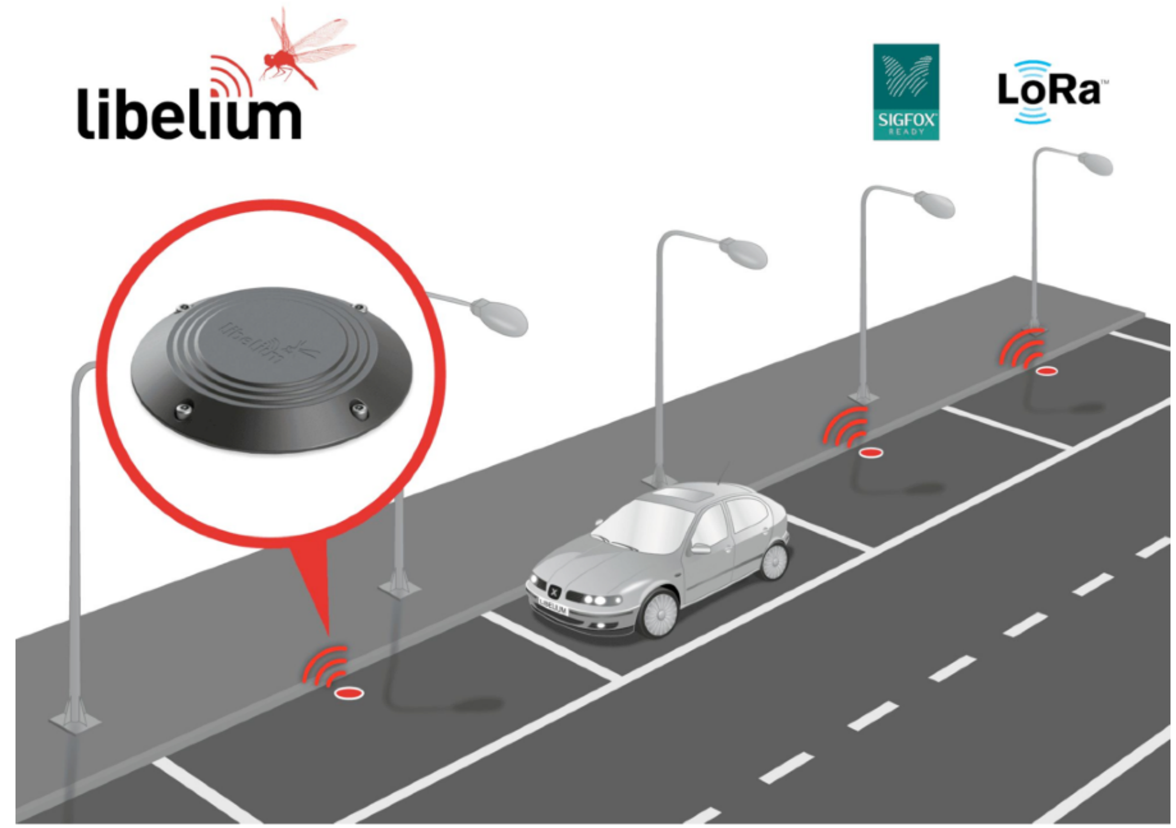
IoT: sensors and actuators connected by networks to computing systems.

- Gartner predicts 20.8 billion IoT devices by 2020.
- IDC projects 32 billion IoT devices by 2020

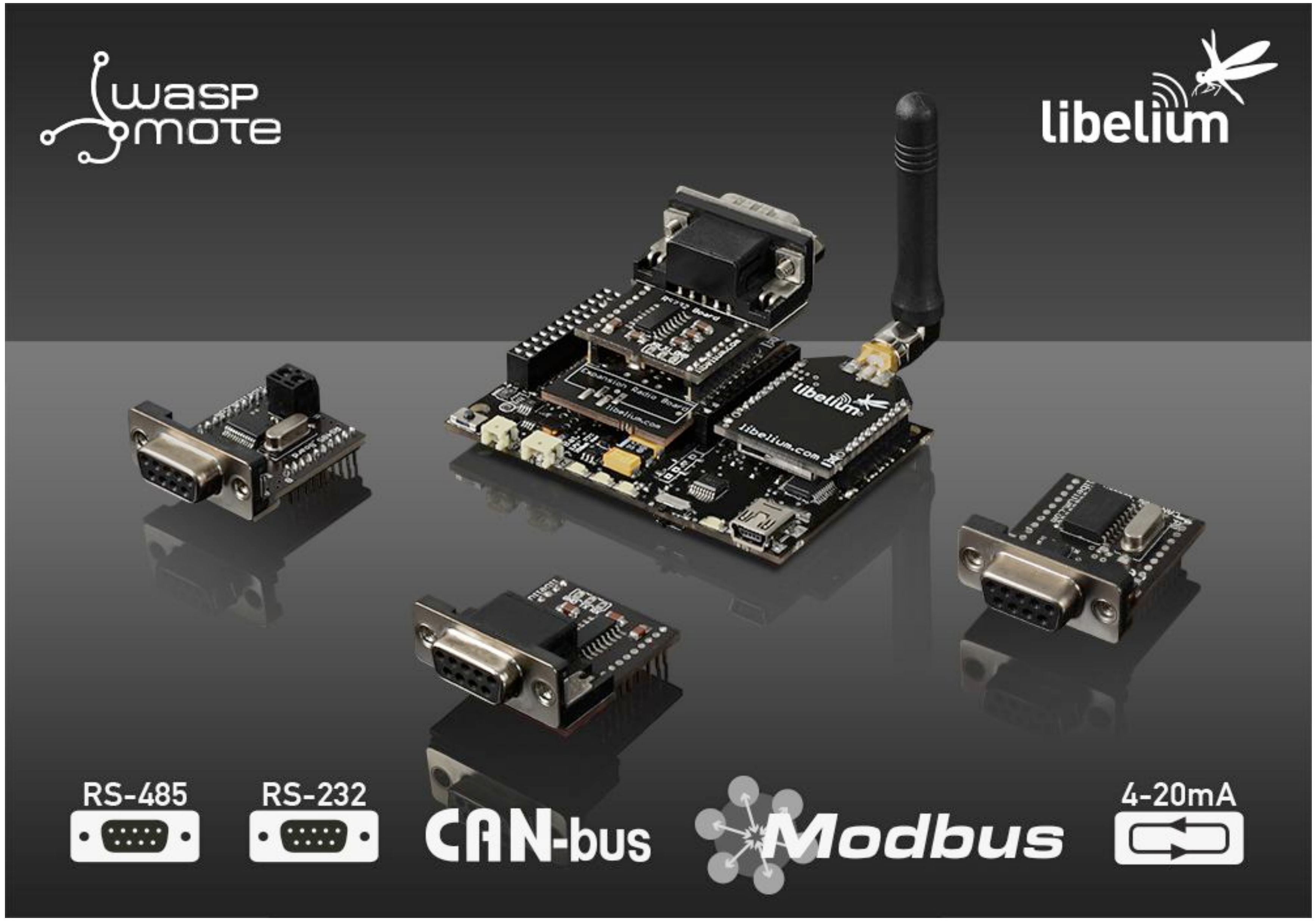
IoT Applications For Energy Management

IoT Applications For Connected/Smart Home





IoT Applications For Smart Cities



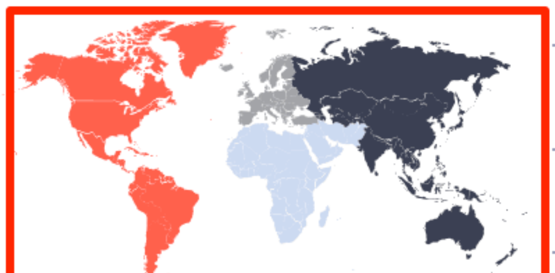
IoT Applications For Industrial Automation

Applications IoT Analytics

IOT ANALYTICS | **Q3/2016** | *Insights that empower you to understand IoT markets*

	IoT Segment	Global share of IoT projects[1]	Americas	Europe	APAC
1	Connected Industry	22%	43%	30%	20%
2	Smart City	20%	31%	47%	15%
3	Smart Energy	13%	49%	24%	25%
4	Connected Car	13%	43%	33%	17%
5	Other	8%	46%	33%	13%
6	Smart Agriculture	6%	48%	31%	17%
7	Connected Building[3]	5%	48%	33%	12%
8	Connected Health	5%	61%	30%	6%
9	Smart Retail	4%	52%	30%	13%
10	Smart Supply Chain	4%	57%	35%	4%

Details — Trend[2]



N = 640 global, publicly announced IoT projects

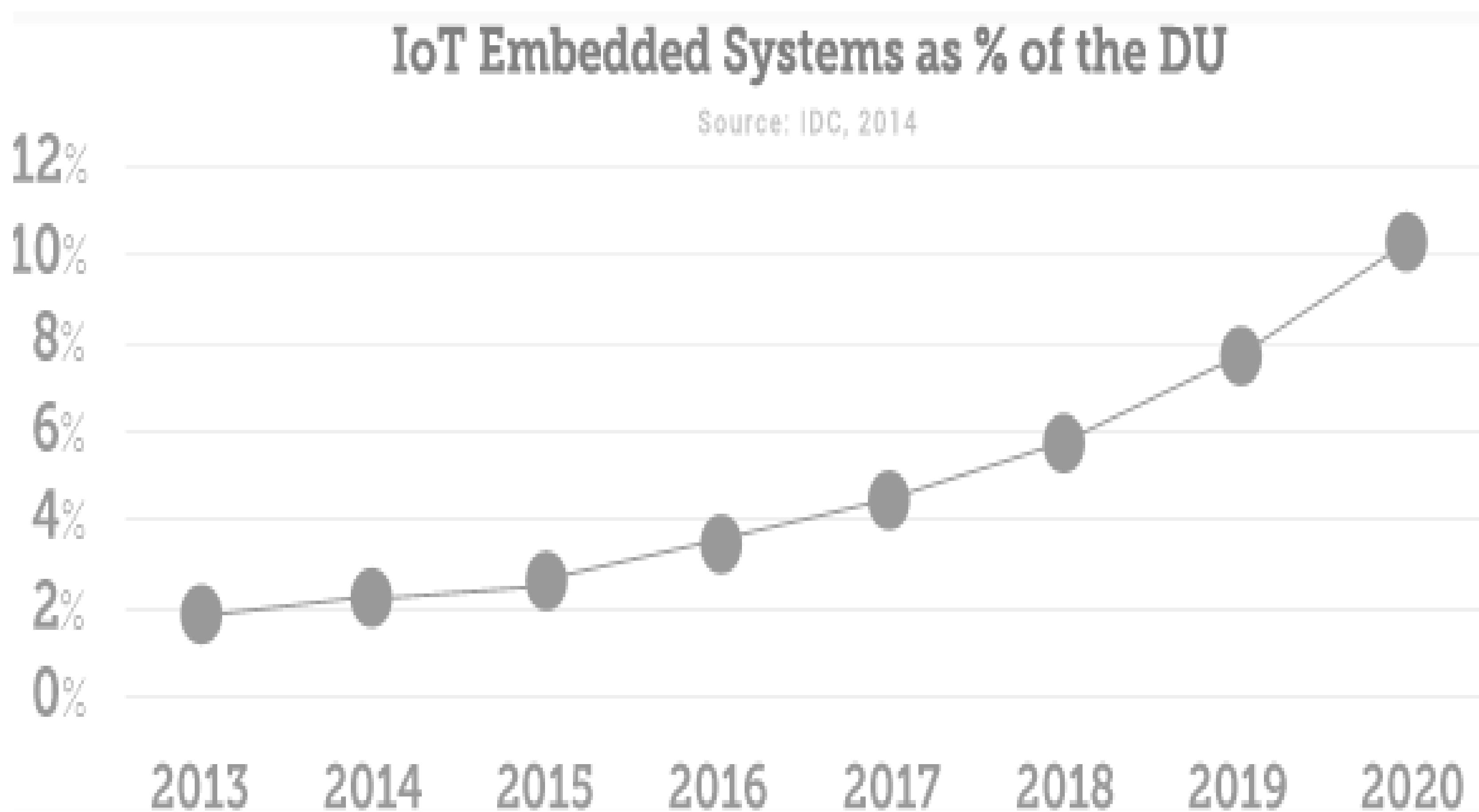
Americas | Europe | APAC | MEA | N/A

1. Based on 640+ publicly known enterprise IoT projects.(Not including consumer IoT projects e.g., Wearables, Smart Home) 2. Trend based on IoT Analytics's Q2/2016 IoT Employment Statistics Tracker 3. Not including Consumer Smart Home Solutions Source: IoT Analytics 2016 Global overview of 640 enterprise IoT use cases (August 2016)

IOT AND INDUSTRY 4.0

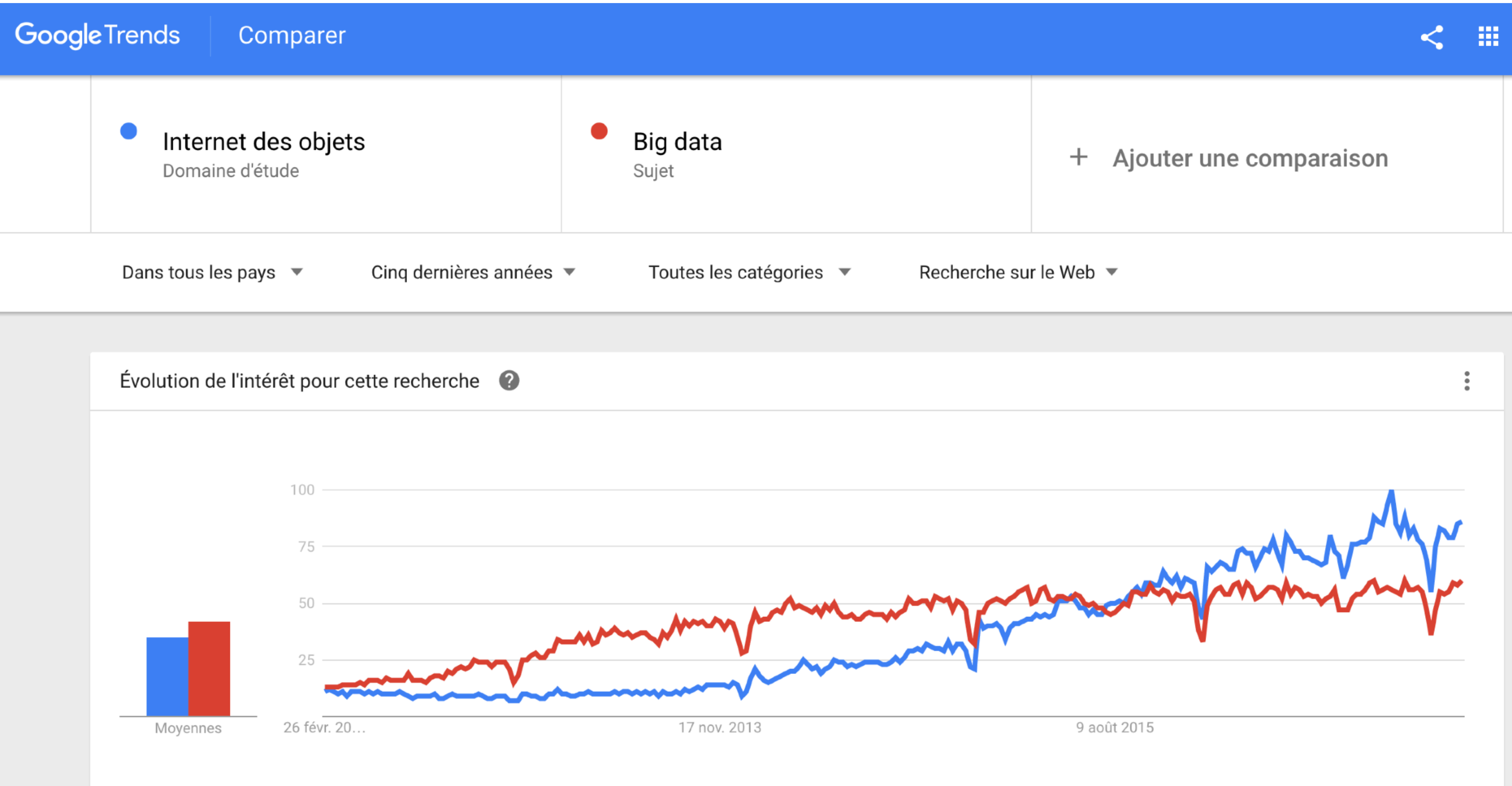
- Interoperability: IoT
- Information transparency: virtual copy of the physical world
- Technical assistance: support human decisions
- Decentralized decisions: make decisions on their own

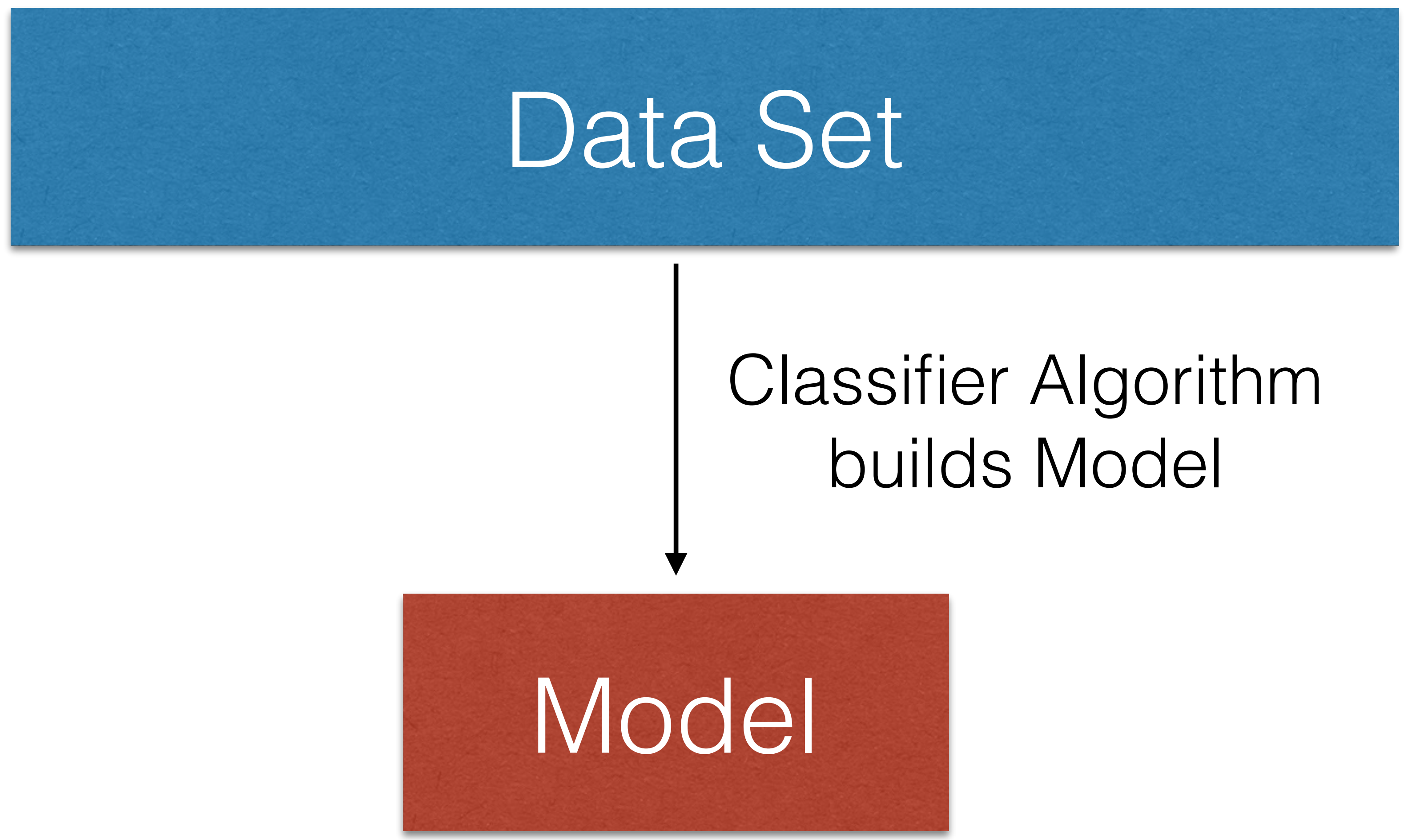
INTERNET OF THINGS



- EMC Digital Universe, 2014

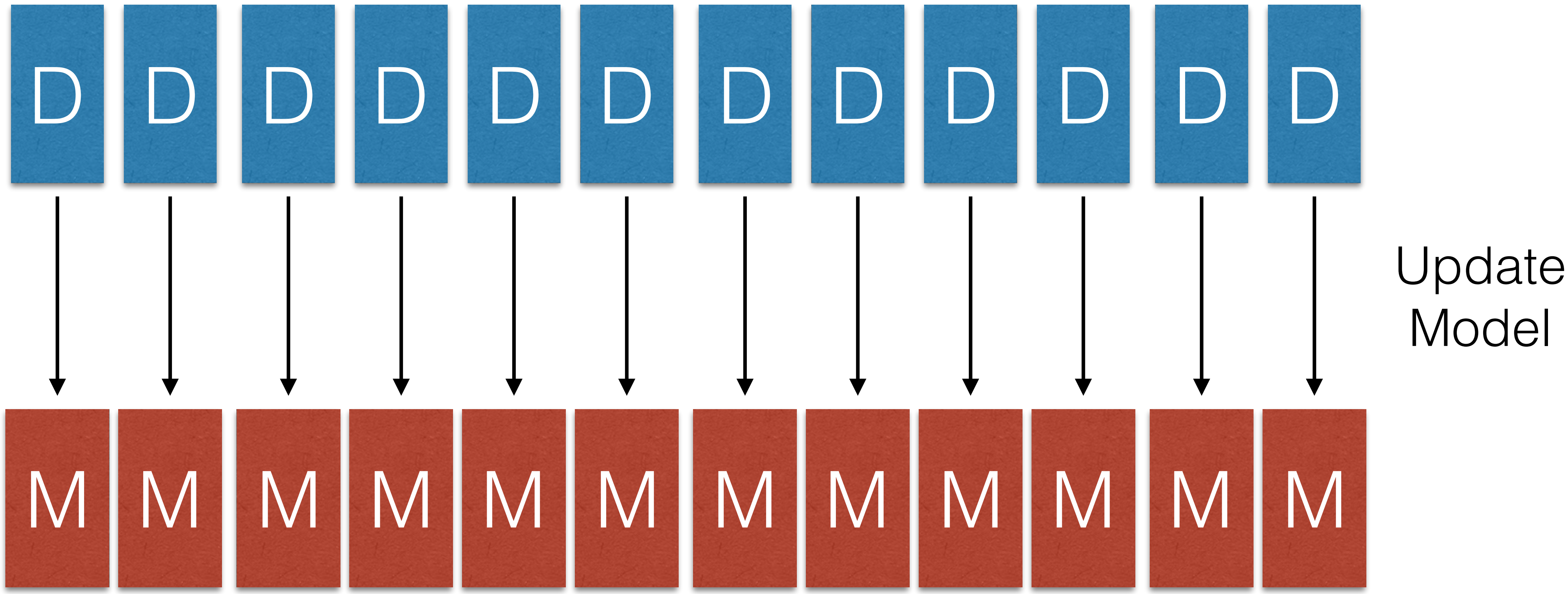
IoT versus Big Data





Analytic Standard Approach

Finite training sets
Static models

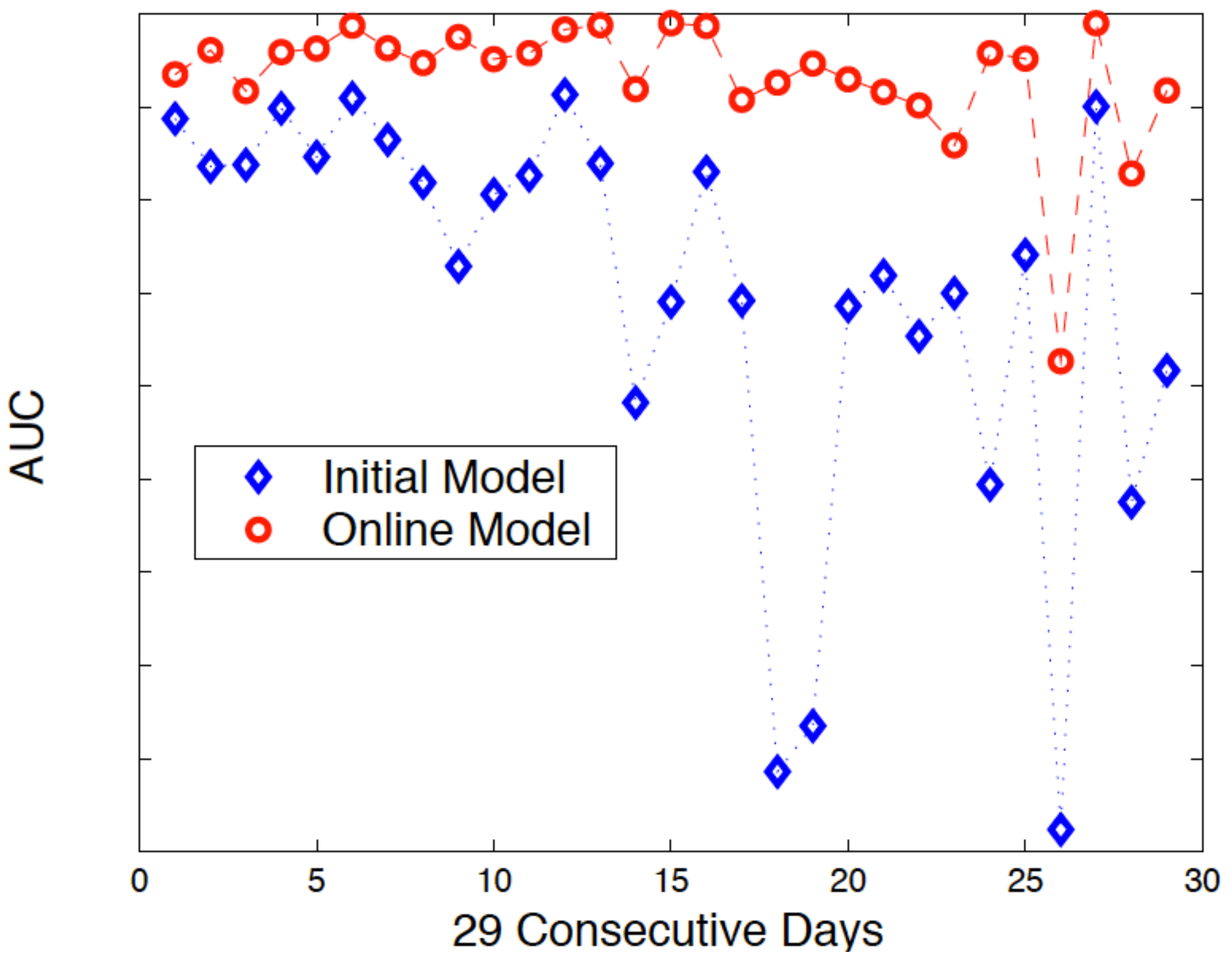


Data Stream Approach

Infinite training sets
Dynamic models

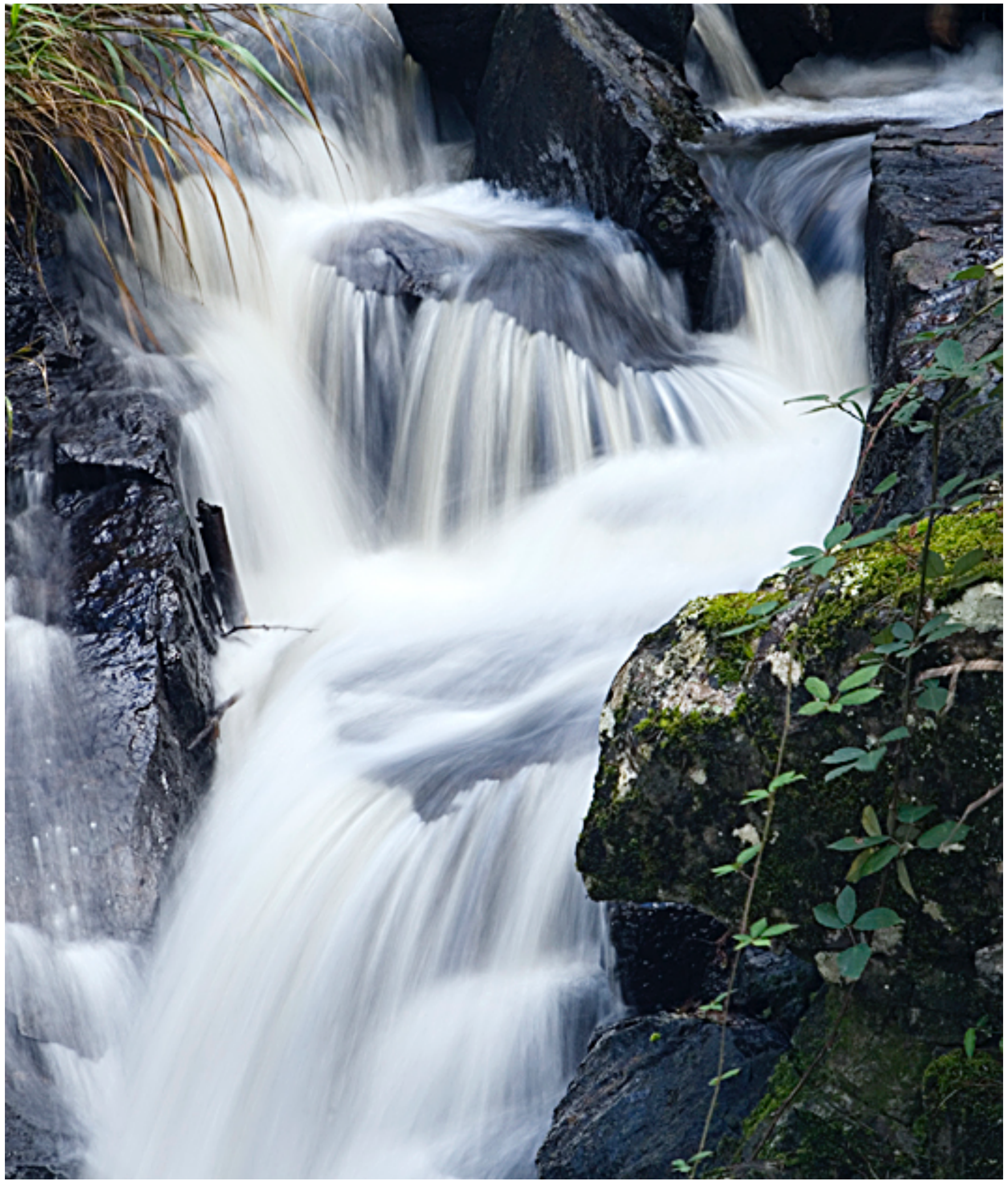
Pain Points

- Need to **retrain!**
  - Things change over time
  - How often?
- Data unused until next update!
  - Value of data wasted



IoT Stream Mining

- Maintain models online
  - Incorporate data on the fly
  - Unbounded training sets
  - Resource efficient
  - Detect changes and adapts
  - Dynamic models



Approximation Algorithms

- General idea, good for streaming algorithms
- Small error ε with high probability $1-\delta$
    - True hypothesis H, and learned hypothesis $\hat{H}$
    - $\Pr[\ |H - \hat{H}| < \varepsilon|H|\ ] > 1-\delta$

Approximation Algorithms

- What is the largest number that we can store in 8 bits?

1	0	1	0	1	0	1	0

Approximation Algorithms

- What is the largest number that we can store in 8 bits?

Programming Techniques

S.L. Graham, R.L. Rivest
Editors

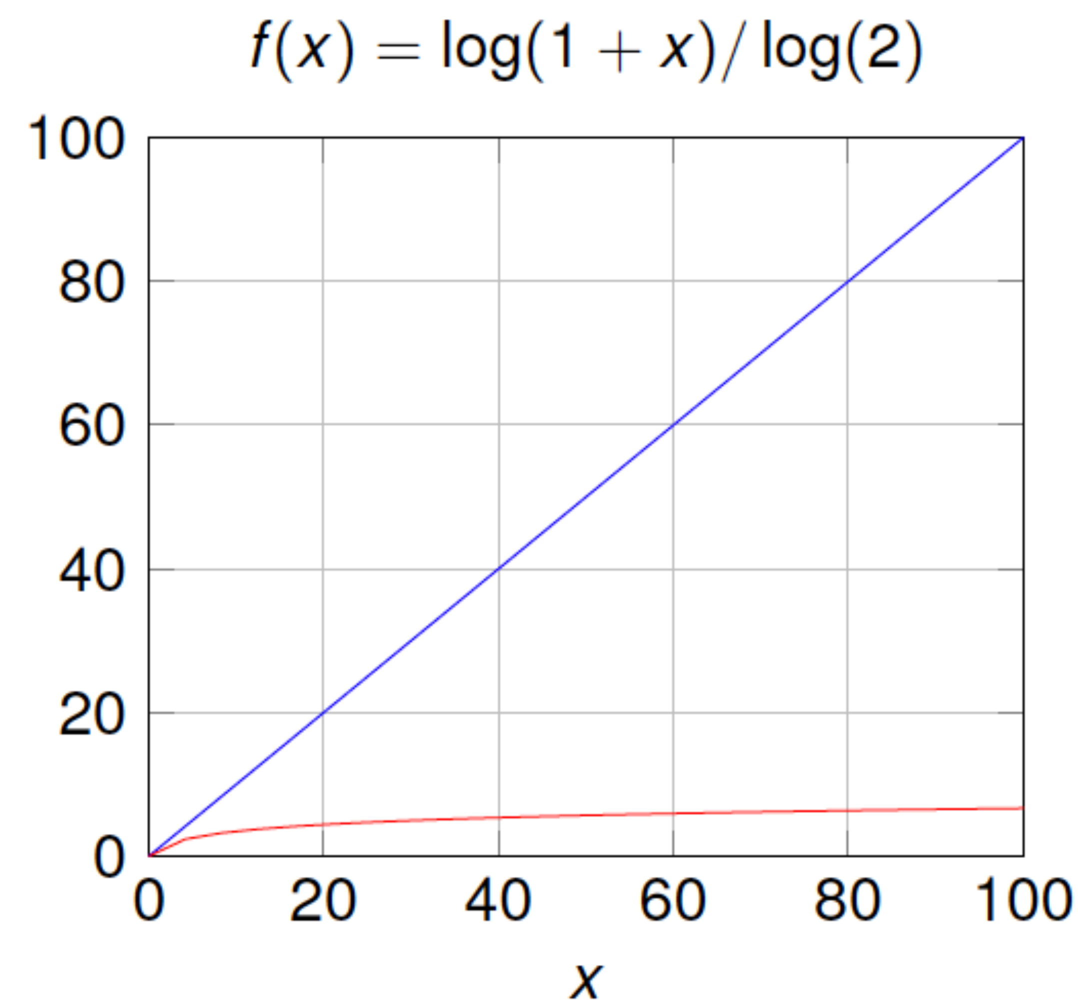
Counting Large Numbers of Events in Small Registers

Robert Morris
Bell Laboratories, Murray Hill, N.J.

It is possible to use a small counter to keep approximate counts of large numbers. The resulting expected error can be rather precisely controlled. An example is given in which 8-bit counters (bytes) are used to keep track of as many as 130,000 events with a relative error which is substantially independent of the number n of events. This relative error can be expected to be 24 percent or less 95 percent of the time (i.e. $\sigma = n/8$). The techniques could be used to advantage in multichannel counting hardware or software used for the monitoring of experiments or processes.

Approximation Algorithms

- What is the largest number that we can store in 8 bits?

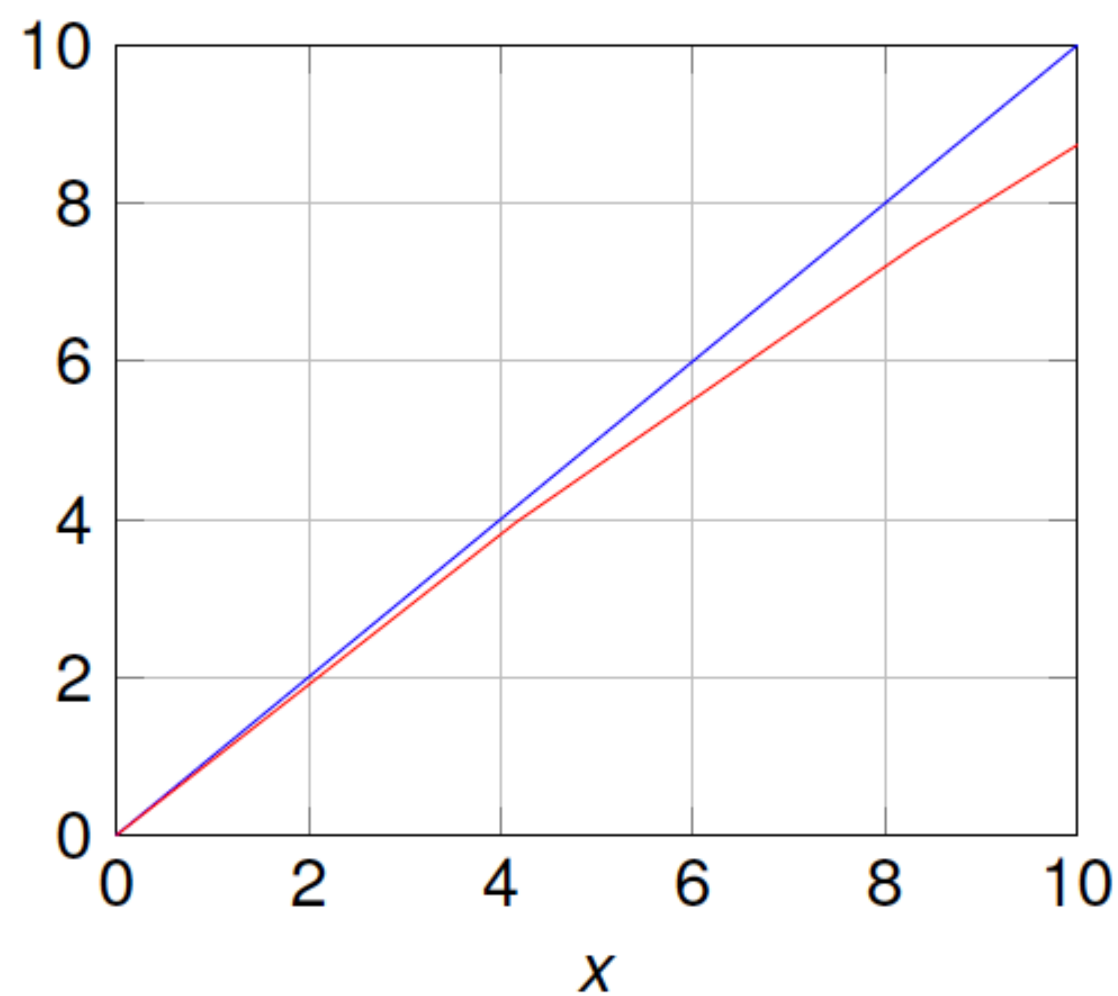


$f(0) = 0, f(1) = 1$

Approximation Algorithms

- What is the largest number that we can store in 8 bits?

$f(x) = \log(1 + x/30)/\log(1 + 1/30)$

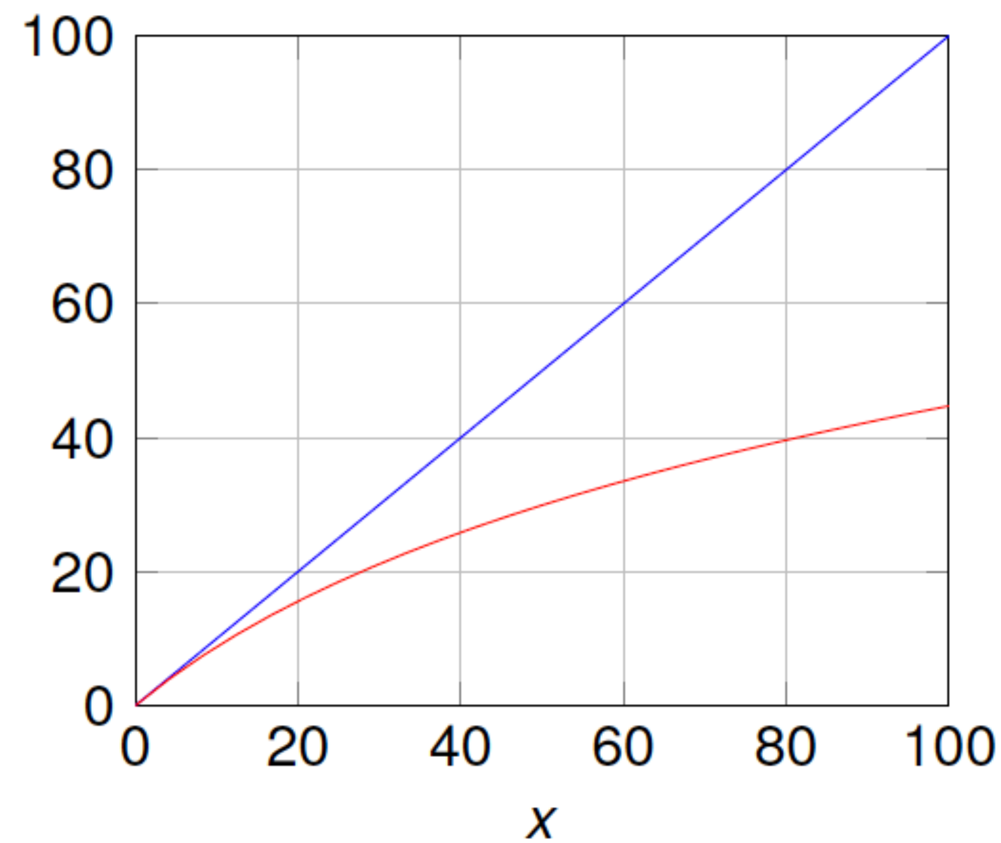


$f(0) = 0, f(1) = 1$

Approximation Algorithms

- What is the largest number that we can store in 8 bits?

$f(x) = \log(1 + x/30)/\log(1 + 1/30)$



$f(0) = 0, f(1) = 1$

Approximation Algorithms

MORRIS APPROXIMATE COUNTING ALGORITHM

```
Init counter c ← 0
for every event in the stream
        do rand = random number between 0 and 1
            if rand < p
              then c ← c + 1
```

- What is the largest number that we can store in 8 bits?

Approximation Algorithms

101100011110101 $\boxed{0111010}$

Sliding Window

We can maintain simple statistics over sliding windows, using $O(\frac{1}{\epsilon}\log^2 N)$ space, where

- N is the length of the sliding window
- ϵ is the accuracy parameter

M. Datar, A. Gionis, P. Indyk, and R. Motwani.
Maintaining stream statistics over sliding windows. 2002

WHAT IS **MOA**?

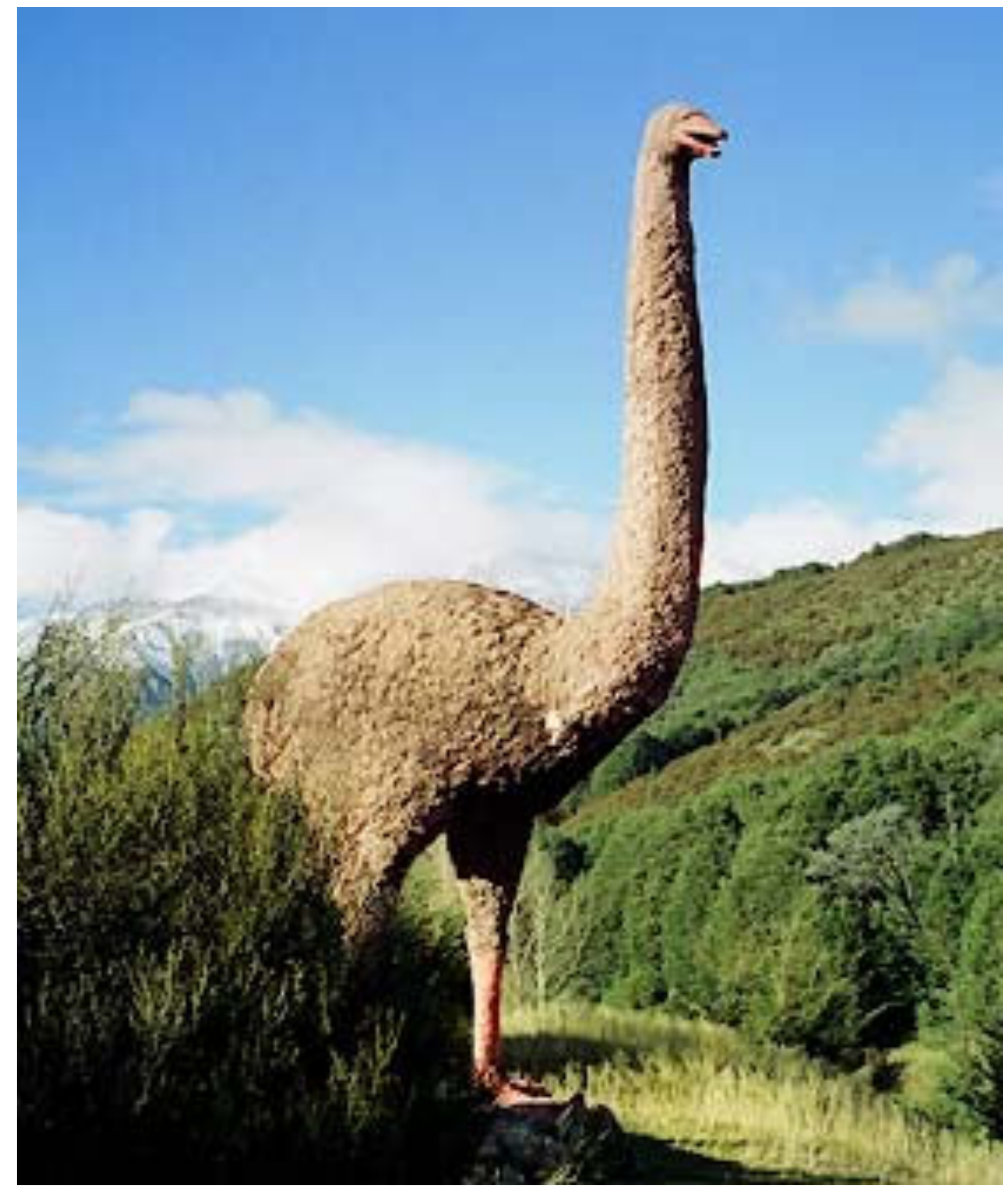
MOA

- {M}assive {O}nline {A}nalysis is a framework for online learning from data streams.
- It is closely related to WEKA
- It includes a collection of offline and online as well as tools for evaluation:
  - classification, regression
  - clustering, frequent pattern mining
- Easy to extend, design and run experiments

WEKA: the bird

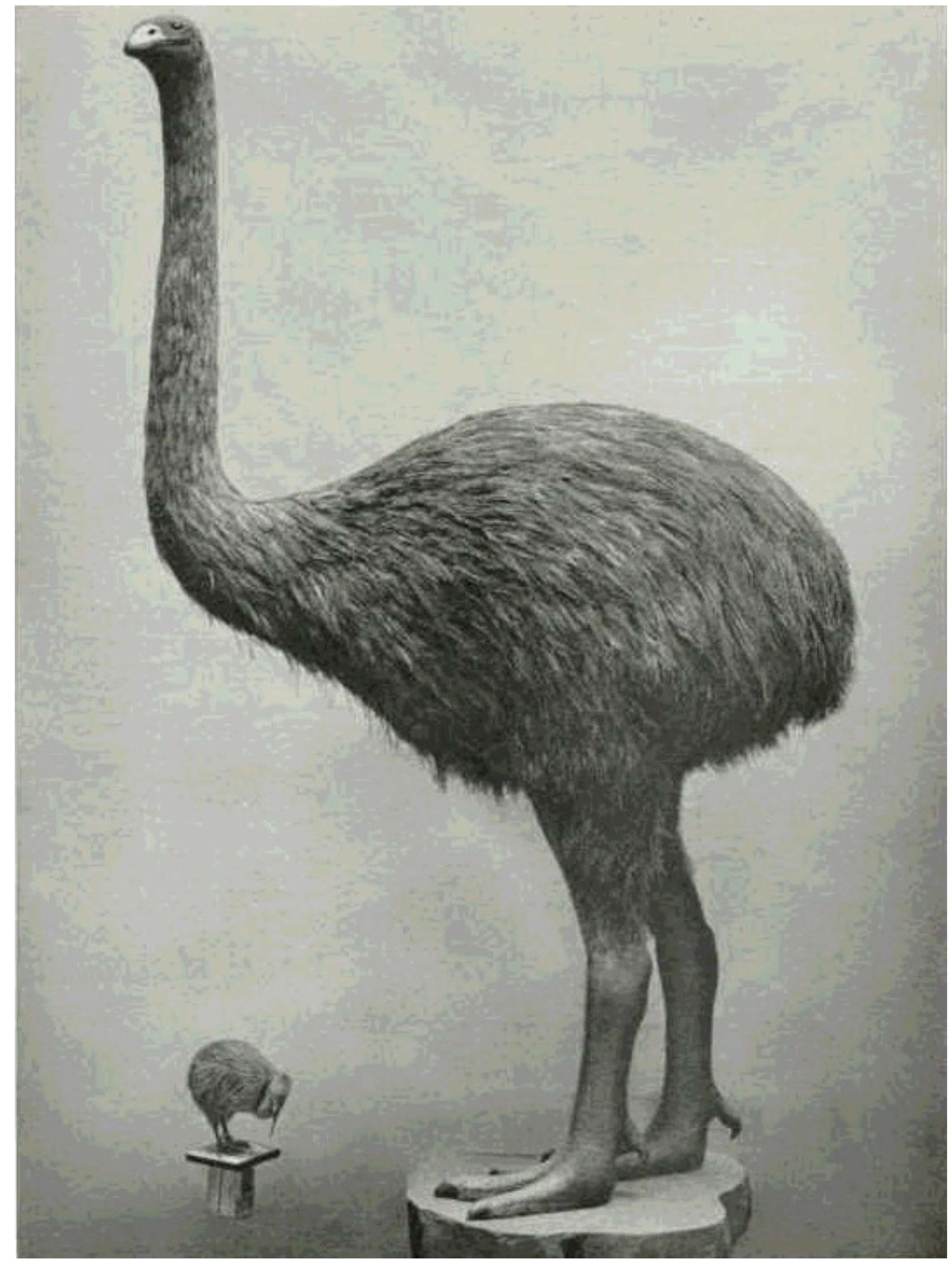
MOA: the bird

The Moa (another native NZ bird) is not only flightless, like the Weka, but also extinct.



MOA: the bird

The Moa (another native NZ bird) is not only flightless, like the Weka, but also extinct.

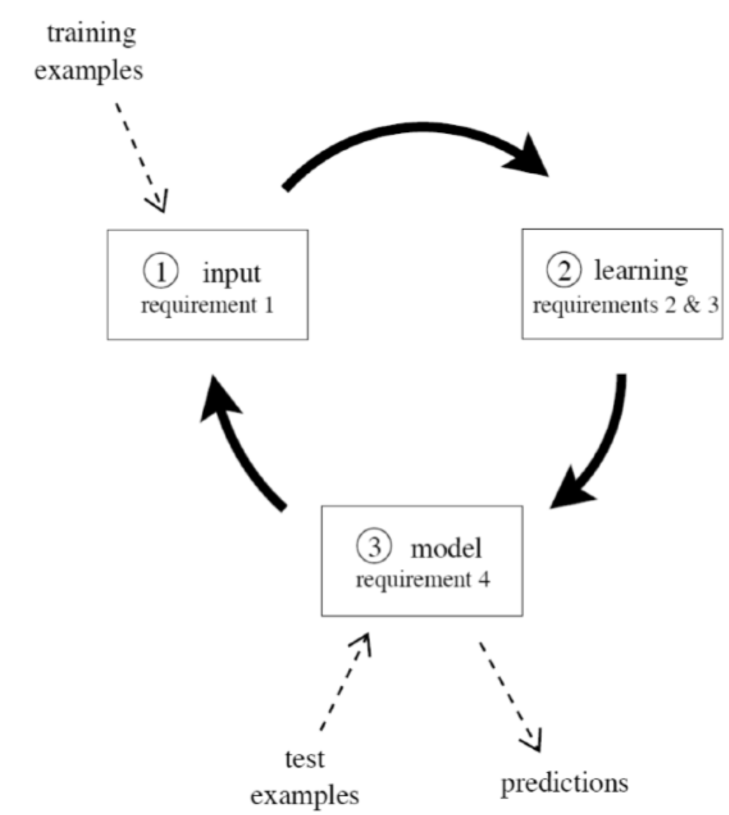


MOA: the bird

The Moa (another native NZ bird) is not only flightless, like the Weka, but also extinct.

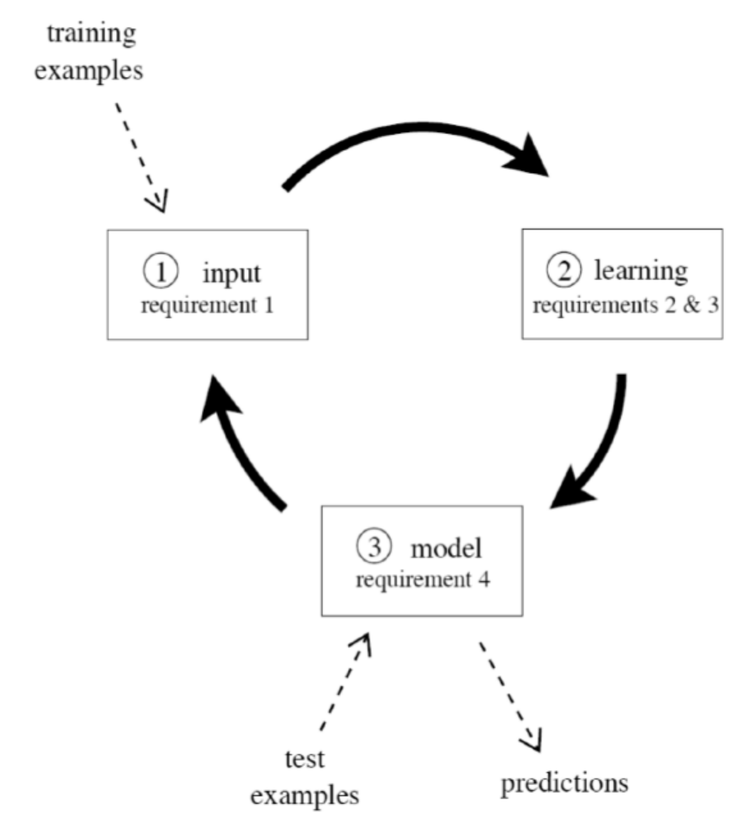
STREAM SETTING

- Process an example at a time,and inspect it only once (at most)
- Use a limited amount of memory
- Work in a limited amount of time
- Be ready to predict at any point



STREAM EVALUATION

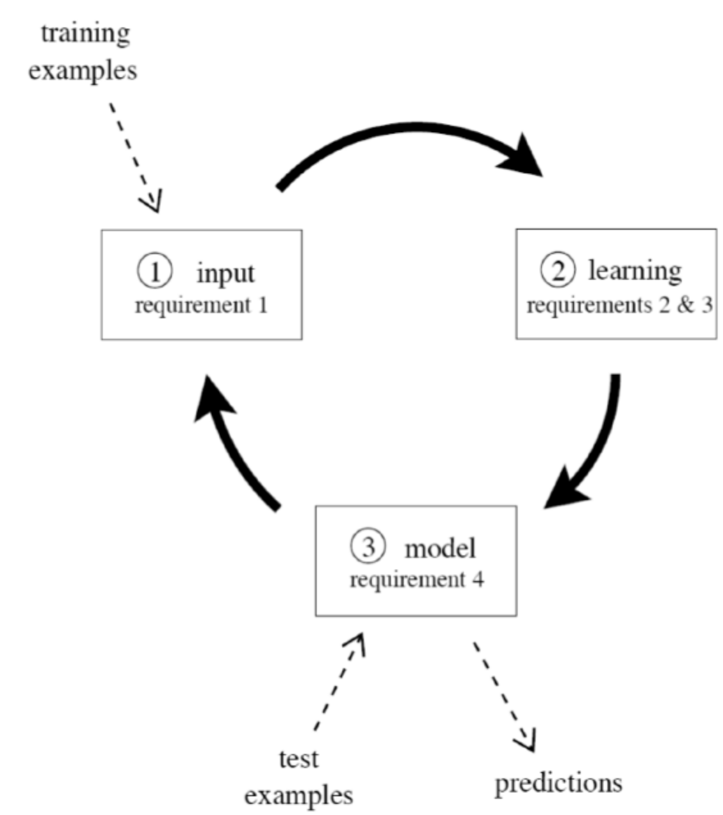
- Holdout Evaluation
- Interleaved Test-Then-Train or Prequential



STREAM EVALUATION

Holdout an independent test set

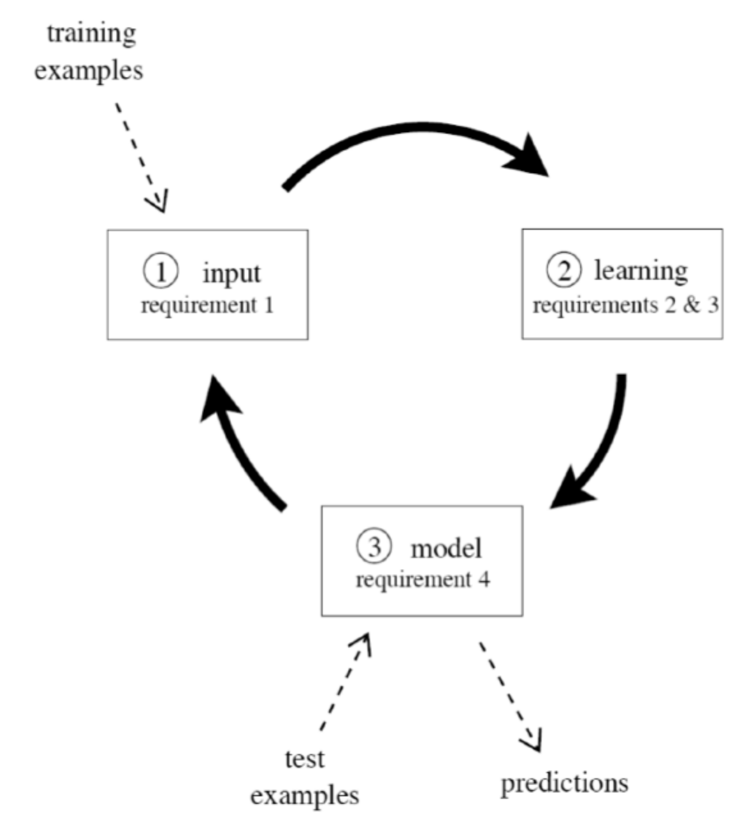
- Apply the current decision model to the test set, at regular time intervals
- The loss estimated in the holdout is an unbiased estimator



STREAM EVALUATION

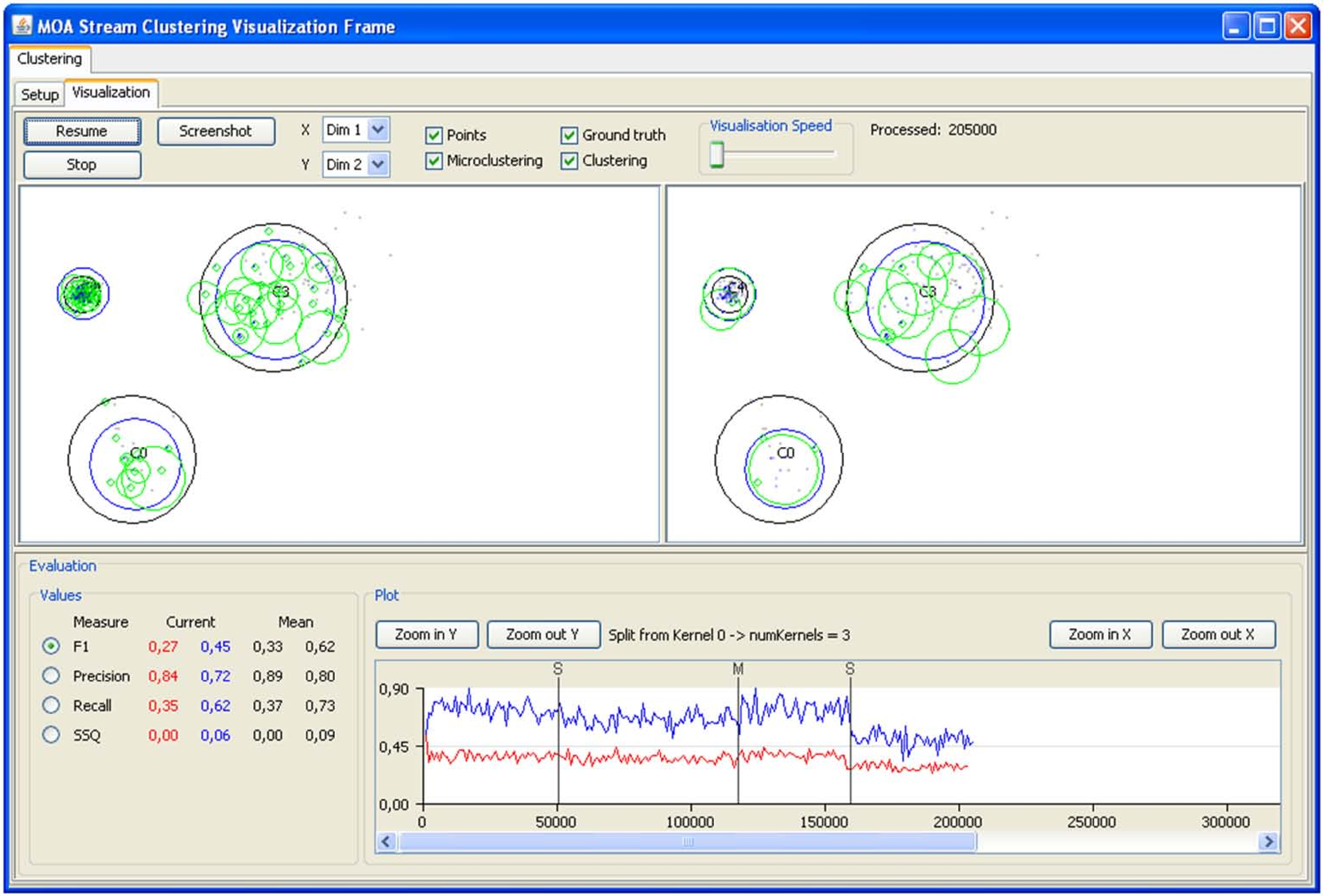
Prequential Evaluation

- The error of a model is computed from the sequence of examples.
- For each example in the stream, the actual model makes a prediction based only on the example attribute-values.



$$S = \sum_{i=1}^{n} L(y_i, \hat{y}_i).$$

GUI


MOA Stream Clustering Visualization Frame
Clustering
Setup
Visualization
Resume
Stop
Screenshot
X Dim 1
Y Dim 2
Points
Microclustering
Ground truth
Clustering
Visualisation Speed
Processed: 205000
C3
C0
C4
Evaluation
Values
Measure
Current
Mean
F1 0,27 0,45 0,33 0,62
Precision 0,84 0,72 0,89 0,80
Recall 0,35 0,62 0,37 0,73
SSQ 0,00 0,06 0,00 0,09
Plot
Zoom in Y
Zoom out Y
Split from Kernel 0 -> numKernels = 3
Zoom in X
Zoom out X
S
M
S
0,90
0,45
0,00
0
50000
100000
150000
200000
250000
300000

COMMAND LINE

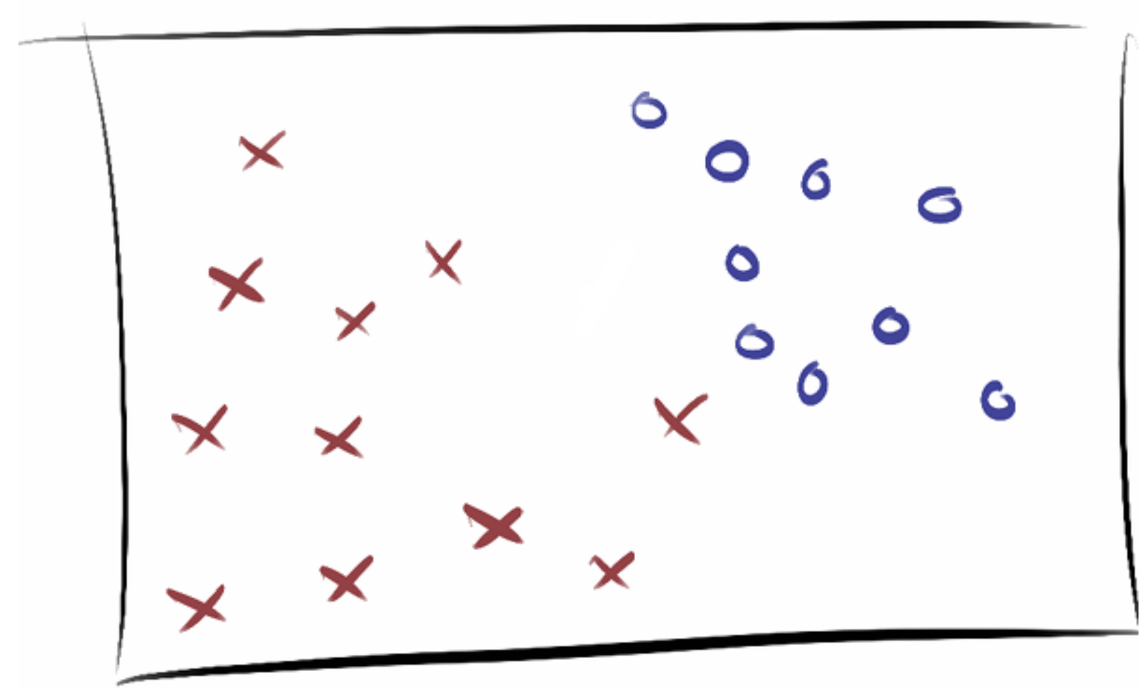
- `java -cp .:moa.jar:weka.jar -javaagent:sizeofag.jar moa.DoTask "EvaluatePeriodicHeldOutTest -l DecisionStump -s generators.WaveformGenerator -n 100000 -i 100000000 -f 1000000" > dsresult.csv`

- This command creates a comma separated values file:
  - training the DecisionStump classifier on the WaveformGenerator data,
  - using the first 100 thousand examples for testing,
  - training on a total of 100 million examples,
  - and testing every one million examples

Classification

Definition

Given a set of training examples belonging to n_C different classes, a classifier algorithm builds a model that predicts for every unlabeled instance x the class C to which it belongs



Examples

- Email spam filter
- Twitter sentiment analyzer

Photo: Stephen Merity http://smerity.com

Naïve Bayes

- Based on Bayes' theorem

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

- Probability of observing feature x_i given class C

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

- Prior class probability P(C)

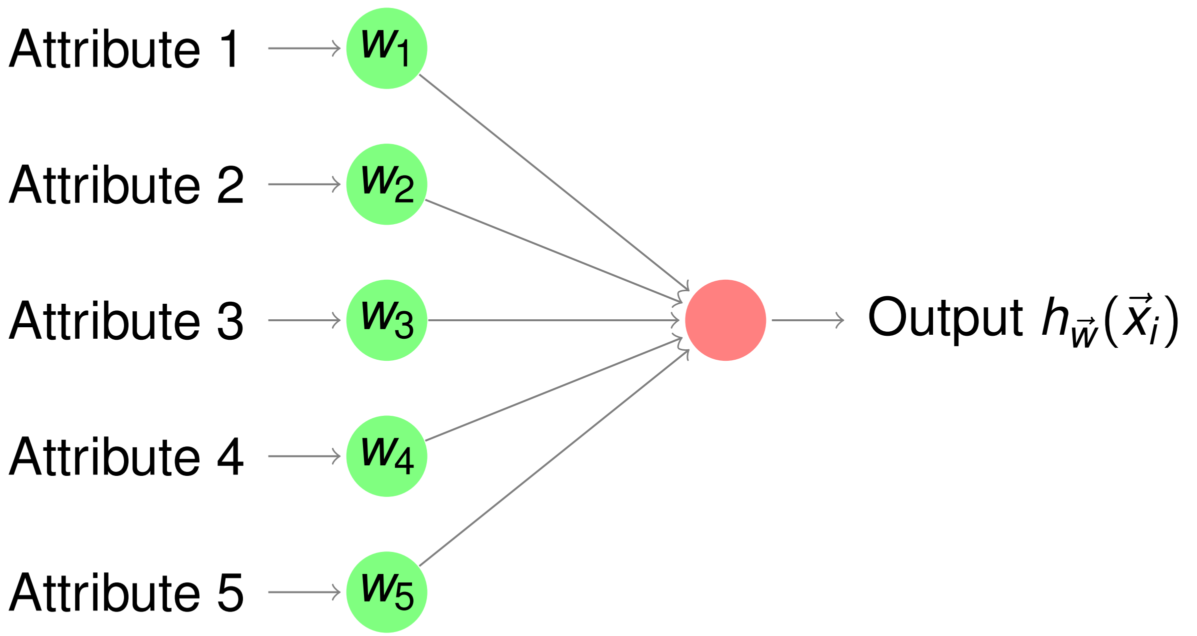
$$P(C|x) \propto \prod_{x_i \in x} P(x_i|C)P(C)$$

- Just counting!

$$C = \arg\max_{C} P(C|x)$$

Perceptron

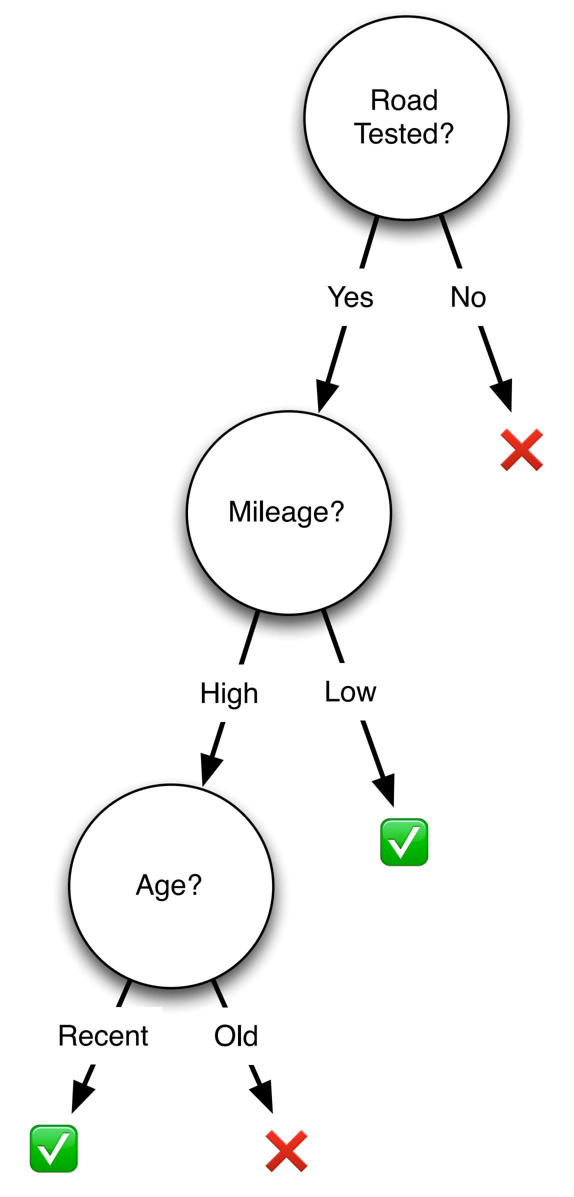
- Linear classifier
- Data stream: $\langle \vec{\mathbf{x}}_i, y_i \rangle$
- $\tilde{y}_i = h_{\vec{w}}(\vec{\mathbf{x}}_i) = \sigma(\vec{\mathbf{w}}_i^T \vec{\mathbf{x}}_i)$
- $\sigma(x) = 1/(1+e^{-x})$ $\sigma' = \sigma(x)(1-\sigma(x))$
- Minimize MSE $J(\vec{\mathbf{w}}) = ½\sum(y_i - \tilde{y}_i)^2$
- SGD $\vec{\mathbf{w}}_{i+1} = \vec{\mathbf{w}}_i - \eta \nabla J \, \vec{\mathbf{x}}_i$
  - $\nabla J = -(y_i - \tilde{y}_i)\tilde{y}_i(1-\tilde{y}_i)$
  - $\vec{\mathbf{w}}_{i+1} = \vec{\mathbf{w}}_i + \eta(y_i - \tilde{y}_i)\tilde{y}_i(1-\tilde{y}_i)\vec{\mathbf{x}}_i$



Decision Tree

- Each node tests a features
- Each branch represents a value
- Each leaf assigns a class
- Greedy recursive induction
  - Sort all examples through tree
  - x_i = most discriminative attribute
  - New node for x_i, new branch for each value, leaf assigns majority class
  - Stop if no error | limit on #instances

Car deal?



P. Domingos and G. Hulten, "Mining High-Speed Data Streams," KDD '00

HOEFFDING TREE

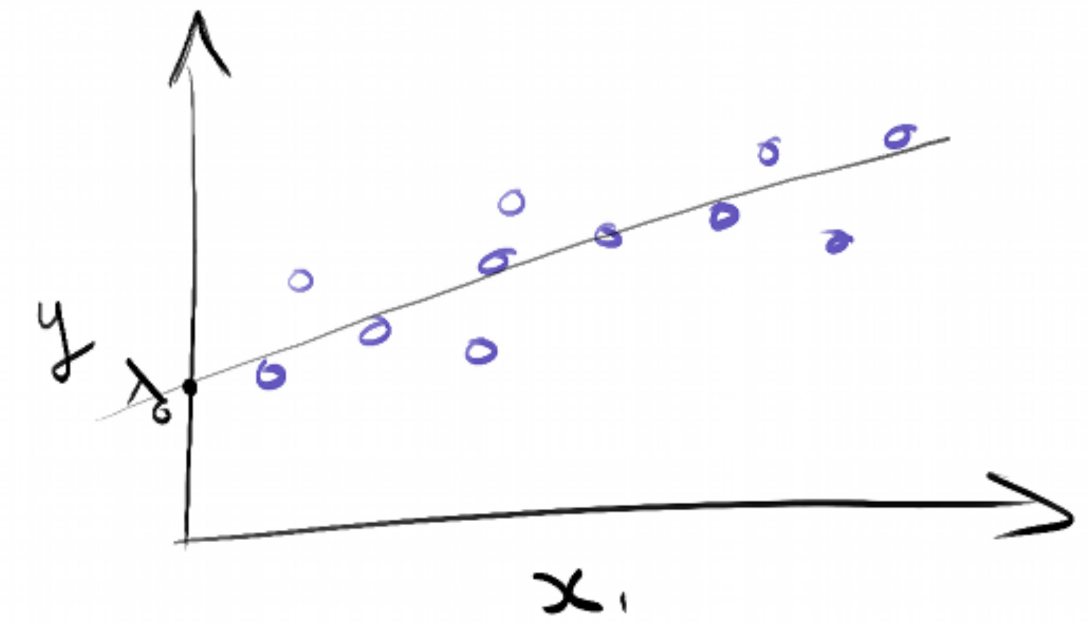
- Sample of stream enough for near optimal decision
- Estimate merit of alternatives from prefix of stream
- Choose sample size based on statistical principles
- When to expand a leaf?
  - Let x_1 be the most informative attribute, x_2 the second most informative one
  - Hoeffding bound: split if $G(x_1) - G(x_2) > \boldsymbol{\varepsilon} = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}$

Regression

Definition

Given a set of training examples with a numeric label, a regression algorithm builds a model that predicts for every unlabeled instance x the value with high accuracy

$$y=f(x)$$



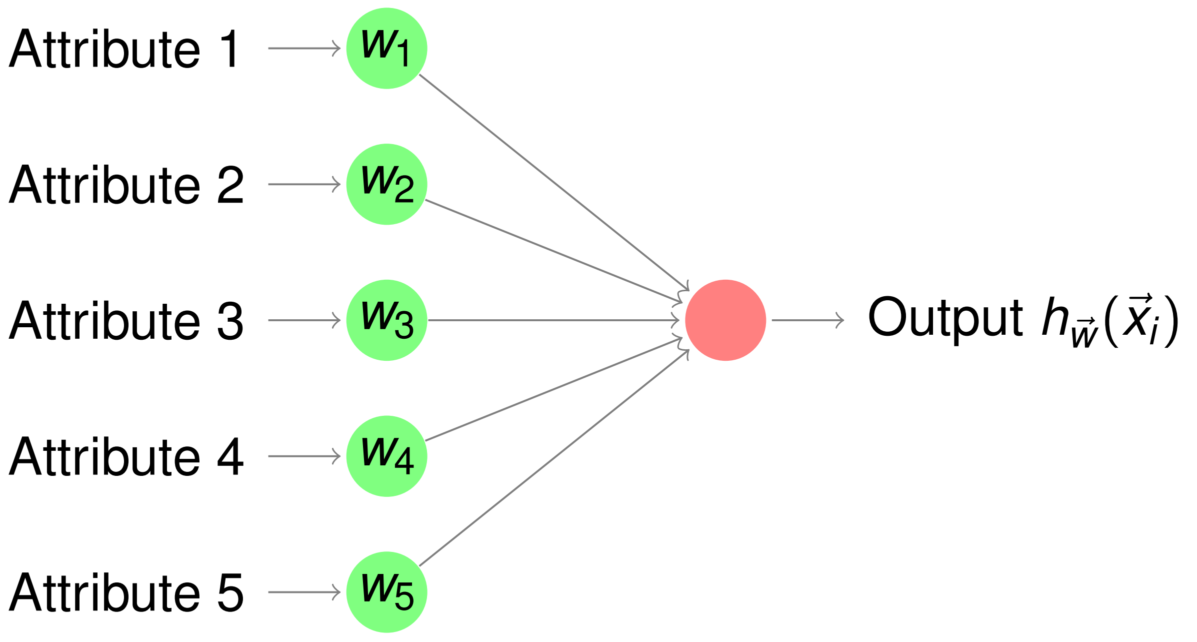
Examples

- Stock price
- Airplane delay

Photo: Stephen Merity http://smerity.com

Perceptron

- Linear regressor
- Data stream: $\langle \vec{\mathbf{x}}_i, y_i \rangle$
- $\tilde{y}_i = h_{\vec{w}}(\vec{\mathbf{x}}_i) = \vec{\mathbf{w}}^T \vec{\mathbf{x}}_i$
- Minimize MSE $J(\vec{\mathbf{w}}) = \frac{1}{2}\sum(y_i - \tilde{y}_i)^2$
- SGD $\vec{\mathbf{w}}' = \vec{\mathbf{w}} - \eta \nabla J \, \vec{\mathbf{x}}_i$
  - $\nabla J = -(y_i - \tilde{y}_i)$
  - $\vec{\mathbf{w}}' = \vec{\mathbf{w}} + \eta(y_i - \tilde{y}_i)\vec{\mathbf{x}}_i$



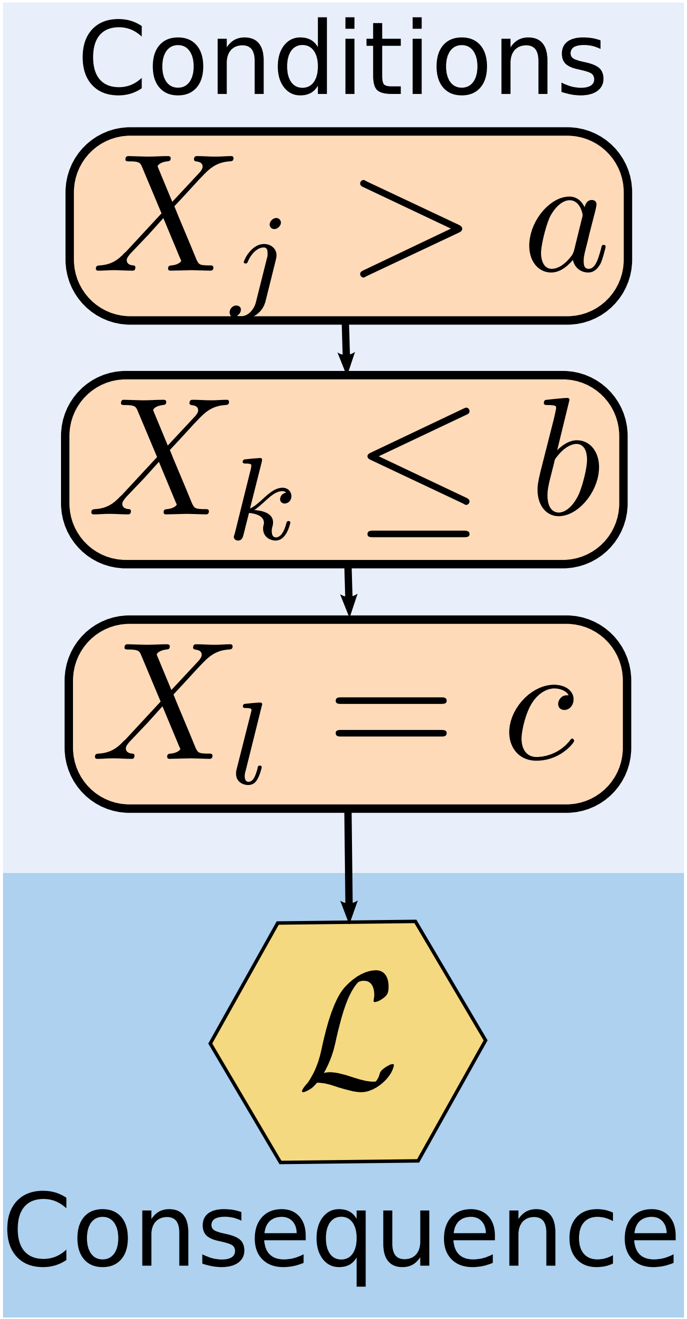
Regression Tree

- Same structure as decision tree
- Predict = average target value or linear model at leaf (vs majority)
- Gain = reduction in standard deviation (vs entropy)

$$\sigma = \sqrt{\sum (\tilde{y}_i - y_i)^2 / (N-1)}$$

Rules

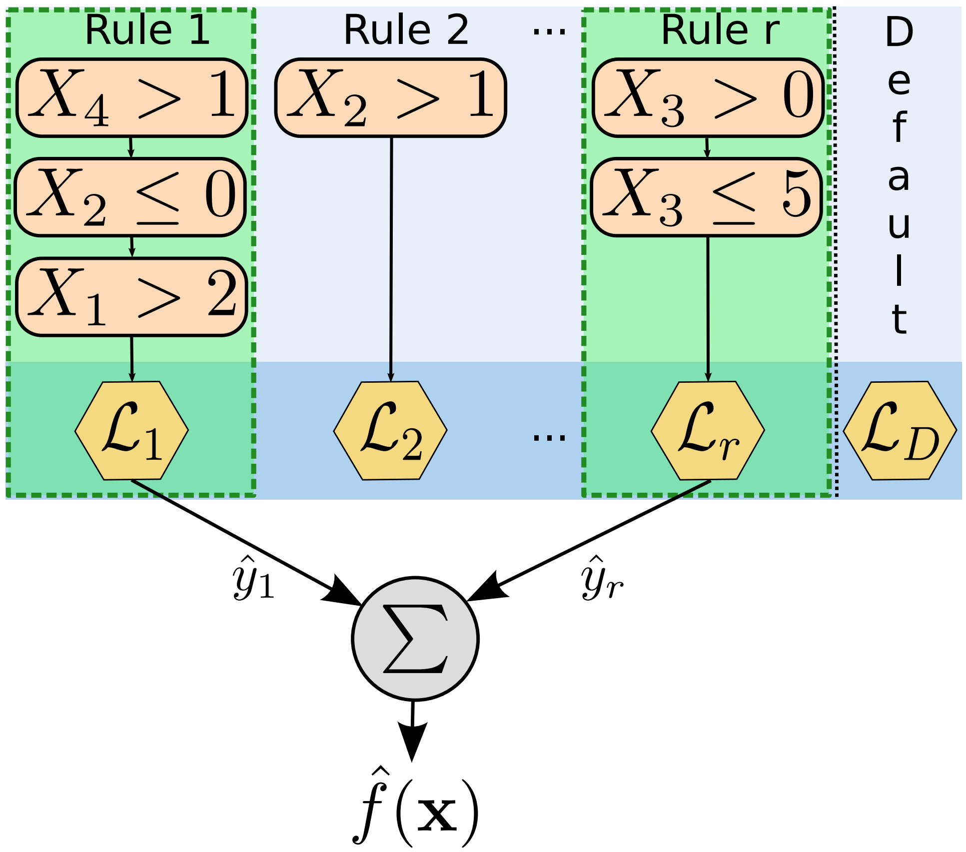
- Problem: very large decision trees have context that is complex and hard to understand
- Rules: self-contained, modular, easier to interpret, no need to cover universe
- $\mathcal{L}$ keeps sufficient statistics to:
    - make predictions
    - expand the rule
    - detect changes and anomalies



Adaptive Model Rules

E. Almeida, C. Ferreira, J. Gama. "Adaptive Model Rules from Data Streams." ECML-PKDD '13

- Ruleset: ensemble of rules
- Rule prediction: mean, linear model
- Ruleset prediction
    - Weighted avg. of predictions of rules covering instance x
    - Weights inversely proportional to error
    - Default rule covers uncovered instances



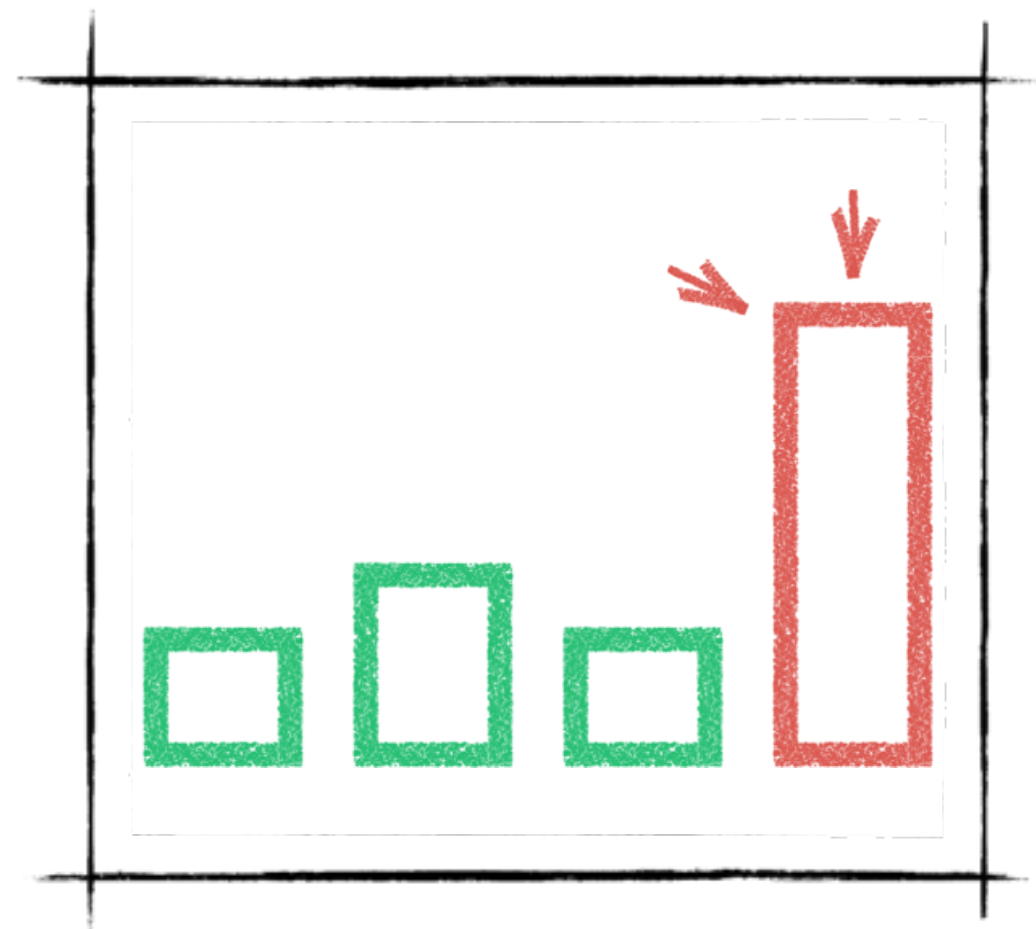
E.g: $\mathbf{x} = [4, -1, 1, 2]$

$$\hat{f}(\mathbf{x}) = \sum_{R_l \in S(\mathbf{x}_i)} \theta_l \hat{y}_l,$$

Concept Drift

Definition

Given an input sequence $\langle x_1, x_2, \ldots, x_t \rangle$, output at instant *t* an alarm signal if there is a distribution change, and a prediction $\hat{x}_{t+1}$ minimizing the error $|\hat{x}_{t+1} - x_{t+1}|$



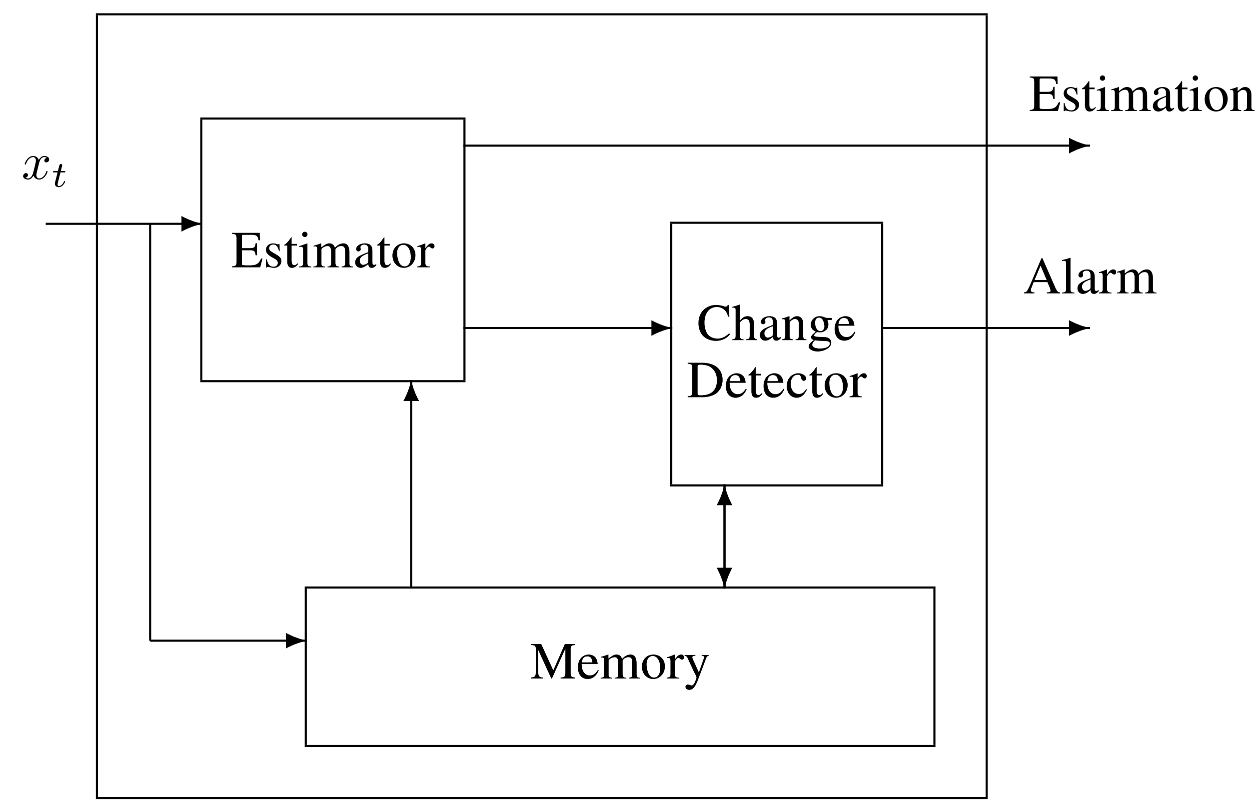
Outputs

- Alarm indicating change
- Estimate of parameter

Photo: http://www.logsearch.io

Application

- Change detection on evaluation of model
- Training error should decrease with more examples
    - Change in distribution of training error
    - Input = stream of real/binary numbers
- Trade-off between detecting true changes and avoiding false alarms



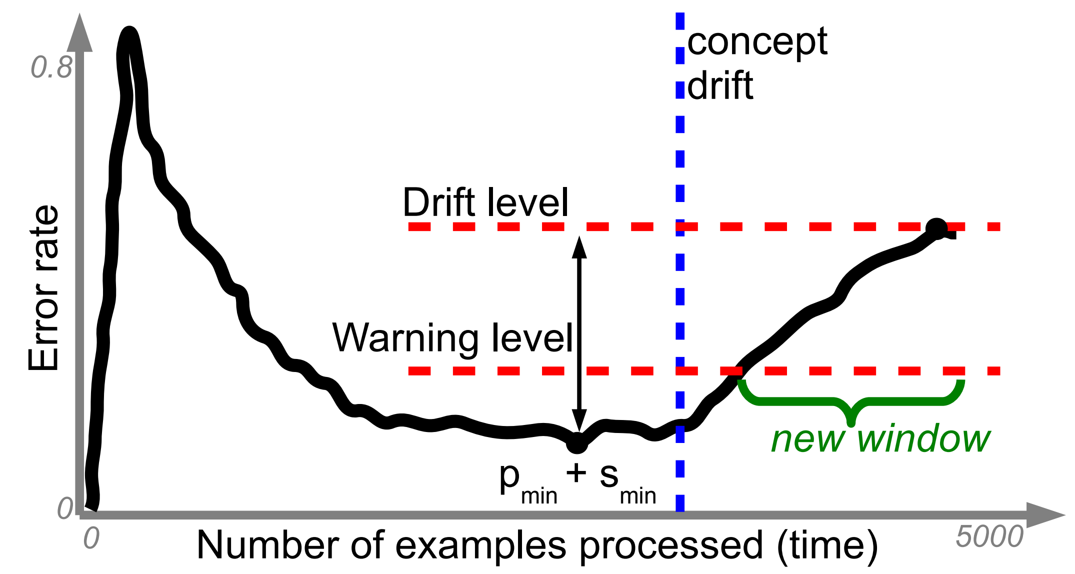
Cumulative Sum

- Alarm when mean of input data differs from zero
- Memoryless heuristic (no statistical guarantee)
- Parameters: threshold h, drift speed v
- $g_0 = 0, \quad g_t = \max(0, g_{t-1} + \varepsilon_t - v)$
- if $g_t > h$ then alarm; $g_t = 0$

Statistical Process Control

J Gama, P. Medas, G. Castillo, P. Rodrigues: “Learning with Drift Detection”. SBIA '04

- Monitor error in sliding window
- Null hypothesis:
  no change between windows
- If error > warning level
  learn in parallel new model
  on the current window
- if error > drift level
  substitute new model for old



Concept-adapting VFDT

G. Hulten, L. Spencer, P. Domingos: "Mining Time-Changing Data Streams". KDD '01

- Model consistent with sliding window on stream
- Keep sufficient statistics also at internal nodes
    - Recheck periodically if splits pass Hoeffding test
    - If test fails, grow alternate subtree and swap-in when accuracy of alternate is better
- Processing updates O(1) time, +O(W) memory
    - Increase counters for incoming instance, decrease counters for instance going out window

A. Bifet, R. Gavaldà: "Adaptive Parameter-free Learning from Evolving Data Streams" IDA (2009)

Hoeffding Adaptive Tree

- Replace frequency counters by estimators
  - No need for window of instances
  - Sufficient statistics kept by estimators separately
- Parameter-free change detector + estimator with theoretical guarantees for subtree swap (ADWIN)
  - Keeps sliding window consistent with "no-change hypothesis"

A. Bifet, R. Gavaldà: "Learning from Time-Changing Data with Adaptive Windowing". SDM '07

ADWIN

ADWIN

An adaptive sliding window whose size is recomputed online according to the rate of change observed.

Problem

Given an input sequence $x_1, x_2, \ldots, x_t, \ldots$ we want to output

- a prediction $\widehat{x}_{t+1}$ minimizing prediction error:

$$|\widehat{x}_{t+1} - x_{t+1}|$$

- an alert if change is detected

ADWIN

Optimal Change Detector and Predictor

- High accuracy
- Fast detection of change
- Low false positives and false negatives ratios
- Low computational cost: minimum space and time needed

`ADWIN`

- Theoretical guarantees
- No parameters needed

ADWIN

Theorem

At every time step we have:

1. (False positive rate bound). *If μ_t remains constant within W, the probability that* ADWIN *shrinks the window at this step is at most δ.*
2. (False negative rate bound). *Suppose that for* some *partition of W in two parts $W_0 W_1$ (where W_1 contains the most recent items) we have $|\mu_{W_0} - \mu_{W_1}| > 2\epsilon_c$. Then with probability $1-\delta$* ADWIN *shrinks W to W_1, or shorter.*

ADWIN tunes itself to the data stream at hand, with no need for the user to hardwire or precompute parameters.

ADWIN

- Classification
  - Adaptive Naive Bayes (Bifet et al. 2007)
  - Decision Trees: Hoeffding Adaptive Trees (Bifet et al. 2009)
  - `ADWIN` Bagging (Bifet et al. 2009)
  - Leveraging Bagging (Bifet et al. 2010)
  - Stacking of Restricted Hoeffding Trees (Bifet et al. 2012)
  - Multilabel Classification (Read et al. 2012)
  - Adaptive kNN (Bifet et al. 2013)
  - Random Forests (Marron et al. 2014)
- Frequent Pattern Mining
  - Frequent Closed Tree Mining (Bifet et al. 2008)
  - Frequent Closed Graph Mining (Bifet et al. 2011)

Adaptive Random Forest

- Why Random Forests?
  - Off-the-shelf learner
  - Good learning performance Related publication

Adaptive random forests for evolving data stream classification.

Gomes, H M; Bifet, A; Read, J; Barddal, J P; Enembreck, F; Pfharinger, B; Holmes, G; Abdessalem, T.

Machine Learning, Springer, 2017.

- Based on the original Random Forest by Breiman

Adaptive Random Forest

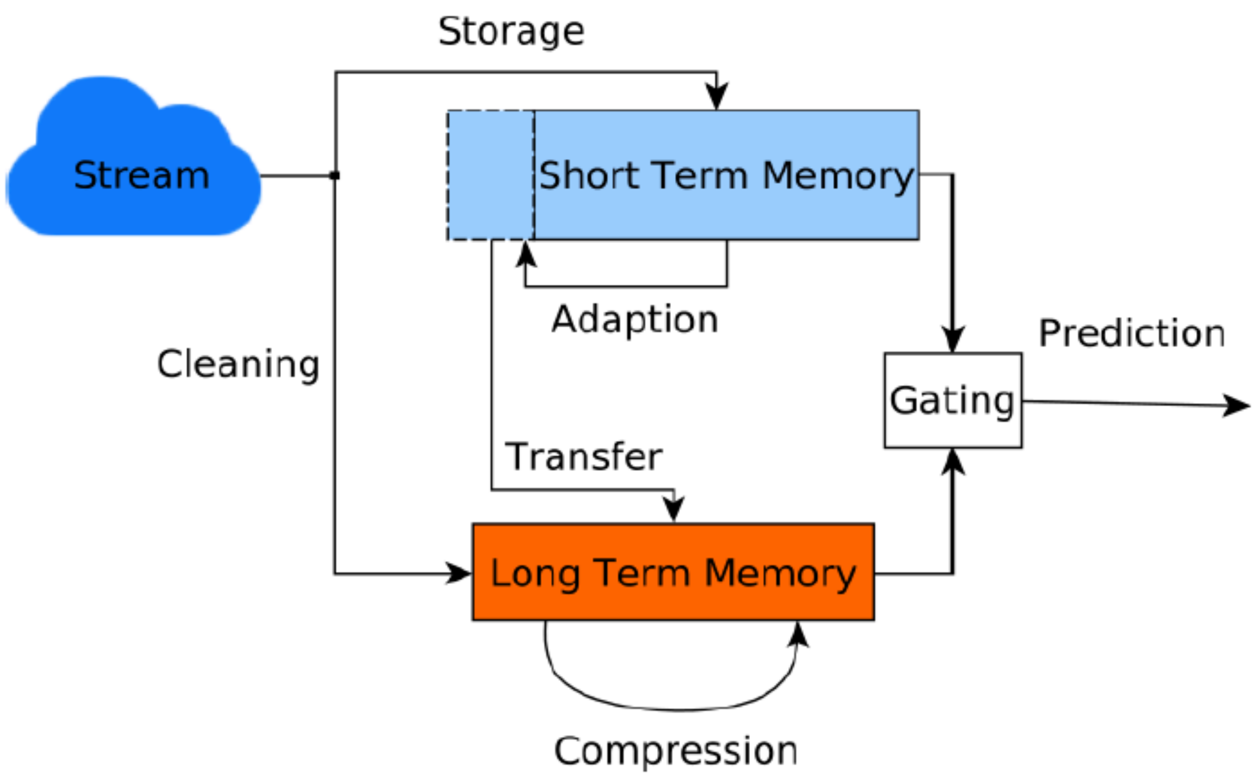
1. Simulates resampling through leveraging bagging
2. Randomly select subsets of features for splits
3. Uses Hoeffding Trees as the base learner
4. 1 drift and 1 warning detector per tree
5. Train trees in the background before adding them
6. Trees are completely independent (can train in parallel)

SAM-kNN

KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift.

Viktor Losing, Barbara Hammer, Heiko Wersing:

Best Paper Award
ICDM 2016: 291-300



Micro-Clusters

Tian Zhang, Raghu Ramakrishnan, Miron Livny: “BIRCH: An Efficient Data Clustering Method for Very Large Databases”. SIGMOD ’96

- AKA, Cluster Features CF
  Statistical summary structure

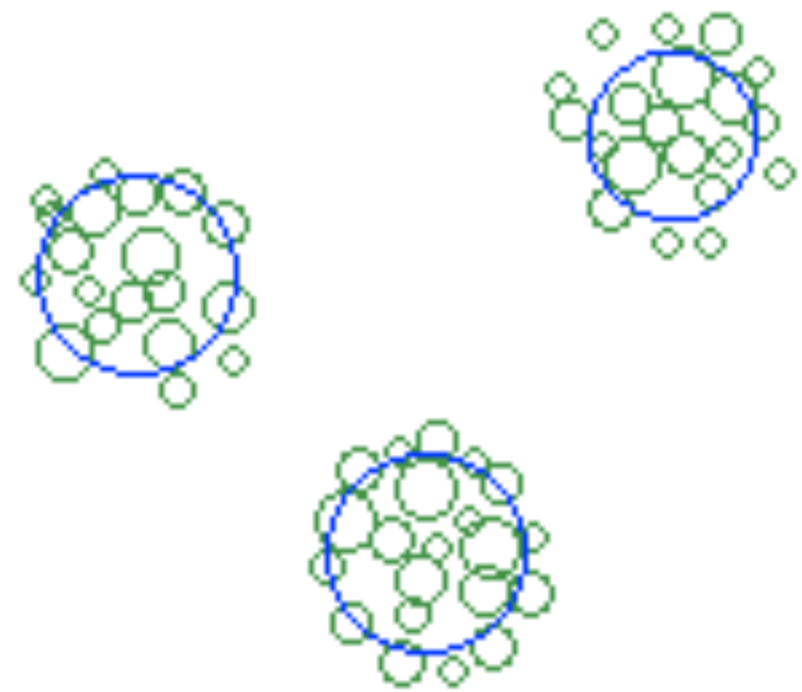
- Maintained in online phase,
  input for offline phase

- Data stream $\langle \vec{x}_i \rangle$, d dimensions

- Cluster feature vector
  N: number of points
  LS_j: sum of values (for dim. j)
  SS_j: sum of squared values (for dim. j)

- Easy to update, easy to merge

- **Constant space irrespective to the number of examples!**

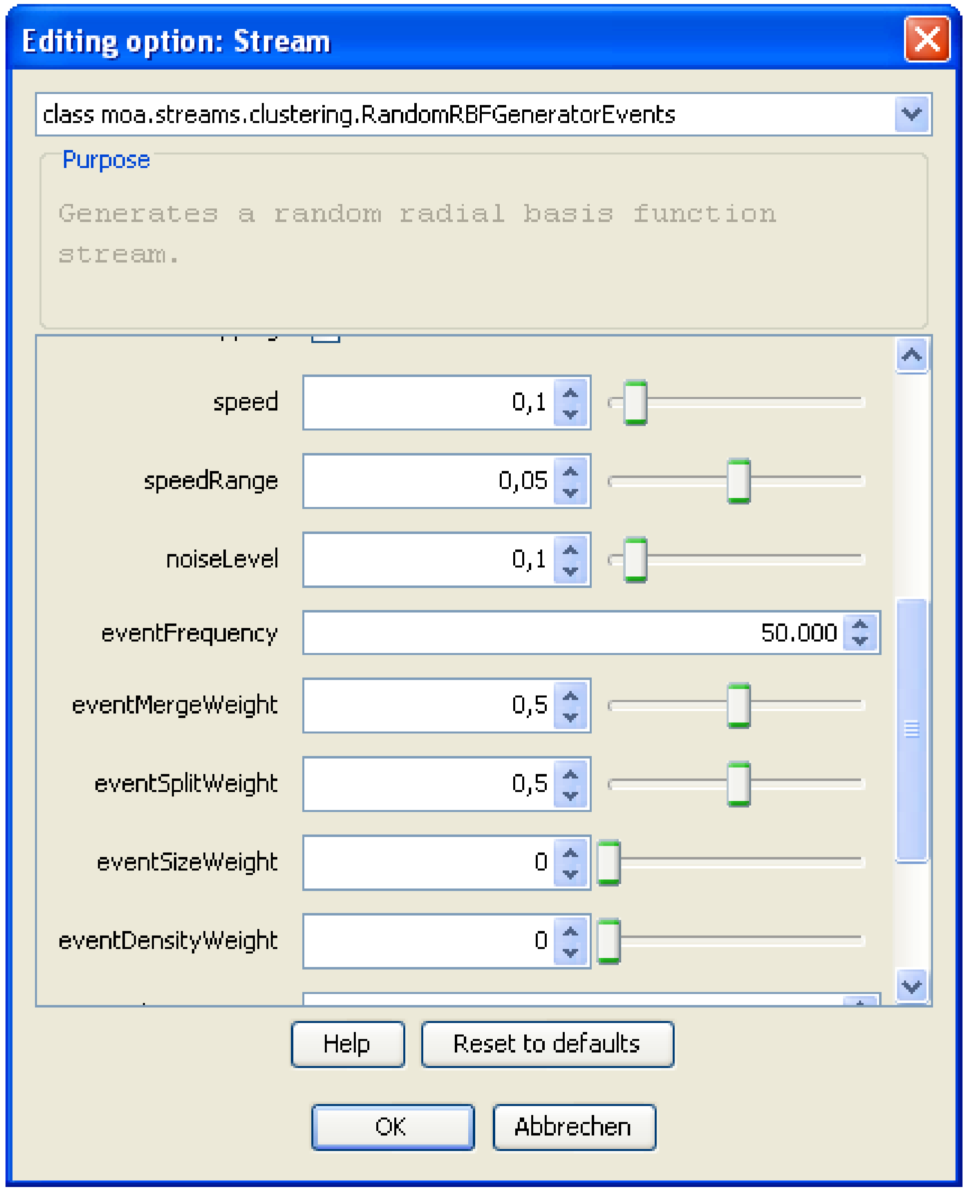


Properties:

- Centroid $= LS/N$
- Radius $= \sqrt{SS/N - (LS/N)^2}$
- Diameter $= \sqrt{\frac{2 \times N * SS - 2 \times LS^2}{N \times (N-1)}}$

MOA Algorithms

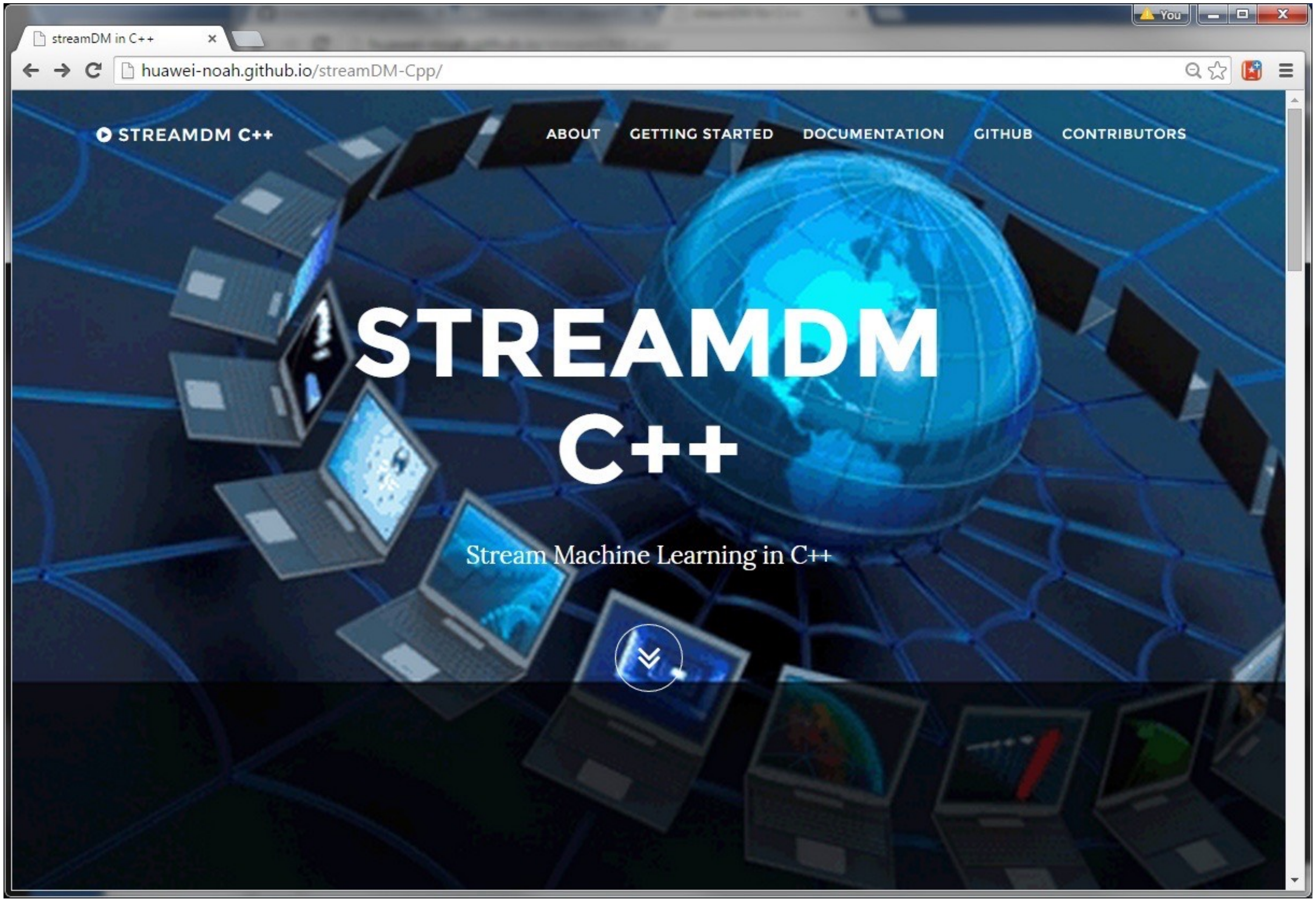
- Multi-label/ Multi-target
- Outlier Detection
- Concept Drift Detection
- Active Learning
- Frequent Itemset Mining
- Frequent Graph Mining
- Recommendation Systems



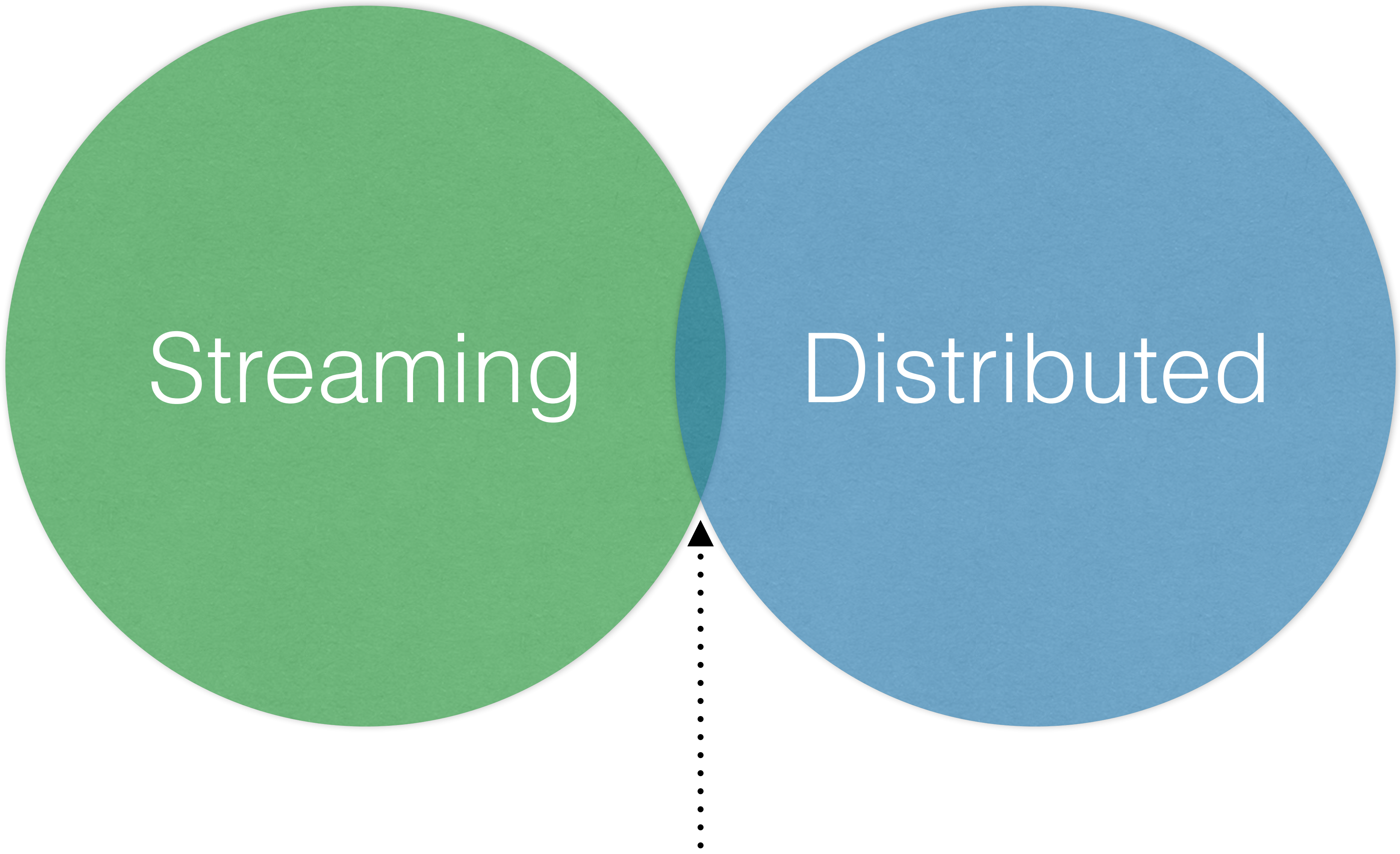
What’s next?

http://huawei-noah.github.io/streamDM-Cpp/

streamDM C++



Vision

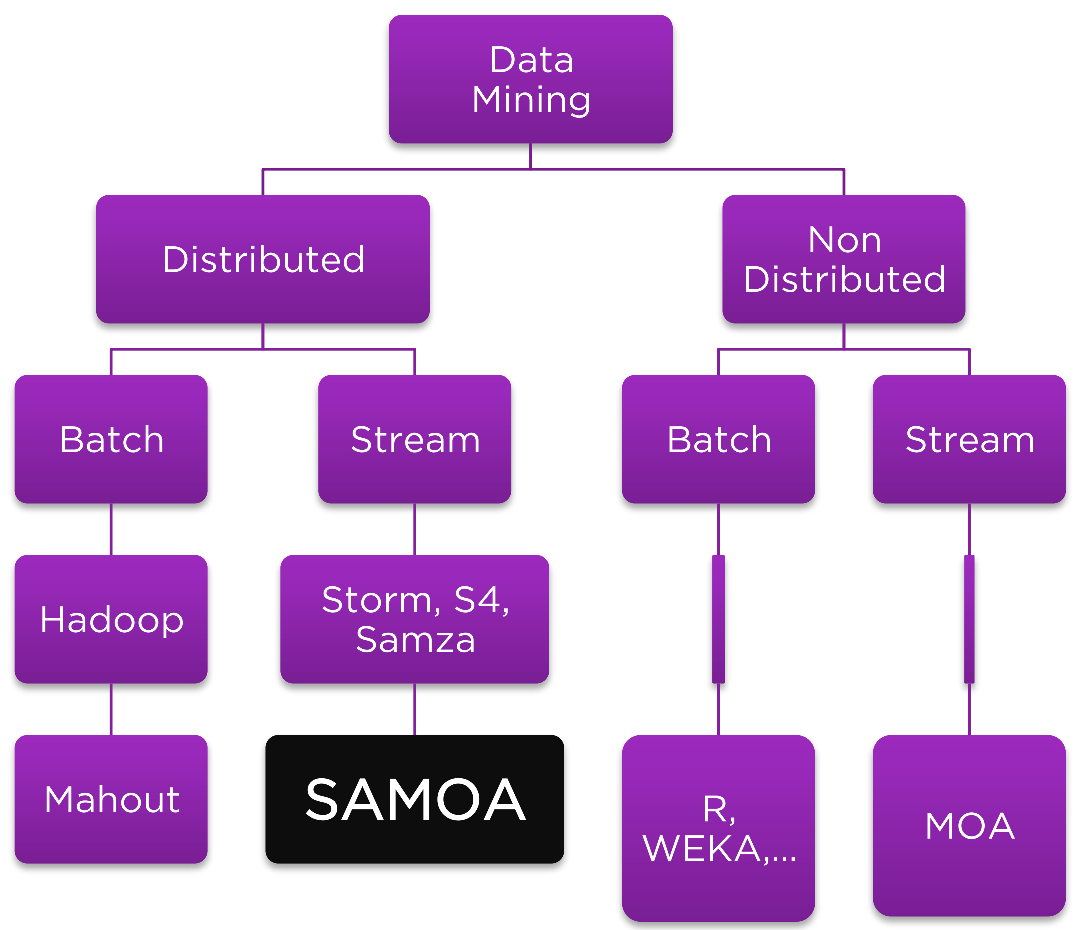


IoT Big Data Stream Mining

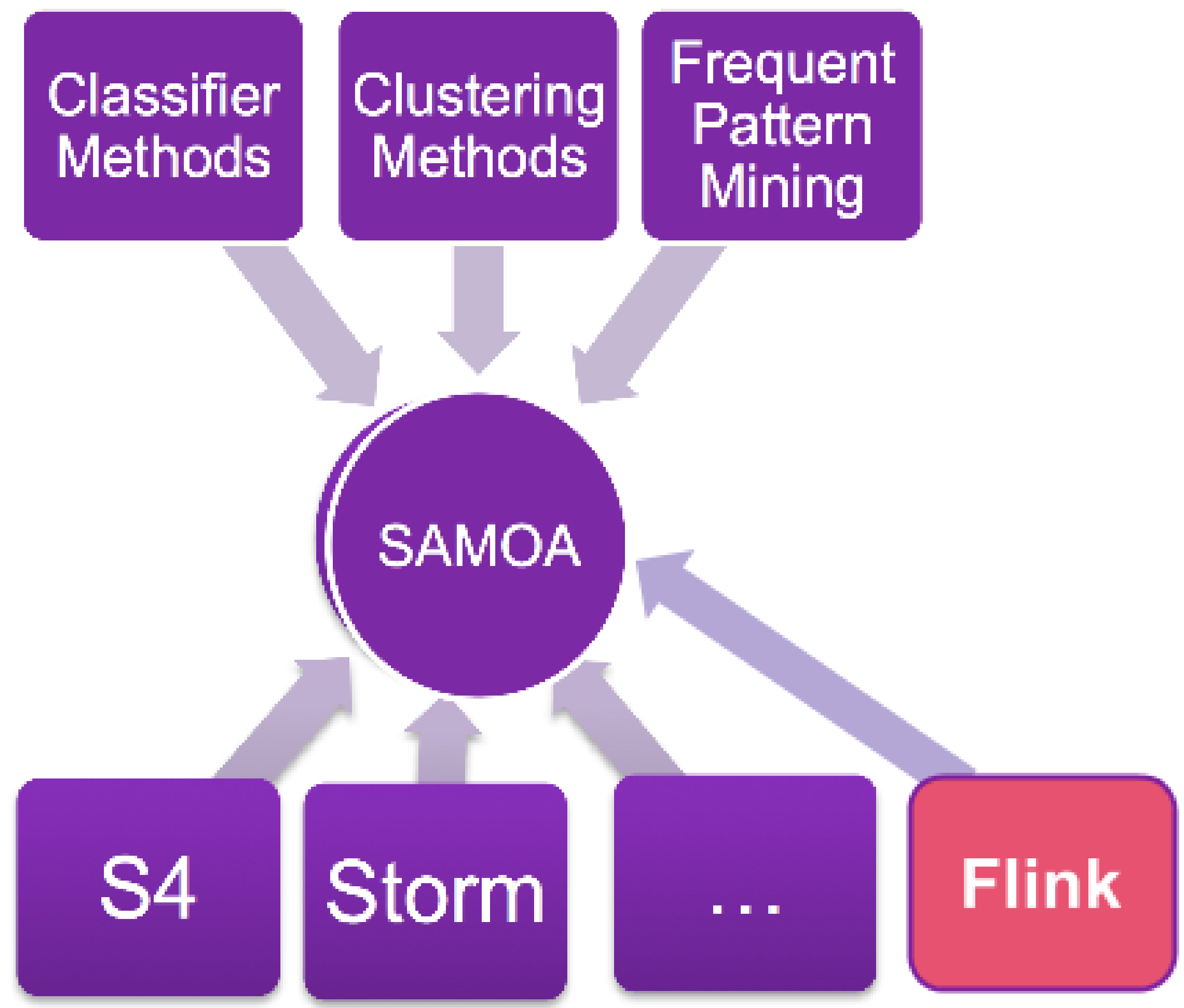
http://samoa-project.net

APACHE SAMOA

G. De Francisci Morales, A. Bifet: “SAMOA: Scalable Advanced Massive Online Analysis”. JMLR (2014)

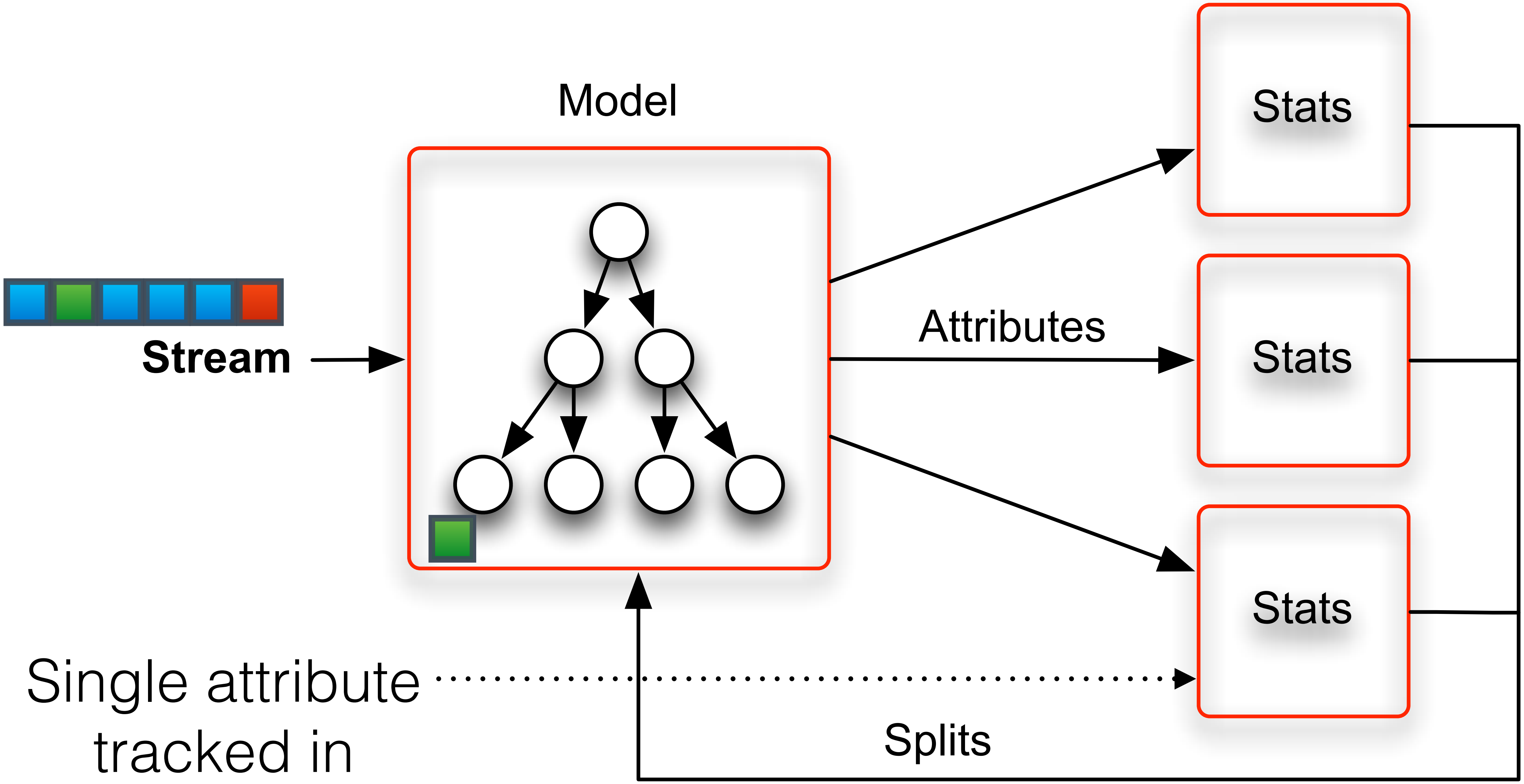


SAMOA ARCHITECTURE



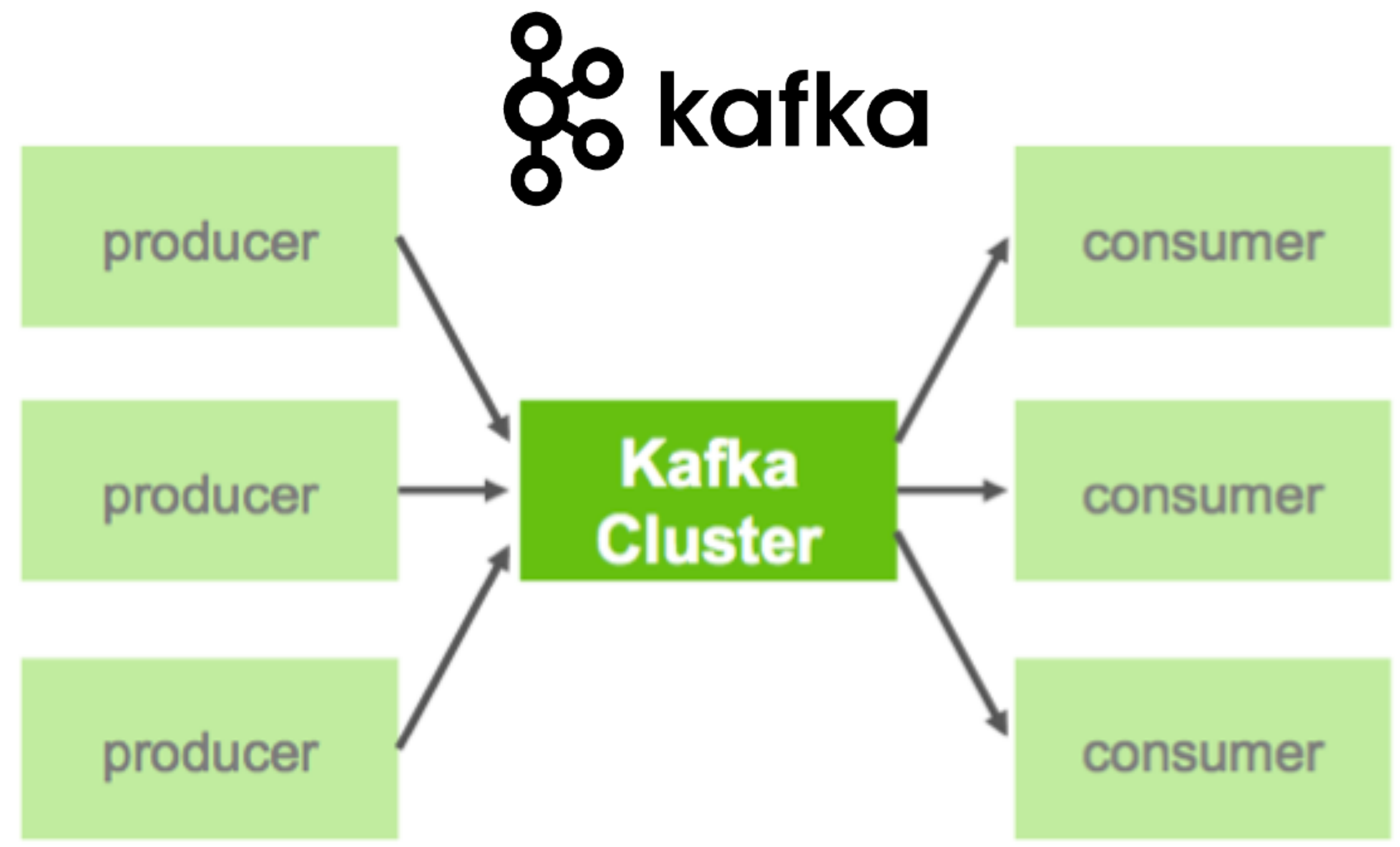
Vertical Partitioning

N. Kourtellis, G. De Francisci Morales, A. Bifet, A. Murdopo: "VHT: Vertical Hoeffding Tree", 2016 Big Data Conference 2016



Single attribute tracked in single node

Kappa Architecture

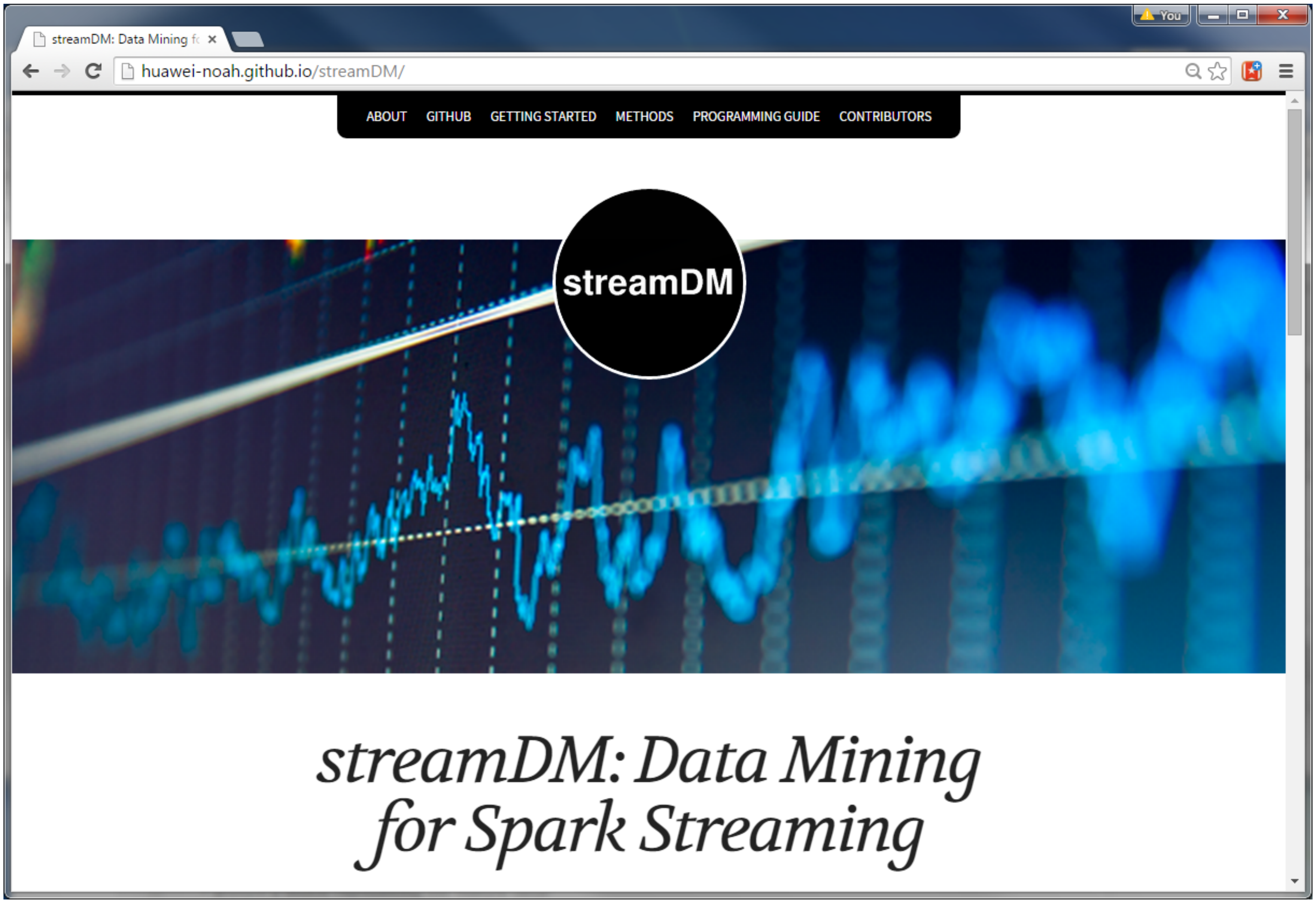


- Apache Kafka is a fast, scalable, durable, and fault-tolerant publish-subscribe messaging system.

SUPPORTING ORGANISATIONS

http://huawei-noah/github.io/streamDM

StreamDM



Summary

- IoT Streaming useful for finding approximate solutions with reasonable amount of time & limited resources
- MOA: Massive Online Analytics
  - Available and open-source
    - http://moa.cms.waikato.ac.nz/
- SAMOA: A Platform for Mining Big Data Streams
  - Available and open-source (incubating @ASF)
    - http://samoa.incubator.apache.org

Open Challenges

- Times Series + Stream Mining
- Structured output
- Millions of classes
- Ease of use
- Applications: Predictive Maintenance, AI for IoT

Thanks!

@abifet