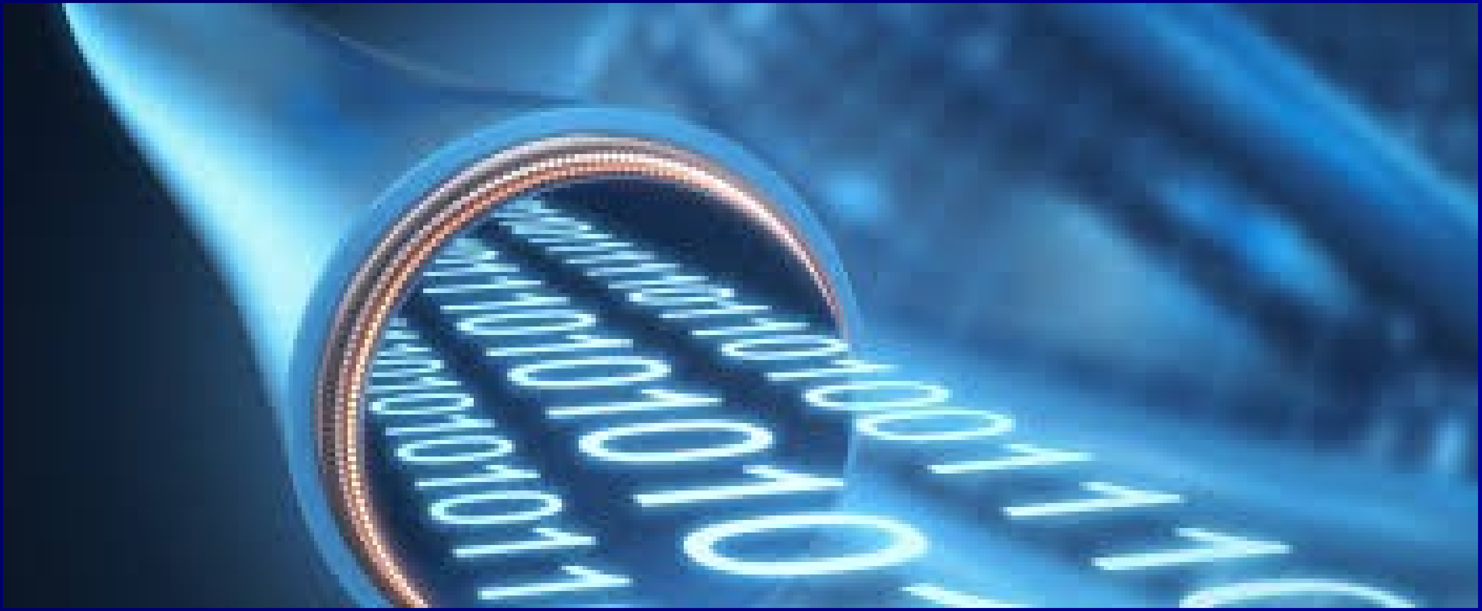
DATA STREAM MINING

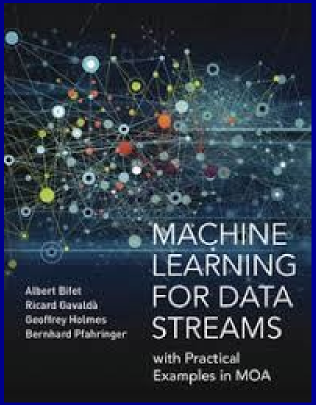


OUTLINE

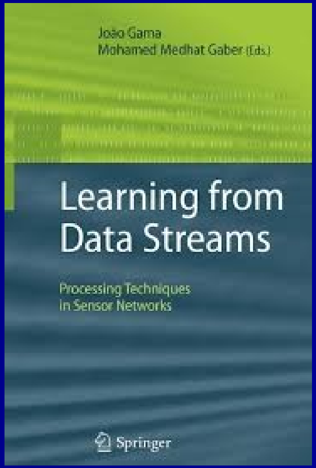
- Data stream → new "data flow" type

- Data stream classification
- Classification with "concept drift"
- Novelty class detection
- Data stream clustering
- Other streaming scenarios

- References and software


Albert Bifet
Ricard Gavaldà
Geoffrey Holmes
Bernhard Pfahringer

MACHINE LEARNING FOR DATA STREAMS
with Practical Examples in MOA


João Gama
Mohamed Medhat Gaber (Eds.)

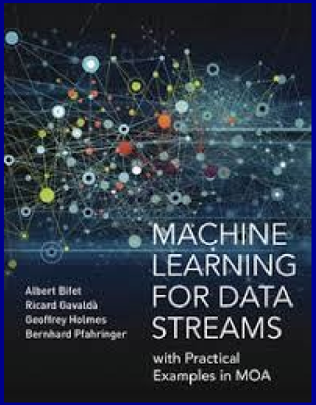
Learning from Data Streams
Processing Techniques in Sensor Networks

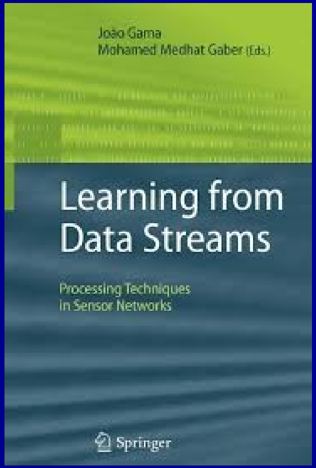
Springer

DATA STREAM – NEW DATA FLOW TYPE -

- "Classic batch-type" data

- All training data "always" available
- Random access to data
- Predefined train + test phases
- No memory + time restrictions


Albert Bifet
Ricard Gavaldà
Geoffrey Holmes
Bernhard Pfahringer
MACHINE LEARNING FOR DATA STREAMS
with Practical Examples in MOA

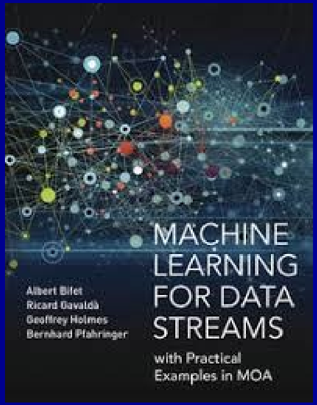

João Gama
Mohamed Medhat Gaber (Eds.)
Learning from Data Streams
Processing Techniques in Sensor Networks
Springer

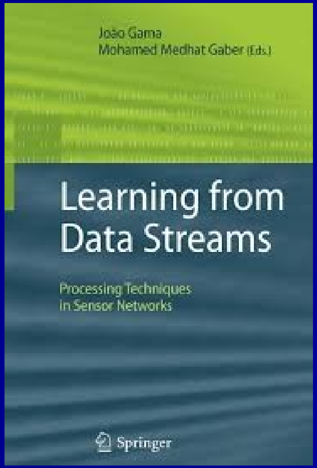
DATA STREAM – NEW DATA FLOW TYPE -

- "Change the chip" !!!!
- "Big Data" pioneers

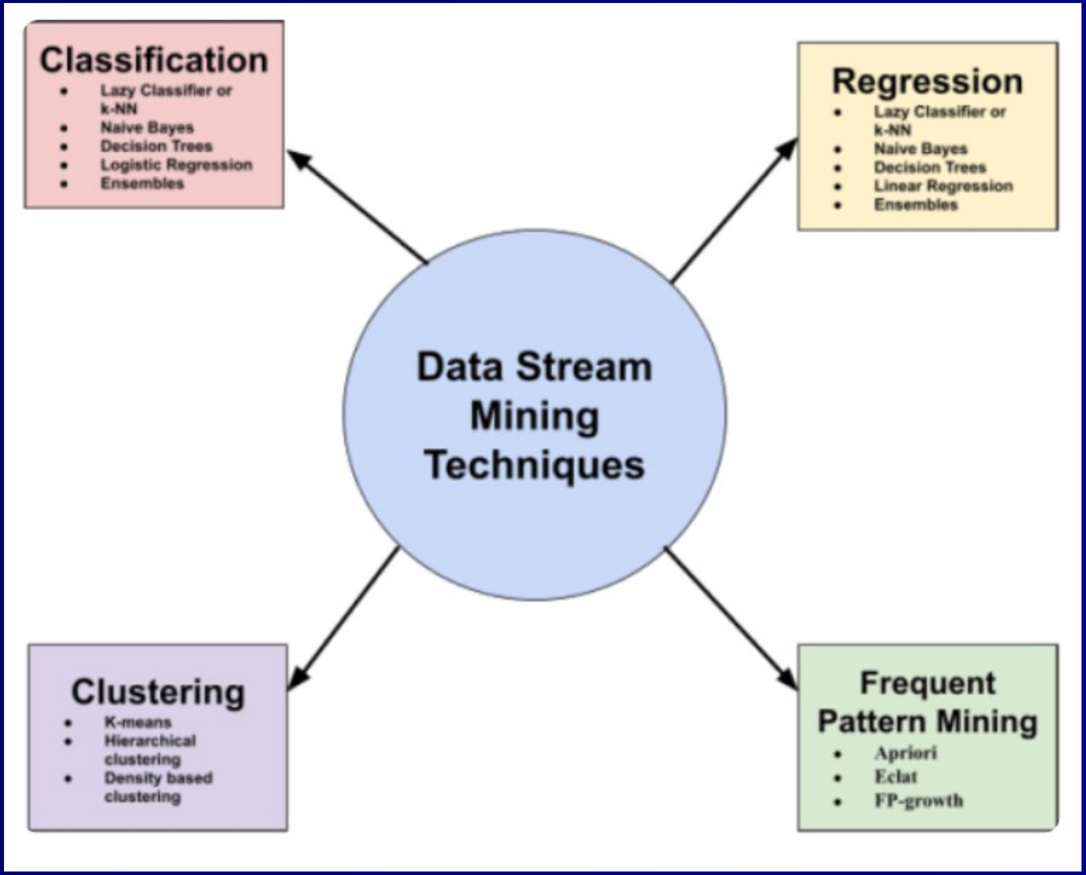
- Data stream data type

- Continuous data flow
- Never all data is available
- Last instance or chunk is only "in memory"
- Time + memory constrains
- Model is only "in memory" → fast update
- Always "ready to predict"
- Model → "on the edge" + "edge computing"
- All data-stream models in practice are different !!

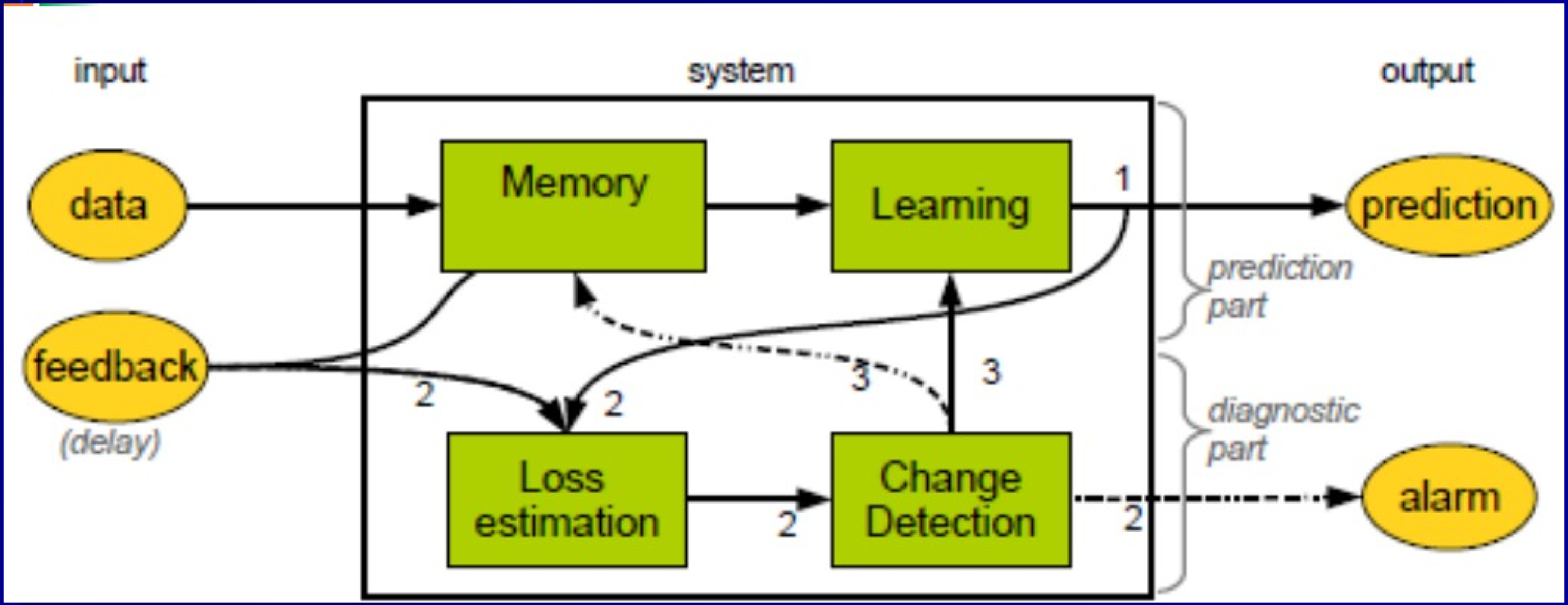

Albert Bifet
Ricard Gavaldà
Geoffrey Holmes
Bernhard Pfahringer
MACHINE LEARNING FOR DATA STREAMS
with Practical Examples in MOA


João Gama
Mohamed Medhat Gaber (Eds.)
Learning from Data Streams
Processing Techniques in Sensor Networks
Springer

DATA STREAM – CLASSIFICATION TASK -



DATA STREAM
– CLASSIFICATION TASK -

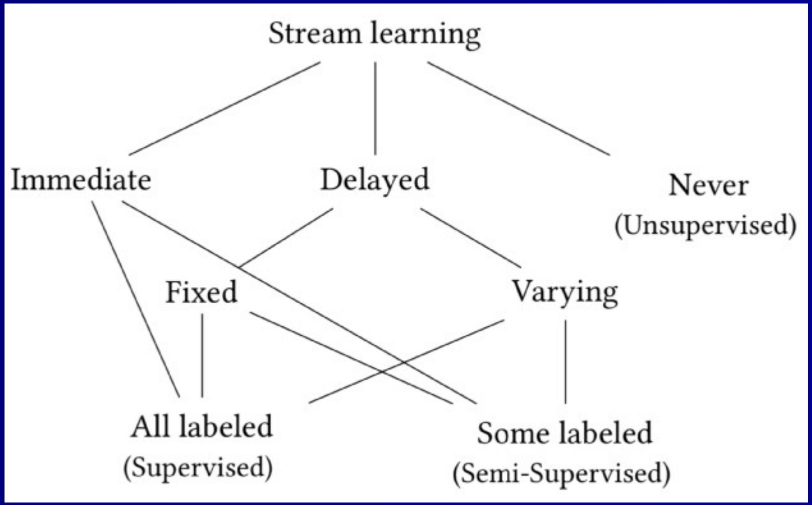


DATA STREAM – CLASSIFICATION TASK –

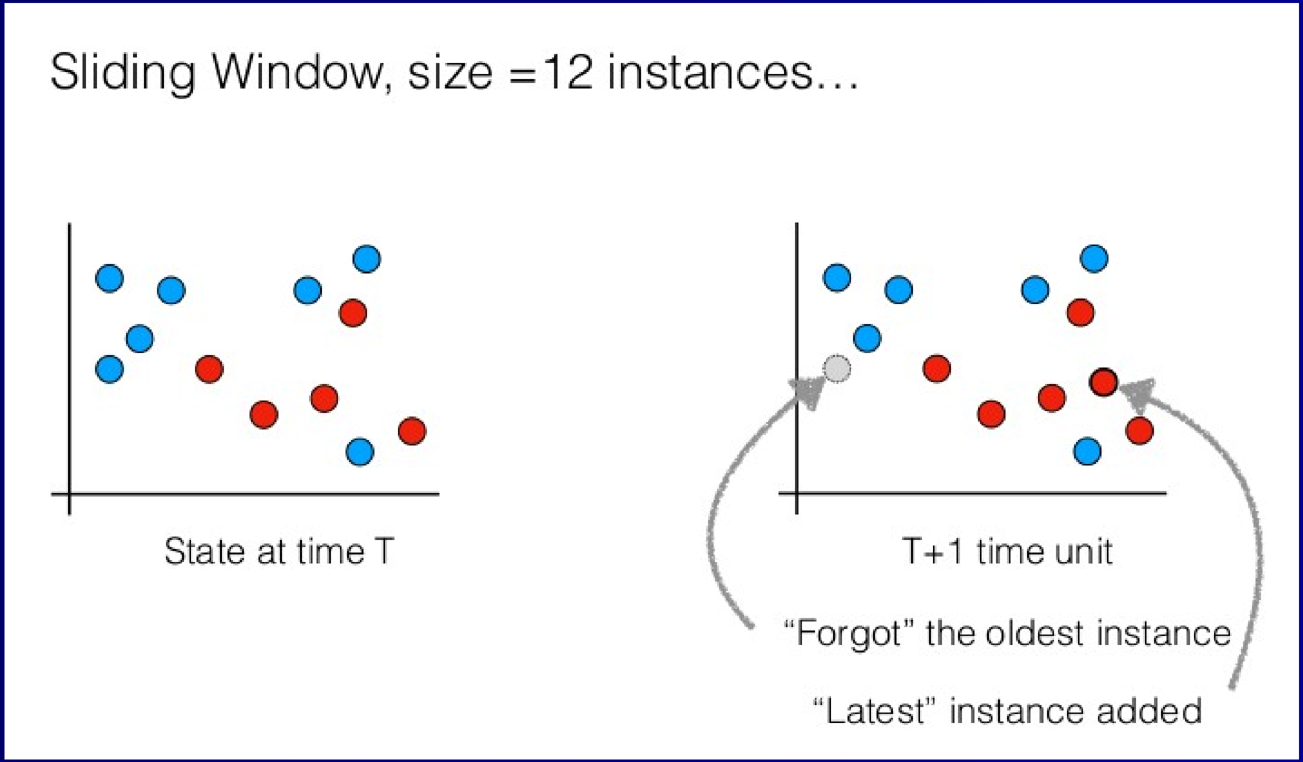
- Labeled stream training data
- Fast model updating
- Be ready always to predict

- Models → K-NN, naive Bayes, Hoeffding trees, ensembles...
- Model evaluation → periodic H, test + train...

- Label arrival? → → →

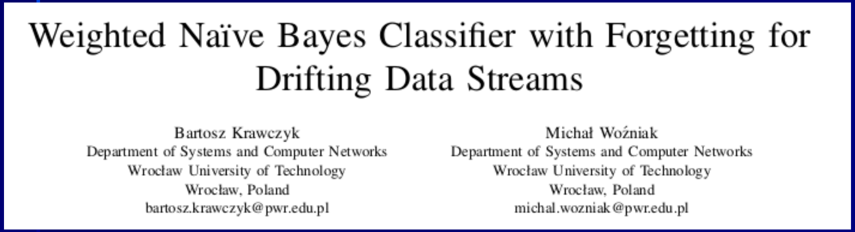


CLASSIFICATION MODELS – SLIDING WINDOW K-NN –


Sliding Window, size =12 instances…
State at time T
T+1 time unit
“Forgot” the oldest instance
“Latest” instance added

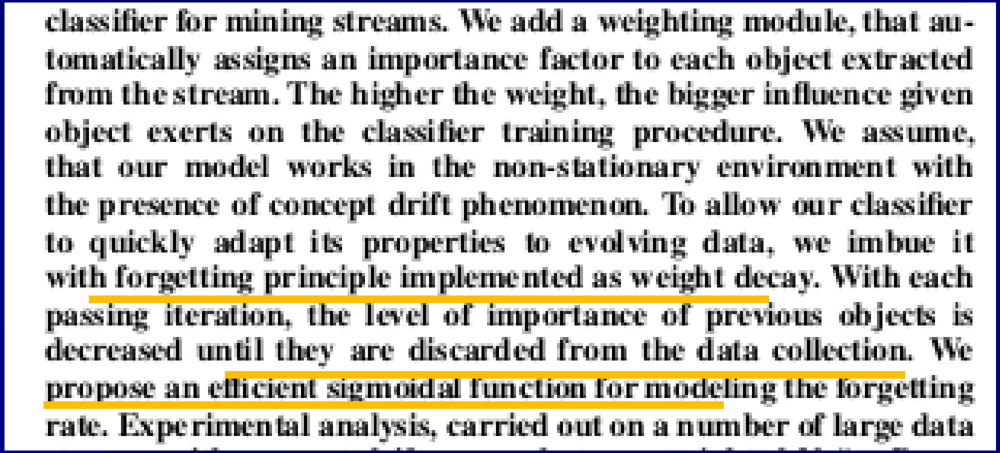
CLASSIFICATION MODELS

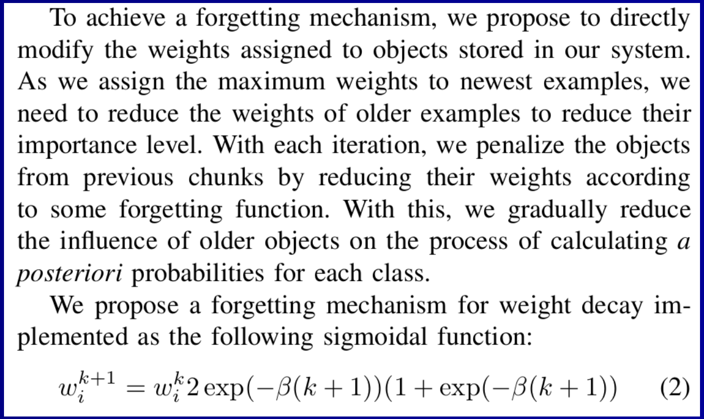
– naive Bayes WITH FORGETTING MECHANISM –


Weighted Naïve Bayes Classifier with Forgetting for Drifting Data Streams

Bartosz Krawczyk
Department of Systems and Computer Networks
Wrocław University of Technology
Wrocław, Poland
bartosz.krawczyk@pwr.edu.pl

Michał Woźniak
Department of Systems and Computer Networks
Wrocław University of Technology
Wrocław, Poland
michal.wozniak@pwr.edu.pl


classifier for mining streams. We add a weighting module, that automatically assigns an importance factor to each object extracted from the stream. The higher the weight, the bigger influence given object exerts on the classifier training procedure. We assume, that our model works in the non-stationary environment with the presence of concept drift phenomenon. To allow our classifier to quickly adapt its properties to evolving data, we imbue it with forgetting principle implemented as weight decay. With each passing iteration, the level of importance of previous objects is decreased until they are discarded from the data collection. We propose an efficient sigmoidal function for modeling the forgetting rate. Experimental analysis, carried out on a number of large data


To achieve a forgetting mechanism, we propose to directly modify the weights assigned to objects stored in our system. As we assign the maximum weights to newest examples, we need to reduce the weights of older examples to reduce their importance level. With each iteration, we penalize the objects from previous chunks by reducing their weights according to some forgetting function. With this, we gradually reduce the influence of older objects on the process of calculating *a posteriori* probabilities for each class.

We propose a forgetting mechanism for weight decay implemented as the following sigmoidal function:

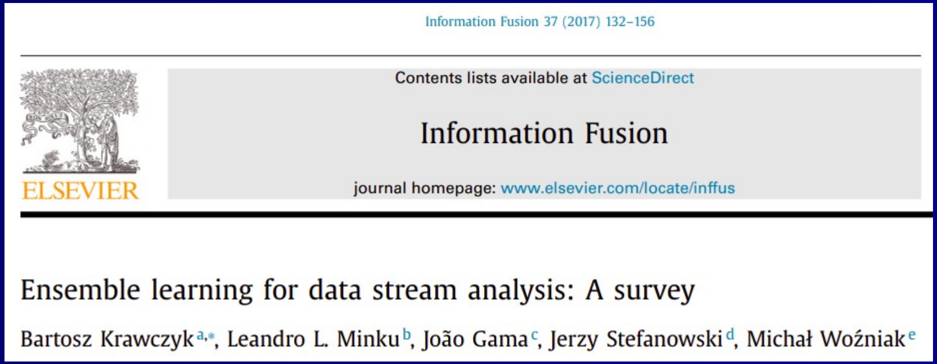
$$w_i^{k+1} = w_i^k 2\exp(-\beta(k+1))(1+\exp(-\beta(k+1)) \qquad (2)$$

$$p(c_m) = \frac{1+\sum_{i=1}^{N} I(c_{x_i} = c_m) w_i}{M + sum_{i=1}^{N} w_i}$$

$$p(a_j|c_m) = \frac{1+\sum_{i=1}^{N} I(a_j = a_{ij}) I(c_{x_i} = c_m) w_i}{n_j + sum_{i=1}^{N} I(a_j = a_{ij}) w_i}$$

CLASSIFICATION MODELS

– ensemble classifiers –



Information Fusion 37 (2017) 132–156

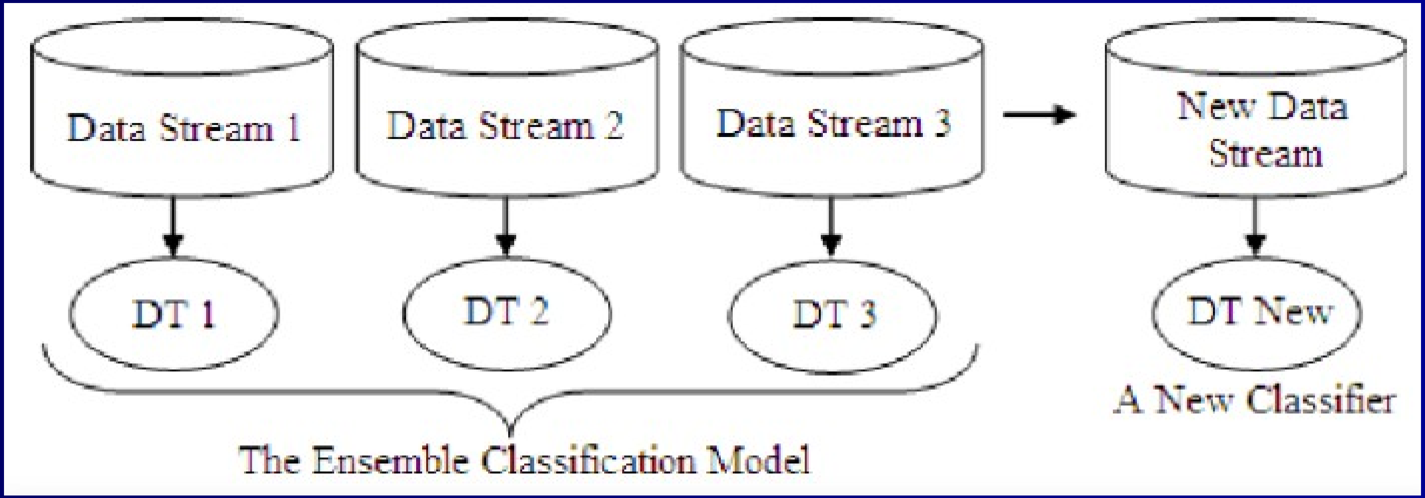
Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Ensemble learning for data stream analysis: A survey

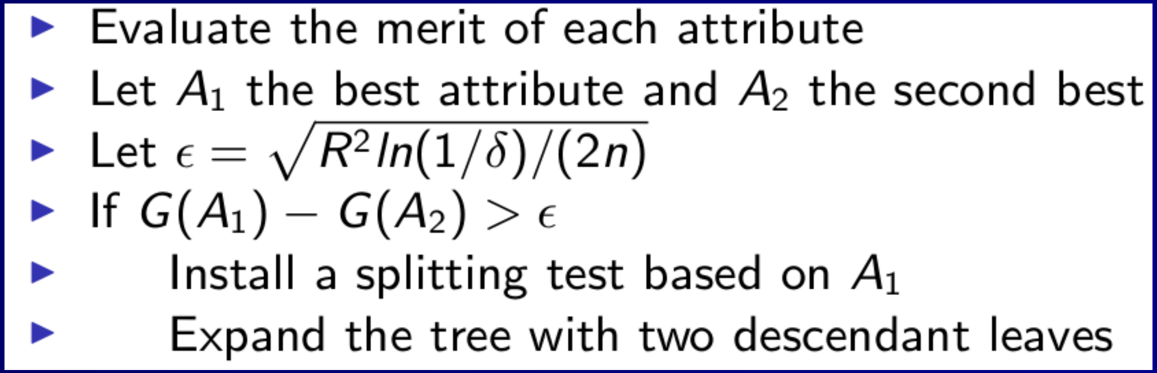
Bartosz Krawczyk[a,*], Leandro L. Minku[b], João Gama[c], Jerzy Stefanowski[d], Michał Woźniak[e]



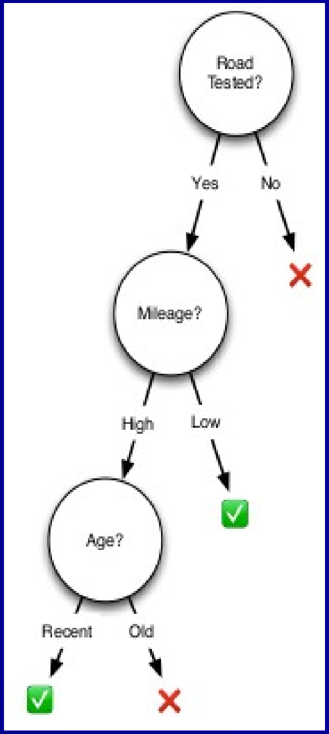
CLASSIFICATION MODELS
– Hoeffding Decision Trees –

- “Classic” decision tree theory

- Key → split-merit calculation
- Only expand a leaf when…
  → enough guarantee to be the “best split”

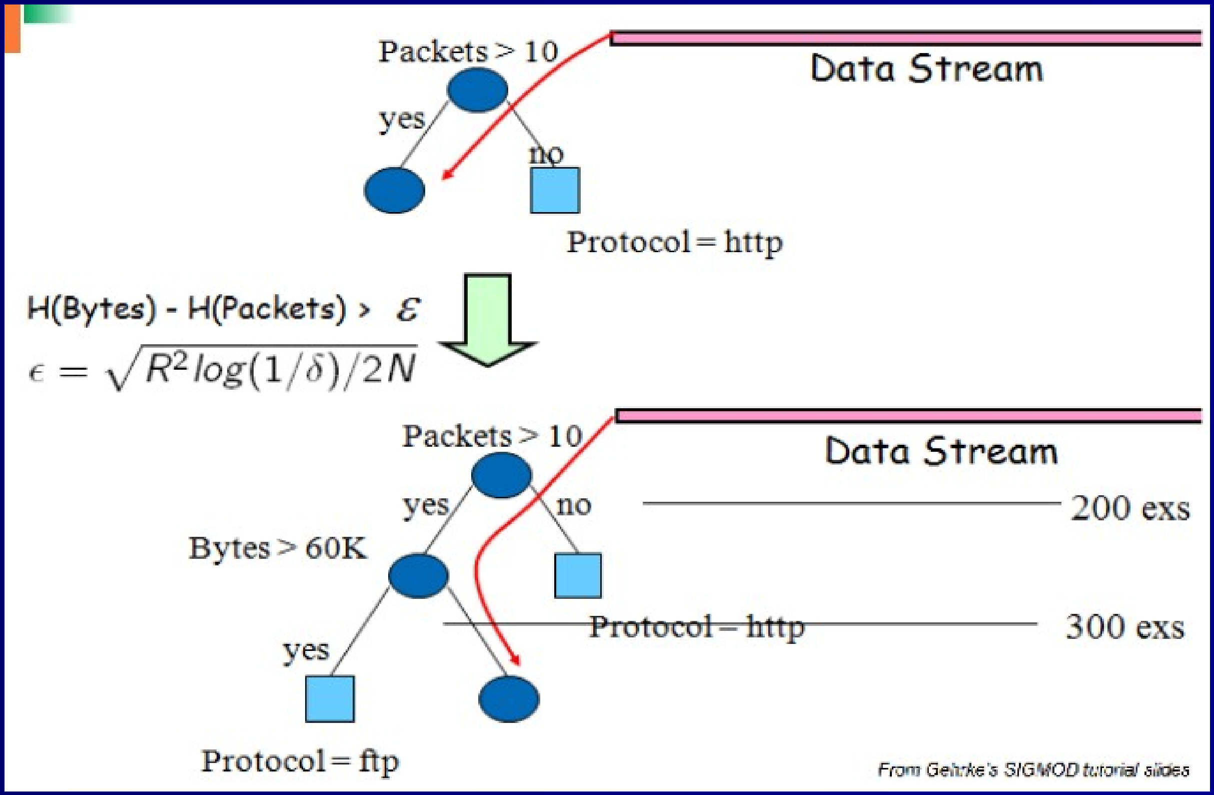


```
▶ Evaluate the merit of each attribute
▶ Let $A_1$ the best attribute and $A_2$ the second best
▶ Let $\epsilon = \sqrt{R^2 ln(1/\delta)/(2n)}$
▶ If $G(A_1) - G(A_2) > \epsilon$
▶       Install a splitting test based on $A_1$
▶       Expand the tree with two descendant leaves
```



CLASSIFICATION MODELS – Hoeffding Decision Trees –

- Only expand leaf when → … enough guarantee to be the "best split"



CLASSIFICATION MODELS – Hoeffding Decision Trees –

An statistical result known as *Hoeffding bound*, which bounds probability that the sum of independent variables (s) deviates from its expected value (X) by more than a certain amount, δ:

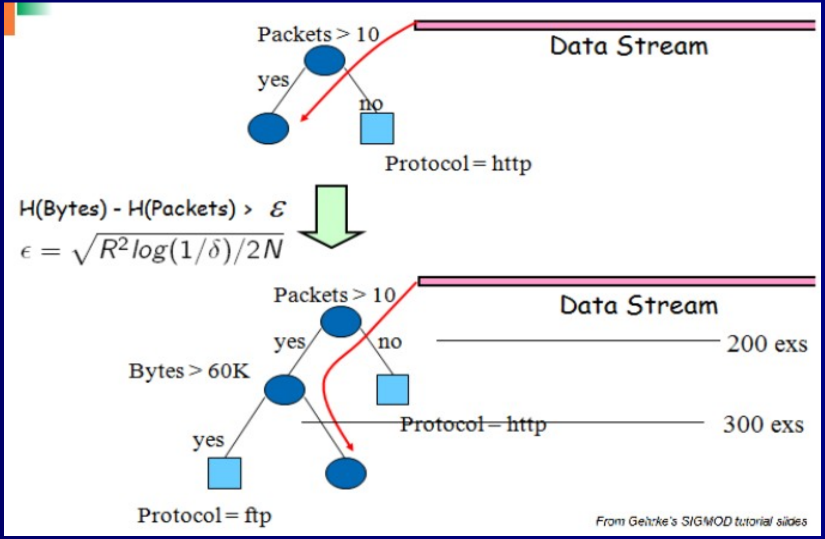
$$p(|X - s| > \epsilon) \leq \delta$$

where $\epsilon = \sqrt{\frac{R^2 \ln \frac{1}{\delta}}{2n}}$, n number of samples y R range of the variable (e.g. $\log c$ for the entropy as 'split' metric in trees, being c number of classes). The value of the upper bound, δ, a low probability value fixed by the user (e.g. 0.05).

Mining High-Speed Data Streams

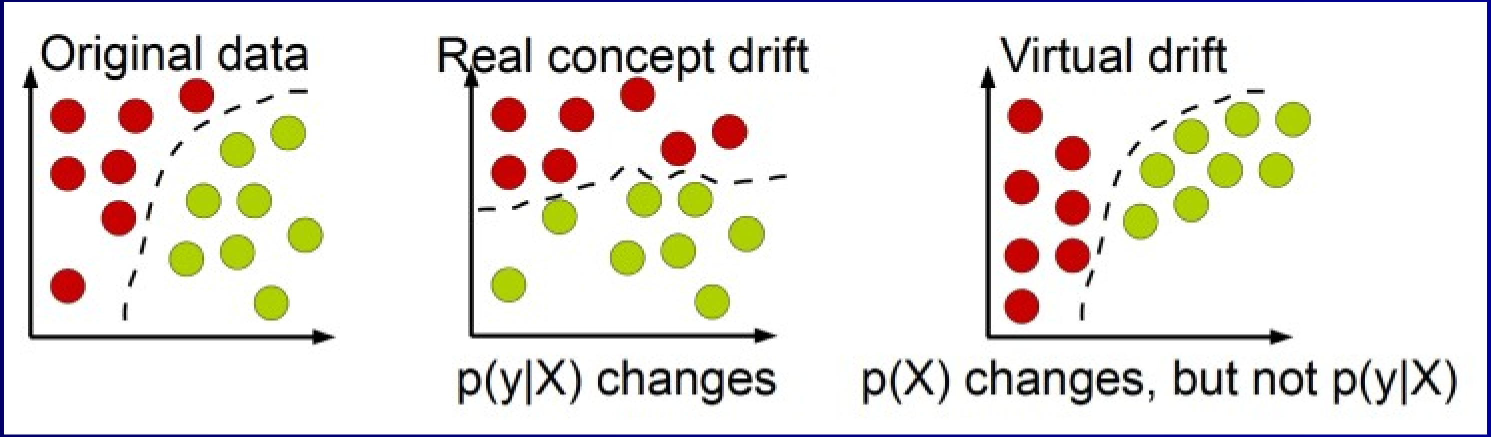
Pedro Domingos
Dept. of Computer Science & Engineering
University of Washington
Box 352350
Seattle, WA 98195-2350, U.S.A.
pedrod@cs.washington.edu

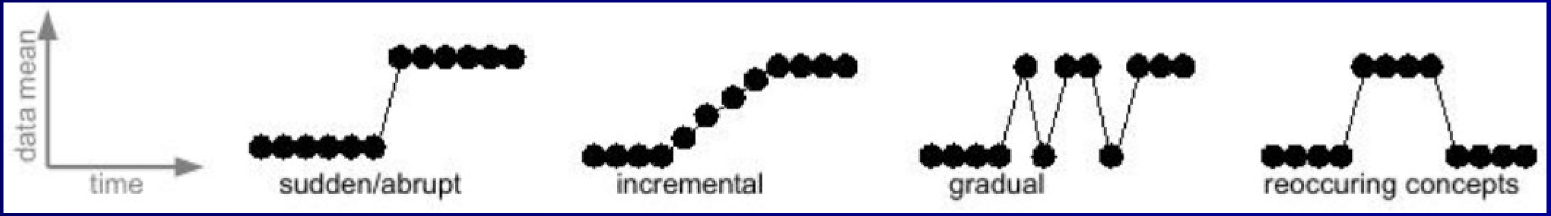
Geoff Hulten
Dept. of Computer Science & Engineering
University of Washington
Box 352350
Seattle, WA 98195-2350, U.S.A.
ghulten@cs.washington.edu



CLASSIFICATION WITH – CONCEPT DRIFT –

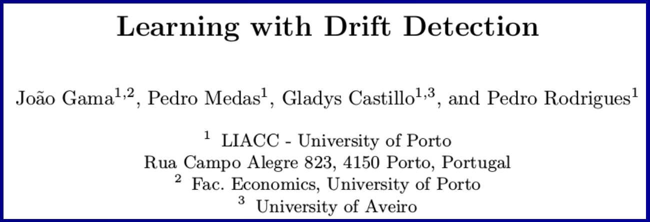
Change on data distribution → “evolving data”





CONCEPT DRIFT

- How detect an “accuracy decay”?
- “Independent” of any classifier
- → “Drift Detection Method” (DDM) →



Learning with Drift Detection

João Gama[1,2], Pedro Medas[1], Gladys Castillo[1,3], and Pedro Rodrigues[1]

[1] LIACC - University of Porto
Rua Campo Alegre 823, 4150 Porto, Portugal
[2] Fac. Economics, University of Porto
[3] University of Aveiro

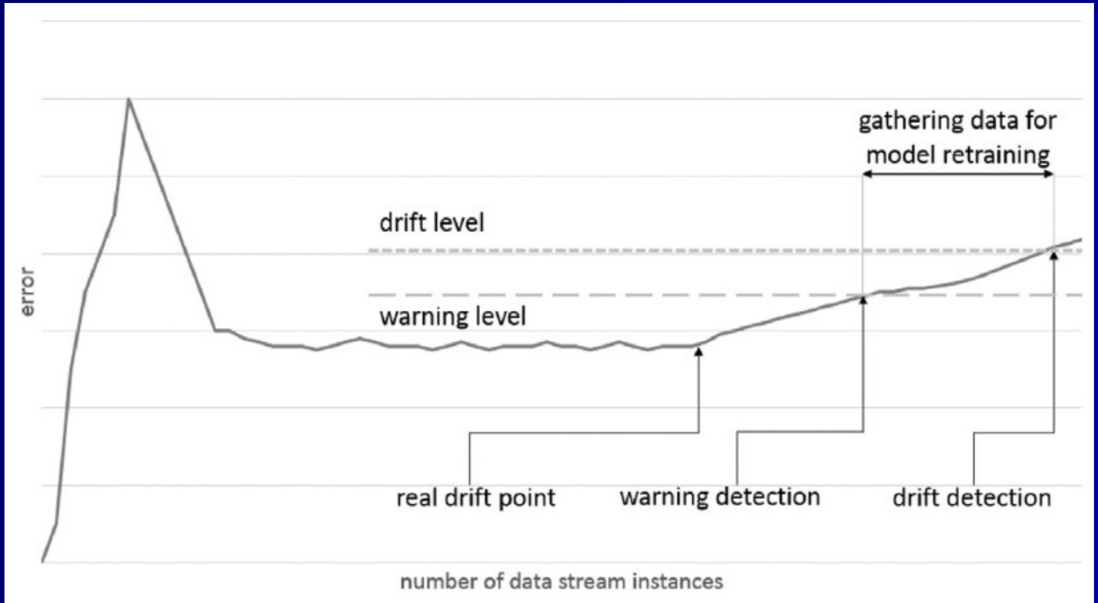


Fig. 5. The idea of drift detection based on tracking classifier errors.

DRIFT DETECTION METHOD – DDM

- p_t → error since last model update
- p_{min} → minimum error since last model update

- "Warning level" → → → → → → → $p_t + s_t \geq p_{min} + 2 \cdot s_{min}$

- "Drift level" → → → → → → → → → $p_t + s_t \geq p_{min} + 3 \cdot s_{min}$

- "Declare drift"!
- Re-learn model → with samples since "warning declare"

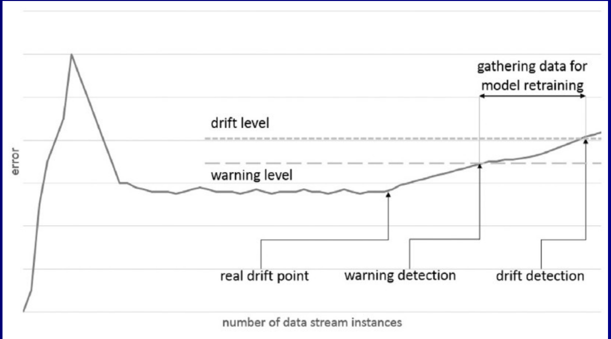


Fig. 5. The idea of drift detection based on tracking classifier errors.

DRIFT DETECTION METHOD – DDM

- p_t → error since last model update
- p_{min} → minimum error since last model update
- "Warning level" → → → → → → → $p_t + s_t \geq p_{min} + 2 \cdot s_{min}$
- "Drift level" → → → → → → → → → $p_t + s_t \geq p_{min} + 3 \cdot s_{min}$
- S_t? S_{min}?
- $p_i \sim Binomial(p_i)$ → → → → → → $s_i = \sqrt{\frac{p_i(1-p_i)}{i}}$

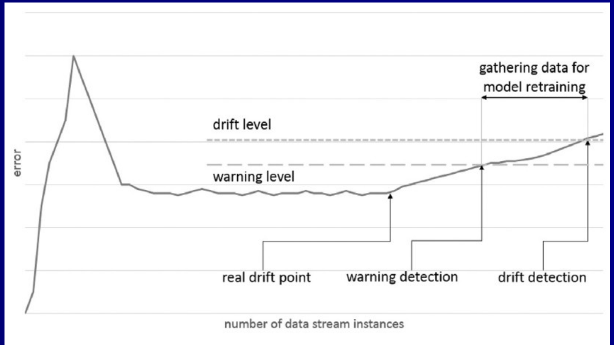


Fig. 5. The idea of drift detection based on tracking classifier errors.

CONCEPT DRIFT
– EARLY DRIFT DETECTION METHOD –

- Consider → distance between 2 errors

$$(p'_i + 2 \cdot s'_i)/(p'_{max} + 2 \cdot s'_{max}) < \alpha \text{ for the warning level}$$

$$(p'_i + 2 \cdot s'_i)/(p'_{max} + 2 \cdot s'_{max}) < \beta \text{ for the drift level}$$

- p_i → average distance between 2 errors
- p_{max} → when p_i reaches its maximum value
  → best approximation the stream data
- α=0.95? β=0.90? → needed to user fix

- Start calculation after 30 errors

- Better for "early + gradual" drift detection

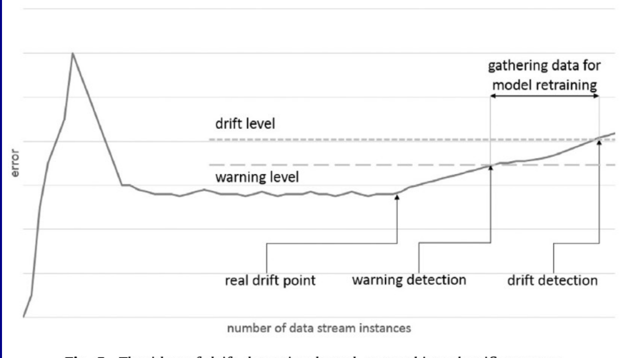


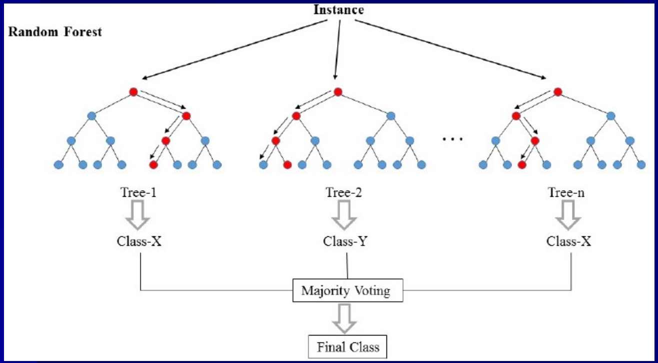
Fig. 5. The idea of drift detection based on tracking classifier errors.

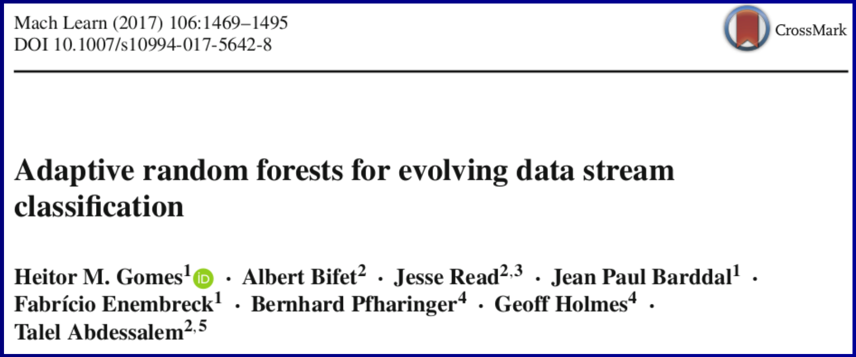
CLASSIFICATION MODELS – ADAPTIVE RANDOM FOREST –

- Ensemble of trees

- Train each tree → in a resampling of stream instances – bagging
- Randomly select subsets of features as candidate split in nodes
  → Adding diversity + Reduce variance

- When a tree arrives to "warning" level
  → start training in background a new tree
  → replacing active tree if "drift" level arrived



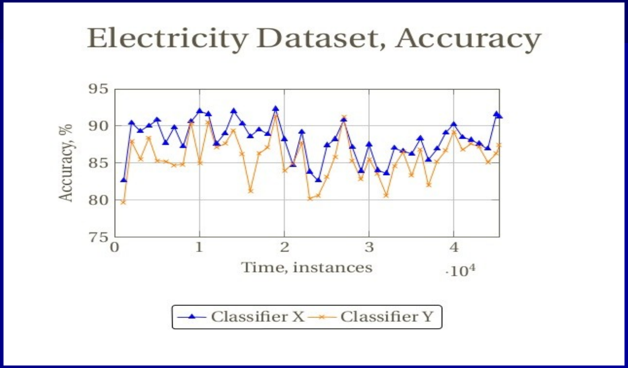

Mach Learn (2017) 106:1469–1495
DOI 10.1007/s10994-017-5642-8

CrossMark

Adaptive random forests for evolving data stream classification

Heitor M. Gomes[1] · Albert Bifet[2] · Jesse Read[2,3] · Jean Paul Barddal[1] · Fabrício Enembreck[1] · Bernhard Pfharinger[4] · Geoff Holmes[4] · Talel Abdessalem[2,5]

CLASSIFICATION MODELS – EVALUATION –



- Periodic Hold-Out
  - Train-stream flows
  - Test with the same test subset
- InterLeaved Test-Then-Train
  - 1. Use stream-instance to predict → update score
  - 2. Use the stream-instance to update model
  - Not in the opposite order!!
- Prequential Evaluation
  - Recent instances → larger weight in score calculation
  - Weighting → "sliding window" + "decay factor"

CLASSIFICATION MODELS – NOVELTY DETECTION –

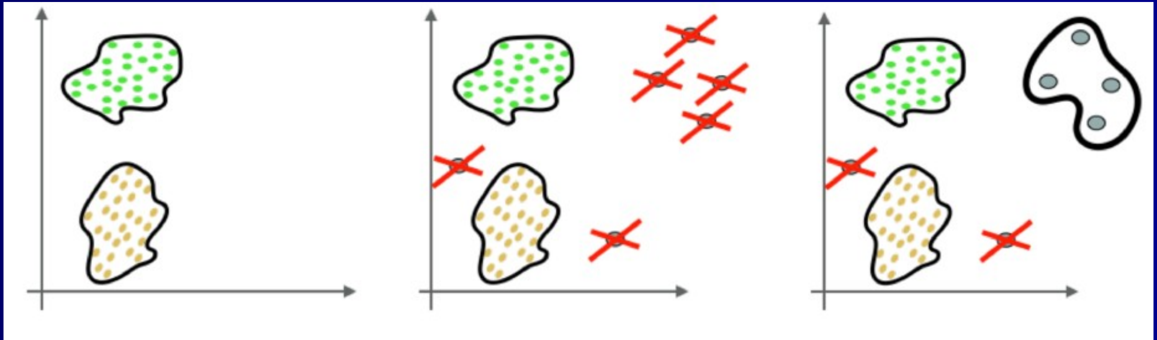
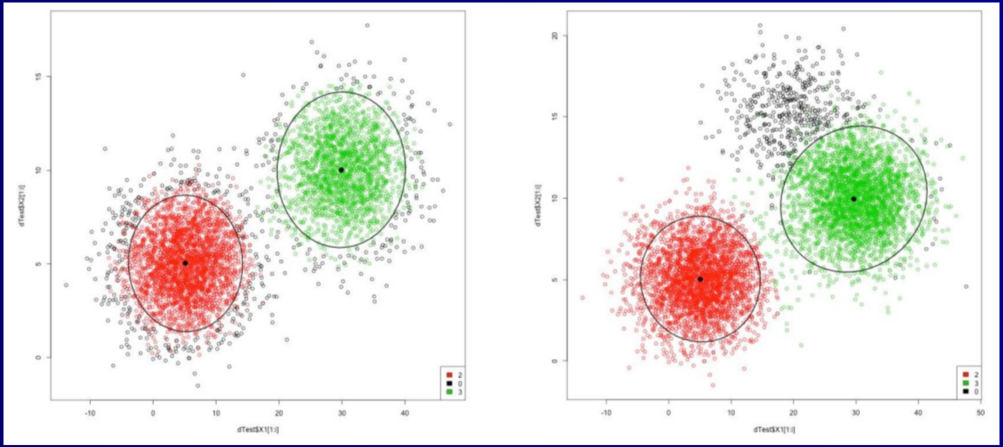
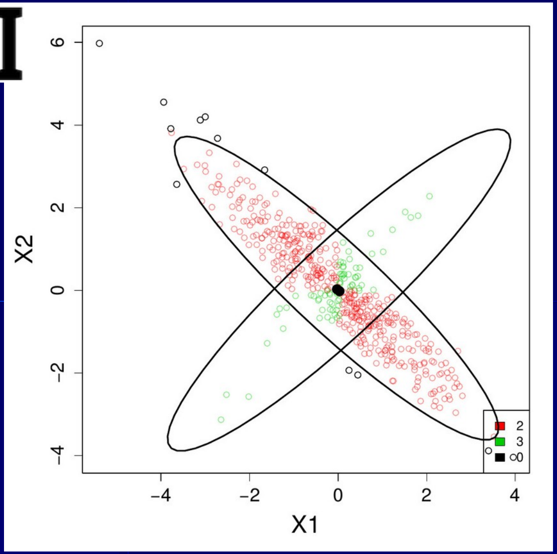


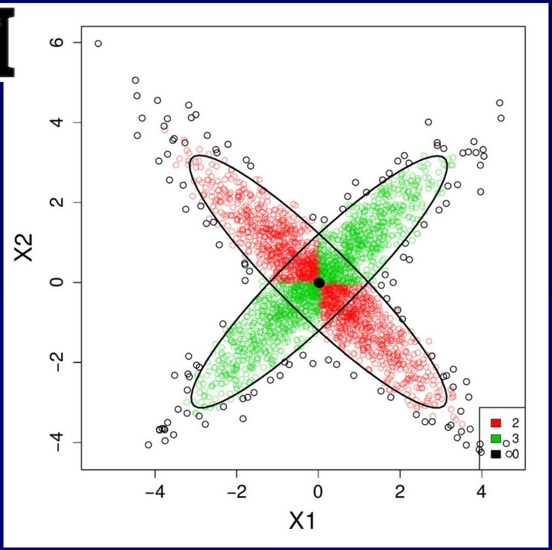
Figure 2: Dynamic novelty detection step by step flowchart.



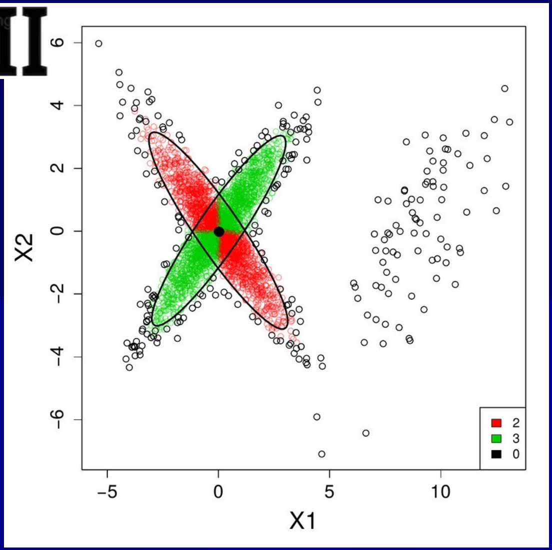
I



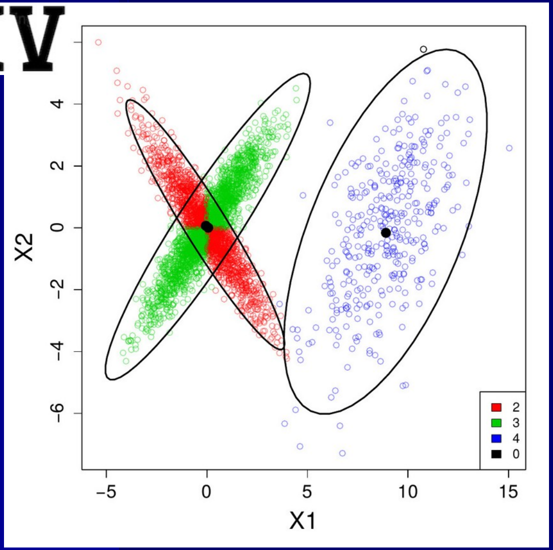
II



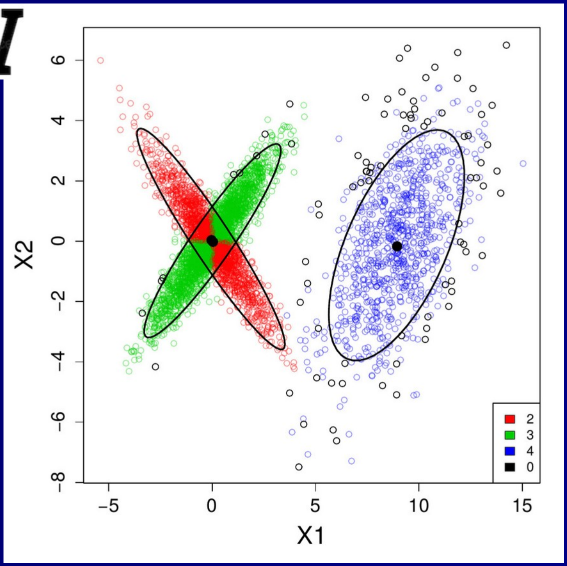
III



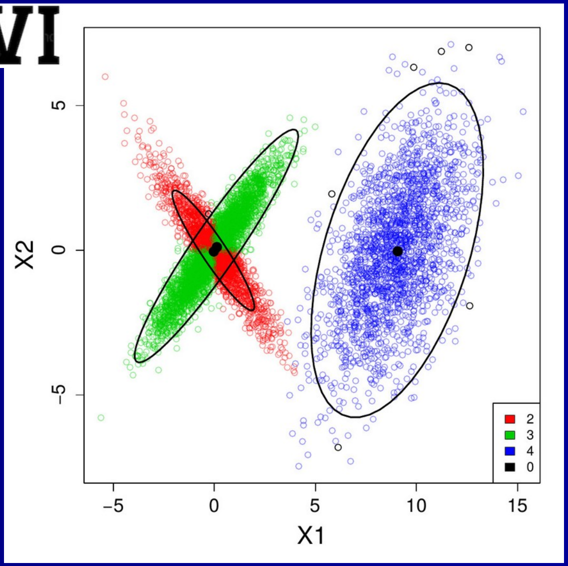
IV



V

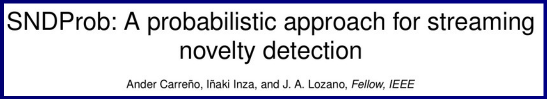


VI



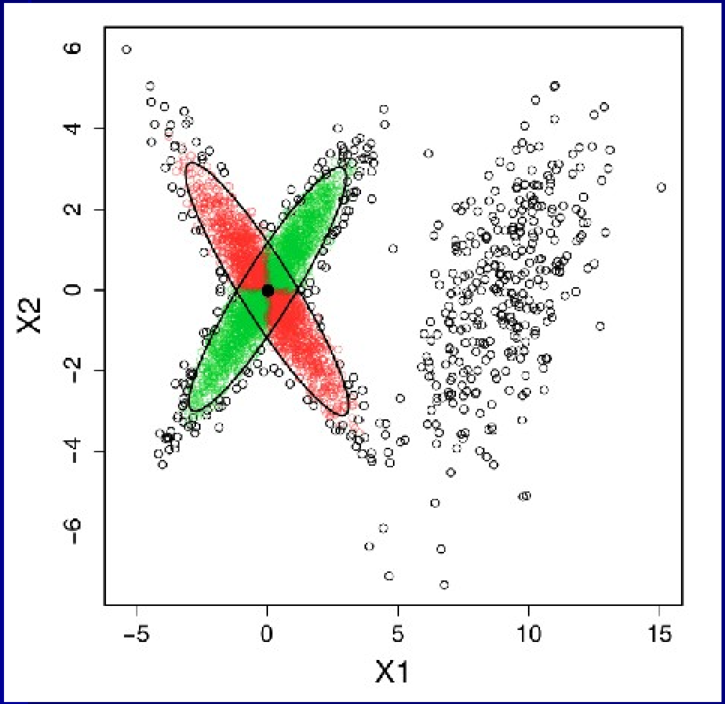
CLASSIFICATION MODELS – NOVELTY DETECTION –

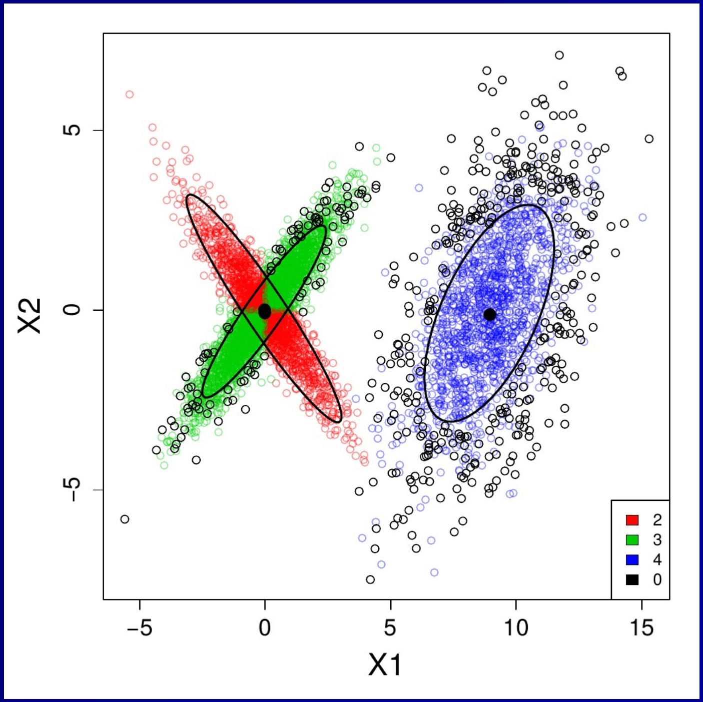
https://github.com/andercarreno/SNDProb



SNDProb: A probabilistic approach for streaming novelty detection

Ander Carreño, Iñaki Inza, and J. A. Lozano, *Fellow, IEEE*





CLASSIFICATION MODELS – NOVELTY DETECTION –

- Other names: "emerging classes"
- Initial off-line phase
  - Learn a multiclass supervised model
- On-line phase
  - Arrival of unlabeled data in streaming form
  - Model update of known classes
  - Consider appearance of new classes → cohesion + separation
  - Consider the removal of outdated classes

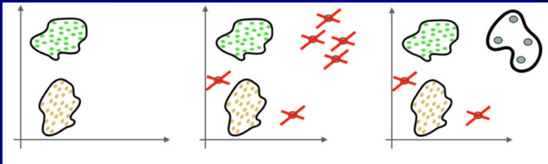
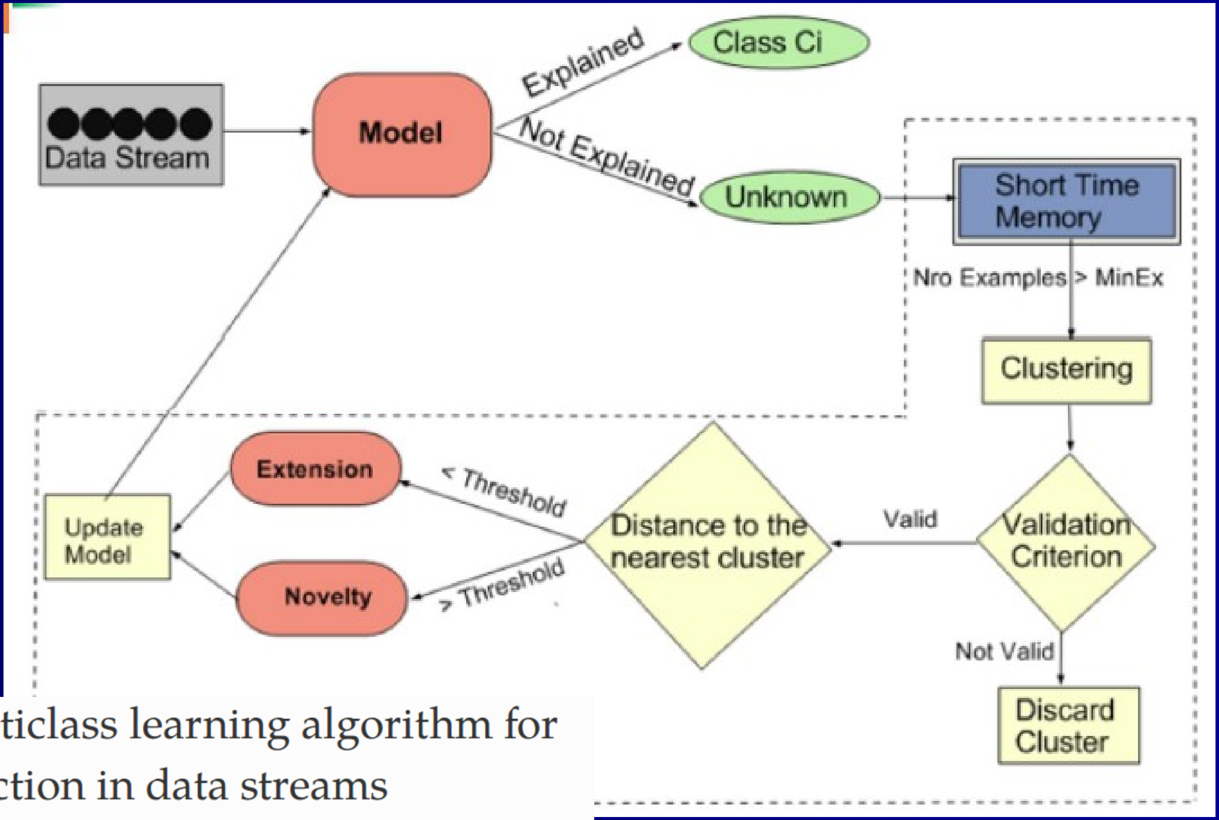
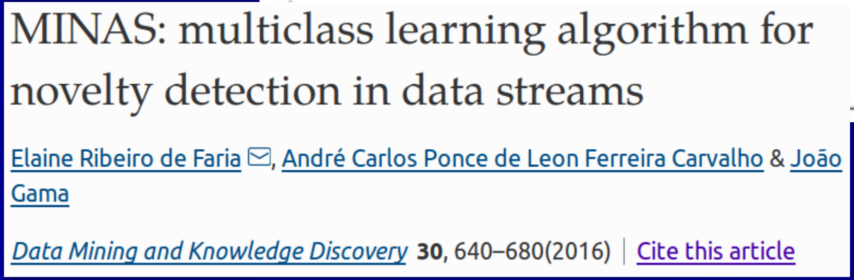


Figure 2: Dynamic novelty detection step by step flowchart.

CLASSIFICATION MODELS – NOVELTY DETECTION –

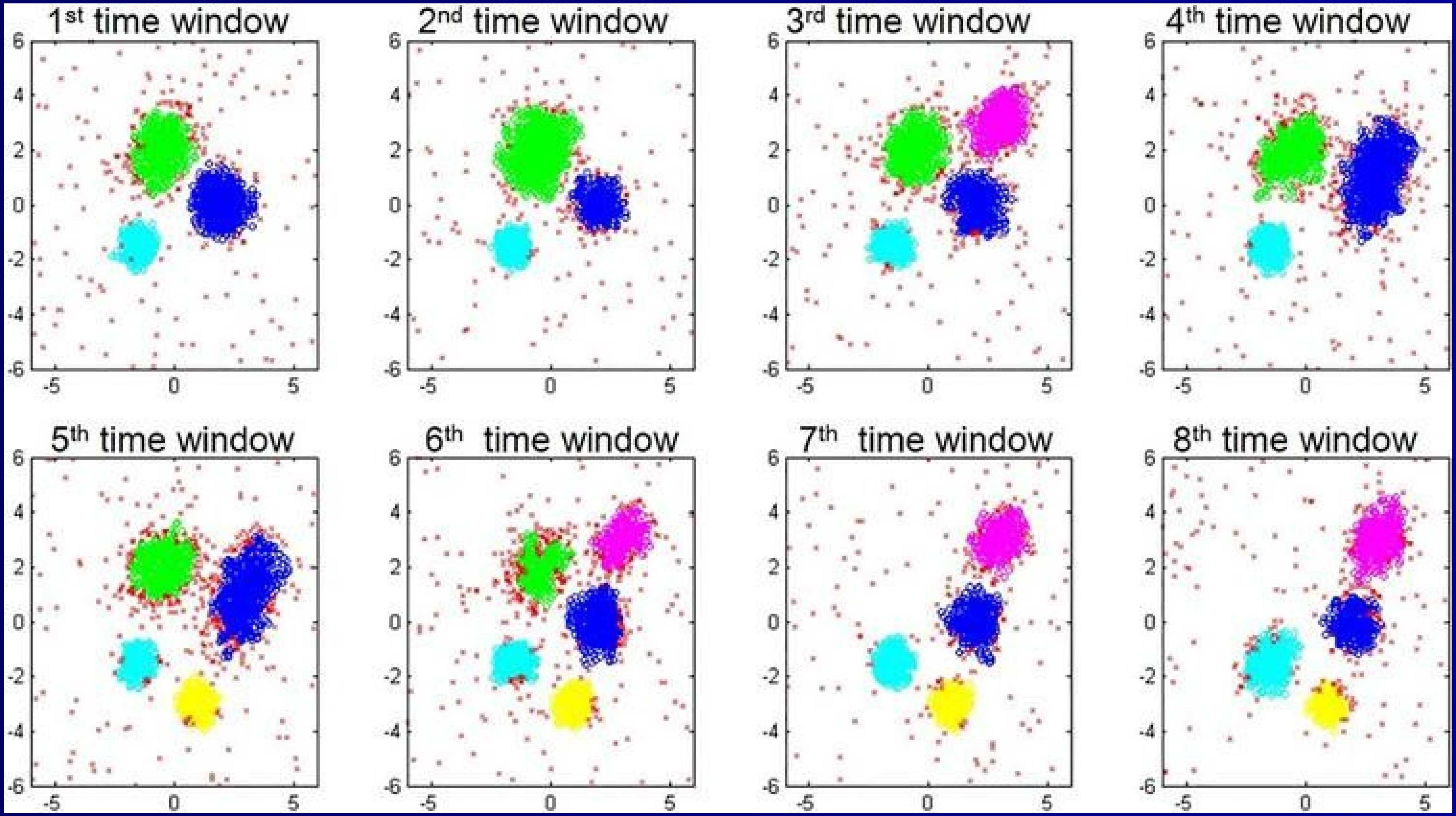



MINAS: multiclass learning algorithm for novelty detection in data streams

Elaine Ribeiro de Faria, André Carlos Ponce de Leon Ferreira Carvalho & João Gama

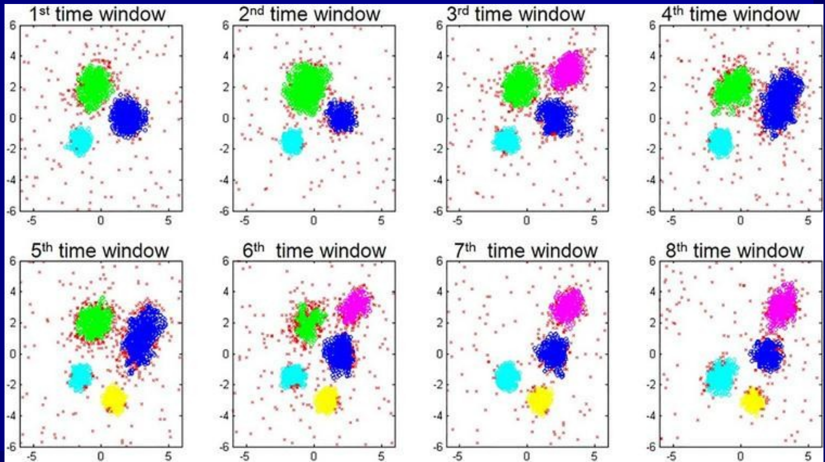
Data Mining and Knowledge Discovery **30**, 640–680(2016) | Cite this article

DATA STREAM
– CLUSTERING TASK –



DATA STREAM – CLUSTERING TASK –

- Two types of algorithms
- Based on partitional clustering (~ k-means)
  - CluStream, BIRCH, ClusTree
- Based on probability density concepts
  - DBScan



DATA STREAM CLUSTERING – PARTITIONAL METHODS –

- CluStream, ClusTree, BIRCH... algorithms

- Based on the "MicroCluster" concept
- MicroCluster → summary representation of data
- Learn 'q' microclusters ~ partitional clustering
- Learn at the beginning of the stream
- q ›› natural number of clusters, k
- q ‹‹ samples in the stream

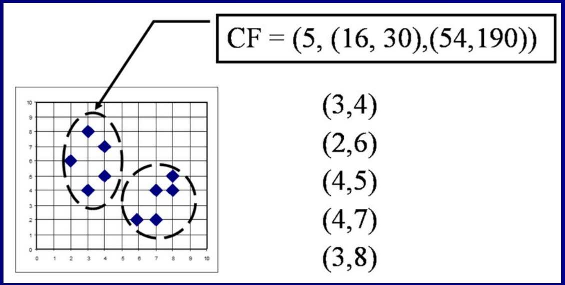
Cluster Feature Vector: $\text{CF} = (N, LS, SS)$

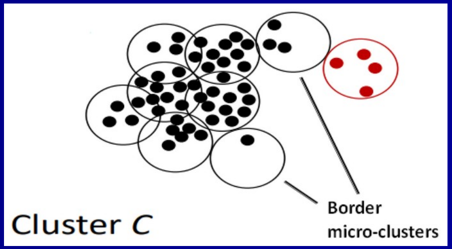
- N: Number of data points
- LS: $\sum_1^N \vec{x_i}$
- SS: $\sum_1^N (\vec{x_i})^2$

$$\text{Centroid} = LS/N$$

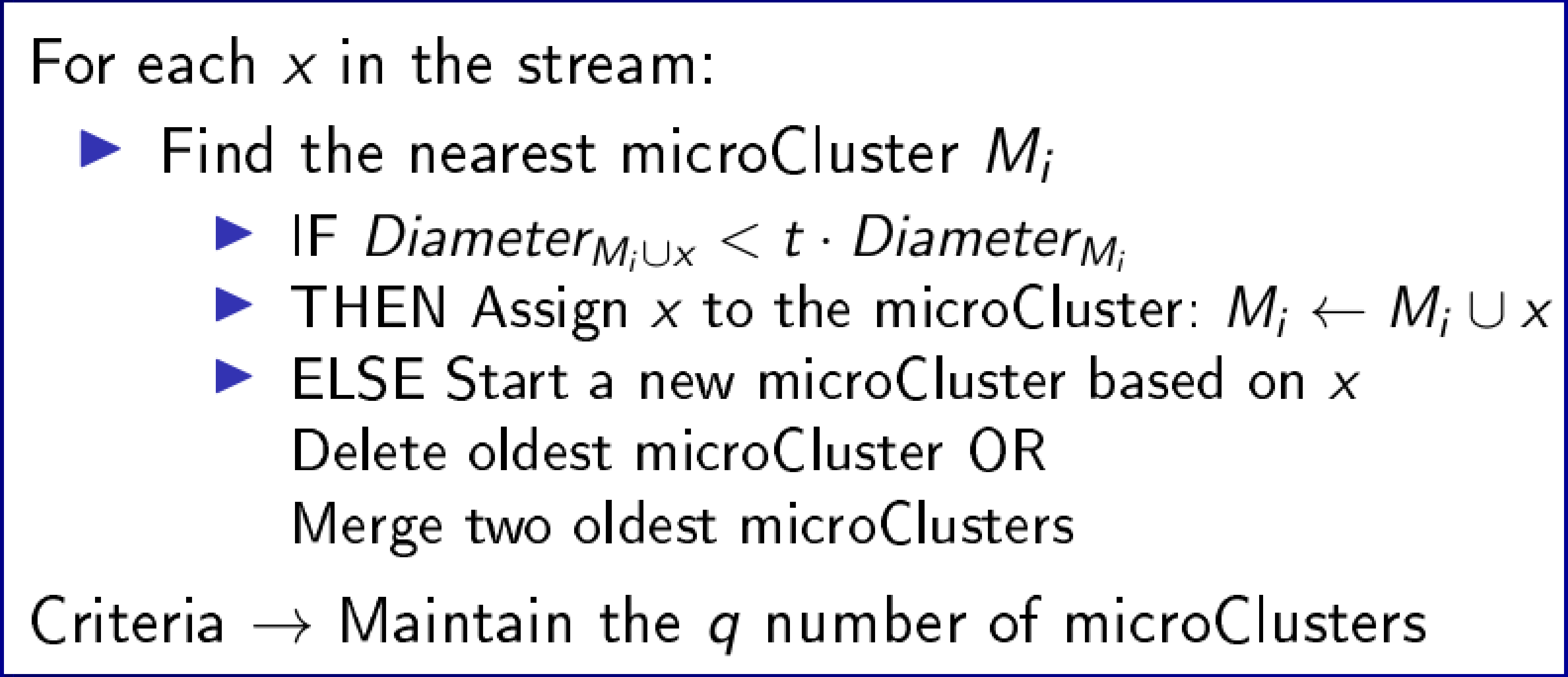
$$\text{Radius} = \sqrt{SS/N - (LS/N)^2}$$

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





– CLUSTREAM ALGORITHM –
– NEW MICROCLUSTER CREATION –



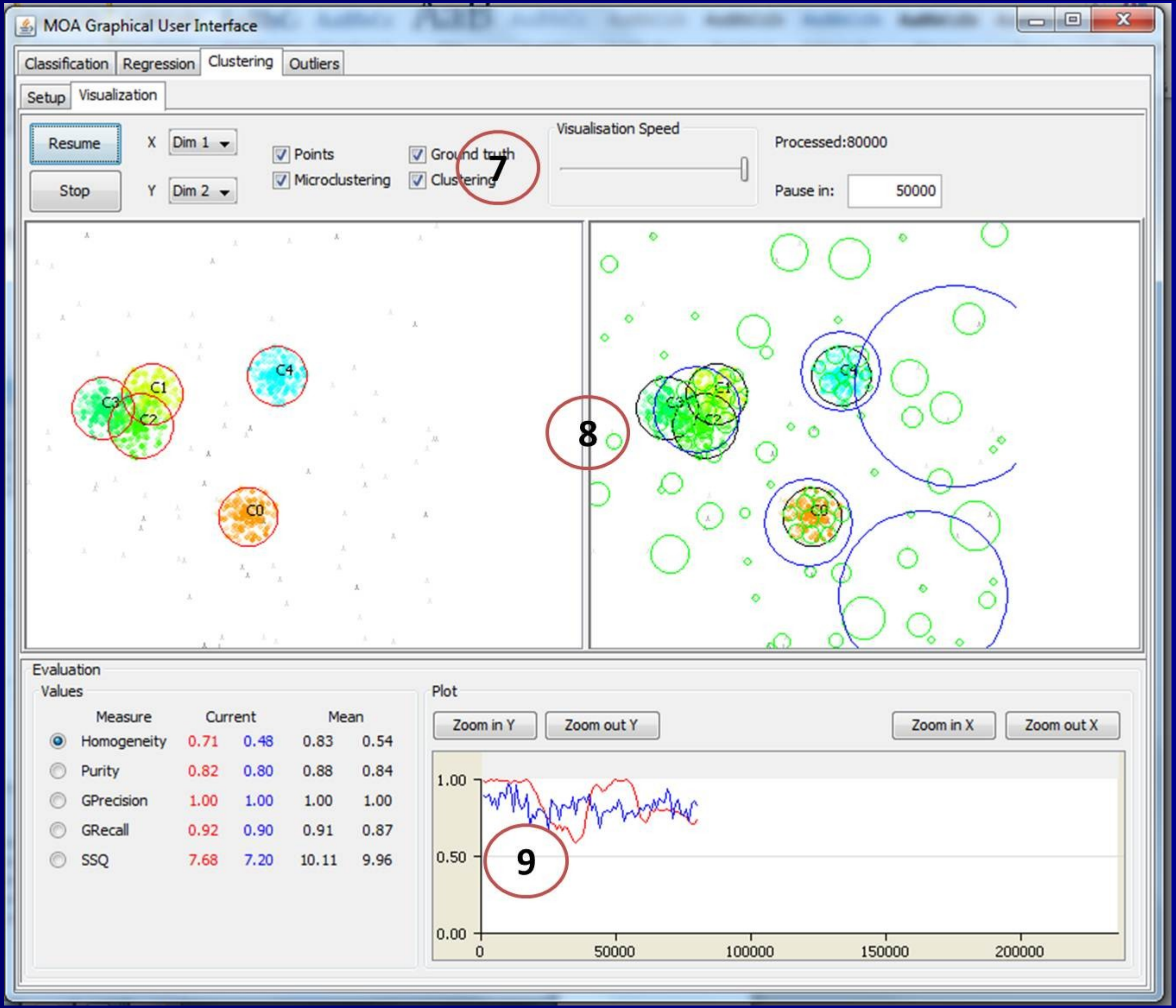
```
For each $x$ in the stream:
  ► Find the nearest microCluster $M_i$
      ► IF $Diameter_{M_i \cup x} < t \cdot Diameter_{M_i}$
      ► THEN Assign $x$ to the microCluster: $M_i \leftarrow M_i \cup x$
      ► ELSE Start a new microCluster based on $x$
        Delete oldest microCluster OR
        Merge two oldest microClusters
Criteria $\rightarrow$ Maintain the $q$ number of microClusters
```

– CLUSTREAM ALGORITHM –
– CREATE FINAL MACROCLUSTERS –

Modify k-means algorithm $\rightarrow$ to obtain final k-macroClusters from q-microClusters ($q > k$)

- Run k-means over microClusters
- Initial seeds are not random $\rightarrow$ sampled with a probability proportional to the number of points in each microCluster
- Distance from the seed to any microCluster $\rightarrow$ distance between the seed and the centroid of the microCluster.
- New seed for a given partition is defined as the weighted centroid of the micro-clusters in that partition.

– CLUSTREAM ALGORITHM –
– CREATE FINAL MACROCLUSTERS –



X: Dim 1
Y: Dim 2

☑ Points
☑ Microclustering
☑ Ground truth
☑ Clustering

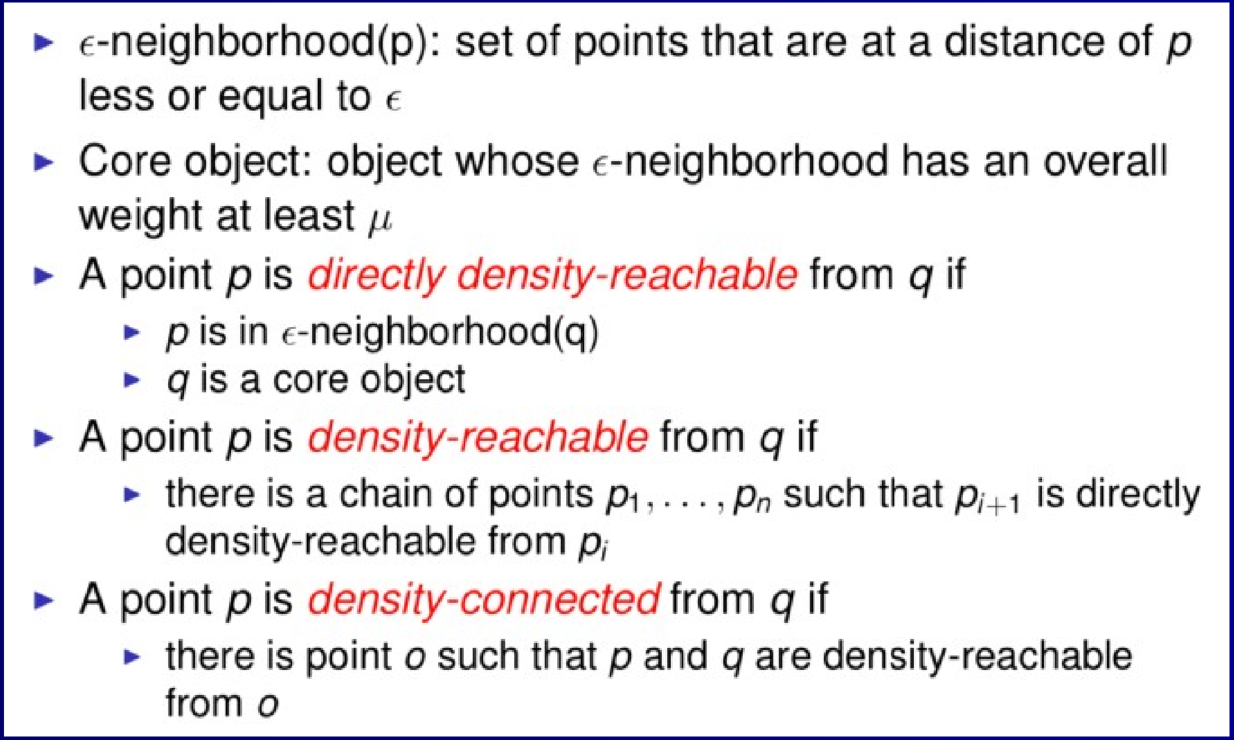
Processed:80000
Pause in: 50000

Values

	Measure	Current		Mean	
●	Homogeneity	0.71	0.48	0.83	0.54
○	Purity	0.82	0.80	0.88	0.84
○	GPrecision	1.00	1.00	1.00	1.00
○	GRecall	0.92	0.90	0.91	0.87
○	SSQ	7.68	7.20	10.11	9.96

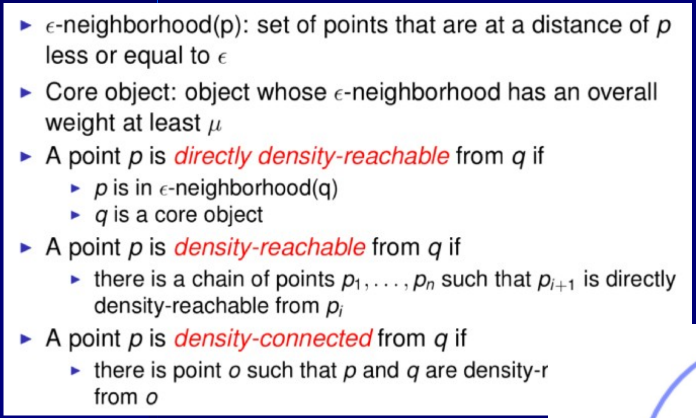
DATA STREAM CLUSTERING – DENSITY-BASED METHODS –

- **DBScan, DenStream…**
- **Based on the “Neighborhood” concept**

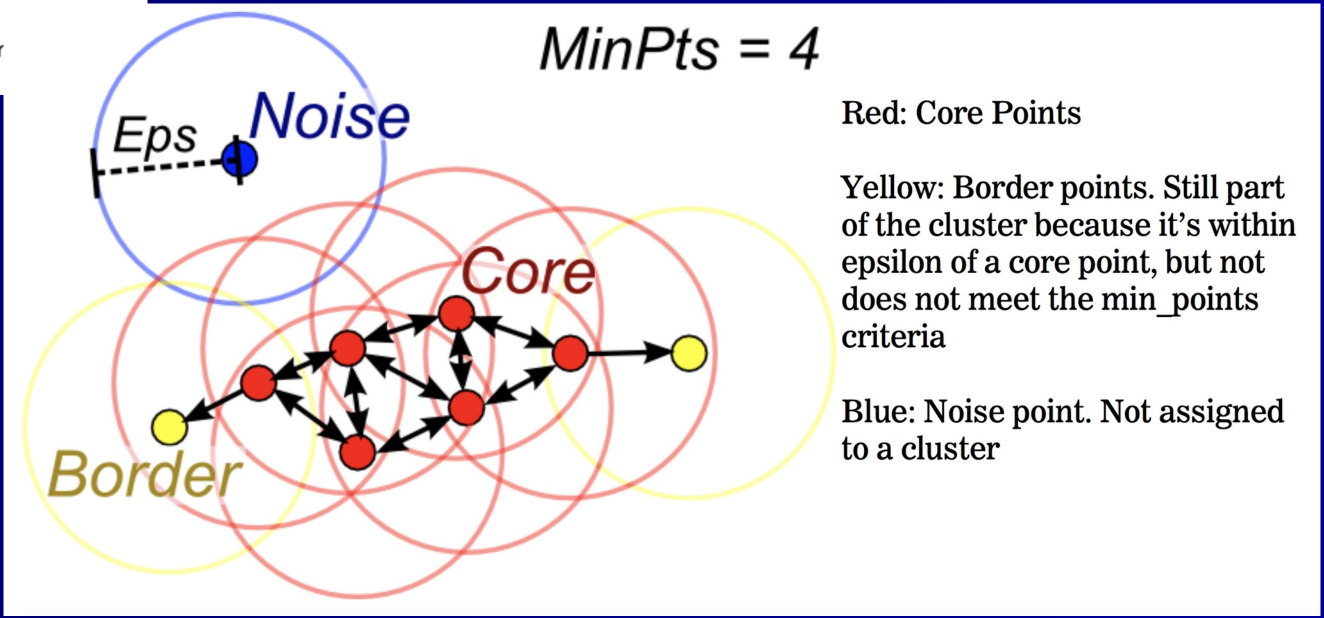


- ϵ-neighborhood(p): set of points that are at a distance of p less or equal to ϵ
- Core object: object whose ϵ-neighborhood has an overall weight at least μ
- A point p is *directly density-reachable* from q if
  - p is in ϵ-neighborhood(q)
  - q is a core object
- A point p is *density-reachable* from q if
  - there is a chain of points $p_1, \ldots, p_n$ such that p_{i+1} is directly density-reachable from p_i
- A point p is *density-connected* from q if
  - there is point o such that p and q are density-reachable from o

DATA STREAM CLUSTERING – DBScan ALGORITHM –



- ϵ-neighborhood(p): set of points that are at a distance of p less or equal to ϵ
- Core object: object whose ϵ-neighborhood has an overall weight at least μ
- A point p is *directly density-reachable* from q if
  - p is in ϵ-neighborhood(q)
  - q is a core object
- A point p is *density-reachable* from q if
  - there is a chain of points $p_1, \ldots, p_n$ such that p_{i+1} is directly density-reachable from p_i
- A point p is *density-connected* from q if
  - there is point o such that p and q are density-r
    from o

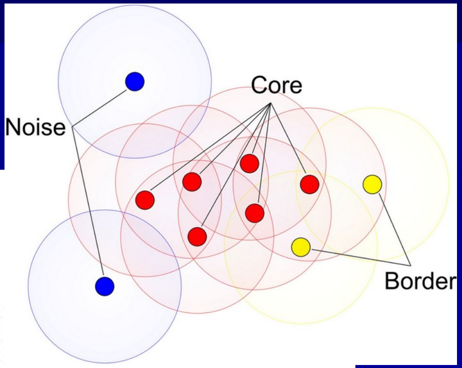


Red: Core Points

Yellow: Border points. Still part of the cluster because it's within epsilon of a core point, but not does not meet the min_points criteria

Blue: Noise point. Not assigned to a cluster

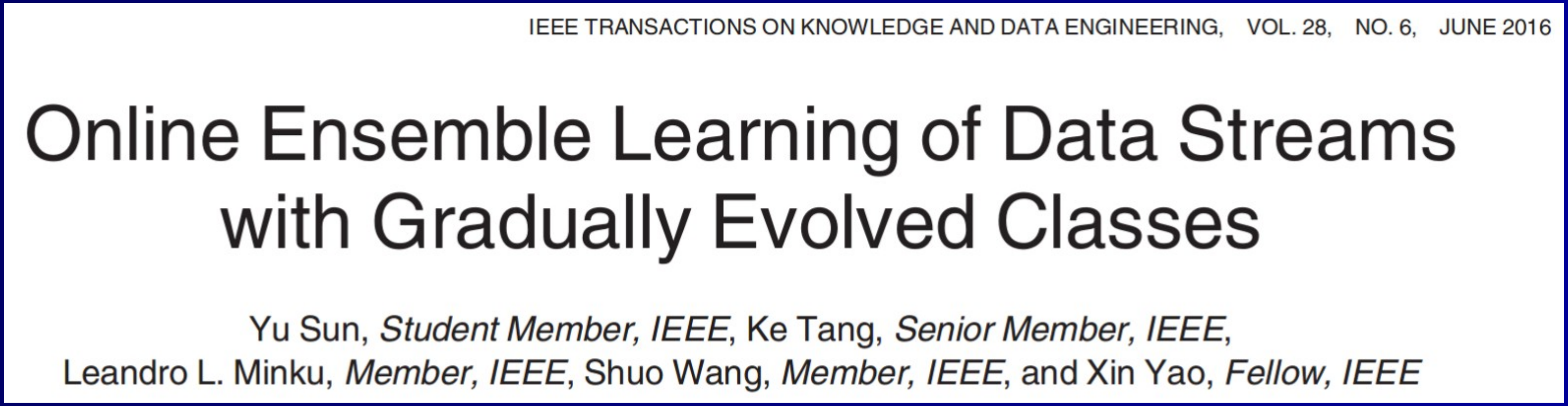
DATA STREAM CLUSTERING – DBScan ALGORITHM –



- select an arbitrary point *p*
- retrieve all points density-reachable from *p*
- if *p* is a core point, a cluster is formed
- If p is a border point
  - no points are density-reachable from p
  - DBSCAN visits the next point of the database
- Continue the process until all of the points have been processed

RELATED SCENARIOS “GRADUALLY EVOLVED CLASSES”

- Annotated stream → supervised scenario
- “Class evolution” → emergence, disappearance, re-ocurrence
- Challenge → initial class-imbalance of emergent classes

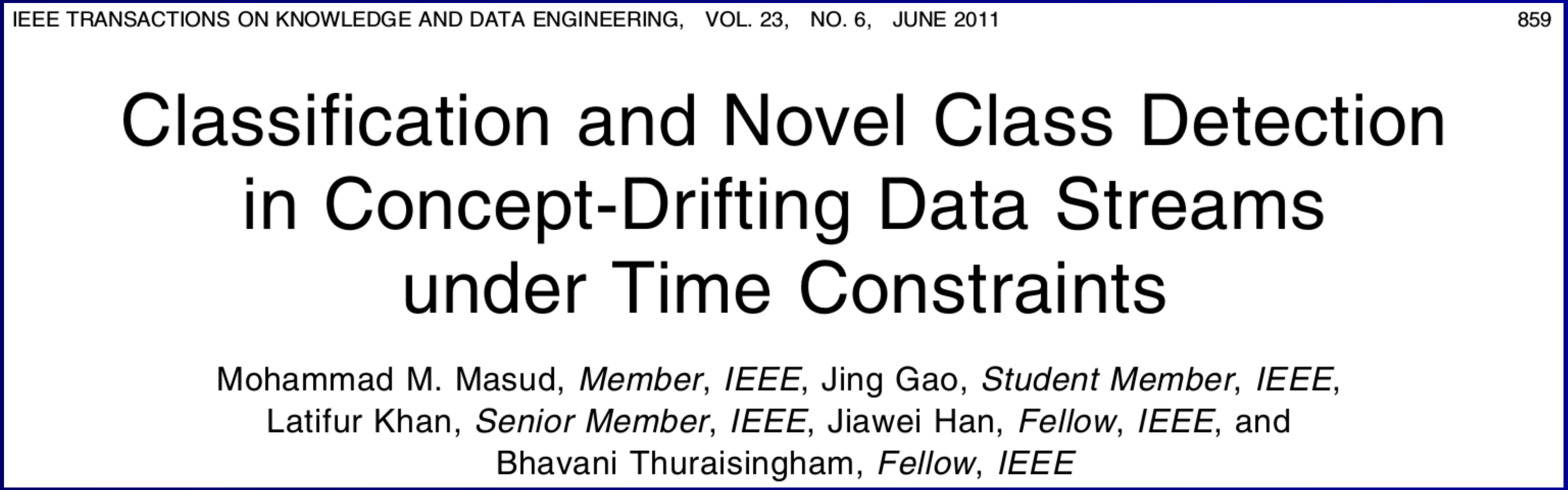

IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 28, NO. 6, JUNE 2016

Online Ensemble Learning of Data Streams with Gradually Evolved Classes

Yu Sun, *Student Member, IEEE*, Ke Tang, *Senior Member, IEEE*, Leandro L. Minku, *Member, IEEE*, Shuo Wang, *Member, IEEE*, and Xin Yao, *Fellow, IEEE*

RELATED SCENARIOS "TIME CONSTRAINTS SUPERVISION"

- Data stream-samples
- Novelty detection → the appearance of new classes
- "Real" label → available after *T* time units → update model

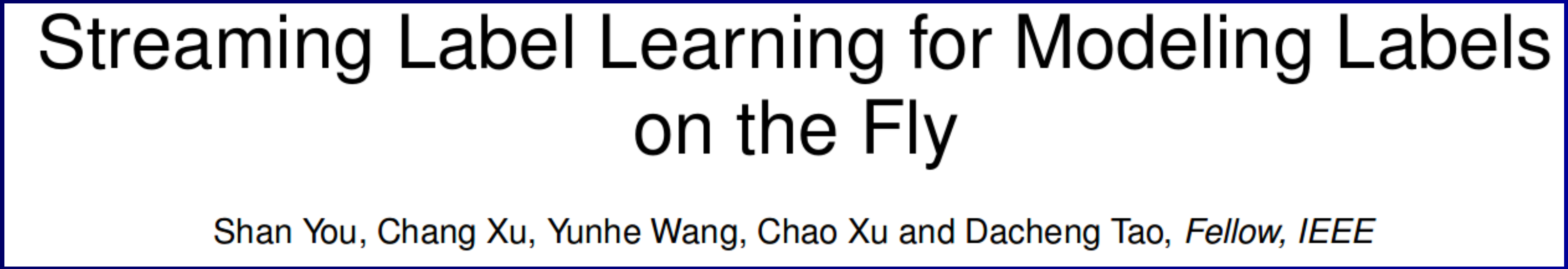

IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 6, JUNE 2011 859

Classification and Novel Class Detection in Concept-Drifting Data Streams under Time Constraints

Mohammad M. Masud, *Member*, *IEEE*, Jing Gao, *Student Member*, *IEEE*, Latifur Khan, *Senior Member*, *IEEE*, Jiawei Han, *Fellow*, *IEEE*, and Bhavani Thuraisingham, *Fellow*, *IEEE*

RELATED SCENARIOS “STREAMING LABEL LEARNING”

- Set of training samples → fixed
- Set of labels → augmented in streaming way
- Update predictive model → without re-training
- Settle “old + new” labels


Streaming Label Learning for Modeling Labels on the Fly

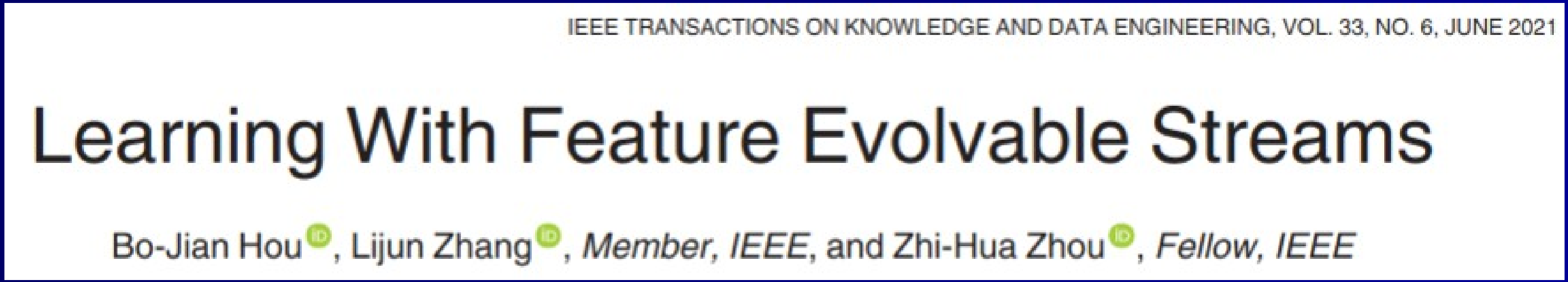
Shan You, Chang Xu, Yunhe Wang, Chao Xu and Dacheng Tao, *Fellow, IEEE*

RELATED SCENARIOS “FEATURE EVOLVABLE STREAMS”

- **Streaming data-instances**

- **Initial features substituted**
- **Learn a mapping: $\text{Set}_{\text{NEW}} \rightarrow \text{Set}_{\text{OLD}}$**

- **Continue using old-features’ model**

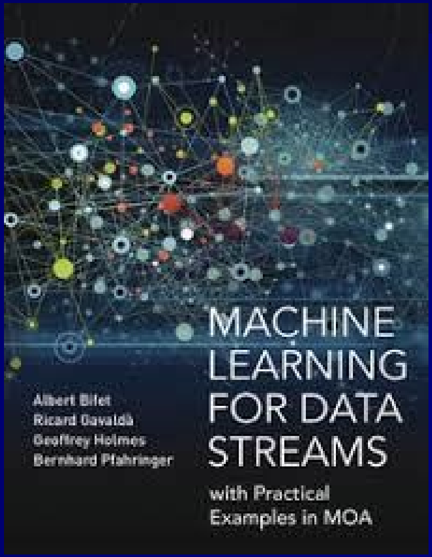


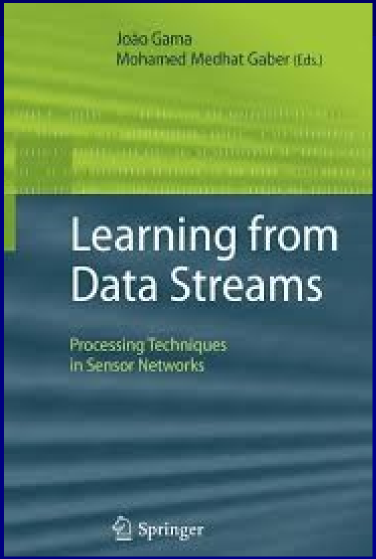
IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 33, NO. 6, JUNE 2021

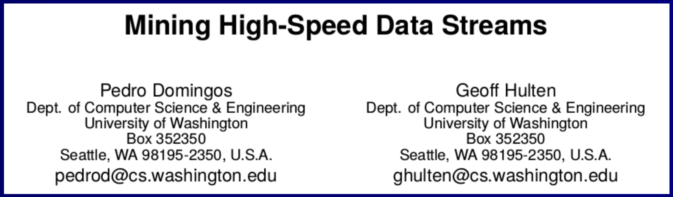
Learning With Feature Evolvable Streams

Bo-Jian Hou, Lijun Zhang, *Member, IEEE*, and Zhi-Hua Zhou, *Fellow, IEEE*

REFERENCES + SOFTWARE


Albert Bifet
Ricard Gavaldà
Geoffrey Holmes
Bernhard Pfahringer
MACHINE LEARNING FOR DATA STREAMS
with Practical Examples in MOA

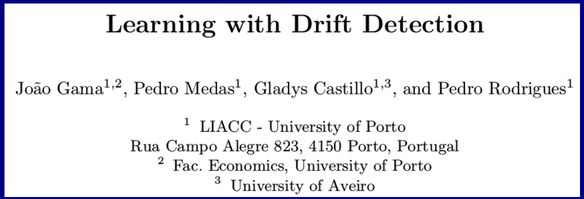

João Gama
Mohamed Medhat Gaber (Eds.)
Learning from Data Streams
Processing Techniques in Sensor Networks
Springer



Mining High-Speed Data Streams

Pedro Domingos
Dept. of Computer Science & Engineering
University of Washington
Box 352350
Seattle, WA 98195-2350, U.S.A.
pedrod@cs.washington.edu

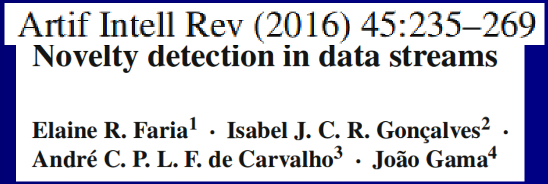
Geoff Hulten
Dept. of Computer Science & Engineering
University of Washington
Box 352350
Seattle, WA 98195-2350, U.S.A.
ghulten@cs.washington.edu



Learning with Drift Detection

João Gama[1,2], Pedro Medas[1], Gladys Castillo[1,3], and Pedro Rodrigues[1]

[1] LIACC - University of Porto
Rua Campo Alegre 823, 4150 Porto, Portugal
[2] Fac. Economics, University of Porto
[3] University of Aveiro



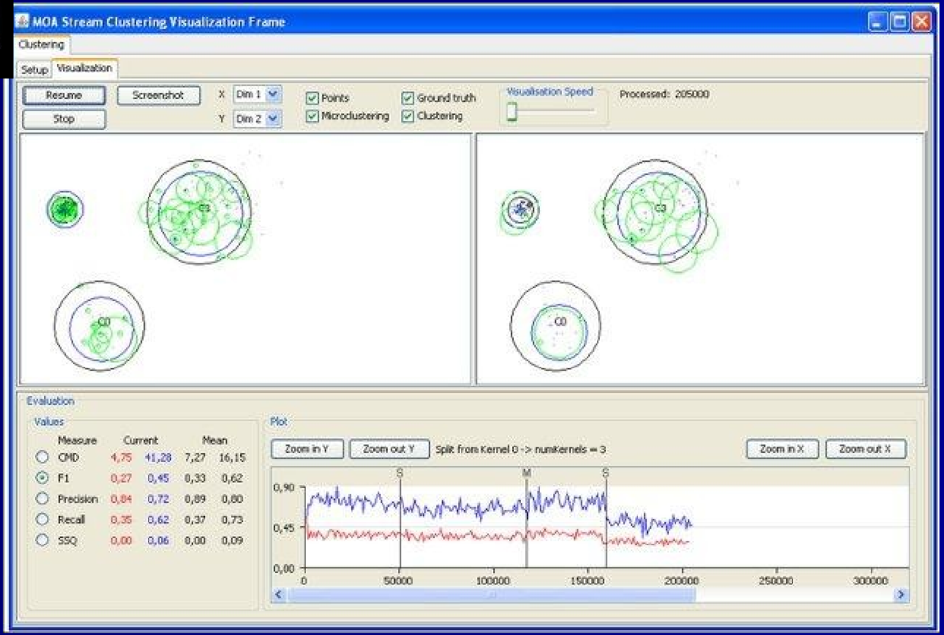
Artif Intell Rev (2016) 45:235–269

Novelty detection in data streams

Elaine R. Faria[1] · Isabel J. C. R. Gonçalves[2] · André C. P. L. F. de Carvalho[3] · João Gama[4]

REFERENCES + SOFTWARE

moa



X Dim 1
Y Dim 2
☑ Points ☑ Ground truth
☑ Microclustering ☑ Clustering
Processed: 205000

	Measure	Current		Mean	
○	CMD	4,75	41,28	7,27	16,15
●	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

Split from Kernel 0 -> numKernels = 3

RDocumentation

stream

River

A Python package for online/streaming machine learning.