Compressed k-Nearest Neighbors Classification for Evolving Data Streams

Maroua Bahri[1], Albert Bifet[1,2], Silviu Maniu[3], Rodrigo F. de Mello[4], Nikolaos Tziortziotis[5]

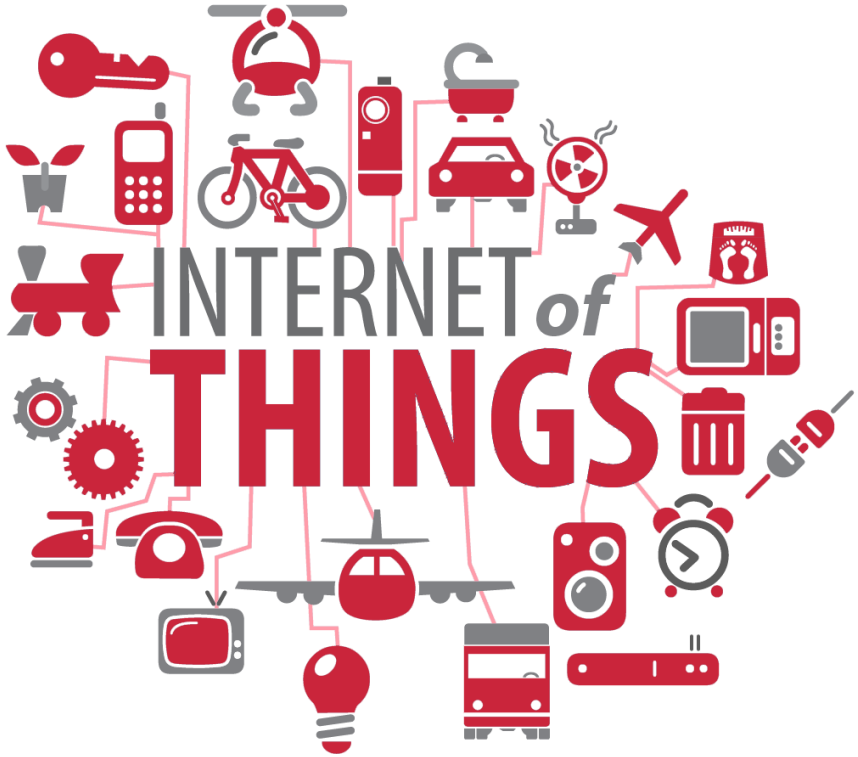
[1]LTCI, Télécom Paris, [2]University of Waikato, [3]LRI, Université Paris-Sud, [4]University of São Paulo, [5]Tradelab

Outline

TELECOM Paris

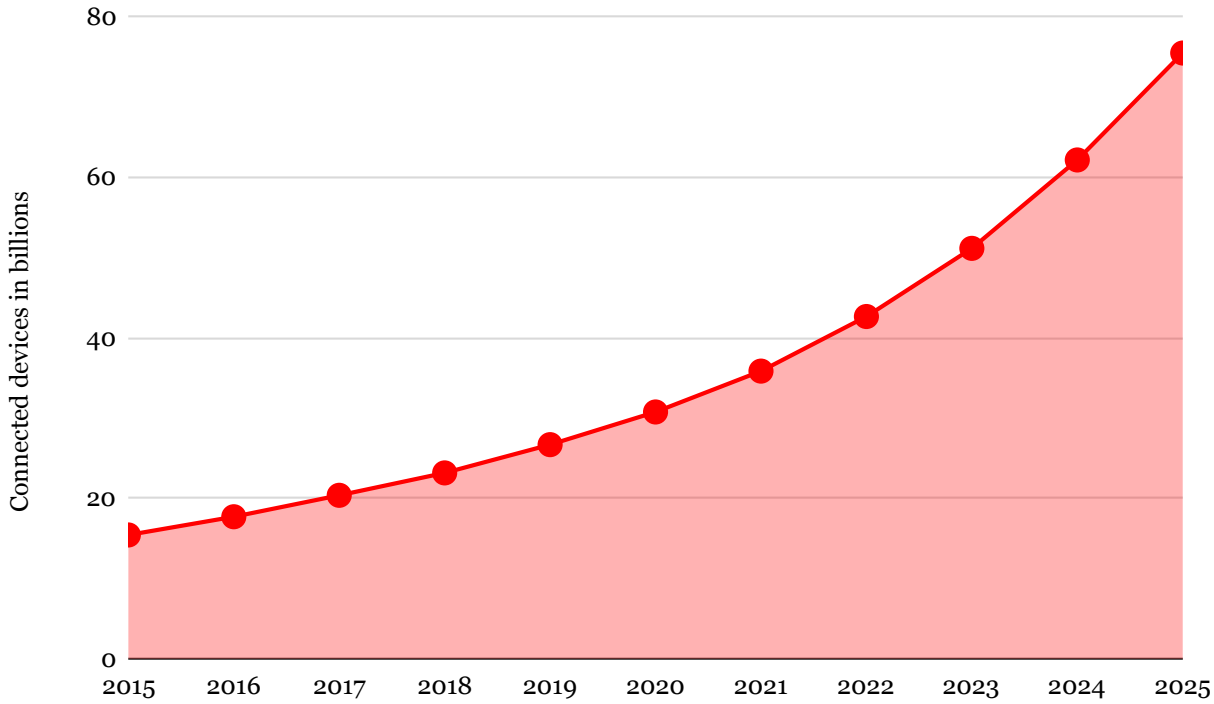
Outline

Internet of Things (IoT)



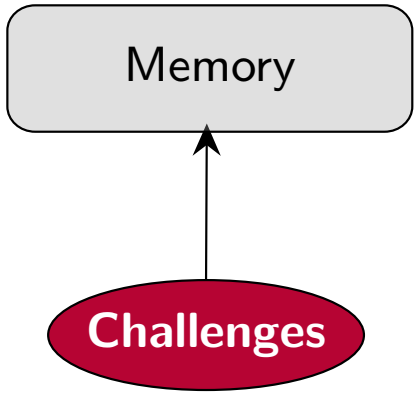
- Network of connected devices

Internet of Things (IoT)



- Statista predicts around 80 billion IoT devices by 2025

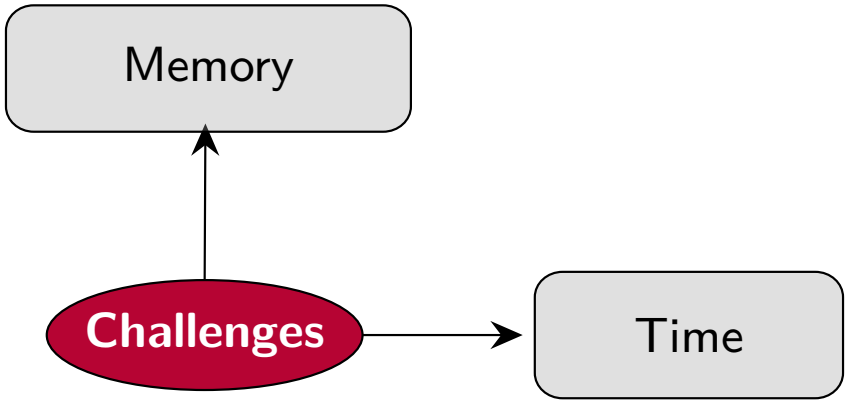
IoT Stream Mining



Memory

- Use a limited amount of memory

IoT Stream Mining



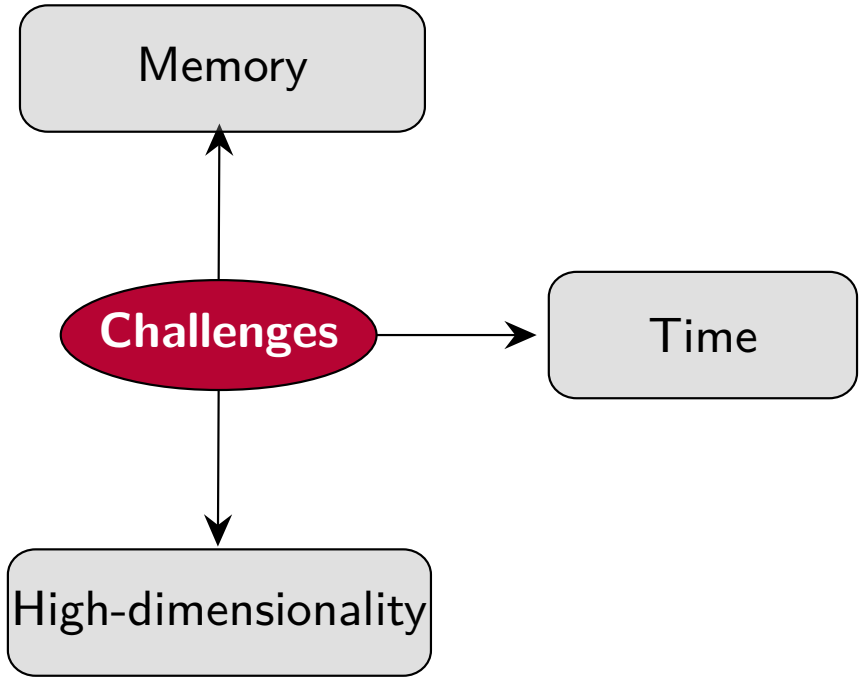
Memory

- Use a limited amount of memory

Time

- Work in a limited amount of time

IoT Stream Mining



Memory

- Use a limited amount of memory

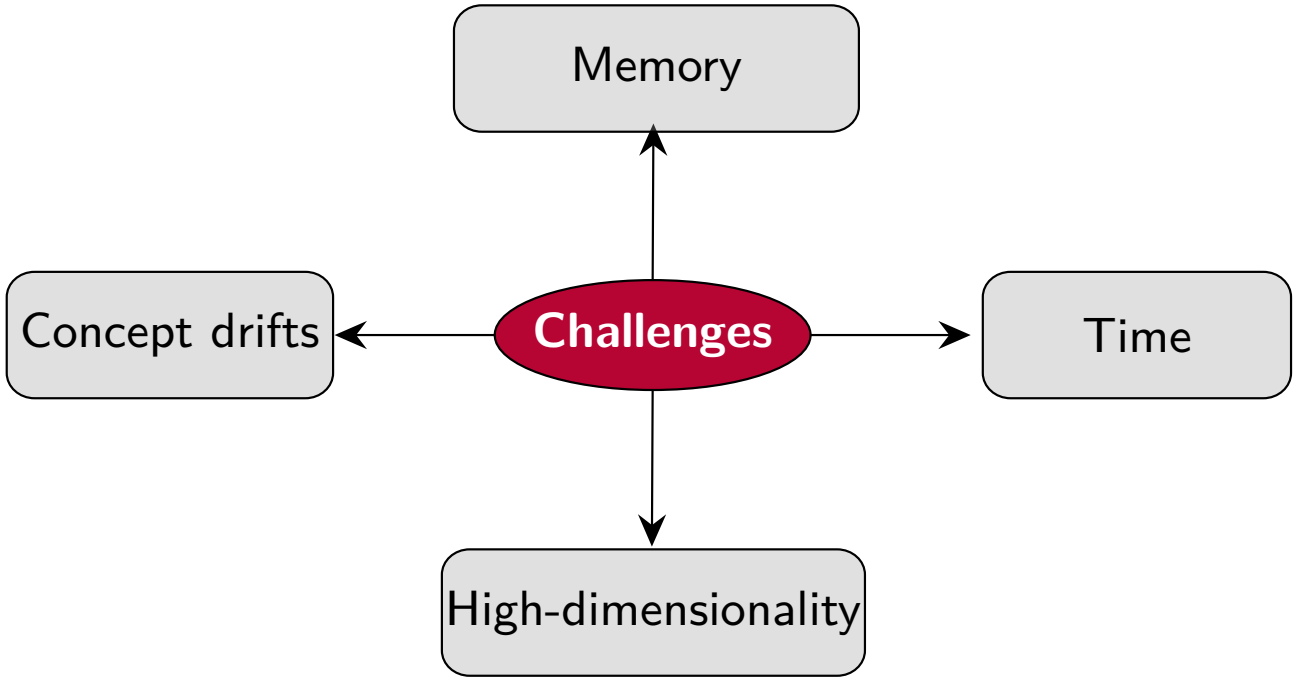
Time

- Work in a limited amount of time

Dimensionality

- Handle high-dimensional data

IoT Stream Mining



Memory

- Use a limited amount of memory

Time

- Work in a limited amount of time

Dimensionality

- Handle high-dimensional data

Concept drifts

- Detect and adapt to changes

IoT Stream Mining

- Incorporate data on the fly
- Single pass, one instance at a time
- Once processed, it is discarded or archived
- Be ready to predict at any instance

Objectives

The available algorithms have significant shortcomings :

- Some are efficient, but do not produce accurate model
- Some produce accurate model, but inefficient
- Costly with high-dimensional data

Objectives

The available algorithms have significant shortcomings :

- Some are efficient, but do not produce accurate model
- Some produce accurate model, but inefficient
- Costly with high-dimensional data

Objectives

- Improve stream algorithms performance
- Guarantee a good precision
- Tradeoff between resources and accuracy

Objectives

The available algorithms have significant shortcomings :

- Some are efficient, but do not produce accurate model
- Some produce accurate model, but inefficient
- Costly with high-dimensional data

Objectives

- Improve stream algorithms performance
- Guarantee a good precision
- Tradeoff between resources and accuracy

$\Rightarrow$ **Sampling, sketching, dimensionality reduction,** $\cdots$

Outline

k-NN Algorithm

- Uses a sliding window as a search space
- Given an unclassified instance X_i from a stream S :
  - Determines the kNN inside the window
  - Predicts the most frequent label

k-NN Algorithm

- Uses a sliding window as a search space
- Given an unclassified instance X_i from a stream S :
  - Determines the kNN inside the window
  - Predicts the most frequent label

- Inefficient at prediction time
- Memory consuming
- Inefficient with high-dimensional data

k-NN Algorithm

- Uses a sliding window as a search space
- Given an unclassified instance X_i from a stream S :
  - Determines the kNN inside the window
  - Predicts the most frequent label

- Inefficient at prediction time
- Memory consuming
- Inefficient with high-dimensional data

$\Rightarrow$ **Dimensionality reduction**

Dimensionality Reduction (DR)

The projection of high-dimensional data into a low-dimensional space by reducing the input features

Objectif : given an instance $X_i \in \mathbb{R}^a$, we wish to obtain $Y_i \in \mathbb{R}^m$, where $m \ll a$

- Principal Component Analysis (PCA)
- Random Projection (RP)
- Compressed Sensing (CS)
- Hashing Trick (HT)

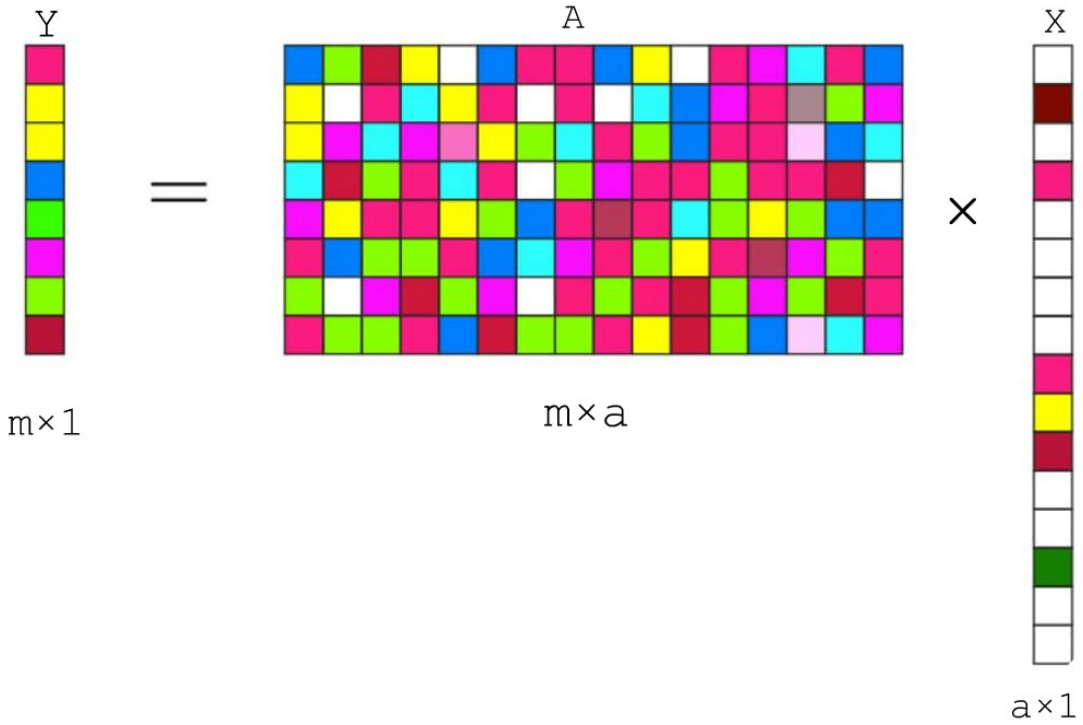
Dimensionality Reduction (DR)

The projection of high-dimensional data into a low-dimensional space by reducing the input features

Objectif : given an instance $X_i \in \mathbb{R}^a$, we wish to obtain $Y_i \in \mathbb{R}^m$, where $m \ll a$

- Principal Component Analysis (PCA)
- Random Projection (RP)
- **Compressed Sensing (CS)**
- Hashing Trick (HT)

Compressed Sensing (CS)



- Data compression method that transforms and reconstructs data from few samples with h.p
- Matrix A used to transform instances from $\mathbb{R}^a \rightarrow \mathbb{R}^m, m \ll a$
  - Fourier transform, random matrices (e.g., Bernoulli, Gaussian)

Donoho, Compressed sensing, 2006.

Compressed Sensing (CS)

CS relies on two principles :

- ***Sparsity*** : expresses the idea that data may be much smaller and are compressible. X is s-sparse if $\|X\|_0 \leq s$

Compressed Sensing (CS)

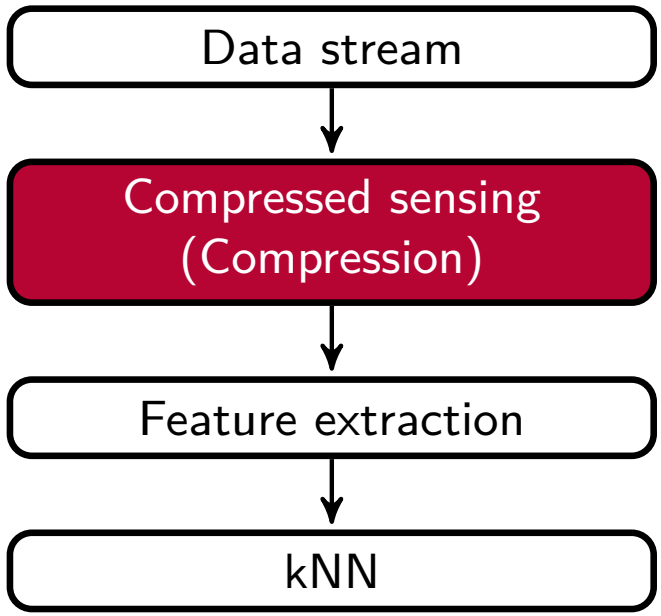
CS relies on two principles :

- ***Sparsity*** : expresses the idea that data may be much smaller and are compressible. X is s-sparse if $\|X\|_0 \leq s$
- ***Restricted Isometry Property (RIP)*** : A satisfies RIP $\forall$ s-sparse instance $X \in \mathbb{R}^a$, if there exists $\epsilon \in [0, 1]$:

$$(1 - \epsilon)\|X\|_2^2 \leq \|AX\|_2^2 \leq (1 + \epsilon)\|X\|_2^2$$

A satisfies the RIP with high probability if $m \geq (s \log(a))$

Compressed Sensing kNN (CS-kNN)



CS-kNN Algorithm

Algorithm Compressed-kNN. **Symbols :** $S = \{X_1, X_2, \ldots\} \in \mathbb{R}^a$: stream ; $S' = \{Y_1, Y_2, \ldots\} \in \mathbb{R}^m$: transformed data ; C : set of labels ; w : window ; k : number of neighbors.

1: **function** CS-kNN(S, k, m)
2: Init $w \leftarrow \emptyset$
3: **for all** $X_i \in S$ **do**
4: $Y_i \leftarrow \mathrm{CS}(X_i, m)$ ▷ apply CS
5: **for all** $Y_j \in w$ **do**
6: compute $D_{Y_j}(Y_i)$
7: $c \leftarrow \mathrm{Predict}_{c \in C} D_{w,k}(Y_i)$

CS-kNN : Theoretical Guarantees

The distance between two instances X_i and X_j is defined as follows :

$$D_{X_j}(X_i) = \sqrt{\|X_i - X_j\|^2}$$

Similarly, the k-nearest neighbors distance is defined as :

$$D_{w,k}(X_i) = \min_{\binom{w}{k}, X_j \in w} D_{X_j}(X_i)$$

$\binom{w}{k}$ denotes the subset of w of size k

CS-kNN : Theoretical Guarantees

The distance between two instances X_i and X_j is defined as follows :

$$D_{X_j}(X_i) = \sqrt{\|X_i - X_j\|^2}$$

Similarly, the k-nearest neighbors distance is defined as :

$$D_{w,k}(X_i) = \min_{\binom{w}{k}, X_j \in w} D_{X_j}(X_i)$$

$\binom{w}{k}$ denotes the subset of w of size k

Theorem

Given a stream $S = \{X_i\}$ and $\epsilon \in [0, 1]$, if there exists a transformation matrix $A : \mathbb{R}^a \rightarrow \mathbb{R}^m$ having the RIP, such that $m = \mathcal{O}(s \log(a))$, where s is the sparsity of data, then $\forall X_i \in w$:

$$(1 - \epsilon) D^2_{w,k}(X) \leq D^2_{w,k}(AX) \leq (1 + \epsilon) D^2_{w,k}(X)$$

$\rightarrow$ CS-kNN captures the neighborhood up to some ϵ-divergence

Outline

MOA

Massive **O**nline **A**nalysis [1] is a framework for online learning from data streams

- It is written in Java
- It includes tools for evaluation and a collection of machine learning algorithms for data streams
- It is an easily extendable framework for new streams, algorithms and evaluation methods

[1] https://moa.cms.waikato.ac.nz/

Results

Accuracy (%)

Dataset	CS-kNN	HT-kNN	PCA-kNN	kNN
Tweet_1	78.82	73.77	**80.43**	79.80
Tweet_2	78.13	73.02	**80.06**	79.20
Tweet_3	76.75	72.40	**81.93**	78.86
RBF	98.90	19.20	**99.00**	98.89
CNAE	70.00	65.00	**75.83**	73.33
Enron	96.02	95.76	94.59	**96.18**
IMDB	69.86	69.65	70.57	**70.94**
Spam	85.39	83.82	**96.00**	81.17
Covt	91.36	77.18	91.55	**91.67**
Overall ∅	82.80	69.98	**85.55**	83.34

Results

Memory (MB)

Dataset	CS-kNN	HT-kNN	PCA-kNN	kNN
Tweet_1	**2.52**	**2.52**	3.03	34.64
Tweet_2	**2.52**	**2.52**	5.97	70.97
Tweet_3	**2.52**	**2.52**	8.84	103.19
RBF	**2.52**	**2.52**	8.86	13.18
CNAE	**2.52**	**2.52**	3.09	61.37
Enron	**2.52**	**2.52**	3.51	70.60
IMDB	**2.52**	**2.52**	8.81	70.65
Spam	**2.52**	**2.52**	245.22	1476.11
Covt	**2.52**	**2.52**	3.02	3.47
Overall ∅	**2.52**	**2.52**	32.26	211.57

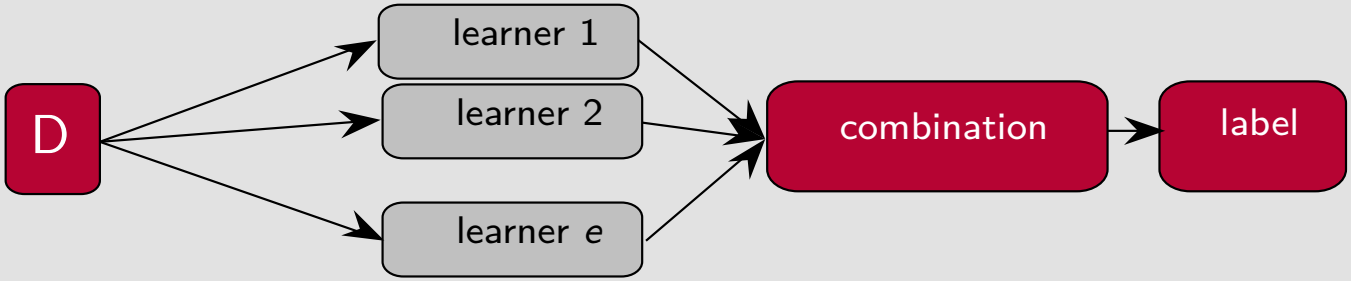
Results

Time (sec)

Dataset	CS-kNN	HT-kNN	PCA-kNN	kNN
Tweet_1	**62.55**	93.24	622.65	1198.78
Tweet_2	**107.48**	120.83	705.71	2029.82
Tweet_3	**126.73**	154.22	988.25	2864.55
RBF	**59.47**	168.31	243.26	284.34
CNAE	**0.87**	0.95	3.97	32.19
Enron	**1.58**	1.81	7.21	86.08
IMDB	**95.62**	125.62	1686.88	7892.96
Spam	**159.92**	194.07	11329.91	34231.45
Covt	**30.94**	88.17	161.00	252.69
Overall ∅	**71.68**	105.25	1749.87	5430.32

Ensemble CS-kNN (CSB)

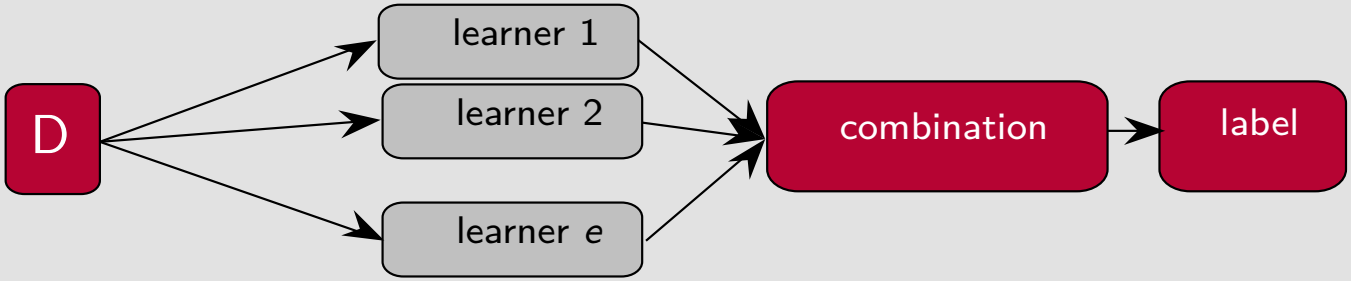
Ensemble-based method



Bifet, *et al.*, Leveraging bagging for evolving data streams, 2010.

Ensemble CS-kNN (CSB)

Ensemble-based method

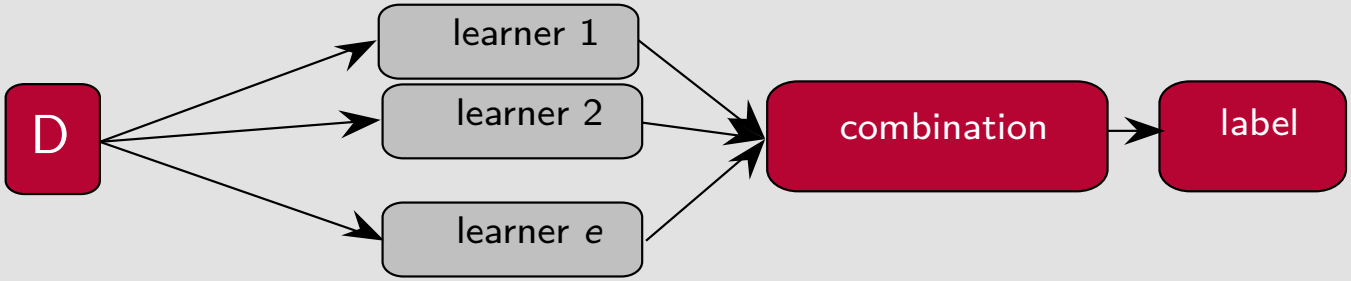


- Uses CS-kNN as a base learner under Leveraging Bagging (LB)
- Uses several random matrices : one for each ensemble member
- Preserves the neighborhood properties of the CS-kNN

Bifet, *et al.*, Leveraging bagging for evolving data streams, 2010.

Ensemble CS-kNN (CSB)

Ensemble-based method



- Uses CS-kNN as a base learner under Leveraging Bagging (LB)
- Uses several random matrices : one for each ensemble member
- Preserves the neighborhood properties of the CS-kNN

⊕ Good accuracy

⊖ Computational resources

Bifet, *et al.*, Leveraging bagging for evolving data streams, 2010.

Outline

Conclusions

- CS-kNN algorithm
  - Reduces the resource usage
  - Preserves the neighborhood
- CSB-kNN ensemble method
  - Improves accuracy
  - Is slow
- Open source contribution :

`https://github.com/marouabahri/CS-kNN`

Thank you

This work was done in the context of IoT Analytics
DigiCosme Project and was funded by Labex DigiCosme.

Compressed k-Nearest Neighbors Classification for Evolving Data Streams

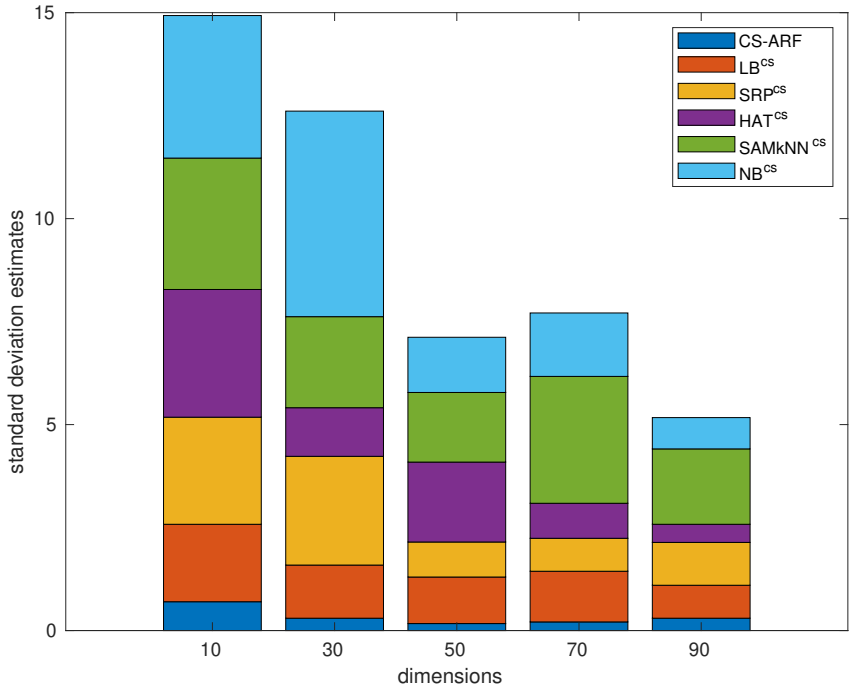
Maroua Bahri[1], Albert Bifet[1,2], Silviu Maniu[3], Rodrigo F. de Mello[4], Nikolaos Tziortziotis[5]

[1]LTCI, Télécom Paris, [2]University of Waikato, [3]LRI, Université Paris-Sud, [4]University of São Paulo, [5]Tradelab

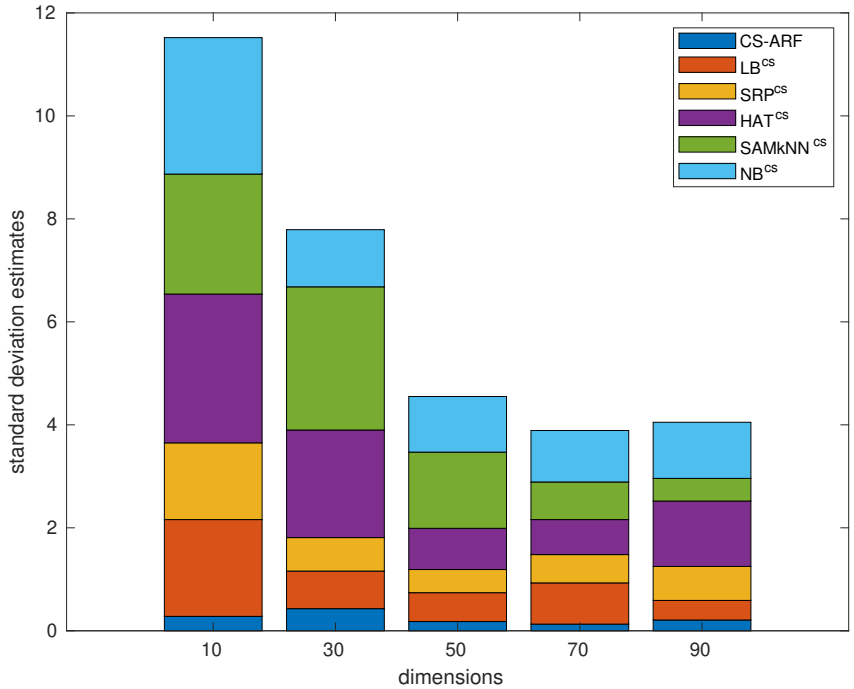
Overview of the data

Dataset	#Instances	#Attributes	#Classes	Type
Tweets_1	1,000,000	500	2	Synthetic
Tweets_2	1,000,000	1,000	2	Synthetic
Tweets_3	1,000,000	1,500	2	Synthetic
RBF	1,000,000	200	10	Synthetic
CNAE	1,080	856	9	Real
Enron	1,702	1,000	2	Real
IMDB	120,919	1,001	2	Real
Spam	9,324	39,916	2	Real
Covt	581,012	54	7	Real

Results : Standard Deviation

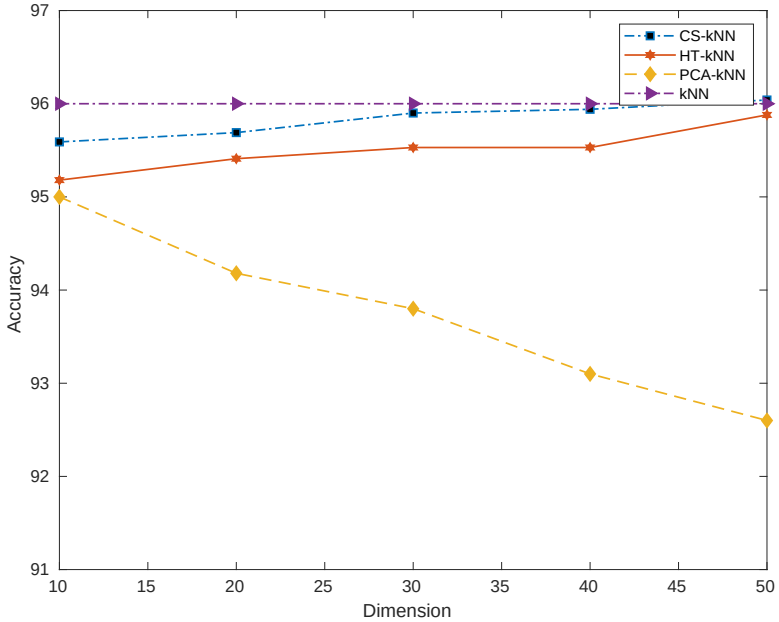


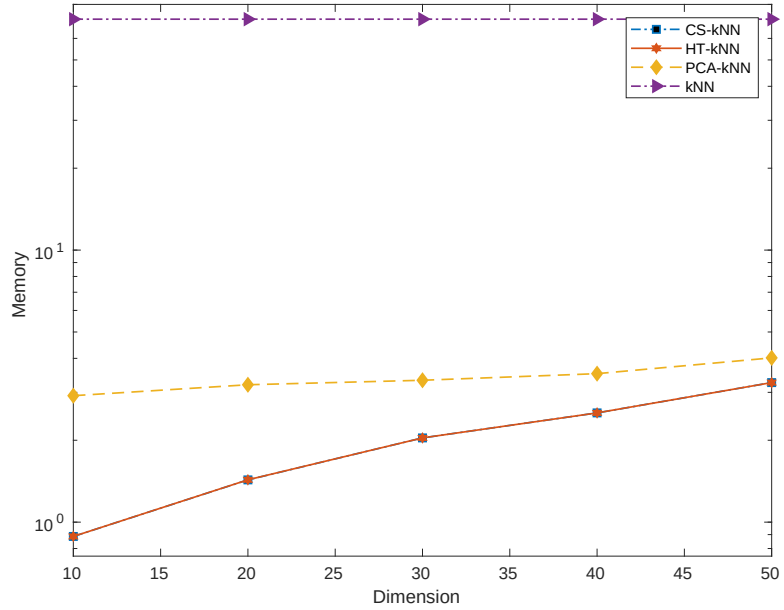
(a) Tweet_3

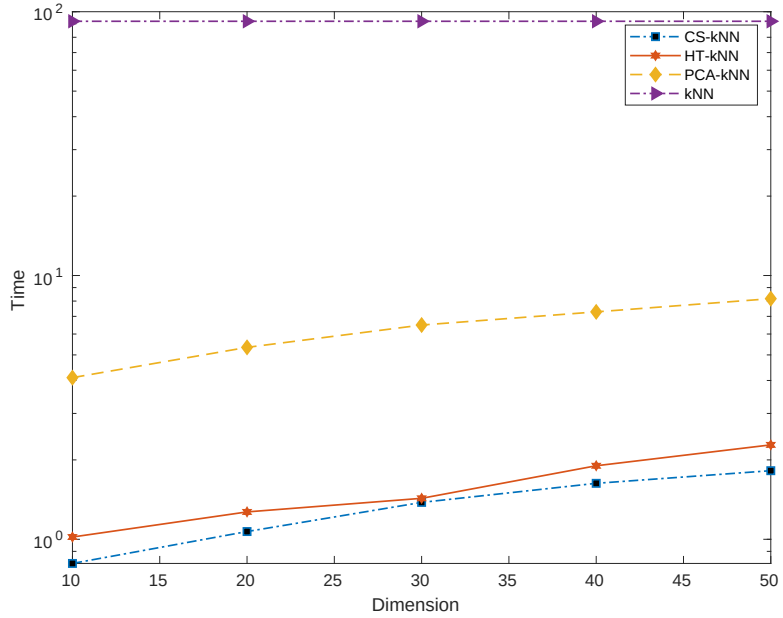


(b) Har

Results : CS-kNN with Enron Dataset

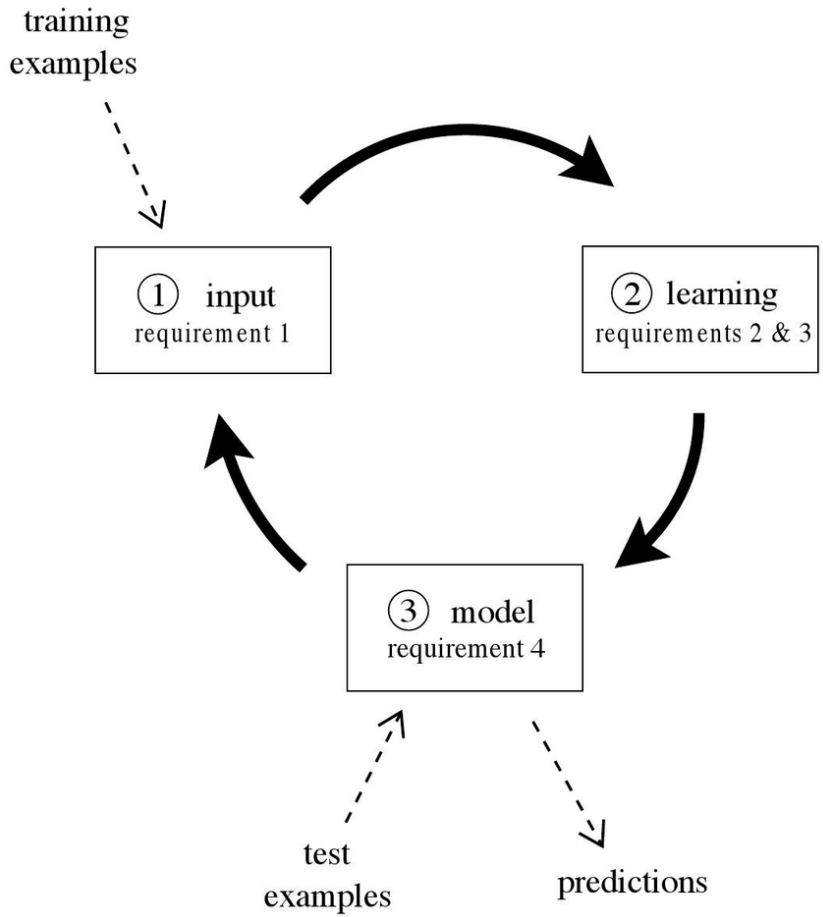






Classification

1. Process an instance at a time
2. Use a limited amount of memory
3. Work in a limited amount of time
4. Be ready to predict at any instance



Bifet, et al., Data stream mining a practical approach, 2009.