

# Data Stream Mining

Nida Meddouri<sup>1</sup>

nida.meddouri@emse.fr

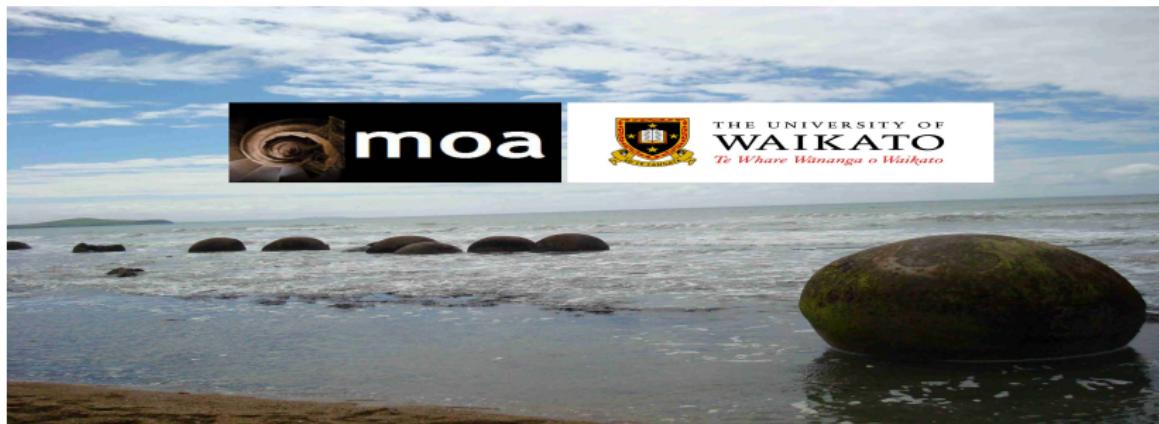
<sup>1</sup>École Nationale Supérieure des Mines de Saint-Etienne  
29 Rue des Frères Ponchardier 42 023 Saint-Etienne.

Novembre 2020

# Why **YOU** should care about Stream Mining

Version préliminaire, merci de signaler toute coquille ou erreur.

Pourquoi le Stream Mining est important ?  
En quoi est-il différent du Machine Learning ?  
Cinq défis en Stream Mining  
Hypothèse de l'IDI  
Trois approches algorithmiques standards  
Tout est une approximation



# DATA STREAM MINING

## A Practical Approach

Albert Bifet, Geoff Holmes, Richard Kirkby and  
Bernhard Pfahringer

May 2011



Pourquoi le Stream Mining est important ?  
En quoi est-il différent du Machine Learning ?  
Cinq défis en Stream Mining  
Hypothèse de l'IDI  
Trois approches algorithmiques standards  
Tout est une approximation



# Massive Online Analysis Manual

Albert Bifet, Richard Kirkby,  
Philipp Kranen, Peter Reutemann

March 2012



## Plan :

- 1 Pourquoi le Stream Mining est important ?
- 2 En quoi est-il différent du Machine Learning ?
- 3 Cinq défis en Stream Mining
- 4 Hypothèse de l'IDI
  - 2 "big" questions
  - Une hypothèse fondamentale en Batch Machine Learning
  - Le monde réel n'est pas I.D.I.
- 5 Trois approches algorithmiques standards
- 6 Tout est une approximation
  - Échantillonnage de données
  - Sliding Window
  - Counting in  $\log(N)$  bits
  - Count-Min Sketch
  - Frequent algorithm

# Les flux de données sont partout

- **SGBD traditionnel** : données stockées dans des ensembles de données **finis et persistants**.
- **Flux de données** : **distribués, continus, illimités, rapides, variables dans le temps, bruyants, ...**



## Data Stream Management :

- Surveillance du réseau et ingénierie du trafic.
- Réseaux de capteurs.
- Enregistrements des détails des appels de télécommunication.
- Sécurité Internet.
- Applications financières.
- Processus de manufacture.
- Les journaux Web et les flux de clics.

## (Massive) Data Streams

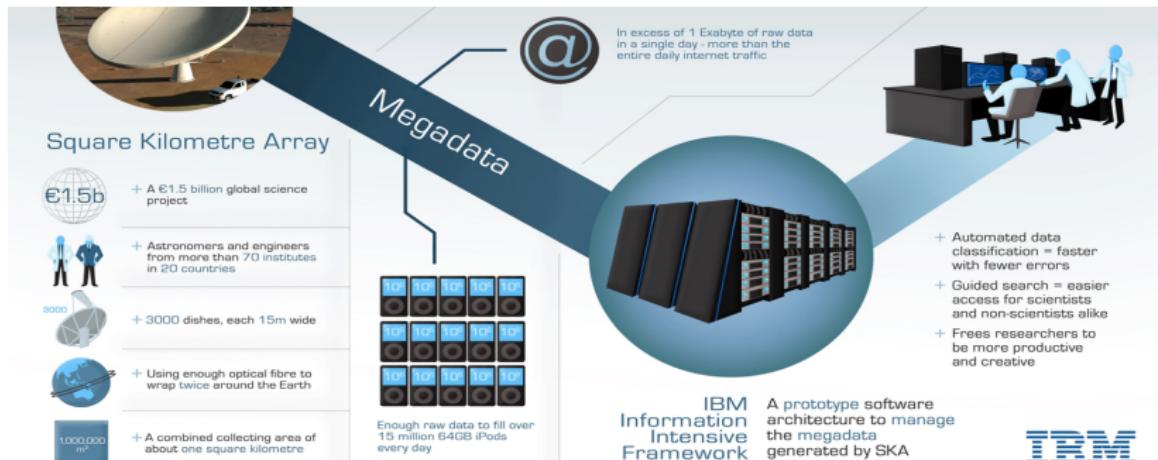
**Les données augmentent continuellement plus vite que notre capacité à les stocker ou à les indexer.**

- **Données scientifiques** : les satellites d'observation de la NASA génèrent des milliards de lectures chaque jour
- **IP Network Traffic** : jusqu'à  $3 \times 10^9$  de paquets par heure et par routeur. Chaque FAI dispose de plusieurs de routeurs.
- **Séquences génomiques** complètes pour de nombreuses espèces maintenant disponible.



# Data Stream Processing

- Recherche scientifique (e.g. surveillance de l'environnement, ...).
- System Management (e.g. détection des pannes, crashes, pannes, ...).
- Business intelligence (e.g. marketing, nouvelles offres, ...).
- Protection des revenus (e.g. fraude téléphonique, abus de service, ..).



## Quelques systèmes actuels

- moa.cs.waikato.ac.nz
  - samoa-project.net
  - spark.apache.org/streaming
  - lambda-architecture.net
- 
- R's stream package (clustering only)  
(plus /r/streamMoa package)
- 
- RapidMiner streams plugin
  - Weka's UpdateableClassifier interface

Pourquoi le Stream Mining est important ?

En quoi est-il différent du Machine Learning ?

Cinq défis en Stream Mining

Hypothèse de l'IDI

Trois approches algorithmiques standards

Tout est une approximation

## Demo : Weka's UpdateableClassifier interface



# J'ai triché un peu ;)

The image shows two error dialog boxes from the Weka software. The left dialog, titled "OutOfMemory", contains a yellow warning icon and text about memory usage. The right dialog, titled "Instances", contains a red error icon and text about a Java exception.

**OutOfMemory**

Not enough memory (less than 50MB left on heap). Please load a smaller dataset or use a larger heap size.

- initial heap size: 128MB
- current memory (heap) used: 1968MB
- max<sup>up</sup> memory (heap) available: 2014MB

Note:

The Java heap size can be specified with the -Xmx option.  
E.g., to use 128MB as heap size, the command line looks like this:  
java -Xmx128m -classpath ...

This does NOT work in the SimpleCLI, the above java command refers to the one with which Weka is started. See the Weka FAQ on the web for further info.

**Instances**

Problem setting base instances:  
java.lang.reflect.InvocationTargetException

OK

## Plan :

- 1 Pourquoi le Stream Mining est important ?
- 2 En quoi est-il différent du Machine Learning ?
- 3 Cinq défis en Stream Mining
- 4 Hypothèse de l'IDI
  - 2 "big" questions
  - Une hypothèse fondamentale en Batch Machine Learning
  - Le monde réel n'est pas I.D.I.
- 5 Trois approches algorithmiques standards
- 6 Tout est une approximation
  - Échantillonnage de données
  - Sliding Window
  - Counting in  $\log(N)$  bits
  - Count-Min Sketch
  - Frequent algorithm

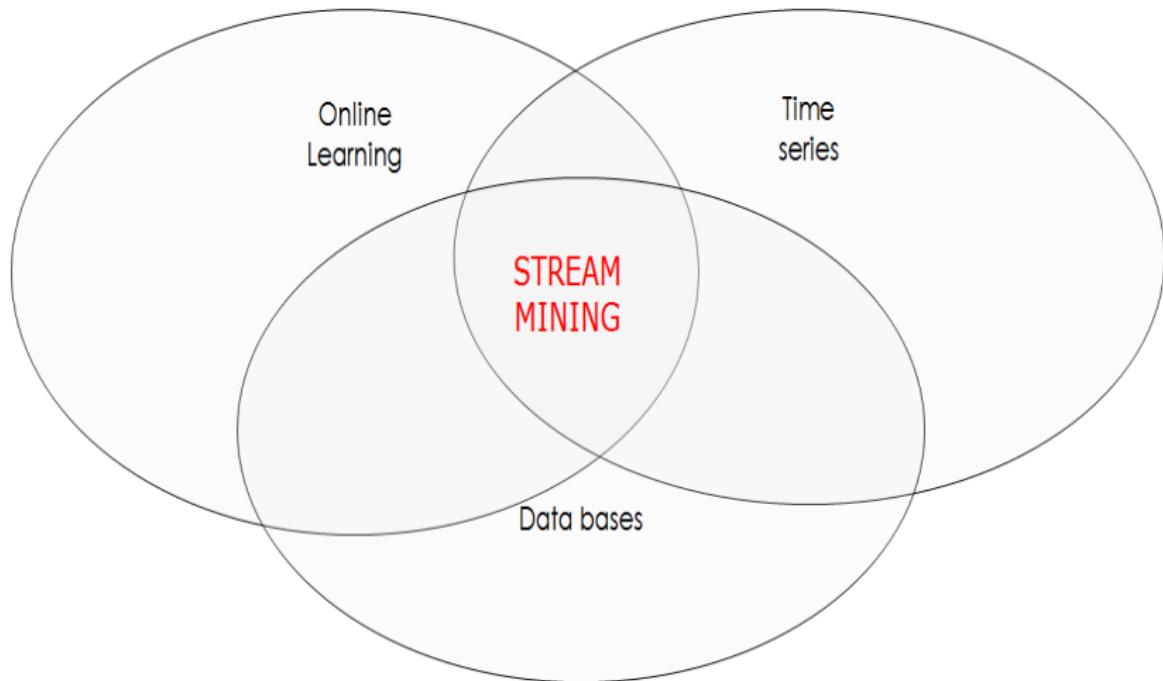


FIGURE 1 – C'est quoi le stream mining ?

## Plan :

- 1 Pourquoi le Stream Mining est important ?
- 2 En quoi est-il différent du Machine Learning ?
- 3 Cinq défis en Stream Mining**
- 4 Hypothèse de l'IDI
  - 2 "big" questions
  - Une hypothèse fondamentale en Batch Machine Learning
  - Le monde réel n'est pas I.D.I.
- 5 Trois approches algorithmiques standards
- 6 Tout est une approximation
  - Échantillonnage de données
  - Sliding Window
  - Counting in  $\log(N)$  bits
  - Count-Min Sketch
  - Frequent algorithm

- ➊ Traiter les instances de manière incrémentielle.
- ➋ Utiliser une quantité très limitée de mémoire et de temps pour traiter chaque instance.
- ➌ Être prêt à prédire à tout moment.
- ➍ Être capable de s'adapter au changement, car l'entrée des données (input) **n'est pas stationnaire**.
- ➎ Gérer les retours (feedback) retardés / limités.

## Plan :

- 1 Pourquoi le Stream Mining est important ?
- 2 En quoi est-il différent du Machine Learning ?
- 3 Cinq défis en Stream Mining
- 4 Hypothèse de l'IDI**
  - 2 "big" questions
  - Une hypothèse fondamentale en Batch Machine Learning
  - Le monde réel n'est pas I.D.I.
- 5 Trois approches algorithmiques standards
- 6 Tout est une approximation
  - Échantillonnage de données
  - Sliding Window
  - Counting in  $\log(N)$  bits
  - Count-Min Sketch
  - Frequent algorithm

- Votre entrée/input(x) est-elle **indépendante et distribuée de manière identique (I.D.I.)**<sup>1</sup> ?
- Vos cibles/targets (y) sont-elles **indépendantes et distribuées de manière identique (I.D.I.)** ?

---

1. independent and identically distributed (I.I.D.)

- Les données d'apprentissage et de test proviennent de la même distribution, elles sont toutes les deux I.D.I.<sup>a</sup>.
- Sinon : les évaluations sont fausses et trompeuses.

a. Indépendante et Distribuée de manière Identique

e.g. les bio-info.. contestent l'insuffisance de la Cross-Validation !

*"The experimental results reported in this paper suggest that, contrary to current conception in the community, cross-validation may play a significant role in evaluating the predictivity of (Q)SAR models."*

[Gütlein et al., 2013]

## Données des accidents de routes à New Zealand (2000-2014)

- $\approx 500\,000$  accidents.
- $\approx 200\,000$  avec "Driver 1 had a significant influence".

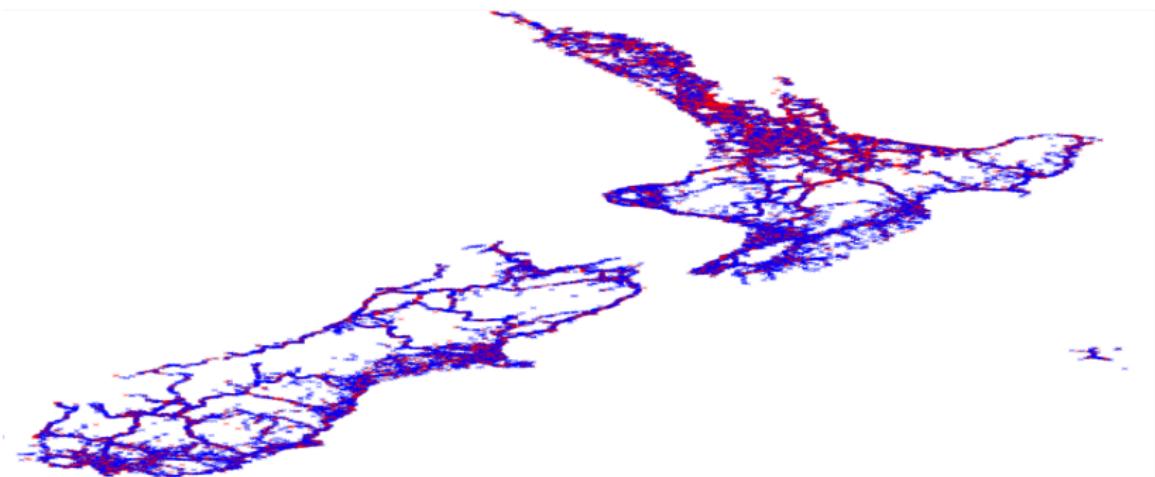


FIGURE 2 – All accidents

Pourquoi le Stream Mining est important ?  
En quoi est-il différent du Machine Learning ?  
Cinq défis en Stream Mining  
Hypothèse de l'IDI  
Trois approches algorithmiques standards  
Tout est une approximation

2 "big" questions  
Une hypothèse fondamentale en Batch Machine Learning  
Le monde réel n'est pas I.D.I.

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. In the 'Classifier' section, 'NaiveBayes' is chosen. Under 'Test options', 'Percentage split' is set to 66%. The 'Classifier output' pane displays the following results:

Time taken to test model on test split: 0.25 seconds

==== Summary ====  
Correctly Classified Instances 11090  
Incorrectly Classified Instances 4316  
Kappa statistic 0.3853  
Mean absolute error 0.3042  
Root mean squared error 0.469  
Relative absolute error 62.2143 %  
Root relative squared error 94.79 %  
Total Number of Instances 15406

==== Detailed Accuracy By Class ====  
IP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Clas.  
0.398 0.039 0.884 0.398 0.549 0.450 0.788 0.766 UP  
0.961 0.602 0.681 0.961 0.797 0.450 0.788 0.817 DOWN  
Weighted Avg. 0.720 0.361 0.768 0.720 0.691 0.450 0.788 0.795

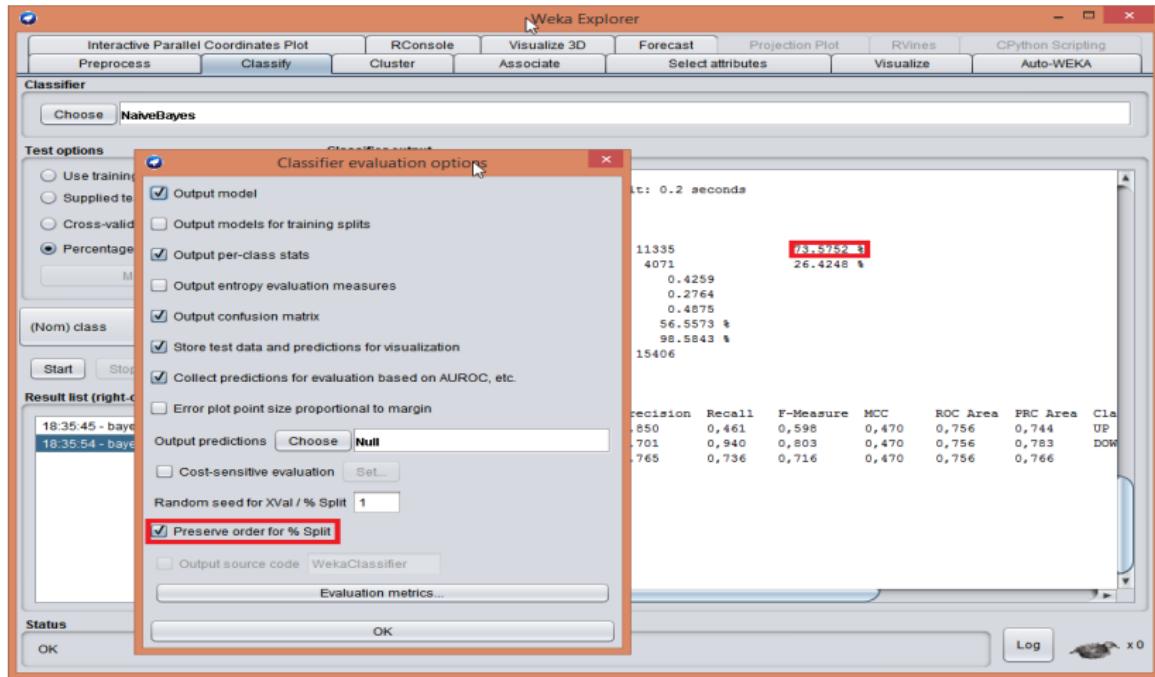
==== Confusion Matrix ====  
a b <-- classified as  
2623 3971 | a = UP  
345 8467 | b = DOWN

The 'Result list' pane shows two entries:  
18:35:45 - bayes.NaiveBayes  
18:35:54 - bayes.NaiveBayes

The 'Status' pane at the bottom left shows 'OK'. The bottom right corner features a toolbar with various icons for navigation and file operations.

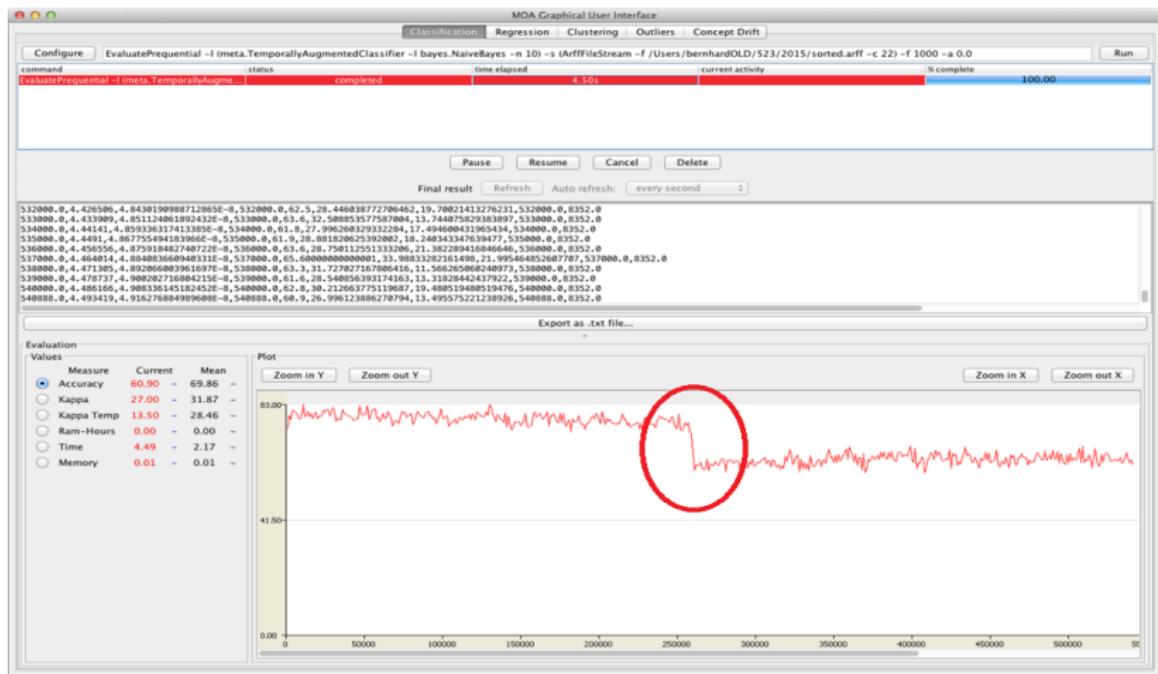
Pourquoi le Stream Mining est important ?  
En quoi est-il différent du Machine Learning ?  
Cinq défis en Stream Mining  
Hypothèse de l'IDI  
Trois approches algorithmiques standards  
Tout est une approximation

2 "big" questions  
Une hypothèse fondamentale en Batch Machine Learning  
Le monde réel n'est pas I.D.I.



Pourquoi le Stream Mining est important ?  
 En quoi est-il différent du Machine Learning ?  
 Cinq défis en Stream Mining  
 Hypothèse de l'IDI  
 Trois approches algorithmiques standards  
 Tout est une approximation

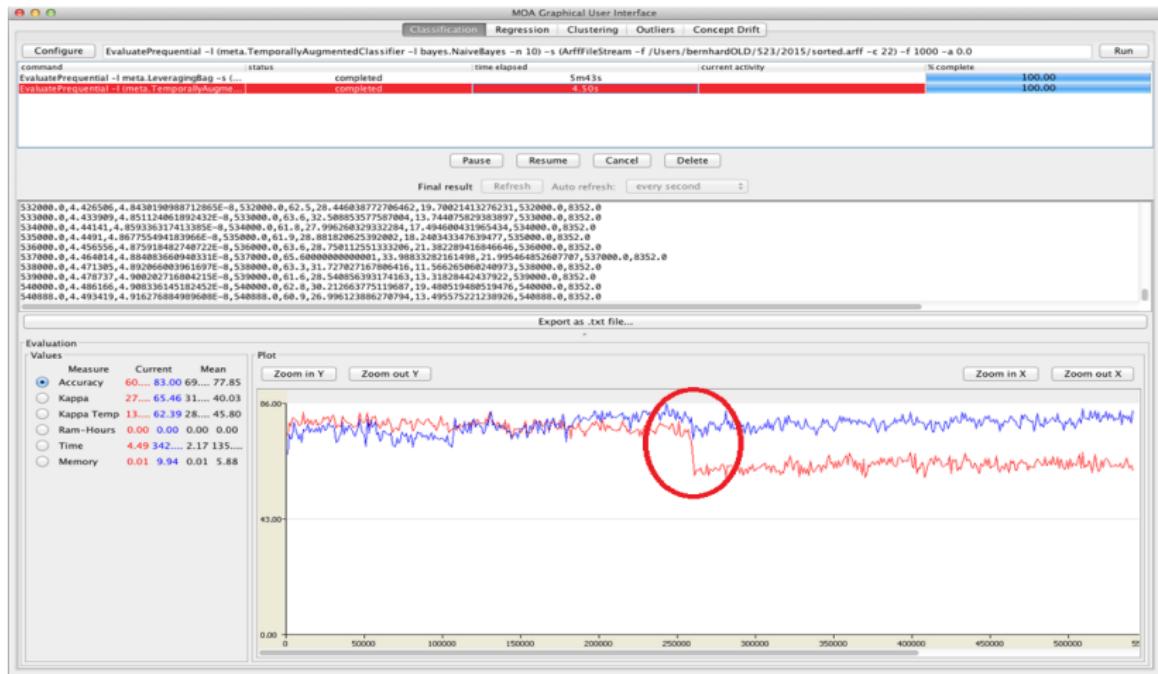
2 "big" questions  
 Une hypothèse fondamentale en Batch Machine Learning  
 Le monde réel n'est pas I.D.I.



\*

Pourquoi le Stream Mining est important ?  
 En quoi est-il différent du Machine Learning ?  
 Cinq défis en Stream Mining  
 Hypothèse de l'IDI  
 Trois approches algorithmiques standards  
 Tout est une approximation

2 "big" questions  
 Une hypothèse fondamentale en Batch Machine Learning  
 Le monde réel n'est pas I.D.I.



\* Bagging with a Change Detector [Adwin]



## LMGTFY<sup>a</sup>

a. Let Me Google That For You

*"Driver and vehicle factor codes were not added to non-injury crashes in the areas north of a line approximately from East Cape, south of Taupo, to the mouth of the Mokau River prior to 2007."*

## Panta Rhei<sup>a</sup> (Heraclitus, ≈500 BC) :

a. Panta Rhei est une formule qui, en grec ancien, signifie littéralement "Toutes les choses coulent" (i.e. "Tout passe").

- Le changement est inévitable ... Acceptez le ... !!
- Des instantanés, suffisamment courts, peuvent cependant sembler statiques.

## Une revendication : la plupart des Big Data sont en streaming

- Actuellement, une manière "**inefficace**" pour faire face :  
"Re-apprendre automatiquement et régulièrement (tous les soirs)  
à partir de zéro"<sup>a</sup>
- Le Stream Mining pourrait offrir une **alternative**.

a. Regularly (every night) retrain from scratch

## Plan :

- 1 Pourquoi le Stream Mining est important ?
- 2 En quoi est-il différent du Machine Learning ?
- 3 Cinq défis en Stream Mining
- 4 Hypothèse de l'IDI
  - 2 "big" questions
  - Une hypothèse fondamentale en Batch Machine Learning
  - Le monde réel n'est pas I.D.I.
- 5 Trois approches algorithmiques standards
- 6 Tout est une approximation
  - Échantillonnage de données
  - Sliding Window
  - Counting in  $\log(N)$  bits
  - Count-Min Sketch
  - Frequent algorithm

## Ré-inventer le Machine Learning

- Batch-incremental.
- Deux niveaux :
  - Online first level, résumés / schémas (summaries / sketches).
  - Offline second level, à la demande ou mis-à-jour régulièrement.
- Fully instance-incremental :
  - Algorithme classique adapté.
  - Nouvel algorithme.

## Plan :

- 1 Pourquoi le Stream Mining est important ?
- 2 En quoi est-il différent du Machine Learning ?
- 3 Cinq défis en Stream Mining
- 4 Hypothèse de l'IDI
  - 2 "big" questions
  - Une hypothèse fondamentale en Batch Machine Learning
  - Le monde réel n'est pas I.D.I.
- 5 Trois approches algorithmiques standards
- 6 **Tout est une approximation**
  - Échantillonnage de données
  - Sliding Window
  - Counting in  $\log(N)$  bits
  - Count-Min Sketch
  - Frequent algorithm

- Probablement vrai pour la plupart des méthodes de Batch Machine Learning.
- Pour le Stream Mining : **être exact est impossible.**

## Un petit jeu ensemble :D

## Sampling

### Reservoir Sampling :

- Collecter les  $k$  premières instances
- Puis, avec probabilité  $\frac{k}{n}$ , remplacer une entrée de réservoir aléatoire par la nouvelle instance.

### Min-Wise Sampling :

- Pour chaque instance de l'ensemble, générer un nombre aléatoire uniformément dans  $[0,1]$ .
- Ne gardez que le "plus petit"  $k$ .

## Sliding Window

- La forme la plus simple d'adaptation aux données changeantes.
- Ne conserver que les  $k$  derniers éléments.
- Quelle structure de données ?

Permet de mettre à jour efficacement les statistiques récapitulatives de la fenêtre des données (moyenne, variance, ...)

## Counting in $\log(N)$ bit

- Quel est le nombre d'éléments distincts dans un data stream ?
- La solution exacte nécessite un espace  $\approx O(N)$  pour  $N$  éléments distincts.
- L'esquisse de hachage (hash sketch) nécessite uniquement  $\log(N)$  des bits
  - Hash chaque élément, extraire la position du 1 le moins significatif dans le hashcode.
  - Garder une trace du  $p$  maximum pour tout élément.
  - $N \approx 2^p$  (Pourquoi ?)

Comment réduire l'erreur d'approximation ?

## Count-Min Sketch

- Compter les occurrences d'éléments dans un flux.  
(e.g. Combien de paquets / flux réseau)
- Solution exacte : hash-table, flow-identifier ⇒ Calcul, Coût de mémoire élevé ...

### Une alternative : utiliser un tableau (de taille fixe) de compteurs

- $c[\text{hachage}(\text{flux})]++$
- Collisions de hachage : le nombre est gonflé  
(mais, **JAMAIS** trop petit, **peut être** trop grand).
- Réduire *approx.error* : utiliser plusieurs fonctions de hachage.
  - Mettre à jour : incrémenter tout
  - Récupération : rapporter la valeur MIN

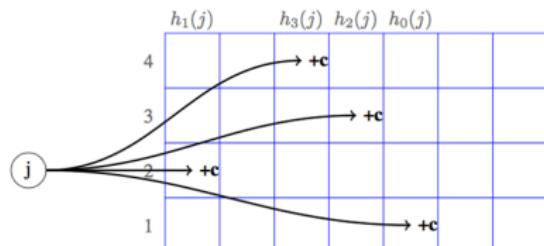


FIGURE 3 – Exemple de Count-Min Sketch

- Inspiré des filtres *Bloom*.
- L'idée : Supprimer les key-info coûteuses (des hachages) sera plus utile.
  - E.g. "*hashing trick*" dans des systèmes comme *Vowpal/Wabbit*<sup>2</sup>

## Frequent algorithm

- Trouvez les  $k - top$  éléments les plus importants en utilisant seulement  $n\_counts$  :  $k < n\_counts \ll N$

Pour chaque élément  $x$  :

- Si ( $x$  a été compté) Alors (Incrémenter le compteur.)
- Sinon : //( $x$  n'a pas été compté)  
Si (un compteur est nul) Alors (Allouer-le à  $x$  et incrémenter.)
- Sinon : //( $x$  n'a pas été compté) && (aucun compteur est nul)  
(Décrémenter tout les compteurs.)

## SpaceSaving algorithm

- Pour chaque élément  $x$  :
  - Si ( $x$  a été compté) Alors : Incrémente.
  - Sinon : trouver le plus petit  $count$ , allouez-le à  $x$  et incrémenter

- Structure de données efficace ?
- Fonctionne bien pour les distributions asymétriques (power laws).

## Résumé

## Stream Mining

**Stream mining = online learning **without** the IID assumption**

- Lots of missing bits.  $\Rightarrow$  Opportunity
- Lots of space for cool R&D.  $\Rightarrow$  Research & Development

Merci pour votre attention

Questions ?  
Remarques ?  
Suggestions ?  
... ?