Economically-Efficient Data Stream Analysis

Roberto Oliveira Jr.
Advisor: Adriano Veloso
Co-advisor: Wagner Meira Jr.

Computer Science Dept - UFMG - Brazil

Data Stream

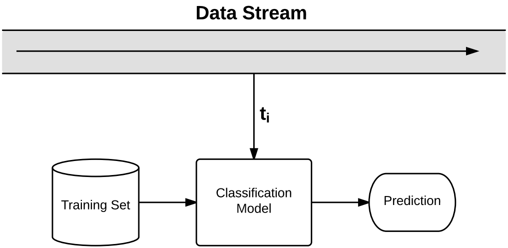
- Definition
  - Fast and possible unbounded sequence of data that arrives at time-varying.
- Motivation
  - It allows us to process huge volumes of data.
- Problem
  - Automatically extraction of relevant patterns and relations from data that is continuously created.
    - Keep track of data streams is useful for systems monitoring, online social network advertising, etc.

Social Networks Streams and Advertising

Superbowl 2013

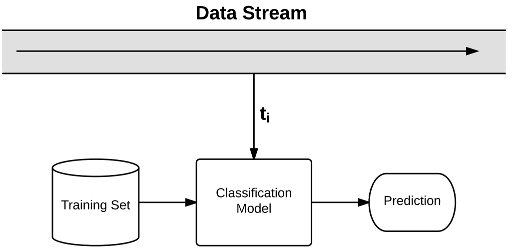
Classification in Data Streams

- Classification models are applied to distinguish between pre-defined labels.



Classification in Data Streams

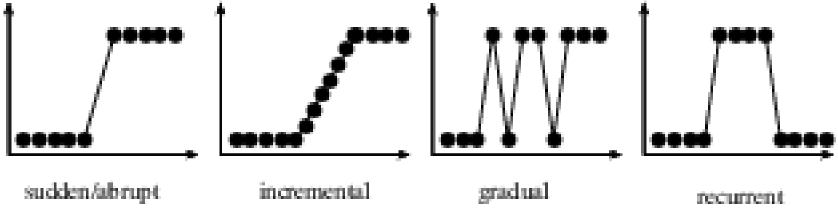
- Classification models are applied to distinguish between pre-defined labels.



- Data characteristics may change with time.

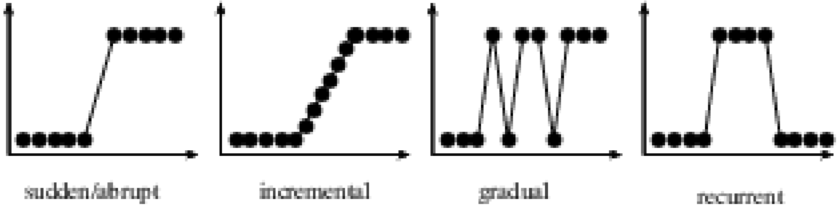
Concept Drifts

- Concept Drift is unforeseen changes in data's nature over time.



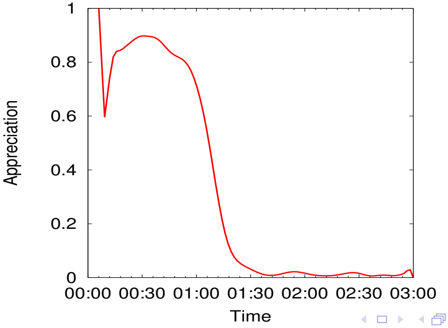
Concept Drifts

- Concept Drift is unforeseen changes in data's nature over time.

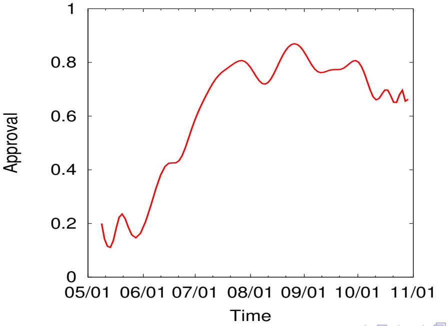


- Data streams contains combination of such patterns.

Sports (WC 2010)

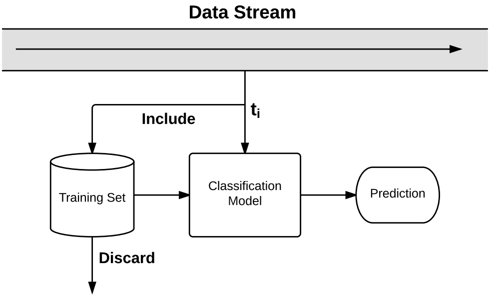


Elections (Brazil 2010)



Classifying Data Streams

- Effective classification requires:
  - Updating the classification model as the stream evolves.
    - Taking into account resources limitation: memory, time and learning requirements.



Research Question

How to deal with concept drifts?

Classification Model

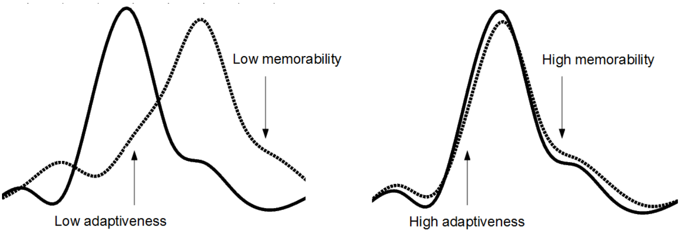
- Classification models are composed by association rules.
  - $\{x \rightarrow y\}$, where $x \in X$ and $y \in Y$
- Models are built on-the-fly:
  - For a given $[x_i, *]$, rules $\{x \rightarrow y\}$ such that $x \in x_i$ are produced.
  - Prediction is performed from the combination of these rules.
- At each time step is produced a model $\mathcal{R}(x_i)$.

Classification Model

- Classification models are composed by association rules.
  - $\{x \rightarrow y\}$, where $x \in X$ and $y \in Y$
- Models are built on-the-fly:
  - For a given $[x_i, *]$, rules $\{x \rightarrow y\}$ such that $x \in x_i$ are produced.
  - Prediction is performed from the combination of these rules.
- At each time step is produced a model $\mathcal{R}(x_i)$.
- Can be updated efficiently as the training set evolves.

Dealing with Drifts

- Two properties are necessary in order to produce classifiers that are robust to drifts:
  - Adaptiveness:
    - The ability to adapt itself to drifts.
  - Memorability:
    - The ability to recover itself from drifts.

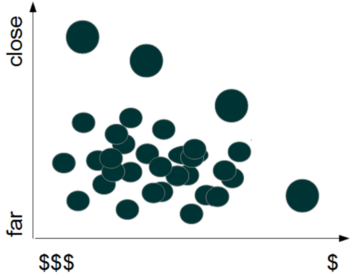


Dealing with Drifts

- Two properties are necessary in order to produce classifiers that are robust to drifts:
  - Adaptiveness:
    - The ability to adapt itself to drifts.
    - The training-set must contain fresh messages.
  - Memorability:
    - The ability to recover itself from drifts.
    - The training-set must contain pre-drift messages.
- Improving both properties simultaneously may lead to a conflict-objective problem.
  - Improve adaptiveness may hurt memorability, and vice-versa.

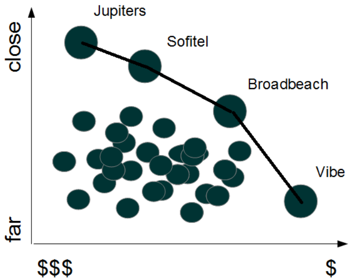
Pareto Efficiency

Example: hotels in Petrópolist.



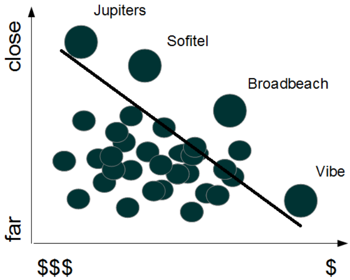
Pareto Efficiency

Pareto frontier



Compensation — Kaldor-Hicks Principle

Region of compensation

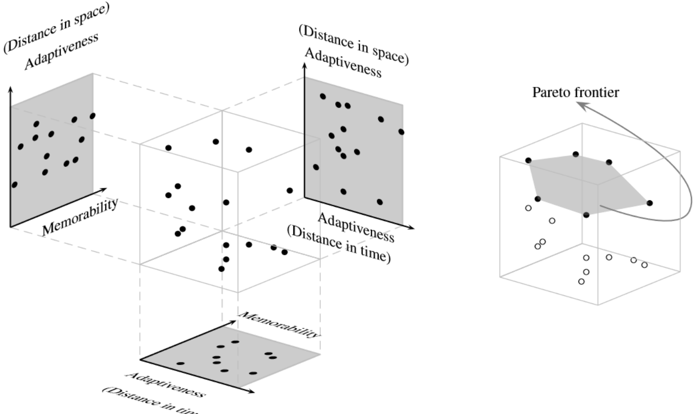


Utility Measures

- Distance in space:
  - How similar message t_j is to the newest message t_n.
  - $U_s(t_j) = \frac{|\mathcal{R}(t_n) \cap \mathcal{R}(t_j)|}{|\mathcal{R}(t_n)|}$
- Distance in time:
  - How fresh is the message.
  - $U_t(t_j) = \frac{\gamma(t_j)}{\gamma(t_n)}$.
    - $\gamma(t_j)$ returns the time in which message t_j arrived.
- Random permutation of messages:
  - $U_r(t_j) = \frac{\alpha(t_j)}{|\mathcal{D}_n|}$
    - $\alpha(t_j)$ returns the position of t_j in the shuffle.
    - $\mathcal{D}_n$ is the training set at time step n.

Utility Measures

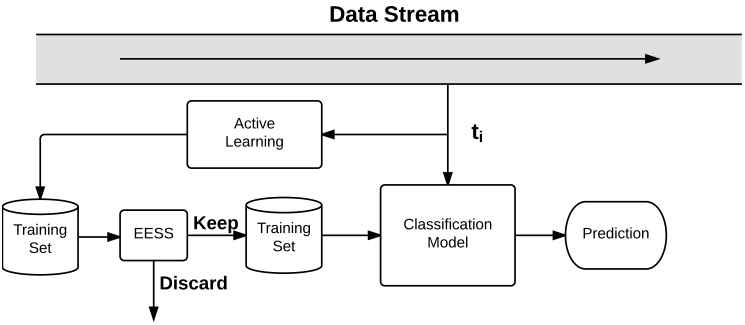
1. At each time step n:
   1. Place candidate messages in the utility space.
   2. Select messages in the Efficiency Region (Pareto-frontier / Kaldor-Hicks Region).



Reducing Labeling Efforts

- Random Active Learning
  - Naive strategy.
  - Simple to integrate.
  - Labeling Effort control: δ.

EESS in Stream Classification Flow



Experimental Evaluation

Setup

- Interleaved Test-Then-Train
- 1% of data provided as training seed;
- Massive Online Analysis (MOA) framework as evaluation environment;
- Baselines:
    - AC – Active Classifiers (KDD 2011)
    - HAT – Hoeffding Adaptive Trees (JMLR 2011)
    - ILAC – Incremental Lazy Classifiers (SIGIR 2011)
- Labeling Efforts (AC and EESS): 10%; **25%**; 50%; 75% and 100%;

Evaluation

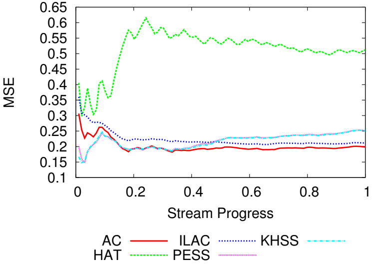
- Measures used:
  - Mean Squared Error.
  - Labeling Effort.
  - Training set site.
  - RAM-Hours:
    - A GB of RAM deployed for 1 hour execution.
- Datasets:

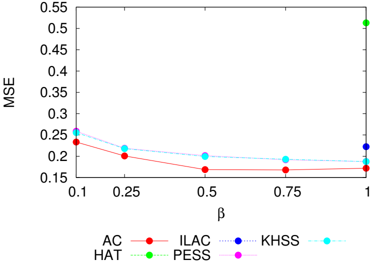
	Concept Drift Pattern			
Dataset	Sudden	Incremental	Gradual	Recurrent
Presidential Elections	-	X	X	-
Person of the Year	-	X	X	-
FIFA World Cup - EN	X	-	-	-
FIFA World Cup - PT	X	-	-	-
Cover Type	X	-	X	X
Spam Filtering	X	-	X	X
Poker Hand	-	-	X	X

Evaluation

Brazilian Presidential Elections

MSE and Labeling Efforts

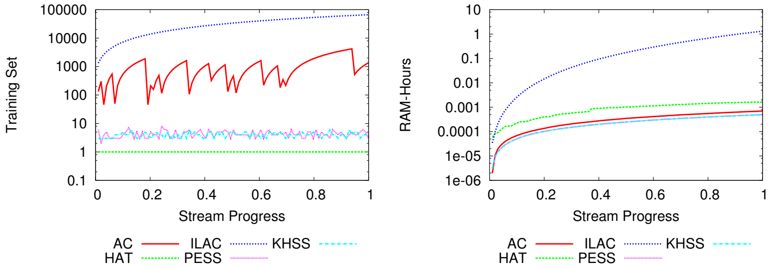




Evaluation

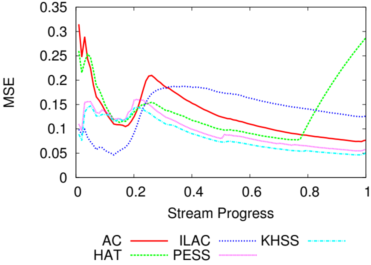
Brazilian Presidential Elections

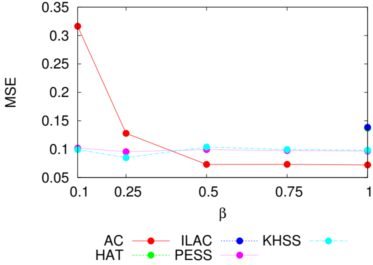
Training Size and RAM-Hours



Evaluation
FIFA World Cup - Portuguese

MSE and Labeling Efforts

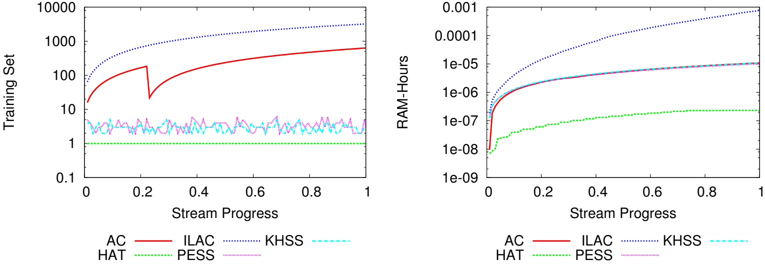




Evaluation

FIFA World Cup - Portuguese

Training Size and RAM-Hours



Conclusions

- Sentiment analysis on Twitter streams.
  - Limited computing and training resources.
  - Sentiment drifts.
- Efficiency and accuracy.
  - Incremental classifiers.
  - Pareto efficiency and compensation principle.
- Our results.
  - 50% reduction in terms of labeling effort without impact on accuracy.
- Future work includes:
  - Other utility measures.
  - Other application scenarios.

Thank you!

adrianov@dcc.ufmg.br