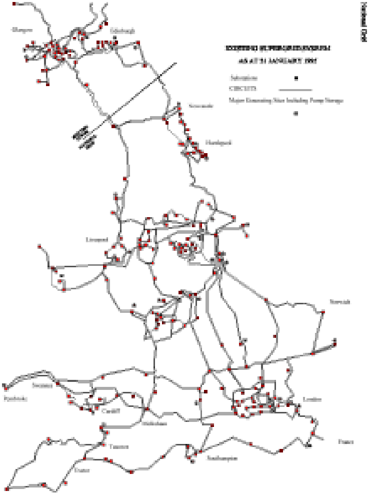
Data Streaming

- When: data, specificities
- What: goals
- How: algorithms

More: see Joao Gama's tutorial,
http://wiki.kdubiq.org/summerschool2008/index.php/Main/Materials

Motivations



Electric Power Network

Data

Input

- Continuous flow of (possibly corrupted) data, high speed
- Huge number of sensors, variable along time (failures)
- Spatio-temporal data

Output

- Cluster: profiles of consumers
- Prediction: peaks of demand
- Monitor Evolution: Change detection, anomaly detection

Where is the problem ?

Standard Data Analysis

- Select a sample
- Generate a model (clustering, neural nets, ...)

Where is the problem ?

Standard Data Analysis

- Select a sample
- Generate a model (clustering, neural nets, ...)

Does not work...

- World is not static
- Options, Users, Climate, ... change

Specificities of data

Domain

- Radar: meteorological observations
- Satellite: images, radiation
- Astronomical surveys: radio
- Internet: traffic logs, user queries, ...
- Sensor networks
- Telecommunications

Features

- Most data never seen by humans
- Need for REAL-TIME monitoring, (intrusion, outliers, anomalies,,,)

NB: Beyond ML scope: data are not iid (independent identically distributed)

Data streaming Challenges

Maintain Decision Models in real-time

- incorporate new information   comply with speed
- forget old/outdated information
- detect changes and adapt models accordingly

Unbounded training sets   Prefer fast approximate answers...

- Approximation: Find answer with factor $1 \pm \epsilon$
- Probably correct: Pr(answer correct) $=$ 1 -δ
- PAC: ϵ, δ (Probably Approximately Correct)
- Space $\approx \mathcal{O}(1/\epsilon^2 log(1/\delta))$

Data Mining vs Data Streaming

	Traditional	Stream
Nr. of Passes	Multiple	Single
Processing Time	Unlimited	Restricted
Memory Usage	Unlimited	Restricted
Type of Result	Accurate	Approximate
Distributed	No	Yes

What: queries on a data stream

- Sample
- Count number of distinct values / attribute
- Estimate sliding average (number of 1's in a sliding window)
- Get top-k elements

Application: Compute entropy of the stream

$$H(x) = \sum p_i log_2(p_i)$$

useful to detect anomalies

Sampling

Uniform sampling: each one out of n examples is sampled with probability 1/n.
What if we don't know the size ?

Standard

- Sample instances at periodic time intervals
- Loss of information

Reservoir Sampling

- Create buffer size k
- Insert first k elements
- Insert i-th element with probability k/i
- Delete a buffer element at random

Limitations

- Unlikely to detect changes/anomalies
- Hard to parallelize

Count number of values

Problem

Domain of the attribute is $\{1, \dots M\}$

Piece of cake if memory available... What if the memory available is $log(M)$?

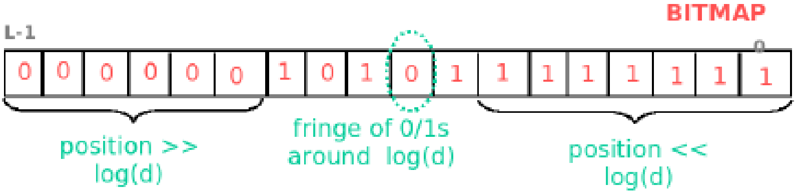
Flajolet-Martin 1983

Based on hashing: $\{1, \dots M\} \mapsto \{0, \dots 2^L\}$ with $L = log(M)$.

$$x \rightarrow \ hash(x) = y \ \rightarrow \ position\ least\ significant\ bit, lsb(x)$$

Count number of values, followed

Init: $BITMAP(\{0, \ldots L\}) = 0$
Loop: Read x, $BITMAP(lsb(x)) = 1$



Result

$$R = \text{ position of rightmost 0 in } H$$

$$M \approx 2^R / .7735$$

Decision Trees for Data Streaming

Principle
Grow the tree if evidence best attribute $>$ second best

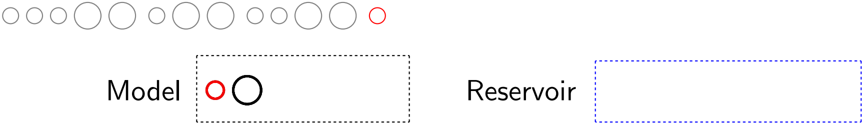
Algorithm parameter: confidence δ (user-defined)

While true
  Read example, propagate until a leaf
  If enough examples in leaf
    Compute IG for all attributes;
    $\epsilon = \sqrt{\frac{R^2 ln(1/\delta)}{2n}}$
    Keep best if IG(best) - IG(second best) $> \epsilon$

Mining High Speed Data Streams, Pedro Domingos, Geoffrey Hulten, KDD-00

Stream clustering

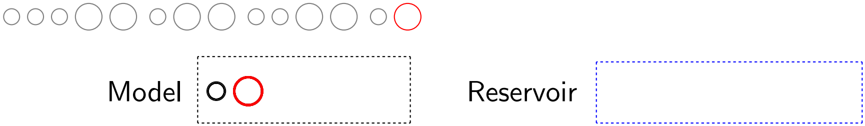
Stream clustering



Does e_t fit the current model ??

- if yes, update the model
- otherwise, put outlier e_t in reservoir

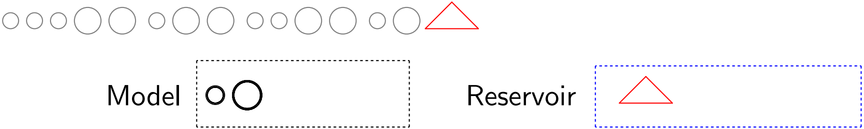
Stream clustering



Does e_t fit the current model ??

- if yes, update the model
- otherwise, put outlier e_t in reservoir

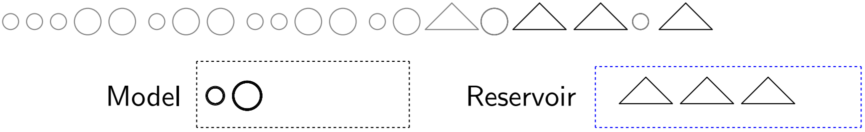
Stream clustering



Does e_t fit the current model ??

- if yes, update the model
- otherwise, put outlier e_t in reservoir

Stream clustering

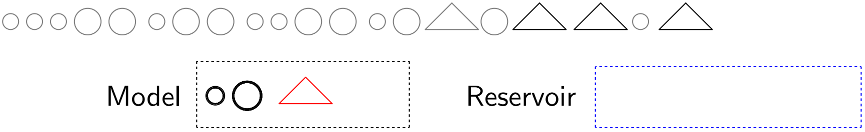


CHANGE TEST

Has the distribution changed ?

- if yes, rebuild the model
- otherwise, continue

Stream clustering

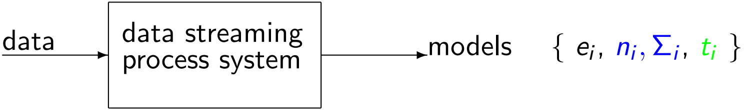


Has the distribution changed ?

- if yes, rebuild the model
- otherwise, continue

CHANGE TEST

Strap



Does e_t fit the current model ?

- if yes, update the model
- otherwise, put e_t in reservoir

Has the distribution changed ?

- if yes, rebuild the model
- otherwise, continue

Update the model

Stream Model: $\{(e_i, n_i, \Sigma_i, t_i)\}$

- e_i examplar
- n_i number of items represented by e_i
- Σ_i sum of distortions incurred by e_i
- t_i last time step when a point was affected to e_i

Update with decay: Δ: time window

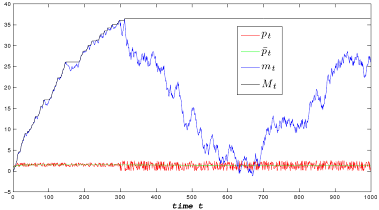
$$
\begin{aligned}
n_i &:= n_i \times \left(\tfrac{\Delta}{\Delta + (t - t_i)} + \tfrac{1}{n_i + 1} \right) \\
\Sigma_i &:= \Sigma_i \times \tfrac{\Delta}{\Delta + (t - t_i)} + \tfrac{n_i}{n_i + 1} d(e_t, e_i)^2 \\
t_i &:= t
\end{aligned}
$$

Rebuild the model

Trigger

- when reservoir is full
- when changes are detected

Page-Hinkley statistic



$$\bar{p}_t = \frac{1}{t}\sum_{\ell=1}^{t} p_\ell$$
$$m_t = \sum_{\ell=1}^{t}(p_\ell - \bar{p}_\ell + \delta)$$
$$PH_t = max\{m_\ell\} - m_t$$

HINKLEY D. Inference about the change-point in a sequence of random variables. Biometrika, 1970

PAGE E. Continuous inspection schemes. Biometrika, 1954

Experimental validation

Data used

- Artificial dataset
- Real world data: KDD99 data
  - intrusion detection benchmark
  - 494,021 network connection records in $\mathbb{R}^{34}$
  - 23 classes: 1 normal + 22 attacks
- Baseline: DenStream

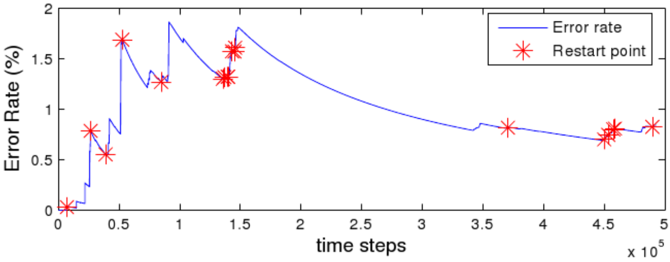
  F. Cao, M. Ester, W. Qian, A. Zhou. Density-Based Clustering over an Evolving Data Stream with Noise. SDM 2006.

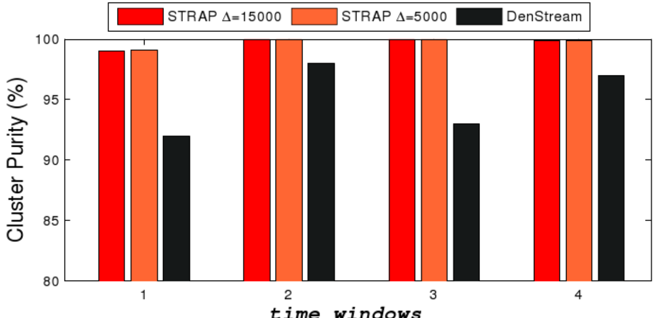
Performance indicator

- Distortion
- Clustering accuracy / Clustering purity (supervised setting)

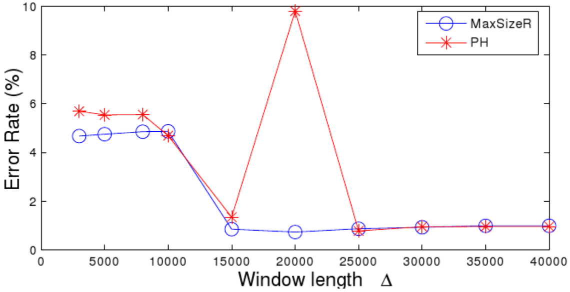
KDD Cup 1999 data: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

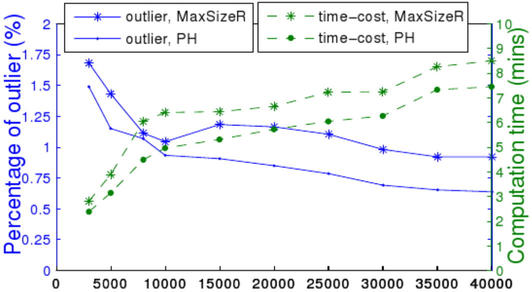
Accuracy along time





Restart criteria: MaxSizeR vs PH





Discussion

Rebuild: ReservoirSize vs PH

- PH is 10% better than ReservoirSize
- PH is less stable

Strap vs DenStream

- Pros
  - better accuracy
  - model available at any time
- Cons
  - DenStream: 7 seconds
  - Strap : 7 mins

Conclusion

Scalability: Hi-WAP

- Reduce complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N^{3/2})$
- iteratively reduce toward $\mathcal{O}(N^{(1+\gamma)})$

Stream clustering: Strap

- Hybridized with an efficient change detection method, Page-Hinkley
- Model available at any time
- BUT: slower than DenStream

Future work Provide an upper bound on the distortion loss caused by Hi-WAP

Open issues

What's new

Forget about iid;

Forget about more than linear complexity (and log space)

Challenges

Online, Anytime algs

Distributed alg.

Criteria of performance

Integration of change detection