ECML 2007 PKDD
WARSAW POLAND

THE 18TH EUROPEAN CONFERENCE ON MACHINE LEARNING
AND
THE 11TH EUROPEAN CONFERENCE ON PRINCIPLES AND PRACTICE
OF KNOWLEDGE DISCOVERY IN DATABASES

STATE-OF-THE-ART IN DATA STREAM MINING

TUTORIAL NOTES

---

presented by
Mohamed Gaber and Joao Gama

September 17, 2007
Warsaw, Poland

Prepared and presented by:
Mohamed Gaber
Tasmanian ICT Centre, CSIRO ICT Centre, Australia
João Gama
Laboratory of Artificial Intelligence and Decision Support,
INESC-Porto, University of Porto, Portugal

Tutorial Summary

Data streams became ubiquitous as many sources produce data continuously and rapidly. Examples of streaming data include customer click streams, telephone records, web logs, multimedia data, and sets of retail chain transactions. Data streams have brought new challenges to the data mining research community. In consequence, new techniques are needed to process streaming data in reasonable time and space. The goal of this tutorial is to present and discuss the research problems, issues and challenges in learning from data streams. We will present the state-of-the-art techniques in change detection, clustering, classification, frequent patterns, and time series analysis from data streams. Applications of mining data streams in different domains are highlighted. Open issues and future directions will conclude this tutorial. The tutorial also points to data stream mining resources.

Specific goals and objectives

- Introducing the area of data stream mining
- Giving a detailed explanation of the major techniques in the area
- Emphasizing the research issues and challenges

Expected background of the audience

Basic knowledge of data mining concepts and techniques is required.

Outline

1. Introduction
2. Data Streams
3. Change Detection
4. Learning Descriptive Models from Data Streams
5. Learning Predictive Models from Data Streams
6. Frequent pattern mining
7. Time series analysis in data streams
8. Applications of mining data streams
9. Future Directions

Prepared and presented by:

Mohamed Gaber
Tasmanian ICT Centre, CSIRO ICT Centre, Australia

Mohamed Medhat Gaber is a research scientist at Commonwealth Scientific and Industrial Research Organization (CSIRO), Australia. He has published more than 40 articles. Mohamed has served in the program committees of several international and local conferences and workshops in the area of data mining. He has also been serving as a reviewer for the special issues of international journals in the area of data stream mining. He was the co-chair of the International Workshop on Mining Evolving and Streaming Data held in conjunction with ICDM 2006. He is the co-chair of the International Workshop on Knowledge Discovery from Ubiquitous Data Streams to be held in conjunction with ECML/PKDD 2007 and the ACM Workshop on Knowledge Discovery from Sensor Data to be held in conjunction with ACM SIGKDD 2007.

João Gama
Laboratory of Artificial Intelligence and Decision Support, INESC-Porto, University of Porto.

Joao Gama is a researcher at LIACC, the Laboratory of Artificial Intelligence and Computer Science of the University of Porto, working at the Machine Learning group. His main research interest is in Learning from Data Streams. He has published several articles in change detection, learning decision trees from data streams, hierarchical Clustering from streams, etc. Editor of special issues on Data Streams in Intelligent Data Analysis, J. Universal Computer Science, and New Generation Computing Co-chair of a series of Workshops on Knowledge Discovery in Data Streams, ECML 2004, Pisa, Italy, ECML 2005, Porto, Portugal, ICML 2006, Pittsburg, US, ECML 2006 Berlin, Germany, SAC2007, Korea, and the ACM Workshop on Knowledge Discovery from Sensor Data to be held in conjunction with ACM SIGKDD 2007.

Joao and Mohamed are editing a book titled: *Learning from Data Streams-Processing Techniques in Sensor Networks* to be published by Springer.

State-of-the-Art in Data Stream Mining (Part I)

João Gama

LIAAD-INESC Porto, University of Porto, Portugal

September 2007

ALES II Adaptive LEarning Systems II (POSC/EIA/55340/2004)

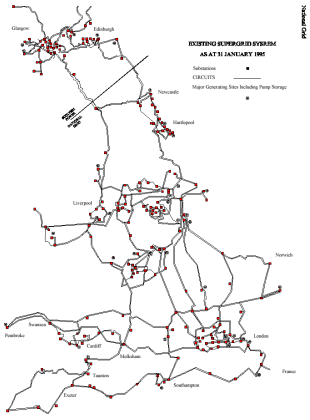
State-of-the-Art in Data Stream Mining (Part I)

João Gama

Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Outline

Scenario



Electrical power Network: Sensors all around network monitor measurements of interest.

Scenario

- Sensors produce continuous flow of data at high speed:
  - Sensors send information at different time scales;
  - Sensors act in adversary conditions: they are prone to noise, weather conditions, battery conditions, etc;
- Huge number of Sensors, variable along time
- Geographic distribution:
  - The topology of the network and the position of the sensors are known.

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Illustrative Learning Tasks:

- Monitoring Evolution
  - Anomaly Detection
  - Extreme Values and Outlier Detection
    - Identification of picks on the demand.
    - Identification of **critical points** in load evolution.
  - Change Detection
    - Detect changes in the behaviour of sensors

Illustrative Learning Tasks:

- Monitoring Evolution
  - Anomaly Detection
  - Extreme Values and Outlier Detection
    - Identification of picks on the demand.
    - Identification of **critical points** in load evolution.
  - Change Detection
    - Detect changes in the behaviour of sensors
- Cluster Analysis
  - Identification of Profiles: Urban, Rural, Industrial, etc.

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Illustrative Learning Tasks:

- Monitoring Evolution
  - Anomaly Detection
  - Extreme Values and Outlier Detection
    - Identification of picks on the demand.
    - Identification of **critical points** in load evolution.
  - Change Detection
    - Detect changes in the behaviour of sensors
- Cluster Analysis
  - Identification of Profiles: Urban, Rural, Industrial, etc.
- Predictive Analysis
  - Predict the value measured by each sensor for different time horizons.
  - Prediction of picks on the demand.

Outline

The Data Stream Phenomenon

- Highly detailed, automatic, rapid data feeds.
  - Radar: meteorological observations.
  - Satellite: geodetics, radiation,.
  - Astronomical surveys: optical, radio,.
  - Internet: traffic logs, user queries, email, financial,
  - Sensor networks: many more *observation points* ...
- Most of these data will never be seen by a human!
- Need for near-real time analysis of data feeds.
- Monitoring, intrusion, anomalous activity, classification, prediction, complex correlations, detect outliers, extreme events, fraud, ...

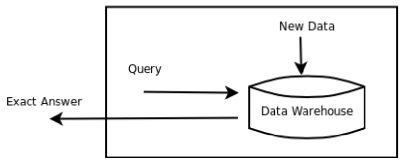
Data Streams

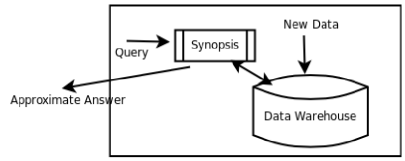
Continuous flow of data generated at **high-speed** in **Dynamic**, **Time-changing** environments.
The usual approaches for querying, clustering and prediction use **batch procedures** cannot cope with this streaming setting.
We need to maintain **Decision models** in **real time**.
Decision Models must be capable of:

- **incorporating** new information at the speed data arrives;
- **forgetting** outdated information;
- **detecting** changes and **adapting** the decision models to the most recent information.

Massive Data Sets

- Data analysis is complex, interactive, and exploratory over very large volumes of historic data, eventually stored in distributed environments.
- Traditional pattern discovery process requires online ad-hoc queries, not previously defined, that are successively refined.
- Due to the exploratory nature of these queries, an exact answer may not be required. A user may prefer a fast approximate answer.





Illustrative Examples

- We see a large number of individual transactions.
  - What are the top sellers today?
- We are monitoring network traffic.
  - Which hosts/subnets are responsible for most of the traffic?
- We have a network of satellites monitoring events over large areas.
  - Which areas are experiencing the most activity over a week / day /hour?

Data Stream Models

In the stream model the input elements $a_1, a_2, \ldots, a_j, \ldots$ arrive sequentially, item by item and describe an underlying function A.

- Insert Only Model: once an element a_i is seen, it can not be changed;
- Insert-Delete Model: elements a_i can be deleted or updated;

Monitoring: Querying Data

- Data continuous flow over time at high speed.
- Computational Resources are limited.
- How to Query data?
  - Continuous Queries
  - Continuous Aggregations
  - Continuous Joins
- Problem: Blocking Operators
  Some SQL Operators (SORT, SUM, COUNT, MIN, ...) only return the first output tuple, after reading all the input records!

Traditional / Stream Processing

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

Approximate Answers

Approximate answers:

Actual answer is within 5 ± 1 *with probability* ≥ 0.9.

- Approximation: find an answer correct within some factor
  - Find an answer that is within 10% of correct result
  - More generally, a $(1 \pm \epsilon)$ factor approximation
- Randomization: allow a small probability of failure
  - Answer is correct, except with probability 1 in 10,000
  - More generally, success probability $(1 - \delta)$
- Approximation **and** Randomization: (ϵ, δ)-approximations

The constants ϵ and δ have great influence in the space used. Typically the space is $O(1/\epsilon^2 log(1/\delta))$.

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Tail Inequalities

Approximate answers:

Trade-off between accuracy of the answer and computational resource required to compute an answer.

Tail inequalities:

General bounds on the tail probability of random variables. The probability that a random variable deviates far from its expectation.

Chebyshev Inequality

if X is a random variable with standard deviation σ, the probability that the outcome of X is no less than $k\sigma$ away from its mean is no more than $1/k^2$:

$$P(|X - \mu| \leq k\sigma) \leq \frac{1}{k^2}$$

No more than $1/4$ of the values are more than 2 standard deviations away from the mean, no more than $1/9$ are more than 3 standard deviations away, no more than $1/25$ are more than 5 standard deviations away, and so on.

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Chernoff Bound

Consider a biased coin. One side is more likely to come up than other, but we don't know which and would like to find it.

- Flip it many times and then choose the side that comes up the most.
- How many times do you have to flip it to be confident that you've chosen correctly?

Example: p=0.6; $\delta = 95\%$

$$n \geq \frac{ln(1/\sqrt{\delta})}{(p-1/2)^2}$$

Hoeffding Bound

Characterize the deviation between the true probability of some event and its frequency over m independent trials.

$$P(|\overline{X} - \mu| \geq \epsilon) \leq 2exp(-2m\epsilon^2/R^2),$$

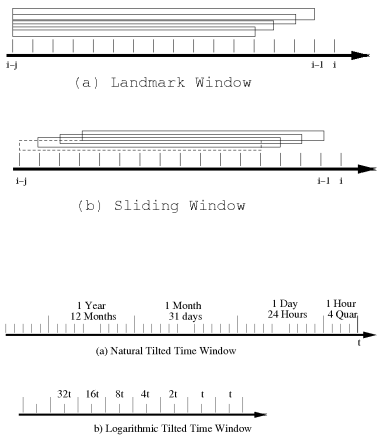
where R is the range of the random variables.

Example: After seeing 100 examples of a random variable X, $x_i \in [0, 1]$, the sample mean is $\overline{x} = 0.6$;
The true mean is with confidence δ in $\overline{x} \pm \epsilon$, where

$$\epsilon = \frac{\sqrt{R^2 ln(1/\delta)}}{2n}$$

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Time Windows



- Instead of computing statistics over all the stream ...
- use only the most recent data points.
- Most recent data is more relevant than older data
- Several Window Models: **Landmark**, **Sliding**, **Tilted** Windows.

Basic Stream Methods

- Sampling
- Data Synopsis:
  - Sketches
  - Synopsis
  - Histograms
  - Wavelets

Sampling

To otain an unbiased sampling of the data, we need to know the lenght of the stream. In Data Streams, we need to modify the approach!

Strategy

- Sample instances at periodic time intervals
- Useful to *slow down* data.
- Involves *loss* of information.

Sampling

To otain an unbiased sampling of the data, we need to know the lenght of the stream. In Data Streams, we need to modify the approach!

Strategy

- Sample instances at periodic time intervals
- Useful to *slow down* data.
- Involves *loss* of information.

Known Problems

Not possible to detect:

- Changes
- Anomalies

The reservoir Sample Technique

Vitter, J.; *Random Sampling with a Reservoir*, ACM, 1985.

- Creates uniform sample of fixed size k;
- Insert first k elements into sample
- Then insert ith element with prob. $p_i = k/i$
- Delete an instance at random.

Illustrative Problems

Illustrative Problems

Illustrative Problems:

- Count the number of distinct values in a stream;
- Count the number of 1's in a sliding window of a binary string;
- Count frequent items above a given support.

Illustrative Problems

State-of-the-Art in Data Stream Mining (Part I)

João Gama

Outline

Motivation

Data Streams

Basic Methods

Change Detection

Predictive Learning

Clustering Data Streams

Predictive Models from Data Streams

Decision Trees

Neural Networks

Illustrative Problems

Illustrative Problems:

- Count the number of distinct values in a stream;
- Count the number of 1's in a sliding window of a binary string;
- Count frequent items above a given support.

Count the Number of Distinct Values in a Stream

Assume that the domain of the attribute is $\{0, 1, \ldots, M-1\}$.
The problem is trivial if we have space linear in M.
Is there an approximate solution is space $log(M)$?

FM Sketches for Distinct Value Estimation

Flajolet and Martin; *Probabilistic Counting Algorithms for DataBase Applications*, JCSS, 1983

- Maintain a *Hash Sketch* = BITMAP array of L bits,, where $L = \mathcal{O}(log(M))$, initialized to 0.
- Assume a hash function $h(x)$ that maps incoming values $x \in [0, \ldots, M-1]$, *uniformly* across $[0, \ldots, 2^{(L-1)}]$.
- Let $lsb(y)$ denote the position of the least-significant 1 bit in the binary representation of y.
- A value x is mapped to $lsb(h(x))$.
- For each incoming value x, set BITMAP$[lsb(h(x))] = 1$.

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

FM Sketches for Distinct Value Estimation

Example:

BITMAP:

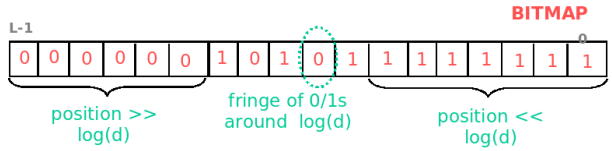
5	4	3	2	1	0
0	0	0	0	0	0

$x = 5 \rightarrow h(x) = 101100 \rightarrow lsb(h(x)) = 2$

BITMAP:

5	4	3	2	1	0
0	0	0	1	0	0

FM Sketches for Distinct Value Estimation



- By uniformity through $h(x)$: $P(BITMAP[k] = 1) = Prob(10^k) = 1/2^{k+1}$
- Let R= position of the rightmost zero in BITMAP
- R is an indicator of $log(d)$
- Flajolet and Martin [FM85] prove that $E[R] = log(\phi M)$, where $\phi = .7735$
- Estimate of $M = 2^R/\phi$

Exponential Histograms

Computing Statistics in a sliding window of incoming examples.
Illustrative Problem: Count the number of 1's from a moving window in a binary string.
Easy if we can store all the elements inside the window.
What if

Exponential Histograms

Maintaining Stream Statistics over Sliding Windows, M.Datar, A.Gionis, P.Indyk, R.Motwani; ACM-SIAM Symposium on Discrete Algorithms;2002
The basic idea:

- Use buckets of different sizes to hold the data
- Each bucket has a timestamp associated with it
- It is used to decide when the bucket is out of the window

Data Structures for Exponential Histograms:

- Buckets: counts and time stamp
- LAST: stores the size of the last bucket.
- TOTAL: keeps the total size of the buckets.

The estimate of the sum of data elements is proven to be bounded within a user-specified parameter.

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Exponential Histograms

Consider a simplified data stream environment where each element comes from the same data source and is either 0 or 1.
When a new data element arrives:

- If the new data element is 0, ignore it
- Otherwise create a new bucket of size 1 with the current timestamp, and increment the counter TOTAL.
- Given a parameter,ϵ, if there are $\lfloor 1/\epsilon \rfloor/2 + 2$ buckets of the same size, merge the oldest two of these same-size buckets into a single bucket of double size.
- The larger timestamp of the two buckets is then used as the timestamp of the newly created bucket.
- If the last bucket gets merged, we update the size of the merged bucket to the counter LAST.

Exponential Histograms

Whenever we want to estimate the moving sum:

- Check if the oldest bucket is within the sliding window.
- If not, we drop that bucket:
  subtract its size from the variable TOTAL and
  update the size of the current oldest bucket to the variable LAST.
- Repeat the procedure until all the buckets with timestamps outside of the sliding window are dropped.
- The estimate of 1's in the sliding window is TOTAL-LAST/2.

Exponential Histograms: Analysis

- The size of the buckets grows exponentially: $2^0, 2^1, 2^2 \ldots 2^h$
- Need only $O(logN)$ buckets.
- It is shown that, for N 1's in the sliding window, we only need $O((logN)/\epsilon)$ buckets to maintain the moving sum and the error of estimating
- The error in the oldest bucket **only**.
- The moving sum is proven to be bounded within a given relative error, ϵ.

Exponential Histograms: Example

Time	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Element	1	1	1	1	0	1	0	1	1	1	1	1	1	1	0

Window length=10
Relative Error=0.5
Merge if 3 buckets of the same size: $\lfloor 1/0.5 \rfloor / 2 / 2$

Time	Buckets	Total	Last
T1	1_1	1	1
T2	$1_1, 1_2$	2	1
T3	$1_1, 1_2, 1_3$	3	1
(merge)	$2_2, 1_3$	3	1
T4	$2_2, 1_3, 1_4$	3	2
...			
T11	$4_4, 2_8, 2_{10}, 1_{11}$	9	4
T12	$4_4, 2_8, 2_{10}, 1_{11}, 1_{12}$	10	4
T13	$4_4, 4_{10}, 2_{12}, 1_{13}$	11	4
T14	$4_4, 4_{10}, 2_{12}, 1_{13}, 1_{14}$	12	4
(Removing out-of-date)			
T15	$4_{10}, 2_{12}, 1_{13}, 1_{14}$	8	4

Current Research on Data Streams

- Basic stream synopses computation
  Samples, Equi-depth histograms, Wavelets
- Sketch-based computation techniques
  Self-joins, Joins, Wavelets, V-optimal histograms
- Advanced techniques
  Sliding windows, Distinct values, Hot lists

Bibliography

- *Data Streams: Algorithms and Applications* (2003) S. Muthukrishnan
- *Stream Data Management* (2005) N. Chaudry, K. Shaw, M. Abdelguerfi, Springer
- *Data Streams and Data Synopses for Massive Data Sets*, Yossi Matias (Invited Talk at ECML-PKDD 05)
- *Models and Issues in Data Stream Systems* (2002), Brian Babcock Shivnath Babu Mayur Datar Rajeev Motwani Jennifer Widom ;PODS
- *Querying and Mining Data Streams: You only get one look*; M. Garafalakis, J. Gehrke, R. Rastagi;
- *Randomized Algorithms*; R.Motwani, P. Raghavan, Cambridge University Press, 1995
- *Data Mining Concepts and Techniques*, J. Hanm M. Kambler, Morgan Kaufmann, 2006

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Outline

Introduction

Data flows continuously over time *Dynamic Environments*.
Some characteristic properties of the problem can change over time.
Machine Learning algorithms assume:

- Instances are generated at random according to some probability distribution $\mathcal{D}$.
- Instances are independent and identically distributed
- It is required that $\mathcal{D}$ is stationary

Examples:

- e-commerce, user modelling
- Spam emails
- Fraud Detection, Intrusion detection

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Introduction

Concept drift means that the concept about which data is obtained may shift from time to time, each time after some minimum permanence.
Any change in the distribution underlying the data

Context: a set of examples from the data stream where the underlying distribution is stationary

The Nature of Change

The causes of change:

- Changes due to modifications in the context of learning due to changes in **hidden variables**.
- Changes in the characteristic properties of the observed variables.

Change Detection in Predictive Learning

When there is a change in the class-distribution of the examples:

- The actual model does not correspond any more to the actual distribution.
- The error-rate increases

Basic Idea: Monitor the evolution of the error rate.
Main Problems:

- How to distinguish Change from Noise?
- How to React to drift?

A Framework based on Statistical Quality Control

Suppose a sequence of examples in the form $< \vec{x}_i, y_i >$
The actual decision model classifies each example in the sequence
In the 0-1 loss function, predictions are either True or False
The predictions of the learning algorithm are sequences:
$T, F, T, F, T, F, T, T, T, F, \ldots$.
The Error is a random variable from *Bernoulli* trials.
The Binomial distribution gives the general form of the probability of observing a F:
$p_i = (F/i)$ and $s_i = \sqrt{p_i(1-p_i)/i}$ where i is the number of trials.

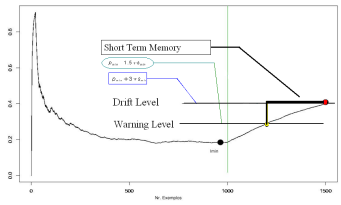
The P-chart Algorithm

The algorithm maintains two registers: P_{min} and S_{min} such that $P_{min} + S_{min} = min(p_i + s_i)$
Minimum of the error rate taking into account the variance of the estimator.
At example j:
The error of the learning algorithm will be

- **Out-control** if $p_j + s_j > p_{min} + \alpha * s_{min}$
- **In-control** if $p_j + s_j < p_{min} + \beta * s_{min}$
- **Warning Level**: if
  $p_{min} + \alpha * s_{min} > p_j + s_j > p_{min} + \beta * s_{min}$

The constants α and β depend on the desired confidence level.
Admissible values are $\beta = 2$ and $\alpha = 3$.

The P-chart Algorithm



At example j the actual decision model classifies the example

- Compute the error and variance: p_j and s_j
- If the error is
  - **In-control** the actual model is updated Incorporate the example in the decision model
  - **Warning zone**: Maintain the actual model
    First Time: the lower limit of the window is: $L_{warning} = j$
  - **Out-Control** Re-learn a new model using as training set the set of examples $[L_{warning}, j]$.

Analysis of the P-chart Algorithm

- Independent of the Learning Algorithm
- Resilient to False Alarms
- Maintain a single Decision Model in Memory

Main Characteristics in Change Detection

- **Data management**
  Characterizes the information about training examples stored in memory.
- **Detection methods**
  Characterizes the techniques and mechanisms for drift detection
- **Adaptation methods**
  Adaptation of the decision model to the current distribution
- Decision model management

Decision model management

Model management characterize the number of decision models needed to maintain in memory.
The key issue here is the assumption that data generated comes from multiple distributions,

- at least in the transition between contexts.
- Instead of maintaining a single decision model several authors propose the use of multiple decision models.

Dynamic Weighted Majority

A seminal work, is the system presented by Kolter and Maloof (ICDM03, ICML05).
The Dynamic Weighted Majority algorithm (DWM) is an ensemble method for tracking concept drift.

- Maintains an ensemble of base learners,
- Predicts using a weighted-majority vote of these *experts*.
- Dynamically creates and deletes experts in response to changes in performance.

Granularity of Decision Models

Occurrences of drift can have impact in part of the instance space.

- **Global models:** Require the reconstruction of all the decision model. (like naive Bayes, SVM, etc)
- **Granular decision models**: Require the reconstruction of parts of the decision model (like decision rules, decision trees)

Outline

Online Divisive-Agglomerative Clustering

State-of-the-Art in Data Stream Mining (Part I)

João Gama

Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Goal: Continuously maintain a clustering structure from evolving time series data streams.

- Incremental clustering of streaming time series;
- Constructs a hierarchical tree-shaped structure of clusters
- Using a top-down strategy.
- The leaves are the resulting clusters: each leaf groups a set of variables.
- The union of the leaves is the complete set of variables.
- The intersection of leaves is the empty set.

Online Divisive-Agglomerative Clustering

Key Concept – Diameter of a cluster: the maximum distance between two variables.

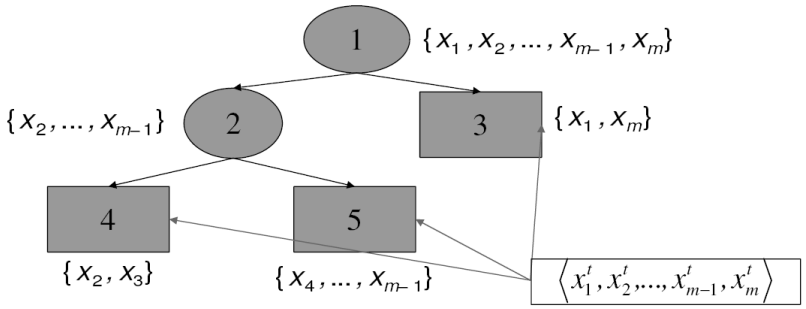
- Incremental system to **monitor clusters' diameters**
- Performs hierarchical clustering of **first-order differences**
- Can **detect changes** in the clustering structure
- Two Operators:
  - Splitting: expand the structure
  - Agglomeration: contract the structure
- Splitting and agglomerative criteria are supported by a confidence level given by the **Hoeffding bounds**.

Main Algorithm [Rodrigues, Gama, 2006]

- ForEver
  - Read Next Example
  - Compute first order differences
  - For all the clusters
    - Update the sufficient statistics
  - Time to Time
    - Verify Merge Clusters
    - Verify Expand Cluster

Feeding ODAC

Each example is processed once.
Only sufficient statistics **at leaves** are updated.
Sufficient Statistics: a triangular matrix of the correlations between variables in a leaf.
Released when a leaf expands to a node.



$C_1 = \{x_2, x_3\}$, $C_2 = \{x_4, \ldots, x_{m-1}\}$, $C_3 = \{x_1, x_m\}$

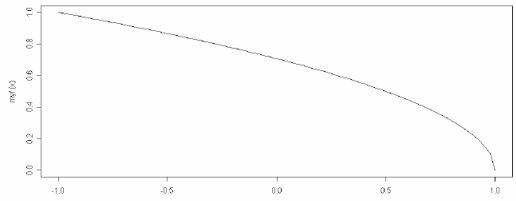
Similarity Distance

Distance between time Series: $rnomc(a,b) = \sqrt{\frac{1-corr(a,b)}{2}}$
where $corr(a,b)$ is the Pearson Correlation coefficient:

$$corr(a,b) = \frac{P - \frac{AB}{n}}{\sqrt{A_2 - \frac{A^2}{n}}\sqrt{B_2 - \frac{B^2}{n}}}$$

The *sufficient statistics* needed to compute the correlation are easily updated at each time step:

$A = \sum a_i,\ B = \sum b_i,\ A_2 = \sum a_i^2,\ B_2 = \sum b_i^2,\ P = \sum a_i b_i$



Splitting Criteria

When should we expand a leaf?
Let

- $d_1 = d(a, b)$ the farthest distance
- d_2 the second farthest distance

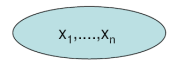
Hoeffding bound:

Split if $d_1 - d_2 > \epsilon$ with $\epsilon = \sqrt{\frac{R^2 ln(1/\delta)}{2n}}$
where R is the range of the random variable; δ is a user confidence level, and n is the number of observed data points.

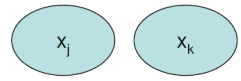
State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Expanding a Leaf

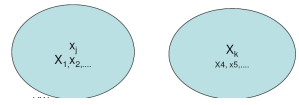
Step 1 Find Pivots:
$x_j, x_k : d(x_j, x_k) > d(a, b)$
$\forall a, b \neq j, k$



Step 2 If Splitting Criteria applies:
Generate two new clusters.

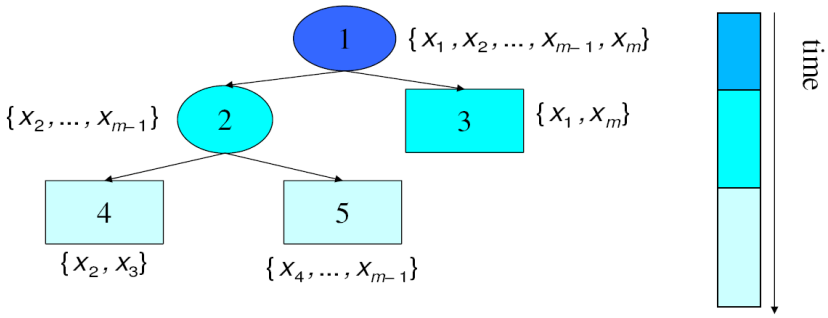


Step 3 Each new cluster attract nearest variables.



Multiple Time-Windows

A multi-window system: each node (and leaves) receive examples from different time-windows.



State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Change Detection

Time Series Concept Drift:

- Change in the distribution generating the observations.
- Clustering Analysis Concept Drift
  - Changing the way time series correlate with each other
  - Change in he cluster Structure.

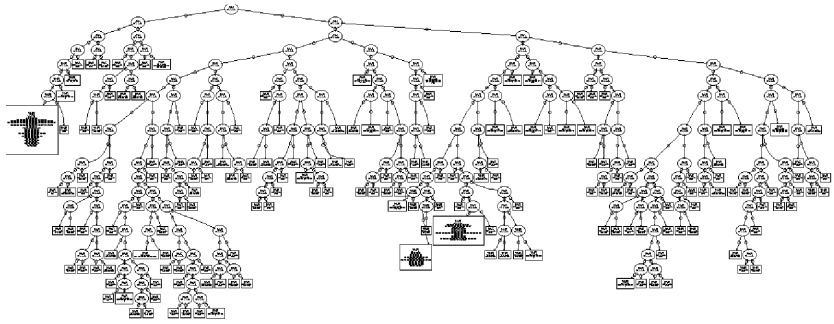
The Splitting Criteria guarantees that cluster's diameters monotonically decrease.

- Assume Clusters: c_j with descendants c_k and c_s.
- If $diameter(c_k) - diameter(c_j) > \epsilon$ OR $diameter(c_s) - diameter(c_j) > \epsilon$
  - Change in the correlation structure!
  - Merge clusters c_k and c_s into c_j.

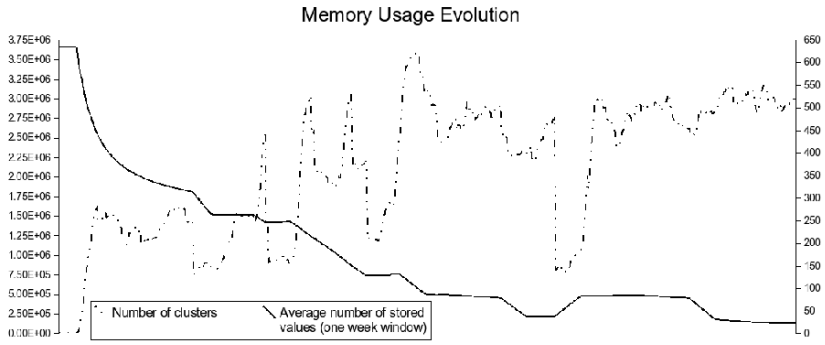
Properties of ODAC

- For stationary data the cluster's diameters monotonically decrease.
- **Constant update time/memory consumption** with respect to the number of examples!
- Every time a **split** is reported
  - the **time** to process the next example **decreases**, and
  - the **space** used by the new leaves is **less than** that used by the parent.

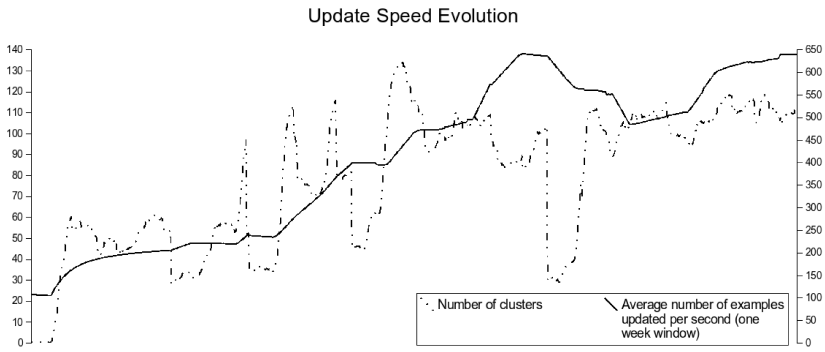
A snapshot - 1 year data, 2500 variables



Memory Usage



Speed in Processing Time



Outline

Desirable properties:

- Processing each example:
  - Small constant time
  - Fixed amount of main memory
  - Single scan of the data
  - Without (or reduced) revisit old records.
  - Eventually using a sliding window of more recent examples
- Processing examples at the speed they arrive
- Classifiers at anytime
- Ideally, produce a model equivalent to the one that would be obtained by a batch data-mining algorithm
- Ability to detect and react to concept drift

Very Fast Decision Trees

Mining High-Speed Data Streams, P. Domingos, G. Hulten; KDD00

The base Idea:
A small sample can often be enough to choose the optimal splitting attribute

- Collect sufficient statistics from a small set of examples
- Estimate the merit of each attribute
- Use Hoeffding bound to guarantee that the best attribute is really the *best*.
  - Statistical evidence that it is better than the second best

State-of-the-Art in Data Stream Mining (Part I)
João Gama
Outline
Motivation
Data Streams
Basic Methods
Change Detection
Predictive Learning
Clustering Data Streams
Predictive Models from Data Streams
Decision Trees
Neural Networks

Very Fast Decision Trees: Main Algorithm

- **Input:** δ desired probability level.
- **Output:** $\mathcal{T}$ A decision Tree
- **Init:** $\mathcal{T} \leftarrow$ Empty Leaf (Root)
- While (TRUE)
  - Read next Example
  - Propagate Example through the Tree from the Root till a leaf
  - Update Sufficient Statistics at leaf
  - If $leaf(\#examples) > N_{min}$
    - 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 Strategies

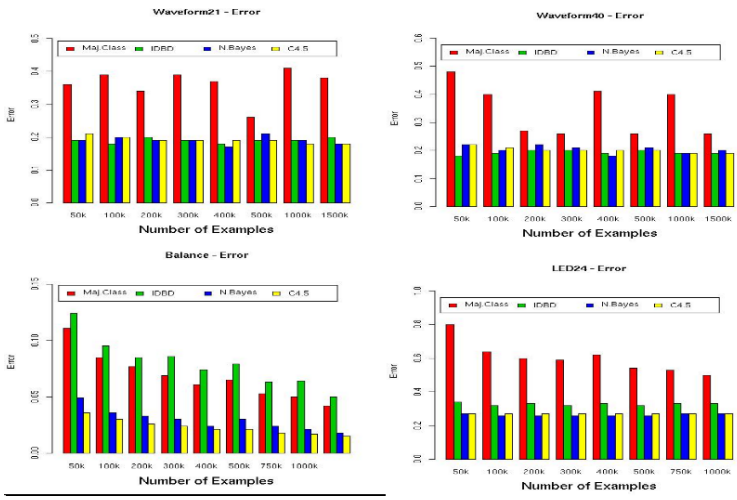
Accurate Decision Trees for mining high-speed Data Streams, J.Gama, R. Rocha; KDD03

- To classify an unlabelled example:
  - The example traverses the tree from the root to a leaf
  - It is classified using the information stored in that leaf

Two classification strategies:

- The standard strategy use ONLY information about the class distribution: $P(Class_i)$
- A more informed strategy, use the sufficient statistics $P(x_j|Class_i)$
  - Classify the example in the class that maximizes $P(C_k|\overrightarrow{x})$
  - Naive Bayes Classifier: $P(C_k|\overrightarrow{x}) \propto P(C_k) \prod P(x_j|C_k)$.
    - VFDT stores sufficient statistics of hundred of examples in leaves.

VFDT: Illustrative Evaluation



VFDT: Analysis

- Low variance models:
  Stable decisions with statistical support.
- Low overfiting:
  Examples are processed only once.
- **Convergence**: VFDT becomes asymptotically close to that of a batch learner. The expected disagreement is δ/p; where p is the probability that an example fall into a leaf.

Neural-Nets and Data Streams

Multilayer Neural Networks

- A general *Function approximation* method;
- A 3 layer ANN can approximate any continuous function with arbitrary precision;
- Fast Train and Prediction:
  - Each example is propagated once
  - The Error is back-propagated once
- No overfitting
  - First: Prediction
  - Second: Update the Model
- Smoothly adjust to gradual changes

State-of-the-art in Data Stream Mining (Part II)

Mohamed Medhat Gaber
Tasmanian CSIRO ICT Centre
Mail: GPO Box 1538, Hobart, TAS 7001, Australia
E-mail: Mohamed.Gaber@csiro.au

Outline

- Frequent Pattern Mining in Data Streams
- Time Series Analysis in Data Streams
- Data Stream Mining Systems
- Applications of Mining Data Streams
- Future Directions
- Open Issues
- Future Vision
- Resources

Outline

- Frequent Pattern Mining in Data Streams
- Time Series Analysis in Data Streams
- Data Stream Mining Systems
- Applications of Mining Data Streams
- Future Directions
- Open Issues
- Future Vision
- Resources

Introduction to Frequent Pattern Mining

- Frequent pattern mining refers to finding patterns that occur greater than a pre-specified threshold value.
- Patterns refer to items, itemsets, or sequences.
- Threshold refers to the percentage of the pattern occurrences to the total number of transactions. It is termed as *Support*

Introduction to Frequent Pattern Mining (Cont'd)

- Finding frequent patterns is the first step for the discovery of association rules in the form of $A \rightarrow B$.
- Apriori algorithm represents a pioneering work for association rules discovery
  - R Agrawal and R Srikant, Fast Algorithms for Mining Association Rules. In Proc. of the 20th International Conference on Very Large Databases, Santiago, Chile, September 1994
- An important step towards improving the performance of association rules discovery was FP-Growth
  - J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD'00), Dallas, TX, May 2000.

Introduction to Frequent Pattern Mining (Cont'd)

- Many measurements have been proposed for finding the strength of the rules.
- The very frequently used measure is confidence.
- Confidence refers to the probability that set B exists given that A already exists in a transaction.
  - Confidence (A→B) = Support (AB) / Support (A)

Frequent Pattern Mining in Data Streams

- The process of frequent pattern mining over data streams differs from the conventional one as follows:
  - The technique should be linear or sublinear (You Have Only One Look).
  - Frequent items (heavy hitters) and itemsets are often the final output.

Frequent Items (Heavy Hitters) in Data Streams

- Manku and Motwani have two master algorithms in this area:
  - Sticky Sampling
  - Lossy Counting

G. S. Manku and R. Motwani. **Approximate Frequency Counts over Data Streams**, in Proceedings of the 28th International Conference on Very Large Data Bases (VLDB), Hong Kong, China, August 2002.

Sticky Sampling

- Sticky sampling is a probabilistic technique.
- The user inputs three parameters
  - Support (s)
  - Error (ε)
  - Probability of failure (δ)
- A simple data structure is maintained that has entries of data elements and their associated frequencies (e, f).
- The sampling rate decreases gradually with the increase in the number of processed data elements.

$$t = \frac{1}{\varepsilon}\log(s^{-1}\delta^{-1}).$$

Sticky Sampling (Cont'd)

- For each incoming element in a data stream, the data structure is checked for an entry.
  - If an entry exists, then increment the frequency
  - Otherwise sample the element with the current sampling rate.
    - If selected, then add a new entry, else the element is ignored.
- With every change in sampling rate, a unbiased coin toss is done for each entry with decreasing the frequency with every unsuccessful coin toss.
  - If the frequency goes down to zero, the entry is released.

Lossy Counting

- Lossy counting is a deterministic technique.
- The user inputs two parameters
  - Support (s)
  - Error (ε)
- The data structure has one more attribute for each entry than the sticky sampling technique (e, f, Δ) where Δ is the maximum possible error in f.
- The stream is conceptually divided into buckets with a width w = 1/ ε.
- Each bucket is labelled by a value of N / w, where N starts from 1 and increases by 1.

Lossy Count (Cont'd)

- For a new incoming element, the data structure is checked
  - If an entry exists, then increment the frequency
  - Otherwise, add a new entry with $\Delta = b_{current} - 1$ where $b_{current}$ is the current bucket label.
- When switching to a new bucket, all entries with $f + \Delta < b_{current}$ are deleted.
- Lossy Count outperforms Sticky Sampling in practice.

Frequent Itemsets in Data Streams

- Manku and Motwani has extended Lossy Counting to find frequent itemsets.

  G. S. Manku and R. Motwani. **Approximate Frequency Counts over Data Streams**, in Proceedings of the 28th International Conference on Very Large Data Bases (VLDB), Hong Kong, China, August 2002.

- The technique follows the same steps with batch processing of transactions according to memory availability.
- All subsets of the stored batch are checked and pruned.
- If the frequency of a new entry is greater than the number of buckets currently in memory, then a new entry is added to the data structure.

Outline

- Frequent Pattern Mining in Data Streams
- Time Series Analysis in Data Streams
- Data Stream Mining Systems
- Applications of Mining Data Streams
- Future Directions
- Open Issues
- Future Vision
- Resources

Introduction to Time Series Analysis

- Time Series Analysis refers to applying different data analysis techniques on measurements acquired over temporal basis.
- Data analysis techniques recently applied on time series include clustering, classification, indexing, and association rules.
- The focus of classical time series analysis was on forecasting and pattern identification

Introduction to Time Series Analysis (Cont'd)

- Similarity measures over time series data represent the main step in time series analysis.
- Euclidean and dynamic time warping represent the major similarity measures used in time series.
- Longer time series could be represent computationally hard for the analysis tasks.
- Different time series representations have been proposed to reduce the length of a time series.

Time Series Analysis in Data Streams

- When data elements (records) in a data stream are processed based on their temporal dimension, we consider the process as time series analysis.
- Time series analysis in data streams are different in two aspects:
  - Several data points are considered to be an entry.
  - The analysis is done in real-time as opposed to traditional time series analysis.

Symbolic ApproXimation (SAX)

- SAX is a fast symbolic approximation of time series.
  - J. Lin, E. Keogh, S. Lonardi, and B. Chiu, **A Symbolic Representation of Time Series, with Implications for Streaming Algorithms**, in proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, San Diego, CA. June 13, 2003.
- It allows a time series with a length n to be transformed to an approximated time series with an arbitrarily length w, where w <<n.
- SAX follows three main steps:
  - Piecewise Aggregate Approximation (PAA)
  - Symbolic Discretization
  - Distance measurement
- SAX is generic and could be applied to any time series analysis technique.

Piecewise Aggregate Approximation (PAA)

- A time series with size n is approximated using PAA to a time series with size w using the following equation.

$$\bar{C}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} C_j$$

Where $\bar{c}_i$ is the i^{th} element in the approximated time series

Symbolic Discretization

- Breakpoints are calculated that produce equal areas from one point to another under Gaussian distribution.
  - A lookup table could be used.
- According to the output of PAA
  - If a point is less than the smallest breakpoint, then it is denoted as “a”.
  - Otherwise and if the point is greater than the smallest breakpoint and less than the next larger one, then it is denoted as “b”.
  - etc.

Distance Measurement

- The following distance measure is applied when comparing two different time series:

$$MINDIST(\hat{Q},\hat{C}) \equiv \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} (dist(\hat{q}_i,\hat{c}_i))^2}$$

- It returns the minimum distance between the original time series.
- A lookup table is calculated and used to find the distance between every two letters.

SAX (Cont'd)

- SAX has been applied to many data mining techniques including
  - Clustering (hierarchical and partitioning)
  - Classification (Nearest neighbour and decision trees)
  - Change detection
- SAX represents the state-of-the-art in time series data streams analysis due to its generality

Hot SAX

- SAX has been used to discover discords in time series. The technique is termed as Hot SAX.
  - Keogh, E., Lin, J. and Fu, A., HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. In the 5th IEEE International Conference on Data Mining, New Orleans, LA. Nov 27-30, 2005.
- Discords are the time series subsequences that are maximally different from the rest of the time series subsequences.
- It is 3 to 4 times faster than brute force technique.
- This makes it a candidate for data streaming applications

Hot SAX (Cont'd)

- The process starts with sliding widows of a fixed size over the whole time series to generate subsequence
- Each generated subsequence is approximated using SAX
- The approximated subsequence is then inserted in an array indexed according to its position in the original time series
- The number of occurrences of each SAX word is also inserted in the array.

Hot SAX (Cont'd)

- The array is then transformed to a tries where the leaf nodes represent the array index where the word appears.
- The two data structures (array and trie) complement each other.

Outline

- Frequent Pattern Mining in Data Streams
- Time Series Analysis in Data Streams
- Data Stream Mining Systems
- Applications of Mining Data Streams
- Future Directions
- Open Issues
- Future Vision
- Resources

Data Stream Mining Systems

- Diamond Eye
  - The aim of the project is to enable remote systems as well as scientists to extract patterns from spatial objects in real time image streams.
  - The system uses a high performance computational facility for processing the data mining request
  - The scientist uses a web interface that uses java applets to connect to the server that requests that images to perform the image mining process.

M. Burl, Ch. Fowlkes, J. Roden, A. Stechert, and S. Mukhtar, Diamond Eye: A distributed architecture for image data mining, in SPIE DMKD, Orlando, April 1999, pp. 197-206

Data Stream Mining Systems (Cont'd)

- MobiMine
  - It is a client/server PDA-based distributed data mining application for financial data streams.
  - The system prototype has been developed using a single data source and multiple mobile clients; however the system is designed to handle multiple data sources.
  - The server functionalities in the proposed system are data collection from different financial web sites and storage, selection of active stocks using common statistics methods, and applying online data mining techniques to the stock data.

Kargupta, H., Park, B., Pittie, S., Liu, L., Kushraj, D. and Sarkar, K, MobiMine: Monitoring the Stock Market from a PDA. ACM SIGKDD Explorations. January 2002. Volume 3, Issue 2. Pages 37--46. ACM Press

Data Stream Mining Systems (Cont'd)

- MobiMine (Cont'd)
  - The client functionalities are portfolio management using a mobile micro-database to store portfolio data and information about user's preferences, and construction of the WatchList and this is the first point of interaction between the client and the server.
  - The server computes the most active stocks in the market, and the client in turn selects a subset of this list to construct the personalized WatchList according to an optimization module.
  - The second point of interaction between the client and the server is that the server performs online data mining and then transforms the results using Fourier transformation and finally sends this to the client.
  - The client in turn visualizes the results on the PDA screen.

Data Stream Mining Systems (Cont'd)

- VEDAS
  - It stands for Vehicle Data Stream Mining System
  - It is a ubiquitous data stream mining system that allows continuous monitoring and pattern extraction from data streams generated on-board a moving vehicle.
  - The mining component is located on a PDA placed onboard the vehicle.
  - VEDAS uses online incremental clustering for modelling of driving behaviour.
  - Hillol Kargupta, Ruchita Bhargava, Kun Liu, Michael Powers, Patrick Blair, Samuel Bushra, James Dull, Kakali Sarkar, Martin Klein, Mitesh Vasa, and David Handy,

    VEDAS: A Mobile and Distributed Data Stream Mining System for Real-Time Vehicle Monitoring, Proceedings of SIAM International Conference on Data Mining 2004

Data Stream Mining Systems (Cont'd)

- EVE
  - It stands for EnVironment for On-Board Processing
  - It is used for astronomical data stream mining.
  - Data streams are generated from measurements of different on-board sensors.
  - Only interesting patterns are sent to the ground stations for further analysis preserving the limited bandwidth.

S. Tanner, M. Alshayeb, E. Criswell, M. Iyer, A. McDowell, M. McEniry, K. Regner, EVE: On-Board Process Planning and Execution**,** Earth Science Technology Conference, Pasadena, CA, Jun. 11 - 14, 2002

Data Stream Mining Systems (Cont'd)

- MAIDS
  - It stands for Mining Alarming Incidents of Data Streams.
  - The system can classify, cluster, count frequency and query over data streams.
  - It is a generic system as opposed to the other data stream mining systems that are application-based.

Y. D. Cai, D. Clutter, G. Pape, J. Han, M. Welge, and L. Auvil, MAIDS: Mining Alarming Incidents from Data Streams, (system demonstration), Proc. 2004 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'04), Paris, France, June 2004

Data Stream Mining Systems (Cont'd)

- Genie of the net
  - It is a mobile agent-based ubiquitous data mining for a context-aware health club for cyclists.
  - The process starts by collecting information from sensors and databases in order to recognize the needed information for the specific application.
  - This information includes user's context and other needed information collected by mobile agents.

S. Pirttikangas, J. Riekki, J. Kaartinen, J. Miettinen, S. Nissila, & J. Roning. Genie Of The Net: A New Approach For A Context-Aware Health Club. In Proceedings of Joint 12th ECML'01 and 5th European Conference on PKDD'01. September 3-7, 2001, Freiburg, Germany.

Data Stream Mining Systems (Cont'd)

- Genie of the net (Cont'd)
  - The main scenario for the health club system is that the user has a plan for an exercise.
  - All the needed information about the health such as heart rate is recorded during the exercise.
  - This information is analyzed using data mining techniques to advise the user after each exercise.

Outline

- Frequent Pattern Mining in Data Streams
- Time Series Analysis in Data Streams
- Data Stream Mining Systems
- Applications of Mining Data Streams
- Future Directions
- Open Issues
- Future Vision
- Resources

Applications of Mining Data Streams

- Analysis of biosensor measurements around a city for security reasons
- Analysis of simulation results and on-board sensors in scientific laboratories and spacecrafts has its potential in changing the mission plan or the experimental settings in real time
- Analysis of web logs and web clickstreams

Applications of Mining Data Streams (Cont'd)

- Real-time analysis of data streams generated from stock markets
- A travelling salesman performing customer profiling
- Continuous monitoring and analyzing of status information received for intrusion detection or laboratory experiments

Applications of Mining Data Streams (Cont'd)

- Analysis of data from sensors in moving vehicles to prevent fatal accidents through early detection
- Performing in-network mining of data streams in a wireless sensor network
- Prediction of climate, weather and geophysical hazards.

Outline

- Frequent Pattern Mining in Data Streams
- Time Series Analysis in Data Streams
- Data Stream Mining Systems
- Applications of Mining Data Streams
- Future Directions
- Open Issues
- Future Vision
- Resources

Future Directions

- Developing analysis algorithms for sensor networks to serve a number of real-time critical applications. SenosrNet (www.sensornet.gov) is one example in this direction.
- Online medical, scientific and biological analysis using data generated from medical, biological instruments and various tools employed in scientific laboratories.

Future Directions (Cont'd)

- Hardware solutions to small devices emitting or receiving data streams in order to enable high performance computation on small devices.
- Developing software architectures that serve the streaming applications.

Outline

- Frequent Pattern Mining in Data Streams
- Time Series Analysis in Data Streams
- Data Stream Mining Systems
- Applications of Mining Data Streams
- Future Directions
- Open Issues
- Future Vision
- Resources

Open Issues

- Interactive mining environment to satisfy user requirements
- The integration between data stream management systems and the ubiquitous data stream mining approaches
- Matching techniques with real world applications
- Data stream pre-processing

Open Issues (Cont'd)

- Model overfitting
- Data stream mining technology
- Real-time accuracy evaluation
- Theoretical foundations of data stream computing

Outline

- Frequent Pattern Mining in Data Streams
- Time Series Analysis in Data Streams
- Data Stream Mining Systems
- Applications of Mining Data Streams
- Future Directions
- Open Issues
- Future Vision
- Resources

Future Vision

- Wireless Sensor Networks provide environmental information.
- Building data mining models from this information according to the current context would contribute to build smart environments.
- Context-aware computing, data stream querying/mining, and wireless sensor networks will bring together the potential of research in this direction
- Examples include: Smart marketplace, smart workplace, smart vehicle and smart house.

The Data Stream Phenomenon

- Highly detailed, automatic, rapid data feeds.
  - Radar: meteorological observations.
  - Satellite: geodetics, radiation,.
  - Astronomical surveys: optical, radio,.
  - Internet: traffic logs, user queries, email, financial,
  - Sensor networks: many more *observation points* ...
- Most of these data will never be seen by a human!
- Need for near-real time analysis of data feeds.
- Monitoring, intrusion, anomalous activity Classification, Prediction, Complex correlations, Detect outliers, extreme events, fraud, ...

The Past of Machine Learning

In the last two decades, machine learning research and practice focus in batch learning using small datasets.

- The whole training data is available to the algorithm, that outputs a decision model after processing the data multiple times.
- This practice assumes that examples were generated at random accordingly to some stationary probability distribution.
- Most learners use a greedy, hill-climbing search in the space of models.
- Learning from small datasets: Emphasis in variance reduction.

What distinguishes current data sets from earlier ones is *automatic data feeds*. We do not just have people entering information into a computer. We have computers entering data into each other.

The Future of Machine Learning

Learning from small datasets: emphasis in variance reduction. Whats about large datasets?

- Increasing data = Variance reduction. Stable statistics estimators
- Learning from large datasets may be more effective using algorithms that places greater emphasis on bias management
- Solutions to these problems require
  - New Sampling and Randomize Techniques,
  - New Approximate, Incremental Algorithms,
  - Management the cost of Model's update and the Gains in Performance.
  - Incorporation of Change Detection Algorithms inside the Learning Process.

Resources

- First International Workshop on Knowledge Discovery from Data Streams (IWKDDS) at ECML/PKDD 2004 on September 24th, 2004, in Pisa, Italy.
  - Organized by:
    - Joao Gama, University of Porto, Portugal
    - Jesus S. Aguilar-Ruiz, University of Seville, Spain
  - Web: http://www.lsi.us.es/~aguilar/ecml2004/
- Second International Workshop on Knowledge Discovery from Data Streams (IWKDDS) at ECML/PKDD 2005 on October 10th, 2005, in Porto, Portugal.
  - Organized by:
    - Jesus S. Aguilar-Ruiz, University of Seville, Spain
    - Joao Gama, University of Porto, Portugal
  - Web: http://www.niaad.liacc.up.pt/~jgama/IWKDDS/

Resources (Cont'd)

- Third International Workshop on Knowledge Discovery from Data Streams (IWKDDS) at ICML 2006 on June 29th, 2006, at Carnegie Mellon University (CMU) in Pittsburgh, PA, USA.
  - Organized by:
    - Joao Gama, University of Porto, Portugal
    - Jesús S. Aguilar-Ruiz, University of Pablo de Olavide, Spain
    - Josep Roure, Carnegie Mellon University, US
  - Web: http://www.cs.cmu.edu/~jroure/iwkdds/iwkdds_icml06.html
- ECML/PKDD 2006 Workshop on Knowledge Discovery from Data Streams
  - Organized by:
    - João Gama,University of Porto, Portugal
    - Jesus S. Aguilar-Ruiz, University of Seville / University of Pablo de Olavide, Spain
    - Ralf Klinkenberg, University of Dortmund, Germany
  - Web: http://www.machine-learning.eu/iwkdds-2006/

Resources (Cont'd)

- International Workshop on Knowledge Discovery from Ubiquitous Data Streams
  - Organized by:
    - João Gama, University of Porto, Portugal
    - Mohamed Medhat Gaber, CSIRO ICT Centre, Australia
    - Jesus S. Aguilar-Ruiz, University of Seville and University of Pablo de Olavide, Spain
  - Web: http://www.niaad.liacc.up.pt/~iwkduds/
- ACM SAC – Data Streams Track (2004 – 2007) – papers could be found at ACM Portal

Resources (Cont'd)

- UCR Time Series Classification/Clustering Datasets
  - Maintained by:
    - Eamonn Keogh, UCR, US
  - Web: http://www.cs.ucr.edu/~eamonn/time_series_data/
- Mining Data Streams Bibliography
  - Maintained by:
    - Mohamed Medhat Gaber, CSIRO ICT Centre, Australia
  - Web: http://www.csse.monash.edu.au/~mgaber/WResources.htm

Master References

- **Books**
  - **Data Streams: Algorithms and Applications (Foundations and Trends in Theoretical Computer Science,)** by S. Muthukrishnan (Now Publishers)
  - **Data Streams: Models and Algorithms (Advances in Database Systems)** by Charu C. Aggarwal (Ed) (Springer)
  - **Learning from Data Streams: Processing Techniques in Sensor Networks** by Joao Gama and Mohamed Medhat Gaber (Eds) (Springer)
- **Seminal Surveys**
  - B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. **Models and Issues in Data Stream Systems,** in Proceedings of PODS, 2002.
  - Gaber, M, M., Zaslavsky, A., and Krishnaswamy, S., **Mining Data Streams: A Review**, in ACM SIGMOD Record, Vol. 34, No. 1, March 2005, ISSN: 0163-5808
  - S. Muthukrishnan, **Data streams: Algorithms and Applications**. Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms, 2003

Researchers

- Charu Aggarwal
- Jesús S. Aguilar-Ruiz
- Yun Chi
- Graham Cormode
- Pedro Domingos
- Wei Fan
- João Gama
- Venkatesh Ganti
- Minos N. Garofalakis
- Johannes Gehrke
- Sudipto Guha
- Jiawei Han
- Geoff Hulten

Researchers (Cont'd)

- Hillol Kargupta
- Eamonn Keogh
- Ralf Klinkenberg
- Nikos Koudas
- Jessica Lin
- Nina Mishra
- Rajeev Motwani
- Muthu Muthukrishnan
- Olfa Nasraoui
- Rajeev Rastogi
- Haixun Wang
- Qian Weining
- Philip S. Yu

Thanks for your attention!

More information:

Sensors J. Gama, R. Pederson; *Predictive Learning from Sensory Data*, Learning from Data Streams – Processing Techniques in Sensor Networks, Springer Verlag, 2007.

Streams Learning from Data Streams – Processing Techniques in Sensor Networks, Editores J. Gama and M. Gaber, Springer Verlag, 2007.

Streams S. Muthukrishnan, *Data Streams: Algorithms and Applications*, Now Publishers, 2003.

VFDT P. Domingos, G. Hulten; *Learning from Infinite Data in Finite Time*, Advances in Neural Information Processing Systems 14. Cambridge, MA: MIT Press, 2002

VFDT J. Gama, R. Fernandes, R. Rocha, *Decision Trees for Mining Data Streams* Intelligent Data Analysis, Vol. 10, Number 1, IOS Press, 2006.

ODAC P. P. Rodrigues, J. Gama and J. P. Pedroso. *ODAC: Hierarchical Clustering of Time Series Data Streams*. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, 2006.

Thanks for your attention!

State-of-the-Art in Data Stream Mining (Part I)

João Gama and Mohamed Gaber

Conclusions and Open Issues