Data Streams

P. Kosina

LIAAD-INESC Porto, FI MU Brno

Contents

Introduction

Basic Approaches and Models

Change Detection

Evaluation of Stream Mining Algorithms

Application Examples

Traditional (Off-line) Data Mining

- 'Data entered by **people** into the **computers**'
- Deals with limited training set size, the whole learning set is available
- Stationary distribution
- Speed of processing data is not the top priority factor
- Aims to make use of all the available data to create the best model possible
- Problems: high variance and over-fitting

Traditional (Off-line) Data Mining

Movie on DVD - you can analyze (evaluate) it, random access...

Was it good or bad?

Data Streams

- '**Computers** feed data into **computers**'
- Streams are potentially unbounded in size
- Examples are usually incoming very fast
- May have non-stationary distribution
- Processing speed per example is very important
- Utilize summarized information from data rather then all the information possible
- Represent most of the real problems
  - TCP/IP traffic, GPS data, emails, sensor networks etc.

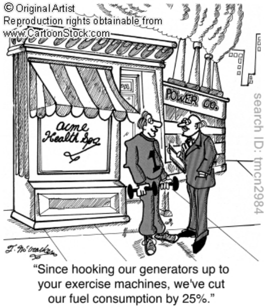
Data Streams

TV series episode - summarized information from previous episodes, sequential access (next episode to be aired)

Good or bad? You can evaluate the episode or what you have seen so far. Whole series only when finished.

Illustrative Example

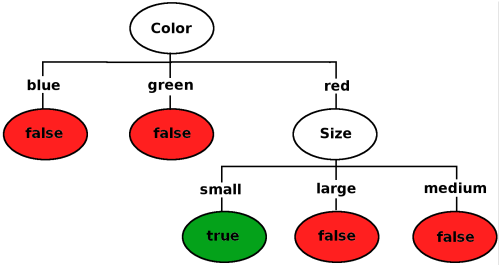
- Gama (2010)
- Electrical power distribution networks use sensors placed along network which produce streams of data at high speed
- Load forecasting is a very important task
    - Planning for different time horizons
    - Support economic decisions to buy/sell electricity (price variation depending on time of trade)


© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com
POWER CO.
Acme Health Spa
search ID: tmcn2984
"Since hooking our generators up to your exercise machines, we've cut our fuel consumption by 25%."

Illustrative Example

- Relevant data mining tasks
  - Cluster analysis
    - Identify profiles: urban, rural, industrial etc.
  - Predictive analysis
    - Predict the value measured by each sensor for different time horizons
    - Predict peaks in demand
  - Monitoring evolution
    - Detect changes in the behavior of sensors
    - Detect failures and abnormal activities
    - Identify peaks in demand
    - Identify critical points in load evolution

Basic Approaches and Models



Rules

1: Color = blue -> false
2: Color = green -> false
3: and(Color = red; Size = small) -> true
4: and(Color = red; Size = large) -> false
5: and(Color = red; Size = medium) -> false

Simple Statistics from Data Streams

- Incremental mean
  - Only two variables in memory to maintain
    - number of observations n
    - sum of values seen so far $\sum_{t=0}^{n} x_t$

  $$\overline{x}_n = \frac{(n-1) \times \overline{x}_{n-1} + x_n}{n}$$
- Incremental standard deviation
  - One more variable
    - sum of the squares $\sum_{t=0}^{n} x_t^2$

  $$\sigma_n = \sqrt{\frac{(\sum_{t=0}^{n} x_t^2) - \frac{(\sum_{t=0}^{n} x_t)^2}{n}}{n-1}}$$
- You can directly apply these statistics to compute incremental Naïve Bayes

Simple Statistics from Data Streams

- Correlation coefficient
  - Given two streams x and y we need to maintain
    - sum of each stream $\sum_{t=0}^{n} x_t$ and $\sum_{t=0}^{n} y_t$
    - sum of the squares $\sum_{t=0}^{n} x_t^2$ and $\sum_{t=0}^{n} y_t^2$
    - sum of the crossproduct $\sum_{t=0}^{n}(x_t \times y_t)$

$$corr(a,b) = \frac{\sum_{t=0}^{n}(x_t \times y_t) - \frac{\sum_{t=0}^{n} x_t \times \sum_{t=0}^{n} y_t}{n}}{\sqrt{\sum_{t=0}^{n} x_t^2 - \frac{\sum_{t=0}^{n} x_t^2}{n}}\sqrt{\sum_{t=0}^{n} y_t^2 - \frac{\sum_{t=0}^{n} y_t^2}{n}}}$$

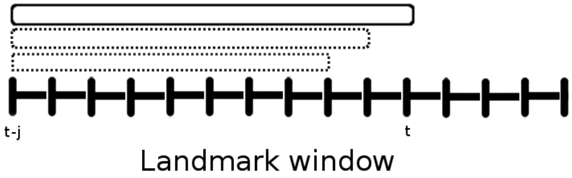
Sliding Windows

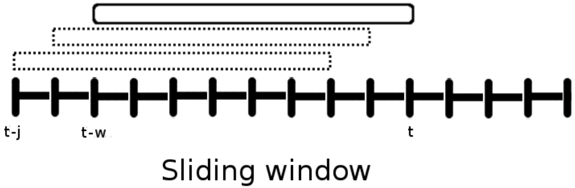
- Situations where *r*ecent past is more interesting than *c*omplete history
- FIFO structure: element x_t observed, element x_{t-w} forgotten; w is window size
- Two basic types
  - **Sequence based**
    - size defined by number of observations
  - **Timestamp based**
    - size defined by duration: elements with timestamp within a time interval T
- Necessary to maintain all the examples within the window in memory
  - e.g., $S = \sum_{t=101}^{150} x_t^2$ updated as $S_{i+1} = S_i + x_{151}^2 - x_{101}^2$

Sliding Windows

- Many variations of windows
  - Landmark window
    - ’growing’ window
    - typically, landmark is a new day
  - Natural tilted time window
  - Logarithmic tilted time window
    - different granularity of stored information (different levels of approximation)
    - recent history more precise (more points), less points for older
  - Adaptive sliding window

Sliding Windows





Reducing Data Size

- Alternative to sliding windows are data reduction techniques
- Sampling
  - Common method to reduce data size and computational costs
  - How to obtain an unbiased sample of data?
    - random sampling
    - reservoir sampling
    - ...
- Synopsis and histograms
- Data transformation: Wavelets, Discrete Fourier Transformation

Data Stream Classifiers

- It must require small constant time per record
- It must use only a fixed amount of main memory
- It must be able to build a model using at most one scan of the data
- It must make a usable model available at any point in time since it may never be done processing

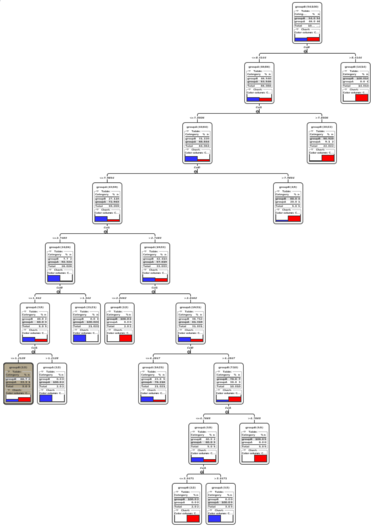
Data Stream Classifiers

- It should ideally produce a model that is equivalent (or nearly identical) to the one that would be obtained by corresponding ordinary database mining algorithm, operating without the above constraints.
- When the data-generating phenomenon is changing over time (e.g. when concept drift is present), the model at any time should be up-to-date, but also include all the information from the past that has not become outdated.

Decision Trees

- Tree - directed acyclic graph
- Node
  - *D*ecision node - has two or more successors, condition base on attribute values
  - *L*eaf node - class label
- High degree of interpretability
- Divide-and-conquer strategy
  - Complex problem recursively divided into simpler sub-problems
  - Solutions combined to provide solution of the complex problem

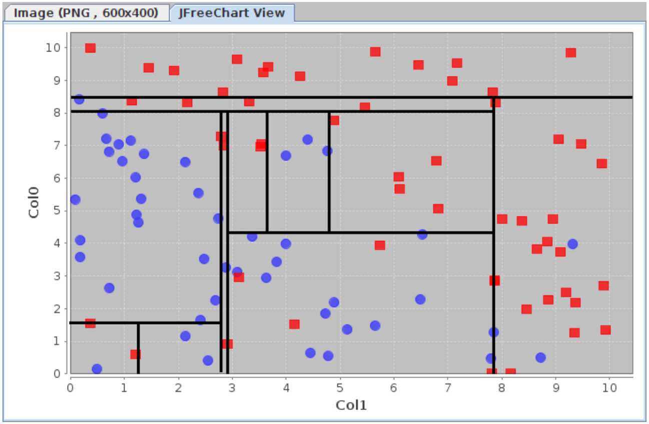
Decision Trees - tree structure



Decision Trees

- Conditions in decision nodes include a splitting attribute, branches are defined by operator (e.g., $\leq, =, \in$ etc.) and value from attribute's domain
    - e.g., **Color** $=$ green
- Test creates hyperplane orthogonal to axis of tested attribute and parallel to all the others
- Each leaf corresponds to a region (hyper-rectangle)
- Regions are mutually exclusive and exhaustive (i.e., fully cover the instance space)

Decision Trees - instance space



Hoeffding Trees a.k.a Very Fast Decision Trees

- Domingos (2000)
- Tree grows over time
- Each incoming example traverses from root to leaf and updates sufficient statistic
    - Statistics for heuristic function to compute the merit of split tests
- Leaf node is replaced by decision node when indicated by statistical support
    - i.e., enough evidence that one split is superior to the others

Fruit Data

VFDT Statistics - Nominal Attributes

- $\vec{v}_{Ai} =< n_1, n_2, \ldots, n_k >$,
  - $\vec{v}_{Ai}$ vector for value i of attribute A of counts for each class $1, 2, \ldots, k$

Table: Attribute Color

	Class		
Value	Apple	Plum	Strawberry
Red	20	0	10
Blue	0	10	0
Green	10	0	0

- Summary of observed instances
- And provides counts necessary to compute entropy (or other measures)

VFDT Statistics - Numerical Attributes

- Numerical attributes are very common and are more challenging to handle
- Different methods for deciding cut point with different demands
- On-line methods avoid sort operation
- Examples of Techniques
    - Gaussian distribution
    - Discriminant analysis
    - Exhaustive method

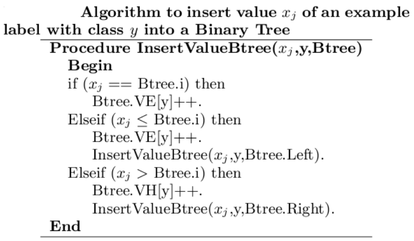
VFDT Statistics - Exhaustive method

- Gama et al. (2003)
- Each leaf contains binary tree structure for each attribute A
- Node in binary tree is identified by value i of the attribute
- Node has two vectors of dimension k, VE and VH, containing counts (for each class) of values $\leq i$ and $> i$ that passed via the node
- To evaluate merit of given attribute it computes information gain for each possible cut point

VFDT Statistics - Exhaustive method computation

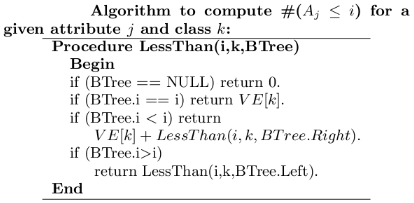
- Minimize:
  $info(A_j(i)) = P(A_j \leq i) \times iLow(A_j(i)) + P(A_j > i) \times iHigh(A_j(i))$
  i is the value of cut point, $iLow(A_j(i))$ information of $A_j \leq i$ and $iHigh(A_j(i))$ information of $A_j > i$
- $iLow(A_j(i)) = -\sum_K P(K = k|A_j \leq i) \times \log_2(P(K = k|A_j \leq i))$
- $iHigh(A_j(i)) = -\sum_K P(K = k|A_j > i) \times \log_2(P(K = k|A_j > i))$

VFDT Statistics - Exhaustive method algorithms



Algorithm to insert value x_j of an example label with class y into a Binary Tree

```
Procedure InsertValueBtree($x_j$,y,Btree)
  Begin
  if ($x_j$ == Btree.i) then
      Btree.VE[y]++.
  Elseif ($x_j \leq$ Btree.i) then
      Btree.VE[y]++.
      InsertValueBtree($x_j$,y,Btree.Left).
  Elseif ($x_j >$ Btree.i) then
      Btree.VH[y]++.
      InsertValueBtree($x_j$,y,Btree.Right).
End
```



Algorithm to compute $\#(A_j \leq i)$ for a given attribute j and class k:

```
Procedure LessThan(i,k,BTree)
  Begin
  if (BTree == NULL) return 0.
  if (BTree.i == i) return $VE[k]$.
  if (BTree.i < i) return
      $VE[k] + LessThan(i, k, BTree.Right)$.
  if (BTree.i>i)
      return LessThan(i,k,BTree.Left).
End
```

Hoeffding Bound

- *After n independent observations of a real-valued variable r with range R, HB ensures with confidence $1-\delta$ that the true mean of r is at least $\overline{r}-\epsilon$ where $\overline{r}$ is the observed mean.*
- Statistical guarantee that one split attribute is better than others
- It decides sufficient sample size of observed examples
- Independent of distribution, but requires more examples
- $H(\cdot)$ evaluation function, R range ($R=1$ for probability, $R=\log_2(\#classes)$ for information gain), confidence $1-\delta$
- $\epsilon=\sqrt{\frac{R^2\ln(1/\delta)}{2n}}$
- After n observed instances in leaf l, x_a and x_b are the attributes with highest $H(\cdot)$ respectively

Hoeffding Bound

- If $\mathbf{H}(\mathbf{x_a}) - \mathbf{H}(\mathbf{x_b}) > \epsilon$, x_a is the best split attribute with confidence $1 - \delta$ otherwise need to observe more examples
- Problem 1: two attributes are equally 'important' and constantly achieve the same $H(\cdot)$
    - Solution: tie-breaking constant. Split for small enough ϵ
- Problem 2: computationally demanding evaluate split with every example
    - Solution: compute after number of observations. Not likely to change with every example.

Extensions

- Popular classifier ⇒ many variation, extensions
- Change reaction (Hulten et al., 2001)
- Functional leaves (Gama et al., 2003)
  - Exploit information in the leaves to use Naïve Bayes prediction
- Option trees (Holmes et al., 2003)
- ...and many more

Change Detection

Introduction to Drift

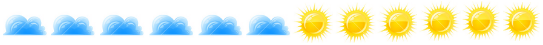
- Previous assumptions: examples generated at random from some stationary probability distribution
- Processes generating data in real world are dynamic and may change due to some external or internal influence
- 'Modern' approaches focus on adaptation
- Examples
  - Seasonal changes
  - Wear of tools in industrial processes
  - Gained resistance to certain type of antibiotics in medicine

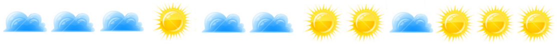
Specification of the Problem

- Data stream as sequences $< S_1, S_2, \ldots, S_m, \ldots >$ where each S_i is generated by some stationary distribution $\mathcal{D}_i$
    - Sequence S is context
- Problem is to detect change points
- Transition phase - examples from both distributions appear
- Example from $\mathcal{D}_i$ is noise for $\mathcal{D}_{i+1}$
- Noise vs. change, combine/balance robustness and sensitivity
- Persistence of distribution

Change Characteristics

- Can be caused by change in one or more attributes
- Or hidden changes (not reflected directly)
- Changes ca be of various rates - commonly two basic types
  - Concept drift - gradual change
  - Concept shift - abrupt/sudden change


Sudden drift

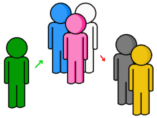

Gradual drift

Types of Approaches

Data management 'Forgetting'

Detection methods 'Detectors'

Decision model management 'Dynamic ensembles'



Data Management

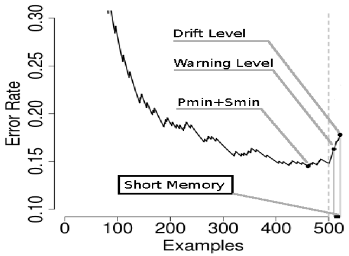
- Information in memory for maintaining consistent model
- Full memory
  - Statistics over all examples
  - Impact of examples weighted by age
    - e.g., exponential $w_\lambda(x) = \exp(-\lambda t_x)$, λ decay factor, if 0 all x have equal weight
  - **Gradual** forgetting
- Partial memory
  - Only most recent examples
  - Sliding windows (fixed, adaptive...)
    - small - fast adaptation to changes
    - large - slower adaptation, better performance for long stable contexts
  - **Abrupt** forgetting

Drift Detection Methods

- Previous methods are **blind**
- Detection methods are **explicit**, provide information about
  - Time points when change appeared
  - Small time windows in which drift occurred
- Two examples of DDM in predictive learning
  - Statistical Process Control
  - Page-Hinkley test

Statistical Process Control (SPC)

- Grant and Leavenworth (1996); Gama et al. (2004)
- Hypothesis: item distribution of data is stationary
  - Error-rate decreases with increasing number of examples
- Error-rate increases significantly $\Rightarrow$ not stationary distribution



Statistical Process Control (SPC)

- p_t is error-rate, s_t is standard deviation, p_{min}, s_{min} respective values corresponding to $min(p_t + s_t)$
- Out-of-Control

$$p_t + s_t > p_{min} + \alpha \times s_{min}$$

- In-Control

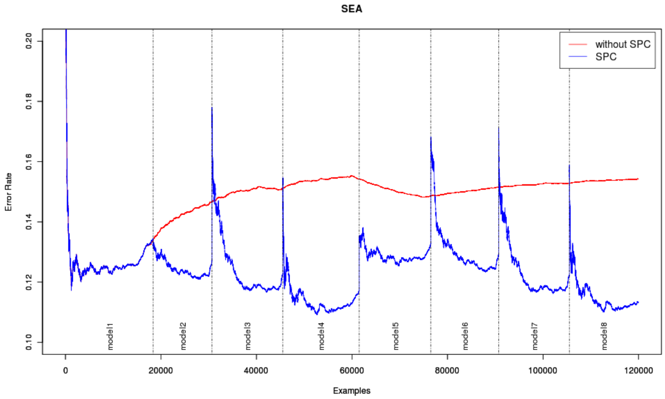
$$p_t + s_t < p_{min} + \beta \times s_{min}$$

- Warning

$$p_{min} + \alpha \times s_{min} > p_t + s_t > p_{min} + \beta \times s_{min}$$

- Constants α and β define confidence level ($\alpha = 3$ and $\beta = 2$ for 99% and 95% resp.)

Statistical Process Control (SPC)



Page-Hinkley

- Hinkley (1970)
- Change detection in signal processing
- Cumulative difference between error-rate values and its mean till n:

$$m_n = \sum_{t=1}^{n} (p_t - \overline{p}_t - \delta)$$

where $\overline{p}_t = \frac{1}{t} \sum_{i=1}^{t} p_i$ and parameter δ is magnitude of change not raising alarm.

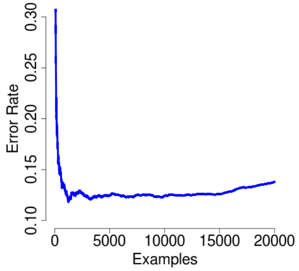
- Minimum:

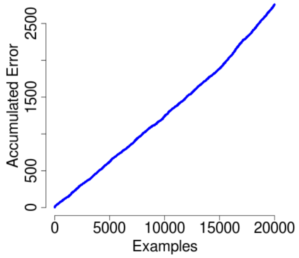
$$M_n = min\,(m_t, t = 1 \ldots n)$$

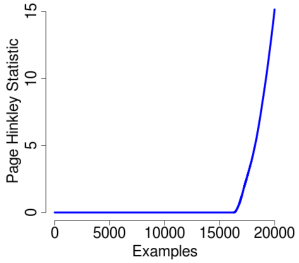
- Change reported when:

$$m_n - M_n > \lambda$$

Page-Hinkley







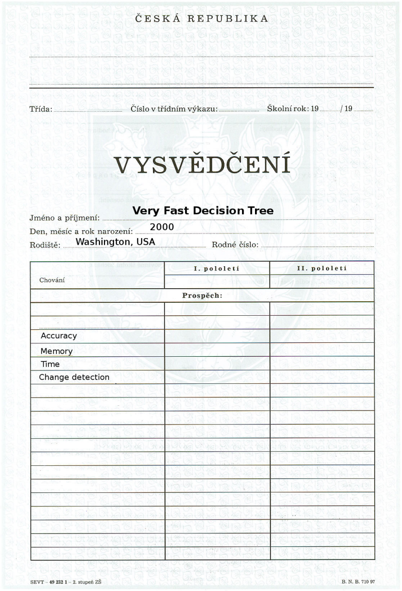
Decision model management

- Data generated from multiple distributions $\Rightarrow$ multiple decision models
- How many models keep in memory?
- Which models keep in memory?
- Dynamic Weighted Majority (DWM) algorithm

DWM

- Kolter and Maloof (2003)
- Ensemble weighted predicting using majority vote
- Models use same algorithm but trained on different data
- Each model has associated weight
- Incorrect prediction decreases weight of model
- Dynamically creates and removes models based on the performance
    - New model is added when global prediction is incorrect
    - Remove models when weight drops below given threshold

Evaluation of Stream Mining Algorithms



ČESKÁ REPUBLIKA

Třída: ……… Číslo v třídním výkazu: ……… Školní rok: 19 ……/ 19 ……

VYSVĚDČENÍ

Jméno a příjmení: **Very Fast Decision Tree**

Den, měsíc a rok narození: 2000

Rodiště: Washington, USA Rodné číslo:

	I. pololetí	II. pololetí
Chování		
Prospěch:		
Accuracy		
Memory		
Time		
Change detection		

SEVT – 49 232 1 – 2. stupeň ZŠ B. N. B. 710 97

Adaptive Learning Evaluation Methods

- Different goals
- High accuracy
- Efficiency
  - Time
  - Space
  - Resources - battery consumption
- Change detection quality
  - Detection delay
  - False positive/negative drift signals

Traditional Evaluation vs. Stream Evaluation

- Off-line training phase terminates within 'reasonable' time, learned model is the 'final product'
  - Testing after training process is finished
  - Various methods and metrics
- On-line task can never be finished learning
  - Must be interactive
  - Outputs from learned model are demanded during the process
  - Performance is required to be evaluated throughout the process

On-line Evaluation

- Gama et al. (2009)
- Holdout evaluation
  - Independent examples kept for test set
  - Current model is evaluated using holdout set periodically
- Prequential (**Pre**dictive Se**quential**) evaluation
  - With every training example, prediction is made before training the model (using only attribute values)
  - Cumulative loss is computed $S = \sum_{t=1}^{n} L(y_t, \hat{y}_t)$
  - Than train the model with given example
- Prequential evaluation can employ fading factor α to elevate the information about most recent performance or compute prequential error using a window of recent examples

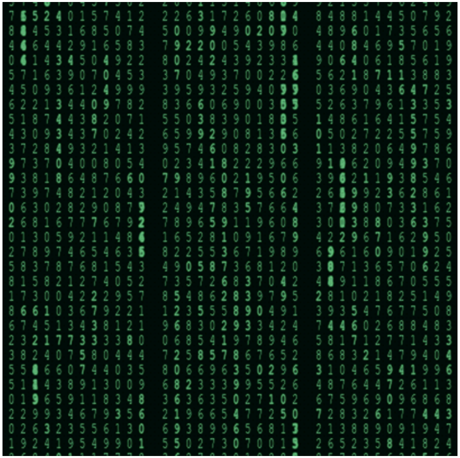
Fading Factor Prequential

- Prequential error with correction factor $E_t = \frac{S_t}{B_t} = \frac{L_t + \alpha \times S_{t-1}}{1 + \alpha \times B_{t-1}}$ where $B_1 = 1$

RAM Hours

- Bifet et al. (2010)
- Stream algorithms aim to achieve high time and space efficiency (more often than off-line algorithms) because of
    - Limited resources
    - Big data
- Time and space measure in one metric
- Every GB of RAM deployed for 1 hour
- Cloud computing rental cost inspiration

Application Examples



Sensors

- Gama and Gaber (2007)
- Usually small devices with very limited resources
  - Memory
  - Computational power
  - Power consumption (battery life)
- Prone to failures or to providing otherwise corrupted data
- Usually set to execute many readings per second (but highly depend on type of application)
- Processing can be done (to some extend) on the device to reduce communication costs

Sensors in Predicting Natural Disasters

- Include avalanche, drought, earthquake, flood, hurricane, tsunami, volcano, wildfire...
- Different sensors remote sensors, satellite images, radars, earth observing sensors
- Various related - measure temperature, rainfall, wind, land cover...
- Represent huge volumes of data incoming at fast rate
- Needs to be processed in real time to provide timely prediction of natural disaster threat

Food Sales

- Having too much of perishable goods $\Rightarrow$ loss when not sold
- Empty shelves $\Rightarrow$ unsatisfied customers
- Static model is not appropriate
  - Holidays
  - Season, temperature
  - Events (sport, cultural)
  - Advertisement, promotions
- Multiple models
- Contextual (meta-learning) approach
- External contextual data (weather etc.)

Marketing Applications

- Recommendation
  - User preferences and interests evolve
  - Influenced by media, season, work etc.
- Web clickstream analysis

Security Applications

- Intrusion detection / Fraud detection
  - Millions of connections/transactions need to be processed
  - New types of attacks appear
- Crime volume prediction
  - Planning support

Spam Detection

- Enormous amounts of Spam sent/received every day
- Feedback from users ('Report unrecognized Spam')
- Spam content is evolving (Viagra, fake Rolex, help a guy from Nigeria, etc.)

Summary

- Data stream setting
    - Most data generated nowadays are continuously incoming, potentially unbounded in size
    - Changes over time
- Characteristics of data stream mining algorithms
    - Sub-linear space, sequential access to data, constant time per example, any-time
    - Very Fast Decision Tree
- Reaction to changes
    - Importance of adaptation
    - Different techniques
- Algorithm assessment during the process
- Variety of applications (many to be explored yet!)

Conclusion

Stream mining is the way to go!

References I

Bifet, A., G. Holmes, B. Pfahringer, and E. Frank (2010). Fast perceptron decision tree learning from evolving data streams. In *Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part II*, PAKDD'10, Berlin, Heidelberg, pp. 299–310. Springer-Verlag.

Domingos, P. (2000). Mining high-speed data streams. pp. 71–80. ACM Press.

Gama, J. (2010). *Knowledge Discovery from Data Streams*. Chapman and Hall/CRC.

Gama, J. and M. Gaber (2007). *Learning from Data Streams – Processing techniques in Sensor Networks*. Springer.

Gama, J., P. Medas, G. Castillo, and P. Rodrigues (2004). Learning with drift detection. In *SBIA Brazilian Symposium on Artificial Intelligence*, pp. 286–295. Springer.

References II

Gama, J., R. Rocha, and P. Medas (2003). Accurate decision trees for mining high-speed data streams. In *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining*. ACM Press, New York, NY.

Gama, J., R. Sebastiao, and P. P. Rodrigues (2009). Issues in evaluation of stream learning algorithms. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, New York, NY, USA, pp. 329–338. ACM.

Grant, E. and R. Leavenworth (1996). *Statistical Quality Control*. McGraw-Hill.

Hinkley, D. (1970). Inference about the change point from cumulative sum-tests. *Biometrika 58*, 509–523.

Holmes, G., R. Kirkby, and B. Pfahringer (2003). Mining data streams using option trees. Technical report, States of America.

References III

Hulten, G., L. Spencer, and P. Domingos (2001). Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 97–106. ACM, New York, NY, USA.

Kolter, J. Z. and M. A. Maloof (2003). Dynamic weighted majority: A new ensemble method for tracking concept drift. In *Proceedings of the 3th International IEEE Conference on Data Mining*, pp. 123–130. IEEE Computer Society.