Mining Data Streams
Problems and Methods

JERZY STEFANOWSKI
Inst. Informatyki PP
Wersja dla TPD 2009
„Zaawansowana eksploracja danych”

Outline of the presentation

1. Motivations
2. Data streams requirements
3. Applications
4. Approximate queries and basic techniques
5. Classification in data streams
6. Clustering
7. Research directions

Acknowledgments

Some of slides are coming from lectures of:

- Mining High Speed Data Streams, talk by P. Domingos, G. Hulten, SIGKDD 2000.
- State of the art in data streams mining, talk by M.Gaber and J.Gama, ECML 2007.
- J.Han slides for a lecture on Mining Data Streams – available from Han's page on his book
- Myra Spiliopoulou, Frank Höppner, Mirko Böttcher - Knowledge Discovery from Evolving Data / tutorial at ECML 2008

The rest is based on my notes and experiments with my students (B.Szopka i M.Kmieciak)

Processing Data Streams: Motivation

- A growing number of applications generate streams of data
  - Performance measurements in network monitoring and traffic management
  - Call detail records in telecommunications
  - Transactions in retail chains, ATM operations in banks
  - Log records generated by Web Servers
  - Sensor network data
- Application characteristics
  - Massive volumes of data (several terabytes)
  - Records arrive at a rapid rate
  - Most od data will never be seen by a human!
  - Need for near-real time analysis of data feeds
- Goal: Mine patterns, process queries and compute statistics on data streams in real-time

What is a data stream ?

- Golab & Oszu (2003): *"A data stream is a real-time, continuous, ordered (implicitly by arrival time or explicitly by timestamp) sequence of items. It is impossible to control the order in which items arrive, nor is it feasible to locally store a stream in its entirety."*
- Structured records ≠ audio or video data
- Massive volumes of data, records arrive at a high rate

Timestamp	Puis. A (kW)	Puis. R (kVAR)	U 1 (V)	I 1 (A)
…	…	…	…	…
16/12/2006-17:26	5,374	0,498	233,29	23
16/12/2006-17:27	5,388	0,502	233,74	23
16/12/2006-17:28	3,666	0,528	235,68	15,8
16/12/2006-17:29	3,52	0,522	235,02	15
…	…	…	…	…

Characteristics of Data Streams

- Data Streams
  - Data streams — continuous, ordered, changing, fast, huge amount
  - Traditional DBMS — data stored in finite, persistent data sets
- Characteristics
  - Huge volumes of continuous data, possibly infinite
  - Fast changing and requires fast, real-time response
  - Data stream captures nicely our data processing needs of today
  - Random access is expensive — single scan algorithm (*can only have one look*)!
  - Store only the summary of the data seen thus far
  - Most stream data are at pretty low-level or multi-dimensional in nature, needs multi-level and multi-dimensional processing

Tradional vs. Stream Processing

	Tradional	Stream
No. of passes	Muliple	Single
Processing Time	Unlimited	Restricted
Memory Usage	Unlimited	Restricted
Type of Results	Accurate	Approximate
Distributed	No	Yes

Traditional DBMS versus DSMS

- Persistent relations
- One-time queries
- Random access
- "Unbounded" disk store
- Only current state matters
- No real-time services
- Relatively low update rate
- Data at any granularity
- Assume precise data
- Access plan determined by query processor, physical DB design

- Transient streams
- Continuous queries
- Sequential access
- Bounded main memory
- Historical data is important
- Real-time requirements
- Possibly multi-GB arrival rate
- Data at fine granularity
- Data stale/imprecise
- Unpredictable/variable data arrival and characteristics

Ack. From Motwani's PODS tutorial slides

Stream Data Applications - More

- Telecommunication calling records
- Business: credit card transaction flows
- Network monitoring and traffic engineering
- Financial market: stock exchange
- Engineering & industrial processes: power supply & manufacturing
- Sensor, monitoring & surveillance: video streams, RFIDs
- Security monitoring
- Web logs and Web page click streams
- Massive data sets (even saved but random access is too expensive)

More on applications of data stream processing

Applications

- Real-time monitoring/supervision of IS (Information Systems) generating large amounts of data
  - Computer network management
  - Telecommunication calls analysis (BI)
  - Internet applications (ebay, google, recommendation systems, click stream analysis)
  - Monitoring of power plants
- Generic software for applications where basic data is streaming data
  - Finance (fraud detection, stock market information)
  - Sensor networks (environment, road traffic, weather forecast, electric power consumption)

IP Network Measurement Data

- IP session data (collected using Cisco NetFlow)

Source	Destination	Duration	Bytes	Protocol
10.1.0.2	16.2.3.7	12	20K	http
18.6.7.1	12.4.0.3	16	24K	http
13.9.4.3	11.6.8.2	15	20K	http
15.2.2.9	17.1.2.1	19	40K	http
12.4.3.8	14.8.7.4	26	58K	http
10.5.1.3	13.0.0.1	27	100K	ftp
11.1.0.6	10.3.4.5	32	300K	ftp
19.7.1.2	16.5.5.8	18	80K	ftp

- AT&T collects 100 GBs of NetFlow data each day!

Network Data Processing

- Traffic estimation
  - How many bytes were sent between a pair of IP addresses?
  - What fraction network IP addresses are active?
  - List the top 100 IP addresses in terms of traffic
- Traffic analysis
  - What is the average duration of an IP session?
  - What is the median of the number of bytes in each IP session?
- Fraud detection
  - List all sessions that transmitted more than 1000 bytes
  - Identify all sessions whose duration was more than twice the normal
- Security/Denial of Service
  - List all IP addresses that have witnessed a sudden spike in traffic
  - Identify IP addresses involved in more than 1000 sessions

J. Gama – Sensor networks

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

- Sensors produce continuous flow of data at high speed:
  - Send information at different time scales;
  - 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.

Sensor networks

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

After P.Domingos's talk

Desiderata

- Small constant time per record
- Fixed amount of main memory
- At most one scan of data
- Results available anytime
- Results equivalent to standard algorithm
- Ability to handle time-changing phenomena

Data Stream – Querying Data

- Generally, algorithms compute approximate answers
  - Difficult to compute answers accurately with limited memory
- Approximate answers - Deterministic bounds
  - Algorithms only compute an approximate answer, but bounds on error
- Approximate answers - Probabilistic bounds
  - Algorithms compute an approximate answer with high probability
    - With probability at least $1-\delta$, the computed answer is within a factor ε of the actual answer

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.

Approximate Answers

- Actual answer is within 5 ± 1 with probability ≥ 0.9
- Trade off between sufficient accuracy of the answer and computational resources required to compute it.
- Probabilistic tail inequalities
  - Chebyshev Inequality
  - Chernoff Bounds
  - 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

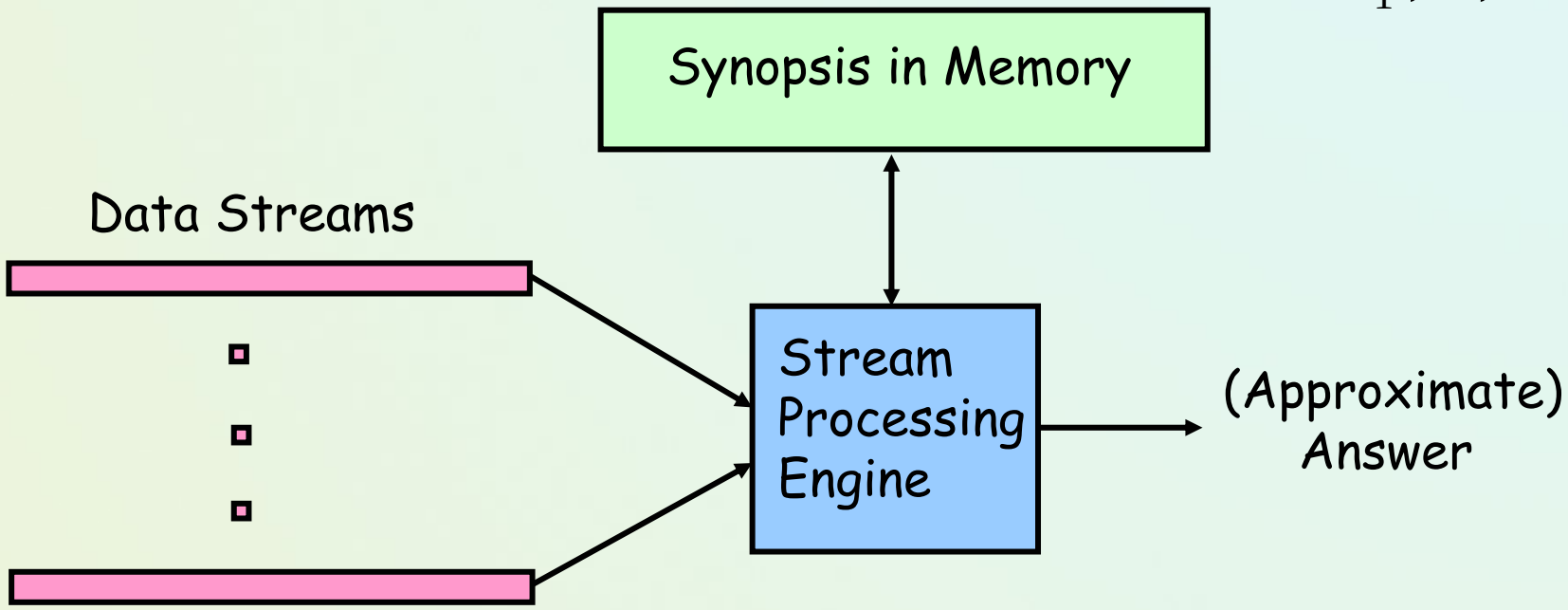
$$\epsilon = \frac{\sqrt{R^2 ln(1/\delta)}}{2n}$$

Basic Techniques for Stream Data Processing

- Major challenges
  - Keep track of a large universe
- Methodology
  - Synopses (trade-off between accuracy and storage)
  - Use *synopsis data structure*, much smaller ($O(\log^k N)$ space) than their base data set ($O(N)$ space)
  - Compute an *approximate answer* within a *small error range* (factor ε of the actual answer)
- Major methods
  - Random sampling
  - Histograms
  - Sliding windows
  - Sketches
  - Radomized algorithms

Computation Model for Approximate Answers

- A data stream is a (massive) sequence of elements: $e_1, \ldots, e_n$



- Stream processing requirements
  - Single pass: Each record is examined at most once
  - Bounded storage: Limited Memory (M) for storing synopsis
  - Real-time: Per record processing time (to maintain synopsis) must be low

Stream Data Processing Methods (1)

- Random sampling (but without knowing the total length in advance)
  - *Reservoir sampling*: maintain a set of *s* candidates in the reservoir, which form a true random sample of the element seen so far in the stream. As the data stream flow, every new element has a certain probability (*s*/N) of replacing an old element in the reservoir.
- Sliding windows
  - Make decisions based only on *recent data* of sliding window size *w*
  - An element arriving at time *t* expires at time *t* + *w*
- Histograms
  - Approximate the frequency distribution of element values in a stream
  - Partition data into a set of contiguous buckets
  - Equal-width (equal value range for buckets) vs. V-optimal (minimizing frequency variance within each bucket)
- Multi-resolution models
  - Popular models: balanced binary trees, micro-clusters, and wavelets

Sampling: Basics

- Idea: A small random sample S of the data often well-represents all the data
  - For a fast approx answer, apply "modified" query to S
  - Example: select agg from R where R.e is odd (n=12)

    Data stream: 9 3 5 2 7 1 6 5 8 4 9 1

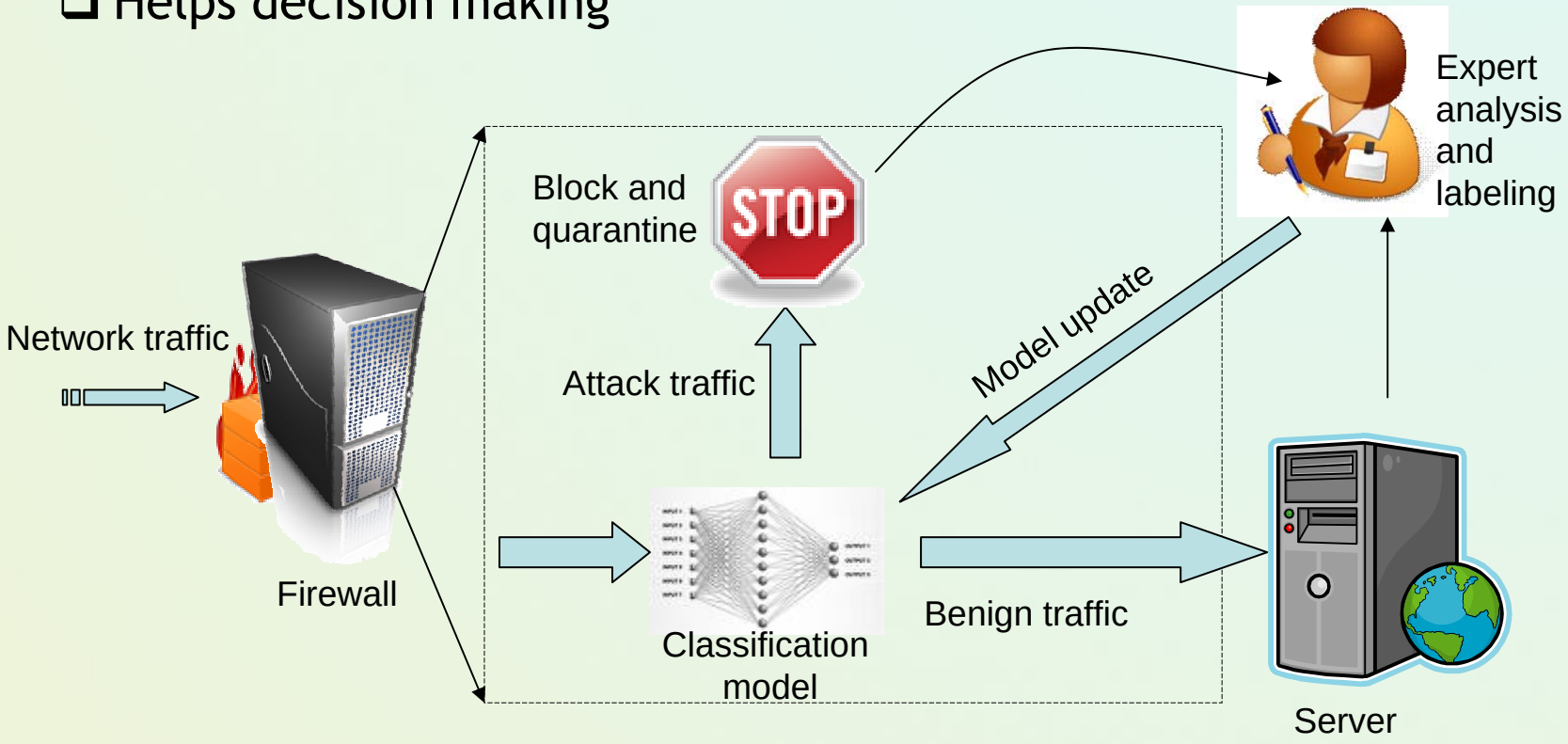
    Sample S: 9 5 1 8

    answer: 5
  - If agg is avg, return average of odd elements in S
  - If agg is count, return average over all elements e in S of
    - n if e is odd
    - 0 if e is even

Unbiased: For expressions involving count, sum, avg: the estimator is unbiased, i.e., the expected value of the answer is the actual answer

Data Stream Classification

- Uses past labeled data to build classification model
- predicts the labels of future instances using the model
- Helps decision making



Few previous research efforts

Older machine learning or AI directions

- Incremental learning vs. batch
  - Neural networks
  - Generalizations of k-NN (Aha's IBL)
  - Bayesian update
- Incremental versions of symbolic knowledge reconstruction
  - Decision trees ID5 (Utgoff)
  - Clustering – COBWEB
- Another heuristic evaluation measures
- Specific sampling for larger data
  - Windowing for trees
  - Sampling for k-means like clustering algorithms

And what …?

- However, we still need new approaches to real on-line learning from massive data sources!

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

Hoeffding Tree: Strengths and Weaknesses

- Strengths
  - Scales better than traditional methods
    - Sublinear with sampling
    - Very small memory utilization
  - Incremental
    - Make class predictions in parallel
    - New examples are added as they come
- Weakness
  - Could spend a lot of time with ties
  - Memory used with tree expansion
  - Number of candidate attributes

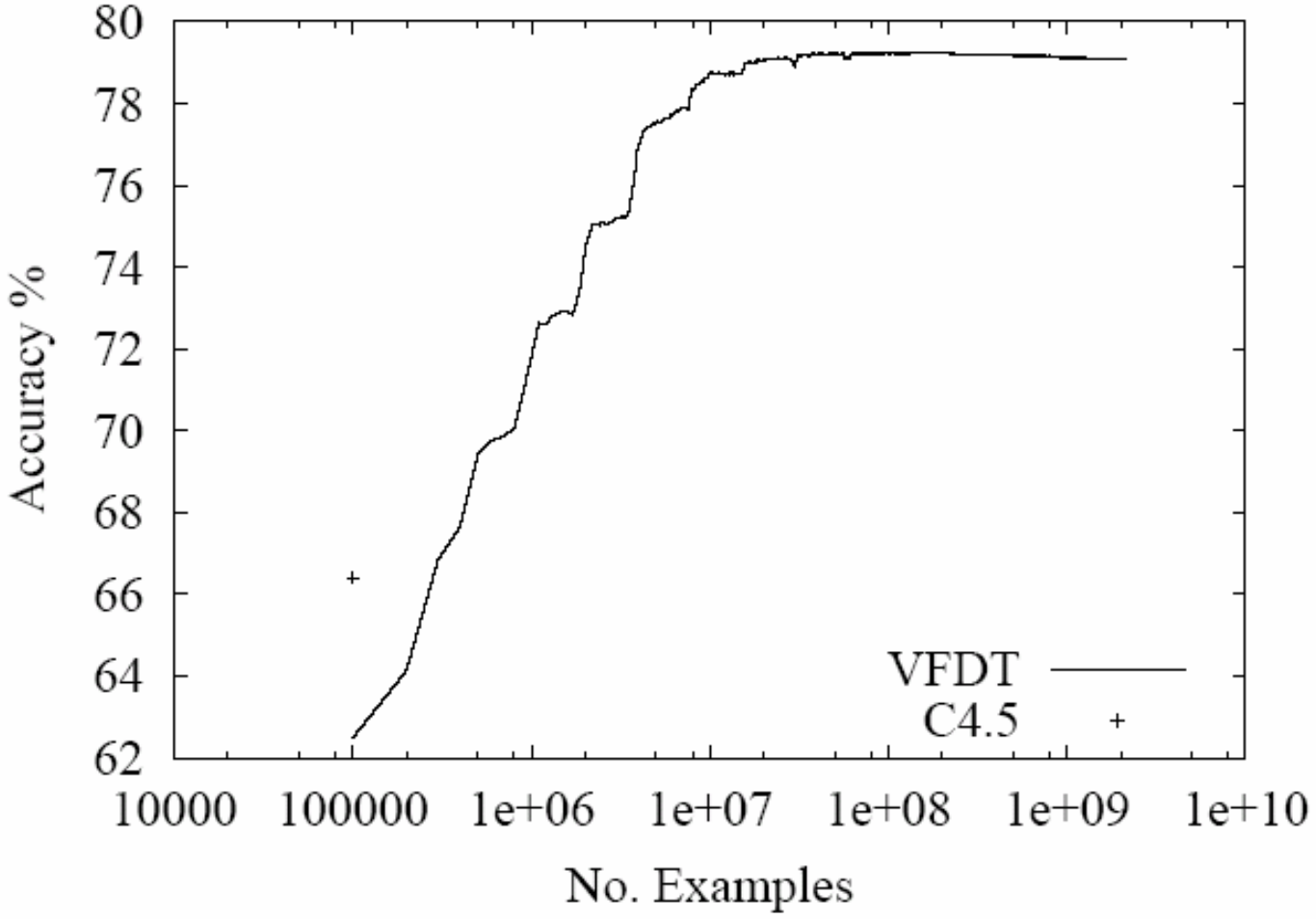
Very Fast Decision Trees: Main Algorithm

- **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

VFDT (Very Fast Decision Tree)

- Modifications to Hoeffding Tree
  - Near-ties broken more aggressively
  - G computed every n_{min}
  - Deactivates certain leaves to save memory
  - Poor attributes dropped
  - Initialize with traditional learner (helps learning curve)
- Compare to Hoeffding Tree: Better time and memory
- Compare to traditional decision tree
  - Similar accuracy
  - Better runtime with 1.61 million examples
    - 21 minutes for VFDT
    - 24 hours for C4.5
- Still does not handle concept drift

VFDT Trained on 2.5 Billion Examples



Classification in Changing Environments

- **Concept drift**
  - As time goes by, different elements belong to the mental categories (concept is not stationary, depends on time)
  - “interesting literature“ -- from novice to expert
  - “reasonably priced“ -- from student to manager
  - „spam email” – new versions arrive
- 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

Change Detection in Predictive Learning

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

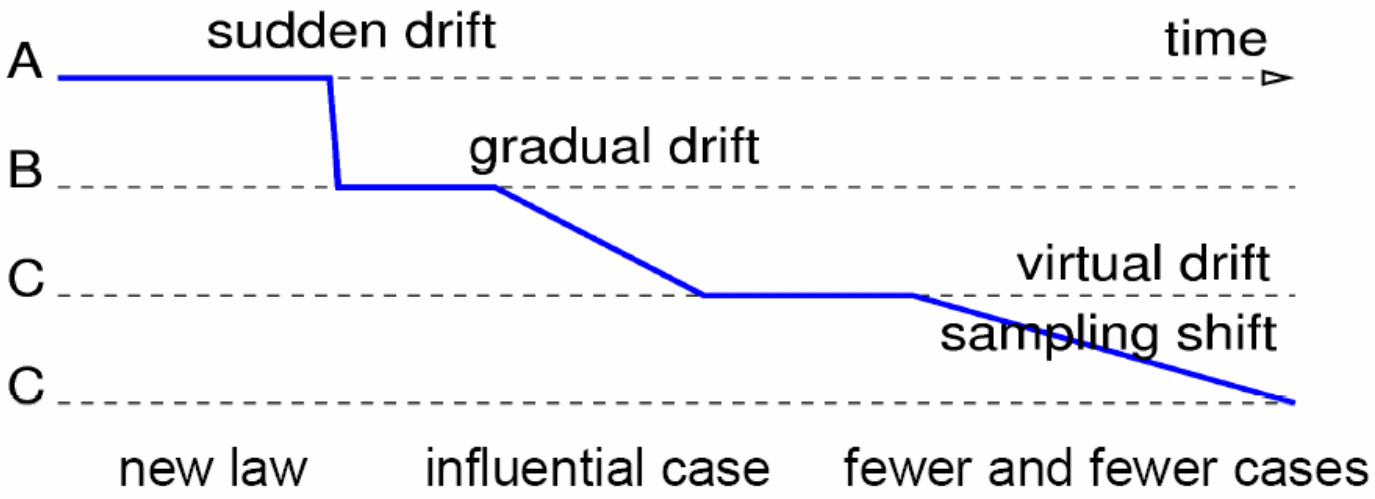
- The actual model does not correspond any more to the actual distribution.
- The error-rate increases

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

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

Types of Change

- "good faith" - not a *defined* concept in legislation (Rissland & Friedman, 1995)



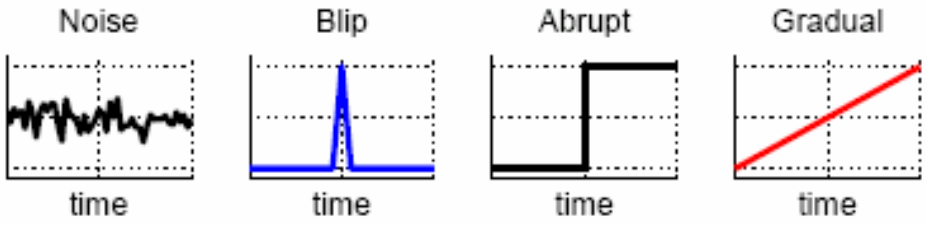
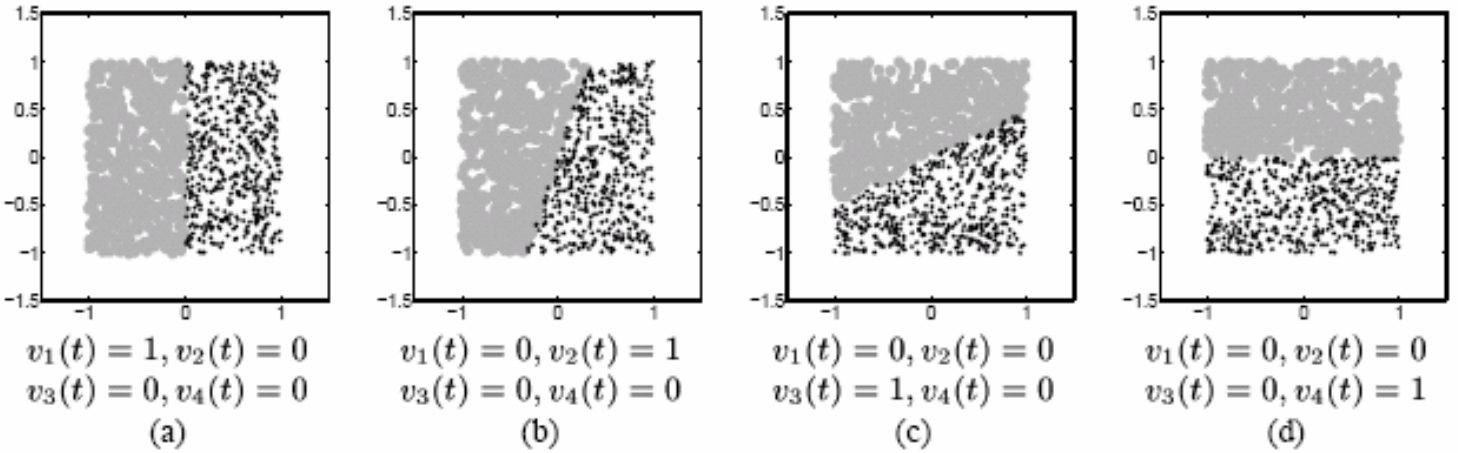


Figure 1. Types of concept change in streaming data

Gradual concept drift

- Rotating decision boundary problem



Handling Concept Drift – The Goal

- Quickly detect and adapt to concept drift
  - reduce time to learn the model (incrementally)
  - reduce time to adopt to change
- Robustness against noise
  - distinguish noise from slowly changing context
- Recognize and treat recurring contexts
  - quickly recover old models if appropriate
- Provide insights into changes
  - interpretability, local vs. global change

Illustration of sudden concept drift

- Experiments with Enron data sets

FIGURE 5.2: The time-spatial diagram of classes' distribution of the mann-k-mod dataset, according to the message timestamp order increasing from left to right. The five manually created partitions are marked.

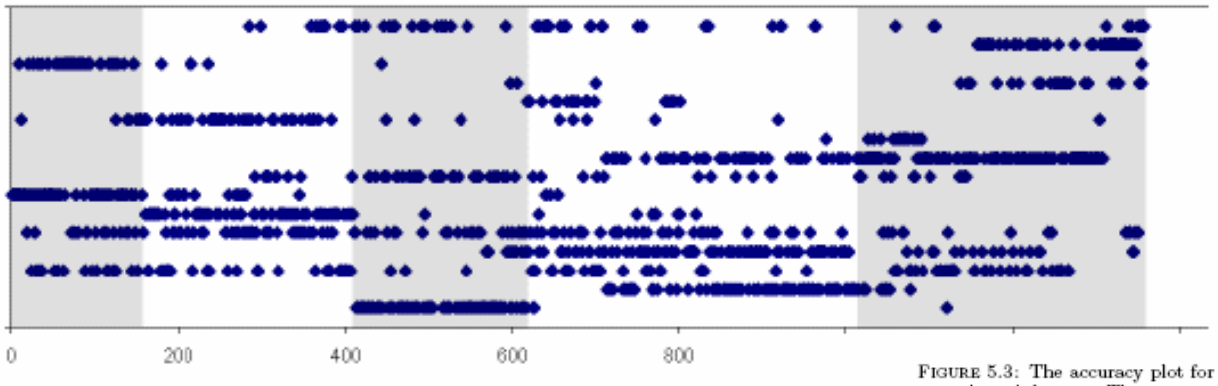
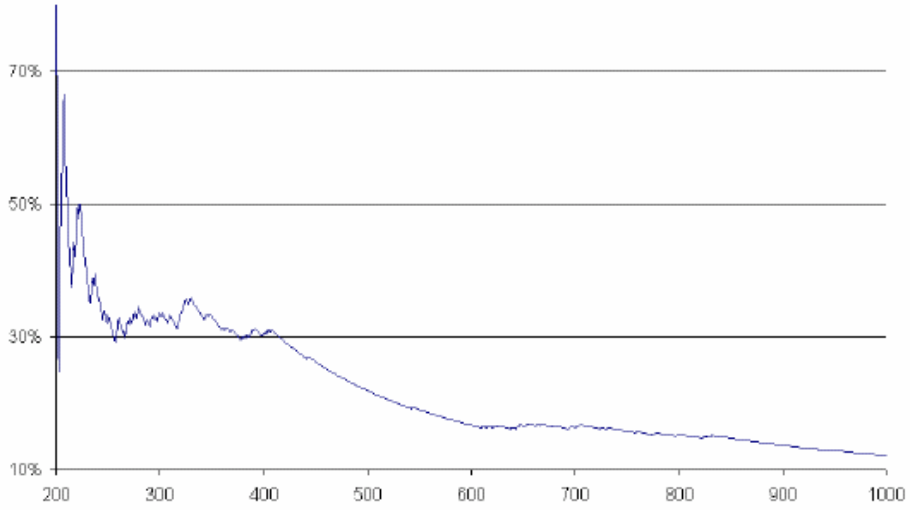
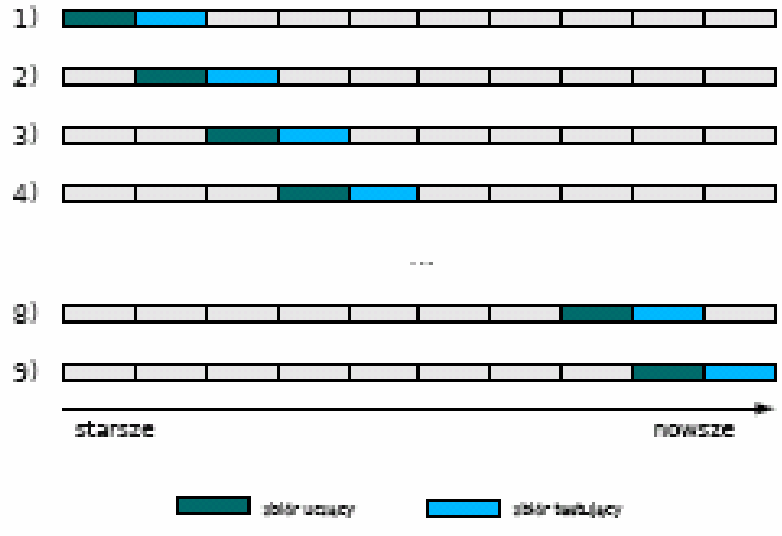


FIGURE 5.3: The accuracy plot for the hold out evaluation scheme applied on the mann-k-mod dataset. The training set contains the $k = 200$ first examples. Each accuracy value corresponds to the particular timestamp number of the test example.



Enron data – different window techniques

- Sliding windows (constant length)



Rysunek 5.2: Przesuwny zbiór uczący

TABLICA 6.6: Porównanie technik podziału na zbiory uczące i testujące na podstawie wyników testów Wilcoxona.

użytkownik	Porównanie technik podziału
beck-s	Przesuwny $\sim$ Rosnący
farmer-d	Przesuwny $\sim$ Rosnący
kaminski-v	Przesuwny $\ll$ Rosnący
kean-s	Przesuwny $>$ Rosnący
kitchen-l	Przesuwny $\sim$ Rosnący
lokay-m	Przesuwny $<$ Rosnący
mann-k	Przesuwny $\sim$ Rosnący
rogers-b	Przesuwny $>$ Rosnący
symes-k	Przesuwny $<$ Rosnący
williams-w3	Przesuwny $\gg$ Rosnący

TABLICA 6.4: Tabela średnich trafności klasyfikacji wraz z odchyleniami standardowymi dla prostych klasyfikatorów.

Rosnący zbiór uczący				
użytkownik	NB	kNN	C4.5	SVM
beck-s	50.94 ± 11.61	50.08 ± 7.84	49.55 ± 11.94	49.94 ± 9.15
farmer-d	69.24 ± 9.50	68.49 ± 9.46	71.28 ± 8.04	71.22 ± 7.17
kaminski-v	48.77 ± 4.63	41.06 ± 7.50	43.58 ± 6.28	46.00 ± 6.93
kean-s	19.62 ± 3.08	26.05 ± 2.91	32.27 ± 4.30	32.25 ± 4.29
kitchen-l	33.34 ± 9.32	30.38 ± 8.24	32.20 ± 10.79	36.74 ± 8.85
lokay-m	65.74 ± 3.49	71.91 ± 6.13	77.19 ± 6.19	78.26 ± 5.68
mann-k	67.32 ± 9.04	58.68 ± 14.64	57.30 ± 12.11	67.03 ± 14.98
rogers-b	67.29 ± 11.76	67.48 ± 11.91	68.85 ± 16.81	70.34 ± 18.15
symes-k	73.30 ± 5.86	77.49 ± 6.23	78.67 ± 7.07	80.39 ± 7.18
williams-w3	87.60 ± 15.69	94.57 ± 7.24	97.40 ± 2.87	95.90 ± 6.20
Przesuwny zbiór uczący				
użytkownik	NB	kNN	C4.5	SVM
beck-s	51.33 ± 10.14	50.28 ± 7.13	49.55 ± 7.57	51.31 ± 9.19
farmer-d	68.38 ± 8.64	67.76 ± 8.45	66.28 ± 9.64	70.23 ± 9.36
kaminski-v	45.46 ± 3.74	37.79 ± 4.18	40.50 ± 5.74	41.54 ± 5.03
kean-s	15.78 ± 5.98	26.65 ± 2.21	33.07 ± 3.06	33.06 ± 3.05
kitchen-l	37.63 ± 10.53	37.23 ± 17.15	41.19 ± 16.57	43.47 ± 18.50
lokay-m	68.52 ± 6.02	70.61 ± 5.42	73.69 ± 7.42	74.74 ± 6.18
mann-k	66.00 ± 13.50	59.90 ± 15.96	57.94 ± 12.42	67.02 ± 17.48
rogers-b	78.57 ± 9.51	74.07 ± 14.41	70.97 ± 21.34	72.44 ± 20.16
symes-k	75.85 ± 8.38	75.30 ± 9.23	77.14 ± 10.74	78.11 ± 9.61
williams-w3	92.06 ± 13.59	96.23 ± 6.2	98.61 ± 2.59	97.18 ± 6.13

How to tune windows …

- Rationale:
  - Recent data more useful for current context, remove old data
- window size
  - „just right“ – sufficient data, up to date model
  - too small – insufficient samples, poor models
  - too large – slow adaptation to changing contexts
- constant window size is unrealistic
  - conservative size – slow adaptation
  - How much data is needed? depends on (unknown) complexity of concept

- Stick to window, but adopt its size
- Window Adjustment Heuristic (Widmer&Kubat '96)
  - Monitor system performance (accuracy)
    - Extremly stable concept: reduce window width
    - Sufficiently stable: leave size constant
    - Otherwise more information needed: increase width *
- Optimal width (Klinkenberg&Joachims, 2000)
  - Identify best window size among a number of possible size by leave-one-out CV

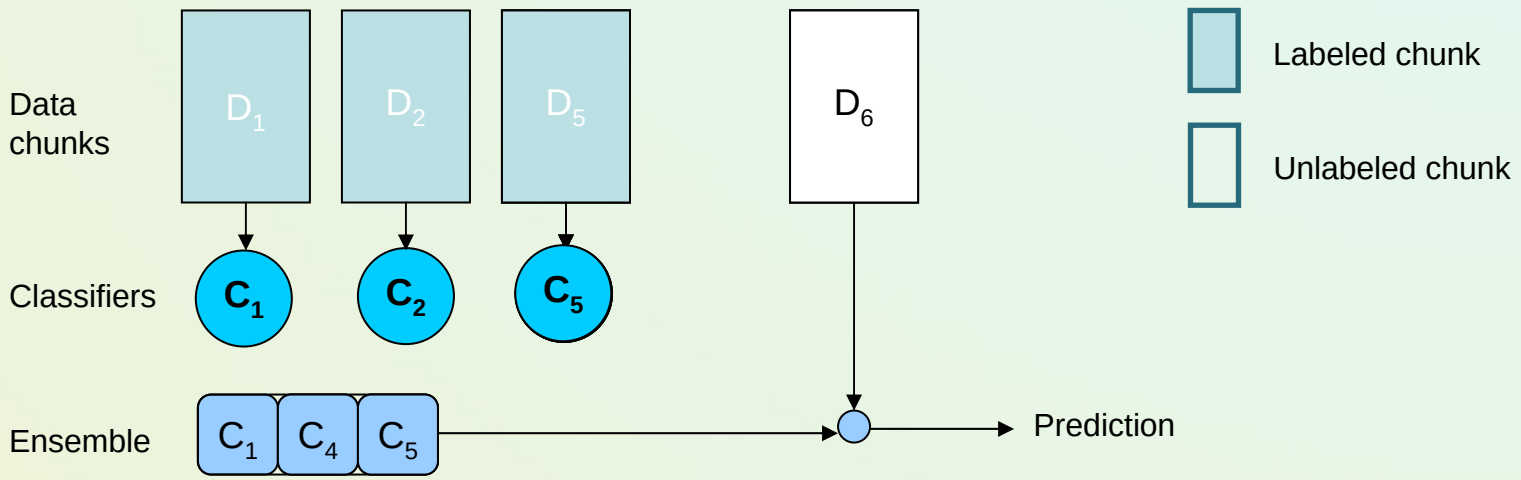
Landmark for classification from changing data

- *Instances versus batches of data.*
  - On-line learning approaches
- *Explicit versus implicit change detection.*
  - After detection $\rightarrow$ an action on the last data block
  - Forgetting heuristics (sliding windows)
- *Classifier-specific versus classifier-free.*
- *Classifier ensembles versus single classifiers.*
- *Completely supervised vs. semi-supervised (or demand) classification*
- *…*

- Concept Drift in Stream Mining
  - data arrives at a high rate and is processed on-line
- A stream of classified examples? And why do we need a classifier, then?
  - user feedback (multiple users, online/web system)
    - main interest in interpreting changes
  - delayed classification (fraud)
    - fast response to new types of fraud

Overview

- The Single-Partition Single-Chunk Ensemble (SPC) Approach
- Divide the data stream into equal sized chunks
  - Train a classifier from each data chunk
  - Keep the best *K* such classifier-ensemble
  - Example: for K= 3



Ensemble of Classifiers Algorithm

- H. Wang, W. Fan, P. S. Yu, and J. Han, “Mining Concept-Drifting Data Streams using Ensemble Classifiers”, KDD'03.
- Method (derived from the ensemble idea in classification)
  - train K classifiers from K chunks
  - for each subsequent chunk

    train a new classifier

    test other classifiers against the chunk

    assign weight to each classifier

    select top K classifiers

On-line algorithms

- Some single algorithms can be easily adopted
  - Neural networks
  - Instance based learning (IBL/CBL)
- New generalizations of ensembles
  - On-line baging and boosting (Oza)
  - Based on special sampling

Stream Data Mining: Research Issues

- Mining sequential patterns in data streams
- Mining partial periodicity in data streams
- Mining notable gradients in data streams
- Mining outliers and unusual patterns in data streams
- Stream clustering
  - Multi-dimensional clustering analysis?
    - Cluster not confined to 2-D metric space, how to incorporate other features, especially non-numerical properties
  - Stream clustering with other clustering approaches?
  - Constraint-based cluster analysis with data streams?

Summary: Stream Data Mining

- Stream data mining: A rich and on-going research field
  - Current research focus in database community:
    - DSMS system architecture, continuous query processing, supporting mechanisms
  - Stream data mining and stream OLAP analysis
    - Powerful tools for finding general and unusual patterns
    - Effectiveness, efficiency and scalability: lots of open problems
- Our philosophy on stream data analysis and mining
  - A multi-dimensional stream analysis framework
  - Time is a special dimension: Tilted time frame
  - What to compute and what to save?—Critical layers
  - partial materialization and precomputation
  - Mining dynamics of stream data

Some References

- S. Muthukrishnan Data Streams: Algorithms and Applications, Foundations & Trends in Theoretical Computer Science, 2005.
- C. Aggarwal, Ed. Data Streams: Models and Algorithms, Springer, 2007
- J. Gama, M. Gaber (Eds), Learning from Data Streams – Processing Techniques in Sensor Networks, Springer Verlag, 2007.
- Mining High Speed Data Streams, talk by P. Domingos, G. Hulten, SIGKDD 2000.
- State of the art in data streams mining, talk by M.Gaber and J.Gama, ECML 2007.
- J.Han slides for a lecture on Mining Data Streams
- L.Kuncheva: Classier Ensembles for Changing Environments, MCS 2004.
- A Survey of Stream Data Mining; E. Ikonomovska, S. Loskovska, D. Gjorgjevik, ETAI 2007.