Knowledge Discovery from Evolving Data

Myra Spiliopoulou [1] , Frank Höppner [2], Mirko Böttcher [1]

[1] Otto-von-Guericke University Magdeburg, Germany

[2] University of Applied Sciences, Braunschweig / Wolfenbüttel

The Presenters

Myra Spiliopoulou

Work group KMD – Knowledge Management & Discovery

Faculty of Computer Science, Otto-von-Guericke-Univ. Magdeburg

http://omen.cs.uni-magdeburg.de/itikmd

Frank Höppner

University of Applied Sciences Braunschweig / Wolfenbüttel

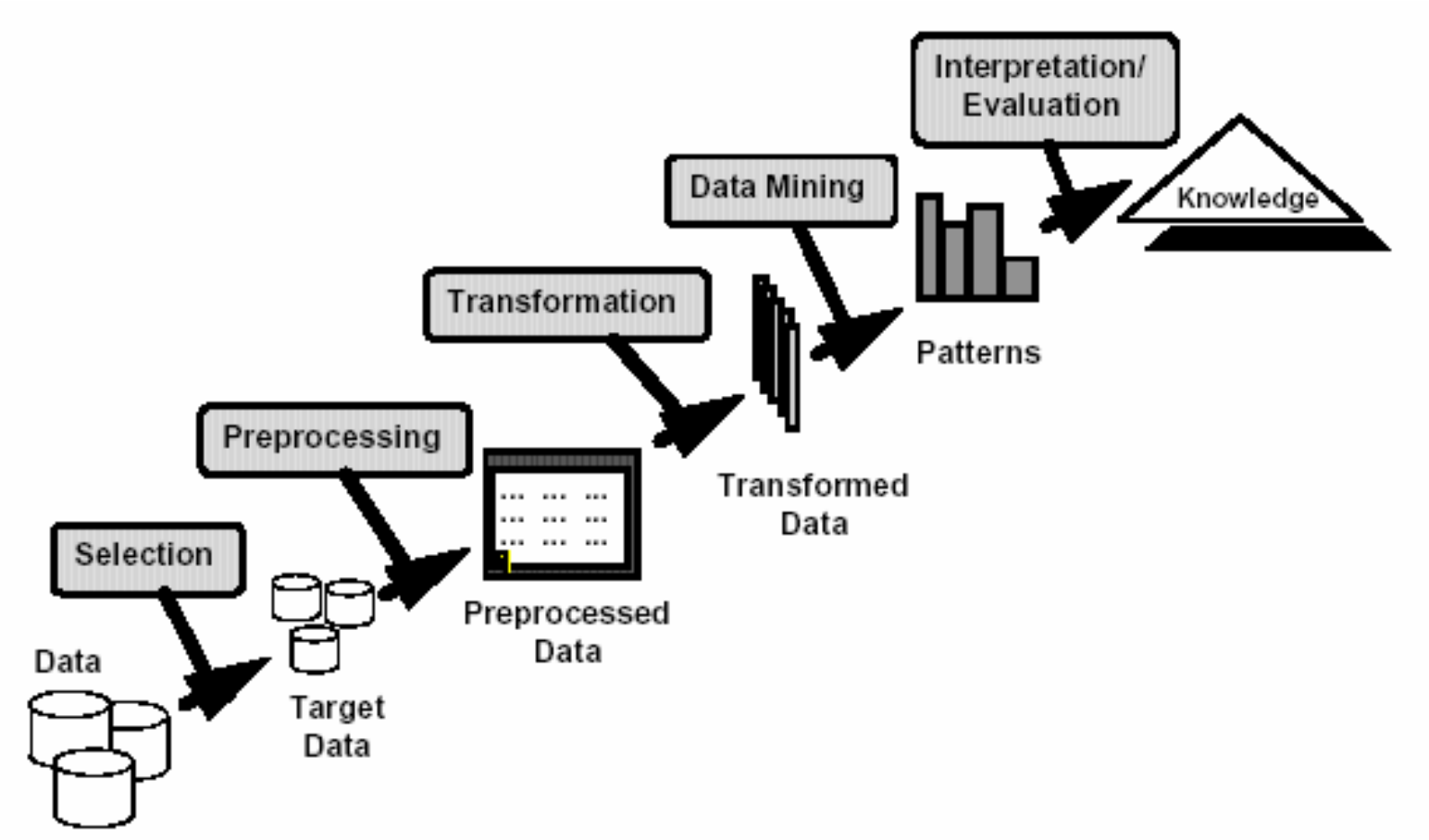
Mirko Böttcher

Work group CI – Computational Intelligence

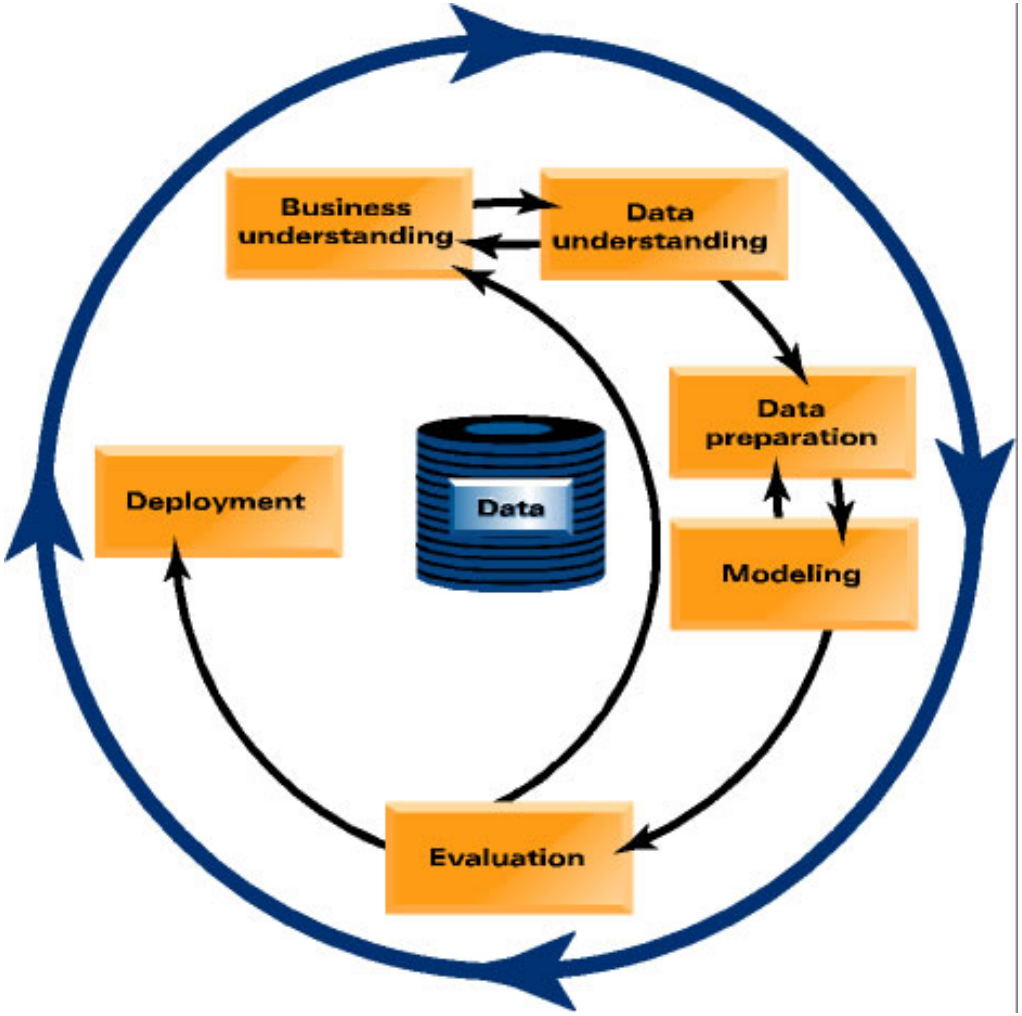
Faculty of Computer Science, Otto-von-Guericke-Univ. Magdeburg

http://fuzzy.cs.uni-magdeburg.de

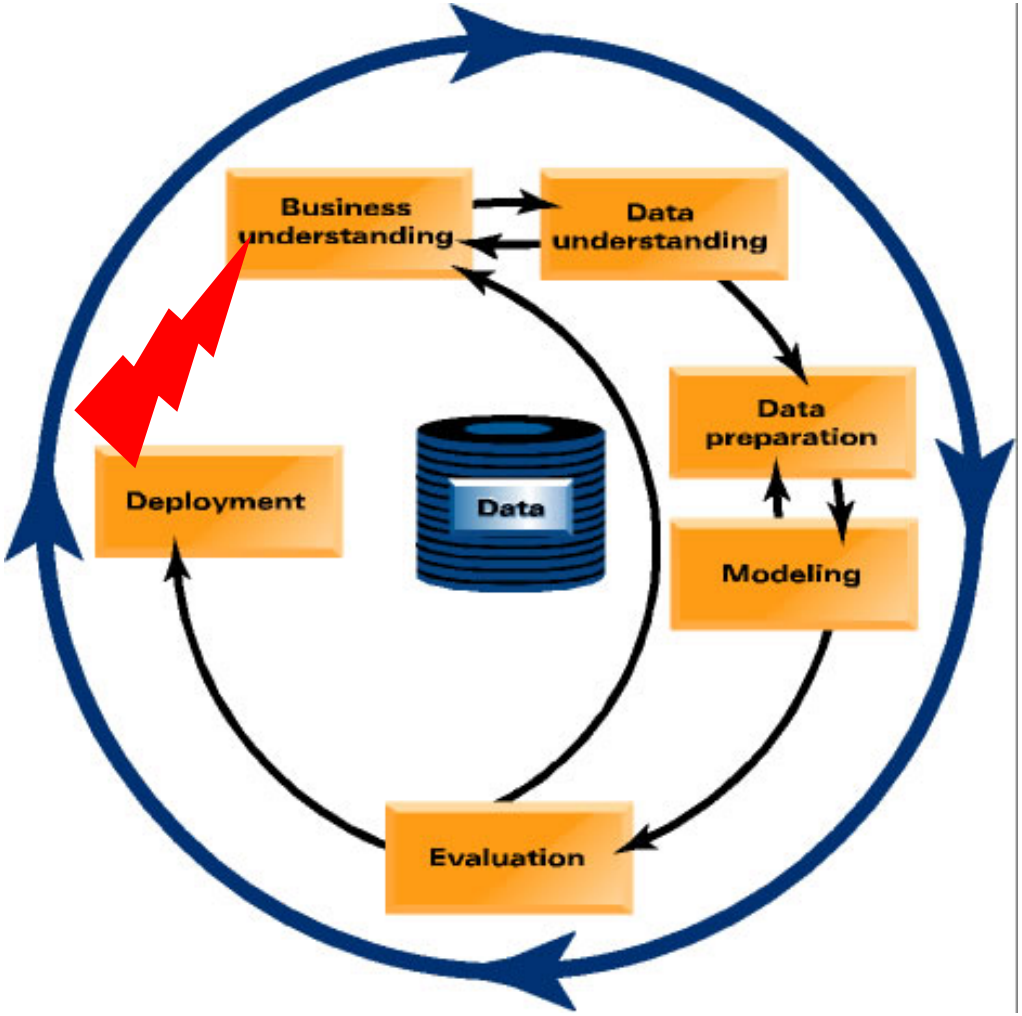
The classic view of Data Mining



CRISP-DM and the virtuous circle of data mining



CRISP-DM: Closing the open circle (sic)



- Deployment of discovered patterns implies influencing the population under observation.
  - Population evolution

A paradigm shift towards evolution

Data stream paradigm

- Evolution of the data means:
  - The models become obsolete – we must adapt them.
  - The models become obsolete – we must understand why.
- Keep in mind:
  - Evolution of the data is caused by factors that may or may not be in our control.
  - Even if we cause the evolution of the data, this does not mean that we know how they evolve.

Paradigms for knowledge discovery from evolving data

- Data stream model of computation (Guha et al)
- Higher Order Mining paradigm (Roddick et al, 2008)
- Change mining paradigm


The data stream model of computation (Guha et al)

- A data stream is a sequence of points $x_1,...,x_n$ that arrive in increasing order of the index i.
  - Algoritms operating upon a stream are subject to memory constraints and
  - must minimize the number of passes over the data.

- Clustering in the data stream model:

  Given a sequence of points
  - maintain a good clustering of the encountered sequence
  - while satisfying constraints on space and computation time.


The data stream model of computation (Guha et al)

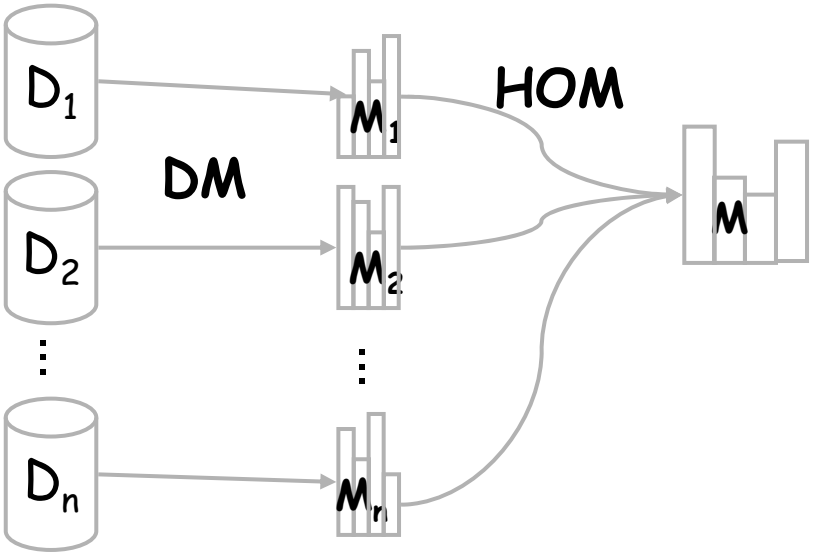
- A data stream is a sequence of points $x_1,...,x_n$ that arrive in increasing order of the index i.
  - Algoritms operating upon a stream are subject to memory constraints and
  - must minimize the number of passes over the data.

- Mining in the data stream model:

  Given a sequence of points
  - maintain a good model of the encountered sequence
  - while satisfying constraints on space and computation time.

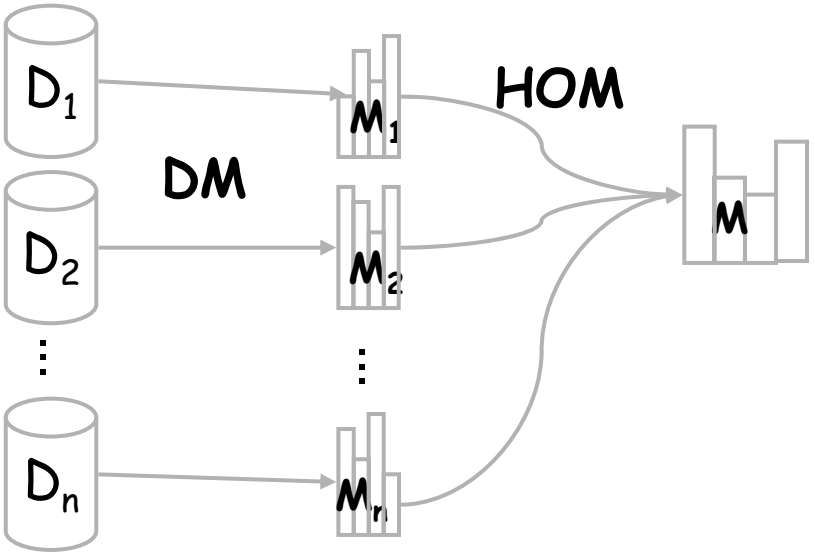

Higher Order Mining Paradigm (Roddick et al)

- Higher Order Mining (HOM)
  is the sub-field of knowledge discovery concerned with mining over patterns/models
  derived from one or more large / complex datasets.



Higher Order Mining Paradigm (Roddick et al)

- Higher Order Mining (HOM)
  is the sub-field of knowledge discovery concerned with mining over patterns/models
  derived from one or more large / complex datasets.



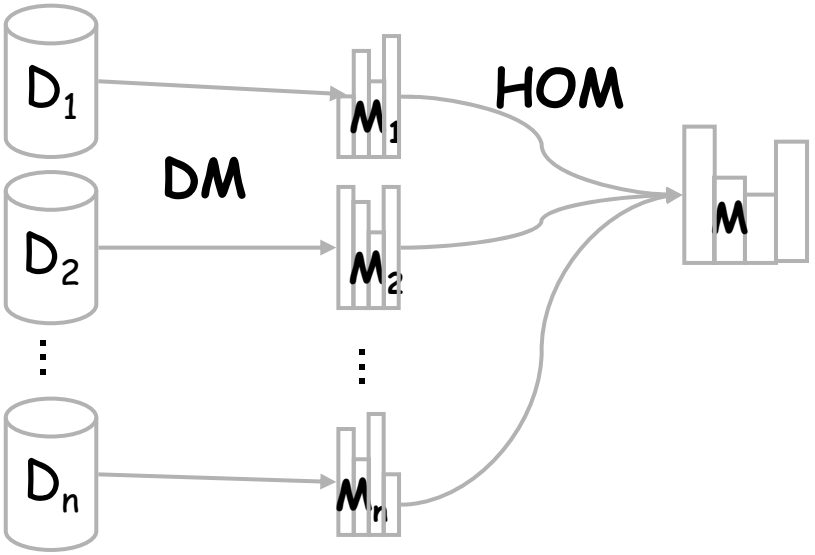
Meta-learning in distributed mining systems (Prodromidis et al):

$M_1,\ldots,M_n$ are classifiers build upon independent datasets.

M is a classifier trained upon the outputs delivered by the $M_1,\ldots,M_n$ for a given training dataset.

Higher Order Mining Paradigm (Roddick et al)

- Higher Order Mining (HOM)
  is the sub-field of knowledge discovery concerned with mining over patterns/models
  derived from one or more large / complex datasets.



Discovering evolutionary patterns from text (Mei & Zhai):

M_i is the clustering built at timepoint i=1...n.

A graph of clusters is built, where an edge connects two clusters X in M_i, Y in M_j, (i < j) if X and Y are similar.

Paradigms for knowledge discovery from evolving data

- Data stream model of computation (Guha et al)
- Higher Order Mining paradigm (Roddick et al, 2008)
- Change mining paradigm
  - Designed for data observed at different snapshots
  - Emphasis on modeling and understanding change – not on pattern adjustment
  - Overlaps with HOM as it uses mining methods on patterns


References (Block 1)

- S. Guha, A. Meyerson, N. Mishra, R. Motwani and L. O'Callaghan (2003). Clustering data streams: Theory and practice. *IEEE Trans. on Knowlede and Data Eng.,* 15(3):515-528.
- J. Roddick, M. Spiliopoulou, D. Lister and A. Ceglar (2008). Higher Order Mining. ACM SIGKDD Explorations, 10(1):5-17.
- Q. Mei and C. Zhai (2005). Discovering evolutionary theme patterns from text – an exploration of temporal text mining. In Proc. of *11th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'05),* 198-207, Chicago, IL.
- A. Prodromidis, P. Chan and S. Stolfo (2000). Meta-learning in distributed data mining systems: Issues and approaches. In *Advances in Distributed and Parallel Knowledge Discovery (H. Kargupta and P. Chan, eds)*, AAAI Press.
- P.-N. Tan, Steinbach and V. Kumar (2004). *Introduction to Data Mining*. Wiley.

Presentation Outline

- ☑ Block 1: Introduction
- Block 2: Evolution in Association Rules
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
  - Incremental Learning of Rules
  - Fundamentals of Change Detection
  - Querying Change
  - Change Pattern Detection
  - Redundancy Detection
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
  - Incremental Learning of Rules
  - Fundamentals of Change Detection
  - Querying Change
  - Change Pattern Detection
  - Redundancy Detection
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


Data Streams

- A data stream is an ordered sequence of items that arrive in timely order
- Data streams are continuous, unbounded and have a data distribution that often changes with time.
- Can be further classified into online and offline streams
  - Offline streams: regular bulk arrivals of data
  - Online streams: real-time updated data that come one by one in time


Challenges for Association Rule Mining

- Not enough time to (multiply) rescan the whole database
- Not enough space to store all the stream data
- Need to adapt to changing data distribution
- Processing should be as fast as possible
- The analysis results of data streams often change as well

➔ Association mining on data streams as an incremental process, i.e. new iterations of mining results are built based on old mining results.


Incremental Association Mining

Three Categories of Approaches (cf. (Jiang et al, 2006)):

- **Landmark Model**: Mines all frequent itemsets over the entire history history of stream data from a specific (landmark) point in time
- **Damped Model:** Mines associations in stream data in which each transactions has a weight which decreases with age
- **Sliding Windows Model**: Finds and maintains associations within a sliding window. Allows the expert to get the most recent analysis results.


Challenges of the Sliding Window Model

- Frequency counts of non-frequent itemsets need to be monitored as well, they may become frequent in the future
- Additional effort for storing non-frequent items in contrast to the need for a condensed and fast representation
- The set of stored itemsets must be dynamically maintained in a very fast data structure. Traditional data structures like prefix trees are inadequat.

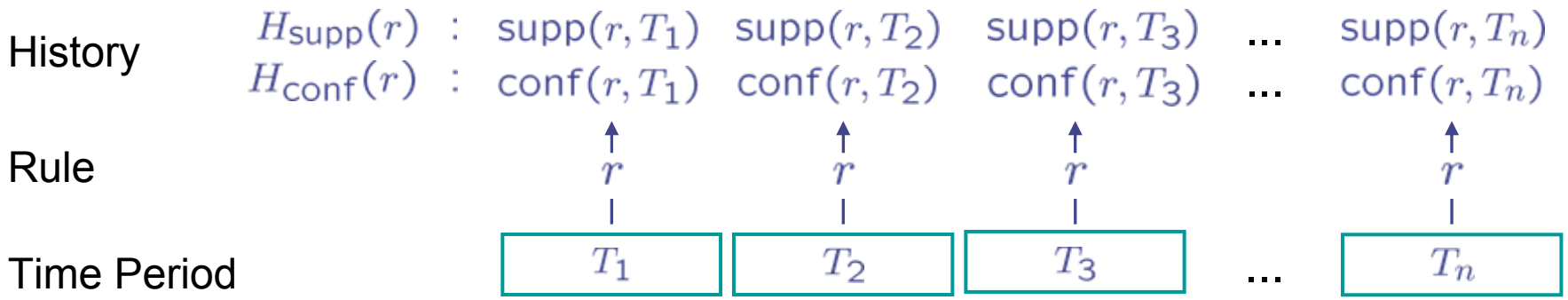

The Moment Algorithm (Chi et al, 2006)

- Mines *closed* frequent itemsets over a data stream sliding window
- *Closed Enumeration Tree* monitors closed frequent itemsets as well as itemsets that form the boundary between the closed frequent and the remaining itemsets
- Proof that every itemset that changes status (e.g. from non-frequent to frequent) must come to this boundary
- The Moment algorithm maintains the tree in an efficient way


Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
  - ☑ Incremental Learning of Rules
  - ❑ Fundamentals of Change Detection
  - ❑ Querying Change
  - ❑ Change Pattern Detection
  - ❑ Redundancy Detection
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook

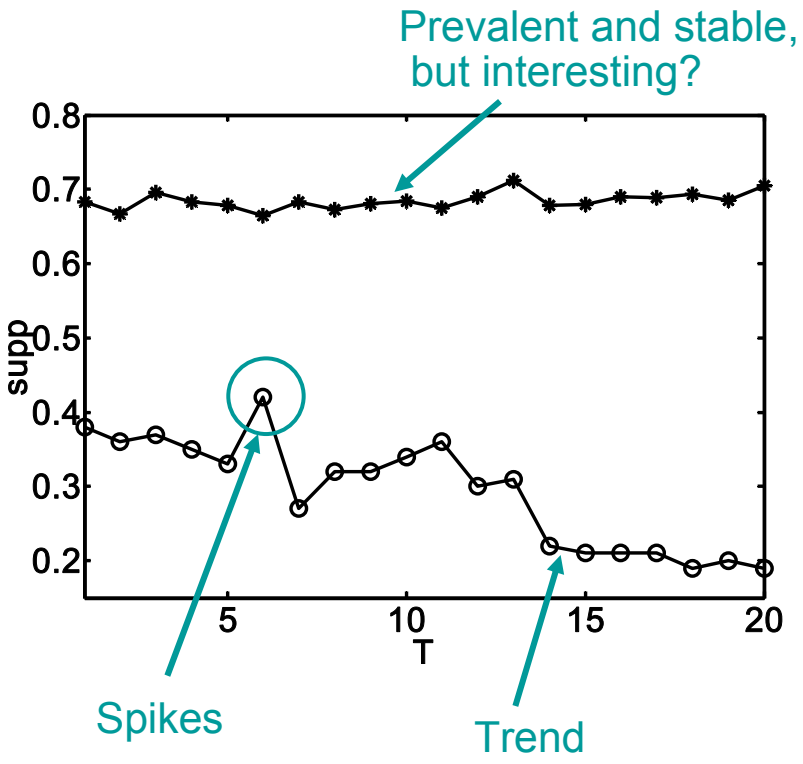

From Association Rules to Rule Histories



Rule Change: Change of a Rule's Statistical Properties,
not the Change of its Symbolic Representation!

Goal: Discover interesting events or patterns within rule histories

Why are Rule Histories Important?



- Prevalent patterns are rarely of interest
- A user is interested in information which is novel or changes
- Change has an intrinsic interestingness
- Not the actual value of a traditional interestingness measure is of interest but how it changes

How to Choose the Length of the Time Periods?

- Small Periods
  - Lead to fine-grained histories
  - Allow for detecting also small change patterns
  - Can lead to noisy histories
- Long Periods
  - Lead to coarse-grained histories, small patterns may be missed
  - Allows for a more reliable calculation of support and confidence due to larger sample sizes


How to Choose the Length of the Time Periods?

- Ideally,
  - the choice of the time period should be done for each rule individually
  - time periods should have different length
- Practically,
  - Time periods are chosen based on the underlying problem or the sampling frequency of data
  - For example, data is often collected on a daily, or weekly basis


Temporal Description Length (Chakrabarti, 1998)

- Method to determine the best time segmentation for each itemset such that the correlation between its constituting items is …
  - maximally homogeneous within, but …
  - inhomogeneous between segments
- Based on the *minimum description length principle*
- Defines a binary coding scheme for time segments which assigns a code length that declines with increasing homogeneity
- The segmentation is then chosen such that the sum of all segment code lengths is minimised
- The overall code length can be used to assess the interestingness of itemsets

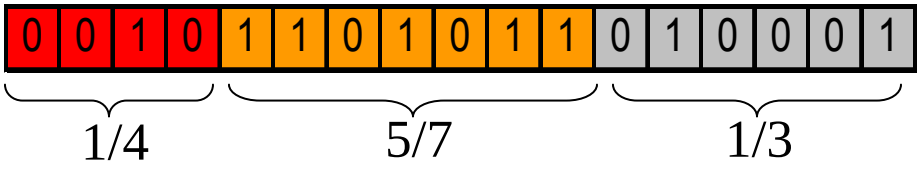

Temporal Description Length (Chakrabarti, 1998)

Underlying Idea: Assume an itemset being represented by a coin

- Players A and B
- A has a set of coins with different biases
- A repeatedly
  - Picks arbitrary coin
  - Tosses it arbitrary number of times
- B observes the number of heads and tails
- B guesses transition points between coins and their biases


Temporal Description Length (Chakrabarti, 1998)

- Given *n* observations of heads and tails
  - Can assume *n* different coins with bias 0 or 1
    - Data fits perfectly (with probability one)
    - Many coins needed (“expensive”)
  - Or assume one coin
    - Likely to fit observations poorly (but “inexpensive”)
- “Best explanation” is a trade-off between fit and expensiveness



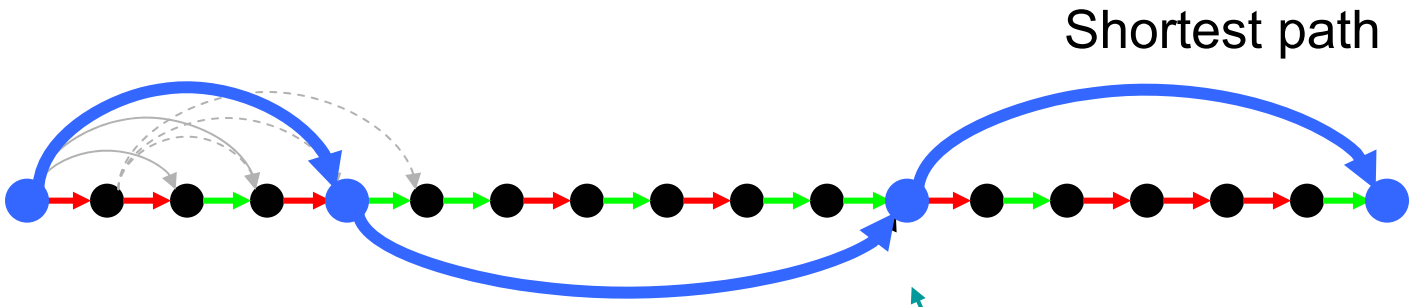
(http://www.cse.iitb.ac.in/~soumen/doc/vldb1998/)

Temporal Description Length (Chakrabarti, 1998)

Sequence of 17 tosses:

0	0	1	0	1	1	0	1	0	1	1	0	1	0	0	0	1

Derived graph with 18 nodes:



Edge cost = model cost + data cost

Data cost for Pr[head] = 5/7, 5 heads, 2 tails

Model cost = one node ID + one Pr[head]

(http://www.cse.iitb.ac.in/~soumen/doc/vldb1998/)

Temporal Description Length (Chakrabarti, 1998)

- Approach is very expensive and does only work for the support of itemsets
- Cannot easily be adopted to other rule measures
- Every itemset has a different time segmentation which makes comparison between histories impossible


Steps for Change Detection
Proposed by (Boettcher et al, 2006)

Association rules have to be discovered, stored and managed.

Change patterns have to be reliably detected

Histories with change patterns have to be analysed for redundancies and evaluated.

Structural Analyser

- Association Rule Mining System
  - incremental algorithm
- Rule History Database

Change Analyser

- Noise Filter
  - Double exponential smoothing
- Trend Detection
  - Mann-Kendall Test
  - Cox-Stuart Test
- Stability Detection
  - χ^2 Test

Interestingness Evaluator

- Temporal Redundancy Filter
- Interestingness Measures

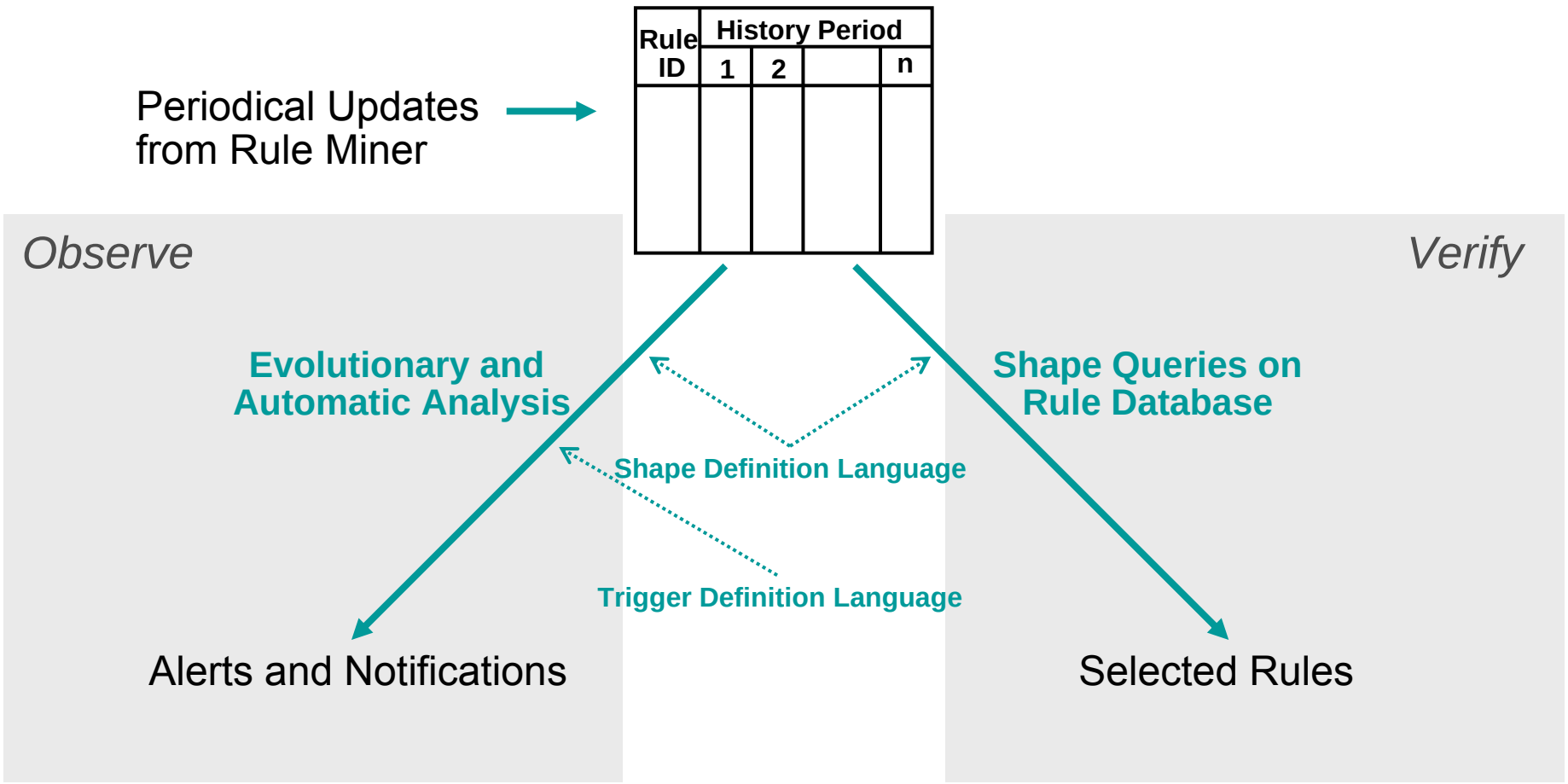

Evolution in Association Rules

- ☑ Incremental Learning of Rules
- ☑ Fundamentals of Change Detection
- ■ Querying Change
- ■ Change Pattern Detection
- ■ Redundancy Detection


Querying Change (Agrawal, 1995)

- Additionally to the vast number of rules the same number of histories needs to be analyzed
- The histories are periodically updated thus what can be detected changes constantly
- Having knowledge about the underlying domain an expert
  - may be able to specify types of change which are more likely to be critical or interesting
  - may have assumptions about which change should be present and needs to verify them


Querying Change (Agrawal, 1995)



Querying Change (Agrawal, 1995)

Shape and Query Definition Language (IBM Quest Project, 1995)

Definition of an upward trend

Based on the number of increasing values within a time window

```
(shape uptrend(width upcnt)
   (in width
       (noless upcnt (any up Up)))
```

Check for trigger condition at each history update

```
(trigger detect_up
    (events updatehistory)
    (condition
       (uptrend(5 4)
         (support (- end 5) end)))
     (actions upward)
)
```

Action to be carried out when trigger condition holds


Evolution in Association Rules

- ☑ Incremental Learning of Rules
- ☑ Fundamentals of Change Detection
- ☑ Querying Change
- ■ Change Pattern Detection
- ■ Redundancy Detection


Time Series Analysis

- The objectives of time series analysis are (Chatfield, 1996):
  - To describe the data
  - To model the data generating process
  - To do forecasting
  - To use the forecasts to control a given process
- Requirements (for the last three objectives):
  - Availability of a valid model
  - Sufficiently large number of values for parameter estimation and testing
  - Human expert for model validation


Properties of Histories

- Vast number of histories, for each rule two or more
  - Very difficult to make any assumptions about the underlying model for each history
  - Impossible to examine by a human expert
- Histories are deeply intertwingelt, e.g. the history of a rule is influenced by the history of its more general rules
- A user has rarely a precise idea about how rules should evolve
- A user needs prediction, if at all, only for a tiny fraction of rules


Best Practices for Change Pattern Detection

- Prefer non-parametric statistical methods over parametric ones
- Use methods which do require as little human intervention (parameter tuning, validation) as possible
- First descriptive analysis to get a smaller set of histories on which predictive methods can be applied
- Trade-off between time complexity, accuracy and sophistication of methods

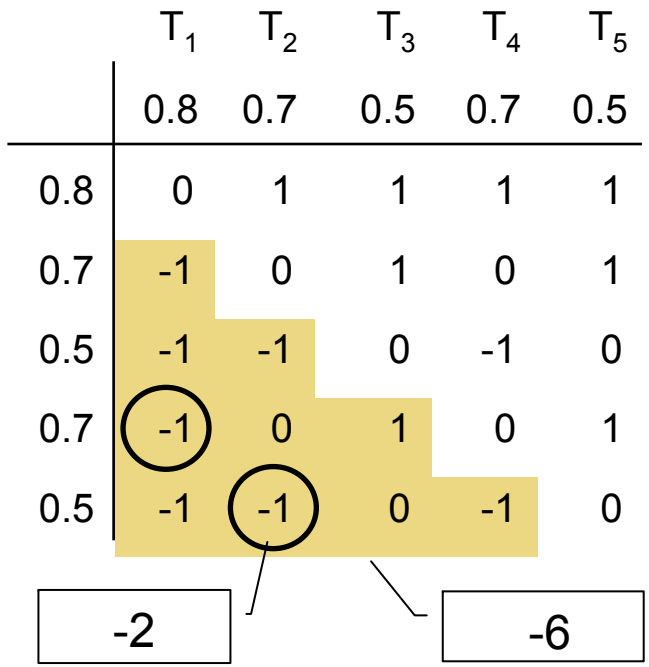

Change Pattern Detection – Trends
Approach used by (Boettcher et al, 2006)

Mann-Kendall Test

- Complexity $O(n^2)$
- noise robustness

Cox-Stuart Test

- Complexity O(n)
- susceptible to noise



For short histories and not time-critical applications the Mann-Kendall Test is more suitable.

Change Pattern Detection – Trends
Approach used by (Liu et al, 2001)

- Support and confidence measure proportions between subsets
- Stable History : Proportions must be equal in all time periods

➔ χ^2 Test : $H_0 \quad : \quad \text{supp}(r, T_1) = \text{supp}(r, T_2) = \ldots = \text{supp}(r, T_n)$
$H_1 \quad : \quad$ the supports are not all equal

Problems of the Approach:

- the hypothesis to reject is H_1
  ➔ „optimistic detection“ of stability
- does not account for the temporal order
  ➔ clear trends may be recognized as stable

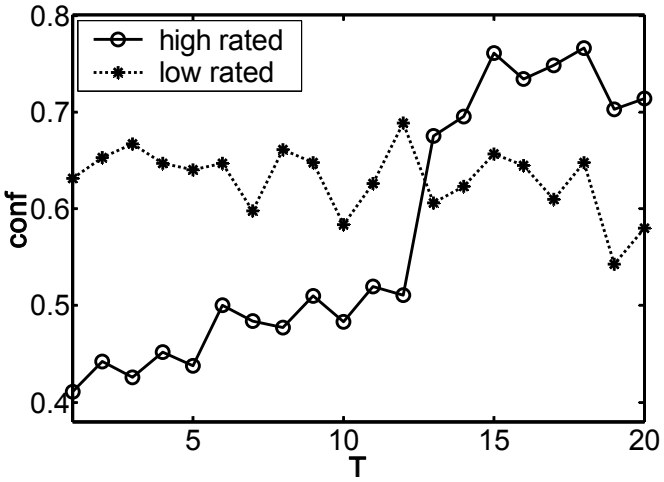
Improvement:

1. Test for any other change pattern, for instance a trend
2. If tests are negative, test for stability using the χ^2 test

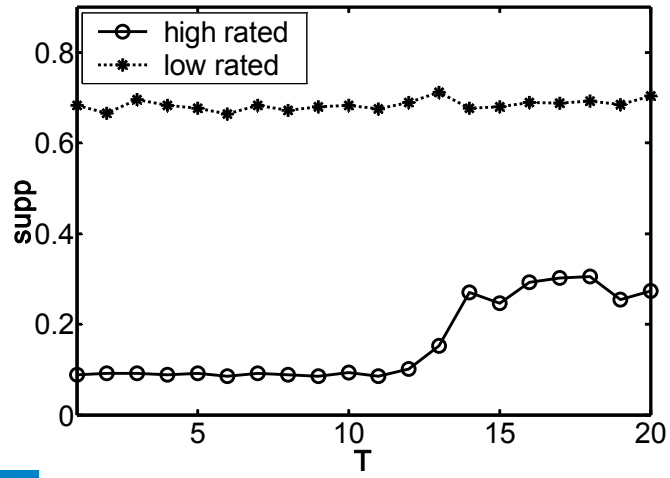
Interestingness Assessment
Approach used by (Boettcher et al, 2006)

- Each history is not like any other
- Each history has its own interestingness
- Interestingness as the discrepancy between a history and general assumptions about change

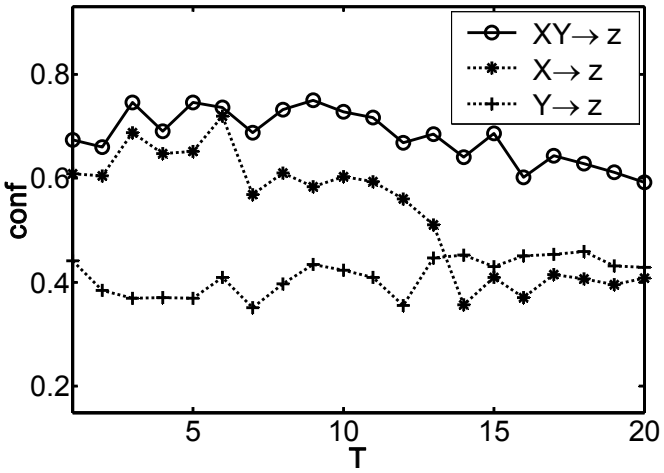
Stability



Non-rapid Change



Homogeneity



Frameworks for change detection:
The Pattern Monitor of Baron et al (ADBIS03;EWMF03;05)

- The Pattern Monitor PAM observes patterns as
  - temporal objects maintained in a DBMS
    - association rules and frequent sequences
    - web usage patterns
    - clusterings
  - associated with statistical properties
    - support, confidence
    - cardinality, homogeneity
  - taking the form of a time series – sequence of timestamps
  - at which the data population has been processed by a mining query.

$$R = (ID, query, body, head, \{timestamp, stats\})$$

Frameworks for change detection: The Pattern Monitor PAM (2)

- PAM has two modi of operation:

1. Observation of existing patterns at each time interval

   The records in the *data slice* of this interval are mapped to the patterns and the pattern statistics for the time slice are recorded:
   - Records supporting existing association rules are found, filling the corresponding support and confidence values.
   - Records are put into the clusters, filling the cluster stats.

2. Pattern re-learning at each time interval

   Data mining is performed upon the data slice.
   - The patterns discovered are compared to those of the previous slice.
   - New patterns are stored to the database as temporal objects.
   - The statistics of old patterns are recorded for the time slice.


Frameworks for change detection: The Pattern Monitor PAM (3)

- For both modi of operation:
  - The time series of each statistical property is monitored.
  - Three interestingness heuristics detect changes and raise alerts.
  - Patterns are characterized in terms of *stability* & *slope of change* :
    - Permanent patterns appear in all/most of the timestamps.
    - Temporary patterns indicate short-lived phenomena.

    whereby stable patterns are of more interest
    while unstable patterns might correspond to local noise.
  - When changes to overlapping patterns are identified, they are mapped to a smaller set of *atomic changes*.

Frameworks for change detection: The Pattern Monitor PAM (4)

Some application areas:

- Exception detection in mailings (Baron et al, ADBIS03)
  - Association rules discovery has been performed over a mailserver log.
  - A few very stable rules have been identified.
  - All of them were trivial.

    If the sender is outside the mailserver's domain, then at least one recipient belongs to the mailserver's domain

  - However, changes in the statistics of those rules indicated potential exceptions (e.g. mailserver misuse).

Frameworks for change detection: The Pattern Monitor PAM (4)

Some application areas:

- Exception detection in mailings (Baron et al, ADBIS03)
- Web usage mining (Baron et al, EWMF03)
  - The stability of association rules in a small web server log was studied
  - identifying a minimal set of changed rules.
- Topic trend monitoring
  - Topic clusters were discovered in a text corpus.
  - The clusters were monitored as new batches of documents arrived (PAM modus 1)

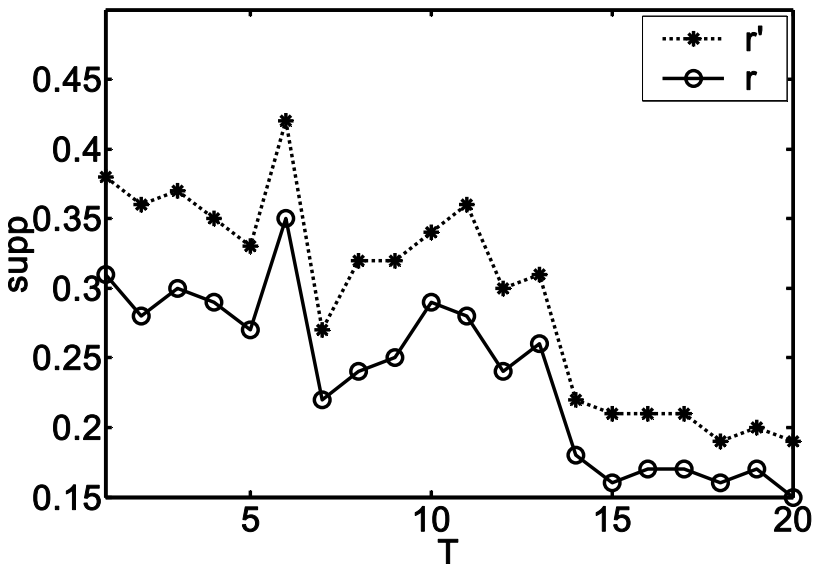

Evolution in Association Rules

- ☑ Incremental Learning of Rules
- ☑ Fundamentals of Change Detection
- ☑ Querying Change
- ☑ Change Pattern Detection
- ■ Redundancy Detection


Temporal Redundancy

r : Product=Broadband,Customer=Male $\rightarrow$ InternetUsage=Low

r' : Product=Broadband $\rightarrow$ InternetUsage=Low



Reason: The fraction of males among all broadband users with low Internet usage remains stable over time

➔ r' is a likely cause for the trend observed in $H_{\text{supp}}(r)$

➔ $H_{\text{supp}}(r)$ can be *derived* from $H_{\text{supp}}(r')$

➔ r is *temporally redundant* to r' with respect to support

Detecting Temporally Non-Redundant Rules
The Approach by (Liu, 2001)

- Detects the fundamental changes between *two* consecutive periods
- Method is basically the same for confidence and support
- Given a rule $r : \mathcal{X} \Rightarrow y$, it can be seen as a combination of two more general rules $r' : \mathcal{X}' \Rightarrow y$ and $r'' : \mathcal{X}'' \Rightarrow y$, whereby $\mathcal{X}' \cup \mathcal{X}'' = \mathcal{X}$ and $\mathcal{X}' \cap \mathcal{X}'' = \varnothing$.
- It is assumed that the proportional relationships between the itemsets of r, r' and r'' should stay the same
- A rule r is called non-fundamental (i.e. redundant) if the assumption is not violated


Detecting Temporally Non-Redundant Rules
The Approach by (Liu et al, 2001)

- The *expected supports* $E_{r'}$ and $E_{r''}$ of r in period T_2 are

$$E_{r'}(\text{supp}(r, T_2)) = \min\left(1, \frac{\text{supp}(r, T_1)}{\text{supp}(r', T_1)}\ \text{supp}(r', T_2)\right)$$

$$E_{r''}(\text{supp}(r, T_2)) = \min\left(1, \frac{\text{supp}_1(r, T_1)}{\text{supp}(r'', T_1)}\ \text{supp}(r'', T_2)\right)$$

- The rule r is defined as non-fundamental if rules r' and r'' exist, such that a $\chi 2$ test fails to reject the following null hypothesis:

$$H_0 : E_{r'}(\text{supp}(r, T_2)) = E_{r''}(\text{supp}(r, T_2)) = \text{supp}(r, T_2)$$
$$H_1 : \text{the three values are not all equal}$$

Temporally Derivative Rules
The Approach by (Böttcher et al, 2005)

Let m be a measure like support or confidence, and $m(T) := m(s, T)$ and $m_i(T) := m(s_i, T)$ its functions over time.

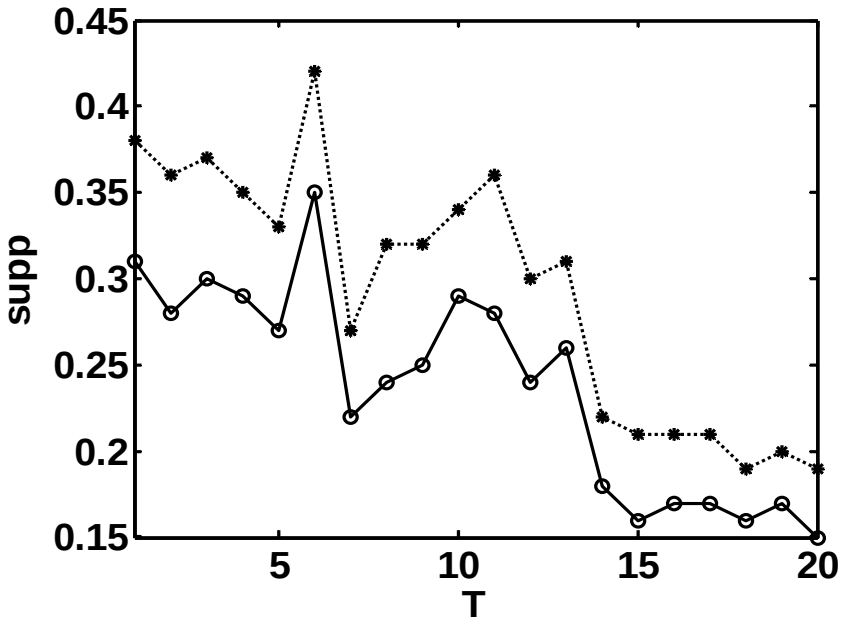
Let s, $s_1, s_2 \ldots s_p$ be rules or itemsets with $s \prec s_i,\ i = 1, \ldots, p$.

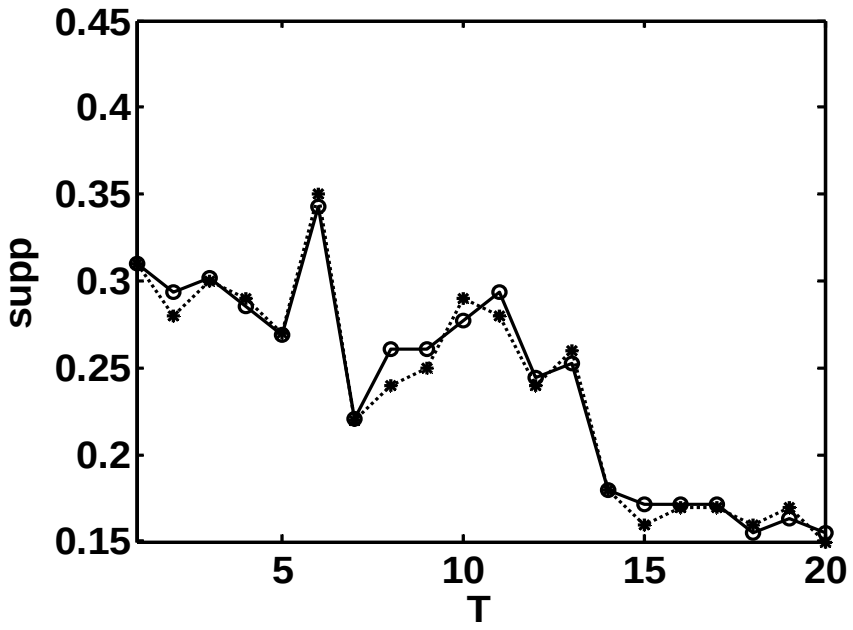
Let $\mathcal{M} := \{g : \mathbb{R} \longrightarrow \mathbb{R}\}$ be the set of real-valued functions over time.

The history $H_m(s)$ regarding the measure m is called *derivative* iff a function $f : \mathcal{M}^p \longrightarrow \mathcal{M}$ exists such that for all $T \in \hat{T}$

$$m(T) = f(m_1, m_2, \ldots, m_p)(T)$$

Temporally Derivative Rules – Example
The Approach by (Böttcher et al, 2005)





By multiplication with a constant factor one history can (apart from noise) be transformed into the other.

Temporally Derivative Rules – Criteria for Support
The Approach by (Böttcher et al, 2005)

An item set X ∪ Y is derivable w.r.t. to *support* if any of the following criteria hold:

Criterion 1:

$$\forall T \; : \; \frac{\text{supp}(\mathcal{XY},T)}{\text{supp}(\mathcal{Y},T)} = \text{const.}$$

- The fraction of transactions containing X additionally to Y grows in the same proportion as Y.
- conf(Y→ X) is constant over time

Criterion 2:

$$\forall T \; : \; \frac{\text{supp}(\mathcal{XY},T)}{\text{supp}(\mathcal{X},T)\,\text{supp}(\mathcal{Y},T)} = \text{const.}$$

- The degree of dependence between X and Y is constant over time.
- lift(X → Y) is constant over time.

Temporally Derivative Rules – Criteria for Confidence
The Approach by (Böttcher et al, 2005)

A rule XY $\rightarrow$ z is derivable w.r.t. to *confidence* if the following criterion holds:

Criterion 3:

$$\forall T \ : \ \frac{\text{conf}(\mathcal{XY}\rightarrow z, T)}{\text{conf}(\mathcal{X}\rightarrow z, T)} = \text{const.}$$

- The contribution of Y in addition to X to predict z relative to the predictive power of X remains stable over time.

References (Block 2)

- Agrawal, R. and Psaila, G. (1995). Active data mining. In Fayyad, Usama, M. and Uthurusamy, R., editors, *Proceedings of the 1st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 3–8, Montreal, Quebec, Canada. AAAI Press, Menlo Park, CA, USA.
- Böttcher, M., Nauck, D., Ruta, D., and Spott, M. (2006). Towards a framework for change detection in datasets. In *Proceedings of the 26th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 115–128. Springer
- Böttcher, M., Nauck, D., and Spott, M. (2005). Detecting Temporally Redundant Association Rules. In *Proceedings of the 4th International Conference on Machine Learning and Applications*, pages 397-403. IEEE Computer Science Press
- Chi, Y., Wang, H., Yu, P. S., and Muntz, R. R. 2006. Catch the moment: maintaining closed frequent itemsets over a data stream sliding window. *Knowl. Inf. Syst.* 10, 3 (Oct. 2006), 265-294
- Chakrabarti, S., Sarawagi, S., and Dom, B. (1998). Mining surprising patterns using temporal description length. In *Proceedings of the 24th International Conference on Very Large Databases*, pages 606–617. Morgan Kaufmann Publishers Inc.


References (Block 2)

- Chatfield, C. (2003). *The Analysis of Time Series: An Introduction, Sixth Edition (Texts in Statistical Science)*. Chapman & Hall/CRC
- Cox, D. and Stuart, A. (1955). Some quick sign tests for trend in location and dispersion. *Biometrika*, 42:80–95
- Jiang, N. and Gruenwald, L. (2006). Research issues in data stream association rule mining. *SIGMOD Rec.* 35, 1 (Mar. 2006), 14-19.
- Liu, B., Hsu, W., and Ma, Y. (2001). Discovering the set of fundamental rule changes. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 335–340.
- Liu, B., Ma, Y., and Lee, R. (2001). Analyzing the interestingness of association rules from the temporal dimension. In *Proceedings of the IEEE International Conference on Data Mining*, pages 377–384. IEEE Computer Society.
- Mann, H. (1945). Nonparametric tests against trend. *Econometrica*, 13:245–259.
- Baron, S., Spiliopoulou, M., and Günther, O. (2003). Efficient monitoring of patterns in data mining environments. In *Proceedings of the 7th East-European Conference on Advances in Databases and Information Systems (ADBIS'03)*, pages 253–265. Springer.


End of Block 2

Thank you!

Questions?

Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
- Block 3: Evolution in Classifiers
  - Concept Drift
  - Spectrum of Solutions
- Block 4: Evolution in Clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


Evolution in Classifiers – Concept Drift

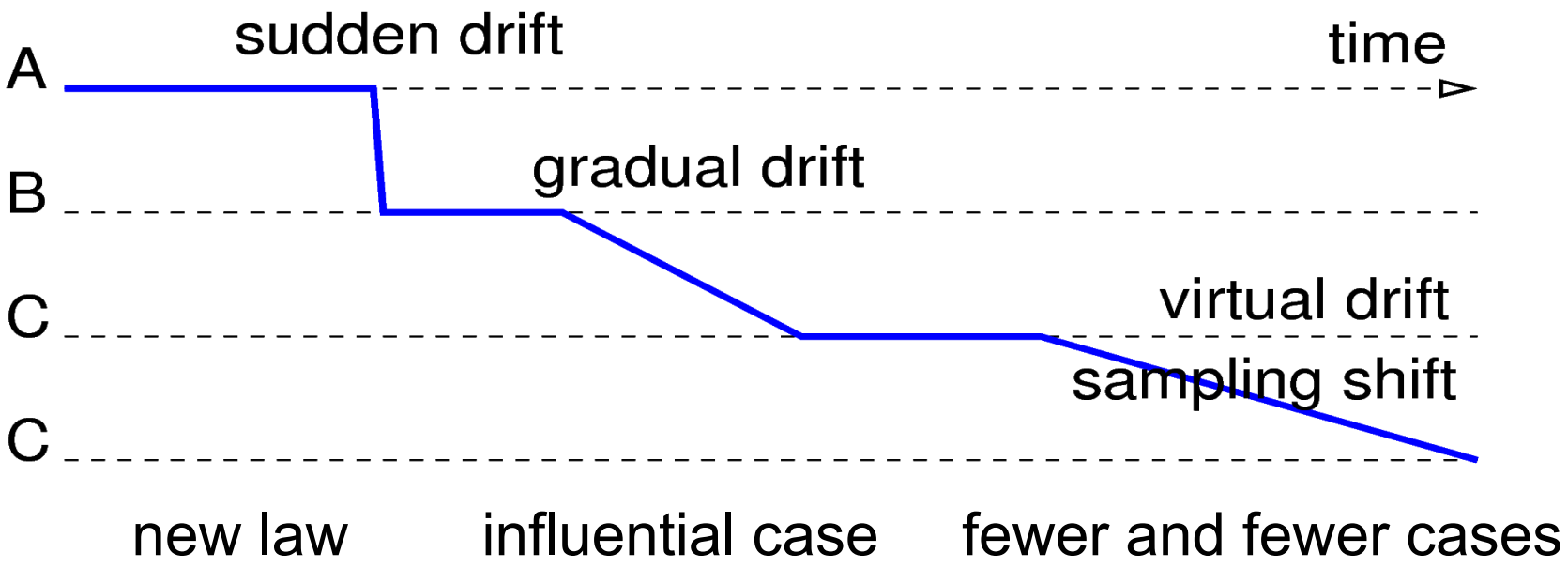
- **Concept**: mental category that help us classify objects
  - “reasonably-priced”
  - “spam-mail”
  - “fraudulent transaction”
  - “interesting literature“
- **Concept Learning**: train to classify objects by being shown a set of examples along with their label


Evolution in Classifiers – 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
- **Context**: period of time where the concept is stationary, context is *unobservable*
- **Concept Drift**: switch between contexts


Types of Change

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



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


Special Case: Concept Drift in Stream Mining

- 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


Frequently Used Learners as Starting Points

- frequently used learners
  - decision trees (beware of instability (Li & Belford, 2002))
  - naïve bayes
  - rules
  - instance-based learner (e.g. k-nearest neighbours)
  - support vector machines
- desirable for stream mining
  - low runtime complexity
  - incremental learner (naïve bayes, instance-based learner)


Spectrum of Approaches

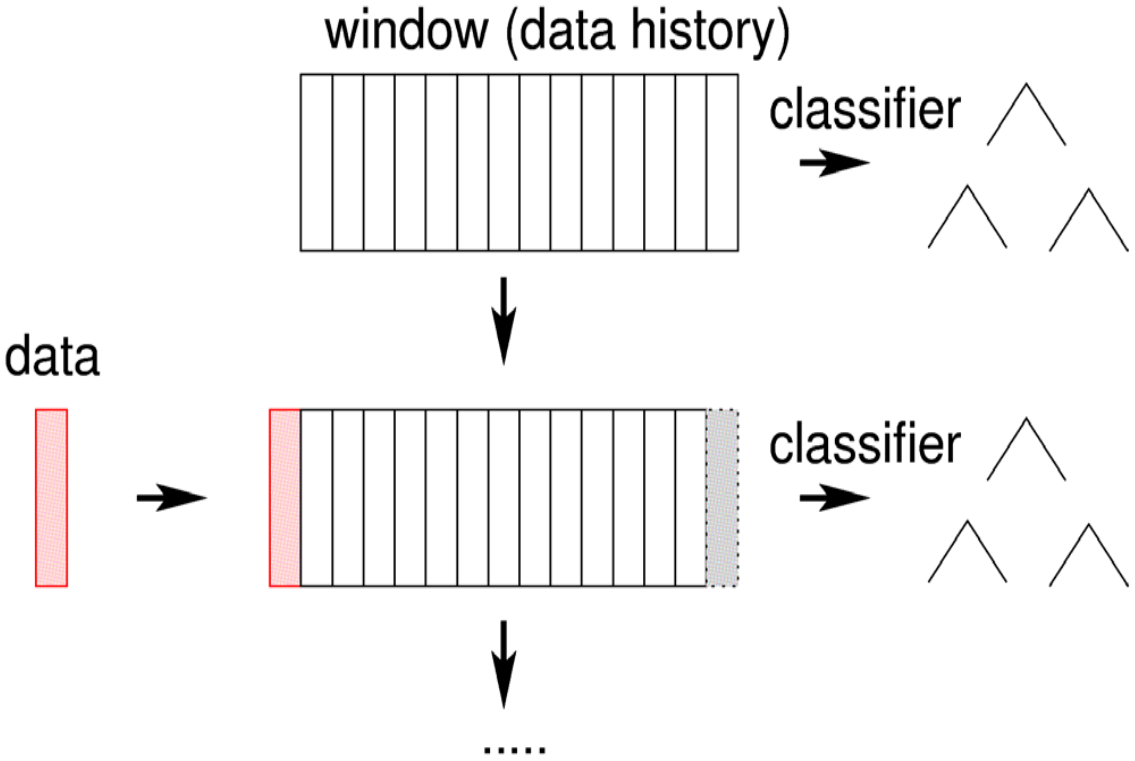
- Continuously Adapt
- Continuously Adapt & Detect
- Ensemble Methods
- Meta-Learning


Spectrum of Approaches

- **Continuously Adapt**
- Continuously Adapt & Detect
- Ensemble Methods
- Meta-Learning


Continuously Adopt Model

- recent data used for training (or storing in IBL)



- Alternatively assign weights (TWF, Salganicoff, 1997)

Continuously Adopt Model

- 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


Adjust Window Size

- 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 sizes by leave-one-out CV

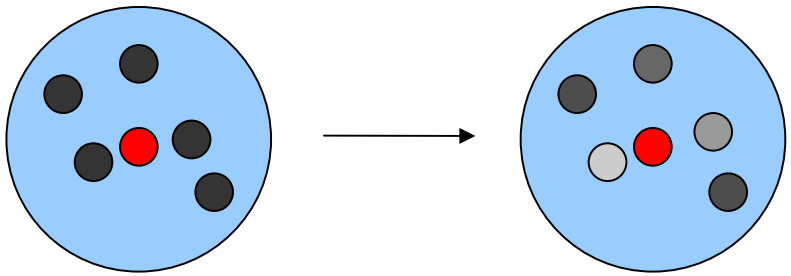
Other Forgetting Policies

- Data may be (ir-) relevant for other reasons
  - Local change – most of the old data may still be useful
  - Redundancy – compress similar data with same label
  - Consistency – remove similar but inconsistent data


Locally Weighted Forgetting (Salganicoff, 1997)

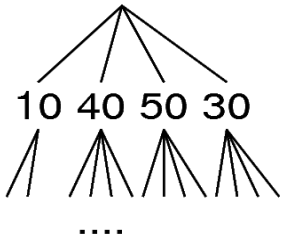
- Forget *redundant* data first:

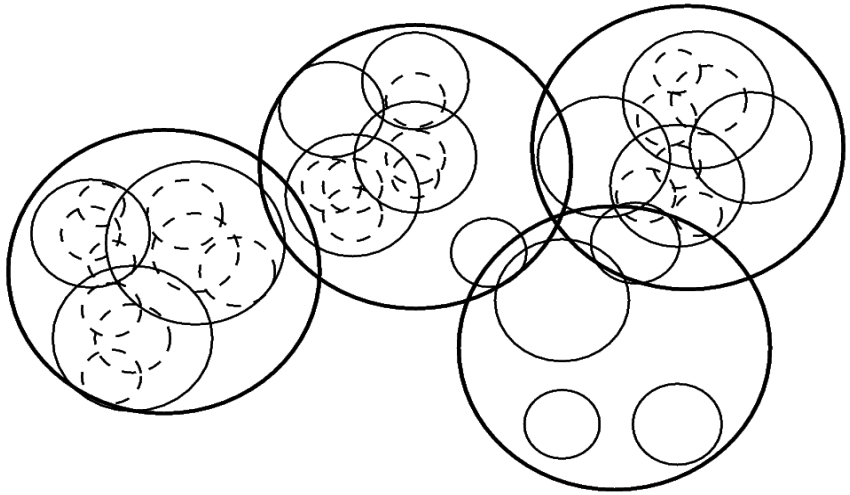
- All data starts with weight 1
- Weights of k nearest neighbours are adjusted
  - The closer the data to the new sample, the more the weight is decayed
  - If weight drops below some threshold, remove data



IBL-DS (Beringer&Hüllermeier, 2007)

- Remove *outliers*, but *keep coverage*:
- Find k nearest neighbors, majority-vote
- Remove old outliers (=wrong label) within k^2+k nearest neighbours
- Other data is removed spatially uniform, but temporally skev





Spectrum of Approaches

- Continuously Adapt
- **Continuously Adapt & Detect**
- Ensemble Methods
- Meta-Learning


Drift Detection

- How can drift detection help?
  - Control window (training data set) size more directly
    - Forget/Ignore more old data if a new context shows up
    - Gather more data if the context remains stable
  - Benefit: faster response

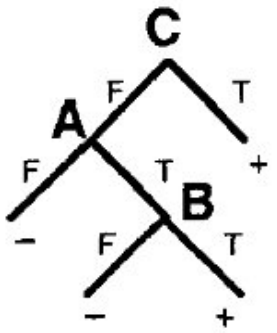

Drift Detection – FLORA (Widmer&Kubat 1996)

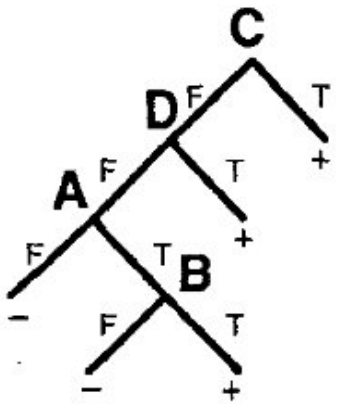
- FLORA (Widmer&Kubat 1996)
- Two drift indicators:
  - Accuracy goes down (on random sample)
  - Exploding number of conditions in hypotheses (rules)
- If drift is suspected, window width is reduced by 20% - thereby oldest data is dropped and adoption is faster


Drift Detection – Structural Instability

- Structural Concept Change (Rissland&Friedman, 1995)
  - Compare decision trees, measure structural instability
  - Depth-weighted sum of level-differences:

    $1.0*0 + 0.8*1 + 0.6*2 + 0.4*4 + 0.2*2 = 4.0$





Address in AB ∨ C	Value at address	Address in AB ∨ ¬C	Value at address
1	C	1	C
1.1	A	1.1	+
1.2	+	1.2	A
1.1.1	−		
1.1.2	B		
		1.2.1	−
		1.2.2	B
1.1.2.1	−		
1.1.2.2	+		
		1.2.2.1	−
		1.2.2.2	+

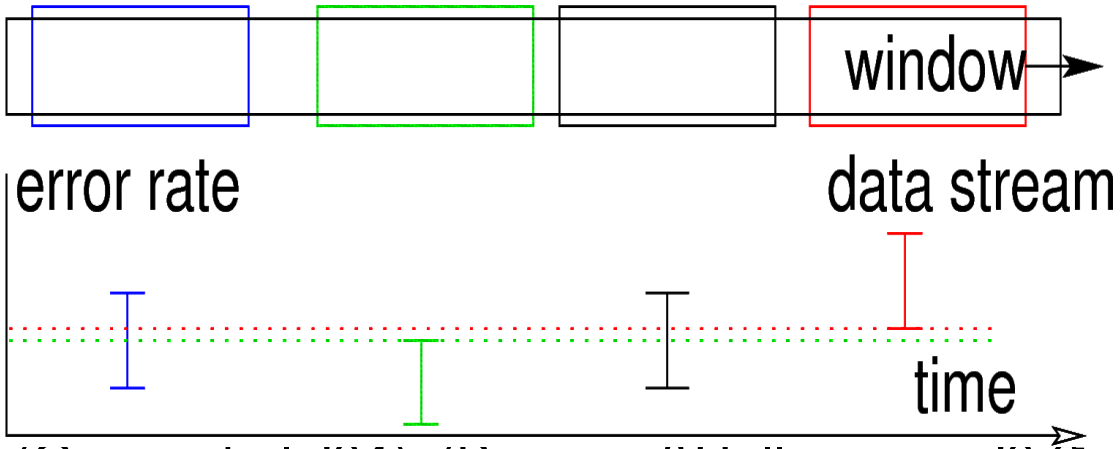
  - Drift Detection: test for positive correlation with time

Statistical Quality Control (Castillo et al, 2003)

- Idea: continuously monitor the stability of some quality characteristics
- If a distribution is normal, 99.7% of observations X lie within $\mu \pm 3\sigma$, so we consider the process being
  - In control: if X is within $\mu \pm 3\sigma$ - stable context
  - Out-of-control: if X is outside $\mu + 3\sigma$ - concept drift
- Here: use error rate as quality characteristics (dichotomous variable, success:0, failure:1)
- Use past data to estimate true μ and σ


P_{min}-Chart (Castillo et al, 2003)

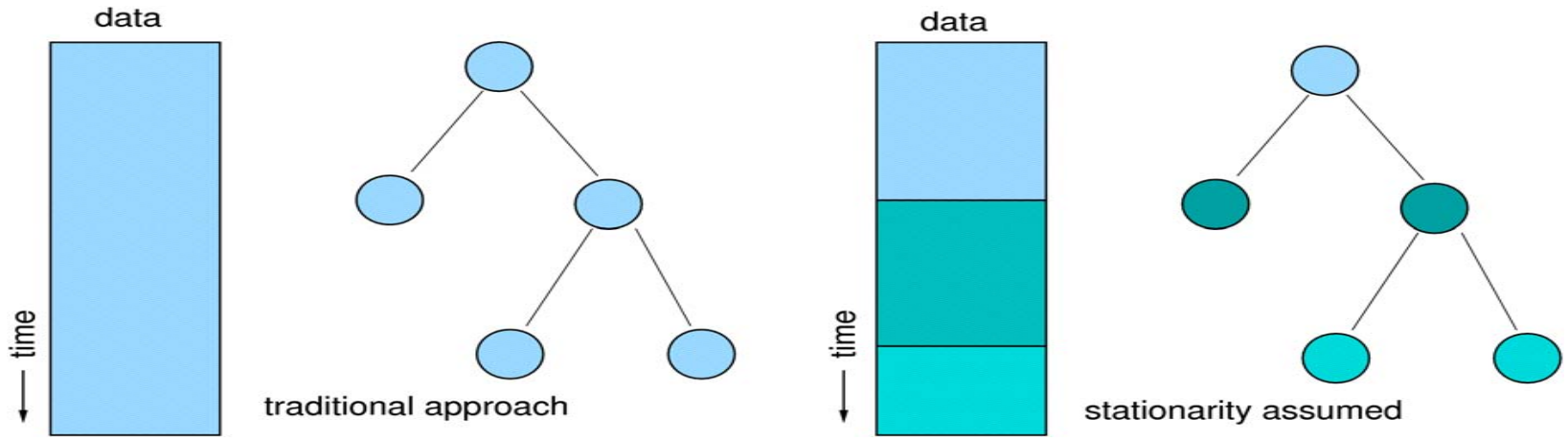
- Divide data into batches, estimate error rate for batch
- Start of new context: downward trend in error rate expected, optimum achieved as $\mu+\sigma$ becomes minimal
- Test for: current error > $\mu_{min}+z_{\alpha}\ \sigma_{min}$ at confidence level α
  shift: single exceedance, drift: series of exceedances



also used in (Gama et al, '04), (Beringer&Hüllermeier, '07),
(Baena-Garcia et al, 2006) use “time between errors” instead

VFDT (Domingos&Hulten, 2000)

- Stream Mining: VFDT = Very Fast Decision Tree Learner



- How much data is needed to safely induce node?
- Hoeffding-bound: estimates #samples necessary to acknowledge: $IG(X_{best}) - IG(X_{2ndbest}) > \varepsilon$, with prob. $1-\delta$

Concept-Adapting VFDT (Hulten et al, 2001)

- VFDT assumes stationarity
- No stationarity: node violates $IG(X_{best}) - IG(X_{2ndbest}) > \varepsilon$
- Drift suspected: start learning alternative **subtree**, exchange subtree if more accurate than current subtree
- Fast Adaptation by Forgetting: maintain window in memory (RAM or disk) to correct sufficient statistics at nodes


Spectrum of Approaches

- Continuously Adapt
- Continuously Adapt & Detect
- **Ensemble Methods**
- Meta-Learning


Ensemble Methods

- Learn a number of models on different parts of the data
- Weigh classifier according to recent performance
- If classifier performance degrades, it is replaced by new classifier


Dynamic Weighted Majority (Kolter & Maloof, 2003)

- Classifiers in ensemble have initially a weight of 1
- For every new instance:
  - If a classifier predicts incorrectly, its weight is reduced
  - If weight drops below threshold, the classifier is removed
  - If ensemble predicts incorrectly, a new classifier is installed
  - Finally, all classifiers are (incrementally) updated to consider the new instance


Accuracy Weighted Majority (Wang et al, 2003)

- Divide stream into chunk
- Learn new classifier from n such chunks, keep k top-performing classifiers
- Use most recent chunk to estimate expected accuracy of each classifier
- Weigh classifiers in the ensemble by expected accuracy (provably better results than averaging decisions)


Spectrum of Approaches

- Continuously Adapt
- Continuously Adapt & Detect
- Ensemble Methods
- **Meta-Learning**

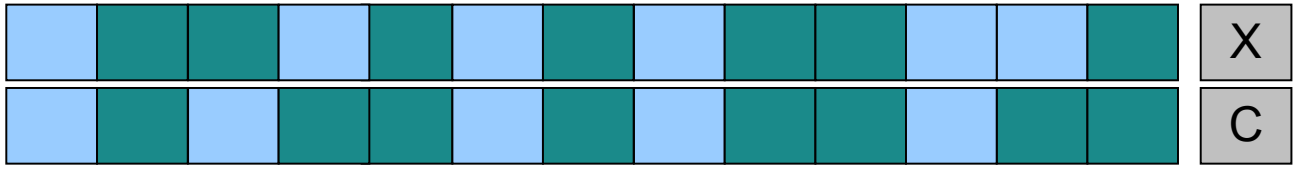

Meta-Learning

- If we could unveil the hidden context...
  - Generate new attributes that help to distinguish contexts
  - Closer to „classical“ problem

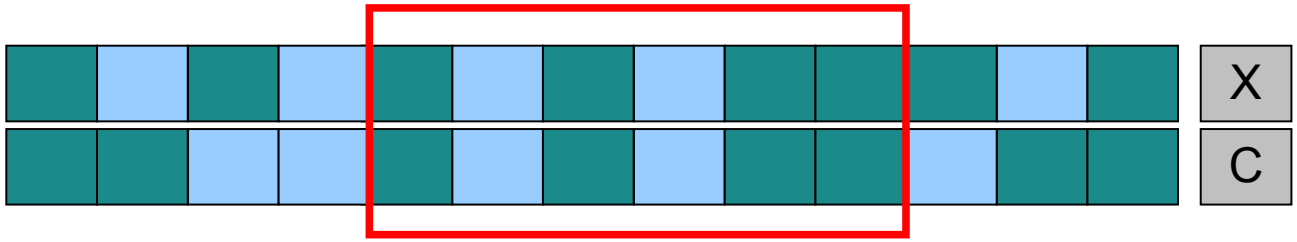
- Learn clues on context changes
  - classifier predicts the concept (one per context)
  - meta-classifier predicts the context
  - e.g. instance weighting and/or attribute weighting depends on the currently suspected context


MetaL(B) / MetaL(IB) (Widmer, 1997)

- Predictive feature: value correlates to class distribution
  - that is: p(class) different from p(class|feature)

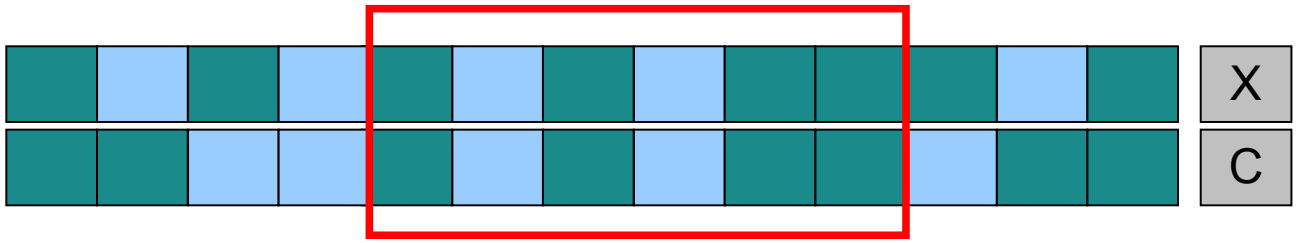


  - What if it is only predictive in some contexts?



MetaL(B) / MetaL(IB) (Widmer, 1997)

- Contextual feature: predictive of predictive features
  - Typically not predictive themselves, but give contextual clues on the (temporal) predictiveness of other features



- Contextual features *focus* on right training set
  - For a prediction in some context, only data from the same context (within the window) is used
  - ..., only the predictive features are considered for classification (feature selection guided by context)

References (Block 3)

- Beringer, Hüllermeier. Efficient Instance-Based Learning on Data-Streams. Intelligent Data Analysis, 2007, 11, 627-650
- Castillo, Gama, Medas. Adaptation to Drifting Concepts. Proc. Extraction of Knowledge from Data Bases, 2003, 2902, 279-293
- Domingos, Hulten. Mining high-speed data streams. KDD 2000, 71-80

  Hulten, Spencer, Domingos. Mining Time Changing Data Streams. KDD 2001, 97-106
- Hulten, Spencer, Domingos. Mining Time Changing Data Streams. KDD 2001, 97-106
- Klinkenberg, Joachims. Detecting Concept Drift with Support Vector Machines. ICML 2000, 487-494

References (Block 3)

- Kolter, Maloof. Dynamic weighted majority: a new ensemble method for tracking concept drift. ICDM 2003, 123-130
- Li, Belford. Instability of decision tree classification algorithms. KDD 2002, 570-575
- Rissland, Friedman. Detecting change in legal concepts. Int. Conf. on Artificial Intelligence and Law, ACM, 1995, 127-136
- Salganicoff. Tolerating Concept and Sampling Shift in Lazy Learning Using Prediction Error Context Switching. Artificial Intelligence Review, 1997, 11, 133-155
- Wang, Fan, Yu, Han. Mining concept-drifting data streams using ensemble classifiers KDD 2003, 226-235
- Widmer. Tracking Context Changes through Meta-Learning. ML, 1997, 27, 259-286
- Widmer, Kubat. Learning in the Presence of Concept Drift and Hidden Contexts ML, 1996, 23, 69-101

End of Block 3

Thank you!

Questions?


Break !

Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
  - incremental clustering
  - stream clustering
  - spatiotemporal clustering
  - change detection in clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


Cluster adjustment and the role of population change

If we assume that the population is not changing,

a clustering $P=\{C_1,C_2,\ldots,C_n\}$ is applied as a classifier:

- For each newly arriving record x
  - return C in P, so that x belongs to C.
  - forget x

Cluster adjustment and the role of population change

If we assume that the population is not changing,

a clustering $P=\{C_1,C_2,\ldots,C_n\}$ is applied as a classifier:

- For each newly arriving record x
  - return C in P, so that x belongs to C.
  - forget x

If the population is changing, then the clustering

$P=\{C_1,C_2,\ldots,C_n\}$ must be adjusted to each new record:

- For each newly arriving record x
  - identify C in P, so that x belongs to C
  - update C and its surroundings **Local methods**

Cluster adjustment and the role of population change

If we assume that the population is not changing,

a clustering $P=\{C_1,C_2,\ldots,C_n\}$ is applied as a classifier:

- For each newly arriving record x
  - return C in P, so that x belongs to C.
  - forget x

If the population is changing, then the clustering

$P=\{C_1,C_2,\ldots,C_n\}$ must be adjusted to each new record:

- For each newly arriving record x
  - identify C in P, so that x belongs to C
  - update P

Global methods


Cluster adjustment: The termini

- Cluster adjustment methods are published as
  - incremental clustering
  - stream clustering
  - topic evolution (documents only)
- and face some common challenges:
  - How to minimize the amount of old data being re-read?
  - When to adjust and when to re-cluster from scratch?
  - What changes can occur?
  - What is a cluster?

... to a cluster ?
... among clusters ?
How many clusters ?


Presentation Outline

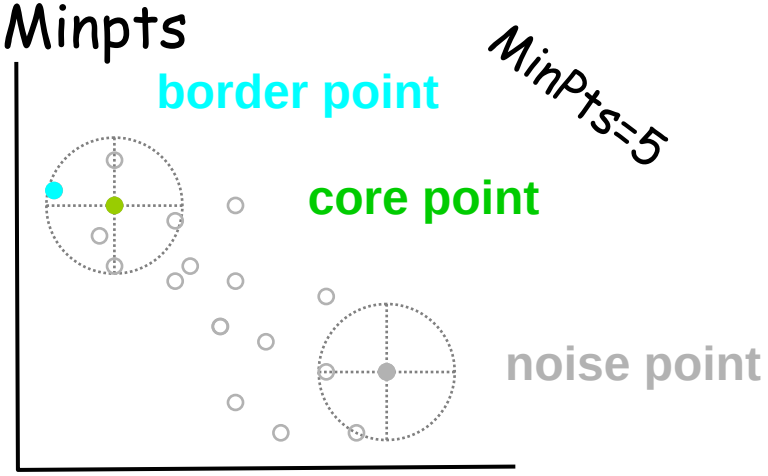
- Block 1: Introduction
- Block 2: Evolution in Association Rules
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
  - incremental clustering
  - stream clustering
  - spatiotemporal clustering
  - change detection in clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


Cluster adjustment with a *local method:* Incremental DBSCAN

- DBSCAN (Ester et al, KDD96) is a clustering algorithm designed for noisy data.
  - It builds *neighbourhoods* of objects.
  - It builds clusters as groups of overlapping neighbourhoods.
- Incremental DBSCAN (Ester et al, VLDB98) extends DBSCAN by allowing insertion and deletion of records:
  - For each insertion, it identifies its impact on the clusters,
  - for each deletion, it identifies its impact on the record's cluster,
  - and adjusts neighbourhoods and clusters accordingly.

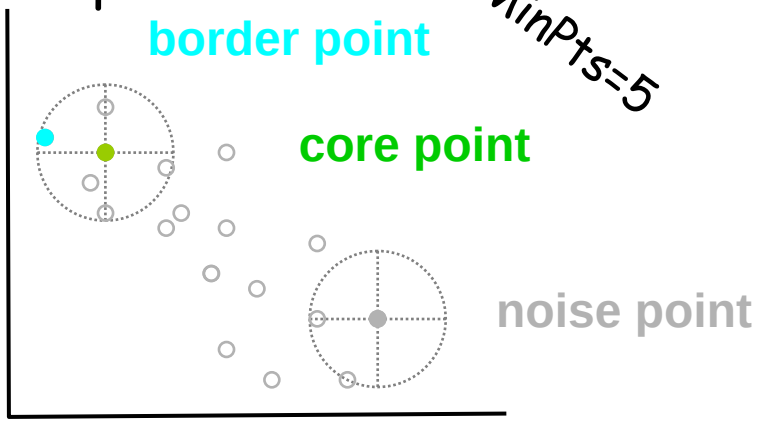

Cluster adjustment with Incremental DBSCAN: Underpinnings of DBSCAN

- *Neighbourhood* around a point p:
  - area of fixed radius *eps*
  - subject to a cardinality threshold *Minpts*



Cluster adjustment with Incremental DBSCAN: Underpinnings of DBSCAN

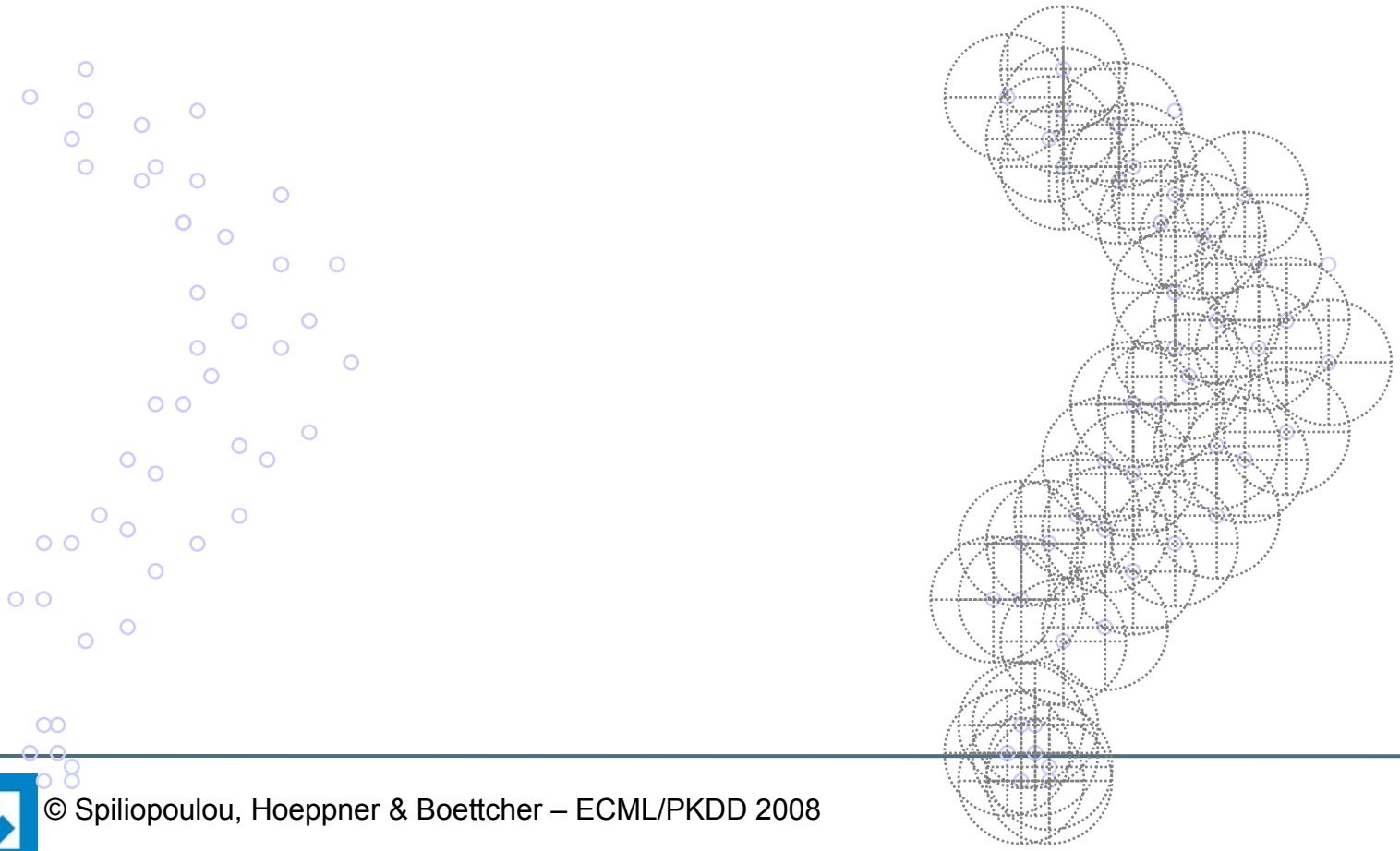
- *Neighbourhood* around a point p:
  - area of fixed radius *eps*
  - subject to a cardinality threshold *Minpts*



- *Cluster* as maximal set of overlapping neighbourhoods

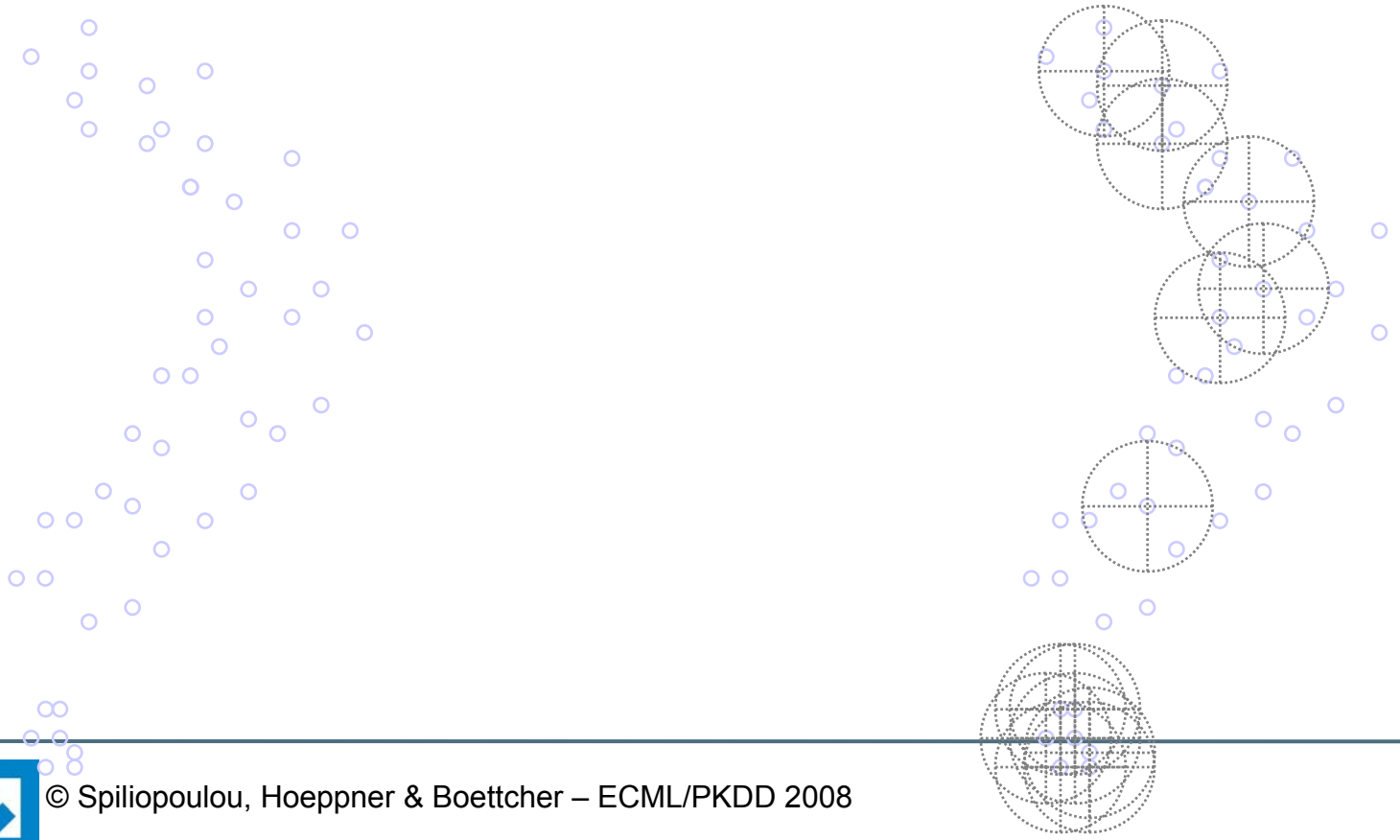
Example of DBSCAN

- MinEps=5, Eps=1



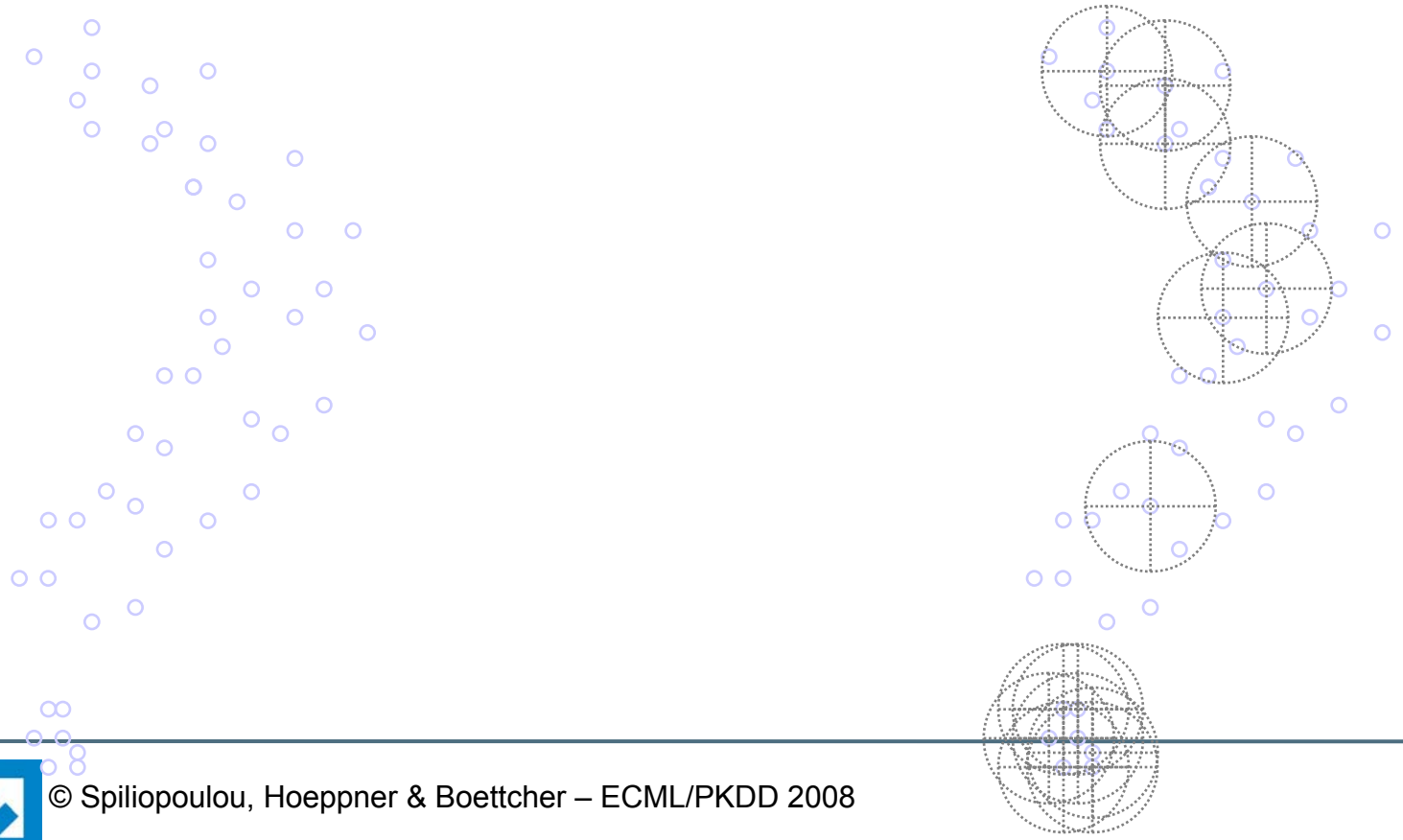
Example of DBSCAN

- MinEps=5, Eps=1



Example of DBSCAN

- MinEps=5, Eps=1

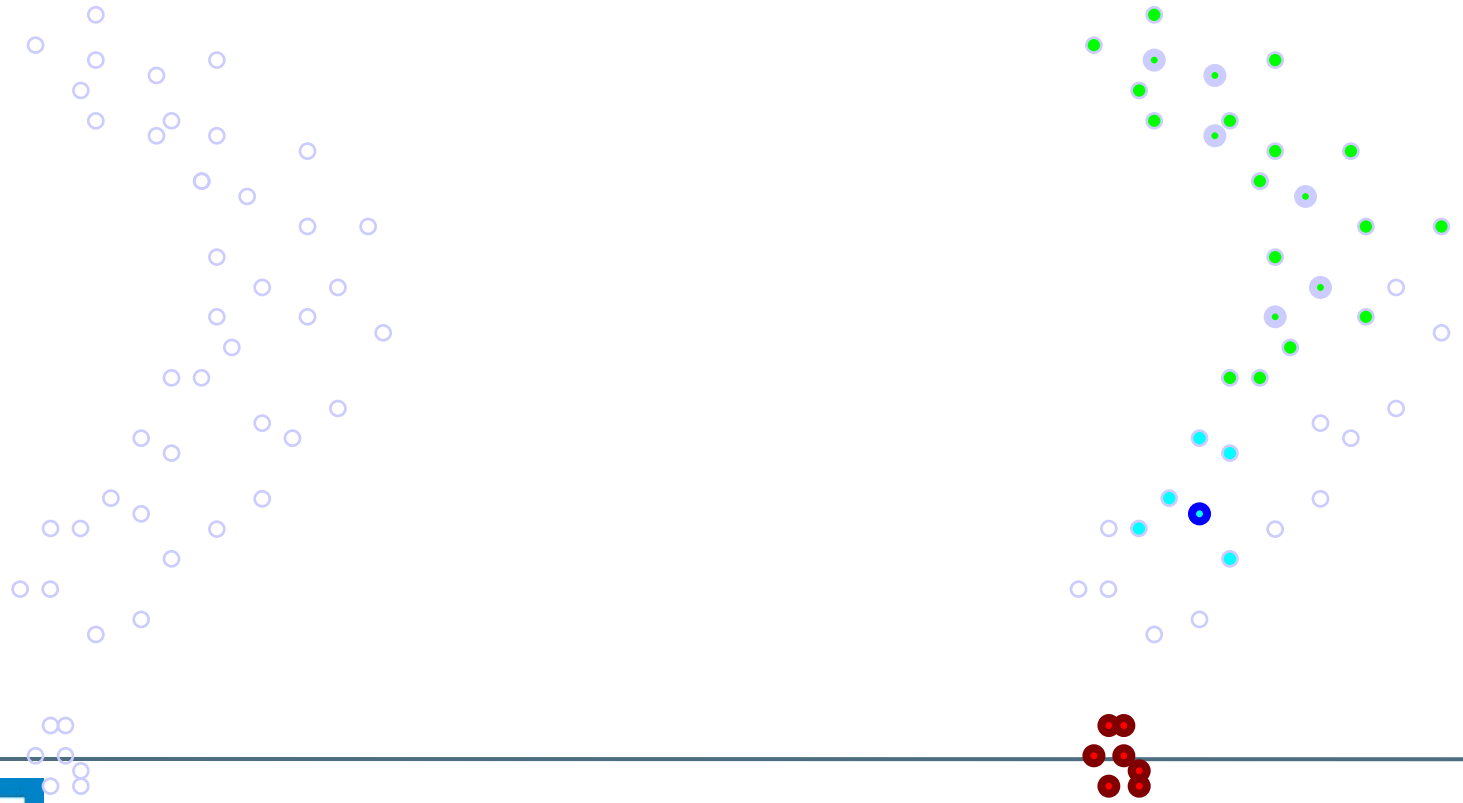


Cluster adjustment with Incremental DBSCAN: Underpinnings of DBSCAN

- A *cluster* C is a set of data points such that
  - any two of its members are density-connected and
  - each data point that is density-connected to a member of C also belongs to C.
- *Density-reachability* and *Density-connectivity:*
  - q is directly density-reachable from p iff
    p is a core point and q is in its neighbourhood
  - q is density-reachable from p iff
    there is a path of core points $p_1,\ldots,p_n$ such that $p=p_1$,
    p_{i+1} is directly density-reachable from p_i (i=1,…,n-1)
    and q is directly density-reachable from p_n
  - p, q are density-connected iff
    there is a core point o from which they are both density-reachable.

Example of DBSCAN

- MinEps=5, Eps=1

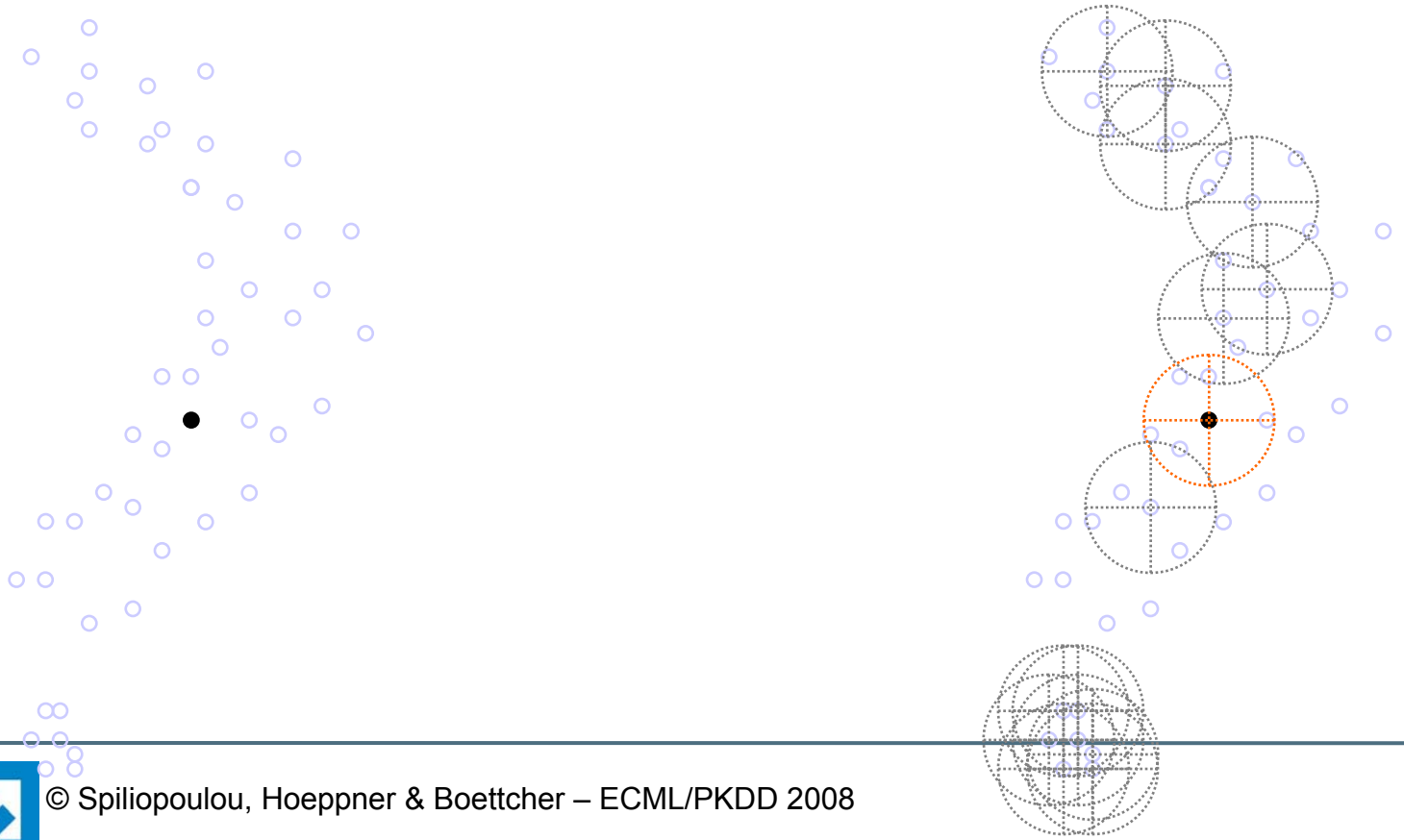


Cluster adjustment with a *local method:* Incremental DBSCAN

- Incremental DBSCAN (Ester et al, VLDB98) extends DBSCAN by allowing insertion and deletion of records:
  - For each insertion, it identifies points that have newly acquired the *core property.*
  - For each deletion, it identifies points that have lost the *core property.*
  - It adjusts neighbourhoods and clusters accordingly.

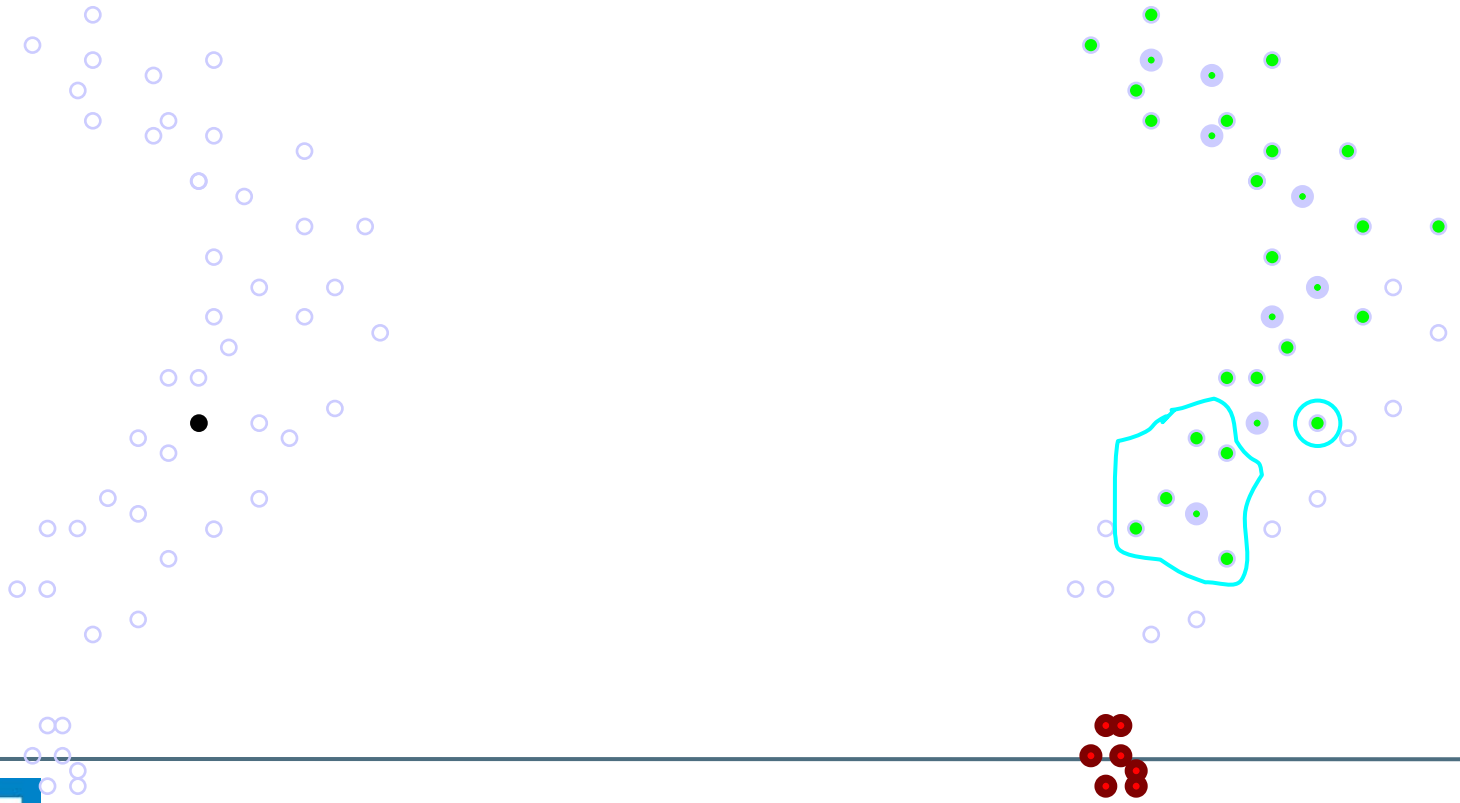

Example of Incremental DBSCAN: Adding one data point

- MinEps=5, Eps=1



Example of Incremental DBSCAN: Adding one data point

- MinEps=5, Eps=1



IncrementalDBSCAN: Computing a minimal number of affected points

- Given is an update Δ of the original dataset D
  i.e. a set of insertions and a set of deletions:

1. Identification of the affected points:
   - O_1: Core points of D that have lost the core property
   - O_2: Non-core points of D that have acquired the core property
2. Computation of the seed points:
   - S_{Ins}: Core points in the neighbourhood of O_2 data points
   - S_{Del}: Core points in the neighbourhood of O_1 data points


IncrementalDBSCAN: Updating the clusters incrementally

- Depending on the contents of the seed S_{Ins} :
  - a new cluster is created,
  - existing clusters are merged,
  - the new point is placed in an existing cluster, or
  - it is marked as noise.
- Depending on the contents of the seed S_{Del} :
  - a cluster may shrink, losing some core points,
  - a cluster may be deleted,
  - a cluster may be split.


IncrementalDBSCAN: The seed after one insertion $D' = D \cup \{p\}$

- The S_{Ins} consists of all core points in the neighbourhood of points that have acquired the core property.
  - S_{Ins} is empty. ➔ p is marked as noise.
  - S_{Ins} consists of noise points of D. ➔ New cluster : S_{Ins}U {p}
  - S_{Ins} consists of points that belonged to one cluster.
    ➔ This cluster absorbs p.
  - S_{Ins} contains points from different clusters.
    ➔ The clusters are merged and p is put to the new cluster.

IncrementalDBSCAN: The seed after one deletion $D' = D - \{p\}$

- The S_{Del} consists of all core points in the neighbourhood of points that have lost the core property.
  - S_{Del} is empty. ➔ A cluster loses some of its core points. If it has no more core points, it is removed.
  - S_{Del} consists of directly density-reachable points. ➔ Some neighbours of p lose the core property.
  - S_{Del} contains points that are not directly density-reachable from each other. ➔ A cluster split may have occurred.
    - If all points are still density-reachable, no split has occurred.
    - If some points are separated from the others, then the fragments of the original cluster become new clusters (subject to MinPts).

Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
  - incremental clustering
  - stream clustering
  - spatiotemporal clustering
  - change detection in clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


Cluster adjustment with a *global method:* Premises

- Optimization problem for a data set :

  Partition the dataset while
  - minimizing the avg distance to the closest cluster center
  - minimizing the maximum cluster radius
  - ...

- Optimization problem for a data stream ?


Stream clustering (Guha et al, 2000): Premises

- Data stream model of computation

Stream: Sequence of records $x_1, \ldots, x_n$ arriving in increasing order of i

  - Space/Memory constraints
  - Computation time constraints
  - Data are seen only once
- Objective of stream clustering:
  - Maintain a good clustering of the sequence seen thus far
  - while satisfying the space and time constraints.


Stream clustering (Guha et al, 2000): A simplified outline

As many data as fit in memory.

Repeat

- divide the data in partitions,
- cluster each partition,
- weight the centers in each partition and then
- cluster the weighted centers **Repeat**

A cluster center is a representative of the data in the cluster.
We use the center hereafter and forget the data.


Stream clustering (Guha et al, 2000): Core operation for O(k)=2k

Repeat

- Read the next m points in sequence
- Cluster them to 2k intermediate medians
- Weight each intermediate median with the number of points assigned to it

until $m^2/(2k)$ points have been seen.

At the end of this step, we have k medians – the level-1 medians.


Stream clustering (Guha et al, 2000): Core operation for O(k)=2k

Repeat

Repeat

- Read the next m points in sequence
- Cluster them to 2k intermediate medians
- Weight each intermediate median with the number of points assigned to it

until $m^2/(2k)$ points have been seen.

At the end of the i^{th} iteration, we have k level-i medians

for u iterations

Cluster the level-u medians to the final k medians.


Cluster adjustment vs Cluster change detection

- Objective of cluster adjustment methods:
  - Maintain a good (optimal) clustering
  - under time and space constraints
  - avoiding to re-read the data

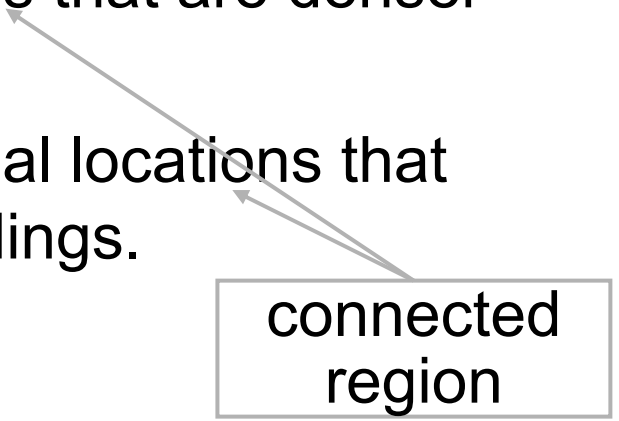
- How does the clustering change?


Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
  - incremental clustering
  - stream clustering
  - spatiotemporal clustering
  - change detection in clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


Spatiotemporal cluster evolution (Aggarwal, 2005): Premises

- Data are points in a multidimensional space.
- Time is an additional dimension
  across which the data population can evolve.
- A cluster is a group of spatial locations that are denser than their surroundings.
- An evolving cluster is a group of spatial locations that change differently from their surroundings.



Spatiotemporal cluster evolution (Aggarwal, 2005): Core ideas

- Model of the data space as a grid of "spatial locations"
- Evolution at a spatial location X:
  - Kernel density estimation at time t
  - Density *change* estimation at (X,t):
    - forward density estimate for the window t+h
    - backward density estimate for the window t-h
  - Velocity density estimation at (X,t) given window h:
    - Difference of forward and backward density estimates

Spatiotemporal cluster evolution (Aggarwal, 2005): Core ideas

- Model of the data space as a grid of "spatial locations"
- Evolution of a connected region of spatial locations:
  - Integral over the locations of the region
- Evolution coefficient for a time slice (t,t'):
  - Integral over the locations of the region and the slice
  - Distinction between
    - global evolution coefficient E for all dimensions
    - coefficient for the *projection* across some dimensions

  to detect dimensions that exhibit faster change than others.


Spatiotemporal cluster evolution (Aggarwal, 2005): Aspects of change

- Data coagulation:
  - A connected region that exhibits higher velocity of change than a positive threshold
- Data dissolution:
  - A connected region that exhibits lower velocity of change than a negative threshold
- Epicenter of coagulation / dissolution:
  - Spatial location that shows the highest positive (resp. lowest negative) velocity of change within the region

Cluster change detection: Changes involving one or more clusters

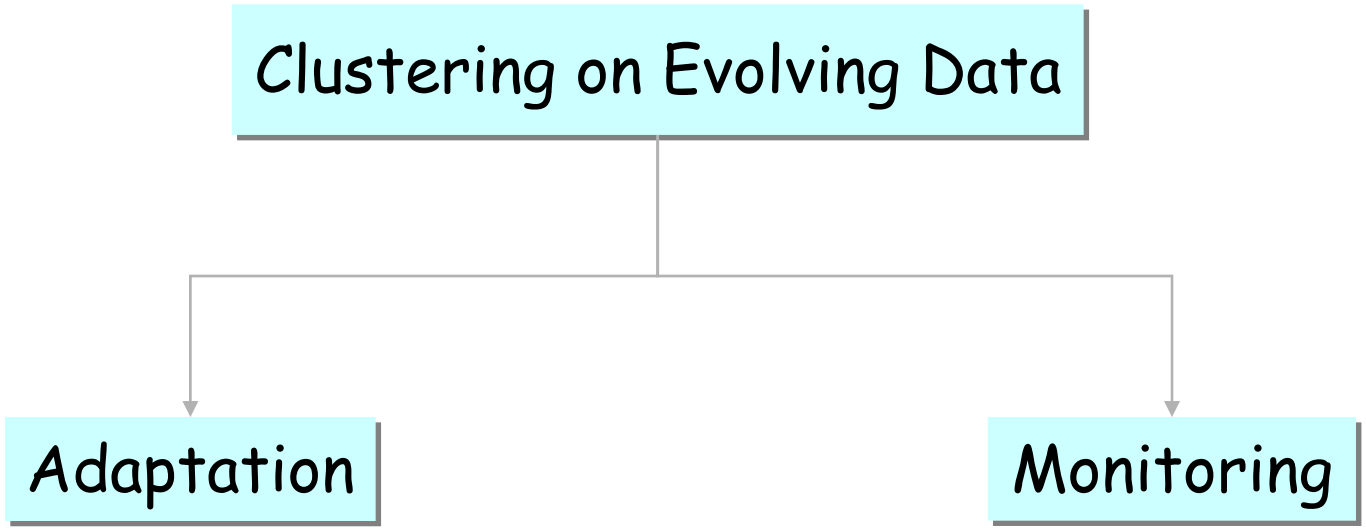
- Incremental DBSCAN (Ester et al, 1998):
  - A cluster may shrink or get split.
  - A cluster may grow.
  - A cluster may merge with another cluster.

- Spatiotemporal cluster evolution (Aggarwal, 2005):
  - A region (as cluster) may dissolve or coagulate.
  - There is an epicenter of change within the region.
  - Some spatial dimensions are more responsible for change than others.


Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
  - incremental clustering
  - stream clustering
  - spatiotemporal clustering
  - change detection in clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


Cluster monitoring for cluster change detection



Adjust the clusters so that they constitute a good clustering of the current population

Detect changes and reason on them to gain insights on the population

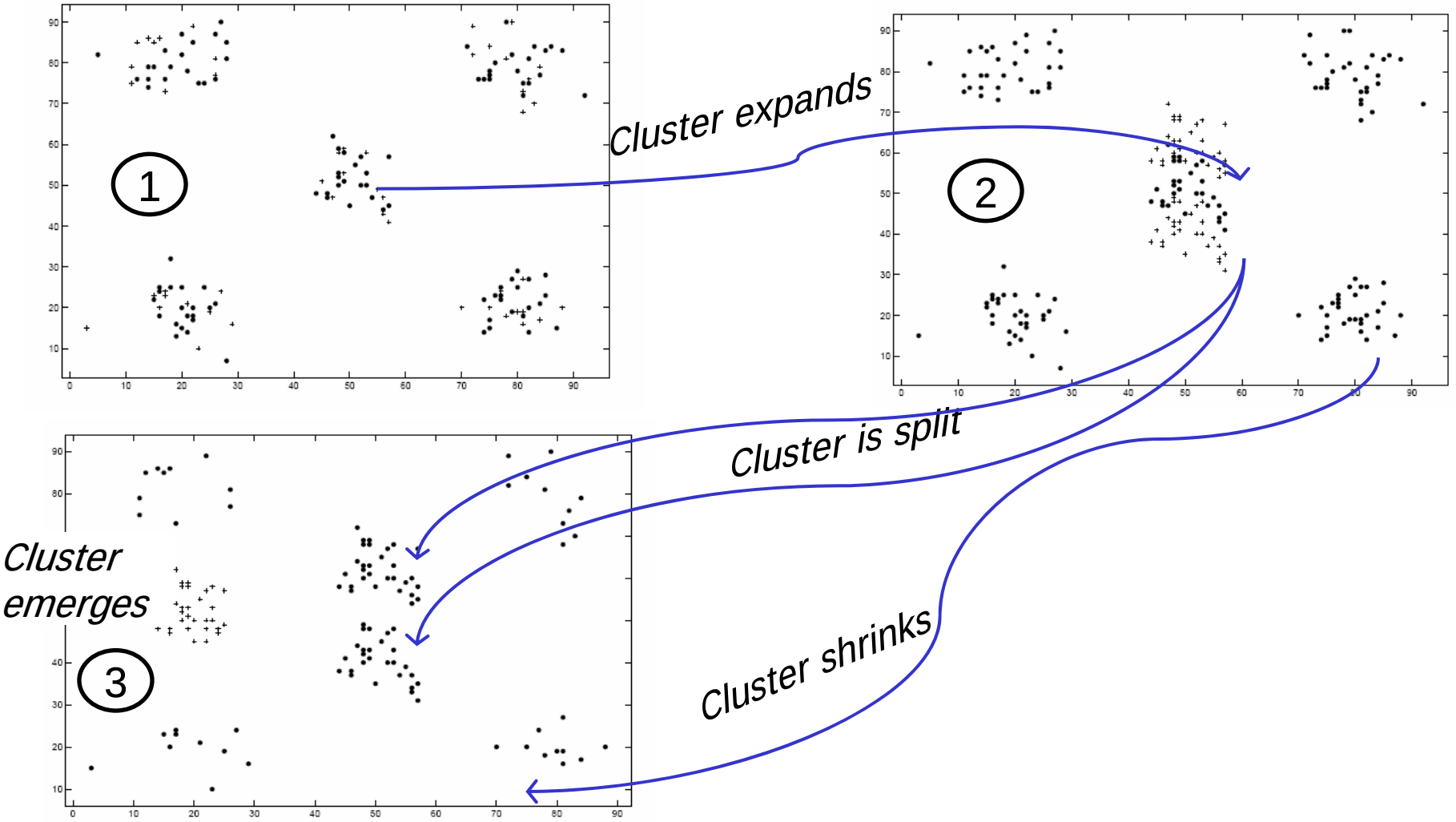
Cluster monitoring: What is a *cluster* ?

- A region in a geometric space:
  - A sphere:
    - K-means: All members of a cluster are closer to its centroid than to other centroids.
  - A group of spheres:
    - (Ester at al, KDD 1996): A cluster is a maximal sequence of overlapping spheres (neighbourhoods).
  - A connected region:
    - (Aggarwal, 2005): A cluster is a connected region of spatial locations that have the same density.
    - (Neill et al, 2005): A cluster is a region that is more homogeneous than the surrounding space.


Cluster monitoring: How does a *cluster change*?

- A region for an evolving population:
  - Tracing changes on spheres:
    - (Abrantes & Markes, 1998): Tracing the motion of centroids
    - (Charikar et al, 1997): Re-computing the minimal number of clusters after an update
  - Tracing changes on groups of spheres:
    - (Ester et al, 1998): Identifying points that lose or acquire the core property
  - Tracing changes in a connected region:
    - (Aggarwal, 2005): Computing the velocity of change, the evolution coefficient and the epicenter of a changing region
    - (Neill et al, 2005): Juxtaposing the density of the cluster to that of its surroundings


Cluster monitoring in a geometric space at three time points



Cluster monitoring: What is a *cluster* ?

- All objects of a region
  - The dimensions and the metric are invariant.
- All objects satisfying a function
  - The dimensions are invariant.
- A set of objects


Cluster monitoring: Premises and challenges for change detection

- We partition the time axis into timepoints t_1, ..., t_m and
- we cluster (or adjust?) the data at each timepoint
- gradually forgetting old data.

Cluster monitoring: Premises and challenges for change detection

- We partition the time axis into timepoints t_1, ..., t_m and we cluster (or adjust?) the data at each timepoint gradually forgetting old data.
- *Matching model:*
  - How to track a given cluster?
  - When is a new cluster a modification of an old one and when is it really new?
- *Transition model:*
  - Is an old cluster associated with exactly one new cluster?
  - How did the cluster change with respect to other clusters?
  - What kind of internal changes did the cluster experience?


MONIC – Matching model

- Given is a cluster X at clustering ξ, built at timepoint t.
- Let ξ' be the clustering built at the next timepoint t'.
- For each Y in ξ', the *overlap* of Y with X is

$$overlap(X,Y) = \frac{\sum_{a \in X \cap Y} age(a, t_2)}{\sum_{x \in X} age(x, t_2)}$$

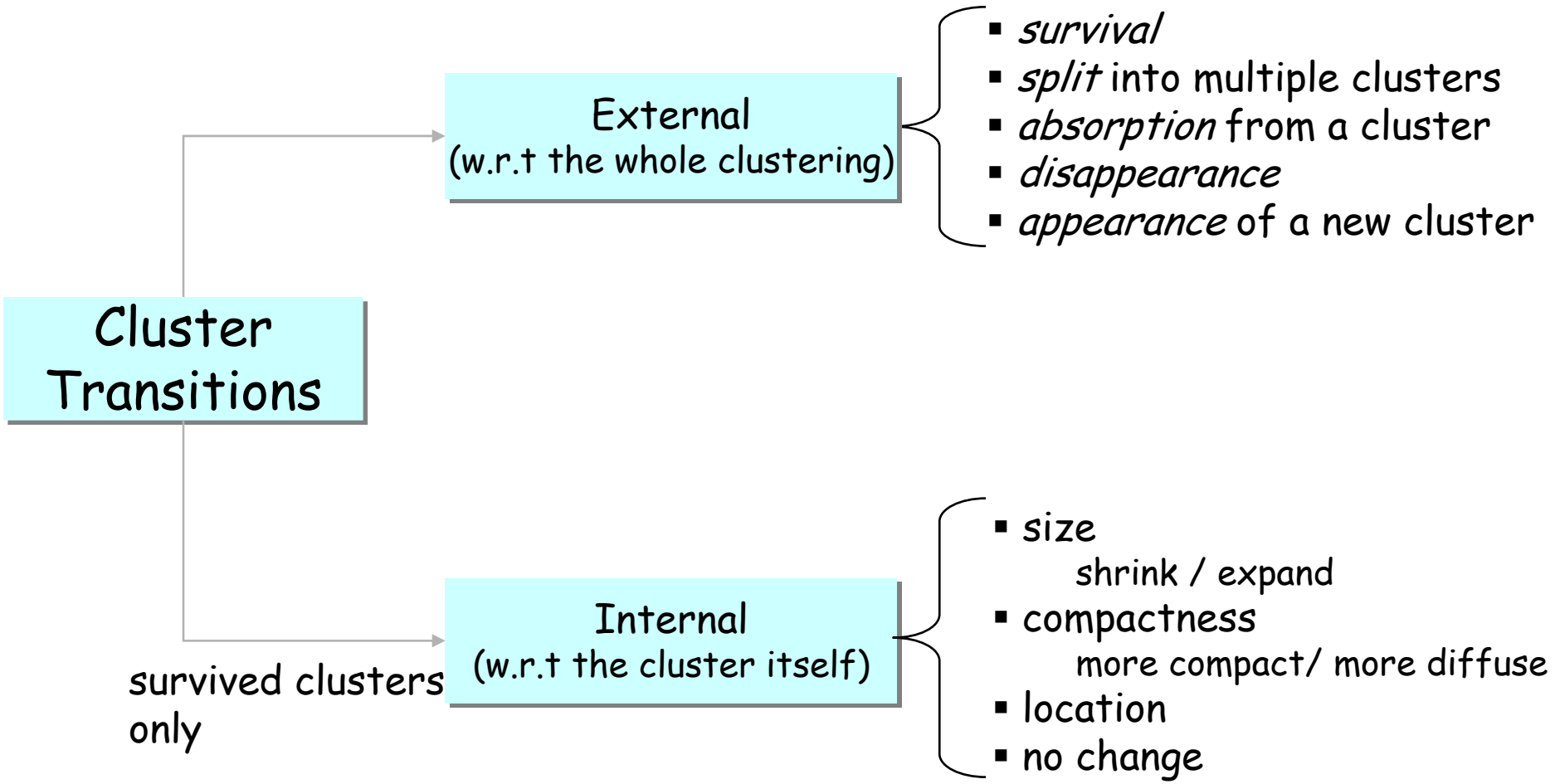
MONIC – Matching model

- Given is a cluster X at clustering ξ, built at timepoint t.
- Let ξ' be the clustering built at the next timepoint t'.
- For each Y in ξ', the *overlap* of Y with X is

$$overlap(X,Y) = \frac{\sum_{a \in X \cap Y} age(a, t_2)}{\sum_{x \in X} age(x, t_2)}$$

- Cluster X may have a non-empty overlap with many clusters in ξ'.
- A cluster Y in ξ' is the *best match* or *match* for X iff
  - There is no cluster in ξ' with higher overlap with X and
  - the overlap exceeds a threshold.


MONIC – Transition model



MONIC – Transition model: External transitions

External transitions of cluster X in ξ towards clustering ξ':

- absorption
- survival
- split
- disappearance
- new cluster

There is a cluster Y in ξ' that is a match for X and for at least one more cluster in ξ.

There is a cluster Y in ξ' that is a match for X and for no other cluster in ξ.

There Y1,...Yp in ξ' that form together a match for X and the overlap of each one with X exceeds a threshold.

There is no match for X in ξ'.

Y is not the result of a transition.

MONIC – Transition model: Internal transitions

Let X be a cluster in ξ at time t that has survived in ξ' at t':

- Size transition := change in cardinality
  - cluster expands
  - cluster shrinks
- Compactness transition:= change in homogeneity
  - Cluster becomes more compact
  - Cluster becomes more diffuse
- Location transition as

  shift of the area

  shift in the distribution
- No change

MONIC – Transition model: Internal transitions and heuristic implementations

Let X be a cluster in ξ at time t that has survived in ξ' at t':

- Size transition := change in cardinality
  - cluster expands $\sum_{y \in Y} age(y,t') > \sum_{x \in X} age(x,t) + \varepsilon$
  - cluster shrinks $\sum_{y \in Y} age(y,t') < \sum_{x \in X} age(x,t) - \varepsilon$
- Compactness transition:= change in homogeneity
  - Cluster becomes more compact $\sigma(Y) < \sigma(X) - \delta$
  - Cluster becomes more diffuse $\sigma(Y) > \sigma(X) + \delta$
- Location transition as

  shift of the area $|\mu(X) - \mu(Y)| > \tau_1$

  shift in the distribution $|\gamma(X) - \gamma(Y)| > \tau_2$
- No change

MONIC – Lifetime of clusters and clusterings

- Lifetime of a cluster X :=

  Number of adjacent timepoints where X has survived
  - Strict lifetime
  - Lifetime under internal transitions
  - Lifetime with absorptions

The Passforward ratio indicates the extend to which a clustering describes the data of the next timepoint.

- Survival ratio of a clustering :=

  Portion of clusters that survive at the next timepoint
- Passforward ratio of a clustering:=

  Portion of clusters that survive or get absorbed at the next timepoint


MONIC – Lifetime of clusters

- Lifetime of a cluster X :=

  Number of adjacent timepoints where X has survived
  - Strict lifetime
  - Lifetime under internal transitions
  - Lifetime with absorptions

For each cluster X encountered in the last timepoint
    set Lifetime(X) = 1
For each earlier timepoint t[i]:
    find each cluster X that has survived at t[i+1] into Y
    set Lifetime(X) = Lifetime(Y)+1

MONIC in experiments: Monitoring thematic evolution in an archive

ACM digital library section H2.8 on Database applications:

- Data Mining — Big class, larger than all other classes together
- Spatial Databases — Large class
- Image Databases — Large class
- Statistical Databases — Small class
- Scientific Databases — Small class
- *unspecified* — Documents that are not assigned to any of the classes

We hide the labels and study the thematic evolution from 1997 to 2004.

MONIC on the ACM section H2.8: Clustering the data …

- Vectorization of the document collection:
  - For each document, we consider
    - the title
    - the list of keywords

    as TFxIDF vectors
  - Feature space: 30 most frequent words
- Data ageing:
  - Sliding window of size 2 (years)
- Clustering:
  - Bisecting K-means, K = 10 (rather than 6)


MONIC on the ACM H2.8: Cluster transitions

- Cluster transitions:
  - No absorptions
  - Many disappearances, many splits
  - Size transitions in all surviving clusters
  - Sensitivity to overlap threshold, no survivals $\tau > 0.7$
  - Maximum lifetime of a cluster is 4 years, clusters are unstable
  - Most survivals are in the early years


MONIC on the ACM H2.8: Passforward ratio

- Passforward ratio = Survival ratio
- K=10 clusters

τ	1999	2000	2001	2002	2003	2004
0.45	4	7	7	1	5	4
0.50	4	5	7	1	3	4
0.55	3	3	3	0	2	3
0.60	3	2	3	0	1	1
0.65	3	0	1	0	0	1
0.70	2	0	1	0	0	0

MONIC on the ACM H2.8: Interpreting cluster evolution and population shift

We label clusters with their two most frequent words, subject to a frequency threshold.

- At each year, there are 2-3 clusters on Data Mining.
  - Cluster *Association Rules*:
    - Before 2002: it grows.
    - 2002: split into *Association Rules* and a noisy cluster.
    - 2003: disappears, 2004: re-appears.
  - Clusters *Data Mining*, *Knowledge Discovery*: Lifetime <= 3
- Clusters *Spatial* and *Image*.
  - Only encountered until 2002.
  - *Image* evolves into *Image Retrieval*.


Cluster monitoring: Interpreting cluster changes and population shifts

- Cluster transitions may be due to
  - Changes in (a part of) the population
  - Shortcomings of the clustering algorithm
  - Shortcomings of the clustering algorithm towards the current distribution of the population
- How to guarantee that a cluster change reflects a population change?
- What changes can occur in a clustering and
- how to detect them?


References and Readings (Block 4)

- A.J. Abrantes and J.S. Marques (1998). "A method for dynamic clustering of data", In Mark Nixon and Joh Carter, editors, Proc. of BMVC98.
- Charu Aggarwal (2005). "On Change Diagnosis in Evolving Data Streams", IEEE TKDE, 17(5):587-600, May 2005.
- C. Aggarwal, J. Han, J.Wang, and P. Yu (2003). "A framework for clustering evolving data streams", Proc. of Int. Conf. on Very Large Data Bases (VLDB'03),Javed Aslam, Katya Pelekhov, and Daniela Rus (1999). "A practical clustering algorithm for static and dynamic information organization" Proc. of SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms), pages 51--60, January 1999.
- Moses Charikar, Chandra Chekuri, Tomas Feder, and Rajeev Motwani (1997). "Incremental Clustering and Dynamic Information Retrieval", Proc. of 29th Annual ACM Symposium on the Theory of Computing, pages 626--635, El Paso, Texas, USA, May 1997. ACM Press.
- Martin Ester, Hans-Peter Kriegel, Joerg Sander, Michael Wimmer, and Xiaowei Xu (1998). "Incremental Clustering for Mining in a Data Warehousing Environment", . Proc. of VLDB'98, pages 323--333, New York City, New York, USA, August 1998. Morgan Kaufmann.
- S. Guha, N. Mishra, R. Motwani, L. O' Callaghan (2000). "Clustering Data Streams", Proc. of ICJNN'2000.


References and Readings (Block 4)

- F.Höppner and M.Böttcher (2007). "Matching partitions over time to reliably capture local clusters in noisy domains", Principles and Practice of Knowledge Discovery in Databases (PKDD'07), pages 479--486, Warsaw, Poland, 2007. Springer.
- P. Kalnis, N. Mamoulis, and S.Bakiras (2005). "On Discovering Moving Clusters in Spatio-temporal Data", Proc. of 9th Int. Symposium on Advances in Spatial and Temporal Databases (SSTD'2005)}, LNCS 3633, pages 364--381, Angra dos Reis, Brazil, Aug. 2005. Springer.
- Olfa Nasraoui, Cesar Cardona-Uribe and Carlos Rojas-Coronel (2003). "Tecno-Streams: Tracking Evolving Clusters in Noisy Data Streams with an Scalable Immune System Learning Method", in Proc. of ICDM'03, Melbourne, Australia.
- Daniel Neill, Andrew Moore, Maheshkumar Sabhnani, Kenny Daniel (2005). "Detection of Emerging Space-Time Clusters", in Proc. of KDD'05, pages 218 - 227, Chicago, IL, Aug. 2005.
- Myra Spiliopoulou, Irene Ntoutsi, Yannis Theodoridis, Rene Schult (2006). "MONIC – Modeling and Monitoring Cluster Transitions", Proc. of KDD'06, pages 706 – 711, Philadelphia, Aug. 2006.
- Hui Yang, Srinivasan Parthasarathy, Sameep Mehta (2005). "A Generalized Framework for Mining Spatio-Temporal Patterns in Scientific Data", in Proc. of KDD'05, pages 716 – 721, Chicago, IL, Aug. 2005.


Some Readings on Topic Evolution in Text Streams

- A. Banerjee and S. Basu (2007). "Topic Models over Text Streams: A Study on Batch and Online Unsupervised Learning", Proc. of SIAM Data Mining Conf., 2007
- D. Blei and J. Lafferty (2006). "Dynamic Topic Models", Proc. of Int. Conf. on Machine Learning (ICML'06).
- Q. Mei and C. Zhai (2005). "Discovering Evolutionary Theme Patterns from Text – An Exploration of Temporal Text Mining", Proc. of 11th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'05), pages 198--207, Chicago, IL, Aug. 2005. ACM Press.
- S. Morinanga and K. Yamanishi (2004). "Tracking dynamics of topic trends using a finite mixture model", Proc. of 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'04), pages 811-816, Aug. 2004. ACM Press.
- S. Schulz, M. Spiliopoulou and R. Schult (2007). "Topic and cluster evolution over noisy document streams", In (F. Masseglia, P.Poncelet and M.Teisseire, eds), *Data Mining Patterns: New Methods and Applications*. Idea Group, 2007.
- X. Wang and M. McCallum (2006). "Topics over Time: A Non-Markov Continuous-Time Model of Topical Trends", Proc. of 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'06), pages 424-433, Aug. 2006. ACM Press.


End of Block 4

Thank you!

Questions?


Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
- Block 5: Change Mining
  - Notion of Change Mining
  - Main Steps in Change Mining
  - Weakest Points
- Block 6: Conclusions and Outlook


Change Mining – Motivation

- Recent work in stream mining is well aware that most data generating processes lack stationarity
- Same is also true for **any (large) dataset** that was collected **over a larger period of time** (not necessarily stream data)

Change Mining – Get more out of the data

- Time-stamps are almost always available
  → "artificial" stream of data (not necessarily at a high rate)
- Generate a sequence of models rather than just one
  → usually more efficient (if complexity above O(n))
- Not only the new adapted model is of value, but what interests people most is: *What has changed and how?*
  → high potential for businesses (customers, market)
- Having accepted the "model of last month", we have an effective means to filter new knowledge
  → only new things are interesting


Change Mining

- Our notion of Change Mining

  Change Mining is a data mining paradigm for the study of *time-associated data*. Its objective is the discovery, modelling, monitoring, prediction and interpretation of changes in the models that describe an evolving population.

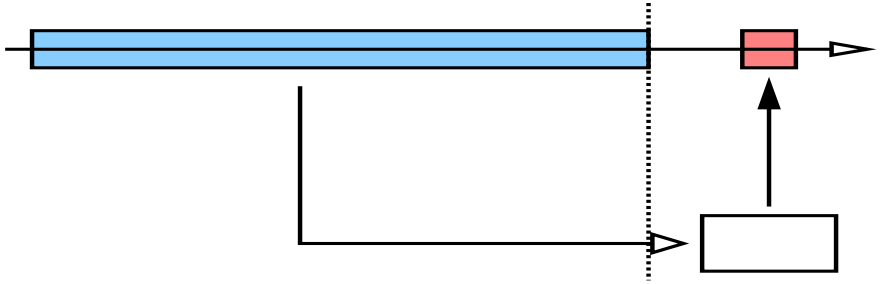
- Subdomain of high-order mining


Involved Tasks

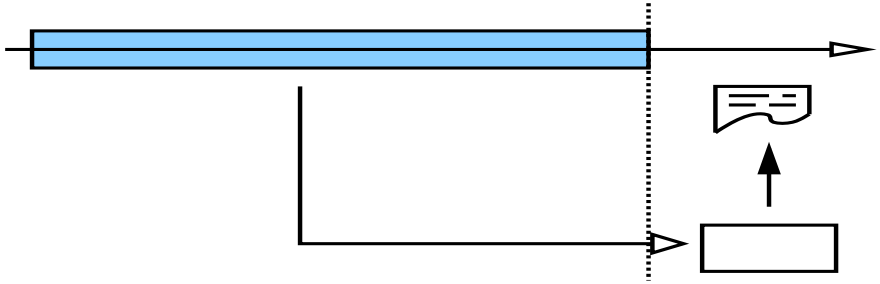
- Determine Goals of Change Mining
- Specify Model of Time
- Specify Objects of Change
- Design Monitoring Mechanism


(1) Goal of Change Mining

- Change (Detection and) Prediction
  e.g. precise temporal location of (past and) future changes



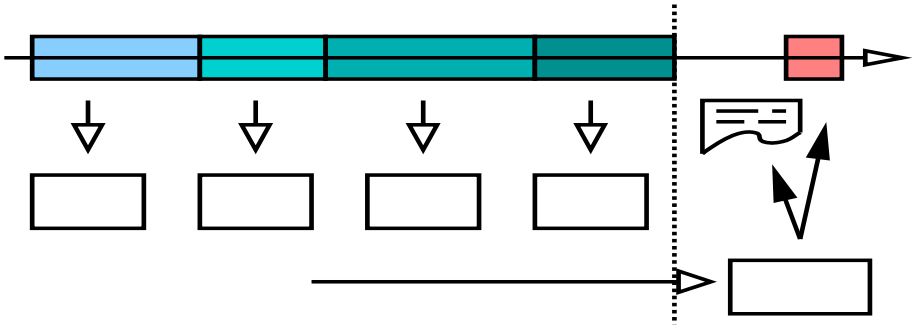
- Change (Detection and) Description
  e.g. textual representation, summary, change history

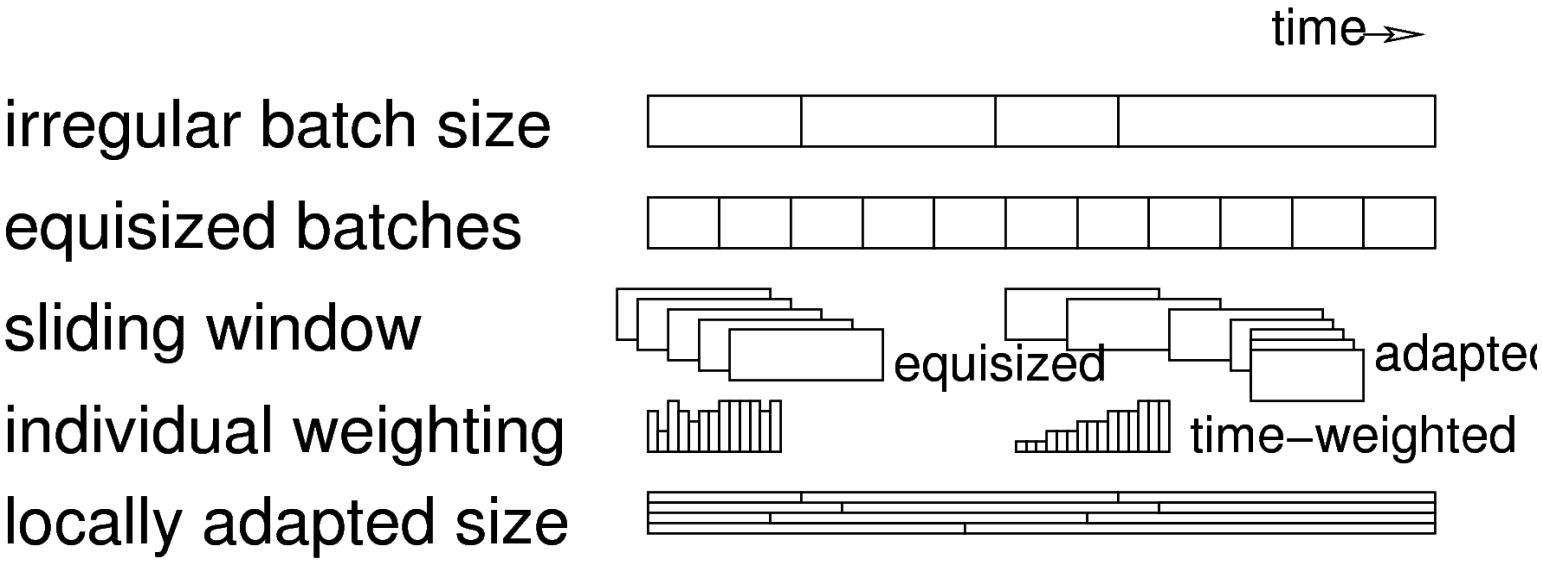


(1) Goal of Change Mining

- Change (Detection and) *Prediction*
  - precise temporal localization of change in past and future
  - prom. techn.: MDL (Chakrabarti et al), HMM (Mei et al)
- Change (Detection and) *Description*
  - Prefer interpretable models
  - exact localization not crucial


(2) Partitioning of time





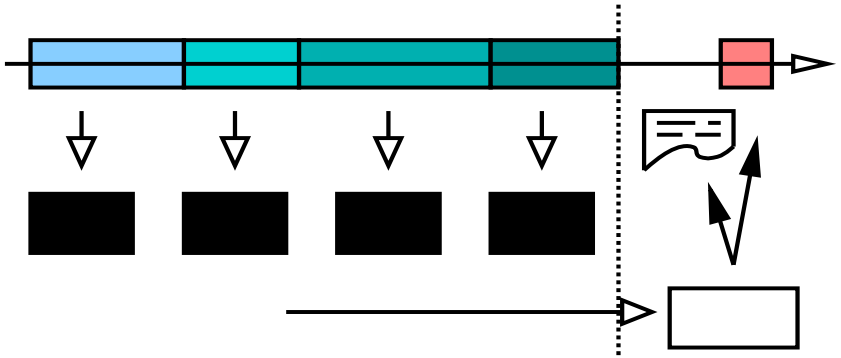
(2) Granularity

- Coarse-grained
  - slow adaptation to change
  - poor localization
  - May be sufficient for description

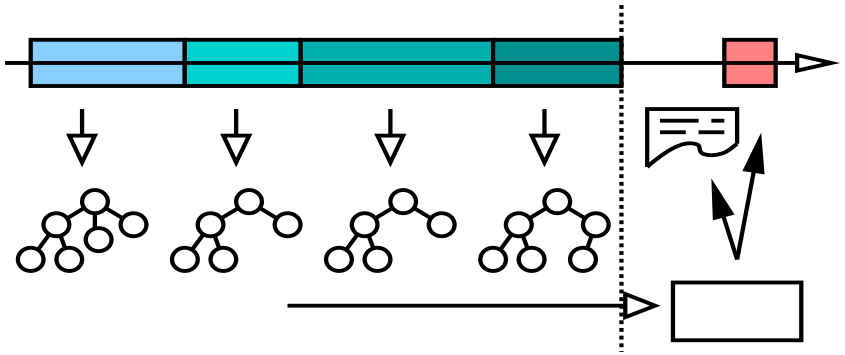
- Fine-grained
  - noise/drift distinction more difficult
  - computationally more expensive

(3) Objects of Change

- Monolithic Approach



- Compositional Approach

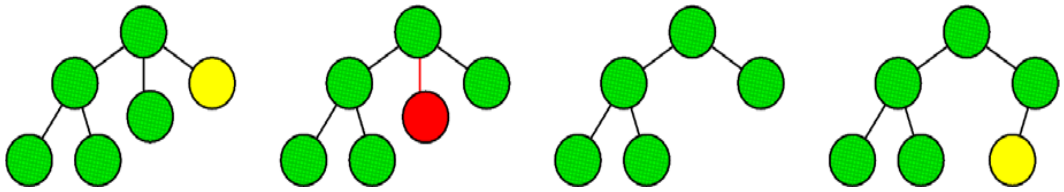


(3) Objects of Change : Monolithic View

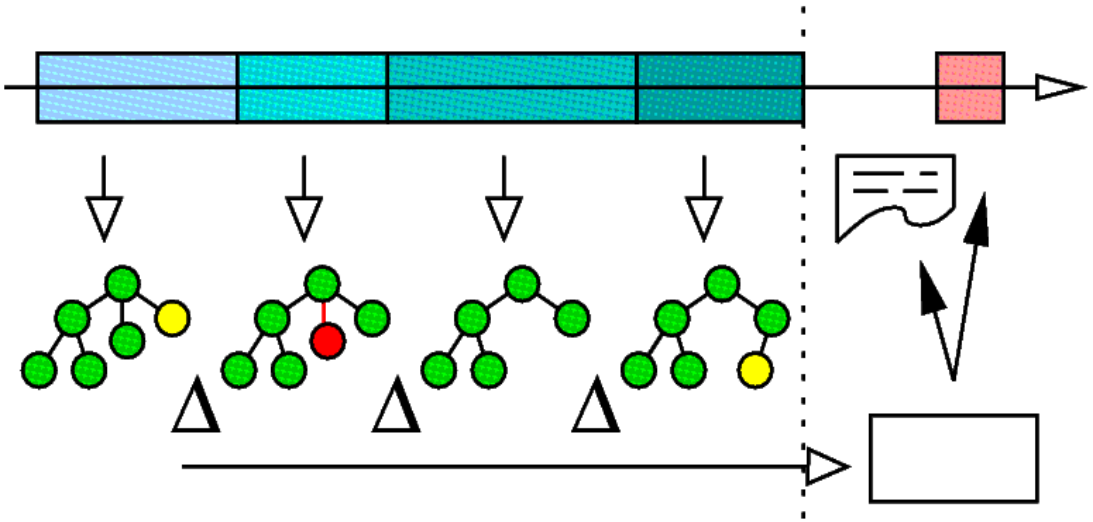
- Monolithic Approach: Change of model “as a whole”
  - e.g. “set of association rules”, “set of clusters”, “classifier”
  - independent of particular, say, classifier
    (possibly multiple types of classifiers simultaneously)
  - unsuited for *change description*


(3) Objects of Change : Compositional View

- Compositional Approach: Change of model components
  - analyse corresponding components of different models
  - requires adaptation to specific kind of model
  - Enables better (spatial) localization of change

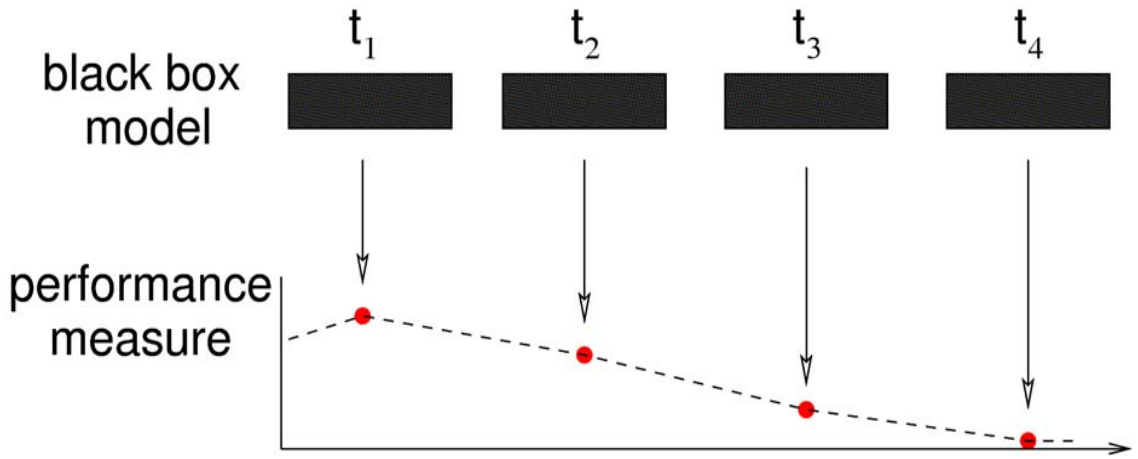


(4) Design Monitoring Mechanism



(4) Design Monitoring Mechanism : Monolithic View

- Monolithic Approach: Change of model “as a whole”
  - assess models via performance measure (e.g. accuracy)
  - analyse measurements instead of models
  - subsequent measurements lead to (short) time series
  - trend, outlier, breakpoint detection in (short) time series

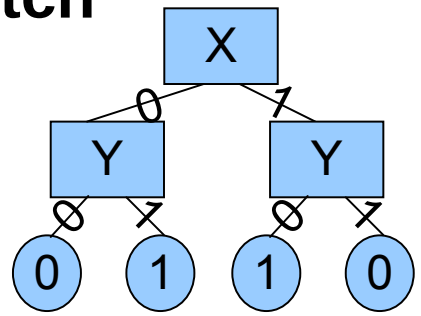


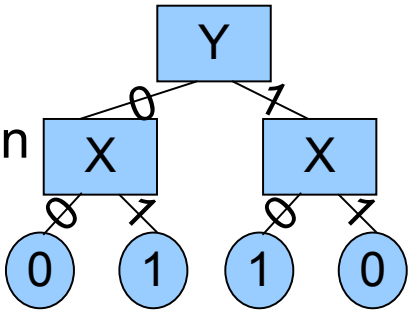
(4) Design Monitoring Mechanism : Composite View

- Compositional Approach: Change of individual parts
  - *match* components of subsequent models (correspondence)
    - results of matching useful for qualitative description: *“stable component”*, *“emerging component”*, *“vanishing component”*
  - *measure/summarize* changes of corresponding components
    - individual components
    - model as a whole (aggregation)
    - Results useful for quantitative description: *“moving component”*, *“shrinking component”*, *“distribution change”*...

(4) Monitoring: Matching Components I

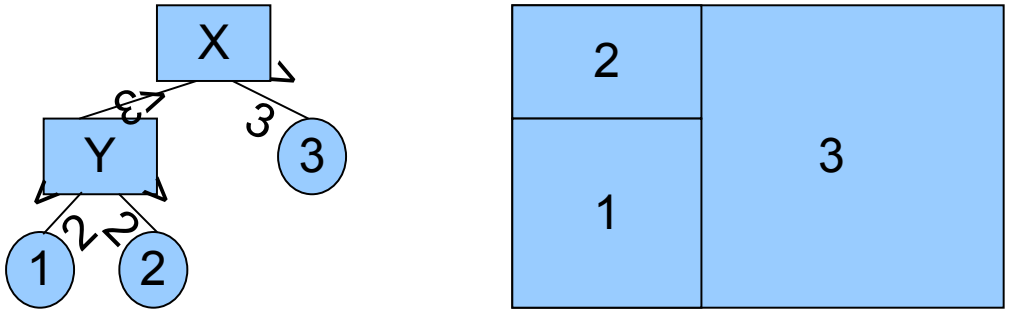
- **Symbolic Representation – Syntactical Match**
  - canonical symbolic representation
    - association rule: A,B,C → E
    - decision tree: (X, (Y, 0,1), (Y, 1,0))
    - cluster: { {#4,#7,#3}, {#1, #2, #5, #6} }
  - Two components match if they are syntactically equivalent
    - Purely syntactic match : only binary result, components have to *match exactly*
    - Beware: same semantics, different representation





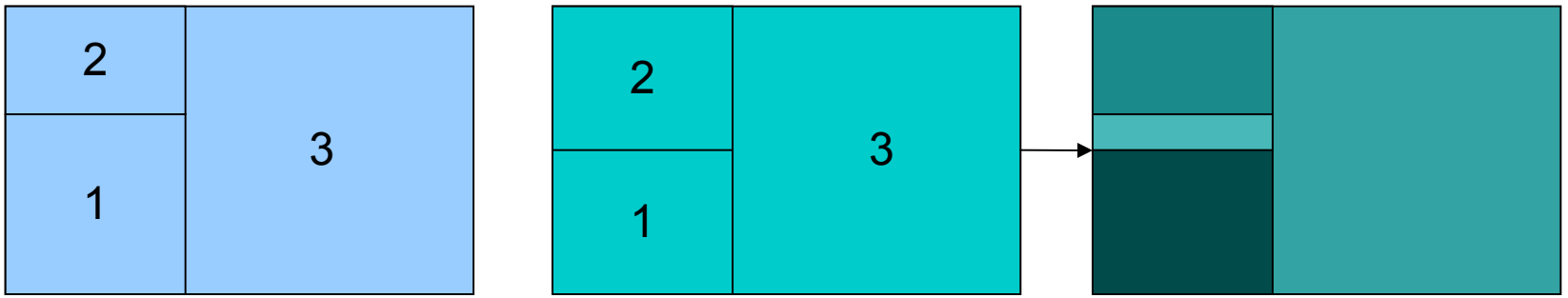
(4) Monitoring: Matching Components II

- **Aligned Decomposition and Difference-Function**
  - model components refer to certain regions in data space



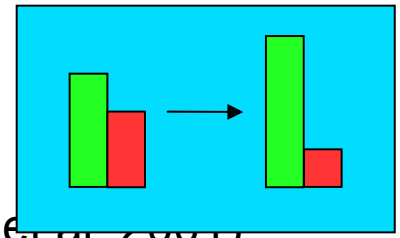
(4) Monitoring: Matching Components II

- **Aligned Decomposition and Difference-Function**
  - model components refer to certain regions in data space
  - construct or align data space regions
    - e.g. tesselation of data space
    - restrict models to regions



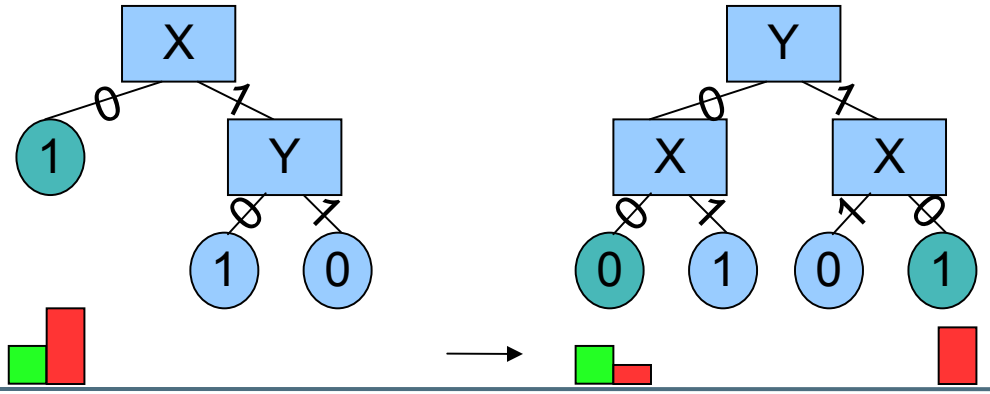
(4) Monitoring: Matching Components II

- **Aligned Decomposition and Difference-Function**
  - model components refer to certain regions in data space
  - construct or align data space regions
    - e.g. tesselation of data space
    - restrict models to regions
  - Similarity function measures model difference within regions
    - optional: aggregation of difference
  - see FOCUS (Ganti et al, 1999) and DEMON (Ganti et al, 2001)



(4) Monitoring: Matching Components III

- **Direct Match via Data Trace**
  - Compare existing components directly
    - feed chunk of data through both models
    - observe at which components data arrives
      - All data from C_i/M_1 in some component C_j/M_2 → stable
      - All data from C_i/M_1 to C_j/M_2 and C_k/M_2 → rule split/refinement



If X=0, then C=1

If Y=0 & X=0, then C=0
If Y=1 & X=0, then C=1

Correspondence Tracing (Wang et al, 2003)

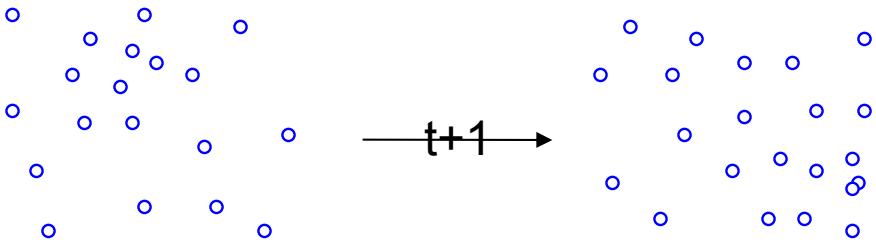
- Focus on interpretability: What has changed?
- Feed data into old and new tree, compare corresp. paths:
  - Qualitative change: rules have changed syntactically
    old: Status=0EUR & Duration>11 & Foreign=yes → no credit
    new: Status=0EUR & Duration>11 & Foreign=yes & ResTime>1 → credit
    new: Status=0EUR & Residence-Time<=1 → no credit
  - Quantitative change: leaf distribution has changed

Weakest points

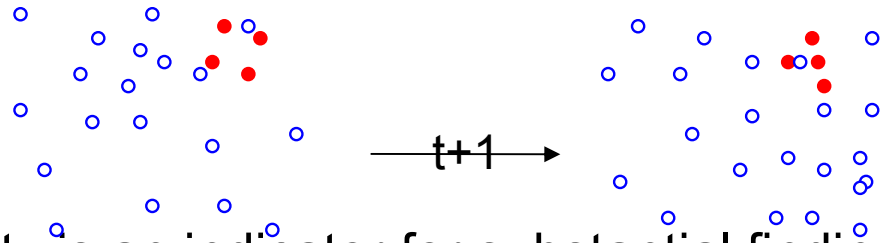
- Model Instability (Model Change ≠ Data Change)
  - Reporting Artefacts rather than true change
    - ID3, C4.5, ... are instable if attributes have similar IG
    - Nnet, k-Means, ... depend strongly on initialization
    - SingleLinkageClustering, ... one record may change a lot
- Reliable Distinction Noise/Change
  - confidence in reported change?
    - introduce certain slack in reporting changes
    - observe change over time : noise will not reproduce itself


PAMALOC (Höppner&Böttcher, 2007)

- Discriminative Power of Time
  - Which clusters are “correct”? (wrt artefacts & noise)
    - random fluctuations in density



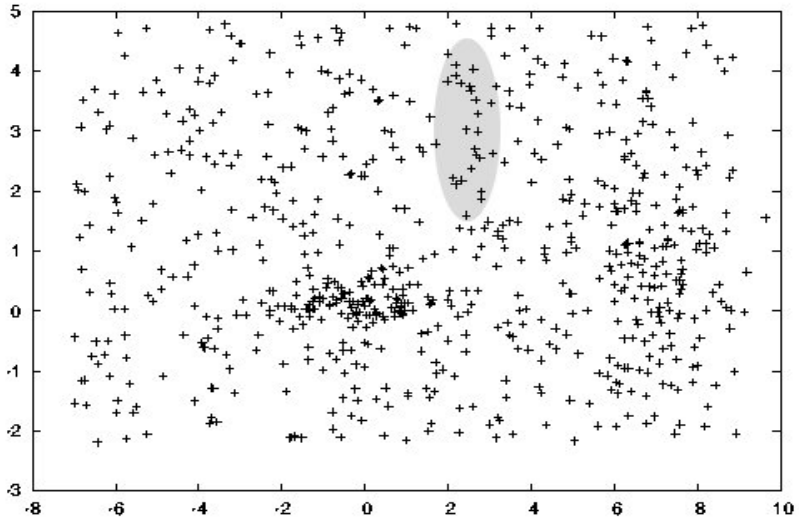
    - only “true structure” will repeat itself

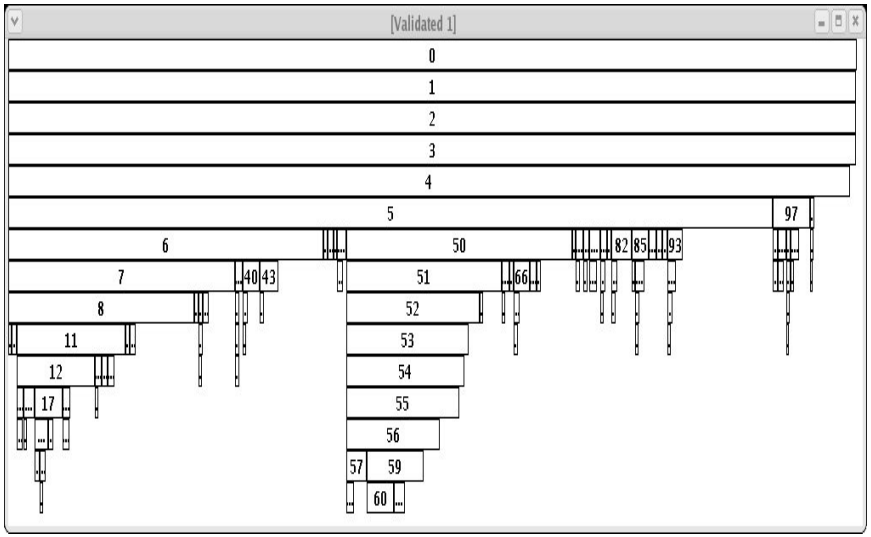


    - stability is an indicator for substantial findings

PAMALOC (Höppner&Böttcher, 2007)

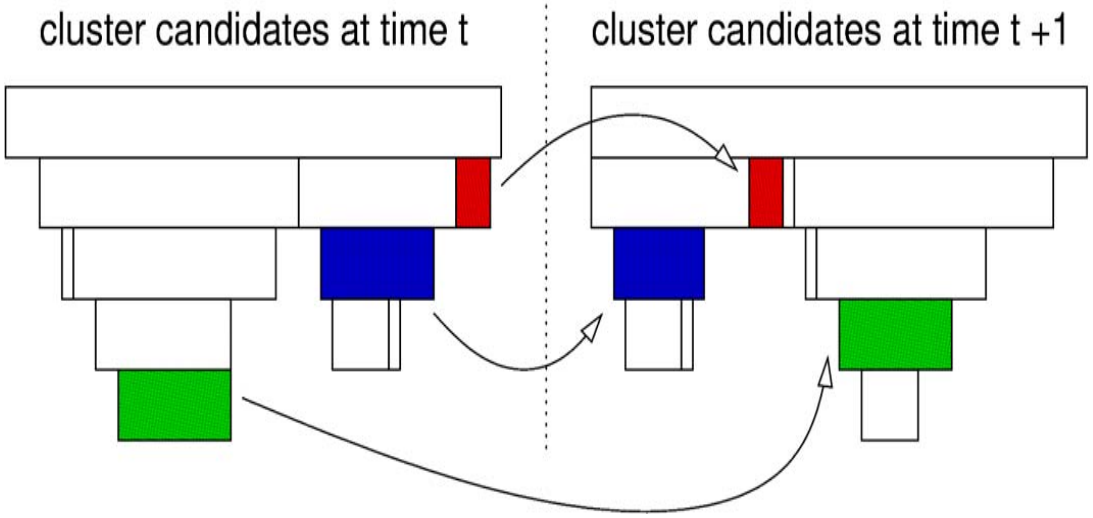
- How can we know at time t, which submodel will survive?
- Only time will reveal, which findings are substantial.
  - Defer decision about correctness
  - Maintain cluster models at multiple scales simultaneously





PAMALOC (Höppner&Böttcher, 2007)

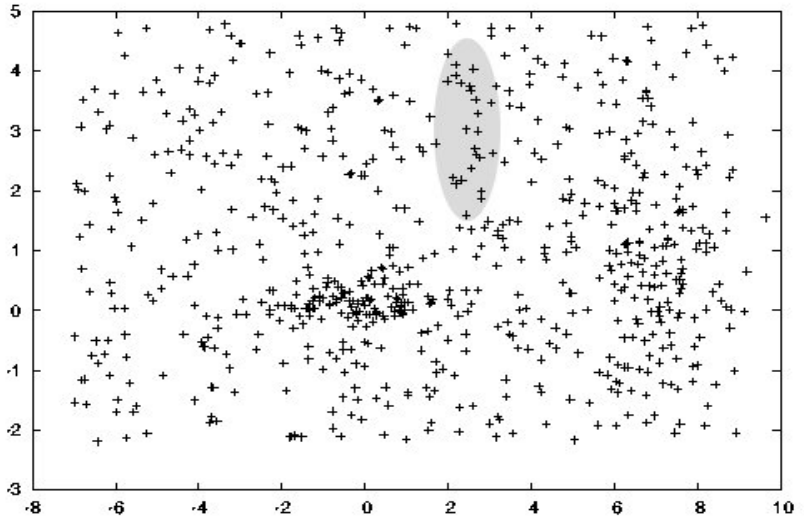
- How can we know at time t, which submodel will survive?
- Only time will reveal, which findings are substantial.
  - Defer decision about correctness
  - Maintain cluster models at multiple scales simultaneously
  - Match all cluster candidates at time t to candidates at t+1

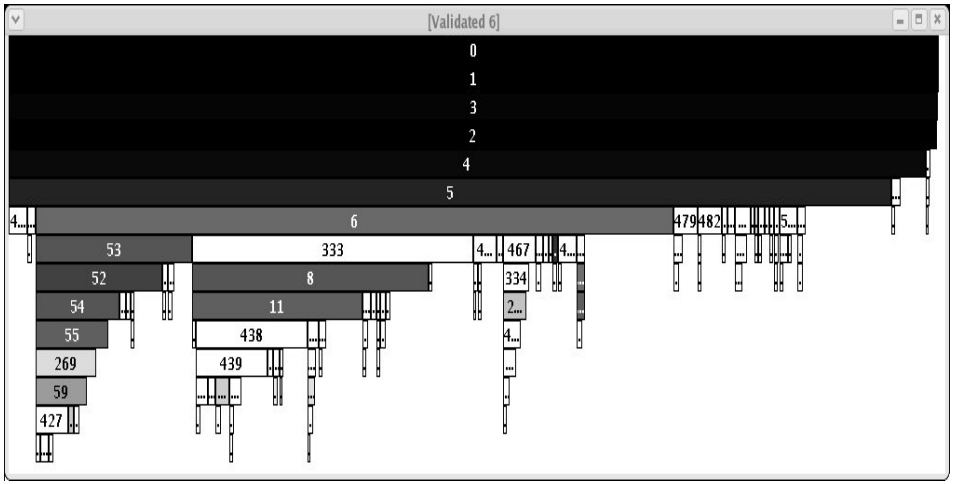


	1 2 3 4 5
1	cluster assign-
2	ment cost
3	matrix
4	
5	

PAMALOC (Höppner&Böttcher, 2007)

- How can we know at time t, which structure will survive?
- Only time will reveal, which findings are substantial.
  - Defer decision about correctness
  - Maintain cluster models at multiple scales simultaneously
  - Match all cluster candidates at time t to candidates at t+1
  - Report only candidates that survive





Change Mining – Relevant & Related Areas

- Model Maintenance (Liu & Tuzhilin, 2008)
  - Store all models learned for later access (Model-DWH)
  - Easy access (model query languages)
- Incremental Mining
  - Fast update of models (DT, NBC, DBScan, ...)
- Novelty Detection
  - Quickly detect irregular (“surprising”) data (change indicator)
- Comparison of Populations
  - Also an issue if time is not involved:
    e.g. compare customers from different countries


References (Block 5)

- Catania, Maddalena, Mazza. PSYCHO: A Prototype System for Pattern Management. Proc. Int. Conf. on Very Large Data Bases, 2005, 1346-1349
- Ganti, Gehrke, Ramakrishnan. A framework for measuring changes in data characteristics Proc. 18th Symp. on Principles of Database Systems, 1999, 126-137
- Ganti, Gehrke, Ramakrishnan. DEMON: Mining and Monitoring Evolving Data. IEEE Trans. Knowl. Data Eng., 2001, 13, 50-63
- Höppner, Böttcher: Matching Partitions over Time to Reliably Capture Local Clusters in Noisy Domains. Proc. Int. Conf. Principles and Practice of Knowledge Discovery in Databases PKDD (2007), 479-486.
- Liu, Tuzhilin. Managing Large Collections of Data Mining Models. Communications of the ACM, 2008, 51, 85-89
- Mei, Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. KDD 2005, 198-207
- Wang, Zhou, Fu, Yu. Mining Changes of Classification by Correspondence Tracing. SIAM Int. Conf. on Data Mining, 2003, 95-106


End of Block 5

Thank you!

Questions?


Presentation Outline

- Block 1: Introduction
- Block 2: Evolution in Association Rules
- Block 3: Evolution in Classifiers
- Block 4: Evolution in Clusters
- Block 5: Change Mining
- Block 6: Conclusions and Outlook


The only thing constant is change

- Data evolves
  - …because the world changes
  - …because the results of data mining influence the underlying domain
- This is a new insight, in the sense that it has been consistently ignored for long time
  - Data is almost always collected over time
  - A time axis is present in almost every data warehouse


Two Challenges for Data Mining

- Quality and complexity challenge
  - Keep a high quality of learned models
  - Keep models up-to-date without relearning them
  - High volumes of data need efficient and time- and storage efficient algorithms
- Knowledge Challenge
  - Discover novel knowledge about how a domain evolves
  - Understand how things change
  - Monitor existing knowledge


Two Challenges for Data Mining

- Quality and complexity challenge
  - Stream Mining
  - Incremental Mining

➔ Relatively mature field

- Knowledge Challenge
  - Higher Order Mining
  - Change Mining

➔ Evolving field


Change Mining

- Could be a key to solve some of the most demanding problems of data mining
  - Interestingness and relevance assessment of patterns
  - Tracing of evolving patterns
  - Robustness in the presence of noise
- First research results by several authors are very promising
- **But:** There is still a lot of research necessary


Research is needed on

- Mechanisms for monitoring patterns
- Methods for studying the evolution of a model (e.g. classifier) as a whole and of its individual parts (e.g. classification rules)
- Methods for capturing, quantifying and predicting change
- Methods for distinguising between real change and artefacts (e.g. due to model instability)


Problems – Short Term

- Unavailability of suitable public benchmark data sets
  - Public timestamped datasets are not appropriate for benchmarking
    - Static concepts only -- as fixed class labels
    - No explicit information on when drift occurs
  - Meaningful artificial data sets are very difficult to generate
  - Existing concept drift simulators are not the optimal choice


Problems – Long Term

- Evolution in the data mining process:
  - Improvements of the quality of gathered data
  - Adjustments and tweaks of parameters
- Effects of this evolution on a model superimpose those of the underlying domain
- Change Mining may also discover the effects of efforts for a better model quality
- How can we separate the effect of efforts for a better model quality from true domain change?
- Can we use such information to analyse the data mining process itself?


Thank you!

Questions?