Knowledge Discovery in Databases II

Lecture 3 – Data Streams

Prof. Dr. Peer Kröger, Yifeng Lu

Sommer Semester 2019

Credits:

Based on material of Eirini Ntoutsi, Matthias Schubert,

Arthur Zimek, Peer Kröger, Yifeng Lu

Table of Contents

Kapitel 1: Intorduction i

A Disclaimer (kind of ...)

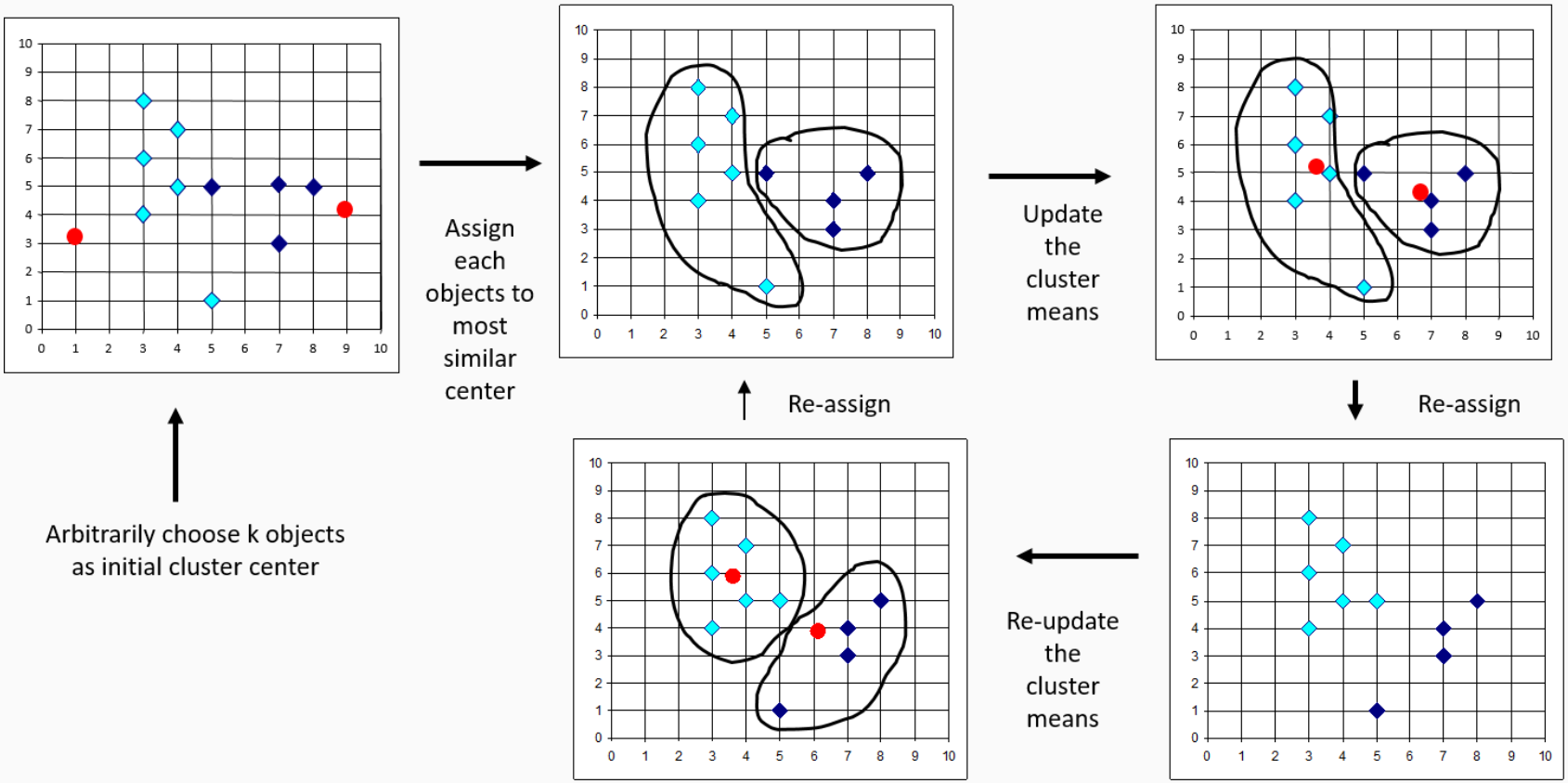
- Data streams usually are a very challenging source of data
- Analysis of data streams require to address several aspects such as
  - The hardware
  - The processing environment (like the operating system, the programming language and the programming schema, . . .)
  - The algorithmic design
  - . . .
- In this lecture, we focus on the algorithmic aspects that are necessary for processing data streams
- The lecture Big Data Management focuses on other aspects

Batch Learning

- Most of the DM algorithms focus on batch learning
  - The complete training/data set is available to the learning algorithm
  - Data instances can be accessed multiple times
  - e.g., for clustering: k-Means, DBSCAN
  - e.g., for classification: decision trees, Naïve Bayes
- Implict assumption: instances are generated by some stationary probability distribution; data is not volatile and so are patterns

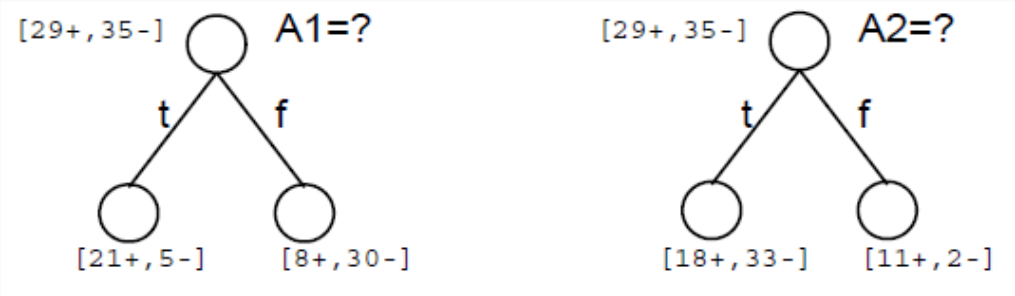
Example: Batch Clustering

- k-means (here $k = 2$) needs full access to the data in each iteration



Example: Batch Classification

- Decision Trees are constructed in a top-down recursive divide-and-conquer manner requiring full access to the data for each split
  - At start, all the training examples are at the root node
  - Select the best attribute for the root
  - For each possible value of the test attribute, a descendant of the root node is created and the instances are mapped to the appropriate descendant node
  - Repeat the splitting attribute decision for each descendant node, so instances are partitioned recursively

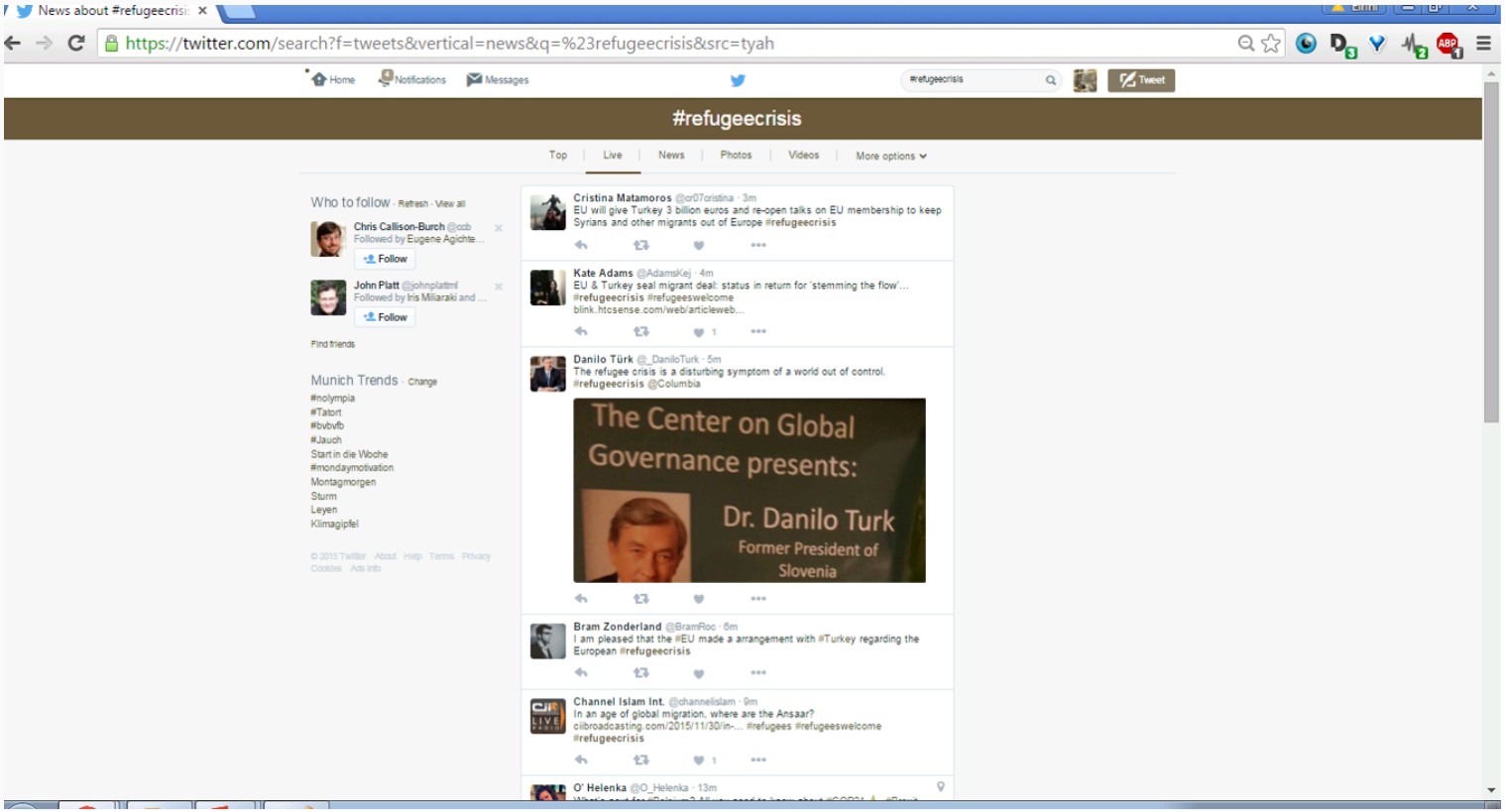


From Batch to Streams

- Many interesting applications nowadays come from dynamic environments where data are generated over time, e.g., customer transactions, call records, customer click data, social media interactions
- Batch learning is not sufficient anymore as
  - Data is never ending. What is the training set?
  - Multiple access to the data is not possible or desirable
- And also, the data generation process is subject to changes over time
  - The patterns extracted upon such sort of data are also evolving
  - Algorithms should respond to change (incorporate new data instances, forget obsolete data instances)

Examples

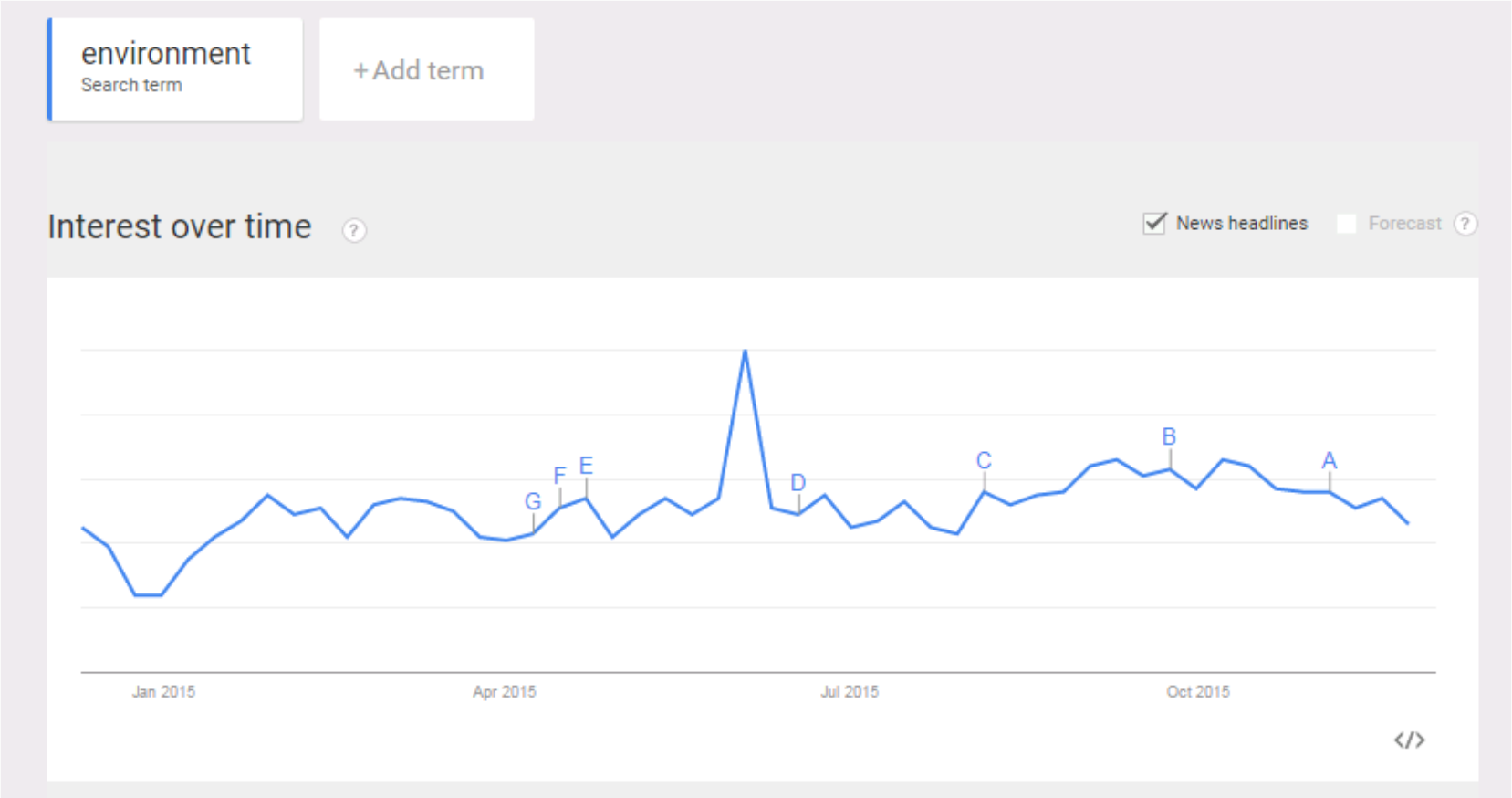
- Twitter stream for hastag “#refugeecrisis”



Source: `https://www.twitter.com/`

Examples

- Trend of the search for “environment”



Source: `https://www.google.com/trends/`

Examples

- Experiments at CERN are generating an entire petabyte (1PB=106 GB) of data every second as particles fired around the Large Hadron Collider (LHC) at velocities approaching the speed of light are smashed together
- “We do not store all the data as that would be impractical. Instead, from the collisions we run, we only keep the few pieces that are of interest, the rare events that occur, which our filters spot and send on over the network”
- This still means CERN is storing 25PB of data every year — the same as 1,000 years’ worth of DVD quality video — which can then be analyzed and interrogated by scientists looking for clues to the structure and make-up of the universe

Source: `http://public.web.cern.ch/public/en/LHC/Computing-en.html`

Source: `http://www.v3.co.uk/v3-uk/news/2081263/cern-experiments-generating-petabyte`

Examples

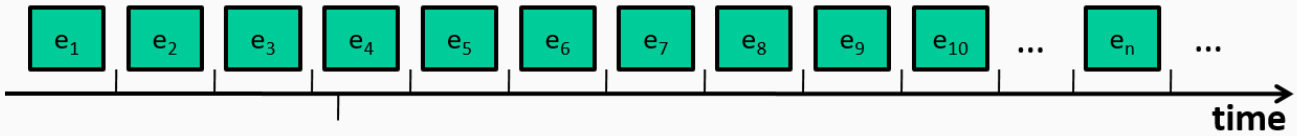
- Network monitoring records e.g. TCP connection records of LAN network traffic
- A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol
- Connections are described in terms of 42 features like duration, protocol type, service, flag, src bytes, dst bytes etc.
- Each connection is labeled as either normal, or as an attack, with exactly one specific attack type
- Most of the connections are usually normal, but occasionally there could be a burst of attacks at certain times

Source (with link to a real data set): `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`

What is a Data Stream?

Everything flows, nothing stands still

Heraclitus (535-475 BC)



- Data evolve over time as new data arrive (and old data become obsolete/irrelevant)
- We can distinguish between:
  - Dynamic data arriving at a low rate (as e.g. in DWs): incremental methods might work for such cases
  - Data streams: possible infinite sequence of elements arriving at a rapid rate: new methods are required to deal with the amount and complexity of these data

Incremental Methods

- Focus is on how to update the current pattern based on the newly arrived data, without re-computing the pattern from scratch
- Requires (limited) access to raw data (i.e., only the data that is affected by the changes)
- Example: incremental DBSCAN (insertion of a new point *p*)

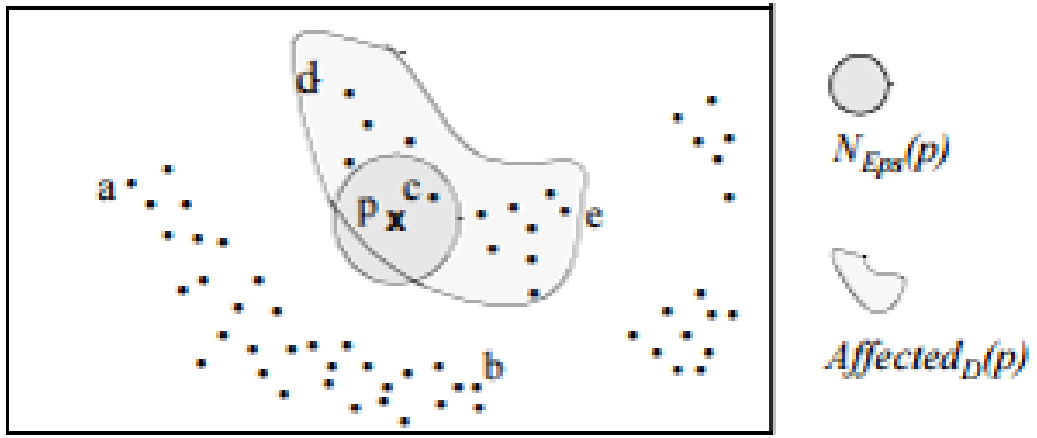


Figure 3: : Affected objects in a sample database

Challenges for Streams

- Data Mining over stream data is more challenging than batch learning
  - Huge amounts of data, thus, only a small amount can be stored in memory
  - Arrival at a rapid rate, thus, no much time for processing
  - The generative distribution of the stream might change over time rather than being stationary, thus, adapt and report on changes
- Requirements for stream mining algorithms
  - Use limited computational resources (bounded memory, small amount of available processing time)
  - No random access to the data but rather only one look at the data (upon their arrival)

From Data Changes to Pattern Changes

Example: cluster evolution over time

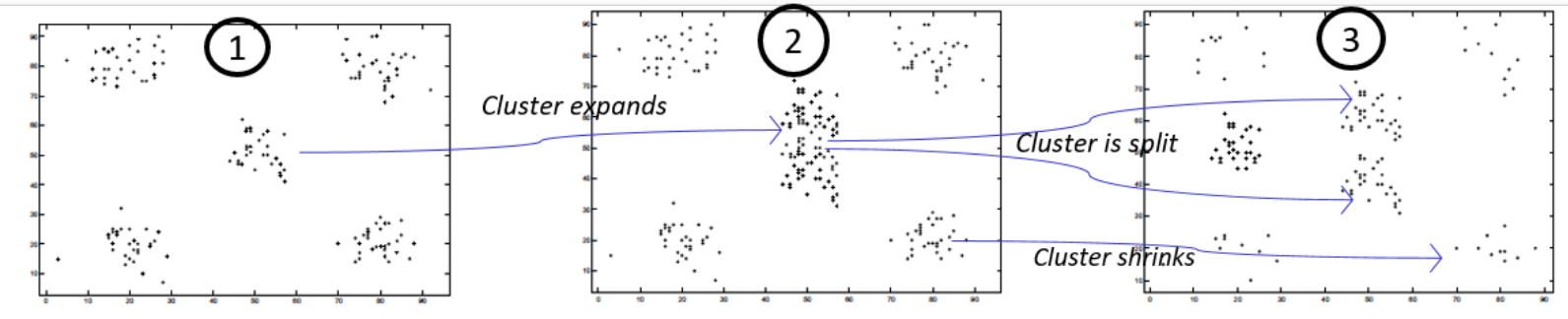


Figure: Data records at three consecutive time stamps, the clustering gradually changes
(from: *MONIC - Modeling and Monitoring Cluster Transitions*, Spiliopoulou et al, KDD 2006)

Example: decision boundary drift over time

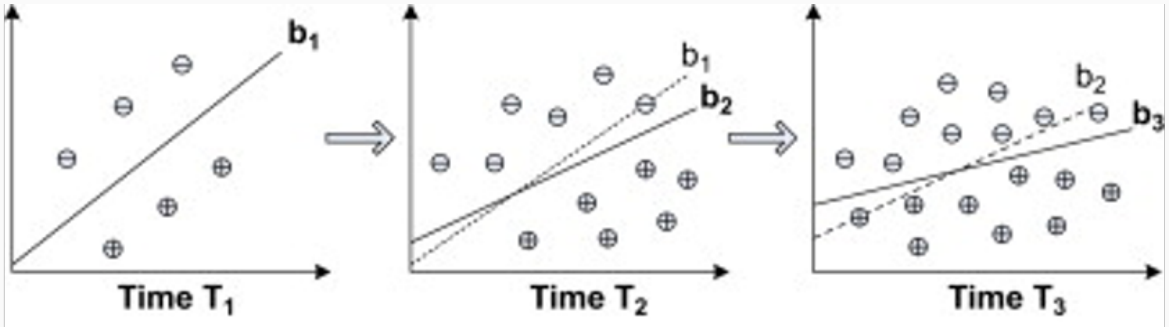


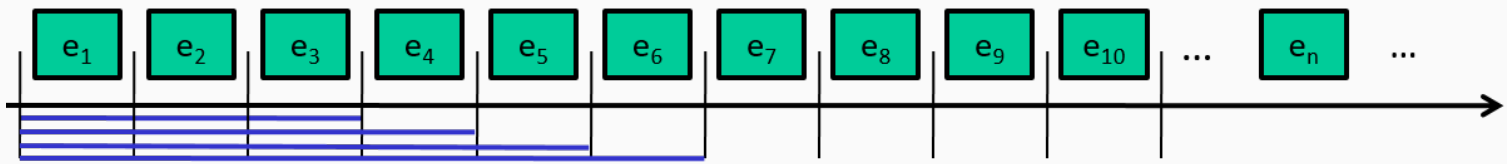
Fig. 1. An illustration of concept drifting in data streams. In the three consecutive time stamps T_1, T_2 and T_3, the classification boundary gradually drifts from b_1 to b_2 and finally to b_3.
(from: *A framework for application-driven classification of data streams*, Zhang et al, Journal Neurocomputing 2012)

Data Ageing

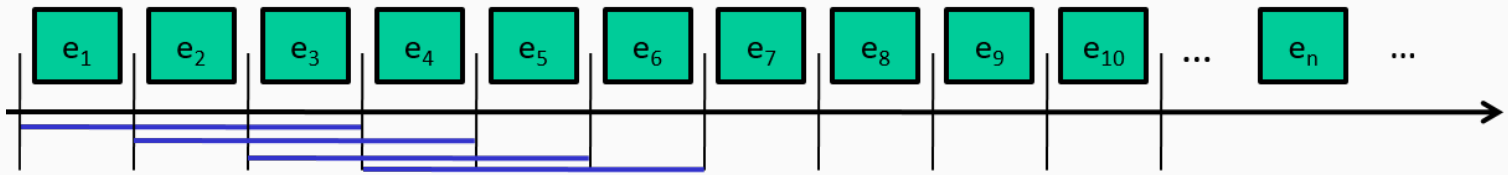
- Usually we are not interested in the whole history of the stream but only in the recent history
- There are different ageing/weighting mechanisms or window models that reflect which part of the stream history is important for learning
  - Landmark window model
  - Sliding window model
  - Damped window model

Data Ageing Models

- Landmark (window) model
  - Include all objects from a given landmark
  - All points have an equal weight (usually $w = 1$)

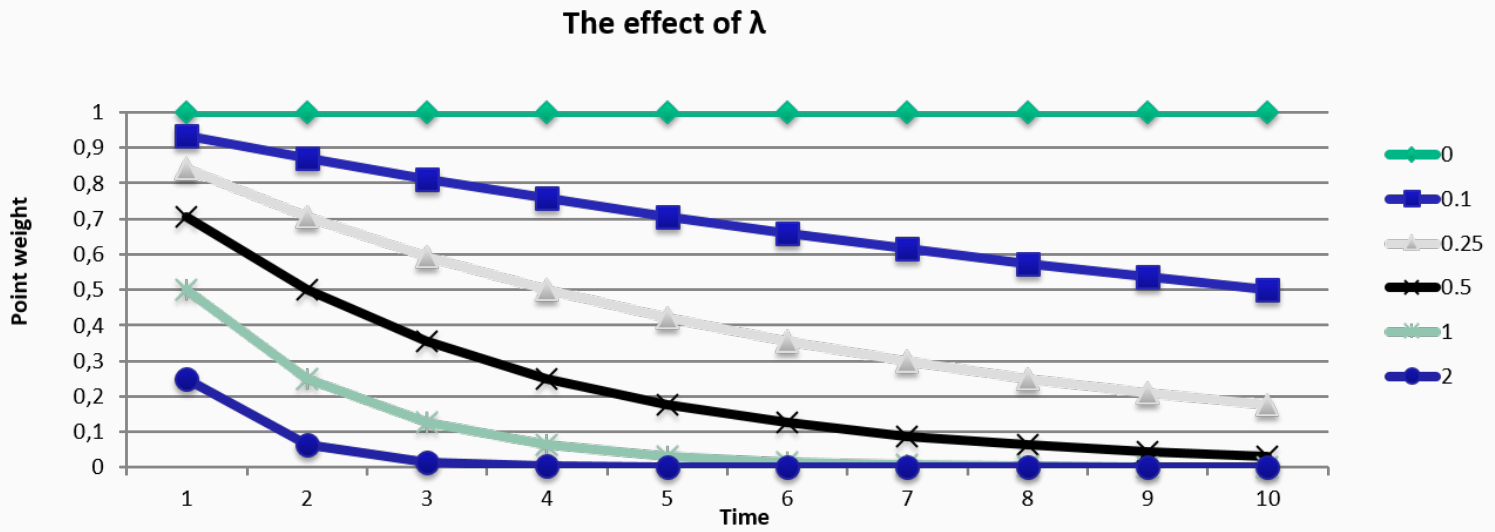


- Sliding window model
  - Remember only the n most recent entries, where n is the window size
  - All points within the window have a weight $w = 1$, for the rest: $w = 0$



Data Ageing Models

- Damped window model
  - Data are subject to ageing according to a fading function $f(t)$, i.e., each point is assigned a weight that decreases with time t via $f(t)$
  - A widely used fading function in temporal applications is the exponential fading function: $f(t) = 2^{-\lambda t}$, where $\lambda > 0$ is the deca rate that determines the importance of historical data (the higher the value of λ, the lower the importance of old data)



Kapitel 2: Classification i