Unsupervised Data Mining: From Batch to Stream Mining Algorithms

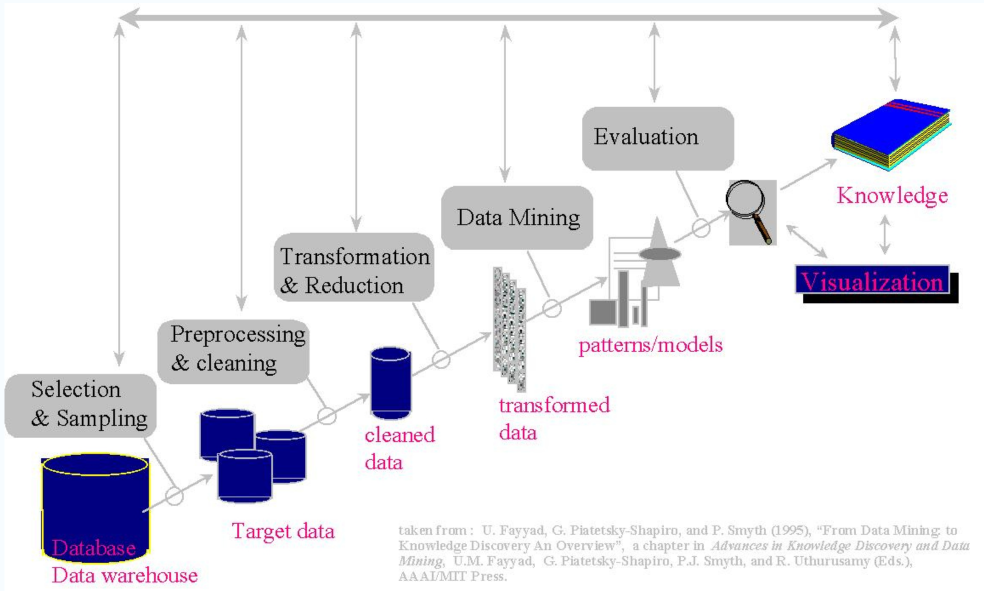
Prof. Dr. Stefan Kramer

Johannes Gutenberg-Universität Mainz

A Brief Introduction to Data Mining and KDD

Knowledge Discovery in Databases

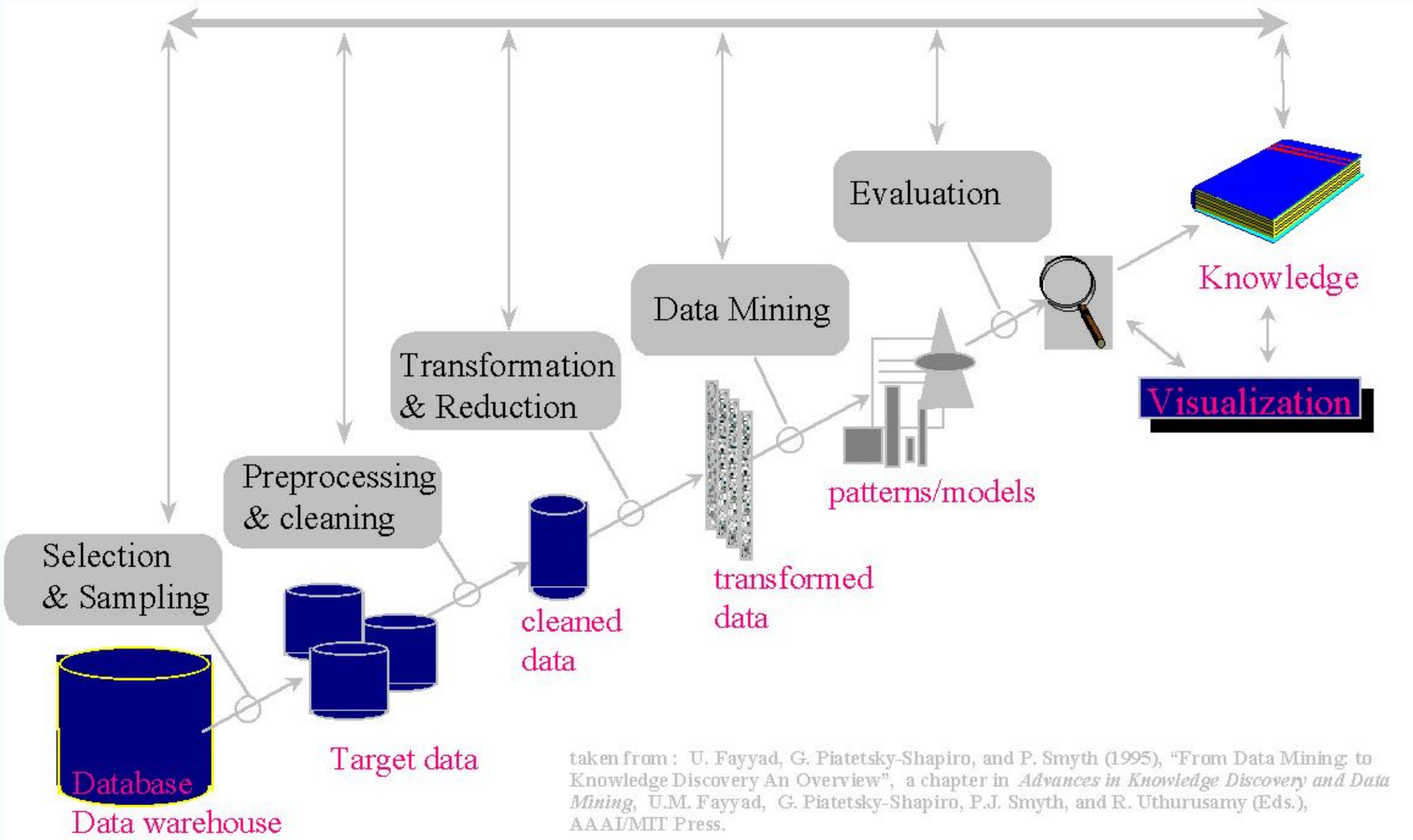
"... is the process of identifying valid, novel, potentially useful and ultimately understandable structure in data."



(Fayyad & Uthurusamy, 1996)

Structure = pattern or model

Knowledge Discovery in Databases and Data Mining



taken from: U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth (1995), "From Data Mining to Knowledge Discovery An Overview", a chapter in *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad, G. Piatetsky-Shapiro, P.J. Smyth, and R. Uthurusamy (Eds.), AAAI/MIT Press.

Data Mining

- Knowledge Discovery in Databases (KDD) (Fayyad 96): “KDD is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.“
- Data Mining: **data analysis step within the KDD process**

Machine Learning

- Learning = improving with experience at some task
  - Improve on task T
  - With respect to performance measure P
  - Based on experience E.
- Learn to play checkers:
  - T: Play checkers
  - P: % of games won
  - E: opportunity to play against oneself

Machine Learning

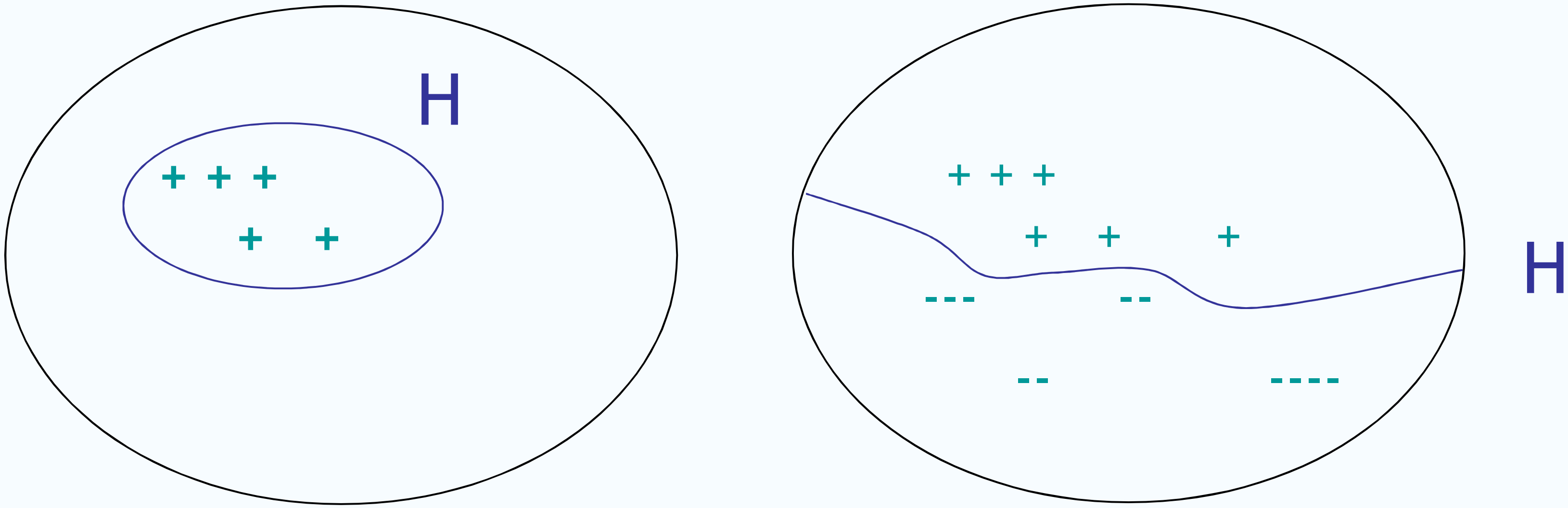
- Learning to classify examples (e.g., gene expression profiles into two subtypes):
  - T: Classifying examples
  - P: % of examples classified correctly
  - E: Training set of examples to learn from
- Machine learning algorithms (such as for classification) often used in Data Mining

Alternative Definitions…

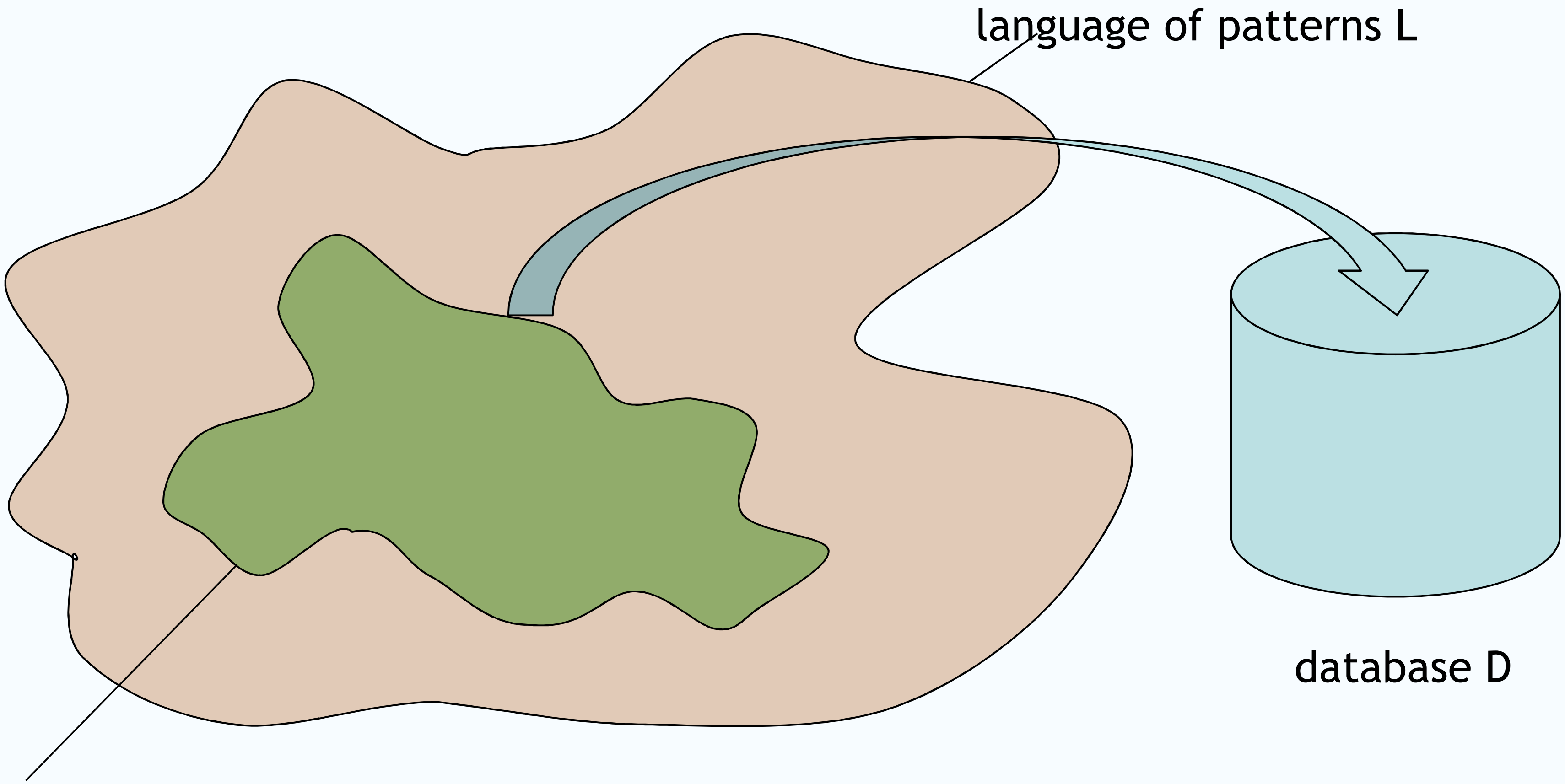
Heikki Mannila:

- *"Knowledge Discovery in Databases is finding the joint probability distribution"*
- "Data Mining is the technology of fast counting"

Descriptive Data Mining, Predictive Data Mining



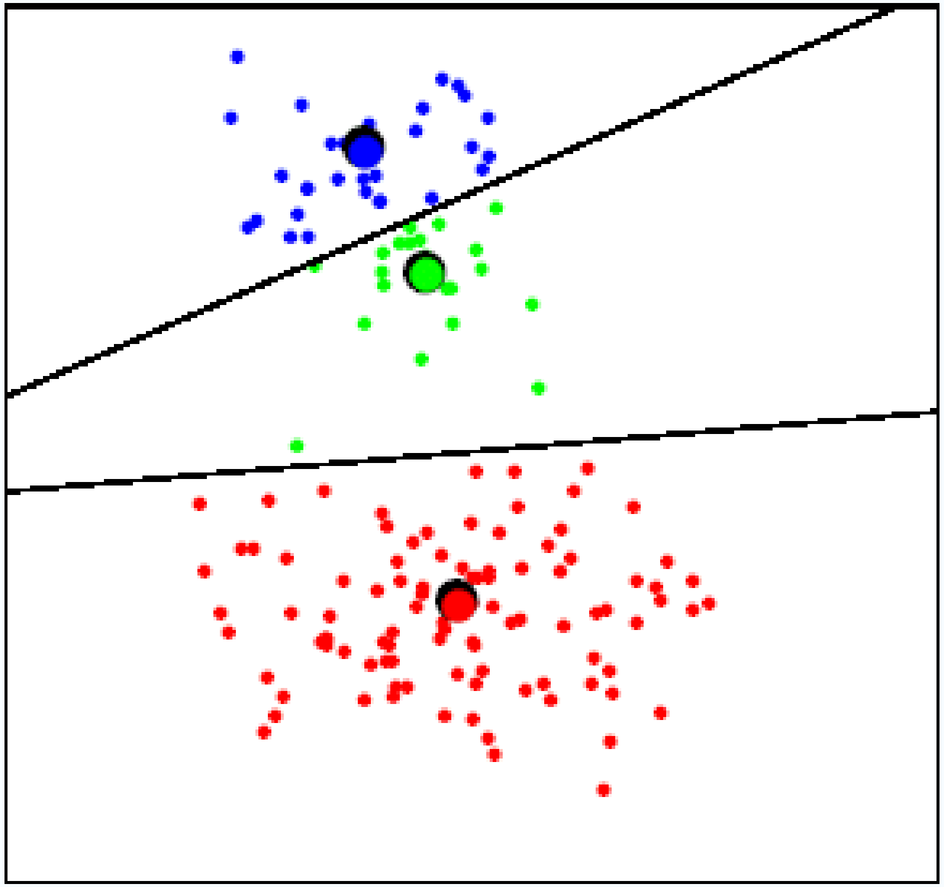
Pattern Mining



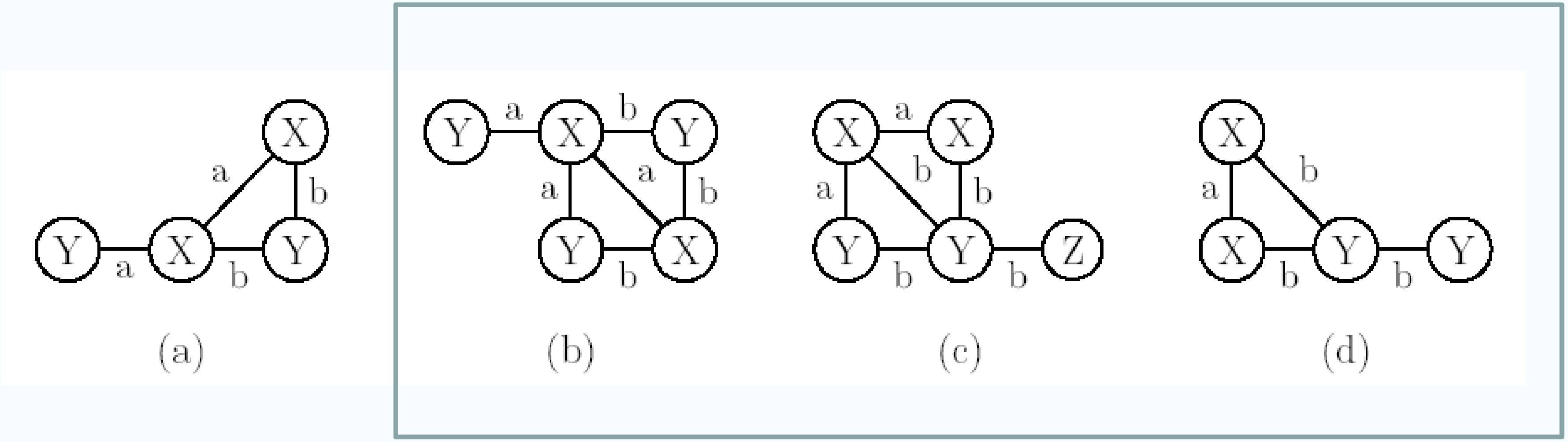
q(p, D) … interestingness predicate: a pattern p from L is interesting wrt. database D

what is interesting? *frequent*, *non-redundant*, *class correlated*, *structurally diverse*, …

Clustering

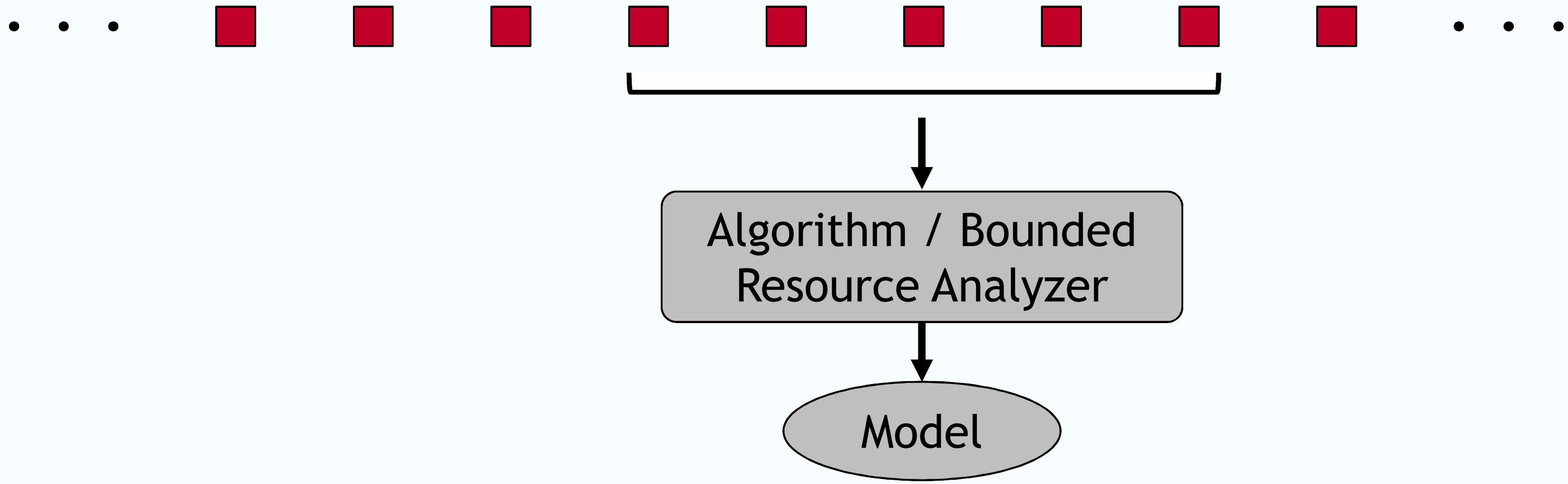


Graph Mining



- Graph database D (graphs (b) to (d))
- Find all subgraphs (patterns) that occur in at least two of the three graphs (examples)
- Example subgraph pattern p shown in (a)

Stream Mining

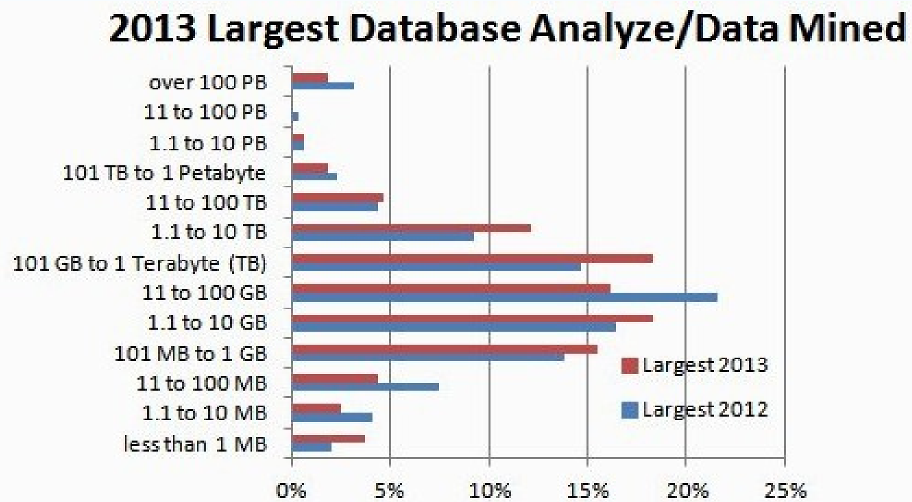


- *Data stream model and philosophy*
- *Not* one *model*
- *Approximations*

Introduction to Stream Mining

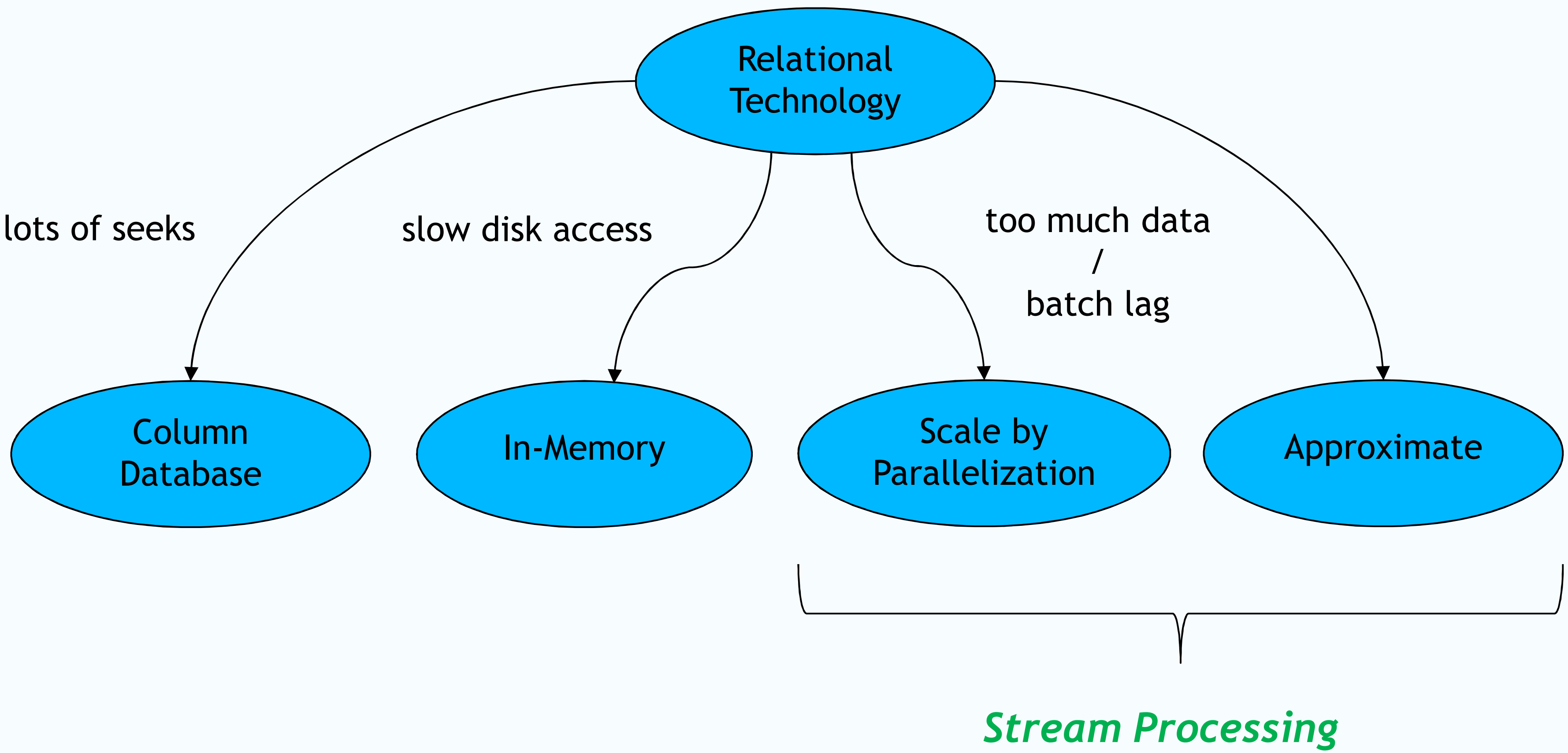
Big (Data) Vs

- Volume
  - depends also on preprocessing or on operations on them (e.g., pairwise comparisons)
  - how much volume really?

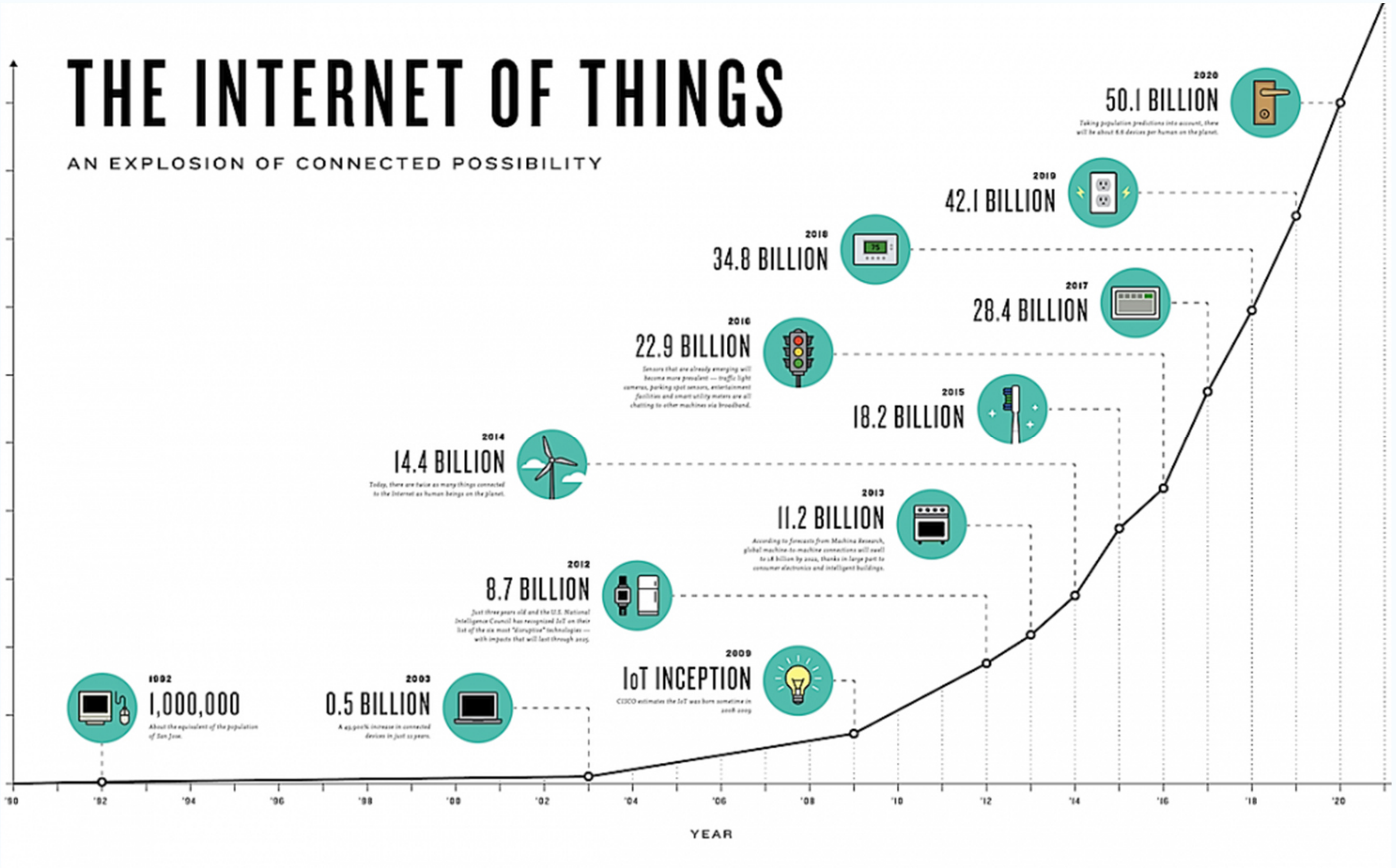


- Variety
  - often underestimated
- Velocity
  - analyzing data as they are generated (“one-touch”), real-time, anytime, ...
- Veracity
  - uncertainty in data, data quality, trust, but also prediction quality

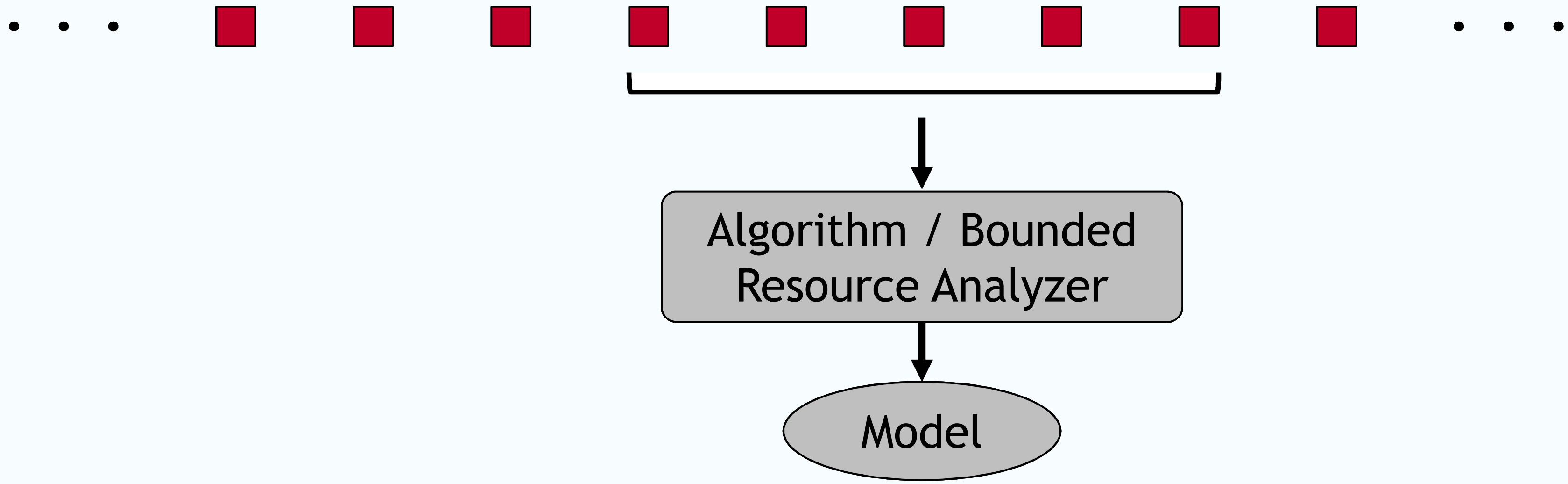
Options for Scaling Up / Out (Partly Inspired by Mikio Braun)



Expected Growth of Connected Devices



Stream Mining



- *Data stream model and philosophy*
- *Not* one *model*
- *Approximations*

Basic Stream Mining Algorithmics

Mean and Variance

Given a stream $x_1, x_2, \ldots, x_n$

$$\bar{x}_n = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$$

$$\sigma_n^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - \bar{x}_i)^2.$$

Mean and Variance

Given a stream $x_1, x_2, \ldots, x_n$

$$s_n = \sum_{i=1}^{n} x_i, \ q_n = \sum_{i=1}^{n} x_i^2$$

$$s_n = s_{n-1} + x_n, \ q_n = q_{n-1} + x_n^2$$

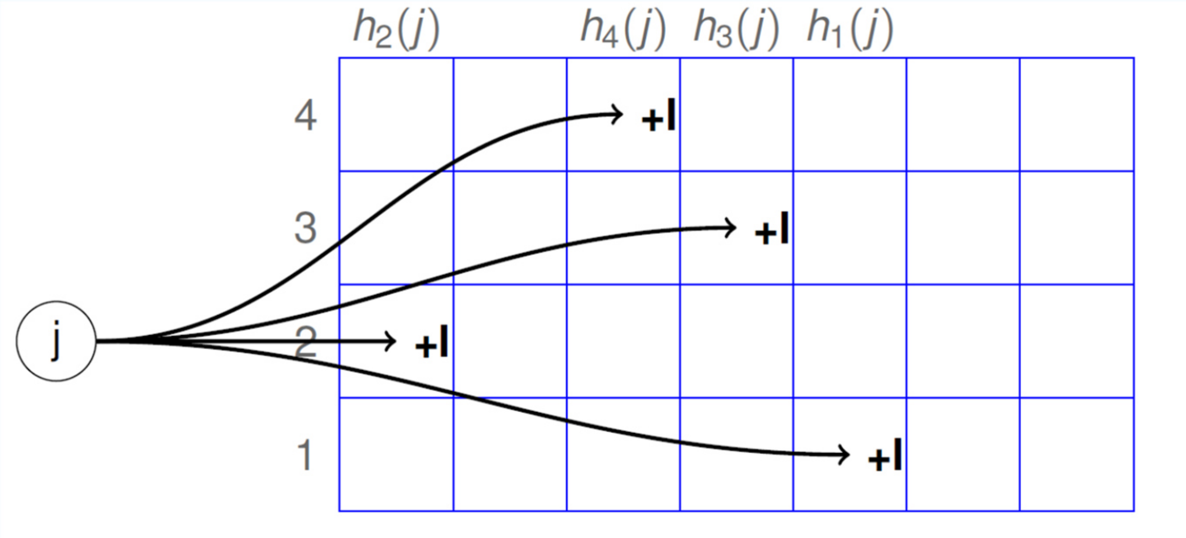
$$\bar{x}_n = s_n/n$$

$$\sigma_n^2 = \frac{1}{n-1} \cdot \left(\sum_{i=1}^{n} x_i^2 - n\bar{x}_i^2\right) = \frac{1}{n-1} \cdot (q_n - s_n^2/n)$$

Count Min Sketch

- Data structure for fast and memory-efficient counting on data streams
- Multiple hash tables and *pairwise independent hash functions* are used to update counts, effect of collisions is alleviated by taking the minimum of the results (i.e., the minimum of the counts from the table)
- Collisions still lead to overcounting, but this is upper-bounded, where the bound depends on the number d and the dimension w of the hash tables used.

CM Sketch Example Structure



- Width of table = dimension of hash tables = w = 7
- Depth of table = number of hash tables = d = 4
- d and w are *derived* from a bound (see below)
- Assume parameters are *set* as follows $\varepsilon = 0.4$, $\delta = 0.02$
- Bound tells us that the count resulting from CM sketch $\hat{a}_i \leq a_i + N^* \varepsilon$ in all but δ cases, with N being the length of the stream up to that point

Determining Size of Table and Determining Count from Table

- If you want overcounting only by maximally N*ε with a probability of 1- δ, then you dimension the table by:

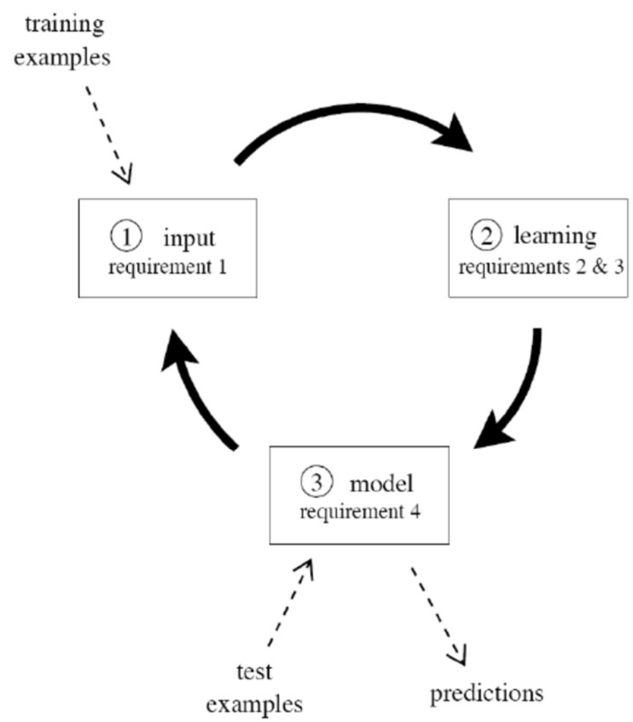
$$w = \left\lceil \frac{e}{\epsilon} \right\rceil, \qquad d = \left\lceil \ln \frac{1}{\delta} \right\rceil$$

- CM Sketch uses space w*d and update time d
- Counts are determined by:
  $\hat{a}_i = \min_j \text{count}[j, h_j(i)]$

Classification on Data Streams and Hoeffding Trees

Data Stream Classification Cycle

1. Process an example at a time, and inspect it only once (at most)
2. Use a limited amount of memory
3. Work in a limited amount of time
4. Be ready to predict at any point



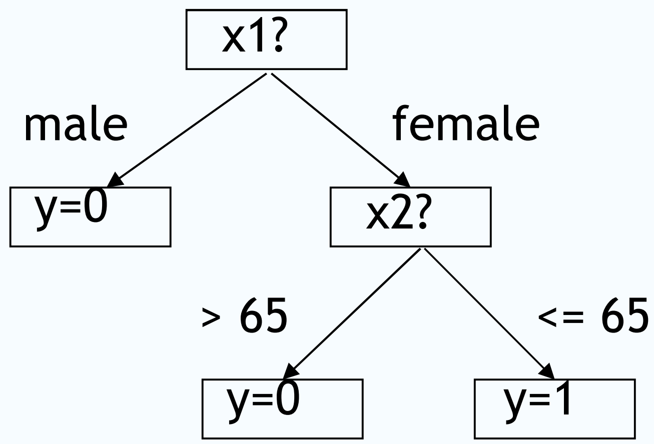
Prequential Testing

- First use new instance from stream to predict/test, then update the model based on it
- Approximates hold-out evaluation (testing on an „external“ test set)
- Estimate accuracy using sliding windows or fading factors

Hoeffding Tree Algorithm (Domingos & Hulten, KDD 2000)

Procedure HoeffdingTree(*Stream*, δ)
Let HT = Tree with single leaf (root)
Initialize sufficient statistics at root
For each example (X, Y) in *Stream*
    Sort (X, Y) to leaf using HT
    Update sufficient statistics at leaf

    Compute G for each attribute
    If G(best) – G(2nd best) > $\varepsilon = \sqrt{\frac{R^2 \ln\left(\frac{1}{\delta}\right)}{2n}}$
    then
      Split leaf on best attribute
      For each branch
          Start new leaf, init sufficient statistics
Return HT



Hoeffding Trees Preliminaries

- Suppose we have n observations of a real-valued random variable whose range is R (we assume R is 1 in the following) and mean is $\bar{X}$.
- The Hoeffding bound states that with probability $1-\delta$, the true mean of the variable is at least $\bar{X}-\varepsilon$, where

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

Hoeffding Trees

- Assume we have some evaluation function G like information gain or Gini index to assess the goodness of a split up to some number of examples n, and the difference between the best evaluated and the second best evaluated attribute is $\Delta\overline{G} = \overline{G}(\text{best})-\overline{G}(2^{nd}\text{ best}) \geq 0$.
- Then, given a desired δ, the Hoeffding bound guarantees that best is the correct choice with probability $1-\delta$ and $\Delta\overline{G}>\varepsilon$.

(Concept) Drift

- One of the main problems with stream mining methods
- Gradual drift, abrupt drift
- Drift in class or in attributes/features
- Recurrence of distributions
- Methods range from simple sliding window based ones (moving average type) to classifier based