

IS 2: The PetaByte Challenge

Arno Siebes



Kilo	$2^{10} = 1,024$
Mega	$2^{20} = 1,048,576$
Giga	$2^{30} = 1,073,741,824$
Tera	$2^{40} = 1,099,511,627,776$
Peta	$2^{50} = 1,125,899,906,842,624$
Exa	$2^{60} = 1,152,921,504,606,846,976$
Zetta	$2^{70} = 1,180,591,620,717,411,303,424$
Yotta	$2^{80} = 1,208,925,819,614,629,174,706,176$

Bytes in Use

▶ Terabyte:

- 3 - 4 standard harddisks: \$ 500
- Library of Congress: 8 Terabytes of Text

▶ Petabyte:

- The Internet Archive Wayback Machine: roughly 1 Petabyte
- NOB Cross Media Facilities: 1.5 Petabyte storage network



Near future: PBs of data:

- ▶ Genomics
- ▶ Transcriptomics
- ▶ Proteomics
- ▶ Medline
- ▶ Health data:
 - personal data
 - diagnostic data
 - treatment data
 - experimental data

Not in one Database

- ▶ Large central servers with curated and annotated data
- ▶ Smaller publically available databases with experimental data
- ▶ In-house data

the data is distributed over many tables in many databases



Data Analysis

Standard Data Analysis requires:

- ▶ all the data in one table
- ▶ a one-one correspondence between rows in the table and objects in the real world
- ▶ the tuples (i.e., the objects) are drawn i.i.d

Here we have multiple tables in multiple databases! Can we put all this data in one table?



Probably not

- ▶ joining all the tables into one big table is easy
- ▶ but keeping the 1-1 relation between tuples and objects is not:
 - this only works iff all tables are 1-1 related
 - but many relations are 1-n or even n-m
- ▶ one object will be related to a set of tuples



Multi Relational Data Mining

- ▶ If we cannot join the relations without losing semantics (the 1-1 relationship)
- ▶ we have to analyse sets of related tables
- ▶ this is what multi relational data mining is about



A General Toolbox

Multi Relational mining contains many popular topics as special case, e.g.,

- ▶ text mining
- ▶ mining sets of time-series
- ▶ multi-media mining
- ▶ web mining



How is this done?

By generalizing the standard approach:

- ▶ some methods have a dependent variable in a *target table*
- ▶ other methods try to model joint probability-distributions
- ▶ they still optimize to, e.g., squared-error reduction, variance reduction, mutual-information preservation, ...

To illustrate, we look at subgroup discovery



Subgroup Discovery

Try to find regions in the input (attribute/feature) space with relatively high (low) values for the target variable.

Applications:

- ▶ Identifying interesting market segments
- ▶ Industrial process control
- ▶ Credit scoring
- ▶ All kinds of *selection* problems



Formal statement of the problem

x_1, x_2, \dots, x_p : input variables, y :target.

Let S_j denote the set of possible values of x_j . The input space is

$$S = S_1 \times S_2 \times \dots \times S_p$$

The objective is to find subregions $R \subset S$ for which

$$\bar{y}_R \gg \bar{y},$$

where \bar{y} is the global mean of y and \bar{y}_R the mean of y in R .



Definition of a box

The regions we are looking for must have “rectangular” shape, hence we call them boxes.

Let $s_i \subseteq S_i$. We define a box

$$B = s_1 \times s_2 \times \dots \times s_p$$

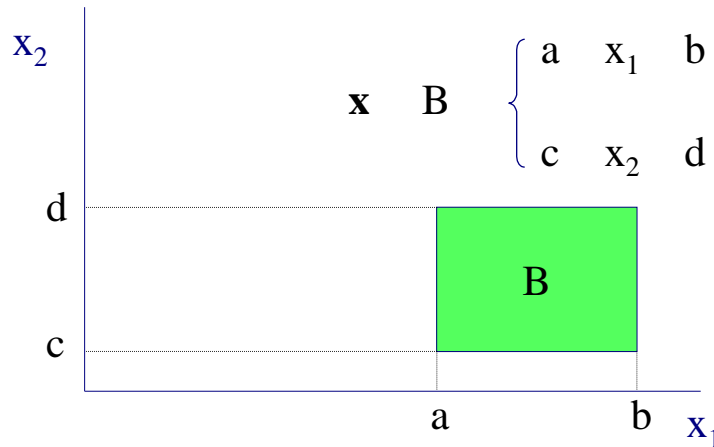
where $\mathbf{x} \in B \equiv \bigcap_{j=1}^p (x_j \in s_j)$.

When $s_i = S_i$, we leave x_i out of the box definition since it may take any value in its domain.



Example of box on two numeric variables

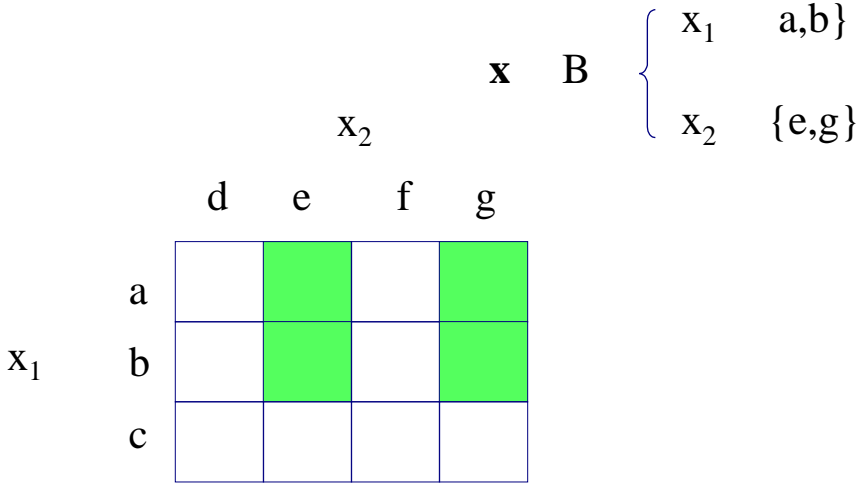
Example of a box defined on two numeric variables, where $\mathbf{x} \in B \equiv x_1 \in [a, b] \cap x_2 \in [c, d]$.



Example of box on two categorical variables

Example of a categorical box where

$$\mathbf{x} \in B \equiv x_1 \in \{a, b\} \cap x_2 \in \{e, g\}.$$



Boxes may also be defined on combinations of numeric and categorical variables.



Covering strategy

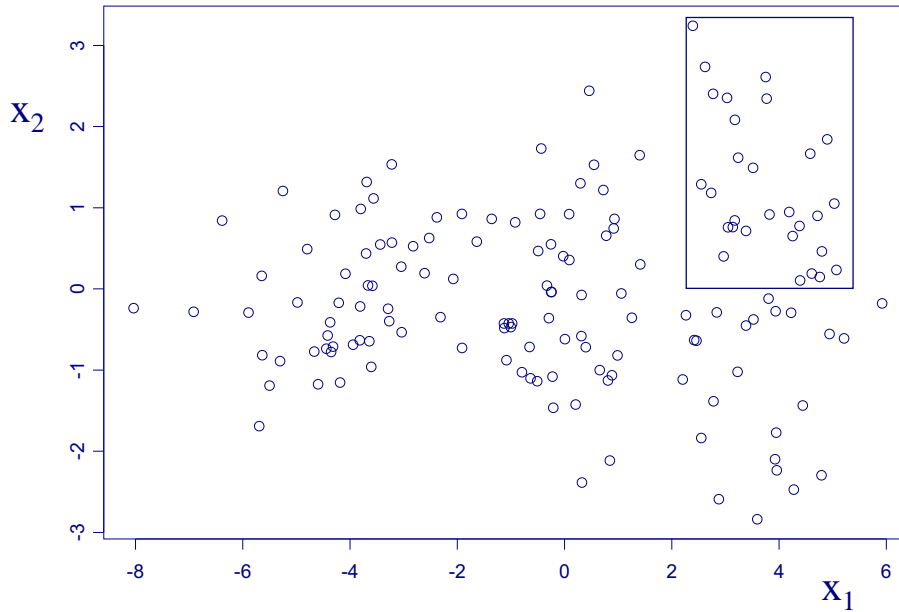
The same box construction (rule induction) algorithm is applied sequentially to subsets of the data:

- ▶ The first box, B_1 , is constructed on the entire data set.
- ▶ For the construction of the second box, B_2 , we remove the data points that fall into B_1 .
- ▶ In general, B_K is constructed on $\{y_i, \mathbf{x}_i \mid \mathbf{x}_i \notin \bigcup_{k=1}^{K-1} B_k\}$.



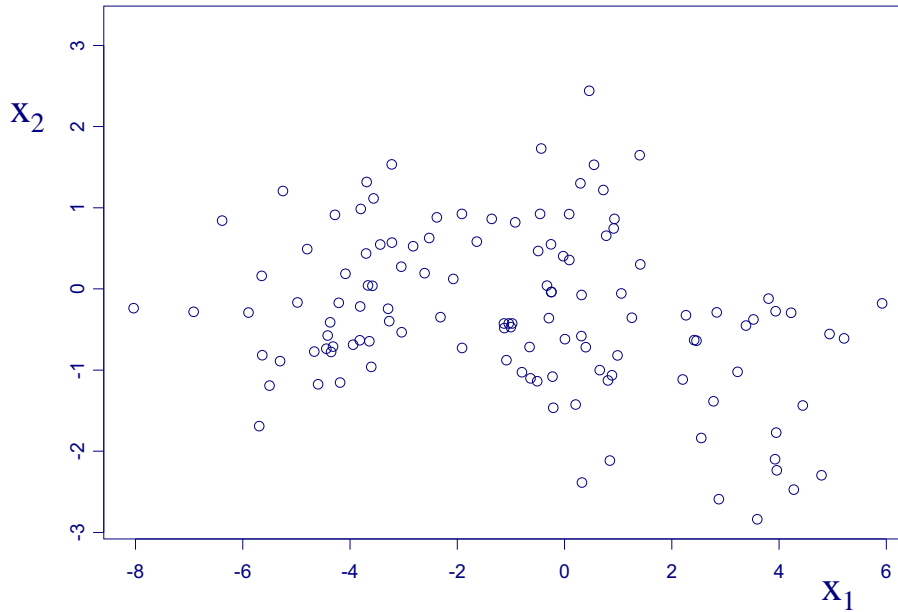
Covering(1)

The first box...



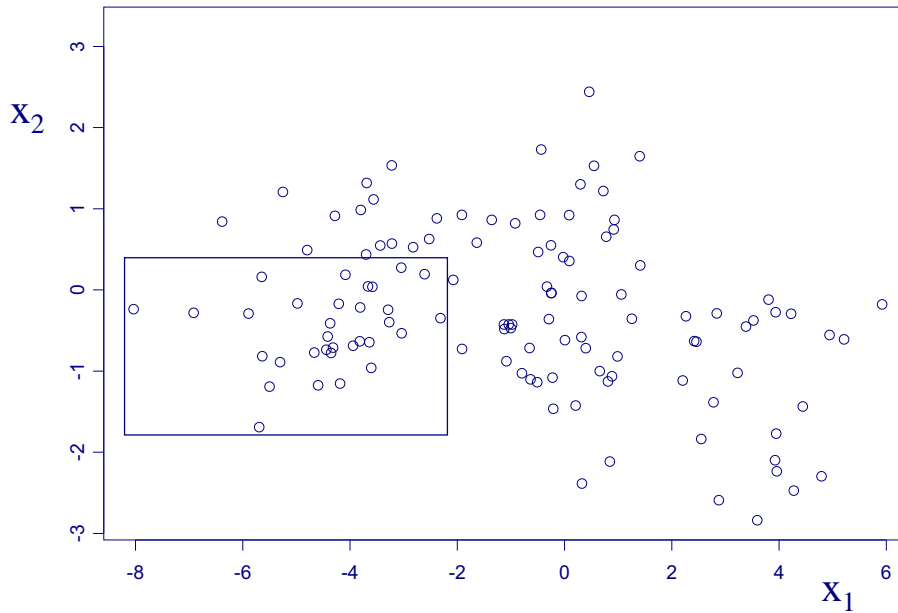
Covering(2)

Data for construction of the second box...



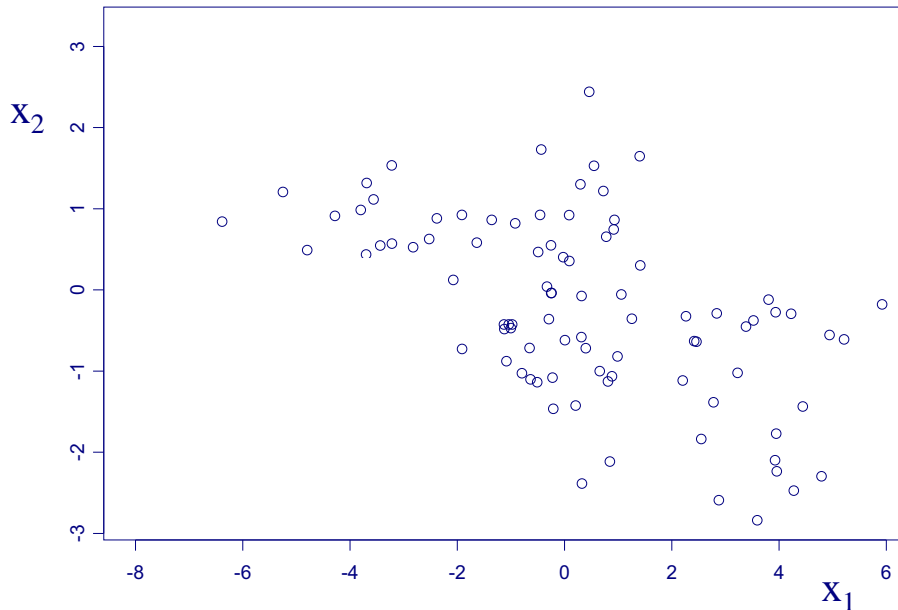
Covering(3)

The second box...



Covering(4)

Data for construction of the third box...



Covering: when do we stop?

Box construction continues until

- ▶ no box in the remaining data with
 - sufficient *support*, and
 - sufficiently high target mean
- ▶ when the user wants to stop!

In computing the *support* of B_K we count the data points that fall into B_K (but not into any of the previous boxes) and divide by number of observation of the entire data set.



Box construction (rule induction)

- ▶ Given the data (or a subset of the data), produce a box with target mean as large as possible
- ▶ Not feasible to consider all possible boxes
- ▶ Apply heuristic search to find a good box
 - zoom in on the data fast, aggressively optimizing the target function
 - search patiently by peeling off only small parts of the initial box



How Do We Generalise?

- ▶ We have a target attribute in the target table
- ▶ for the target table nothing changes
- ▶ for this talk we assume (transitively) 1-n relationships between the target table and the other tables
- ▶ so, for each tuple in the target table we have a set of tuples in the other tables
- ▶ What information can we get from a set of tuples?



Aggregates

- ▶ Since subgroup mining is attribute value based, i.e.,
- ▶ $A_i \in \{v_{i,1}, \dots, v_{i,k}\}$
- ▶ it seems a good idea to compute *aggregates* over attributes in the related tables,
- ▶ e.g., Sum, Max, Count, ...
- ▶ aggregates are functions that compute one value from a set of values.
- ▶ the aggregate functions are computed on the fly and the best one is chosen.



The Relational Generalisation

- ▶ One table is the *target* table: we search for subgroups of the target table.
- ▶ subgroups can be defined by aggregated values on associated tables

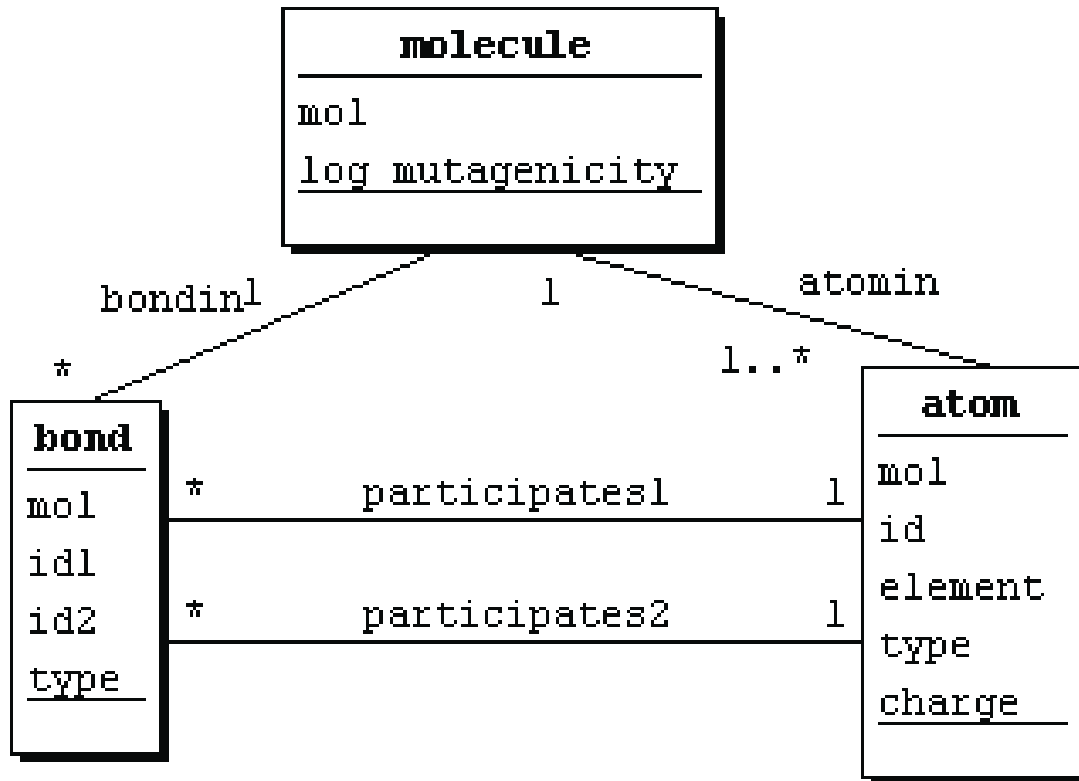


Concrete Example

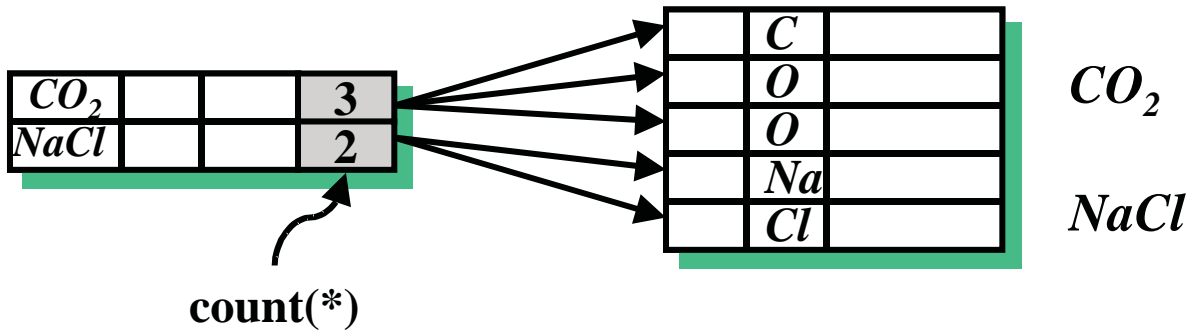
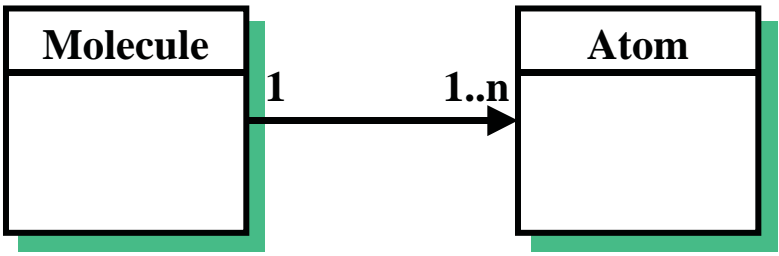
- ▶ Predict the mutagenicity of a set of 230 aromatic and heteroaromatic nitro compounds.
- ▶ Mutagenicity is measured using the Ames test using *S. typhimurium* TA98.



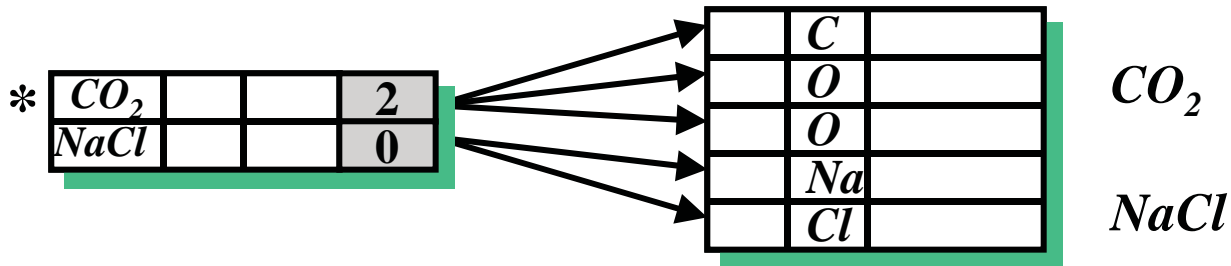
The Data



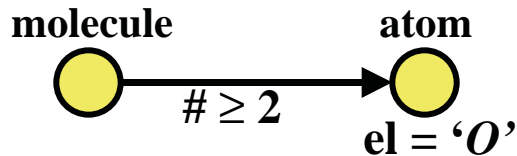
Aggregates



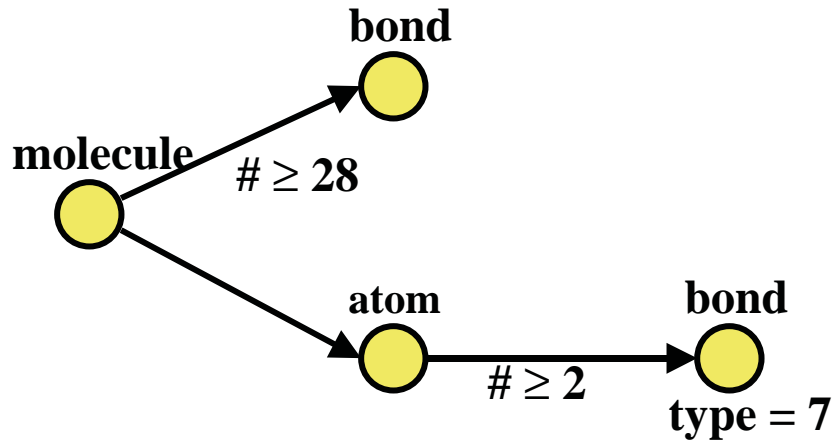
A Graphical Notation



‘all molecules that contain at least 2 *O*-atoms’



A 100% Result



The project

UU multi-relational generalisations of: Bayesian
Networks, Classification and Clustering

CWI database support



Database Support?

- ▶ Joins and Aggregates are computed on the fly
- ▶ One mining algorithm generates tens of thousands of queries
- ▶ If one query takes 10 seconds (not bad on a petabyte) one run will last for days
- ▶ database optimization is crucial!



The Context

UU many related projects (the first PhD was awarded last month)

CWI a long research tradition, Monet has proven its worth for data mining

Joint history means we speak each others language

