Maximally Informative k-Itemsets and their Efficient Discovery

Arno J. Knobbe , Eric K.Y. Ho

Presented by: Marling Engle

Background – Entropy

- Box of balls
  - H(All balls different colors) = MAX
    - Equal probability in all cases is maximum entropy
    - H(More red than others) < MAX
  - Begin drawing balls randomly
    - First ball drawn gives most <u>information</u>
    - Entropy decreases as you continue to draw
      - Last ball : zero entropy, you *know* which ball is there
        - Zero information gained by drawing the ball
        - Certain outcome = zero entropy

Background – Joint entropy

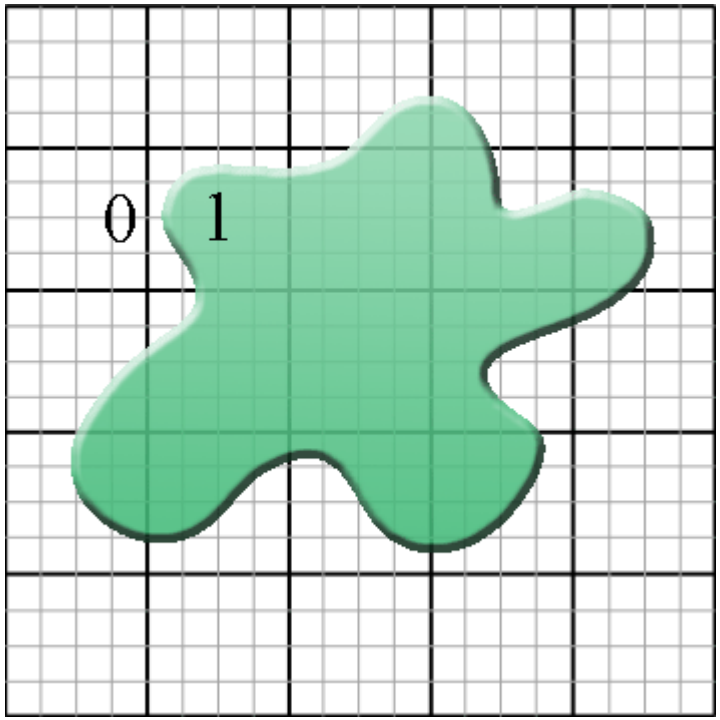
- Measure of entropy in a joint system with 2 random variables
- $H(X,Y) \geq H(X)$ and $H(X,Y) \geq H(Y)$
- $H(X,Y) \leq H(X) + H(Y)$
- Used in paper as a quality measure
  - Maximally informative k-itemsets
    - Informative, in terms of entropy

Motivation

- <u>Subgroup discovery</u>
- Mining binary data
- Current methods produce large numbers of rules
- How to obtain a smaller rule set
  - Must cover as much of the initial rules as possible
  - Results shown in the first experiment

Motivation – Extension

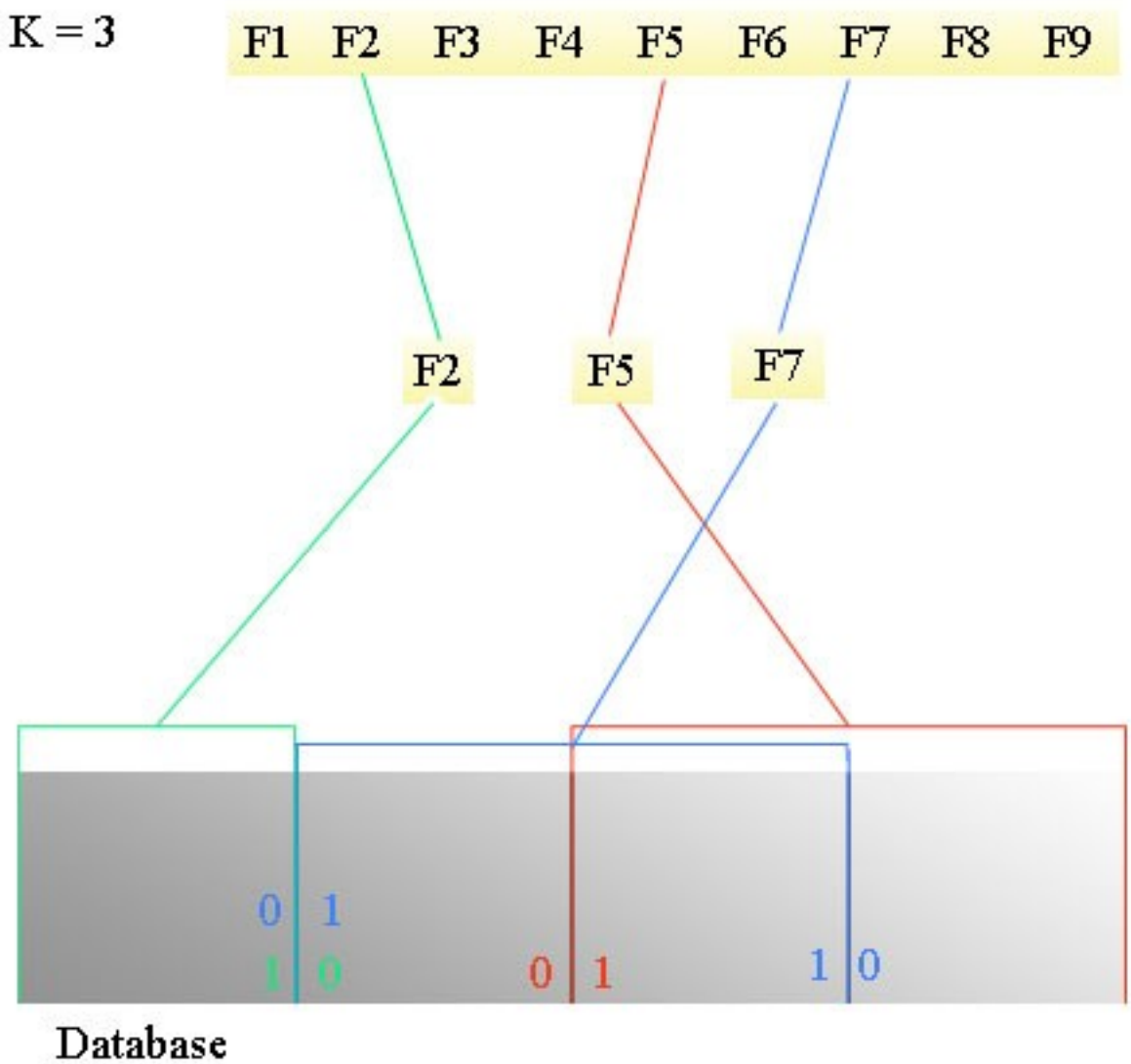
- Treating binary feature (item) as a means of creating 2 subsets from database
  - Works for patterns and more complex rules as well
- Additional binary classifiers
  - Anything that splits database into mutually exclusive parts
  - Pattern
    - Under pattern or not



The Problem

- Select subset (size k) from the itemset such that:
  - Database is partitioned as uniformly as possible over the parts
  - Subset is maximally informative
- Maximize distinctive power (information)/entropy
- Minimize redundancy within feature set

Example Miki



- 3 itemset chosen
- F2,F5,F7 are maximally informative
  - Distribution even
- 2 itemset Miki:
  - F5,F7
  - Why?

Algorithm

Algorithm **ExhaustiveMiki(*k, n*)**

$X \leftarrow [1,\ldots,k]$

$h_{\max} \leftarrow$ JointEntropy(X)

$$H(X) = -\sum_{B \in \{0,1\}^k} p(x_1 = b_1,\ldots,x_k = b_k) \lg p(x_1 = b_1,\ldots,x_k = b_k)$$

$Y \leftarrow X$

while LexicographicSuccessor(X, n) $\neq$ "undefined"

  $X \leftarrow$ LexicographicSuccessor(X, n)

  $h \leftarrow$ JointEntropy(X)

  if $h \geq h_{\max}$

    $h_{\max} \leftarrow h$

    $Y \leftarrow X$

return Y

Example

- Items A to C all have equal numbers of 1' s and 0' s, hence
- H(A) = 1, H(B) = 1, H(C) = 1. H(D) = -(3/8) lg(3/8) -(5/8)lg(5/8) = 0.96.
- The itemset {A, B, C} is a maximally informative k-itemset of cardinality 3.

A	B	C	D
1	1	1	0
1	1	0	0
1	1	1	0
1	0	0	0
0	1	1	0
0	0	0	1
0	0	1	1
0	0	0	1

Challenges

- Continually scanning the dataset to calculate the entropy of given itemsets
  - (Scans the entire database N choose k times)
    - (N=total number of items, k=length of Miki to find)
- Providing exact solutions becomes impractical
  - Increasing the number of items to be chosen
    - Must calculate the joint entropy of every k-sized combination of the items
    - Only possible where k is small, and the

Solutions

- Too many database reads problem
  - Scan the database once, storing precomputed entropies for the itemsets
- Increasing k problem
  - Use a forward selection
    - Chooses only a subset of the possible combinations
  - IE: progressively add one item at a time, greedily choosing the one which joint entropy is maximized

Forward Scan

- ForwardScan was final algorithm given by authors
  - Quickly produces results which are comparable to exact solutions
- Example:
    - ABCDE items
    - AB are already selected, searching for 4 item sets
    - Try: ABC, ABD, ABE.. choose max (ABD, for example)
      - Try ABDC, ABDE
        - Highest entropy = approximate Miki

Miki vs. Frequent Itemset (FIS)

- Miki treats 0 and 1 symmetrically
  - Most frequent itemset only use a minimum threshold for support
    - Would need a *maximum* support threshold as well
  - Minimum and maximum support thresholds lost in favor of joint entropy
    - The features with the most information are used

Miki vs. FIS (cont'd)

- FIS favor positively associated items
  - Miki favors independence of items within the Miki
  - Example:
    - A and B appear together very commonly
      - Shows up as a FIS, because they are closely associated
      - Will *not* show up in MIS: if I know A, I don't need B
- FIS sensitive to items with high support
  - Miki ignores them, frequent = little information

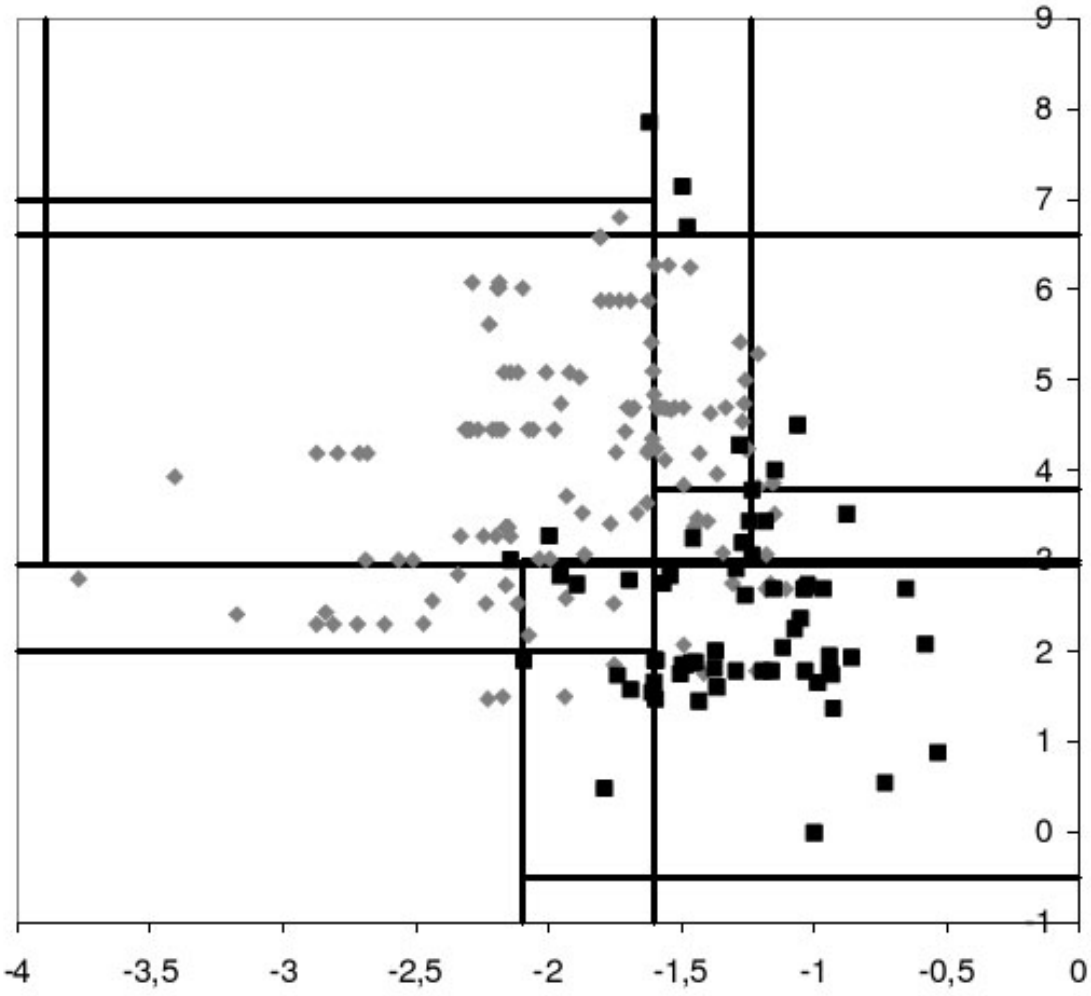
Applications

- Feature Selection
  - Reduce features of a database to essential feature set – improve efficiency for other datamining algorithims
- Subgroup Discovery
  - Usually produces a large number of interesting patterns – can be filtered, to remove redundant patterns while retaining maximum information

Experiment

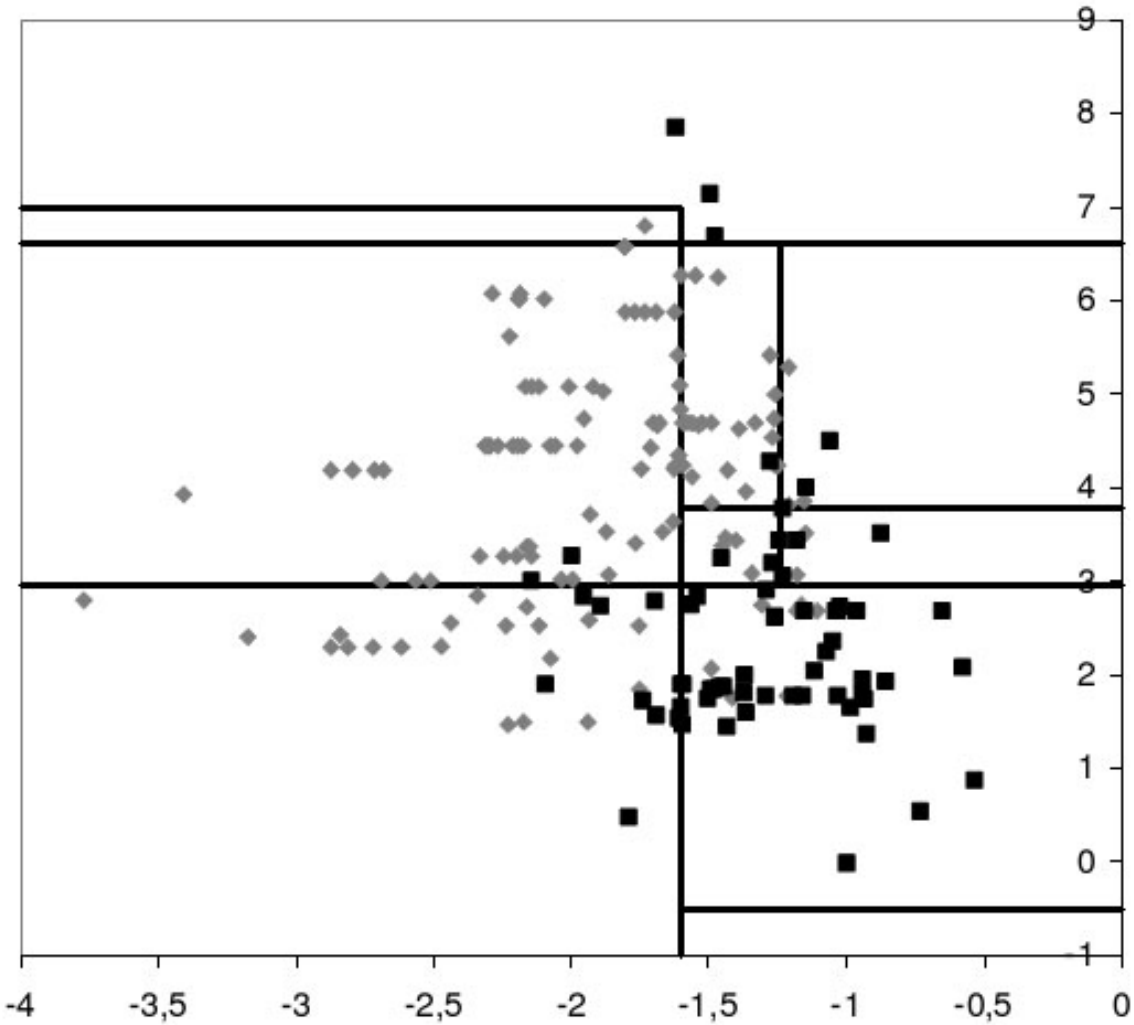
- Demonstrating usefulness of miki's in the context of subgroup discovery
  - (Authors main motivation)
- Shows 2 numeric features (lumo en logp) associated with 188 molecules
  - Appears in *Mutagensis* database [21]
- Molecules appear in two classes
  - Mutagenic (grey dots)
  - Non-mutagenic (black dots)

Experiment - Before



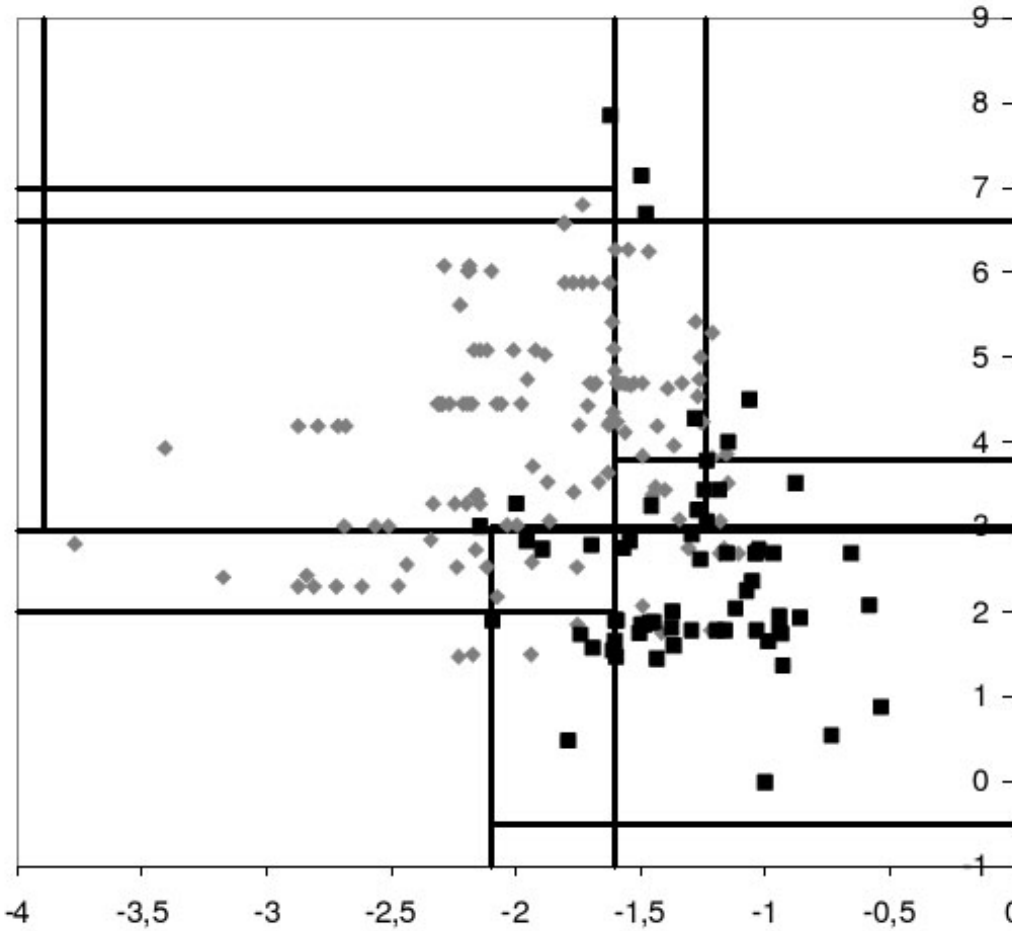
- Axis-parallel lines represent decision boundaries
  - Formed by collective 82 subgroups (rules)
  - Discovered by Safarii [13][20] (datamining package)

Experiment - After

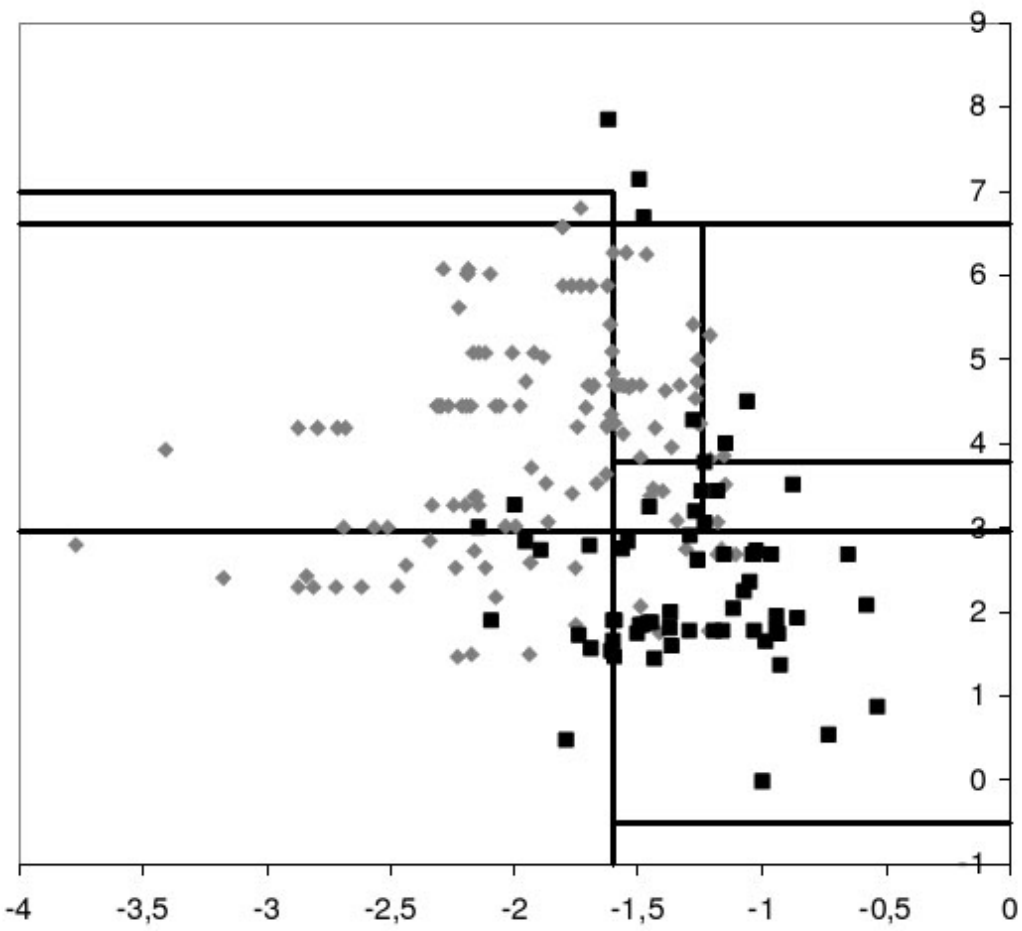


- Each rule = item
- After miki's are found, 4 rules were used
- This 4 rule subset captures almost same as 82 rules

Experiment – Before and After



Before (82 rules)



After (4 rules)

Related Work

- Feature Selection
  - Most other algorithms use supervised learning: selecting only features relevant for predicting a classifier
    - Must be user defined
  - Mikis allow unsupervised feature-set reduction

References

- [21] Srinivasan, A., Muggleton, S.H., Sternberg, M.J.E., King, R.D., Theories for mutagenicity: A study in first-order and feature-based induction, Artificial Intelligence, 85(1,2), 1996
- [13] Knobbe, A.J., Multi-Relational Data Mining, Ph.D. Dissertation, 2004, http://www.kiminkii.com/thesis.pdf
- [20] Safarii Multi-Relational Data Mining environment, http://www.kiminkii.com/safarii.html, 2006