Mining for Patterns Based on Contingency Tables by KL–Miner First Experience

Jan Rauch[(1,3)], **Milan Šimůnek**[(2,4)], **Václav Lín**[(1)]

[(1)] *Department of Knowledge and Information Engineering,* [(2)] *Department of Information Technologies,*
[(3)] *EuroMISE centrum - Cardio,* [(4)] *LISp*
University of Economics Prague, nám. W. Churchilla 4, 130 67 Praha 3, Czech Republic
E-mail: rauch@vse.cz, simunek@vse.cz, xlinv05@vse.cz

Presented at ICDM 2003 workshop *Foundations and New Directions of Data Mining* see
`http://www.cs.sjsu.edu/faculty/tylin/icdm03_workshop.html`

Abstract

A new datamining procedure called KL–Miner is presented. The procedure mines for various patterns based on evaluation of two–dimensional contingency tables, including patterns of statistical nature. The procedure is a result of continued development of the academic LISp-Miner system for KDD.

Keywords

Data mining, contingency tables, the system LISp–Miner, statistical patterns

1 Introduction

Goal of this paper is to present first experience with data mining procedure KL-Miner. The procedure mines for patterns of the form

$$R \sim C/\gamma \, .$$

Here R and C are categorial attributes, the attribute R has *categories* (possible values) $r_1, \ldots, r_K$, the attribute C has categories $c_1, \ldots, c_L$. Further, γ is a Boolean attribute.

The KL-Miner procedure deals with data matrices. We suppose that R and C correspond to columns of the analysed data matrix. We further suppose that the Boolean attribute γ is somehow derived from other columns of the analysed data matrix and thus that it corresponds to a Boolean column of the analysed data matrix.

The intuitive meaning of the expression $R \sim C/\gamma$ is that the attributes R and C are in relation given by the symbol $\sim$ when the condition given by the derived Boolean attribute γ is satisfied.

The symbol $\sim$ is called *KL-quantifier*. It corresponds to a condition imposed by the user on the contingency table of R and C. There are several restrictions that the user can choose to use (e.g. minimal value, sum over the table, value of the χ^2 statistic, and other).

We call the expression $R \sim C/\gamma$ a *KL-hypothesis* or simply *hypothesis*. The KL-hypothesis $R \sim C/\gamma$ is *true* in the data matrix $\mathcal{M}$ if the condition corresponding to the KL-quantifier $\sim$ is satisfied for the contingency table of R and C on the data matrix $\mathcal{M}/\gamma$. The data matrix $\mathcal{M}/\gamma$ consists of all rows of data matrix $\mathcal{M}$ satisfying the condition γ (i.e. of all rows in which the value of γ is TRUE).

Input of the procedure KL-Miner consists of the analysed data matrix and of several parameters defining a set of potentially interesting hypotheses. Such a set can be very large. The procedure KL-Miner automatically generates all potentially interesting hypotheses and verifies them in the analysed data matrix. The output of the procedure KL-Miner consists of all hypotheses that are true in the analysed data matrix (i.e. supported by the analysed data).

Some details about input of the KL-miner procedure are given in the section 2. KL-quantifiers are described in section 3.

Implementation of the KL-Miner procedure is based on a bit string approach [3, 4]. The principles of the KL-Miner procedure implementation are described in section 4. An example of application is given in section 5. Some remarks on scalability are in section 5.2 .

The KL-Miner procedure is a part of the LISp-Miner system[1][4, 6]. The LISp-Miner system consists of several data mining procedures that can be combined in various ways to enhance the mining power.

Let us remark that the KL-Miner is a GUHA procedure in the sense of the book [1]. Therefore, we shall use the terminology introduced in [1]. The potentially

[1] See `http://lispminer.vse.cz`

interesting hypotheses will be called *relevant questions*, and the hypotheses that are true in the analysed data matrix will be called *relevant truths*. Furthermore, the use of the term *quantifier* for the symbol $\sim$ in the expression $R \sim C/\gamma$ is inspired by [1]. The cited book contains rich enough theoretical framework to build a formal logical theory for the KL-Miner-style data mining; however, we shall not do it here. Furthermore, KL-Miner is related to some (by now obsolete) GUHA procedures from 80's, namely to the *COLLAPS* and *CORREL* procedures, see [2]. Let us also remark that the KL-Miner procedure is related to the procedure 49er [8].

2 KL-Miner Input

Please recall that the KL-Miner procedure mines for hypotheses of the form $R \sim C/\gamma$ where R and C are categorial attributes, γ is a Boolean attribute and $\sim$ is a KL-quantifier. The attribute R is called the *row attribute*, the attribute C is called the *column attribute*.

Input of the KL-Miner procedure consists of

- the analysed data matrix
- a set $\mathcal{R} = \{R_1, \ldots, R_u\}$ of row attributes
- a set $\mathcal{C} = \{C_1, \ldots, C_v\}$ of column attributes
- specification of the KL-quantifier $\sim$
- several parameters defining set Γ of *relevant conditions* (i.e. derived Boolean attributes) γ, see below.

The KL-Miner procedure automatically generates and verifies all relevant questions

$$R \sim C/\gamma \,,$$

such that $R \in \mathcal{R}$, $C \in \mathcal{C}$ and $\gamma \in \Gamma$. The set of all relevant questions is denoted RQ.

Each relevant condition γ is a conjunction of several *partial conditions*. Each partial condition is a conjunction of *literals*. Literal is an expression of the form $B(\omega)$ or $\neg B(\omega)$, where B is an attribute (derived column of the analysed data matrix) and ω is a subset of all possible values (i.e. categories) of B. The subset ω is called a *coefficient* of the literal $B(\omega)$ (or $\neg B(\omega)$). The literal $B(\omega)$ is called *positive literal*, $\neg B(\omega)$ is called *negative literal*.

$B(\omega)$ is a Boolean attribute that is true in the row o of analysed data matrix iff the value in the column B in the row o belongs to the set ω. $\neg B(\omega)$ is a Boolean attribute that is a negation of $B(\omega)$.

The set Γ of relevant conditions to be automatically generated is given by definitions of all particular partial conditions (recall that each relevant condition is a conjunction of partial conditions). That is,

$$\Gamma = \Big\{ \bigwedge_{i=1}^{t} \gamma_i \big| \gamma_1 \in \Gamma_1, \gamma_2 \in \Gamma_2, \ldots, \gamma_t \in \Gamma_t \Big\} \,,$$

where $\Gamma_1, \Gamma_2, \ldots, \Gamma_t$ are sets of partial conditions. Each set Γ_i of partial conditions (i.e. conjunctions of literals) is defined by

- a minimal and maximal *lenght* (i.e. number of literals) of conjunctions in the set,
- a list $\mathcal{A} = \{A_1, \ldots, A_w\}$ of attributes from which literals will be automatically generated, some of these attributes are marked as *basic* (partial condition must contain at least one literal derived from a basic attribute),
- a simple definition of the set of all literals to be generated from each attribute from $\mathcal{A}$.

Let us remark that both minimal and maximal lengths of conjunctions can be 0, this results in conjunctions of zero length. The value of the conjunction of zero length is always TRUE. Thus the conjunction $\bigwedge_{i=1}^{t} \gamma_i$ can be seen as the conjunction $\bigwedge \gamma_{i_j}$ where γ_{i_j} are all the γ_i with positive length for $i = 1, \ldots, t$.

The set of all literals to be generated for a particular attribute is given by:

- a *type* of coefficient. There are six types of coefficients available: *subsets*, *intervals*, *left cuts*, *right cuts*, *cuts*, *one particular value*.
- *coefficient lenght* – a minimal and maximal number of categories (i.e. values) in the coefficient.
- positive/negative literal option:
  - generate only positive literals
  - generate only negative literals
  - generate both positive and negative literals.

Let us give examples of particular types of coefficients for an attribute A with categories $\{1, 2, 3, 4, 5\}$:

- **subsets**: definition of subsets with 2-3 categories defines literals $A(1,2)$, $A(1,3)$, $A(1,4)$, $A(1,5)$, $A(2,3)$, …, $A(3,4)$, …, $A(4,5)$, $A(1,2,3)$, $A(1,2,4)$, $A(1,2,5)$, $A(2,3,4)$, …, $A(3,4,5)$ (We write $A(1,2)$ instead of $A(\{1,2\})$ etc.)
- **intervals**: definition of intervals with 2-3 categories defines literals $A(1,2)$, $A(2,3)$, $A(3,4)$, $A(4,5)$, $A(1,2,3)$, $A(2,3,4)$ and $A(3,4,5)$

- **left cuts**: definition of left cuts with maximally 3 categories defines literals $A(1)$, $A(1,2,3)$ and $A(1,2,3)$
- **right cuts**: definition of right cuts with maximally 4 categories defines literals $A(5)$, $A(5,4)$, $A(5,4,3)$ and $A(5,4,3,2)$
- **cuts** means both left cuts and right cuts.

An example of a relevant question is expression

$$R_1 \sim C_1/A_1(1,2) \wedge A_2(3,4) \ .$$

Here R_1 is the row attribute, C_1 is the column attribute and the condition is $A_1(1,2) \wedge A_2(3,4)$. This condition means that value of the attribute A_1 is 1 or 2, and value of the attribute A_2 is 3 or 4. An example of KL-Miner input is in section 5.

The KL-Miner is a part of the LISP-Miner system [6]. Another part of the LISP-Miner system is the 4ft-Miner procedure [4, 6]. The 4ft-Miner procedure mines for association rules of the form

$$\varphi \approx \psi$$

and for conditional association rules of the form

$$\varphi \approx \psi/\gamma$$

where φ, ψ and γ are derived Boolean attributes (conjunctions of literals). Intuitive meaning of $\varphi \approx \psi$ is that φ and ψ are in relation given by the symbol $\approx$. Intuitive meaning of $\varphi \approx \psi/\gamma$ is that φ and ψ are in relation given by the symbol $\approx$ when the condition γ is satisfied.

Symbol $\approx$ is called *4ft–quantifier*. It corresponds to a condition concerning four fold contingency table of φ and ψ. Various types of dependencies of φ and ψ can be expressed this way. An example is the classical association rule with confidence and support, another example is a relation corresponding to the χ^2-test of independence.

The left part of the association rule, i.e. φ, is called *antecedent*, the right part of the association rule, i.e. ψ, is called *succedent*, and γ is called *condition*.

The input of the 4ft-Miner procedure consists of

- the analysed data matrix
- several parameters defining set Φ of relevant antecedents φ
- several parameters defining set Ψ of relevant succedents ψ
- several parameters defining set Γ of relevant conditions γ
- the 4ft-quantifier $\approx$.

Parameters defining sets Φ, Ψ and Γ have the same structure as the parameters defining the set Γ of relevant conditions in the input of the KL-Miner procedure, see above. This fact is very important from the point of view of implementation of the procedure KL-Miner, see section 4.

3 KL-quantifiers

The KL-Miner mines for hypotheses of the form

$$R \sim C/\gamma$$

where R and C are categorial attributes with admissible values $r_1, \ldots, r_K$ and $c_1, \ldots, c_L$, respectively. γ is a relevant condition.

Hypothesis $R \sim C/\gamma$ is true in the data matrix $\mathcal{M}$ if a condition corresponding to the KL-quantifier $\sim$ is satisfied for the contingency table of R and C on the data matrix $\mathcal{M}/\gamma$. The data matrix $\mathcal{M}/\gamma$ consists of all rows of data matrix $\mathcal{M}$, for which the value of γ is TRUE.

We suppose that the contingency table of attributes R and C on the data matrix $\mathcal{M}/\gamma$ has the form of Tab. 1, where:

- $n_{k,l}$ denotes the number of rows in data matrix $\mathcal{M}/\gamma$ for which $R = r_k$ and $C = c_l$
- $n_{k,*} = \sum_l n_{k,l}$ denotes the number of rows in data matrix $\mathcal{M}/\gamma$ for which $R = r_k$
- $n_{*,l} = \sum_k n_{k,l}$ denotes the number rows in data matrix $\mathcal{M}/\gamma$ for which $C = c_l$
- $n = \sum_k \sum_l n_{k,l}$ denotes the number of all rows in data matrix $\mathcal{M}/\gamma$

We shall also use relative frequencies:

- $f_{k,l} = n_{k,l}/n$
- $f_{k,*} = \sum_l f_{k,l} = n_{k,*}/n$
- $f_{*,l} = \sum_k f_{k,l} = n_{*,l}/n$

Semantics of the KL-quantifier $\sim$ is determined by the user, who can choose to set lower / upper threshold values for several functions on the table of absolute or relative frequencies. These functions among others include:
Simple aggregate functions:

$$\min_{k,l}\{n_{k,l}\} \ , \ \max_{k,l}\{n_{k,l}\} \ , \ \sum_{k,l} n_{k,l} \ , \ \frac{1}{KL}\sum_{k,l} n_{k,l}$$

$\mathcal{M}/\gamma$	c_1	...	c_L	Σ_l
r_1	$n_{1,1}$	...	$n_{1,L}$	$n_{1,*}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$
r_K	$n_{K,1}$	...	$n_{K,L}$	$n_{K,*}$
Σ_k	$n_{*,1}$	...	$n_{*,L}$	n

Table 1: Contingency table of R and C on $\mathcal{M}/\gamma$ - absolute frequencies

A simple non-statistical measure expressing the fact, that C is a function of R:

$$Fnc_S = \frac{1}{n}\sum_k \max_l\{n_{k,l}\}\ .$$

Fnc_S takes values from $\langle L^{-1}, 1\rangle$. It is $Fnc_S = 1$ iff for each category r_k of R, there is exactly one category c_l of C, such that $n_{k,l}$ is nonzero; it is $Fnc_S = L^{-1}$ iff $(\forall k \forall l) n_{k,l} = \max_j\{n_{k,j}\}$ (i.e. each row's distribution of frequencies is uniform).

Statistical and information theoretic functions:

Information dependence[5]:

$$ID = 1 - \frac{\sum_k f_{k,*}\log_2 f_{k,*} - \sum_{k,l} f_{k,l}\log_2 f_{k,l}}{-\sum_l f_{*,l}\log_2 f_{*,l}}\ .$$

Note that ID corresponds to

$$1 - \frac{-H(R) + H(C,R)}{H(C)} = 1 - \frac{H(C|R)}{H(C)}\ ,$$

where $H(.)$, $H(.,.)$ and $H(.|.)$ denote entropy, joint entropy and conditional entropy, respectively. ID takes values from $\langle 0, 1\rangle$; $ID = 0$ iff C is independent of R (i.e. $H(C|R) = H(C)$), and $ID = 1$ iff C is a function of R (i.e. $H(C|R) = 0$).

The Pearson χ^2 statistic[5]:

$$\chi^2 = \sum_{k,l} \frac{\big(n_{k,l} - n_{k,*}n_{*,l}/n\big)^2}{n_{k,*}n_{*,l}/n}\ .$$

Some further measures are planned to be implemented, the most prominent of them being the Kendall's coefficient and the mutual information.

Kendall's coefficient[5] is

$$\tau_b = \frac{2(P-Q)}{\sqrt{\big(n^2 - \sum_k n_{k,*}^2\big)\big(n^2 - \sum_l n_{*,l}^2\big)}}\ ,$$

where

$$P = \sum_{k,l} n_{k,l}\sum_{i>k}\sum_{j>l} n_{i,j}\ ,\ Q = \sum_{k,l} n_{k,l}\sum_{i>k}\sum_{j<l} n_{i,j}\ .$$

τ_b takes values from $\langle -1, 1\rangle$ with the following interpretation: $\tau_b > 0$ indicates positive ordinal dependence[2], $\tau_b < 0$ indicates negative ordinal dependence, $\tau_b = 0$ indicates ordinal independence, $|\tau_b = 1|$ indicates that C is a function of R.

Mutual information

$$I_m = \frac{\sum_{k,l} f_{k,l}\big(\log_2 f_{k,l} - \log_2 f_{k,*}f_{*,l}\big)}{\min\big[-\sum_l f_{*,l}\log_2 f_{*,l}, -\sum_k f_{k,*}\log_2 f_{k,*}\big]}$$

corresponds to $\big(H(C) - H(C|R)\big)/\min\big[H(C), H(R)\big]$ and has the following properties: $0 \le I_m \le 1$, $I_m = 0$ indicates independence, $I_m = 1$ indicates that there is a functional dependence of C on R or of R on C.

Values of χ^2, ID, τ_b and I_m have reasonable interpretation only when working with data of statistical nature (i.e. resulting from simple random sampling).

For an example of KL-quantifier, see section 5.1.

4 KL-Miner Implementation

Implementation of the KL-Miner procedure is based on the bit-string approach [3, 4]. Software tools developed earlier for the LISp-Miner system [6] are utilized.

We shall use the data matrix shown in Fig. 1 to explain the principles of implementation of the KL-Miner procedure.

row	R	C	X	Y	Z
o_1	r_2	c_L	x_2	y_3	z_4
o_2	r_K	c_4	x_7	y_2	z_6
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
o_m	r_1	c_1	x_p	y_9	z_3

Figure 1: Data matrix $\mathcal{M}$

We suppose to have only one row attribute R with categories $r_1, \ldots, r_K$, and one column attribute C with categories $c_1, \ldots, c_L$. We also suppose that the relevant conditions will be automatically generated from attributes X, Y and Z with categories $x_1, \ldots, x_p$, $y_1, \ldots, y_q$ and $z_1, \ldots, z_r$, respectively.

Data matrix $\mathcal{M}$ (see Fig. 1) has m rows $o_1, \ldots, o_m$; value of the attribute R in the row o_1 is r_2, value of the attribute C in the row o_1 is c_L etc.

The KL-Miner has to automatically generate and verify relevant questions of the form

$$R \sim C/\gamma$$

[2] i.e. high values of C often coincide with high values of R and low values of C often coincide with low values of R

where the relevant condition γ is automatically generated from attributes X, Y and Z.

The key problem of the KL-Miner implementation is fast computation of contingency tables of attributes R and C on the data matrix $\mathcal{M}/\gamma$ for particular relevant conditions γ, see also Tab. 1.

It means that we have to compute the frequencies $n_{k,l}$ for $k = 1, \ldots, K$ and $l = 1, \ldots, L$. Let us remember that $n_{k,l}$ denotes the number of rows in data matrix $\mathcal{M}/\gamma$ for which $R = r_k$ and $C = c_l$, see section 3. In other words the frequency $n_{k,l}$ is the number of rows in data matrix $\mathcal{M}$ for which the Boolean attribute

$$R(r_k) \wedge C(c_l) \wedge \gamma$$

is true. Recall that $R(r_k)$ and $C(c_l)$ are Boolean attributes - literals, see section 2.

We use a bit-string representation of the analysed data matrix $\mathcal{M}$. Each attribute is represented by *cards* of its particular categories, i.e. the attribute R is represented by cards of categories $r_1, \ldots, r_K$.

For $k = 1, \ldots, K$, the card of the category r_k of the attribute R is denoted $R[r_k]$. $R[r_k]$ is a string of bits. Each row of $\mathcal{M}$ corresponds to one bit in $R[r_k]$. There is "1" in such a bit if and only if there is the value r_k in the corresponding row of the column R. Cards of the categories of the attribute R are shown in Fig. 2.

	attribute	cards of categories of R			
row	R	$R[r_1]$	$R[r_2]$	...	$R[r_K]$
o_1	r_2	0	1	...	0
o_2	r_K	0	0	...	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
o_m	r_1	1	0	...	0

Figure 2: Cards of categories of the attribute R

Cards of categories of attributes C, X, Y and Z are denoted analogously.

Card $\mathcal{S}(\gamma)$ of the Boolean attribute γ is a string of bits that is analogous to card of a category. Each row of data matrix corresponds to one bit of $\mathcal{S}(\gamma)$ and there is "1" in this bit if and only if the Boolean attribute γ is true in the corresponding row.

Clearly, for arbitrary Boolean attributes γ_1 and γ_2

$$\mathcal{S}(\gamma_1 \wedge \gamma_2) = \mathcal{S}(\gamma_1) \dot{\wedge} \mathcal{S}(\gamma_2) \, .$$

Here $\mathcal{S}(\gamma_1) \dot{\wedge} \mathcal{S}(\gamma_2)$ is a bit-wise conjunction of bit-strings $\mathcal{S}(\gamma_1)$ and $\mathcal{S}(\gamma_2)$. Similarly it is

$$\mathcal{S}(\gamma_1 \vee \gamma_2) = \mathcal{S}(\gamma_1) \dot{\vee} \mathcal{S}(\gamma_2) \, .$$

It is important that the bit-wise Boolean operations $\dot{\wedge}$ and $\dot{\vee}$ are realised by very fast computer instructions. Very fast computer instructions are also used to realise a bit-string function $Count(\xi)$ returning number of values "1" in the bit-string ξ.

These bit-string operations and function are used to compute the frequency $n_{k,l}$:

$$n_{k,l} = Count(R[r_k] \dot{\wedge} C[c_l] \dot{\wedge} \mathcal{S}(\gamma)) \, .$$

The only task we have to solve is the task of fast computation of $\mathcal{S}(\gamma))$. Recall that γ is a conjunction of particular conditions and that each particular condition is conjunction of literals.

We will use an example to show how the $\mathcal{S}(\gamma)$ is computed. We will deal with the condition

$$X(x_1, x_2) \ \wedge \ Y(y_1, y_2, y_3, y_4) \, .$$

It is clear that

$$\mathcal{S}(X(x_1, x_2) \ \wedge \ Y(y_1, y_2, y_3, y_4))$$

is equivalent to

$$\mathcal{S}(X(x_1, x_2)) \, \dot{\wedge} \, \mathcal{S}(Y(y_1, y_2, y_3, y_4)) \, ,$$

and further we use the simple facts

$$\mathcal{S}(X(x_1, x_2)) = X[x_1] \dot{\vee} X[x_2]$$

and

$$\mathcal{S}(Y(y_1, y_2, y_3, y_4)) = Y[y_1] \dot{\vee} Y[y_2] \dot{\vee} Y[y_3] \dot{\vee} Y[y_4] \, .$$

There are some simple tricks how to decrease the number of necessary bit-string operations. E.g. if we generate the conditions in the following order:
$X(x_1)$
$X(x_1, x_2)$
$X(x_1, x_2) \wedge Y(y_1)$
$X(x_1, x_2) \wedge Y(y_1, y_2)$
$X(x_1, x_2) \wedge Y(y_1, y_2, y_3)$,
we can use the already generated cards such that, e.g.
$\mathcal{S}(Y(y_1)) = Y[y_1]$
$\mathcal{S}(Y(y_1, y_2)) = \mathcal{S}(Y(y_1)) \dot{\vee} Y[y_2]$
$\mathcal{S}(Y(y_1, y_2, y_3)) = \mathcal{S}(Y(y_1, y_2)) \dot{\vee} Y[y_3]$.
The triple

$$< \mathcal{S}(Y(y_1)), \mathcal{S}(Y(y_1, y_2)), \mathcal{S}(Y(y_1, y_2, y_3)) >$$

can be used to compute cards of further literals derived from the attribute Y. It is e.g.:
$\mathcal{S}(Y(y_1, y_2, y_4)) = \mathcal{S}(Y(y_1, y_2)) \dot{\vee} Y[y_4]$
$\mathcal{S}(Y(y_1, y_2, y_5)) = \mathcal{S}(Y(y_1, y_2)) \dot{\vee} Y[y_5]$

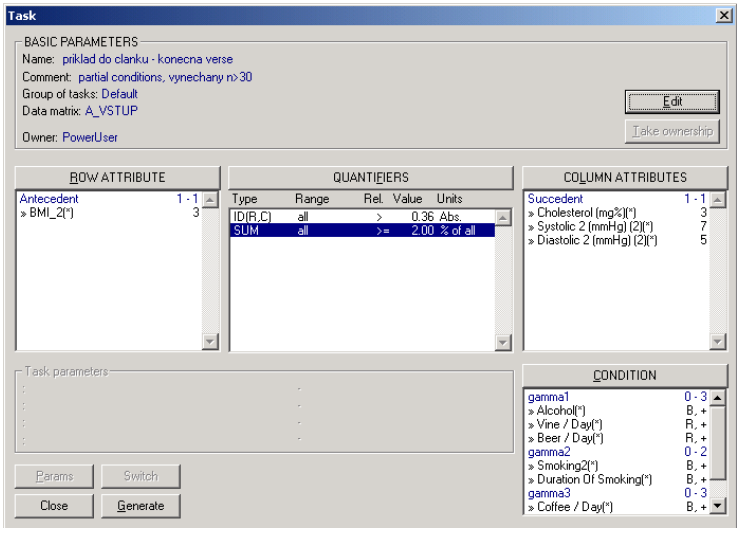

Figure 3: Task definition in the KL-Miner

…,
$\mathcal{S}(Y(y_1, y_3)) = \mathcal{S}(Y(y_1))\dot{\vee}Y[y_3]$
etc.

The resulting algorithm is very fast. Some further optimizations are used, their description is but out of the range of this paper. Results of some experiments are in section 5.2.

5 KL-Miner Application Example

5.1 Mining in the STULONG data set

We shall present an application example concerning the STULONG data set (see [7]). The data set consists of several data matrices, comprising data from cardiology research. We will work with data matrix called ENTRY. The matrix results from observing 219 attributes on 1 419 middle-aged men upon their entry examination.

Our example task concerns relation between patients' body mass index (BMI) and patients' level of cholesterol or blood pressure (diastolic or systolic), conditioned by patients' vices (smoking, alcohol consumption, etc.). Definition of the task in KL-Miner's GUI is shown in Fig. 3.

The set RQ of relevant questions is given by:

- a set of row attributes: $\mathcal{R} = \{\texttt{BMI}\}$,
- a set of column attributes:
  $\mathcal{C} = \{\texttt{diast\_bp}\,, \texttt{syst\_bp}\,, \texttt{cholesterol}\}$,
- a specification of KL-quantifier (see sect. 3): we set conditions $ID > 0.36$ & $\sum_{k,l} n_{k,l} \geq 0.02 * m$ (m is number of rows of the ENTRY data matrix),
- definition of the set Γ of relevant conditions, see below.

Recall (see sect. 2) that each relevant condition is a conjunction of partial conditions, each partial condition being a conjunctions of literals. In our task, we have defined the following three sets of partial conditions:

- set Γ_1 of conjunctions of lenght $0, \ldots, 3$ of literals defined in Tab. 2,
- set Γ_2 of conjunctions of lenght $0, \ldots, 2$ of literals defined in Tab. 3,
- set Γ_3 of conjunctions of lenght $0, \ldots, 3$ of literals defined in Tab. 4.

Only positive literals were allowed. The set Γ of all relevant conditions is defined as

$$\Gamma = \{\gamma_1 \wedge \gamma_2 \wedge \gamma_3 | \gamma_1 \in \Gamma_1, \gamma_2 \in \Gamma_2, \gamma_3 \in \Gamma_3\}\,.$$

Let us give a brief account of cardinalities of the above defined sets: $|\Gamma_1| = 49$, $|\Gamma_2| = 64$, $|\Gamma_3| = 288$, $|\Gamma| = |\Gamma_1| \times |\Gamma_2| \times |\Gamma_3| = 903\,168$, $|RQ| = |\mathcal{R}| \times |\mathcal{C}| \times |\Gamma| = 2\,709\,504$. (See Tab. 5 to confirm these numbers). To solve the task, KL-Miner searched the set RQ and found one relevant truth. Due to optimizations in the algorithm, only $78\,717$ contingency tables were actually evaluated.

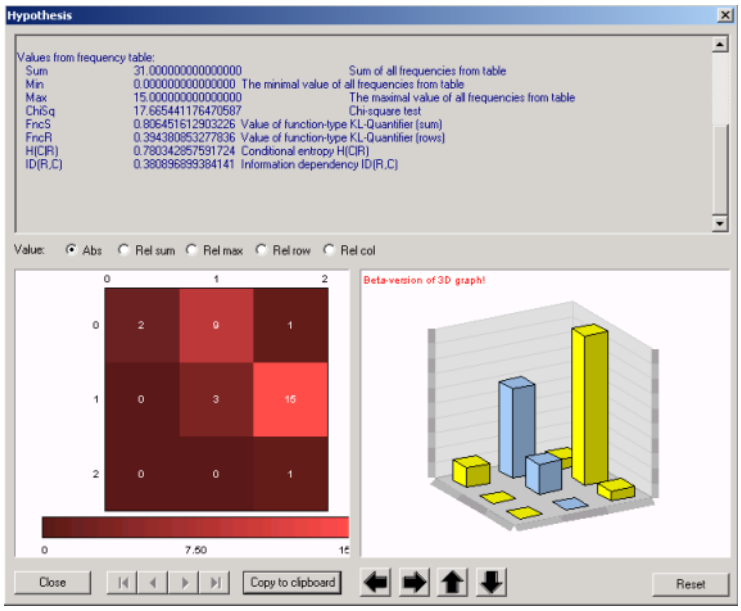

Figure 4: Hypothesis output in the KL-Miner

Attribute	coef. type	coef. lenght	Basic?
`alcohol`	subset	1	yes
`vine/day`	subset	1	no
`beer/day`	subset	1	no

Table 2: Literals for the Γ_1 set of partial conditions

Attribute	coef. type	coef. lenght	Basic?
`smoking`	interval	$1, 2$	yes
`sm_duration`	interval	$1, 2$	yes

Table 3: Literals for the Γ_2 set of partial conditions

Attribute	coef. type	coef. lenght	Basic?
`coffee/day`	subset	1	yes
`tea/day`	interval	$1, 2$	yes
`sugar/day`	interval	$1, 2$	yes

Table 4: Literals for the Γ_3 set of partial conditions

Attribute	No. of admissible values
`alcohol`	3
`vine/day`	3
`beer/day`	3
`smoking`	4
`sm_duration`	4
`coffee/day`	3
`tea/day`	3
`sugar/day`	6

Table 5: Numbers of admissible values of attributes

Execution of the task took 8 seconds, see Tab. 6. The true hypothesis is

$$\texttt{BMI} \sim \texttt{cholesterol}/\gamma^* \,,$$

here γ^* is conjunction of the following literals:

- $\texttt{tea/day}(\texttt{0 cups})$
- $\texttt{smoking}(> \texttt{14 cigarettes/day})$.
- $\texttt{sugar/day}(1 - \texttt{4 lumps})$
- $\texttt{coffee/day}(1 - \texttt{2 cups})$
- $\texttt{alcohol}(\texttt{occasionally})$

This means that strength of the correlation between attributes `BMI` and `cholesterol` measured by ID exceeds 0.36 among those observed patients, who satisfy the condition γ^* (i.e. they do not drink tea, smoke more than 14 cigarettes a day, etc.). Furthermore, the number of patients who satisfy γ^* exceeds $0.02 * m$. The output of this hypothesis in the procedure KL-Miner is shown in Fig. 4. To further examine the dependence, we used the 4ft-Miner procedure (see sect. 2) and found association rules

$$\texttt{BMI}(< 25, 30)) \rightarrow \texttt{cholesterol}(< 260, 530 >)/\gamma^* ,$$

with $confidence = 0.83$ and $support = 0.48$, and

$$\texttt{BMI}(< 15, 25)) \rightarrow \texttt{cholesterol}(< 200, 260))/\gamma^* ,$$

with $confidence = 0.75$ and $support = 0.29$. Here $confidence$ and $support$ are computed w.r.t. the data matrix ENTRY$/\gamma^*$

5.2 An Example on Scalability

We have conducted some preliminary experiments concerning scalability. We have run the task from the preceding section on the ENTRY data matrix magnified by the factor of 10, 20, etc.. Please recall that execution of this particular task consists of generation and evaluation of more than 75 thousands of contingency tables. The experiments were conducted on PC with Pentium IV on 1600MHz, with 512 MB of operational memory. See Tab. 6 for results. It appears that the execution time is almost linear in the number of rows (see the "Difference" column); however, a more detailed examination of computational properties of KL-Miner's core algorithms is yet to be done.

Factor	No. of rows	Exec. time	Difference
1	1419	8 sec.	–
10	14190	30 sec.	22 sec.
20	28380	65 sec.	35 sec.
30	42570	101 sec.	36 sec.
40	56760	142 sec.	41 sec.
50	70950	180 sec.	38 sec.

Table 6: An example on scalability

6 Conclusions and Further Work

We have presented a new data mining procedure called KL–Miner. Purpose of the procedure is to mine for patterns based on evaluation of two–dimensional contingency tables. To evaluate a contingency table, combinations of several functions can be used – these functions range from simple conditions on frequencies to information theoretic measures.

We have also outlined principles of the bit string technique used to optimize the mining algorithm. Application of the procedure was shown on a simple example.

As for the further work, we plan namely to apply the procedure KL-Miner to further data sets and to implement new interestingness measures (τ_b and I_m, see section 3).

Acknowledgement: The work described here has been supported by project LN00B107 of the Ministry of Education of the Czech Republic and by project IGA 23/03 of University of Economics Prague.

References

[1] P. Hájek and T. Havránek, *Mechanising Hypothesis Formation - Mathematical Foundations for a General Theory*, Springer-Verlag: Berlin - Heidelberg - New York, 1978.

[2] P. Hájek, T. Havránek and M. Chytil, *The GUHA Method* (in Czech), Academia: Prague, Czechoslovakia, 1983.

[3] J. Rauch, "Some Remarks on Computer Realisations of GUHA Procedures". International Journal of Man-Machine Studies, vol. 10, pp. 23–28, January 1978.

[4] J. Rauch and M. Šimůnek, "Alternative Approach to Mining Association Rules," in Proc. ICDM02 Workshop The Foundation of Data Mining and Knowledge Discovery, Maebashi, Japan, 2002, pp. 157–162.

[5] J. Řehák and B. Řeháková, *Analysis of Categorized Data in Sociology* (in Czech), Academia: Prague, Czechoslovakia, 1986.

[6] M. Šimůnek, "Academic KDD Project LISp-Miner" in Abraham A., and all (Eds.), Advances in Soft Computing - Intelligent Systems Design and Applications, Springer, Tulsa, Oklahoma, pp. 263-272, 2003

[7] M. Tomečková, J. Rauch and P. Berka, "STULONG - Data from Longitudinal Study of Atherosclerosis Risk Factors", in Proc. ECML/PKDD02 Discovery Challenge, edited by P. Berka, Helsinky, Finland, 2002.

[8] R. Zembowicz and J. Zytkow, "From Contingency Tables to Various Forms of Knowledge in Databases," in Advances in Knowledge Discovery and Data Mining, edited by U. M. Fayyad and all, AAAI Press: Menlo Park, California, pp. 329 - 349, 1996