FINANCIAL DATA CHALLENGE

Petr Mikšovský, Filip Železný, Olga Štěpánková, Michal Pěchouček

{miksovsp, zelezny, step, pechouc}@labe.felk.cvut.cz

The Gerstner Laboratory for Decision Making and Control
Department of Cybernetics, Faculty of Electrical Engineering
Czech Technical University
Technická 2, CZ 166 27, Prague 6

1 INTRODUCTION

"Once upon a time, there was a bank offering services to private persons. The services include managing of accounts, offering loans, etc. The bank wants to improve their services…." [1]. Initially, if a bank wants to improve their services they need to understand their internal processes. It was the main motivation of two problems described in this paper.

The first problem is oriented towards understanding relationships among bank affiliated branches (i.e. search for relational dependency between increase/decrease of average month-balance of branches in different regions). Here, we were trying to find clusters of regions exhibiting "similar" behaviour or which differ from the average.

The second problem is to identify indications of successful or unsuccessful loan before it is created. This problem seems to be very important because the data shows that about 11 percent of loans are unsuccessful. Unsuccessful loan is such that it was not paid or client was in debt (so he paid with problems).

2 DATA PRE-PROCESSING

Since the original dataset was distributed in plain text format any analysis had to be started by conversion into a database form. We have decided for Borland Paradox table as this table can be quickly and easily queried by common database techniques. Converted data needs to be checked for consistency and preliminary explored using SQL.

Further pre-processing is aimed at obtaining characteristic account parameters like minimal, maximal, and average balance etc. These parameters can be calculated from table *transactions*, which contains balances after each transaction. But for calculation of valid average balances it is necessary to fill in balances for those days where no transaction had been performed. It means that from the original dataset, containing about million transactions, we select about 800 thousand day-balances. Within the next step we append balances for the rest of days with respect to date where an account was created. This action increases the number of records representing day-balances to about 5.6 millions and consumes more than 6 hours[1].

These steps represent general pre-processing necessary for the future analysis, further data fitting continues with respect to the problems to be solved and will be described together with them.

[1] Hardware equipment: PIII – 450MHz, RAM 128MB, HDD 8GB

3 FEATURES OF THE DATASET

During the pre-processing and exploration phases several interesting or unusual features of the dataset were found. Let us point to the most important ones:

- Considering that the client is identified by his *client_id*, one account can be accessed by one or two clients but one client can access only one account.
- Each outgoing transaction contains code of the target bank and target account. Analysing amount of target accounts per one source account could identify interesting accounts, e.g. account of revenue authority, insurance company, employees of the same company, etc. We search for those accounts, which were obtaining money from many different sources. They correspond to nodes with high degree in a graph depicting transactions between all accounts. Surprisingly, this graph has low degree. In most cases money is transferred only between two accounts, i.e. one source account receives/sends money from/to one target account. That is why none of the expected relations (specified above) could be identified.
- Loans are paid to different banks (derived from attribute *bank* in table *transactions*). Besides others, this is the case of the loan 6863, which is paid to bank GH, on the other hand loan 5895 is paid to bank KL.
- Loan's status is *A*, i.e. contract finished, no problems [1], but total amount paid (calculated from transactions) differs from the amount loaned. In some cases clients paid several percent more but in two cases clients paid less (loan 5161 – 12% and loan 6995 – 25%). We took to account differences greater than 1%.
- Table *transactions* contains records described as "SANKC. UROK" (i.e. sanction interest if negative balance [1]). There are 11 accounts, which paid this sanction interest, though they did not have negative balance (before sanction interest was paid). These accounts are: 1012, 1498, 2099, 2572, 3953, 7445, 9337, 9814, 10694, 10788, and 11123.

All these surprising features have been uncovered using SQL and other standard database techniques.

4 PROBLEM 1: INTERDEPENDENCIES AMONG BANK AFFILIATED BRANCHES

Let us assume that in the beginning there is a network of affiliated branches with unknown internal dependencies in unknown area. First of all we need to check if there is any influence of balance decrease in one district to the others and describe course of overall balance in process of time.

4.1 DATA PRE-PROCESSING

The problem requires to pre-calculate monthly-based parameters, we expect as important, for each district. Such parameters are minimal, maximal, total, and average balance per moth.

4.2 ANALYSIS

ILP Analysis

In the beginning we have expected more complicated chain-like dependencies. That is why data was prepared so that PROGOL system [3] can be applied. PROGOL is the ILP-based (Inductive Logic Programming) tool for induction of rules, in the form of Prolog program, which allow to distinguish between elements from the set positive examples and those from the set of negative examples. It also allows using domain knowledge that could be helpful for a solution. This information is called *background knowledge*.

In this case we suppose that one district (call it *dist1*) roughly follows the trends (increases/decreases) of another one (*dist2*) with a certain delay, say two months. The resulting hypothesis might acquire a very simple form:

```
increase(dist1, M1) :- increase(dist2, M2), month_ago(m1, N), month_ago(n, M2).
decrease(dist1, M1) :- decrease(dist2, M2), month_ago(m1, N), month_ago(n, M2).
```

To get to this result, we need to define the predicate *month_ago*, e.g. as

```
month_ago(X, Y) :- Y is X-1.
```

where month is represented by ordinal number. This definition is presented as background knowledge. Consequently, we had to adapt input examples into the following form *increase(name_of_district, month)* and *decrease(name_of_district, month)*. We define decreasing balance examples as follows: - l*arge_decrease(name_of_district, month)* for 20-30% decreases and *small_decrease(name_of_district, month)* for 10-20% decreases. Percentage values are calculated with respect to the previous month.

Analysis of the data took about 4 hours using SUN workstation. The resulting rules identified 3 clusters of districts. Two of them seemed to follow similar patterns, namely districts 1, 8, 10, 11, 13, 25, 27,42,62, 69, 77 and 2, 14, 37, 43, 48, 50, 71, 75. Each of both clusters can be represented by its first member. An attempt to interpret these results was based on graphical representation. We expected to see some differences between behaviour of both representatives of clusters. Surprisingly, this visualisation did not prove existence of any significant difference between behaviour of both considered regions. The main reason for that can be that there is no strong dependency among regions.

On the other hand, simple visualisation of the created data seems to be able to provide interesting view on time development of district balances.

Visualisation Analysis

We prepare dataset containing monthly total and average balance for each district and display appropriate graphs (using Microsoft Excel).

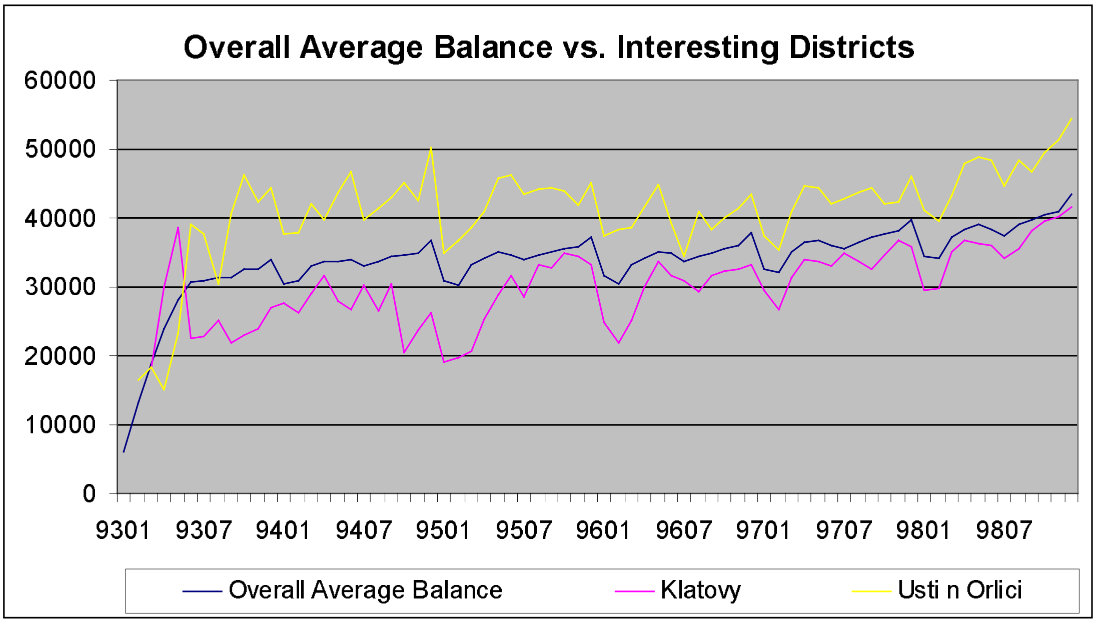

Figure 1

The output graphs confirm that there is not evident dependency among regions and also demonstrate that course of monthly total and average balance in time looks very similar. There appear periodic substantial balance increases and decreases. Except first year, large decreases of balance appear every January. Every year there is a small decrease of balance between larger ones. It emerges every June/July (see Figure 1). Visualisation comparing courses of district values to the average easily highlighted rich and indigent regions, from the bank point of view.

As it shown on the Figure 1, every large decrease o balance is anticipated by a temporary increase. It seems that accounts' owners expect higher spending and cumulate money in advance.

5 PROBLEM 2: LOAN PREDICTION

This problem is rooted in a practical task of predicting successful and unsuccessful loans. We consider that the key parameters for this analysis should be derived from "behaviour" of the individual accounts.

As a training sub-task we look for a description of loan classes. There are 4 types of loan status: *A* if contract has been finished without problems, *B* if contract finished but loan not paid, *C* for running contract, OK so far, and *D* for running contract, where client is in debt. The data analysis is performed using account parameters evaluated 6 month back from the last part payment.

The second sub-task is to predict unsuccessful loans. The unsuccessful loan is a loan, status of which is either *B* or *D*. In this case, the data analysis is carried out using account parameters evaluated 6 month back from the month where loan has been created.

5.1 DATA PRE-PROCESSING

The both sub-tasks described above require similar pre-processing. In the beginning we calculate parameters for each account. The parameters are minimal, maximal, and average amount paid/received in a moth and minimal, maximal, and average balance in a moth.

Within this step, calculating of parameters is restricted to start since month where the account was opened. During the next step we formed the dataset for analysis. Using SQL-based tool (programmed in Delphi) we gather a table containing loan description and account's parameters. While in case of the first sub-task we use 6-month back history since the last payment, in the second case we use account's parameters 6-month back before the loan was created. Furthermore, we add a logical attribute describing whether several clients can access the account or not. The result from the pre-processing is a table containing 682 records that describe all loans.

5.2 ANALYSIS

Attributes introduced in the previous paragraph are mostly numerical ones. Therefore the most natural analysis is applying program C5.0 [2], which can easily handle numeric values without precedent discretisation.

Description of loan classes

We first try to use C5.0 but it does not result in any understandable description of loan classes therefore we perform visualisation (see Figure 2). The visualisation is realised using Microsoft Excel.

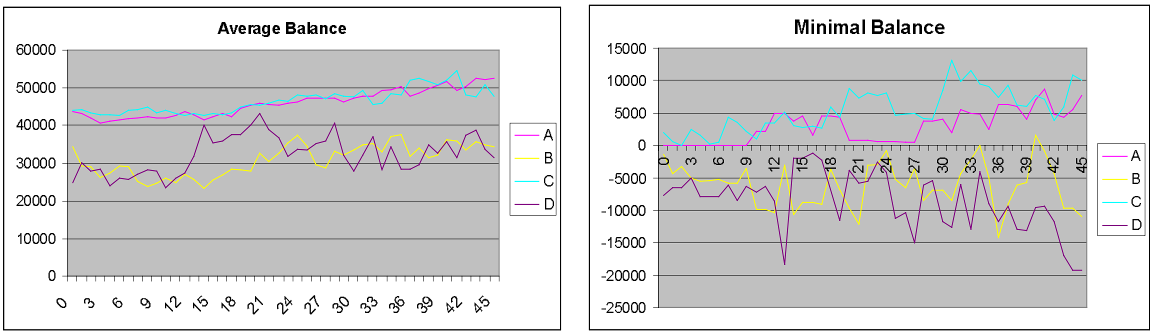

Figure 2. Average and minimal balance for all accounts with loans during the time (month 0 means the month where a loan was created)

As it is shown on the figure it is natural to distinguish only between two classes instead of four. If we know the history of account balances it is not difficult to derive transparent rules for a loan class (status). These rules are e.g.

- class of a loan will be *A* or *B* if the average balance of the corresponding account is higher than 40000 see Figure 2a (error rate 8.4%), or
- class of a loan will be *A* or *B* whenever the minimal balance of the corresponding account is ever non-negative see Figure 2b (error rate 2.1%).

Prediction of unsuccessful loans

In this case we assume that a client who applies for a loan should satisfy some parameters of his account. We consider last 6 month before a loan was created to allow detection of possibly risky loans.

Besides account and loan parameters we also append demographic data and applied C5.0 algorithm. Originally, the dataset consists of 69 attributes and 682 cases. With respect to 10 times more examples of successful loans we have to define misclassification costs. This option helps to compensate this very unbalanced dataset. Result of applying C5.0 was a tree, the upper nodes of which were denoted by attributes *MULTIACC, IB0, IB2, IB6, SA1, SA3* most often. Where attribute *MULTIACC* means that an account can be accessed by several persons (multiple records in table *disp* [1]). Parameters *IBx* and *SAx* represent minimal balance and sum of amounts (incoming minus outgoing) in month *-X* before a loan was created.

Finally, C5.0 analysis of the reduced dataset using only the chosen attributes results in the following set of rules:

```
Rule 1: (cover 225)
  SA1 > -27521, IB2 > 20851, IB6 > 25061                         ->  class A
Rule 2: (cover 242)
  SA1 > -11765, IB2 > 14757, IB6 > 25061                         ->  class A
Rule 3: (cover 160)
  IB0 <= 35385, IB2 > 14757, SA3 > -17286, IB6 > 25061           ->  class A
Rule 4: (cover 145)
  MULTIACC = Y                                                   ->  class A
Rule 5: (cover 130)
  IB0 > 16242, IB2 > 14757, SA3 > -2436, SA3 <= 16680, IB6 <= 25061   ->  class A
Rule 6: (cover 98)
  IB0 > 44495, IB2 > 14757                                       ->  class A
Rule 7: (cover 68)
  IB0 > 7888, IB2 > 14757, IB6 > 6268, IB6 <= 19287              ->  class A
Rule 8: (cover 43)
  IB2 > 14757, SA3 > -8764, SA3 <= -3824, IB6 > 6268             ->  class A
Rule 9: (cover 41)
  IB0 > 7888, IB0 <= 42783, IB2 > 32529, SA3 > 17128             ->  class A
Rule 10: (cover 28)
  IB0 > 5250, SA1 <= 2378, IB2 > 600, IB2 <= 14757, SA3 > -25633, IB6 > 7789,
  IB6 <= 48681                                                   ->  class A
```

```
Rule 11: (cover 21)
  IB0 > 7888, IB0 <= 22550, IB2 > 14757, SA3 <= -2436, IB6 <= 25061   ->  class A
Rule 12: (cover 15)
  IB0 > 32059, IB2 <= 32529, SA3 > 16680, SA3 <=31978,                ->  class A
Rule 13: (cover 10)
  MULTIACC = N, IB0 <= 32059, IB2 <= 32529, SA3 > 16680, IB6 <= 25061 ->  class B
Rule 14: (cover 9)
  MULTIACC = N, IB2 <= 600                                            ->  class B
Rule 15: (cover 3)
  MULTIACC = N, IB2 <= 32529, SA3 > 31978, IB6 <=25061                ->  class B
Rule 16: (cover 3)
  MULTIACC = N, SA1 > 2378, SA1 <= 3533, IB2 <= 14757, IB6 > 7789     ->  class B
Rule 17: (cover 2)
  MULTIACC = N, IB0 > 42783, IB0 <= 44495, IB2 > 32529, SA3 > 16680   ->  class B
Rule 18: (cover 34)
  MULTIACC = N, IB0 <= 7888                                           ->  class B
Rule 19: (cover 21)
  MULTIACC = N, IB2 <= 14757, IB6 <= 7789                             ->  class B
Rule 20: (cover 1)
  SA3 > 16680, SA3 <= 17128, IB6 <= 25061                             ->  class B
Rule 21: (cover 1)
  MULTIACC = N, IB0 > 35385, SA1 <= -27521, IB6 >25061                ->  class B
Rule 22: (cover 16)
  IB2 <= 14757, SA3 <= -25633                                         ->  class B
Rule 23: (cover 15)
  MULTIACC = N, SA1 <= -11765, IB2 <= 20851                           ->  class B
Rule 24: (cover 25)
  MULTIACC = N, IB0 > 22550, IB0 <= 44495, SA3 <= -2436, IB6 <= 25061 ->  class B

Default class: B
```

Evaluation of the rules on data (682 examples) results in misclassification of 52 examples of class *A* (from 606 examples). All 76 examples of class *B* have been classified correctly. Consequently, classification accuracy is 7.6%.

6 CONCLUSION

During the pre-processing and exploring phases we discovered several suspicious features of the dataset, the most interesting of which we present in the paper. These features correspond to the issues we solved or consider to solve.

The analysis has been performed using ILP tool (PROGOL) as well as ID3-based tool (C5.0). In general, the dataset is a typical example of the practical dataset where there are no strong and easy-to-find dependencies without deep expert domain knowledge. It was probably the main reason why as the most successful technique appeared to be visualisation. Visualisation helped to find several interesting results, e.g. periodical decreases of balance in the bank or simple rule that identifies loan status.

Within construction of prediction rules we discovered very interesting attribute *MULTIACC*, that describes whether an account can be accessed by several people or by the owner only. Surprisingly, there is a substantial dependency between this attribute and decision if a loan will be successful.

Finally, we have to mention that there is a lot of attributes, which can be potentially interesting for an analysis but there are only 682 records describing loans. This amount of data is not sufficient for larger number of attributes.

7 REFERENCES

[1] PKDD'99 Discovery Challenge: http://lisp.vse.cz/pkdd99/

[2] C5.0: http://www.rulequest.com

[3] PROGOL: http://www.comlab.ox.ac.uk/oucl/groups/machlearn/PProgol

[4] Železný F: ILP experiments for PKDD99 Data Challenge. Research Report GLM-17/99

[5] Kouba Z., Matoušek K.: Data Warehousing with Graphical Models. In proc. 3rd IEEE Int. Conference On Intelligent Engineering Systems, INES'99, Stará Lesná, Slovakia 1999