Discovery Challenge, Financial Data

Wim Pijls

Erasmus University Rotterdam, Faculty of Econmics.
P.O.Box 1738, 3000 DR Rotterdam,
The Netherlands. e-mail: *pijls@few.eur.nl*

1 Introduction

As a staff member of a Faculty of Economics, I have concentrated on the set of financial data containing data of a Czech bank. My only goal was to find something which might be interesting to the managers of the bank. I proposed no specific goal in advance.

2 How to handle the data set?

The data set comprises eight tables each of which is in ascii-format. There is one table with a huge size: 67Mb. It turned out that an ascii-file of this size is not manageable for any Windows tool. It is true that this file can be loaded into the Word-97 program, but any non-trivial action fails. For instance, replacing Czech words with my own short codes was not possible. The vi-editor under Unix was unable as well to handle this file. My conclusion was that visual or interactive tools are not appropriate to deal with such a huge size. Therefore, I have decided to do everything in command mode. Dealing with other sets in the past, my experience with classical Unix tools like *sed* and *awk* etc. was beneficial. So, everything I have done with the financial data set is performed using just the Unix tools: *wc grep, sed, awk, lex, yacc, sort, uniq, join.* (The Gnu versions of *lex* and *yacc* are called *flex* and *bison* respectively.) Furthermore, *cmp* and *diff* were used for checking some results. For counting frequency, *grep* and *awk* are very convenient. An excellent guide (apart from the manuals) for the aforementioned tools is [1]. Although *perl* is an even more powerful language in this domain, there was no need to utilize this language.

3 Preprocessing steps

The necessary initial task was to reduce the large file size (67 Mb) of the relation *trans.asc*. To that end the following preprocessing steps were executed.

- Czech expressions like "VKLAD", "PREVOD Z UCTU", etc. were replaced by a one-character code in lower case. Since Unix tools are case-sensitive, there will not be any confusion with the two-capital code in the field containing the partner bank.
- Empty strings and strings consisting of one space in the *operation* or the *k_symbol* field were also replaced by a one-character code.

- The *trans_id* field was removed

- Using *yacc* (originally designed as a parser generator), it can be checked whether the structure of each record complies with the structure described in the text accompanying the data. There were two discrepancies. First, if the *operation* field contains the term VYBER, sometimes the *type* field contains this term as well. Second, some records containing VYDAJ and VYBER as type and operation respectively have an extra string "0" at the end of the line.

- The *type* field is functionally dependent on the *operation* field and is redundant therefore; this field is removed.

- The records which have another bank (denoted by two capitals) included were examined. It appeared that the occurrence of bank codes of distributed uniformly. Each bank code has a frequency of about 20.000, which can be checked easily using *grep* and *wc*. Hence the bank code itself is not relevant. Therefore, the code of the bank of the partner was discarded, along with the account of the partner.

- The date field was split up into three separate fields for year, month and day respectively.

The above operations on the file *trans.asc* result into a new file with seven fields: account, year month day, amount, balance, and a two-letter code representing the values in the fields *operation* and *k_symbol* respectively. The size of the new file is about 28 Mb, approximately 40% of the original size. Of course, the number for records is unchanged (1.056.320). The new file in zip-format can be found on [2].
The set of two-letter combinations occurring in the last field has 14 elements. It is curious that SLUZBY, standing for payment for statement, and sanction interest are always combined with VYBER (withdrawal in cash) as operation.

The other files have a small size and do not need any reduction (apart from replacing Czech words by English-like mnemonics).

4 Results

First of all we have investigated thoroughly the new version of *trans.asc*. The results are presented in this section. Second we also examined some features of the other files.

The transactions file, the operation and the k_symbol.
Focusing on the fields *operation* and *k_symbol* we noticed the following points.

- The most frequent mode in the file of transactions is VYBER (withdrawal in cash). About 40% (434918 records) of the transactions has this term mentioned in the mode field. Among the VYBER transactions, a large majority has an empty string or a string of just one space in the k_symbol field. Among the VYBER transactions, the most frequent k_symbol string is SLUZBY (payment for statement).

- The most frequent combinations for the fields *operation*/*k_symbol* are:
VYBER (withdrawal in cash) with an empty k_symbol, 274675 records;

UROK (interest credited), 183114 records; this value as k_symbol has only empty values as operation; conversely, only if interest is credited, the operation field is empty;
VKLAD (credit in cash), 156743 records; this operation has only empty values as k_symbol;
VYBER in combination with SLUZBY (payment for statement), 155832 records.

The above combinations largely exceed the frequencies of other combinations.

- The number of remittances to another bank largely surpasses the number of collections from another bank.

- The number of accounts is exactly 4500.

- Every account has at least one record including a VKLAD operation (credit in cash). As mentioned above, VKLAD as operation is always combined with an empty k_symbol. There are only 1606 accounts which have collections from another bank. We conclude that most accounts have their contributions in cash.

- Every account has at least one record including a withdrawal in cash combined with an empty k_symbol. There are 3602 accounts which have remittances to another bank. Comparing this number with the number of accounts with any collection from another bank (1606), we see again that, in the interactions to other banks, outgoing transactions outnumber incoming transactions.

- The utilization of credit cards is poor.

The conclusion is that cash operations (credit and withdrawal in cash) and automatic operations (interest and payment for statement) are by far most frequent.

The transaction file with time information.
We now present some information in relations to dates.

- About 30% of the transactions is executed on the 30th of 31th of a month. This is due to the interest which is credited only on the last day of a month.

- Another frequent action on the last day of the month is withdrawal in cash combined with the value SLUZBY (payment for statement) in the k_symbol field. Like crediting interest, this action is only performed on the last day of a month.

- Remittances to other banks take place only on the 5th, the 6th, ..., through the 14th of each month. On these days, the number of credits in cash is significantly higher.

- Loan payments (UVER as k_symbol) are only executed on the 12th of each month. The value UVER in the k_symbol only occurs in combination with the terms PREVOD NA UCET (remittance to another bank) in the operation field.

- The number of remittances to other banks is increasing during each year. The summit is in December.

- January exhibits a significantly higher number of withdrawals in cash than any other months does (74484 against 43641, the second highest). The frequencies of the related k_symbols show the same pattern as in other months.

- The number of credits in cash each month is distributed virtually uniformly.

The average balance of an account.
We have computed the average balance for each account. Based upon the value of the average balance we have divided the set of 4500 accounts into 9 classes each of which comprises 500 accounts. Class no 9 contains the top 500 accounts, class no 8 contains the second 500 accounts, etc. We did not found any correlation between the class number and the values in the *operation* or *k_symbol* field.
This classification was also used while investigating other tables. There was no correlation between the class number of an account and its frequency in the table *order.asc*. Neither did we find a correlation between the class number of an account and the average salary in the district of the owner. We discovered that the rate of credit cards is significantly higher as the class number is higher.

Other tables.
We joined the tables *account, client* and *disposition*. There was no striking difference between the genders. The year of birth is distributed uniformly in the range 1940 to 1980. Put another way, every value in this range has the same frequency as year of birth.
Most loans were OK in table *loan* according to the *status* field. There are only 892 accounts connected to a credit card.

5 Concluding remarks

Everything bank managers might wish to know is gathered quickly (almost instantly) using the aforementioned Unix tools. When using the transformed file, the file sizes are no longer problematic. Most tasks are executed in only a few minutes on a Sun Sparcstation 10. If desired, some statistical computations (for instance correlation and regression analysis) can be executed subsequently.
Recently a new algorithm for mining frequent item sets was released. See [3] which is available from [2]. This algorithm is suited to just boolean data sets. It is being extended to quantitative (numerical) data sets. After completion of the extended version, I hope to apply the new algorithm to the financial data set, which is still challenging me for discoveries.

References

[1] Brian W. Kernighan, Rob Pike, *The UNIX Programming Environment*, Prentice Hall, 1984.

[2] Wim Pijls' Homepage, www.few.eur.nl/few/people/pijls/

[3] Wim Pijls and Jan C. Bioch , *Mining frequent itemsets in memory-resident databases.* Accepted for Netherlands/Belgium Conference of Artificial Intelligence 1999.