Data mining techniques in the experimental analysis of dependability

András PATARICZA - Béla TOLVAJ
Budapest University of Technology and Economic Sciences
Department of Measurement and Information Systems

1 Introduction

The experimental analysis of failure log data is an important part of fault tolerance validation measures and/or evaluation of dependability characteristics is. These experiments can be carried out either during the normal operation of a system or in a combination with fault injections in order to increase the frequency of fault occurrences.

In the evaluation of such experiments performed on complex systems crucial difficulties result from the large number of patterns to be recorded in experiments and/or in ob servations,

- partially in order to have statistically meaningful estimates from a sufficiently large and representative failure pattern set,
- moreover due the typically highly sequential nature of the objects under test (like composite hw/sw systems) a sufficiently long period of system activity must be recorded.

Similarly the proper understanding of the error propagation mechanisms necessitates the recording of as much observable factors (signals and data values), as possible. Frequently the experimentation in the form of a post mortem analysis or fault injection experiment aim to reveal the correlation between some potential faults and the failure triggered by them. Note, that pattern sensitive failures are the most typical ones in software based systems.

Therefore the analysis of complex systems must cope with a large population of system logs, where each individual pattern is large both in the terms of temporal duration, and in the number of signals and values to be recorded simultaneously.

In the current paper we summarize our preliminary experiences for using data mining in the analysis phase of such error logs. The pilot experience was carried out on error logs from a Web server, however it must be pointed out, that a very similar approach based on the same technology can be useful in a variety of dependability-related fields as well, like fault injection based testing, stratified testing, etc.

Data mining was originally invented to help the analysis of large scale business records, however the underlying mathematical techniques, like

- relevance filtering (to estimate a subset of data fulfilling some user defined criteria),
- data clustering, aggregation and drilldown (merging or partitioning sets of data according of predefined criteria),
- hierarchical and automated estimation of the correlation between different factors,

- and different pattern search functions

make this technology to a favorite candidate in technology related applications as well.

2 Data mining

Recently, the concept of data mining managed to become a standard component in the toolset of every database technology provider.
This popularity originates in the widespread use of computers in business processes. These applications collect a large amount of business transaction logs. This huge amount of information contains additionally to its original objective (like to record the sales) a fair amount of "hidden knowledge" about the business process itself, that way provide potential answers to questions concerning e.g. the market trends. However, the extraction and interpretation of this knowledge can be barely performed by the traditional means due to its heavy dependence on numerous factors.

Data mining is a methodology suitable to interactively reveal hidden data correlation in large-scale data sets in an automated way. Machine learning techniques developed on the theoretical background of artificial intelligence are integrated into the commercial data mining tools, as an effective support of handling feature extraction in large data and state spaces.

Mathematically, the objective of this analysis is to derive the membership relation, defining the subset relevant from the point of view of the given evaluation objective, from the originally large scale data set given in the form of a list of elements.

The result of the mining process describes the correlation between the objects within the database in the form of various models, decision trees or logical connections, that can give answers to questions that would be difficult to be answered otherwise, like

- identification of the processes in the background;
- modelling of the hidden internal relations of events,
- quantitative of estimation of the effect of changes.

The structure of the start-off data is (not necessarily but expediently) of a data warehouse nature, providing a uniform method to store both the set of collected basic and aggregated data as well, together with their history (time stamps).

2.1 The data mining as a process

Data mining is a complex process that can be typically subdivided into the following four phases [2], all of which can be carried out interactively, without any programming of queries, opposite to the traditional database programming:

1. **Planning** specifies the objectives of the subsequent analysis. (What goals are to be reached by data mining? What are the expected results?)

2. **Preparation** aims at the reduction of the set of data to be dealt with to a manageable amount by filtering out the data fulfilling some user defined criteria from the huge start-off data pile into the working database. These criteria confine the data to be processed to the ones approximately relevant to the objective defined in Phase 1.
3. **Mining** means the actual search for relevant samples The essential steps of this phase are:
    - *Subsampling* aims at a fast and approximate estimation of the objective characteristics upon a heuristically selected subset of the records in the sample pile. This can be useful for several reasons, e.g. it can increase the performance, or in case of a redundant data source it can serve to check the relevance of the data selected in the previous step.
    - *Feature selection* is the user action specifying the features relevant from the point of view of the sample search.
    - *Selection of the algorithms* commercial data mining tools offer a variety of algorithms at the users' disposal. The selection criteria to choose the proper one best fitting the predefined objective will be described in the next chapter.
    - *Transformation:* In this step the coding of the individual attributes will be changed in order to fit to the data mining algorithms, like encoding enumerative types by integers or normalizing and/or discretizing quantitative data.
    - *Parameterization:* Many algorithms make it possible to set up parameters that affect the operation (e.g. in machine learning the trade-off between the accuracy of the model and the computational time can be tuned by the user).
    - *Analysis*, i.e. the application of the selected algorithm on the prepared data, constitutes the actual core of the data mining process. As a result of this phase, mostly huge quantities of complex sample descriptions are provided.
4. **Evaluation** of the results aims at the transformation of the sample descriptors into a mathematically equivalent, but human understandable form. The steps of this phase are:
    - *Interpretation:* Here the user or a built-in mechanism of the tool must select from the set of samples those, that are most relevant from the point of view of the original objective (post-filtering) by evaluating their statistical relevance.
    - *Displaying the results* performs the transformation of the relations identified into a visual form that is comprehensible for the human. Frequently used presenting forms are rules, decision trees as well as PROLOG programmes. Graphical forms of presentation are more easily to perceptible.

2.2 Categorization of data mining techniques

In traditional database query technologies the execution of each analysis must be preceded by a preparatory phase for the formulation of the question in a query language (e.g. SQL); moreover, there is no support of an interactive and iterative analysis process, as no mechanism supports the reuse of the results from a previous analysis in a new query. Accordingly, the main goal of the data mining technologies is to provide a high degree of automated analysis without any requirement of programming, and to support a hierarchical query process [1].

Data mining tools differ in the degree of automation (Table 1).

- OLAP (Online Analytical Processing) offers to the user flexible mechanisms to filter, aggregate and partition data, navigate in multi-dimensional data structures (drilling down and up), and visualize the results, but the knowledge extraction itself is not automated. Its basic data structure is a multidimensional cube with dimensions and measures associated. All hierarchy levels are pre-computed. For instance, in an ad-hoc query a change in the resolution of the dimensions is accelerated by predefined aggregations.
- Data miners offer a considerable level of automation for the search of relations in the sample set. Here the main task of the user is reduced to the selection of the algorithm to be used and to the proper setting of its parameters [1]. These tools completely automate the search of patterns in the database. The patterns can be:
    - *Rules*, which describes the relations between the attributes of an object;
    - *Associations* describes, which objects occur most frequently together;
    - Frequently occurring *sequences;*
    - *Clustering* aims to build groups from objects with similar properties;
    - *Classification* means ordering object to predefined classes.

The capability of handling of multiple samples in the database simultaneously allows the analysis of temporal sequences simply by taking the time stamps associated with the individual samples as a guiding dimension.

Tool categories	**Example of methods**	**Example of tools**
Query languages and report generators	QBE and SQL-front-ends	MS-Access Cognos Impromptu
Table computing	What – if analysis	MS-Excel
Multi dimensional tools	OLAP	DecisionSuite from the Information Advantage Cognos PowerPlay Synchrony
Statistic tools	Cluster-analysis Factor-analysis	IBM Intelligent Miner for Data
Induction of decision trees	ID3-algorithm	XpertRule Profiler Cognos Scenario IBM Intelligent Miner for Data
Neural nets and genetic algorithms	Neural nets	Predict, IBM Intelligent Miner for Data

Table1. Data mining problems and solutions

3 Analysis of a web server log file

The aim of the pilot experiment was to show how data mining techniques can be put into action in the field of solving diagnostic problems. The subject of the experiment was the HTTP service of a Lotus Domino server, having an internal functional and architectural complexity sufficiently large to be a representative example in revealing hidden diagnostic knowledge.

The internal architecture of Domino [3] is built around a database of documents offered to the Web by generating HTML pages dynamically. Internally, documents are composed from smaller design elements (fields, views, etc.).

The Web browser on the client side is asking for these presentation services by URL commands; that is, it sends the commands embedded in the URLs. Similarly, a failure report from the server is sent to the user by generating a failure specific page and a corresponding item is registered in the server log file.
As a result of this service invoking convention the commands and the result of their execution are observable by examining the sequence of URLs and the log file records.

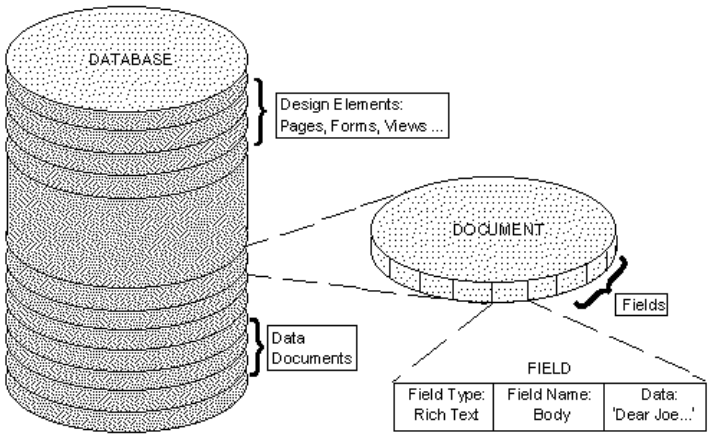

Figure 1. Architecture of a Lotus Domino database

A typical Domino URL command has the syntax of

`http://Host/Database/DominoObject?Action&Arguments,`

where **Host** is a DNS entry or an IP address, **Database is** the database in which the DominoObject resides, **DominoObject** corresponds to a Domino design element (e.g. view, document), **Action** is the desired operation on the specified DominoObject (e.g. `?OpenView`, `?OpenDocument`), while **Arguments** qualify the action, (e.g., `Count = 10` combined with the `?OpenView` action limits the number of rows displayed in a view to 10.).

The Domino log file records all server activities and contains among others the following information about each individual HTTP request:

- Date and time at the request was made
- User's identification (IP address, username for non-anonymous services)
- HTTP request sent to the server from the browser
- RFC2068 status code returned to the browser by the server upon completion of the request
- Workload parameters related to the request (processing time, length of the information sent to the client)
- Type of content accessed by the user (e.g., text/html or image/gif)

This information can be partitioned to enable more detailed analysis. This partitioning process is a kind of transformation. For instance, in this case two fields of the log records can be splitted. The first one is the content type, which has a hierarchical structure. At the first level the partitioning can happen into text and image, while the log records belonging to text can be further splitted into text/HTML and text/XML, accordingly a two level hierarchy in an OLAP cube can be built along the dimension corresponding to this field.

Similarly, the Request field describing the type of the request served by the server contains the URL command and the HTTP method (POST or GET).

After the presentation of a possible transformation of the data let's take a view of how they will be reachable with the help of data mining tools.

3.1 The estimination of failure distribution by the content type with OLAP

The example shows how a log file can be explored with the help of OLAP. The objective is the estimation of the relation between the content type and the failure codes that arose during execution. To do this the dimensions of the OLAP cube presented were selected as {error code, content type}, while the number of requests was used as a measure. This way all the remaining dimensions were neglected and all the corresponding data were aggregated into a three dimensional presentation (Fig. 2). Obviously, the most frequent failure type is "404" (file not found), and the most common content type is "text". In order to check the validity of the conclusion *"Failure 404 occurs most commonly by the text type requests"*, the subset of data having a "Failure 404" indication was filtered (Fig 3.)

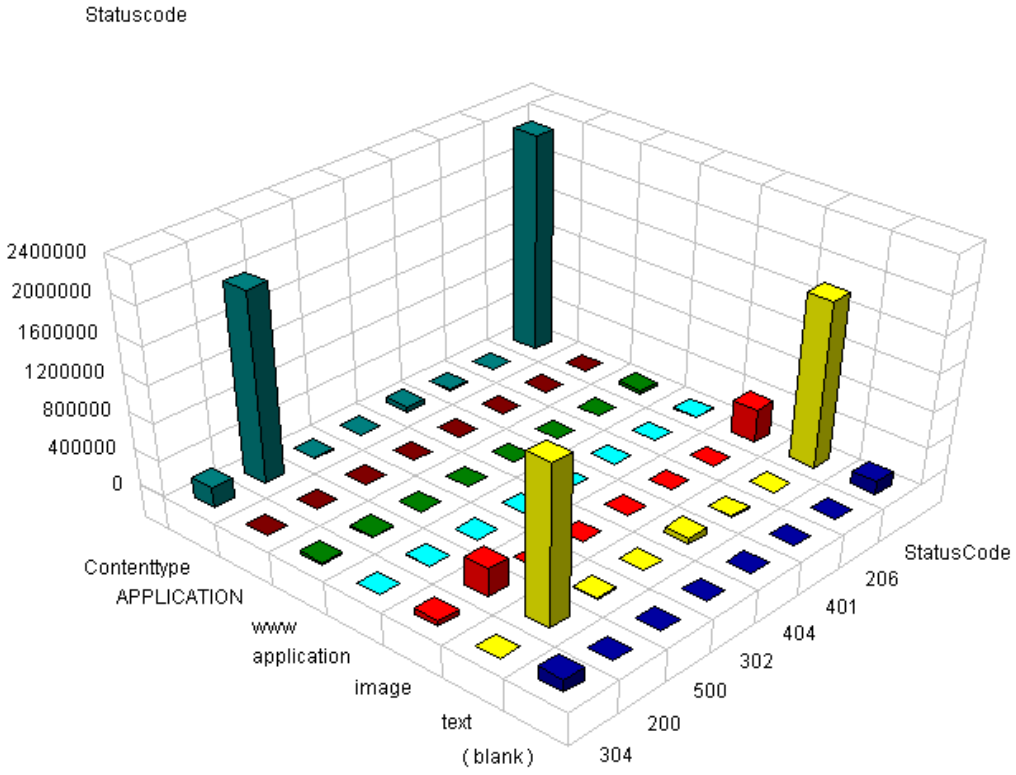

Figure 2. Error distribution by content type and error code

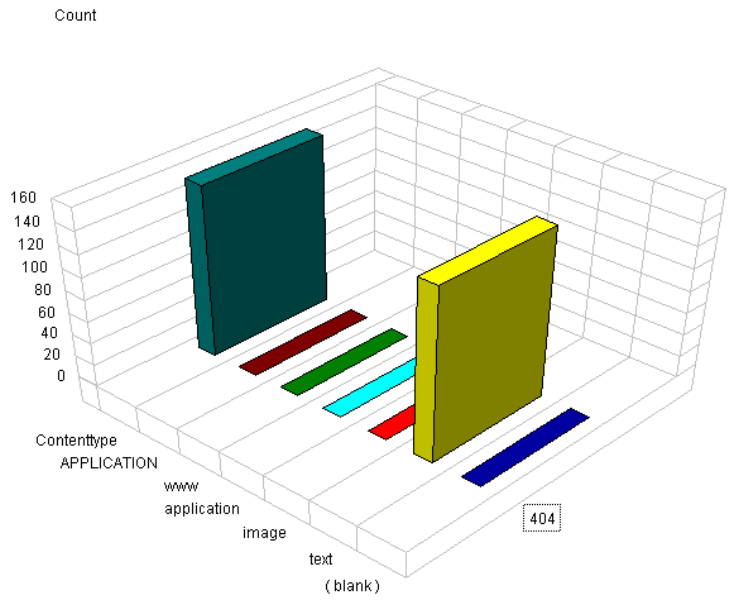

Figure 3. Error distribution by content type of failure “404”

Failure 404 occurs only in commands resulting in text files. As a proof of the more detailed hypothesis *“404 failures originate in operator faults (incorrect URL typed in)”*, a drill-down to the content type of the text requests shows the proportion of XML and HTML requests within the “404” failures (Figure 4). The validity of the hypothesis appears very probable, as failures in HTML requests dominate over XML requests, that are not issued by the user.

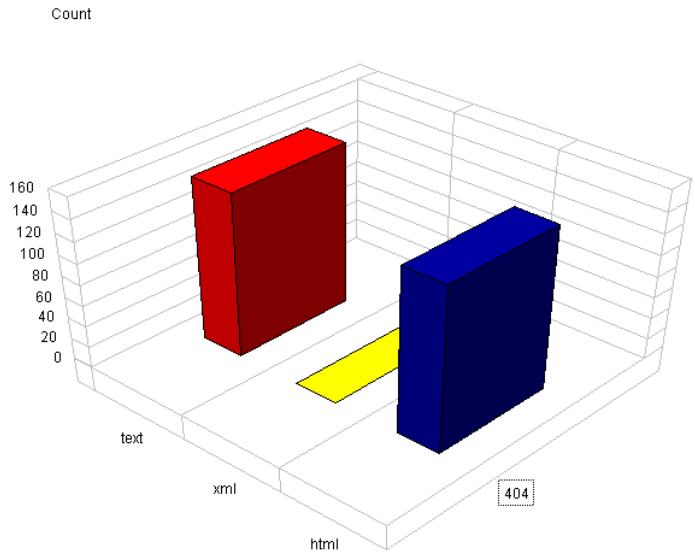

Figure 4. Error distribution by content type of text wihin failure “404”

3.1.1 OLAP summary

The main strength of OLAP according to the pilot study lies in its flexibility and productivity. The use of multi-layered OLAP data structures allows a high level analysis output for the effective processing of a very large set

of data. This way OLAP is an attractive candidate for interactive and adaptive diagnostics. However, OLAP does not support the analysis of complex temporal sequences of records.

3.2 Use of a Data Miner for log file analysis

The second example illustrates a problem that can be solved by the more intelligent data miners. As it has been already mentioned before, these tools support pattern identification additionally to the traditional statistical analysis. The result of these gives practically the stochastic temporal relations that have been outlined in the introduction.

3.2.1 Example for the search for the frequently occurring sequences

The typical failure code sequences generated as internal server errors during a user session will be estimated in this example.

The dynamic part of the Domino web client applications deals usually with posting CGI-based forms to the server. The server typically reacts by sending an URL redirection in the form of a single URL to the browser in order to force the client to open a new design element (e.g. to show the entire list of documents in a view after saving a document). In this case the content is only a URL to be opened by the browser, and the next design element referred by this URL to be opened can even start a server-side program. This way an entire chain of server/client interaction can be triggered by an initial client action.

In the server it is logged as failure 500 when a failure occurs during this process, and the URL redirections as failure 302. Unfortunately, the successful completion is not logged. Thus, the original initiator of the redirection chain has to be estimated in a sequence of URL redirections terminating with a server failure code.

All irrelevant failure codes (like those signaling a correct transaction) were omitted by subsampling in the second phase.

The necessary algorithm to solve this problem can be found under the name of “sequential patterns” in the selected tool [4], which needs the definition of the following input fields:

- **Transaction group** field

  The “Transaction group” field represents a scope of seeking for patterns. As the objective is the examination of failure patterns within a user session, the username must be assigned here. Unsing this parameter, the algorithm can split the input data into subsets corresponding to the individual users, and search for patterns in them. The result of this step is the aggregated set of patterns matching the sample found in the individual user logs.
- **Transaction** field

  The Transaction field identifies the individual transactions in the log. The timestamp of the transaction can be used here as unique key.
- **Item** field

  The Item field represents the individual content of a transaction. In order to examine the failure code, the appropriate code must be given here.

The result of the mining algorithm is a list of pairs. The first element of a pair is the probability of the sequence identified by the second element of the pair. From this set of we have selected some interesting ones:

Pattern 1:	18.667%	[302]
		[500]
Pattern 2:	16.667%	[302]
		[302]
		[500]

This results illustrate, that in the first case the redirection had an immediate consequence in the form of a server failure, while in the second case a sequence of redirections was the trigger of the failure report. Additionally, the analysis delivers cumulative measures about the relative frequency of the occurrence of the individual sequences leading to a system failure report by URL redirection. After identifying this to critical cases a more in/depth analysis can be carried out in a similar fashion, as described by the previous section about OLAP.

3.2.2 Intelligent data mining algorithms, summary

The primitive example described above shows the appropriateness of data miners in the estimation of sequential patterns of interest, even if the relations are hidden in a huge data set. Such problems are unsolvable by simple statistical methods or by OLAP tools, only by the more advanced data miner systems. However, the elimination and explanation of the usefulness of the relations recognized by the computer, as we have seen in the examples, still remains the job of an expert.

4 Conclusions

We hope that we managed to show that different data mining algorithms can provide an essential support in all kinds of dependability related experiments especially in their evaluation phase. Simple OLAP technologies are useful primarily in the estimation of spatially correlated patterns (the simultaneous occurrence of different signals). The more advanced category of data miners can additionally used for the estimation of temporal sequences of interest.

In the field of dependability evaluation we see large perspectives in using them in the analysis of field log data, in the evaluation phase of stratified testing, and fault injection experiments. The main advantages that they provide an essential support for an adaptive and interactive control even if the registered set of samples is as large as several tens of gigabytes. In the case of dependability evaluation this can help in identifying rare and pattern sensitive fault mechanisms.

References

[1] Markus Lusti, Date Warehousing und Data Mining: Eine Einführung in entscheindungsunterstützende Systeme, Springer 1999

[2] Johann Petrak, Data Mining – Methoden und Anwendungen, ÖFAI 1997

[3] Lotus Domino Designer Help

[4] IBM DB2 Intelligent Miner for Data, Version 6.1