Mining Insurance Data at Swiss Life

J.-U. Kietz U. Reimer M. Staudt

Swiss Life, Information Systems Research (CH/IFUE), CH-8022 Zurich, Switzerland
{kietz,reimer,staudt}@swisslife.ch

Abstract

Huge masses of digital data about products, customers and competitors have become available for companies in the services sector. In order to exploit its inherent (and often hidden) knowledge for improving business processes the application of data mining technology is the only way for reaching good and efficient results, as opposed to purely manual and interactive data exploration. This paper reports the first steps of a project initiated at Swiss Life for mining its data resources from the life insurance business. Based on the Data Warehouse MASY collecting all relevant data from the OLTP systems for the processing of private life insurance contracts, a Data Mining environment is set up which integrates a palette of tools for automatic data analysis, in particular machine learning approaches.

1 Introduction

The exploding amount of available digital data in most companies due to the rapid technical progress of hardware and data recording technology has even more increased the tradeoff between just managing the data on the one hand and analyzing resp. exploiting the knowledge hidden in the data for business purposes on the other hand. The supply side of data management is characterized by huge data collections with a chaotic structure, often erroneous, of doubtful quality and only partially integrated. On the demand side we need abstract and high-level information that is tailored to the user's (mostly management people) needs and can be directly applied for improving the decision making processes, for detecting new trends and elaborating suited strategies etc. In order to bridge the gap between both sides, i.e. to find a reasonable way for turning data into information, we need (efficient) algorithms that can perform parts of the necessary transformations automatically. There will always remain interactive steps for this data analysis and information gathering task, e.g. selections of data subsets and contexts in which the whole analysis process takes place, and evaluation of the resulting hypotheses. However, the automatic processing should cover all those parts that can not be handled properly by human beings due to the size of transformation input and output.

Knowledge discovery in databases (KDD) aims at the automatic detection of *implicit*, previously *unknown* and potentially *useful* patterns in the data. One prerequisite for employing automatic analysis tools is a consolidated and homogenized set of data as input. Data Warehouses provide exactly this so that they form the ideal first steps in setting up a KDD process. The data analysis part of KDD is *discovery driven*, i.e. does not start with given hypotheses (as it is the case with using OLAP) but searches for new ones (within a given hypothesis space). The *Data Mining* algorithms constitute the kernel within the often cyclic and multi-step KDD process. Other than classic statistical approaches the exploitation of AI technology for data mining tasks was neglected for a long time. However, techniques from the areas of Machine Learning (esp. Inductive Logic Programming), Fuzzy Systems and Neural Networks promise more elaborate kinds of knowledge to be discovered in huge data collections.

In order to explore how Data Mining tools can complement its Data Warehouse, Swiss Life set up the project DAWAMI. This project is concerned with the design and implementation of a Data Mining environment. In particular, DAWAMI aims at enabling end users to execute mining tasks as independently as possible from data mining experts' support (*Goal 1*) and at making a broad range of data mining applications possible (*Goal 2*).

The rest of this paper is organized as follows: Sec-

Proceedings of the 23rd VLDB Conference Athens, Greece, 1997

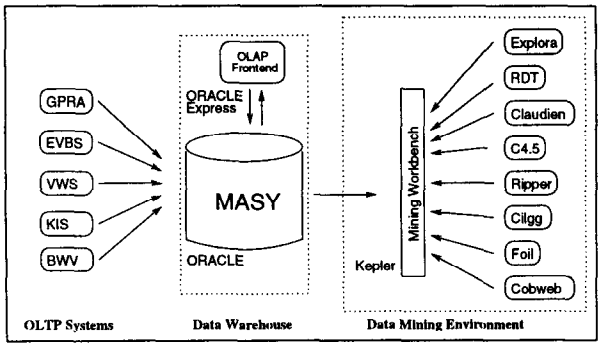

Figure 1: Sources, Warehouse, Mining Environment

tion 2 gives some information about the Data Warehouse MASY. In Section 3 we describe components of the Data Mining environment, in particular some of the algorithms employed for various analysis tasks, and their integration. Section 4 enumerates several concrete applications for DAWAMI and concludes with an outlook to further steps of the project.

2 The Data Warehouse MASY

The masses of digital data available at Swiss Life – not only data from insurance contracts but also from external sources, such as information about the socio-demographic structure and the purchasing power of the population in the various parts of the country – led to the development of the central integrated Data Warehouse MASY [Fri96]. This Warehouse takes over all data from the OLTP systems relevant for analysis tasks. From a database point of view a Data Warehouse is a collection of materialized views on the data in the source systems. Additional views on the warehouse data realize different levels of abstractions required for different analysis goals and end user needs.

MASY comprises data from six OLTP systems: contract data (about 650,000 contracts, some 500,000 clients) plus externally available data collections. Some of the data sources are shown in Figure 1. The basic insurance contract data e.g. stem from the system EVBS while GPRA contains (personal) data about all Swiss Life clients and partners. The system BWV is a publicly available catalogue of all (3 Million) Swiss households

MASY is implemented on top of ORACLE and follows a ROLAP warehouse architecture, i.e. employs on top of the relational structures a multidimensional OLAP-frontend. The database itself has both a normalized warehouse scheme gained from integrating the schemas of all source systems, and a derived (redundant) denormalized Galaxy schema intended to efficiently support multi-dimensional access. The first version of the warehouse in operation is expected to contain around 20 Giga Bytes of data, distributed over approximately 30 tables and 600 attributes.

3 Data Mining Technology

Based on the homogenized data in MASY the Data Mining environment offers tools and a methodology for supporting the information extraction task. The overall architecture of the environment and its relationship to MASY and the OLTP sources respectively is sketched in Figure 1.

In order to realize Goal 2 stated in Section 1 the data mining environment integrates a broad range of data mining approaches including classical statistical ones as well as techniques from the areas of Machine Learning and Neural Networks.

3.1 Algorithms

Figure 1 shows some examples for algorithms that are considered as candidates for becoming components of the mining environment. With respect to their output we can distinguish between the following categories:

1. detection of associations, rules and constraints: a common application of these techniques are e.g. market basket analyses.
2. identification of decision criteria: based on decision trees as one possible result we can support tasks like credit assignments.
3. discovery of profiles, prototypes and segmentations: classes of customers with similar properties e.g. can be grouped together and handled in a uniform way.

Another categorization concerns the kind of input data allowed:

a. sets of attribute-value pairs describing properties of certain data objects represented in one single relation (*attribute-based approaches*) or
b. input tuples from different relations and background knowledge, e.g. Datalog programs (*relational approaches*).

The latter category in particular allows to include additional background knowledge and arbitrary combinations of different classes of data objects. While statistical algorithms as well as most Machine Learning and commercially available Data Mining algorithms are attribute-based, Inductive Logic Programming approaches fall into the second category of relational approaches [Kie96]. Compared to solely applying statistical methods considering Machine Learning approaches has the following advantages (in addition to giving up the restriction to attribute-based input):

- more adequate treatment of nominal (categorial) attributes, esp. for hierarchically ordered values,
- faster heuristic methods, e.g. for clustering,
- more comprehensible results.

Following the two orthogonal categorizations above the algorithms in Figure 1 can be classified as follows:

Algorithm	Output	Input	Reference
Explora	1	a	[HK91]
RDT	1	b	[KW92]
Claudien	1	b	[RB93]
C4.5	2	a	[Qui93]
Ripper	2	a	[Coh95]
Cilgg	2	b	[Kie96]
Foil	2	b	[QCJ93]
Cobweb	3	a	[Fis87]

All the mentioned mining algorithms are based on pure main-memory processing. The main argument for disk-based data mining algorithms (e.g. [MAR96]) is the assumption that analysing more data produces better results.

However, there is usually a maximal accuracy any mining approach can reach on a dataset, in particular due to noise or because the attributes available allow only a certain degree of approximation of the mining goal. As a consequence, only a maximal number of examples is useful for learning a hypothesis space of a given complexity [Nat91]. An extended set of examples would contain more noise and tends to produce an overfit of the data, which may even lead to a decrease of accuracy.

More complicated hypothesis spaces to be searched demand also for more examples. However, in this case also more tests against the sample data have to be executed, which is orders of magnitude slower when the mining approach is disk-based. In summary, we think that a size of the input data which cannot be handled without disk support is necessary only for hypothesis spaces which on the other hand are anyway too complex for disk-based mining algorithms, i.e. the required hypothesis tests are much too slow.

We assume that the main memory in todays workstations (more than 1 GB) should be sufficient to hold the data needed to determine optimal results in hypothesis spaces searchable in reasonable time by todays processing power. Therefore, it seems more appropriate to combine data selection and sampling techniques as preprocessing before the actual mining phase is initiated with verification of the final hypotheses on the whole data sets after mining.

3.2 Integration Paradigm

Offering a great variety of different mining methods requires an open framework that allows the integration of quite different types of algorithms and a high extensibility. The *heart* of the DAWAMI architecture is the Data Mining workbench KEPLER [WWSE96] which relies on the idea of "Plug-Ins": Its tool description language and a basic API enable the building of wrappers around a given implemented mining algorithm and its inclusion into the applicable set of tools.

Accessing and preprocessing the data sources
To access source data, KEPLER allows to directly communicate with relational databases, or to read low-level format descriptions which are converted to relations by its generic parser component. The latter case is particularly intended for data from non-relational sources or sources not available online. A third possibility is the incorporation of application-specific conversion routines. In any of these cases, schema information about the arity and attribute types of each relation must be specified in a given format.

Additionally, KEPLER offers a set of operators for preprocessing relational data and configuring the required input for the mining tools:

- the classical set and relational algebra operations;
- randomized selection: a random number between 0 and 1 is generated for each tuple and leads to tuple elimination if it is greater than a given threshhold;
- introducing new attributes resp. values: numerical values can be converted to discrete ones, missing or additional values can be synthesized;
- various scaling operators for numerical data;
- a specific DINUS operator [KD94] which generates an amalgamation (if possible) of an input relation with a background knowledge relation such that the result has the same number of tuples. This operator works similarly to joining two relations together where the key of the second (background) occurs as component in the first. However, here such a foreign key relationship does not have to be given explicitly in the schema but is recognized from the current database state. A variant of the operator introduces a new (e.g. boolean) attribute for the input relation which states whether a certain value constellation occurs in the matching tuple of the background relation;
- an open (Prolog) programming interface allows for user defined operators.

Tool interaction
Basically, the specification for a tool wrapper consists of facts and some general (Prolog) procedures that have to be defined for each tool such that KEPLER can generate input masks for the source data and allow for parameters to be set. For the mining results a corresponding description states their type (e.g. rule or decision tree). A result handling clause called by KEPLER specifies whether the tool itself presents the result, KEPLER does it, or an external display tool should be started. For this purpose several libraries for data visualization (plotting, diagramms etc.) are offered. The mining tools themselves run as independent processes. Data exchange with KEPLER works with command line arguments and files.

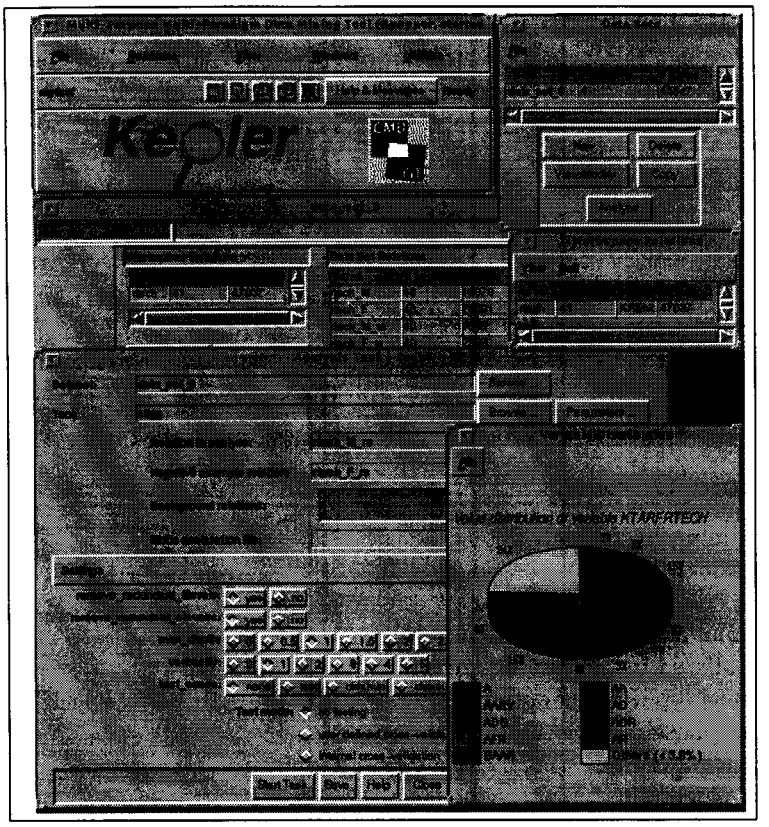

Figure 2: Interfacing CILGG with KEPLER

3.3 End User Support

According to Goal 1 (cf. Sec.1), it is our intention to push the mining task from the pure scientific specialists to real end users as far as possible. For Swiss Life those are e.g. the people at the sales front who are interested in selectively addressing new customers, PR managers who want to launch new promotion activities, or marketing people who plan new, attractive products. Probably, we need some mediator in between: The data analyst – in fact Swiss life has a staff group dedicated to all kinds of data analysis and market observation – performs the mining task by working with the integrated environment. The end users play the role as executors of already mined knowledge, e.g. applying learnt decision trees to new input data.

Even the data analysts cannot be provided with just the integration framework and algorithm implementation. Instead, they have to be supported with something like a *Mining Assistant* as a front-end for the application methodology as an extension of the interface already offered by KEPLER. A first aspect where such a tool should offer advice is the presentation of main problem categories and mining approaches. The support could itself consist in some primitive form of a decision tree, relating rough problem structures (including typical examples) with available algorithms. A second aspect concerns support for the necessary preprocessing steps. A relevant criterion during the preprocessing phase is e.g. the distribution of attribute values. Thus, clustering with unique key attributes obviously does not make much sense. Scaling of numerical attributes and selection of distance metrics is also an important topic. With the operators mentioned above powerful data restructuring and reconfiguring procedures can be executed. However, the user must have guidelines at hand how to combine them in order to meet the input requirements of the intended mining algorithms and respect their specific properties. E.g., several algorithms might not be able to handle numerical values and therefore eliminate them automatically. Others might leave out tuples with NULL values for certain attributes. In practice, a comparison of data and algorithm properties often rules out a selected algorithm and requires taking an appropriate alternative. To give an impression of how the analyst interacts with the mining environment Figure 2 shows the parameter input mask generated by KEPLER from the interface specification for Cilgg (window "Analysis Task 1"). The original relation `rtech` (41 attributes and 47037 tuples) which contains pure pension insurance contracts, constitutes the workspace for a certain mining task. During preprocessing several derived data set relations were generated, among others the relations `rtech_M_rs` and `rtech_F_rs`. Both relations serve as

input for a Cilgg analysis. Additionally required parameters and a number of specific options control the mining step. Pressing the button "Start Task" would cause KEPLER to call Cilgg as an independent analysis process, provide it with the necessary input values, and either allow the mining tool to independently present its mining results or return them by activating a default text-editor-based display. The pie diagram demonstrates the visualization facilities for attribute value distributions (attribute KTARFRTECH stating the tariff category of the insurances). These features are currently being extended to also cover density diagrams.

4 Applications and Outlook

For Swiss, Life Data Mining has a high potential to support marketing initiatives that preserve and extend the market share of the company. In order to experiment with different mining algorithms and to develop a fixed palette of tools offered in the mining environment a number of concrete applications were identified and selected for the first phase of DAWAMI:

Potential Clients: One might think that the wealthy inhabitants of a certain geographical area are the most promising candidates for acquiring new contracts, but this is usually not the case. An interesting data mining task is therefore to find out what the typical profiles of Swiss Life customers are with respect to the various insurance products. An insurance agent can then use these profiles to determine from a publicly available register of households those non-customers of his geographic area which are quite possibly interested in a certain kind of life insurance.

Customer Losses: One way to reach lower cancellation rates for insurance contracts is via preventive measures directed to customers that are endangered through personal circumstances or better offers from competitors. By mining the data about previous cancellations, using as background knowledge unemployment statistics for specific regions and questionnaire data, we expect to obtain classification rules that identify customers that may be about to terminate their insurance contracts.

Other mining tasks concern the identification of differences between the typical Swiss Life customers and those of the competitors, and the segmentation of all persons in the MASY warehouse into the so called *RAD 2000 Target Groups* based on a set of fuzzy and overlapping group definitions developed by the Swiss Life marketing department some years ago.

All mentioned applications above stem from one of the two main Swiss Life business areas, namely the private life insurance business. In Switzerland, all companies are obliged to provide old-age insurances to their employees in additon to the public pension fund and private provisions. Pension fund management and the development of pension schemes is the second insurance market segment Swiss Life is engaged in. Certainly, we will identify a number of questions in this area, too, where Data Mining technology can improve the acquisition and management processes.

Acknowledgements: We would like to thank the KEPLER team at GMD St. Augustin for their collaboration and support.

References

[Coh95] W. W. Cohen. Fast effective rule induction. In *Machine Learning: Proceedings of the Twelfth International Conference (ML95)*, 1995.

[Fis87] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139 – 172, 1987.

[Fri96] M. Fritz. The employment of a data warehouse and OLAP at Swiss Life (in German). Master's thesis, University of Konstanz, Dec. 1996.

[HK91] P. Hoschka and W. Klösgen. A support system for interpreting statistical data. In Piatetsky-Shapiro and Frawley, editors, *Knowledge Discovery in Databases*. 1991.

[KD94] J.-U. Kietz and S. Džeroski. Inductive logic programming and learnability. *SIGART Bulletin*, 5(1), 1994.

[Kie96] J.-U. Kietz. *Inductive Analysis of Relational Data (in German)*. PhD thesis, Technical University Berlin, Oct. 1996.

[KW92] J.-U. Kietz and S. Wrobel. Controlling the complexity of learning through syntactic and task-oriented models. In S. H. Muggleton, editor, *Inductive Logic Programming*, pages 335 – 360. Academic Press, 1992.

[MAR96] M. Metha, R. Agrwal, and J. Rissanen. SLIQ: A fast scalable classifier for data mining. In *Proc. 5th Int. Conf. on Extending Database Technology*, 1996.

[Nat91] B. Natarajan. *Machine Learning — A theoretical Approach*. Morgan Kaufmann, 1991.

[QCJ93] R. Quinlan and R. M. Cameron-Jones. Foil: A midterm report. In P. Brazdil, editor, *Proceedings of the Sixth European Conference on Machine Leaning (ECML-93)*, pages 3–20. LNAI 667,Springer.

[Qui93] J. R. Quinlan. *C4.5: Programms for Machine Learning*. Morgan Kaufmann, 1993.

[RB93] L. De Raedt and M. Bruynooghe. A theory of clausal discovery. In S.H. Muggleton, editor, *The Third International Workshop on Inductive Logic Programming*, 1993.

[WWSE96] S. Wrobel, D. Wettschereck, E. Sommer, and W. Emde. Extensibility in data mining systems. In *Proc. of the 2nd Int. Conf. On Knowledge Discovery and Data Mining*. AAAI Press, 1996.