USABILITY ISSUES IN DATA MINING SYSTEMS

Fernando Berzal, Juan-Carlos Cubero, Nicolás Marín, José-María Serrano
Dept. Computer Science and Artificial Intelligence, E.T.S. Ingeniería Informática, University of Granada
C/ Periodista Daniel Saucedo Aranda, s/n. 18071 Granada, Spain
Email: fberzal@decsai.ugr.es, jc.cubero@decsai.ugr.es, nicm@decsai.ugr.es, jmserrano@decsai.ugr.es

Ignacio Blanco
Dept. Languages and Computation, University of Almería, Ctra. Sacramento s/n 04120 La Cañada, Almería, Spain
Email: iblanco@ual.es

Keywords: Data Mining, usability, component-based systems, design patterns

Abstract: When we build data mining systems, we should reflect upon some design issues which are often overlooked in our quest for better data mining techniques. In particular, we usually focus on algorithmic details whose influence is minor when it comes to users' acceptance of the systems we build. This paper tries to highlight some of the issues which are usually neglected and might have a major impact on our systems usability. Solving some of the usability problems we have identified would certainly add to the odds of successful data mining stories, improve user acceptance and use of data mining systems, and spur renewed interest in the development of new data mining techniques. Our proposal focuses on integrating diverse tools into a framework which should be kept coherent and simple from the user's point of view. Our experience suggests that such a framework should include bottom-up dataset-building blocks to describe input datasets, expert systems to propose suitable algorithms and adjust their parameters, as well as visualization tools to explore data, and communication and reporting services to share the knowledge discovered from the massive amounts of data available in actual databases.

1 INTRODUCTION

Data mining techniques (Han, 2001) allow us to analyze huge datasets, cluster data, build classification models, and extract associations and patterns from input data. The data miner has to analyze large datasets and she needs to make use of data mining tools to perform her task. Data is gathered and data mining algorithms are used in order to build models which summarize the input data. Those models may provide the information our user needs, or they may just suggest new ways to explore the available data.

When researchers build data mining systems, they should take into account several design issues which are often overlooked in the quest for better algorithms and techniques. In particular, every effort aimed at the development of data mining systems should keep system usability high among its priorities.

As any other software system, data mining systems are used by people and system usability is critical for user acceptance, provided that the knowledge workers who will make use of data mining systems are not necessarily knowledgeable about computers. In fact, "usability is arguably the quintessential measure of software quality… The hope is to build software that better supports the work of real people, that serves useful purposes, and makes tasks easier or simpler." (Constantine, 2001)

This paper focuses on some of the issues which are usually neglected and have a negative effect on the overall system usability. Apart from the design qualities which should be common to any human artifact, such as consistency, visibility, simplicity, and error-resilience, which have been extensively covered elsewhere (Norman, 1988), and the practical techniques and guidelines which should be observed in the development of interactive software systems (Schneiderman, 1998), we have found some peculiarities which tend to render data mining systems useless. Even the most advanced data mining and OLAP systems suffer from some of the design mishaps we try to highlight in the following sections.

2 DEFINING THE INPUT

Let us begin taking into account the case of assembling the datasets which are fed as input into the data mining systems in order to build knowledge models.

These datasets may come from heterogeneous information sources, although data mining tools usually work with tables in the relational sense. Each table contains a set of fixed-width tuples which can be obtained either from relational databases or any other data source (ASCII or XML files, for example). All tabular datasets have a set of columns (also called attributes), each one of them with a unique identifier and an associated data type.

A powerful data mining system should allow the specification of order relationships among attribute values and the grouping of attribute values to define concept hierarchies. It should also be capable of performing heterogeneous queries over different databases and information sources. The independently-retrieved datasets, in fact, might be processed further in order to join them with other datasets (data integration), to standardize concept representations and eliminate redundancies (data cleaning), to compute aggregations (data summarization), or just to discard part of them (data filtering).

Formal models and query languages can be used to perform all the aforementioned operations involving datasets. However, typical users are not prepared to use such models and languages to define the customized datasets they need. They will probably reject a system which requires them to learn any complex formalism, even if it seems logical and simple for us as computer scientists.

In order to improve system acceptance, for example, we could use a bottom-up approach to build the datasets from their original data sources. A small family of dataset-building components should provide the user with all the primitives she needs to build her own datasets from the available data sources:

- **Wrappers** are responsible for providing uniform access to different data sources. Data stored as sets of tables in relational databases can be retrieved performing standard SQL queries through call-level interfaces such as JDBC or ODBC. Data stored in other formats would obviously require specific wrappers. Anyhow, data access would be simple and uniform from the user's point of view.

- **Joiners** are used to join multiple datasets. They allow the user to combine information coming from different sources. Joiners are also useful to include lookup fields into a given dataset (as in data warehouse star schemas) and to specify relationships between two datasets from the same source (e.g. master/detail relationships).

- **Aggregators** summarize datasets in order to provide a higher-level view of the available data. Aggregations are useful in a wide range of OLAP applications, where trends are much more interesting than particular details. Common aggregation functions include MAX, MIN, TOP, BOTTOM, COUNT, SUM, and AVG.

- **Filters** perform a selection over the input dataset to obtain subsets of the original input dataset. In Data Mining applications, filters can be used to perform samplings, to build training datasets (e.g. when using cross-validation in classification problems), or just to select the data we are interested in for further processing.

- **Transformers** are also needed to modify dataset columns. We could distinguish two kinds of transformers: *encoders* and *extenders*. While encoders just encode input data and are useful for data cleaning and integration (to ensure that real-world entities are always represented in the same way even when represented differently in different data sources), extenders can be used to append new fields to a given dataset, fields whose values are completely determined by the other field values in the same tuple. Those fields, a.k.a. calculated fields, are useful for managing dates and converting measurement units.

The above components can be easily combined in tree-like structures to build highly personalized datasets. As when using formal query languages, the resulting datasets are amenable to standard query optimization techniques, although even computer illiterate users are able to use complex data mining systems just by linking previously defined dataset-modeling components.

As shown in Figure 1, our proposed dataset-building blocks can be viewed as particular instances of three well-known design patterns (Gamma et al., 1995). Wrappers give us access to heterogeneous data sources. Joiners help us define complex datasets using the composite design pattern. Finally, aggregators, filters, and transformers let us modify

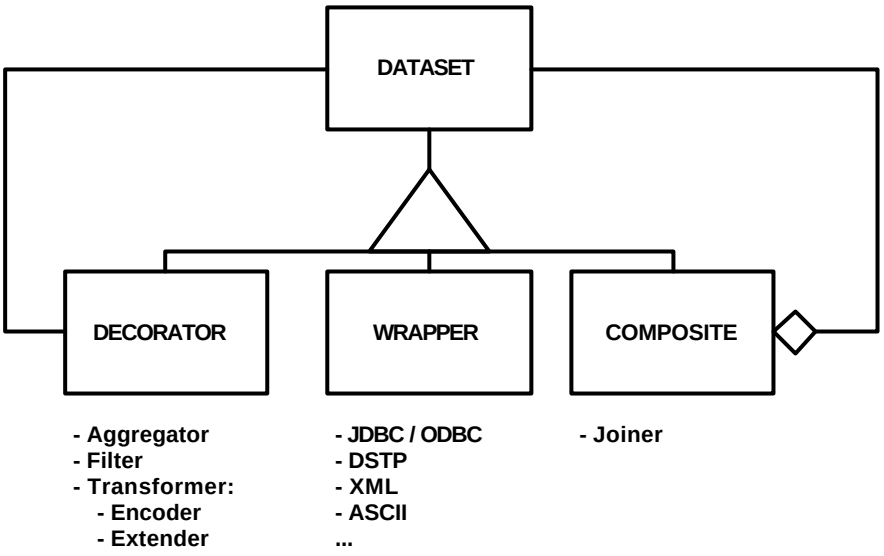

Figure 1: Dataset-building blocks as examples of well-known design patterns.

the content of existing datasets, so they act as decorators using the common terminology in the pattern community.

3 MINING DATA

Once users have defined the datasets they will use as starting points in their quest for knowledge discovery in databases, they usually face an insurmountable obstacle when they find they have a myriad tools and techniques available to analyze their data. Moreover, each tool or technique usually has plenty parameters to adjust its performance. Given this scenario, users feel themselves surpassed by the situation and probably do not know how to begin their analysis.

Data mining systems should provide some guidance on the kind of tools users should employ by analyzing the nature of the users' datasets and should also automatically set the parameters needed for the data mining task at hand. The aforementioned goals could be accomplished by integrating expert systems into the data mining framework and studying heuristics which could help in setting the parameters of data mining algorithms, as we have done with ART, a decision tree-based classification model we have proposed (Berzal et al., 2003). Unfortunately, users are often overwhelmed by the complexity of the systems they are supposed to manage.

Therefore, transparency is a must for a wider deployment of data mining systems in modern enterprises, both for users and for programmers. Users should not need to be aware of the underlying complexity of the data mining system, while programmers should be able to create new data mining algorithms, techniques, and tools just by implementing a core set of well-defined interfaces, without understanding the nitty-gritty details of their data mining infrastructure. Future advances in this direction will benefit both researchers and practitioners in the field.

4 EXPLORING RESULTS

Lucky users have been able to define the precise datasets they want to use and they have been even succeeded setting the all the parameters required by the algorithms they carefully selected among the vast range of existing data mining techniques. However, once they think they have finished their mining efforts, they find themselves deluged by the huge amount of data their algorithms generate. Sometimes, the output is bigger and more complex than the input it was derived from (as happens, for example, when mining association rules from transactional data).

The above situation, when output is too large to be grasped by the poor human user, is referred to as a second-order data mining problem. The

unfortunate user is left again with a huge amount of data he has to analyze by herself, maybe with the help of more data mining tools she will have to master.

In situations like this, summarization and visualization tools could provide invaluable insight into the data. In fact, "we explore in order to make maps, and eventually develop a map that is close enough to the territory to represent it for practical purposes. Visual tools are often the best way to work by successive approximations" (Weinberg, 1989).

5 SHARING RESULTS

If users are able to complete their data mining endeavors, they still need a final step to succeed in the use of data mining systems. They must be able to represent and communicate the insight they have obtained from their analysis.

In order to accomplish this final step, users should be able to store anything they can reuse in the future, so that they do not have to repeat the steps they made to obtain the results they already have. But, above all, they must be able to share the information they obtain after their data mining effort.

This final communication step must also be addressed by data mining system developers, who often forget the big picture and get stuck in the algorithmic details of the techniques they develop. Appropriate reporting mechanisms are needed in data mining systems if you want to get your data mining systems out of the lab into production. Groupware issues are especially important in data mining applications, where the discovered knowledge must be properly represented and communicated in order to be leveraged into the real world.

6 CONCLUSIONS

We all have had frustrating experiences using ill-designed software. Since "people's productivity and comfort relate directly to the usability of the software they use" (Juristo et al., 2001), we should pay greater attention to the usability problems which plague the systems we blindly develop trying to improve qualities which do not have as much influence as usability on software products.

Solving some of the usability problems we have identified in the preceding sections would certainly increase the odds of successful data mining stories, improve user acceptance of data mining systems, and spur interest in the development of new data mining techniques. Any of these reasons make data mining systems usability a worthwhile research area by itself.

In order to improve system usability, data mining systems should be built using a multidisciplinary approach instead of focusing on just a handful of partially related techniques (as happens in most current systems). Our proposed approach would lead to the integration of diverse tools into a unified framework, coherent and simple from the user's point of view. Our experience suggests that such a framework should include a bottom-up dataset-building module to define input datasets, expert systems to recommend algorithms and tune their parameters, visualization tools to explore data, and communication and reporting facilities to share the knowledge discovered from the huge amounts of data available in actual databases.

REFERENCES

Berzal, F., Cubero, J.C., Sánchez, D., and Serrano, J.M., 2003. *ART: A Hybrid classification model*. Machine Learning Journal, to be published.

Constantine, L.L., 2001. *The Peopleware Papers: Notes on the human side of software.* Prentice Hall PTR, ISBN 0-13-060123-3

Gamma, E., Helm, R., Johnson, R., and Vlissides, J., 1995. *Design patterns: Elements of reusable object-oriented software.* Addison Wesley, ISBN 0-201-63361-2.

Han, J., and Kamber, M., 2001, *Data Mining: Concepts and techniques*. Morgan Kaufmann Publishers, ISBN 1-55860-489-8

Juristo, N., Windl, H., and Constantine, L. (eds.), 2001. *Special section on Usability Engineering*. IEEE Software, Vol. 18, No. 1, January/February 2001.

Norman, D.A., 1990. *The design of everyday things*. Doubleday / Currency, ISBN 0-385-26774-6.

Schneiderman, B., 1998. *Designing the user interface: Strategies for effective human-computer interaction*. Addison Wesley, ISBN 0-201-69497-2.

Weinberg, G.M., 1989. *Exploring requirements: Quality before design*. Dorset House Publishing, ISBN 0-932633-13-7.

Work partially supported by
Eureka! project FIT-070000-2001-792