AUSTRIAN JOURNAL OF STATISTICS
Volume 31 (2002), Number 2&3, 231-239

Equational Reasoning as a Tool for Data Analysis

Michael Bulmer
University of Queensland, Brisbane, Australia

Abstract: A combination of deductive reasoning, clustering, and inductive learning is given as an example of a hybrid system for exploratory data analysis. Visualization is replaced by a dialogue with the data.

Keywords: Automated Deduction, Clustering, Induction.

1 Introduction

The aim of this paper is to show how data analysis may be enriched by the use of an automated deduction system, and simultaneously how a system for automated deduction may be extended to carry out data analysis. Of course there are many ways that this could be done and many frameworks on which it could be built. Here we will use *equational reasoning* and show how its discrete algebraic language may be used to analyze data with discrete and continuous variables.

In Section 2 we introduce the basics of equational reasoning using a term algebra and show how information is represented in this language and how deductions and inductive conjectures can be made. The data analysis will begin with some form of inductive learning and then use deduction to make predictions from the conjectures and existing knowledge. Section 3 describes the discretization process that is necessary for analyzing continuous data, with Section 4 giving a second method of learning information from a database of observations. This discrete form of reasoning fits well with a simplified natural language, allowing a user to have a *dialogue* with the software rather than using visualization to analyze data. Example dialogues are given in Section 5. Finally, in Section 6 we summarize the potentials of this approach and detail some of its shortcomings.

2 Equational Reasoning

We begin by giving an introduction to the equational approach to deduction and a simple functional learning method that follows from it. The language of our equational reasoning is that of a *term algebra* generated by a *signature*. Figure 1 gives an example signature for representing knowledge about family relationships. Here Person and Sex are *sorts* while father and Alice are *words*. Terms of the algebra are generated by making paths in the signature, which we will write in reverse order. For example, Figure 1 gives rise to the terms father Alice : Ground $\rightarrow$ Person and sex mother : Person $\rightarrow$ Sex, as well as the singleton Clare : Ground $\rightarrow$ Person. For a term $f : \sigma \rightarrow \tau$ we call σ the *domain* of f, $\mathrm{dom}(f)$, and τ the *codomain* of f, $\mathrm{cod}(f)$.

Terms whose domain is Ground are the *elements* of the language and we will usually write, for example, Clare $\in$ Person to reflect this. Terms with domains other than Ground are *functions*. All sorts σ have the *erasing* term ! : $\sigma \rightarrow$ Ground. That is, if $g \in \sigma$ then

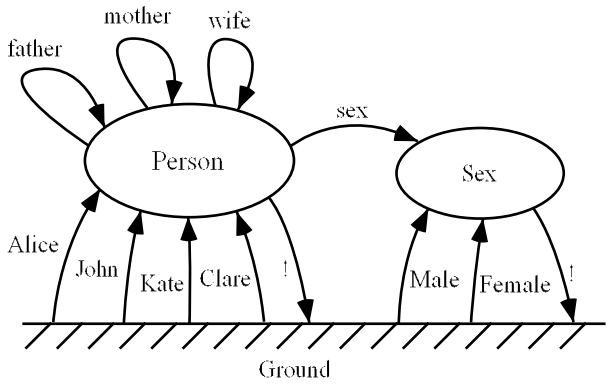

Figure 1: A signature for describing family relationships

the term $f\ !\ g$ is equal to f. Thus $f\ !$ is a constant function on the elements of σ. For example,

$$\mathsf{Male}\ !\ \mathsf{John} = \mathsf{Male}.$$

Knowledge is captured by giving equations between terms in the language. For example, the *ground* equation $\mathsf{father}\ \mathsf{Alice} = \mathsf{John}$ represents the statement that "Alice's father is John". *Functional* equations, such as $\mathsf{father}\ \mathsf{sister} = \mathsf{father}$ ("the father of a person's sister is their father") and $\mathsf{sex}\ \mathsf{father} = \mathsf{Male}\ !$ ("the sex of any person's father is male"), capture universal statements since they hold for any element of the domain. For example, if we believe the three equations

$$\left\{ \begin{array}{l} \mathsf{father}\ \mathsf{sister} = \mathsf{father}, \\ \mathsf{father}\ \mathsf{Alice} = \mathsf{John}, \\ \mathsf{sister}\ \mathsf{Kate} = \mathsf{Alice} \end{array} \right\}$$

then we may deduce

$$\mathsf{father}\ \mathsf{Kate} = \mathsf{father}\ \mathsf{sister}\ \mathsf{Kate} = \mathsf{father}\ \mathsf{Alice} = \mathsf{John}.$$

This equational deduction, which can be done mechanically using rewrite rules and the Knuth-Bendix algorithm, has a rich history in automated theorem proving (Hsiang et al., 1992). However, little work has been done on using equational reasoning for the inductive learning required by data analysis.

Yet equational induction is straightforward in the functional setting. Consider again the above deduction. Suppose instead we knew

$$\left\{ \begin{array}{l} \mathsf{father}\ \mathsf{Alice} = \mathsf{John}, \\ \mathsf{sister}\ \mathsf{Kate} = \mathsf{Alice}, \\ \mathsf{father}\ \mathsf{Kate} = \mathsf{John} \end{array} \right\}.$$

Then we would observe that

$$\mathsf{father}\ \mathsf{sister}\ \mathsf{Kate} = \mathsf{father}\ \mathsf{Kate},$$

and so might conjecture the functional equation

$$\mathsf{father}\ \mathsf{sister} = \mathsf{father}$$

since the two functions have the same output when applied to the singleton Kate.

In general, suppose we have a database of ground equations. If it is observed that for all singletons $a \in \text{dom}(f)$ there is a singleton $b \in \text{cod}(f)$ such that $fa = b$ and $ga = b$, then we conjecture that $f = g$. This is a *summative* conjecture since complete information about the function values exists in the database. A weaker form of induction, allowing predictions, is to conjecture $f = g$ if there is at least one singleton $a \in \text{dom}(f)$ such that $fa = b$ and $ga = b$ and there is no $a' \in \text{dom}(f)$ such that $fa' = b$ and $ga' = c$, where $b \neq c$. For example, given the data

$$D = \left\{ \begin{array}{l} \mathsf{father\ Alice = John}, \\ \mathsf{mother\ Alice = Clare}, \\ \mathsf{wife\ John = Clare}, \\ \mathsf{father\ Paul = Peter}, \\ \mathsf{wife\ Peter = Mary} \end{array} \right\}$$

we could conjecture that wife father = mother. We could then predict from D that

$$\mathsf{mother\ Paul = wife\ father\ Paul = wife\ Peter = Mary}.$$

This method has been shown to be successful in certain problem domains (Bulmer, 1996a). In particular, the bidirectional nature of equations makes this representation very effective. In the above example, if we observed mother Paul = Mary instead of wife Peter = Mary, we could use the same conjecture to predict wife Peter = Mary.

We find that D also gives rise to the erasing conjecture mother = Clare !, since the single observation suggests that mother is a constant function. Such conjectures are sometimes important, such as sex father = Male !, but can also arise in this trivial manner; as the database grows, the trivial ones will usually evaporate. However, such erasing conjectures will be the main focus of the use of subsets in Section 4.

Note that in the context of D we face the problem of deciding between two predictions for mother Paul, either Mary (from wife father = mother) or Clare (from mother = Clare !). Such issues have been addressed by looking at *consistent theories* for the data (Bulmer, 1996b); our implicit requirement above that the induction algorithm only generates conjectures that are individually consistent with the data is to allow such analysis of consistency amongst sets of conjectures. However, this leads to poor results for data sets that have overlapping classes or which are noisy. Error reduction methods, such as bootstrap aggregating (Breiman, 1996), can help overcome this in practice.

3 Data Analysis and Discretization

So far we have dealt only with discrete data. For practical data analysis we need to be able to work with continuous observations as well. The approach we will use here is to preprocess continuous variables by collapsing them into a collection of discrete groups.

This discretization has an important role in many data mining methods, and indeed is an intrinsic part of all decision trees. Methods use some criterion, such as maximizing information gain or minimizing the description length (see Fayyad and Irani, 1993), to find the best splitting of a continuous attribute with respect to a target attribute. However,

here our aim is not necessarily the same as that of a classification problem. Instead we would like to discretize variables individually, without reference to any existing groups.

Thus discretization in this paper is simply a univariate clustering problem, to which we can apply many existing tools. One of the simplest and fastest is the *k-means* algorithm (see, for example, Berry and Linoff, 1997), and we have used this for our examples. Choosing the number of discrete groups in clustering is a difficult problem in general and in our case the variables we are discretizing may be quite homogenous. We have taken our cue from natural language and have arbitrarily split all continuous variables into five groups, since five is an easy number of groups to give names to. For example, heights will be split into "very short", "short", "medium", "tall", "very tall".

4 Subset Induction

Consider an alternative set of observations about the family in Figure 1:

$$\left\{ \begin{array}{l} \mathsf{father\ Alice} = \mathsf{John}, \\ \mathsf{mother\ Alice} = \mathsf{Clare}, \\ \mathsf{father\ Kate} = \mathsf{John} \end{array} \right\}$$

Here it is plausible that Kate's mother is also Clare, since her father is the same as Alice's father and so she might belong to the same "class" as Alice. This is precisely the setting of many classification problems. However, the simple functional induction of Section 2 fails to make this prediction since there are no functions which have the same outputs (except for the trivial conjectures $\mathsf{father} = \mathsf{John}\,!$ and $\mathsf{mother} = \mathsf{Clare}\,!$). To overcome this we need to extend our language and equation processing to capture such reasoning.

Any functional equation can be used as a *characteristic function* to describe a subset of a sort. (The use here is related to subobjects in category theory; (see Goldblatt, 1983,for details).) For example, from the equation $\mathsf{father\ Alice} = \mathsf{John}$ we could create the subset $\mathsf{fatherJohn}$ from the classifier $\mathsf{father} = \mathsf{John}\,!$, as illustrated in Figure 2.

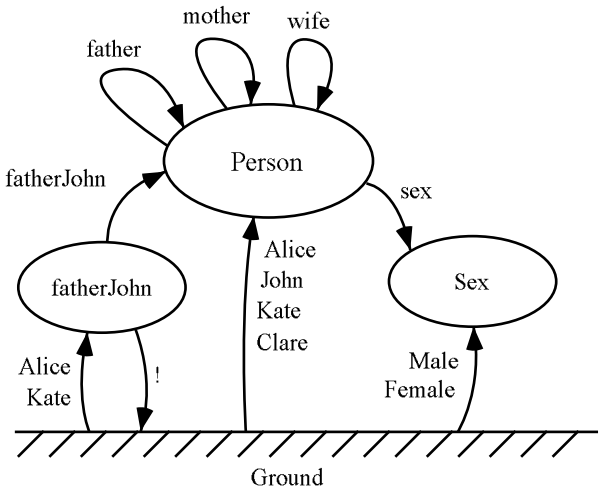

Figure 2: Signature for families with a subset

The equation processing system is augmented to treat Alice and fatherJohn Alice interchangeably, a process that is similar to the handling of associative and commutative domains (Dershowitz, 1989). With subsets present, the learning algorithm can then conjecture

$$\text{mother fatherJohn} = \text{Clare}\ !.$$

From this it can make the prediction

$$\begin{aligned}\text{mother Kate} &= \text{mother fatherJohn Kate}\\ &= \text{Clare}\ !\ \text{Kate}\\ &= \text{Clare}.\end{aligned}$$

The main advantage of this system is that it fits into the broader scope of equational reasoning and induction. For example, consider the database

$$\left\{\begin{array}{l}\text{father Alice} = \text{John},\ \text{mother Alice} = \text{Clare},\\ \text{wife John} = \text{Clare},\ \text{father Kate} = \text{John},\\ \text{father Paul} = \text{Peter},\ \text{mother Paul} = \text{Mary}\end{array}\right\}$$

This database gives conjectures under both the functional induction described in Section 2 and the subset induction described here. We obtain the conjectures

$$\left\{\begin{array}{l}\text{wife father} = \text{mother},\\ \text{mother fatherJohn} = \text{Clare}\ !\end{array}\right\}.$$

From these conjectures we can predict mother Kate $=$ Clare in two different ways, and wife Peter $=$ Mary, and also make a range of other deductions.

5 Dialogues

The use of equational reasoning moves the emphasis of data analysis from visualisation to a kind of dialogue with the data. In this section we present two such dialogues, generated automatically by an implementation of the system described here.

5.1 Heights

The first data set is a simple one obtained from student responses to a survey used in the teaching of an introductory statistics subject (see Bulmer, 1999, for the data). Among other variables, each student gave their sex and height. After loading the data from the web page, the dialogue in Table 1 was recorded.

The reasoner begins with 89 observations of heights, which are automatically split into the 5 groups. There was no measurement for Alice, so no deductive consequences about Alice's height can be made. Once the reasoner is told Alice's height this is discretized and can be used to make deductions. (Any statement made by the reasoner is a deductive truth unless it is preceded by "I think that".) When asked about sex, no split is given since its value is already discrete. However, based on observations about height and sex,

Table 1: A dialogue about height and sex

User	Reasoner
Tell me about height.	I have 89 observations about height. The values are split into very short, short, average height, tall, and very tall.
How tall is Alice?	I don't know.
Alice's height is 156.	Okay.
How tall is Alice?	Alice is very short.
Tell me about sex.	I have 89 observations about sex. I think that if a person's height is very tall then that person is male. I think that if a person's height is very short then that person is female.
What sex is Alice?	I think that Alice is female.

the reasoner has formed two subset conjectures about sex. These can be used to predict Alice's sex.

It is worth noting here that the subsets are *crisp*; the induction could be extended to use fuzzy clustering and fuzzy subsets. For example, most of the people clustered in Short by the reasoner are also female, and so the reasoner could make predictions with varying degrees of certainty.

5.2 Irises

The second data set is Fisher's famous iris data (Fisher, 1936). The data includes measurements of the petals and sepals of 50 specimens each of 3 varieties of iris. Figure 3 gives part of the signature for representing this data. Table 2 gives a dialogue with the iris data set, showing the discretization of petal length and the resulting conjectures about determining petal lengths and iris varieties. Again these can be used to make predictions about a new plant.

6 Conclusions

The approach described here has been applied to a variety of data settings in the UCI Repository (Black and Merz, 1998). As a classification tool it does not perform as well as directed methods such as discriminant analysis and decision tools, but it does do reasonably well on most problems, as suggested by Holte (1993). In addition, the equational framework allows the easy inclusion of background functional knowledge, and can also be used to generate abductive hypotheses.

However, the aim of this paper has been mainly to show the potential of embedding data analysis within a framework for automated deduction. As there are a broad range of

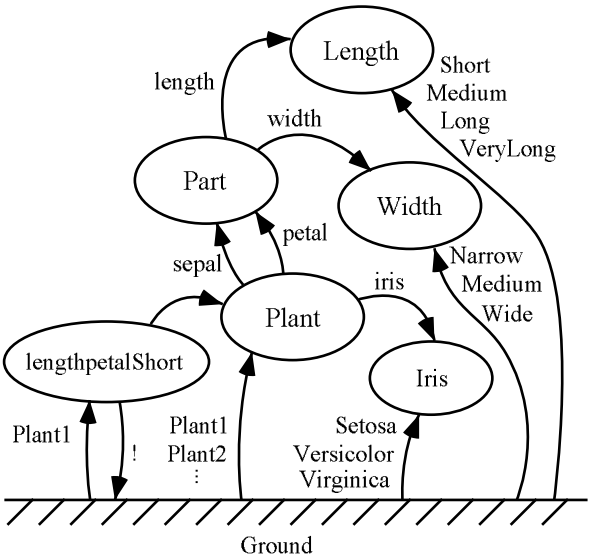

Figure 3: Signature for iris data

approaches to both tasks, there is naturally a large number of possible systems that could be developed in this way. We have demonstrated how one method, equational reasoning, can be used to enrich the analysis of data.

References

M.J.A. Berry and G. Linoff. *Data mining techniques for marketing, sales, and customer support*. John Wiley & Sons, 1997.

C.L. Black and C.J. Merz. UCI repository of machine learning databases, 1998. www.ics.uci.edu/˜mlearn/MLRepository.html.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

M. Bulmer. Inductive equational reasoning. In *Pacific Rim International Conference on Artificial Intelligence (PRICAI'96): Topics in Artificial Intelligence*, number 1114 in Lecture Notes in Artificial Intelligence, pages 13–22. Springer-Verlag, 1996a.

M. Bulmer. Inductive theories from equational systems. In D.L. Dowe, K.B. Korb, and J.J. Oliver, editors, *Information, Statistics and Induction in Science (ISIS'96)*, pages 260–268. World Scientific, 1996b.

M. Bulmer. Survey data from students in "MS113 Elementrary Statistics 1", 1999. http://www.maths.uq.edu.au/cgi-bin/mrb/fetch?1999a,ms113,survey.

N. Dershowitz. Completion and its applications. In Aït Kaci and M. H. & Nivat, editors, *Resolution of Equations in Algebraic Structures*, volume 2, pages 31–85. Academic Press, 1989.

Table 2: A dialogue about irises

User	Reasoner
Tell me about petal length.	I have 150 observations about petal length. The values are split into very short, short, medium length, long, and very long. I think that if a plant's sepal is very long then that plant's petal is very long. I think that if a plant's sepal is very wide then that plant's petal is very short. I think that if a plant's petal is very narrow then that plant's petal is very short. I think that if a plant is Setosa then that plant's petal is very short.
Tell me about iris.	I have 150 observations about iris. I think that if a plant's sepal length is very long then that plant is Virginica. I think that if a plant's sepal width is very wide then that plant is Setosa. I think that if a plant's petal length is very short then that plant is Setosa. I think that if a plant's petal length is short then that plant is Versicolor. I think that if a plant's petal length is very long then that plant is Virginica. I think that if a plant's petal width is very narrow then that plant is Setosa. I think that if a plant's petal width is narrow then that plant is Versicolor. I think that if a plant's petal width is very wide then that plant is Virginica.
The petal length of Plant151 is 5.8.	Okay.
What iris is Plant151?	I think that Plant151 is Virginica.

U. Fayyad and K. Irani. Multi-interval discretization of continuous valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.

R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

R. Goldblatt. *Topoi, the Categorical Analysis of Logic*. North-Holland, 1983.

R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

J. Hsiang, H. Kirchner, P. Lescanne, and M. Rusinowitch. The term rewriting approach to automated theorem proving. *Journal of Logic Programming*, 14:71–99, 1992.

Author’s address:

Dr. Michael Bulmer
Department of Mathematics
University of Queensland
Queensland 4072
Australia

Tel. +61 7 3365 7905
Fax +61 7 3365 1477
E-mail: mrb@maths.uq.edu.au
http://www.maths.uq.edu.au/˜mrb