MetaData for Database Mining

John Cleary, Geoffrey Holmes, Sally Jo Cunningham, and Ian H. Witten
Department of Computer Science
University of Waikato
Hamilton, New Zealand.

Abstract: At present, a machine learning application is accomplished by carefully crafting a single table from an often complex, multi-table database. The metadata necessary to create this table is rarely formally recorded, and is sometimes implicit in the structure of the database or the typing of the attributes. We categorize the types of metadata that we have encountered in our work with machine learning applications in agriculture, and describe a first generation tool that we have built to aid in the recording and use of metadata in database mining.

1. Introduction

Database mining is the process of discovering previously unknown and potentially interesting patterns in large databases [1]. The extracted information is typically organized into a prediction or classification model of the data. Techniques for finding the interesting patterns range from applications of computational models such as machine learning, fuzzy logic and artificial neural networks to statistical models such as regression trees and Bayesian networks.

These techniques all perform some form of sub-optimal search of the database to recover the patterns implicit in the data. In most cases, interesting data is extracted as a two-dimensional view and patterns or rules are discovered within this view. This process is repeated over many different views of the data, and has led to the belief among some researchers that data mining is, to all intents and purposes, file mining — the structure and the semantics of the data are being ignored in constructing the two-dimensional views.

We have extensive experience in the process of file mining using machine learning algorithms [2,3], and it is our belief that this process could be dramatically improved by taking advantage of a metadata model of the data in the process of discovering interesting patterns. In this paper we motivate through examples three separate kinds of metadata information — data type, relational and rule-based — that we feel are useful in assisting a machine learning algorithm to discover more meaningful models of the data. Some, but not all, of this information can be re-engineered from existing database system catalogues [4].

Most state-of-the-art machine learning programs have no ability to utilise knowledge of data in a principled fashion. The level of metadata information that is currently being exploited by these algorithms is principally related to data type — whether attributes are discrete with unordered nominal values or continuous with numeric values. Until the learning process itself is adapted to take into account more information about the data we must work "by hand" to exploit metadata information in the extraction of the two-dimensional views in order to make the overall discovery of patterns more effective.

To support this extraction process we are engineering a software tool which makes explicit use of metadata information, and a prototype implementation is integrated into our WEKA workbench of machine learning programs. Our long term aim is to develop software that will semi-autonomously navigate a relational database to discover interesting patterns while taking into account metadata information.

In this paper we first establish a computational context for our work in developing applications of machine learning to agricultural domains. The three kinds of metadata information arising from this context are then discussed alongside real-world examples that we have processed with our support tool. We conclude with a discussion of the problems we have encountered and the ways these types of metadata information could be used in the future.

2. Computational Context

The process model we use for developing machine learning applications is presented as a data flow diagram in Figure 1. We receive data from a data provider (in our work, either agricultural research institutes or private companies). They supply data, knowledge of the data, and the context in which it was obtained; we provide expertise in applying machine learning schemes to that data. The data itself comes to us in a variety of formats, such as spreadsheets, relational database tables, and files of records generated by a programming language. Eventually, each of these formats is converted to a single table with a common attribute/value format.

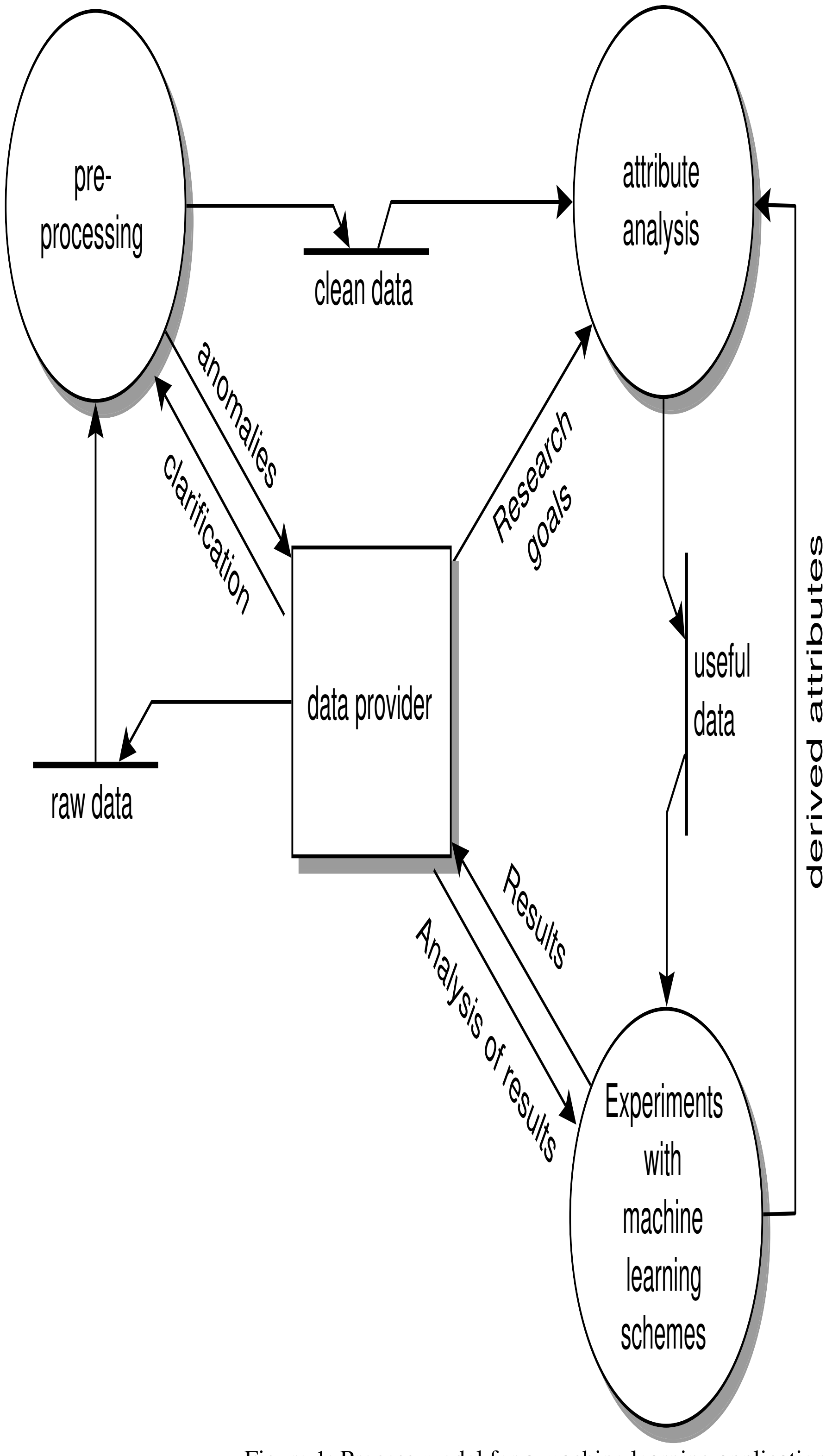

Figure 1: Process model for a machine learning application

The process flows in a clockwise direction around the model presented in Figure 1. At each of the major stages of processing information about the data is required at the metadata level; however, it is only at the attribute analysis stage that this information is currently implemented.

In pre-processing we need to understand, for example, what missing values for an attribute really mean (ie, are they crucial), whether or not it make sense to compute an average for a given attribute, and so on. This information is not used at this stage, but is carried forward to the next stage where it can have an impact on the selection of data items to be presented to the machine learning schemes.

The attribute analysis phase is where metadata information in our model has the most impact. Here we can use the information garnered from the data providers about the meaning of the data together with more data *implicit* information such as whether a given attribute can be manipulated mathematically, and whether co-dependencies and/or causal relationships exist between attributes. This information has a direct bearing on the views we provide the machine learning schemes, and our experience has shown that the better the view the more likely a good result will emerge.

The final process of performing experiments usually results in the need for more information about the data so that new data can be tried and new experiments can be run. Thus the final stage includes aspects of the other two processes, and means that some of the metadata information must be supplied dynamically as new datasets are prepared for experimentation.

We have developed two pieces of software to support this process model: an attribute editor and an experiment editor. The attribute editor allows a user to compute derived attributes from existing data, while the experiment editor allows a user to run several machine learning schemes over several instances of a dataset. The two programs share a symbiotic relationship, but only the attribute editor processes metadata-level information. This aspect of the attribute editor is described in Section 4.

In this paper we focus our attention on the derivation of new attributes from existing data, and the process of learning rules (for machine learning algorithms). Parenthetically, metadata is also important for data visualisation. In our initial attempts to get a feel for the data, we wish to know whether it makes sense to display certain attributes in particular ways; for example, whether two attributes should be plotted against each other, or which members of a set of attributes could be presented effectively in a table. The problem of developing a system that can use metadata in constructing effective data visualizations is considered in [5], and we are currently attempting to incorporate this functionality into our machine learning workbench.

3. Types of Metadata Information

Data type information

Conventional database system catalogues contain basic data type information (such as whether an attribute is real, integer, string, or a date). From these basic types it is possible to derive certain properties that are useful to know, and when they are not derivable we need a mechanism for specifying the properties of the data types directly.

Consider the basic division of data into continuous and discrete types. For continuous data (of type real) we know that the type is numeric and ordered but we do not always know if a type has a zero point or that it is linear (radians are real but not linear), and so we would have to rely on a user specifying this information.

The situation is worse for discrete data types. Discrete data may or may not be ordered, may or may not have a zero point, may or may not be linear and might be numeric, alphabetic or enumerated. Further, if the data is ordered then this implies that a high value and a low value of that type define a range. Each of these properties will have an impact on the operations that we can and cannot perform which is especially important when deriving new values from existing data.

At this point, we present three examples illustrating issues in data type information handling for machine learning:

- The data type *age* (in years) is discrete, ordered, has a zero point and is linear. The fact that a data type is discrete allows us to group data together so that we can compute a new attribute derived from one or more existing attributes (vegetarians grouped by age, for example). Ordered data can be meaningfully separated by the use of relational operations such as less than or greater than. Zero points indicate that a collection of data can be scaled or that absolute values can be taken, and linearity enables averages, standard deviations and basic arithmetic operations to be performed on the data.

- The data type *date* (also in years) is discrete, ordered and linear but does not contain a zero point. The operations on this type are, therefore, more restricted than the *age* data type above. It is interesting to note, however, that we frequently compute a new attribute which is the difference of two *date* data types, and the type of the resulting attribute is *age*. Thus it is in this sense that, in this application, metadata is dynamically generated as the system is used.

- An enumerated data type is one of the easiest to mis-handle. Consider, for example, a data type *colour*□□which has symbolic information coded as integers (red1=, blue = 2, and so on). It is clear that mathematical operations should not be performed on this data even though it is possible to do so. A machine learning scheme will happily split this data into categories which are greater than blue and less than orange, without any reference to whether or not this splitting makes any sense. To overcome this problem we have to be very careful with selecting and manipulating enumerations, and when necessary to force the machine learning algorithms to disregard arithmetic operations (for example, by converting the coded integers into symbols before invoking the machine learning schemes).

Relational metadata

We define three types of relational metadata, each of which specifies a relationship between two or more attributes: meaningful, causal, and functional. This type of metadata works as a constraint system on the rules that a machine learning program generates.

A meaning relationship between two attributes x and y indicates that if y is included in a rule discovered by a machine learning scheme, then x should also be present in that rule. Further, if y is the class of things we are trying to learn then x should be used in all of the rules. In this situation the person directing the machine learning experiments knows *a priori* that two attributes only make sense when used together. For example, in some agricultural data that we have analysed there is an attribute *milk production* which measures how much milk an individual cow produces, and this attribute has a meaning relationship with three other attributes: *cow-identifier*, *herd-identifier* and *farmer-identifier*. In other words, a milk production value can only be understood in the context of the cow which produced the milk, and the cow is further linked to a specific herd owned by a given farmer.

Causal relationships denote that x causes y, and so in a system that is trying to predict y, for example, we know that we have to include x in order to do this prediction meaningfully. In many applications a whole causal chain exists. For example, in the dairy data there exists a chain from the farmer, herd and cow identifiers through measured attributes such as milk production down to the attribute that records whether a particular cow was retained or sold by the farmer. Rules which fail to establish this chain or which attempt to break it are meaningless.

Functional relationships occur in many databases. In database theory attempts are made to locate these dependencies for the purposes of normalisation. In our application the meaning of a functional dependency is that if x is discovered in part of a rule then there is no need to consider y. Many machine learning schemes will often attempt to rediscover functional dependencies that are already known. The effect of this is twofold: meaningless rules are generated, and other patterns which are not part of functional dependencies will be ignored in favour of the functionally related items.

Statistical metadata

Statistical metadata information on attribute values is helpful for massaging data for more effective analysis: for example, the discovery of outliers (data points that have highly atypical values and which could represent data processing errors). This data has to be specified in a meaningful way so that only true outliers are discovered and removed from consideration.

Most machine learning schemes require the class attribute (the right hand side of a learned rule) to be discrete. This attribute is often continuous in the raw data, and must be discretized by making use of distribution and standard deviation information to form appropriate discrete blocks of data values. A data set may not have a *natural* attribute to use as the class, and so one must be found by searching through the attributes for those that are best predicted by other attributes in the data set. If statistical metadata is specified describing how each attribute can be discretized then this process will not only be faster, but will also be more likely to generate acceptable results.

The attribute editor we describe in Section 4 provides the weakest support for this aspect of metadata specification. We are currently working on a program to automatically discover natural classes in datasets by making direct use of statistical metadata.

Operations on metadata

Metadata has an influence on the derivation of attributes and the generation of rules from the machine learning component of our system. In terms of rule generation, the metadata provides the following constraints on the relational operations that are used in the left hand sides of the rules:

Property	Relational Operations that make sense
ordered	greater than or less than
zero point	(greater than or equal to zero) or (less than or equal to zero)
discrete	equal to

plus any combination of these properties.

For deriving new attributes we use the properties of the base attributes to constrain the valid operations that can be used to form new (and meaningful) attributes:

Property	Valid Operations
ordered	discretize to form new type that is discrete and ordered
linear	grouping, average, standard deviation, etc to form new type that is also linear. Some operations may change they type (for example, difference operations may not produce a linear type
zero point	absolute values and scaling.
discrete	grouping by values.

plus any combination of these properties.

4. Attribute editor

Traditional database catalogues are stored as relations so that query languages can be used to query, update and maintain the catalogue. The information they contain is typically in the form of descriptions of the relation names, attribute names, attribute domains (their data types), keys and indexing information, security information, and so forth. Given the dynamic nature of the metadata information necessary for a successful application of machine learning techniques, it seems prudent to maintain this organisation even though the catalogue does not necessarily describe a relational database system.

It is important for us to track the evolution of our data through the various experiments that are performed on it, just as it is important for a database system to document the database design process. However, this type of facility is more commonly associated with a data dictionary, which is a more general purpose utility than a system catalogue. In our system, experiments are documented alongside the results they produce, and so our implementation of metadata specification is aligned more with a system catalogue than a general data dictionary system.

The attribute editor for our WEKA machine learning workbench is shown in Figure 2 below. The topmost section describes the data set under consideration, followed by a description of each of the attributes in the data, their type, and whether they are part of the original data or derived from it. The user is invited in the "Formula" section of the screen to define a new attribute, using elements selected from the "Operators" and "Functions" sections.

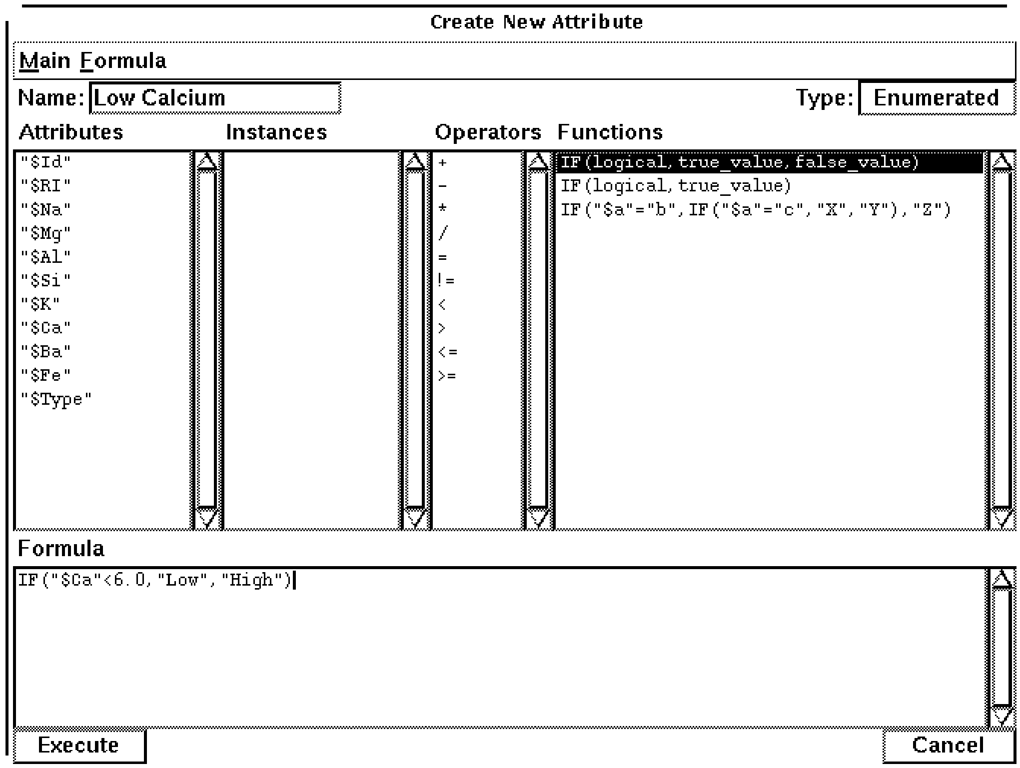

Figure 2. Attribute editor

5. Conclusion

At present, successful machine learning applications depend on a careful hand-crafting of appropriate tables by researchers who are aware of the (often not formally recorded) metadata affecting a given database. Our attribute editor currently hard codes information on data types and data values in base types. Other, more complex, metadata that is known by the user is supplied interactively when creating new attributes. In some cases, this metadata is still implicit in the tables supplied to machine learning schemes. For example, the fact that a particular enumerated attribute cannot be used comparatively (that in this application it makes no sense to state that color blue < red) is indicated by the encoding of that attribute value in symbols, which are generally not ordered by machine learning schemes, as opposed to encoding color values as integers, which are usually treated as order-able.

In the future, we would like to explicitly record more of the metadata associated with a database, and to provide a higher degree of automation in selecting appropriate attribute sets for processing by machine learning algorithms. In particular, we hope to provide warning messages when potentially erroneous attribute values and attribute combinations are selected. A larger problem lies in the extremely sparse amount of metadata that is utilized by the standard machine learning schemes; they take into account very little information about the relational and type aspects of the attributes being processed, and instead rely on human pre-processing to remove potential problems and nudge the algorithms into the most promising analysis paths.

References

[1] Piatetsky-Shapiro, G., and Frawley, W. J., eds. (1991) *Knowledge Discovery in Databases*. Menlo Park, CA: AAAI Press.

[2] Holmes, G., Donkin, A., and Witten, I.H. (1994) "Weka: a machine learning workbench." *Proceedings of the Second Australian and New Zealand Conference on Intelligent Information Systems*, Brisbane, Australia, pp 357-361.

[3] Garner, S., Cunningham, S.J., Holmes G., Nevill-Manning, C.N., and Witten, I.H. (1995) "Applying a Machine Learning Workbench: Experience with Agricultural Databases." *Machine Learning in Practice Workshop of the 12th International Conference on Machine Learning*, Tahoe City, California, U.S.A.

[4] Rosenthal, A., and Reiner, D. (1994) "Tools and transformations — rigorous and otherwise — for practical database design." *ACM Transactions on Database Systems 19 (2)*, 167-211.

[[5] Humphrey, M. (1996) *A graphical notation for the design of information visualization.* D.Phil. thesis, Department of Computer Science, University of Waikato (Hamilton, New Zealand).