Techniques for Dealing with Missing Values in Classification

W.Z. Liu[1], A.P. White[2], S.G. Thompson[1] and M.A. Bramer[1]

[1] Artificial Intelligence Research Group, Department of Information Science, University of Portsmouth, Locksway Road, Milton, Hampshire PO4 8JF, U.K.
[2] School of Mathematics and Statistics, University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K.

Abstract. A brief overview of the history of the development of decision tree induction algorithms is followed by a review of techniques for dealing with missing attribute values in the operation of these methods. The technique of dynamic path generation is described in the context of tree-based classification methods. The waste of data which can result from casewise deletion of missing values in statistical algorithms is discussed and alternatives proposed.

Keywords: Missing values, Dynamic path generation, Intelligent data analysis, Inductive learning, Knowledge discovery, Data mining, Machine learning.

1 Introduction

In the information age, data is generated almost everywhere: satellites orbiting the moons of Jupiter; submarines in the deepest ocean trench; even electronic point of sale machines in the high street produce data. All of these systems generate millions of megabytes of data every day. Some of these data contain information that could lead to important discoveries in science; some data contain the knowledge that could predict a company's growth or collapse and other data contain knowledge that could mean the difference between life and death.

In order to analyse these important data and uncover hidden relationships and knowledge within the data, some sort of data mining approach is required. In the past, statistical methods such as logistic regression or discriminant analysis were the only tools available for such a task. Unfortunately, they are somewhat cumbersome in the sense that the *form* of the model needs to be specified beforehand, which is often not really feasible for an exploratory analysis involving a large number of variables. More recently, the massive increase of interest and research in this area has made a number of innovative techniques available, which have their origins in computer science, rather than mathematical statistics. These techniques include tree-based methods, neural networks, genetic algorithms, case-based reasoning and so on. They offer the possibility of automating the process of knowledge discovery to a greater degree than appears possible with traditional statistical approaches. Because of their greater simplicity and transparency, tree-based classification techniques are of particular interest in this context.

Unlike some of the other newer techniques, tree-based classification methods have two points of academic origin. The first of these was the study of inductive learning. The influence of computer science in this field did not develop until the second half of this century.[3] Hunt was one of the early pioneers who modelled a theory of human concept learning using computer programs. He developed a series of algorithms called 'concept learning systems' (CLS-1 to CLS-9), described in Hunt (1962) and Hunt et al. (1966). Quinlan's well-known ID3 algorithm (Quinlan, 1979), was descended from these systems. Basically, ID3 was a procedure for discriminating between two classes in domains which were entirely free from uncertainty. (In fact, ID3 was developed initially for performing chess endgame analysis, discriminating between winning and non-winning positions).

The second point of origin lay in the discipline of mathematical statistics. It is interesting to note that, among mathematical statisticians, there has been a minority interest in these techniques for the last thirty years. However, this interest did not really come to the fore until the last twelve years or so, prompted by the work of Breiman et al. (1984) and the associated CART software for performing classification and regression using binary trees. In the last few years, tree-based classification and regression procedures have been incorporated into multi-purpose statistical software packages. For example, the statistical package 'S', recently developed for use by statisticians themselves and described by Clark & Pregibon (1992), includes a set of procedures for conducting classification and regression tasks by the use of binary trees. Similarly, the well-known package SPSS has recently become available with CHAID (CHi-squared Automatic Interaction Detector). This is based on earlier work by Kass (1980), which used multiple, rather than binary, branching.

In the statistical field, tree-based methods dealt with uncertainty from the beginning but, in computer science, the adaptation of this type of algorithm to deal with noisy domains took place much more recently. Quinlan (1986) extended the ID3 system, producing the C4.5 algorithm, to deal with the usual statistical situation in which the attributes (independent variables) provide probabilities of class membership, rather than definitive indications. Initially, computer scientists were unaware of the penalties of constructing over-large trees, which is actually equivalent to constructing models with more than an optimal number of parameters, which is understood by statisticians as 'overfitting'. However, Quinlan (1986) rediscovered the problem and dealt with it by incorporating a pruning phase into the algorithm. Thus, an over-large tree was grown to begin with and then cut back to protect against overfitting.

In retrospect, it is obvious that the applicability of the first generation of knowledge discovery systems of computer science ancestry, such as ID3, was very limited. In fact, they could be applied only in deterministic domains, such as chess endgame analysis, in which there is no noise or uncertainty involved. Now, it is increasingly apparent that, in order for a knowledge discovery system to be able to deal with real-world applications, it must be able to handle noise. This is

[3] A more detailed review of these methods from a computer science perspective is given in Liu & White (1991).

because noise is inevitable in most real-world applications. The data collected in the real-world are based either on measurements or subjective judgements. Both of these are subject to error. In order for the knowledge extracted from the data to be useful in helping future decision making, the knowledge obtained must be based on intrinsic relationship or structure in the data, rather than some *ad hoc* features of the data such as noise. A less obvious source of error lies in the relationship itself, which links the dependent variables with the attributes. In many real-world examples, the independent variables available provide only an incomplete indication of the value of the dependent variable, even when no errors are present in the dataset itself. Statistical models typically concatenate all these sources of error and express them as a single *error term* on the right-hand side of an equation specifying the model.

About ten years ago, Quinlan (1986) made some useful modifications to ID3 to deal with noise. He pointed out that two modifications to ID3 are necessary if it is to be able to operate with a noise-affected training set:

1. The algorithm must be able to deal with clashes (when two or more cases have identical values for each attribute but belong to different classes);
2. The algorithm must be able to decide when the testing of further attributes will not improve the predictive performance of the decision tree, i.e. to determine when to stop adding further branches to the decision tree.

The first goal is achieved by using probabilistic induction. When the branching process stops, if the cases at any given terminal node are not all of the same class, then probabilities for membership of the various classes are assigned instead. The conversion of these probabilities to predictions of class membership may then be done either by using the obvious strategy of selecting the most likely class at each terminal node or, if differential mis-classification costs are operating, either by some sort of cost minimisation procedure such as described by Breiman et al. (1984), or else by selecting an appropriate discrimination point on an ROC curve (or its equivalent), in the manner described elsewhere by Liu et al. (1994, 1996) and White & Liu (1997). There are two possible ways to achieve the second goal. What Quinlan suggested doing, is to use some kind of 'stopping rule' to prevent over-large decision trees being grown. The second solution to the problem is to grow an over-large tree to begin with, and to prune it back to the right size.[4]

The various techniques for dealing with uncertainty are very important and, in the past decade more and more research has been focused on problems in this area. However, comparatively little attention has been paid to methods of handling some special types of noise, such as missing values. Where data have been collected for a particular purpose, known beforehand, it is often possible to minimise, or even completely avoid, the occurrence of missing values for data items. On the other hand, where data are collected as a by-product of some other activity and subsequently subjected to some sort of data mining operation,

[4] A review of pruning techniques can be found in Mingers (1989).

missing values are much more likely to be present in substantial proportions. The intention of this paper is to review and summarise techniques for dealing with missing values that are used in tree-based classification methods and to discuss the possibility of adapting these techniques to other knowledge discovery approaches.

2 Decision-Tree Based Inductive Learning

The principle of tree-based inductive learning (Quinlan, 1986; Liu & White, 1991) is well-known. Basically, the idea is to build a learning algorithm to induce classification rules in the form of a decision tree, by operating on a training set. A *training set* usually consists of a set of past decision-making examples, each of which is comprised of a number of attributes (variables) and a class membership indicator. The decision tree obtained can then be used to classify future cases of unknown class membership.

The task of constructing a decision tree from a training set is typically handled by a recursive partitioning algorithm which, at each non-terminal node, branches on that attribute which discriminates best between the cases filtered down to that node. In order to decide which attribute to select to branch on, some suitable attribute selection measure is needed (Liu & White, 1994). There are many such measures which can be used for this purpose, such as transmitted information[5] and χ^2. Definitions for both these measures are given in White & Liu (1994, 1997). The importance of these criteria lies with their ability to measure the association between the class and the other independent variables. This enables the induced tree to reflect the classification structure of the original data.

In situations where there are no missing values in the training set, tree building can proceed in the expected manner. However, if missing values do exist in the training set, the way these missing values are dealt with will have some effect on the tree building process. In the next section, various techniques for handling missing values in such situations are reviewed.

After obtaining a classification tree, the next step is to use the tree to predict the class membership of test cases. Again, this is very simple if there are no missing values for the attributes of the case undergoing classification. However, if the value of a particular attribute is required in order to classify a particular case and that attribute has a missing value for that case, then simple classification immediately becomes impossible because we do not know which branch to take in order to classify the case. In order to carry out classifications under these circumstances, other methods for handling missing values have to be used. Some of these techniques are described in Section 4.

[5] Transmitted information (H_T) is actually algebraically equivalent to information gain, as described by Quinlan (1986). However, its formulation in the former terms is particularly useful because it represents information about class membership transmitted by the attribute concerned.

3 Dealing with Missing Values in Training Cases

As mentioned earlier, at each non-terminal node of the decision tree, that attribute which gives the strongest association with class is selected to branch on. In situations when there are no missing values for an attribute, the calculation of association between class and that attribute is quite simple. It starts with cross-tabulating class against that particular attribute in the following way. Suppose that we are dealing with a problem with k classes and that an attribute, A, with l distinct values is under consideration at a particular node. The following contingency table (Table 1) can be constructed which represents the cross-tabulation of class and attribute values for A:

Table 1. A cross-tabulation of class and attribute values, for attribute A.

	a_1	a_2	$\dots$	a_l	
C_1	n_{11}	n_{12}	$\dots$	n_{1l}	n_{1*}
C_2	n_{21}	n_{22}	$\dots$	n_{2l}	n_{2*}
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
C_k	n_{k1}	n_{k2}	$\dots$	n_{kl}	n_{k*}
	n_{*1}	n_{*2}	$\dots$	n_{*l}	n_{**}

where C_i (i=1, 2, ..., k) and a_j (j=1, 2, ..., l) represent class and attribute values respectively; n_{ij} (i=1, 2, ..., k; j=1, 2, ..., l) represent the frequency counts of cases with attribute value a_j and class C_i and:

$$n_{i*} = \sum_{j=1}^{l} n_{ij}$$

$$n_{*j} = \sum_{i=1}^{k} n_{ij}$$

$$n_{**} = \sum_{i=1}^{k} \sum_{j=1}^{l} n_{ij} = N$$

There are several ways to get around the problem of missing values of cases in the training set. Obviously, the simplest way to deal with unknown attribute values is just to ignore the cases containing them and base the calculation of association on the contingency table constructed from only those cases which have known values on this attribute. This is the method used in PREDICTOR (White, 1987).

The second type of technique in dealing with missing values is to try to determine these values using other information. For example, Kononenko et al.(1984)

used class information to estimate missing attribute values. Let us assume that the case with missing value on attribute A is of class C. The idea is to assign the most probable value, a_i, of attribute A to the missing value, given the class membership of the case concerned. Another method suggested by Shapiro and described by Quinlan (1986) is to use a decision tree approach to decide the missing values of an attribute. It considers the subset S', of the training set S, which consists of those cases whose value of attribute A is known. In S', the original class is regarded as another attribute while the value of attribute A becomes the 'class' to be determined. Using S', a classification tree can be built for determining the value of attribute A from the other attributes and the class. Then, this tree can be used to classify each object in the set $S - S'$. Consequently, each missing value can be estimated. This is a very thorough technique and makes good use of all the information available from the class variables and all the other independent variables. However, it would appear that the technique is appropriate only for sparse concentrations of missing values. Difficulties arise if the same case has missing values on more than one attribute.

Quinlan (1986) proposed another two different methods. The first method is to treat 'unknown' as a new possible value for each attribute and deal with it in the same way as other values. However, this is appropriate only when the missing values are informative, e.g. values recorded as missing because they were too small or too large to be measured. Usually, missing values are missing at random and, in these circumstances, the value 'unknown' does not have the same status as a proper attribute value, i.e. whether or not a particular attribute has a known value for a particular case does not provide information about class membership of that case. Thus, this method cannot really be regarded as a *general* solution to the problem of missing values. The second method is based on the idea that cases with unknown values are distributed across the values of A in proportion to the relative frequency of these values in the training set. Consider a simple 2×2 contingency table with similar notation to that described earlier, with m_i ($i = 1, 2$) cases with missing value on attribute A, for each class respectively. Then each frequency count of the contingency table is adjusted as follows:

$$n'_{ij} = n_{ij} + m_i \frac{n_{*j}}{n_{**}}$$

where $i, j = 1, 2$. The attribute selection criterion is then calculated using the adjusted frequency counts. When an attribute has been chosen by the selection criterion, cases with unknown values of that attribute are discarded before going to the next step of branching. This method can be too conservative. The following example shows how it attenuates association between attribute and class. Consider the following 2×2 contingency table of those cases whose value on attribute A is known:

	a_1	a_2	
C_1	5	0	5
C_2	0	5	5
	5	5	10

where C_i and a_i ($i = 1, 2$) represent class and attribute values respectively. Suppose there are another five cases of class 1 and five cases of class 2 with missing values on attribute A. Then, if we adjust the frequency counts according to the column proportions (as in the formula above), the following table can be derived:

	a_1	a_2	
C_1	7.5	2.5	10
C_2	2.5	7.5	10
	10	10	20

The χ^2 and H_T of the first table are 10 and 1, while those of the second table are only 5.556 and 0.236 respectively. This is obviously undesirable and misleading.

The reason why this method can give such unsatisfactory estimates for missing values is revealed if we take a statistical view of the process involved. To put the matter simply, the procedure takes no account at all of the structure in the data set. Missing value estimates are assigned merely on the basis of prior attribute value probabilities. Kononenko's method is better, because the estimates are made conditional upon class membership. Thus, in the example just considered, the adjusted frequencies become:

	a_1	a_2	
C_1	10	0	10
C_2	0	10	10
	10	10	20

This is clearly preferable.

4 Dealing with Missing Values in Test Cases

The other half of the story is how missing attribute values are dealt with during classification of test cases. When classifying a case, if the value of a particular attribute which was branched on in the tree is unknown, then classification immediately becomes impossible because we do not know which branch to take in order to classify this case. In order to carry out classifications under these circumstances, other methods for handling missing values have to be used.

The procedure implemented by Quinlan (1986) is to explore all branches (below the current node) and take into account that some are more probable than others. This seems to be very clumsy and unsatisfactory. The other method suggested by Breiman et al. (1984) is to use a *surrogate split* when a missing value is found in the attribute originally chosen. The surrogate attribute is the one which has the highest correlation with the original attribute. The efficacy of this method obviously depends on the magnitude of the correlation in the database between the original attribute and its surrogate.

There is another method, called *dynamic path generation*, proposed by White (1987), which can offer great flexibility in dealing with missing values of this type. Instead of generating the whole decision tree beforehand, the dynamic path generation method produces only the path (i.e. the rule) required to classify the case currently under consideration. This approach can deal with missing values very flexibly. Once a missing value is found to be present in an attribute of a new case, such an attribute is never branched on when classifying the case. In more detail, let us consider the process of building a classification rule (i.e. a path in a classification tree) to classify a new case O_1. At each step, the inductive algorithm chooses the most informative attribute on which to branch. However, if the value of the selected attribute is missing in case O_1, then this attribute cannot be branched on and the algorithm tries with the second most informative attribute. Thus, path generation is strictly *dynamic*. Of course, this approach is somewhat expensive in computational terms. However, if N-fold cross-validation is required, then the technique becomes much more economical, in comparative terms (Liu & White 1994). This is because, for N-fold cross-validation, a fresh model needs to be constructed for each case, whatever method is used. When combined with dynamic path generation, only a fresh *path* needs to be constructed for the classification of each case. In other words, with dynamic path generation, cross-validation imposes no extra cost penalty.

The approach of dealing with missing values in test cases in this way is also referred to as the *lazy decision tree* method (Friedman et al., 1996). The reason why this approach is called the lazy decision tree approach is because the creation of a single 'best' tree is deferred. Instead, it constructs the 'best' tree for each test instance. (In fact, only a classification path needs to be generated).

5 Discussion

In many real-world applications, missing values are often inevitable. Therefore, every intelligent data analysis tool should be equipped with facilities to deal with missing values. Unfortunately, many systems which have been built so far still have very limited power in dealing with missing values. For example, most orthodox statistical packages deal with missing values on a casewise deletion basis, for most statistical procedures. This means that if *any* of the available variables has a missing value for a particular case, then that case is omitted from the analysis. Clearly, this may cause a huge waste of data and, as a result, may not be satisfactory in some circumstances. For example, in the medical database reported by White et al. (1996), if the casewise deletion method is used, there are only 632 cases consisting entirely of non-missing values – fewer than a quarter of the 2692 cases available in the original database. By contrast, if the dynamic path generation method is used, missing values can be dealt with very simply.

In fact, some of the techniques for dealing with missing values in decision tree induction (reviewed in the previous sections) can be easily adapted to many other model-based methods of data analysis. Take some statistical classification method such as linear discriminant analysis or logistic regression as an example. It is clear that:

- If an overall model is required in the training phase, then it is always possible to estimate missing values by one of the techniques mentioned in Section 3.
- In order to deal with missing values in test cases, the 'lazy' approach could be easily adapted. Instead of producing a single set of linear discriminant functions (or a regression equation in the case of logistic regression) in the training phase, we could construct a set of discriminant functions or a regression equation for each test instance. This would ensure that variables with missing values on a particular test case did not occur in the model constructed to classify that case. In this way, missing values are handled very naturally.

To conclude, there is no fundamental reason why the lazy approach could not even be extended to other techniques, such as genetic algorithms, in order to prevent waste of information. Of course, the lazy approach can be expensive in computational terms. However, with modern computer technology this is becoming less and less of a problem.

References

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and regression trees.* Belmont: Wadsworth.

Clark, L.A. & Pregibon, D. (1992). Tree-based models. In *Statistical Models in S*, edited by J.M. Chambers & T.J. Hastie, pp. 377–419. California: Wadsworth & Brooks/Cole.

Friedman, H.F., Kohavi, R. & Yun, Y. (1996). Lazy decision trees. in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 717–724, AAAI Press/MIT Press.

Hunt, E.B. (1962). *Concept learning: an information processing problem.* New York: Wiley.

Hunt, E.B., Marin, J. & Stone, P.J. (1966). *Experiments in induction.* New York: Academic Press.

Kass, G.V. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, **29**, 119–127.

Kononenko, I., Bratko, I. & Roskar, E. (1984). Experiments in automatic learning of medical diagnostic rules. *Technical Report.* Jozef Stefan Institute, Ljubjana, Yugoslavia.

Liu, W.Z. & White, A.P. (1991). A review of inductive learning. In *Research and Development in Expert Systems* VIII, edited by I.M. Graham and R.W. Milne, pp. 112–126. Cambridge: Cambridge University Press.

Liu, W.Z. & White, A.P. (1994). The importance of attribute selection measures in decision tree induction. *Machine Learning*, **15**, 25–41.

Liu, W.Z. White, A.P. & Hallissey, M.T. (1994). Early screening for gastric cancer using machine learning techniques. In *Machine Learning: ECML-94*, edited by F. Bergadano and L. De Raedt, pp. 391–394. Springer-Verlag, Berlin.

Liu, W.Z., White, A.P., Hallissey, M.T. & Fielding, J.W.L. (1996). Machine learning techniques in early screening for gastric and oesophageal cancer. *Artificial Intelligence in Medicine*, **8**, 327–341.

Mingers, J. (1989). An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, **4**, 227–243.

Quinlan, J.R. (1979). Discovering rules by induction from large collections of examples. In *Expert Systems in the Micro-Electronic Age,* edited by D. Michie, pp. 168–201. Edinburgh: Edinburgh University Press.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, **1**, 81–106.

White, A.P. (1987). Probabilistic induction by dynamic path generation in virtual trees. In *Research and Development in Expert Systems* **III**, edited by M.A. Bramer, pp. 35–46. Cambridge: Cambridge University Press.

White, A.P. & Liu, W.Z. (1994). Bias in information-based measures in decision tree induction. *Machine Learning*, **15**, 321–329.

White, A.P., Liu, W.Z., Hallissey, M.T. & Fielding, J.W.L. (1996). A comparison of two classification techniques in screening for gastro-oesophageal cancer. *Applications and Innovations in Expert Systems* IV, edited by A. Macintosh and C. Cooper, pp. 83–97. Cambridge: Cambridge University Press.

White, A.P. & Liu, W.Z. (1997). Statistical properties of tree-based approaches to classification. In *Machine Learning and Statistics: the Interface*, edited by R. Nakhaeizadeh and C. Taylor, pp. 23–44. ISBN 0-471-14890-3, John Wiley & Sons, Inc.