Discovering Inter-Attribute Relationships

Geoffrey Holmes[1]

[1]Department of Computer Science, University of Waikato, Dunedin, New Zealand
geoff@cs.waikato.ac.nz

Abstract

It is important to discover relationships between attributes being used to predict a class attribute in supervised learning situations for two reasons. First, any such relationship will be potentially interesting to the provider of a dataset in its own right. Second, it would simplify a learning algorithm's search space, and the related irrelevant feature and subset selection problem, if the relationships were removed from datasets ahead of learning. An algorithm to discover such relationships is presented in this paper. The algorithm is described and a surprising number of inter-attribute relationships are discovered in datasets from the University of California at Irvine (UCI) repository.

1 Introduction

In applied machine learning a dataset is viewed as a list of items (the size of the list can be anything from a few dozen to several thousand), each item consisting of a number of attribute values and an associated classification. The attributes can be numeric, ranging over the set of real numbers, or nominal, ranging over some finite set of values. For some items some of the attribute values might be missing. Practical algorithms have been developed to take such a list and compute a model or theory of the data as a set of rules, or as a tree that branches on attribute values to leaf nodes labelled with classification values.

In our experience of applying such algorithms to "real-world" datasets [Garner *et al.*, 1995] it has become apparent that a natural classification for a given dataset may not be readily available. Data providers may not know which attribute to use for classification, and may not be necessarily looking for a theory that describes their data in its entirety. Rather, they may be looking for information about how the attributes within a dataset relate to each other.

Aside from the knowledge discovery aspects of finding inter-attribute relationships, there are technical reasons why such relationships should be found and eliminated from the machine learning process. Feature subset selection algorithms [John *et al,* 1994] provide evidence that the task of performing supervised machine learning can be substantially improved if irrelevant (and in some cases even relevant attributes) are removed. Intuitively, one would expect all machine learning algorithms to benefit from a simpler search space, where they would operate faster and induce better models of their data.

Further, with recent advances in empirical learning to the problem of numeric class prediction [Quinlan, 1992; Wang and Witten, 1997] there is now no need to perform unsupervised quantisation on numeric data prior to learning. Because quantisation is unsupervised, illogical classes tend to be generated [Dougherty *et al.*, 1995] and the knowledge discovery aspect of the process is lost. The output of the numeric classifier used in this paper is a model tree, a binary decision tree with linear regression functions at the leaf nodes, which can be viewed for the knowledge it contains.

The next section describes the algorithm for inter-attribute discovery, its treatment of missing values, and a worked example of the discovery process. Experimental results of the algorithm applied to a number of standard datasets taken from the UCI collection [Merz and Murphy, 1996] are reported in Section 3, and discussed in Section 4.

2 Discovery Algorithm

The algorithm for discovering inter-attribute relationships involves two iterations. The first iterates through a dataset setting each attribute in turn as the class attribute using the remaining attributes to predict the chosen class. The second iteration occurs once an attribute has been chosen as the class. This iteration is a tenfold cross-validation step to ensure that the inter-attribute relationships are not spurious relationships that only hold for particular train and test datasets.

The type of the attribute chosen as the class determines which empirical learning scheme will be used in the cross-validation. If the type is numeric then the M5' [Wang and Witten, 1997] algorithm is used to predict the continuous values of the class. Otherwise the type of the class attribute is nominal and C4.5 [Quinlan, 1993] is used. These algorithms are used for two reasons. First, they are both state-of-the-art and will therefore predict, on average, as well as any other empirical learning scheme. Second, they both produce interpretable output which is essential for finding out the nature of the inter-attribute relationship. If a "black-box" was used instead then it would be possible to discover that a relationship existed, and with some effort what the dependent attributes were, but it would not be possible to say what the relationship was precisely.

Each scheme outputs results of one cross-validation run using a randomly selected training and test set from the data using the chosen attribute as the class. Ten of these runs are performed and the average predicted error

is written to a file along with the name of the attribute and, if the attribute is nominal, the default accuracy for that attribute. Once the algorithm has finished these results are examined to see if any "low" error scores are present. Zero error implies that it is possible to completely predict one attribute from some of the others. Small error rates (in the zero to one percent range) are worthy of pursuit as there are probably only one or two contradictory examples in the dataset and these outliers could either be data entered incorrectly or genuine exceptions to an interesting rule.

2.1 Discovery Shell Script

The current version of the discovery algorithm has been implemented as a Unix shell script. An outline of the script is given below:

```
set class=1
while ( $class <= $num_attributes )
  class_type = get_type( class )
  randomise_dataset
  if (class_type == numeric)
    if (missing_values_in_class) delete_missing
    foreach f (1 2 3 4 5 6 7 8 9 10)
      split_dataset_into_train_test(train_$f, test_$f)
      m5 -c class -t train_$f -T test_$f >> logfile
    end
    output class, avg_error_from_logfile
  else
    if (missing_values_in_class) add_missing_to_class
    foreach f (1 2 3 4 5 6 7 8 9 10)
      split_dataset_into_train_test(train_$f, test_$f)
      c4.5 -c class -t train_$f -T test_$f >> logfile
    end
    output class, avg_error_from_logfile, default_accuracy
  endif
  set class=class+1
end
```

2.2 Missing Values

As can be seen, the treatment of missing values and the output reported is different in each of the two attribute types (it is unclear at present what, if any, computation could be returned in the numeric attribute case to indicate the significance of the error rate). If the class attribute is numeric and it contains missing values then those values are removed from the randomized dataset prior to cross-validation. Missing values occur in datasets for many reasons [Quinlan, 1989]. For example, data may not be available for a particular attribute because it may not be relevant. For continuously-valued attributes there is little that can be done with such "information".

For nominal attributes it is possible to treat missing values as a further enumerated value. This provides more information for the post-shell script review of results than merely discarding the missing values. In this case it is possible to discover related values in other attributes to missing values in the class.

2.3 Worked Example

The "credit-rating" [Quinlan, 1993] dataset describes credit card applications made in Australia. There are 16 attributes in total including a "class" attribute which specifies whether or not a credit card was approved. The names of the attributes have been changed to meaningless symbols to protect confidentiality of the data. The dataset is interesting because it contains a good mix of attributes—continuous, nominal with small numbers of values, and nominal with larger numbers of values. There are also a few missing values, and so the dataset fully tests the discovery algorithm.

Table 1 shows the results of running the algorithm of Section 2.1 on this dataset.

Table 1. Results for credit-rating dataset

Attribute	Type	Avg Error Rate	Default Accuracy
A1	nominal	36.08	67.83
A2	numeric	70.22	
A3	numeric	73.40	
A4	nominal	0	75.20
A5	nominal	0	75.20
A6	nominal	72.92	19.90
A7	nominal	30.57	57.80
A8	numeric	65.17	
A9	nominal	13.89	52.30
A10	nominal	0	57.20
A11	numeric	66.74	
A12	nominal	42.75	54.20
A13	nominal	9.69	90.60
A14	numeric	87.19	
A15	numeric	80.66	
class	nominal	15.04	55.51

Many of the attributes in this table do not have any clear relationship to the others. This is true if either the average error rate is high (for numeric attributes), in this example this applies to the set {A2, A3, A8, A11, A14, A15} or if the sum of the average error rate and the default accuracy (for nominal attributes) is approximately 100%, in this example this applies to the set {A1, A6, A12, A13}. Attribute 13 nicely demonstrates the need to compute the default accuracy. Without this information it could be assumed that this attribute is worthy of further consideration, in fact, the default rule is used for each of the cross-validation runs. The nominal attributes that are worthy of further consideration are A4, A5, A7, A9, A10 and the class.

Attributes 4, 5 and 10 are predicted perfectly by one or more of the other attributes. The exact nature of the relationship is determined by running either C4.5 or M5' once more, using the entire dataset for training[1], and recording the result. In this case attributes 4, 5 and 10

[1] This provides a further test to verify that the relationship holds for all instances in the dataset. In small datasets, cross-validation can lead to the discovery of bogus relationships (see Section 4), and these are trapped at this stage in the process.

are nominal and so C4.5 is re-run. The relationships discovered are:

```
A4  = '?'   =>  A5 = '?'        (6)
A4  = 'u'   =>  A5 = 'g'        (519)
A4  = 'y'   =>  A5 = 'p'        (163)
A4  = 'l'   =>  A5 = 'gg'       (2)
A4  = 't'   =>  A5 = 'g'        (0)

A5  = '?'   =>  A4 = '?'        (6)
A5  = 'g'   =>  A4 = 'u'        (519)
A5  = 'p'   =>  A4 = 'y'        (163)
A5  = 'gg'  =>  A4 = 'l'        (2)

A11 <=  0   =>  A10 = false     (395)
A11 >   0   =>  A10 = true      (295)
```

The numbers in brackets are the number of instances covered by the rule. Attributes 4 and 5 share a symbiotic relationship, always occurring in pairs together, including those instances (6 of them) where their values are missing, justifying our approach (see Section 2.2). Attribute 11 completely determines attribute 10, but not *vice versa*. For classification purposes it is now no longer necessary to retain attributes 4 (or 5) and 10 in the dataset.[2]

3 Experimental Results

The algorithm outlined in Section 2 was applied to 16 datasets taken from the UCI repository. Table 2 shows a summary of the datasets used in the experiment. The range of datasets was chosen to test all aspects of the discovery algorithm. Most of the datasets involve a mixture of numeric and nominal attributes, most have missing values and there is a representative range of sizes.

Table 2. Datasets used for the experiment

Dataset	Instances	Missing values %	Numeric attributes	Nominal attributes
audiology	226	2.0	0	70
soybean-sm	47	0.0	0	36
anneal	898	65.0	6	32
autos	205	1.1	15	10
horse-colic	368	23.8	7	15
credit-rating	690	0.6	6	9
german	1000	0.0	6	14
lymphogphy	148	0.0	3	15
prim-tumor	339	3.9	0	17
soybean-lg	307	6.6	0	35
echocardio	132	7.7	10	3
mushroom	8124	1.3	0	23
hypothyroid	3163	6.5	7	19
chess	3196	0.0	0	37
brst-cancer	286	0.3	0	10
iris	150	0.0	4	0

[2] C4.5 does not use attributes 5 or 10 in its model of the class attribute, but does use attributes 4 and 11.

Table 3 shows the results of applying the discovery algorithm to the datasets in Table 2.

Table 3. Summary of relationships discovered

Dataset	Number of zero error attributes	Number of interesting attrs	Number of constant attributes
audiology	0	6	0
soybean-sml	19	13	14
anneal	8	7	18
autos	1	21	0
horse-colic	0	7	0
credit-rating	3	6	0
german	0	2	0
lymphogrphy	0	1	0
primry-tumor	0	2	0
soybean-large	1	8	0
echocardio	1	3	1
mushroom	8	10	0
hypothyroid	0	6	0
chess	2	12	0
breast-cancer	0	3	0
iris	0	2	0

Some of the zero error attributes in Table 3 are constant, the same value is used in each instance in the dataset. In the *echocardiogram* dataset the attribute *name* was made constant to protect patient confidentiality. The high number of attributes in the *soybean-small* and the *anneal* datasets which have only one possible value, are harder to explain, and remain a mystery.

The most complex relationships were found in the *mushroom* dataset (Table 4). It is clear that the relationship between the attributes in this dataset are very involved, and it would be very interesting to send these results back to the mycologist who first supplied the data.

Table 4. Dependencies found in the *mushroom* dataset

Attribute	Dependent attributes
ring_number	stalk_shape, spore_print_color, gill_size, odor
stalk_root	ring_type, stalk_color_below_ring, bruises?, gill_spacing, stalk_shape, gill_size, odor, ring_number stalk_surface_below_ring, stalk_surface_above_ring, class
ring_type	bruises?, stalk_shape, odor, gill_size, spore_print_color, ring_number, stalk_root
class	odor, spore_print_color, ring_number, gill_spacing, population
bruises?	ring_type, ring_number, gill_spacing, gill_attachment, stalk_surface_above_ring, gill_size, odor, stalk_shape, habitat
gill_size	class, habitat, gill_spacing, ring_number, gill_attachment, ring_type, spore_print_color,
stalk_shape	ring_type, bruises?, stalk_color_below_ring, stalk_root, ring_number, gill_spacing, class

In all, inter-attribute relationships were discovered in 8 of the 16 datasets. Some were "constants" discovered as a

side-effect of the algorithm. Non-constant relationships were found in the *soybean-small, autos, credit-rating, soybean-large, mushroom* and *chess* datasets. All of these relationships and the non-zero error but interesting attributes, should be pursued with the data providers to see if these results are of interest.

4 Discussion

All of the zero error results presented in Section 3 are related to the classification of nominal attributes. This is not due to the fact that C4.5 outperforms M5' as a classifier but has more to do with the fact that predicting a numeric class perfectly is intrinsically more difficult. M5' is a new class of algorithm that has only just emerged in machine learning, and so there is little experience of interpreting its results. Many of the attributes returning small error rates are numeric and these will receive closer scrutiny in the future.

An interesting question with the cross-validation process is whether ten zero error classification runs guarantee that a relationship holds true for the entire dataset. To date, this has been proven false on only one occasion with the *soybean-small* dataset. The attribute *external-decay* gave zero error on cross-validation but when the whole dataset was used to find the relationship, one instance was classified incorrectly. This can happen because different models are built for different train/test splits of the data. Models can be built that contain errors on training data but not on test data. When the entire dataset is used for training and testing the error will surface, so the second stage of the discovery process returns models of attribute relationships and verifies that those relationships hold throughout the dataset.

5 Conclusion

An algorithm for the discovery of inter-attribute relationships has been presented and tested on a number of datasets from the UCI repository. It is perhaps surprising, given the exhaustive testing that takes place on these datasets, that so many relationships were discovered. However, it is almost always the case that the specified class attribute is used for testing the performance of a new algorithm rather than discovering any other relationships in the data.

Knowledge of attribute dependencies can be useful when filtering attributes prior to learning and all learning schemes benefit from a cleaner search space, but the most gain from this information is in the knowledge that a relationship exists and can be presented, as possibly new knowledge, to a data provider.

In the practical application of machine learning it has been our experience that a data provider does not necessarily know which attribute to use as the class attribute. Typically, they want to know about all relationships that exist in the data, and more importantly, they want a description of those relationships.

The algorithm presented in this paper shows some promise in discovering inter-attribute relationships. It uses state-of-the-art algorithms which are capable of describing the knowledge they induce which is an important aspect of the task. The algorithm is expensive computing ten runs of a learning algorithm per attribute. The current implementation is used in batch mode with results analysed by hand.

By analyzing the manner in which promising results are pursued, and more generally coming to an understanding of how to interpret numeric attributes, it should be possible to extend the algorithm to determine inter-attribute relationships automatically.

Acknowledgments

The Waikato Machine Learning group, supported by the New Zealand Foundation for Research, Science, and Technology, has provided a stimulating environment for this research.

References

1. Garner S.R., Cunningham S.J., Holmes G., Nevill-Manning C.G. and Witten I.H. Applying a Machine Learning Workbench: Experience with Agricultural Databases. *Proceedings of Machine Learning in Practice Workshop*, Machine Learning Conference, Tahoe City, CA, USA, pp. 14-21 (1995).
2. John G., Kohavi R., and Pfleger K. Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Machine Learning Conference* (pp 121-129). New Brunswick, NJ: Morgan Kaufmann (1994).
3. Quinlan, R. Learning with continuous classes. *Proceedings Australian Joint Conference on Artificial Intelligence*. World Scientific, Singapore, pp 343-348 (1992).
4. Wang Y. and Witten I.H. Induction of model trees for predicting continuous classes, *Proceedings of the European Conference on Machine Learning* Prague, Czechoslovakia, April (1997).
5. Dougherty J., Kohavi R., and Sahami M. Supervised and Unsupervised Discretisation of Continuous Features. *Proceedings of the Twelfth International Conference on Machine Learning,* Tahoe City, California, pp 194-202 (1995).
6. Merz C.J. and Murphy P.M. *UCI Repository of machine learning databases* [http://www.ics.uci.edu/mlearn/MLRepository.html] Irvine, CA: University of California, Department of Information and Computer Science (1996).
7. Quinlan R. *C4.5: Programs for Machine Learning* San Mateo: Morgan Kaufmann (1993).
8. Quinlan R. Unknown Attribute Values in Induction. In A. Segre (Ed.) *Proceedings of the 6th International Workshop on Machine Learning* (pp 164-168) Morgan Kaufmann (1989).