

What do Constructive Learners Really Learn?

Chris Thornton

Cognitive and Computing Sciences
University of Sussex
Brighton
BN1 9QH
UK

Email: Christopher.Thornton@firenet.uk.com

Tel: (44)1273 678856

May 21, 2003

Abstract

In *constructive induction* (CI), the learner's problem representation is modified as a normal part of the learning process. This may be necessary if the initial representation is inadequate or inappropriate. However, the distinction between constructive and non-constructive methods appears to be highly ambiguous. Several conventional definitions of the process of constructive induction appear to include all conceivable learning processes. In this paper I argue that the process of constructive learning should be identified with that of relational learning (i.e., I suggest that what constructive learners really learn is *relationships*) and I describe some of the possible benefits that might be obtained as a result of adopting this definition.

1 Introduction

Constructive induction (CI) is of use when the initial representation for a problem obstructs the application of ordinary inductive methods [1]. Wnek and Michalski [2] have divided constructive induction methods into several types including hypothesis-driven (HCI) methods, data-driven (DCI) methods and knowledge-driven (KCI) methods. Practical methods introduced in recent years include FRINGE [3], AQ17-HCI [Wnek and Michalski, AQ17, 1994] and CN2-MCI [Kramer, 1994].

Almost all CI methods seek to transform the initial representation space by introducing new *features*. However, in the literature, the term 'feature' has been used ambiguously. In most cases it has been used to denote any construct or mechanism which imposes a new partition(ing) on the representation space.

However, this usage cannot be taken too literally since all supervised learning algorithms attempt to implement the *target* partitioning on the space and would thus all be potentially classified as constructive.

Explicitly discounting such degenerate cases still leaves us with methods such as C4.5 [4], Backpropagation [5] and CN2 [6, 7] which all make use of intermediate constructs (i.e., constructs not directly involved in production of output) that identify partitions on the representation space.¹ These methods would seem to satisfy the criterion of being ‘feature generators’ in a non-degenerate sense and yet they are typically described as ‘selective’ and thus *non*-constructive (see [8,9,10,11,12] for a selection of views).

The decision criterion that is being applied in such cases is not completely clear. But it appears to be grounded in an evaluation of the *simplicity* of the partitioning introduced by the feature. Constructs that introduce simple, local partitions (e.g., the axis-parallel partitions created by internal, decision-tree nodes) tend *not* to be considered features and algorithms such as C4.5 are thus effectively eliminated from the constructive class. But the notion of ‘simple partitioning’ is poorly defined and as a result, the whole enterprise of constructive induction is placed on an insecure footing.

In this paper, I will introduce an analysis of inductive justification and show how it permits the distinction to be recast in a theoretical context. I will also show how the process of CI can be properly motivated as a necessary response to hard, relational learning problems.

2 Bayesian analysis of inductive justification

In recent years, researchers have made rapid progress in the theoretical analysis of learning. Early work by Gold [13] and Valiant [14,15] established a tradition which grew to encompass theoretical constructs such as VC-dimension [16] and led to the theoretical advances of Haussler and others, e.g., [17, 18, 19, 20, 21, 21, 22]. Much of this work is directed towards the goal of analyzing the complexity of learning but, at the time of writing, measuring the hardness of arbitrary learning *problems* (e.g., specific training sets) remains problematic [23]. However, it turns out that a useful, qualitative measure of problem hardness can be obtained via a Bayesian analysis of justification sources for generalisation.

Consider the following example. D is the body of data shown in Table 1. Each datum in D (i.e., each row in the table) is made up of the values of variables $x_1, x_2, x_3 \dots x_n$. One of the values of x_4 is missing (see the ‘?’ in the fourth column). Can we use observations on the other data to predict this missing value? In other words, can we empirically *induce* the missing value?

If we find that every possible value of the relevant variable has an equal observed probability then we clearly cannot make any prediction at all. If all values do *not* have the same probability then we should predict the missing

¹In the case of C4.5 the intermediate constructs are the internal decision-tree nodes while in the case of backpropagation the constructs are the hidden units in the network.

x1	x2	x3	x4	x5	x6
2	7	3	5	0	1
0	7	2	6	3	4
0	8	1	6	3	0
1	7	4	6	3	0
1	8	4	5	0	4
2	8	2	?	0	4
0	8	3	5	0	4
1	7	2	6	4	0
1	8	2	6	4	4
2	7	3	5	0	4
2	7	1	5	0	4

Figure 1: Sample body of data

value to be the one which has the highest observed probability. However, there are several ways in which we can work out observed probabilities.

We can look at the unconditional probability of seeing a particular value v of x_i .

$$P(x_i = v)$$

Unfortunately, this does not help since both possible values of x_4 turn out to have the same observed probability. This is simply the chance value

$$P(x_i = v) = \frac{1}{|V|}$$

where V is the set of all possible values of x_i . (In this case the chance value is 0.5 since there are only two possible values.)

We can also look at the probability of seeing a particular value conditional on explicit instantiations of the other values, i.e.,

$$P(x_i = v_a | x_j = v_b \dots)$$

where v_a and v_b are possible values and ‘...’ denotes the optional inclusion of other instantiations. This is more rewarding since it turns out that

$$P(x_4 = 5 | x_5 = 0) = 1$$

which is the observation that we always see $x_4 = 5$ whenever we see $x_5 = 0$.

Finally, we can look at the probability of seeing a particular value conditional on there being an *implicit* property among the instantiations of other variables:

$$P(x_i = v | g(X) = v_g)$$

Here X is the entire datum and v_g is the value of a function g , which evaluates the implicit property. Looking at this sort of probability might have been rewarding if, for example, the missing value had been a value of x_2 , since it turns out that

$$P(x_2 = 7 | \text{duplicatesin}(X) = 0) = 1$$

where the *duplicatesin* function tests whether there are duplicated values in the datum and the 0 value indicates a false result. (This probability is observed because 7 appears as the value of x_2 whenever there are *no* duplicates among the remaining values.)

This third form of probability represents a distinct category if and only if the implicit property cannot be reduced to some set of explicit properties (i.e., the sort of properties which are referenced in probabilities of the first and second form). Thus, we know that values of the function cannot depend on combinations of *absolute* values, since in this case the third-form probability could be rewritten in terms of some set of probabilities of the first and second forms. Values of the function must, therefore, depend on *relative* properties, i.e., relationships among observed values. To put it bluntly, the function g must be a *relational* function.

What the analysis is really telling us is, then, is that there are two quite different ways of predicting missing information from given data. Predictions may be based on observations of absolute values *or* upon relationships observed among them. This conclusion can readily be confirmed by considering even very simple examples. For instance, imagine we are told that

- bricks with apples are yellow
- bricks with trees are blue
- apples with cups are yellow
- cups with trees are blue

If we are then asked whether apples with bricks are blue or yellow, we may say ‘blue’ on the grounds that this seems to be the colour associated with trees, or alternatively ‘yellow’ on the grounds that this is the colour associated with combinations of edible and inedible objects. In the one case we have based the prediction on observations about absolute values. In the other we have based the prediction on observations about a particular relationship. Either prediction may be right or wrong. But *any* prediction we make necessarily ‘sources’ one or other (or some combination of) of the two types of effect. This is a simple and direct consequence of the fact that any effect must involve either absolute values or relationships, or some combination of the two.

3 Empirical v. relational learning

The Bayesian analysis allows us to divide induction methods (i.e., prediction methods) into two basic types: methods which attempt to exploit *absolutes* and

methods which attempt to exploit *relationships*. These groupings correspond roughly to the well established categories of **empirical learning** and **relational learning** [24, 25]. The distinction is key from the point of view of complexity. Methods in the first (empirical) category confront a much easier learning task than methods in the latter (relational) category. A method that attempts to exploit observations of absolutes need only consider cases that are *explicitly* observed in the data. There are a finite number of these. The task thus involves consideration of a *finite* number of objects. A method that attempts to exploit probabilities based on relationships, on the other hand, must grapple with the fact that there are, in general, an *infinite* number of relationships that might be relevant. The general conclusion is that exploiting probabilities based on absolutes is a task of finite complexity while exploiting probabilities based on relationships is a task of infinite complexity.²

For present purposes, the interesting thing about the empirical/relational distinction is the light it sheds on the issue of constructive learning. And there is, in fact, a clear suggestion that we may be able to *identify* the class of constructive learners with the class of relational learners. Practical CI methods, it turns out, are often based on some form of relational search process. Oliveira and Sangiovanni-Vincentelli [12] describe a system that searches for a minimal set of features each of which tests for a particular relationship among the input variables. Kramer [10] describes a system that tries to detect and exploit relations over variables which tend to appear together in useful rules. Matheus [27] describes the CITRE system which, to a first approximation, tries to capitalize on the presence of disjunctive regions in decision tree descriptions. Other systems involve a search (i.e., operator-based) process working with some sort of relational description language, e.g., [28, 11,9]. In several recent cases, this type of approach has focussed on what are known as ‘counting’ or *M-of-N* features, i.e., features which effectively count the number of occurrences of a particular variable value, cf. [29,30,2,8 31,32].

Equating constructive induction with relational learning is also compatible with the intuition (noted above) that constructive induction involves the creation of ‘non-local’ partitions. Any feature/function whose values depend on absolutes will tend to define a partitioning involving contiguous regions of the original input space, whereas a feature which computes a relational property will not do so. The values of the former type of function are, by definition, correlated with the absolute values of the inputs to that function. Thus values of the function are correlated with absolute ‘coordinates’ of the representation space and the function must, therefore, define sets of objects which tend to share the same coordinates, i.e., be located within the same region of that space. In contrast, the values of a relational function are not correlated with the absolute values of the function’s inputs and thus not correlated with absolute coordinates of the representation space. The feature implemented by the function will therefore tend to instantiate a complex, non-local partition of the space.

²It is no surprise to find that practical learning methods tend to be predominantly empirical rather than relational [26].

But some difficulties remain. It is arguable that the identification of constructive learning with relational learning flies in the face of convention. One of the best known definitions of constructive induction — given in [33] — is ‘constructive induction is any form of induction that generates new descriptors not present in the input data.’ And more recently, Tom Mitchell has defined constructive induction as ‘the process of augmenting the set of predicates, based on background knowledge.’ [34]. These definitions refer merely to the production of new predicates rather than to the identification of relational functions (predicates). They therefore provide a definition which is weaker and more general than the one proposed. But as was noted in the introduction, the conventional definition appears to be operationally ambiguous and thus cannot be regarded as entirely satisfactory.

A further advantage of the identification of constructive learning with relational learning is the way it opens up possibilities for defining special cases of the construction process. There are a number of ways in which we can define subcategorisations of relational learning. Under the proposed identification, all such classifications become available for the purposes of drawing distinctions between different varieties of constructive learning.

One of the more interesting directions we might take this involves the utilisation of feedback within the learning process. In the simplest case, a relational learner engages in a one-pass approach. It simply attempts to identify some set of relationships which are relevant to the given input data. But in fact relational learners are always potentially *recursive*. The identification of any set of relational effects involves the application of evaluations (functions) to the original data. This creates new values and thus, potentially at least, new data. These new data can themselves be processed for exploitable effects in a recursive manner. The process of recursively exploiting relational effects is manifestly a *constructive* process. But in the context of the proposed identification of constructive learning with non-recursive relational learning, we would have to classify recursive relational learning as a ‘higher-order’ constructive process.

4 Concluding comments

It is widely agreed that existing learning methods are extremely effective provided an appropriate representation of the problem is used. Thus constructive induction — the problem of *finding* the right representation — is a key challenge. In this paper I have shown that the conventional definition of the term constructive learning is ambiguous and over inclusive. As an alternative, I have proposed that the process of constructive learning be defined in terms of relational learning. This approach specialises the original definition considerably and thus reduces the size of its extension. But the elements which remain appear to be exactly the learning processes which are conventionally identified as constructive, i.e., those which engage in the identification of complex, ‘non-local’ partitions within the input space. Combining the categories of constructive and relational learning may also produce interesting cross-fertilisation effects in the

form of relational concepts applied to constructive scenarios, and vice versa.

References

- [1] Michalski, R. (1983). A theory and methodology of inductive learning. In R. Michalski, J. Carbonell and T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach*. Palo Alto: Tioga.
- [2] Wnek, J. and Michalski, R. (1994). Hypothesis-driven constructive induction in AQ17-HCI: a method and experiments. *Machine Learning*, 14 (p. 139). Boston: Kluwer Academic Publishers.
- [3] Pagallo, G. (1989). Learning DNF by decision trees. *Proceedings of The Eleventh Joint Conference on Artificial Intelligence* (pp. 639-644). Morgan Kaufmann.
- [4] Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann.
- [5] Rumelhart, D., Hinton, G. and Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, 323 (pp. 533-6).
- [6] Clark, P. and Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, 3 (pp. 261-283).
- [7] Clark, P. and Boswell, R. (1991). Rule induction with CN2: some recent improvements. In Y. Kodratoff (Ed.), *Proceedings of the Fifth European Working Session on Learning*. No. 482 of Lecture Notes in Artificial Intelligence (pp. 151-163). Springer-Verlag.
- [8] Sazonov, V. and Wnek, J. (1994). A hypothesis-driven constructive induction approach to expanding neural networks. *Proceedings of ML-COLT'94*.
- [9] Pfahringer, B. (1994). Cipl 2.0: a robust constructive induction system. *Proceedings of ML-COLT'94*.
- [10] Kramer, S. (1994). CN2-MCI: a two-step method for constructive induction. *Proceedings of ML-COLT'94*.
- [11] Japkowicz, N. and Hirsh, H. (1994). Towards a bootstrapping approach to constructive induction. *Proceedings of ML-COLT'94*.
- [12] Oliveira, A. and Sangiovanni-Vincentelli, A. (1992). Constructive induction using a non-greedy strategy for feature selection. In D. Sleeman and P. Edwards (Eds.), *Proceedings of the Ninth International Workshop on Machine Machine Learning (ML92)* (pp. 355-360). San Mateo, California: Morgan Kaufmann Publishers.
- [13] Gold, E. (1967). Language identification in the limit. *Information and Control*, 10 (pp. 447-74).

- [14] Valiant, L. (1984). A theory of the learnable. *Communications of the ACM*, 27 (pp. 1134-42).
- [15] Valiant, L. (1985). Learning disjunctions of conjunctions. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 560-566). Los Altos: Morgan Kaufmann.
- [16] Vapnik, V. and Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theor. Probab. Appl.*, 16, No. 2 (pp. 264-280).
- [17] Haussler, D. (1986). Quantifying the inductive bias in concept learning. UCSC-CRL-86-25, University of California at Santa Cruz.
- [18] Blumer, A., Ehrenfeucht, A., Haussler, D. and Warmuth, M. (1987). Occam's razor. *Information Processing Letters*, 24 (pp. 377-380).
- [19] Haussler, D. (1988). Quantifying inductive bias: AI learning and valiant's learning framework. *Artificial Intelligence*, 36 (pp. 177-221).
- [20] Haussler, D. (1987). Bias, version spaces and valiant's learning framework. *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 324-336). University of California, Irvine: (June 22-25).
- [21] Baum, E. and Haussler, D. (1990). What size net gives valid generalization?. In J.W. Shavlik and T.G. Dietterich (Eds.), *Readings In Machine Learning* (pp. 258-262). San Mateo, California: Morgan Kaufmann.
- [22] Haussler, D., Kearns, M. and Schapire, R. (1992). Bounds on the sample complexity of bayesian learning using information theory and the VC dimension. UCSC-CRL-91-44, University of California at Santa Cruz.
- [23] Kearns, M. (1990). *The Computational Complexity of Machine Learning*. The MIT Press.
- [24] Michalski, R., Carbonell, J. and Mitchell, T. (Eds.) (1983). *Machine Learning: An Artificial Intelligence Approach*. Palo Alto: Tioga.
- [25] Michalski, R., Carbonell, J. and Mitchell, T. (Eds.) (1986). *Machine Learning: An Artificial Intelligence Approach: Vol II*. Los Altos: Morgan Kaufmann.
- [26] Stone, J. and Thornton, C. (1995). Can artificial neural networks discover useful regularities?. *Proceedings of ICANN-95*. Cambridge.
- [27] Matheus, C. (1990). Adding domain knowledge to SBL through feature construction. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 803-808). Boston, MA.: MIT Press.

- [28] Aronis, J. and Provost, F. (1994). Efficiently constructing relational features from background knowledge for inductive machine learning. *Proceedings of ML-COLT'94*.
- [29] Spackman, K. (1988). Learning categorical decision criteria in biomedical domains. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 36-46). San Mateo, CA: Morgan Kaufmann.
- [30] Fawcett, T. and Utgoff, P. (1991). A hybrid method for feature generation. *Proceedings of the Eighth International Workshop on Machine Learning* (pp. 137-141). Evanston, Ill.
- [31] Seshu, R. (1989). *Solving the Parity Problem*. University of Illinois at Urbana-Champaign, Inductive Learning Group.
- [32] Murphy, P. and Pazzani, M. (1991). ID2-of-3: constructive induction of m-of-n concepts for discriminators in decision trees. *Proceedings of the Eighth International Workshop on Machine Learning (ML91)*. San Mateo, CA: Morgan Kaufmann.
- [33] Dietterich, T. and Michalski, R. (1983). A comparative review of selected methods for learning from examples. In R. Michalski, J. Carbonell and T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach*. Palo Alto: Tioga.
- [34] Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.