Unsupervised Constructive Learning

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

Email: Chris.Thornton@cogs.susx.ac.uk
WWW: http://www.cogs.susx.ac.uk
Tel: (44)1273 678856

November 27, 1996

Abstract

In *constructive induction* (CI), the learner's problem representation is modified as a normal part of the learning process. This is useful when the initial representation is inadequate or inappropriate. In this paper, I argue that the distinction between constructive and non-constructive methods is unclear. I propose a theoretical model which allows (a) a clean distinction to be made and (b) the process of CI to be properly motivated. I also show that although constructive induction has been used almost exclusively in the context of supervised learning, there is no reason why it cannot form a part of an *unsupervised* regime.

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 [2] and CN2-MCI [4].

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* 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?

x_1	x_2	x_3	x_4	x_5	x_6
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

Table 1: Sample body of data.

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 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.,

$$P(x_i = v_a | x_j = v_b ...)$$

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

$$P(x_2 = 7|duplicatesin(X) = 0) = 1$$

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.)

Methods which attempt to discover and exploit such probabilities for inductive purposes, without using any other source of information, are **empirical learning algorithms**. There are a large number of these, see [24, 25, 26].

3 Statistical v. relational learning

The analysis of justification sources allows us to divide methods of inductive learning into two basic types. A method that attempts to exploit either of the first two forms of probability confronts a tractable task. Only cases that are *explicitly* observed in the data need to be taken into account. There are a finite i.e., they tend to exploit probabilities of the first and second form, rather than of the third form. [27]

Interestingly, we can deduce that the evaluation function used in the third form must measure a *relational* property of its inputs. To understand why, we need to think about the way in which the function differentiates different types of input. Let us say that the function produces a particular value whenever the input variables have certain *absolute* values. In this case, the evaluation is effectively a label for an explicit case. If all the values of the function are derived this way, the conditional probability can be reduced to a set of probabilities of the second form. If the probability is a valid example of the third form, the evaluation function must therefore measure a non-absolute — i.e., *relational* — property of its inputs.

Learning problems whose solutions involve exploiting probabilities of the third form are thus **relational**. Problems which involve exploitation of probabilities for explicit cases are **statistical**, since they simply involve the derivation of frequency statistics over a finite dataset. Learning *methods* can be classified the same way. Learning procedures which exploit probabilities of the first and second form are statistical while ones which exploit probabilities of the third form are relational.

4 Recursive relational exploitation is constructive learning

It is important to note that relational learners are 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 new data. These new data can themselves be processed for statistical and relational effects in a recursive manner. Thus the process of recursively exploiting relational effects is manifestly a *constructive* process. Since the process of exploiting statistical effects is manifestly *non-constructive* — and since there are no alternative methods of exploitation — it can be deduced that the process of constructive induction is, precisely, the process of recursive exploitation of relational effects. This provides an operational definition of the the distinction between constructive and non-constructive methods. It also gives the former type of process a clear motivation: constructive methods are necessitated just in case the relevant problem is of the hard, relational type.

Equating 'constructive induction' with recursive, relational learning is generally compatible with conventional interpretations of the meaning of the term. In particular, it is compatible with the intuition that constructive induction involves the creation of 'non-local' partitions. Any feature/function which computes a statistical property of its arguments will tend to define a partitioning involving contiguous regions of the original input space, whereas a feature which

process of *unsupervised* constructive learning may have any useful application.

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] Kramer, S. (1994). CN2-MCI: a two-step method for constructive induction. *Proceedings of ML-COLT'94*.

[5] Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann.

[6] Rumelhart, D., Hinton, G. and Williams, R. (1986). Learning representations by back-propagating errors. *Nature, 323* (pp. 533-6).

[7] Clark, P. and Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning, 3* (pp. 261-283).

[8] 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.

[9] Sazonov, V. and Wnek, J. (1994). A hypothesis-driven constructive induction approach to expanding neural networks. *Proceedings of ML-COLT'94*.

[10] Pfahringer, B. (1994). Cipf 2.0: a robust constructive induction system. *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] Shavlik, J. and Dietterich, T. (Eds.) (1990). *Readings in Machine Learning.* San Mateo, California: Morgan Kaufmann.

[25] Michalski, R., Carbonell, J. and Mitchell, T. (Eds.) (1983). *Machine Learning: An Artificial Intelligence Approach.* Palo Alto: Tioga.

[26] Michalski, R., Carbonell, J. and Mitchell, T. (Eds.) (1986). *Machine Learning: An Artificial Intelligence Approach: Vol II.* Los Altos: Morgan Kaufmann.

[27] Stone, J. and Thornton, C. (1995). Can artificial neural networks discover useful regularities?. *Proceedings of ICANN-95*. Cambridge.

[28] 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.

[29] Aronis, J. and Provost, F. (1994). Efficiently constructing relational features¡br¿from background knowledge for inductive machine learning. *Proceedings of ML-COLT'94*.

[30] 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,.

[31] 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.

[32] Seshu, R. (1989). *Solving the Parity Problem*. University of Illinois at Urbana-Champaign, Inductive Learning Group.

[33] 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.