Is Transfer Inductive?

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

Work is currently underway to devise learning methods which are better able to transfer knowledge from one task to another. The process of knowledge transfer is usually viewed as logically separate from the inductive procedures of ordinary learning. However, this paper argues that this 'seperatist' view leads to a number of conceptual difficulties. It offers a task analysis which situates the transfer process *inside* a generalised inductive protocol. It argues that transfer should be viewed as a subprocess within induction and not as an independent procedure for transporting knowledge between learning trials.

1 Introduction

Where learning tasks are closely related, it seems reasonable to expect a learner to be able to improve its performance on a particular learning task by reapplying knowledge gained on some previous learning task. The learner should, we feel, be able to *transfer* knowledge from one task to another. Unfortunately, popular learning methods such as backpropagation [1] often exhibit erratic transfer effects [2]. Sometimes positive transfer effects are obtained[1] but sometimes they

[1] In fact Harvey and Stone [3] argue that there is *always* a positive, initial transfer effect with backpropagation learning

are exactly reverse of what we want: the acquisition of new knowledge appears to catastrophically interfere with existing knowledge [4].

Many workers are engaged in the attempt to realise the benefits of knowledge tranfer within learning [cf. 5, 6, 7, 8].[2] However, there seems to be some residual fuzziness in our thinking about the relationship between transfer and learning. In particular, different assumptions are made about the way in which these two processes interact.

In some cases the role of learning is simply rote storage (i.e., memorisation) of presented data. However, in most cases learning involves going beyond presented data, i.e., it involves some form of induction. Where the goal of learning is some form of behaviour then producing high performance means doing the right thing at the right time. But we can, of course, always see this as a kind of induction simply by treating the motor commands to be learned as the 'target outputs' in a conventional induction problem.

If we accept the idea that learning can usually be viewed as some sort of inductive process, we have to ask how transfer fits in. A common view is that transfer is an operation which takes place between learning tasks. This suggests that the process is somehow independent and separated from normal inductive activity. On the other hand, transfer seems pointless unless it contributes in some way to learning (i.e., inductive) performance. This seems to imply that we should view transfer as being a part of an higher-level inductive process.

There are thus conceptual problems to deal with whether we treat transfer as separate from induction or as closely integrated with it. To try to resolve these I present a task analysis of induction [9]. This differs from some theoretical treatments of learning (e.g., COLT treatments such as [10]) since it concentrates exclusively on properties of the induction problem and ignores possible solutions altogether. Interestingly, it leads to a view of induction which gives a clear role to a transfer process[3] and also allows us to formulate a criterion for deciding when and if such transfer has occurred. The paper thus provides theoretical ammunition for those who take the view that transfer should be treated as an aspect of induction rather than a separate activity (i.e., the 'anti-separatist' view).

The paper divides up into four main sections. The next section (section two) provides the task analysis of induction. The third section shows how the task analysis supports a particular conceptualisation of the learning process. The fourth section shows how this conceptualisation leads to a new appraisal of the role transfer plays in induction.

[2] Some recent work was presented at the NIPS-95 workshop on 'learning to learn and transfer'. Lori Pratt has a WWW page giving a useful set of pointers, see `http://vita.mines.colorado.edu:3857/lpratt/transfer.html`

[3] Not necessarily *the* transfer process.

2 A task analysis of induction

Imagine we have a body of data D, as shown in Table 1. Each datum in D (i.e., each row) is made up of the values of variables x_1, x_2, x_3, x_4 and x_5. One of the values of x_3 is missing (see the '?' in the x_3 column). Can we use the other data to predict this missing value? In other words, can we empirically *induce* the missing value from the data which are provided?

x_1	x_2	x_3	x_4	x_5
c	d	f	a	b
a	b	h	d	b
e	c	h	d	e
c	b	f	a	e
a	c	f	d	e
c	c	?	a	e
b	c	f	a	e
b	d	h	d	e
e	d	f	a	c
a	c	h	d	c
c	d	h	a	c

Table 1: Sample induction problem.

If we observe that every possible value of the relevant variable has the same probability then we clearly cannot make any prediction at all. If all values do *not* have the same probability then we will rationally 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'. First, we can look at the unconditional probability of seeing a particular value v of x_i.

$$P(x_i = v)$$

In the present case this is not productive since both possible values of x_3 have the same unconditional probability. This is just the chance value of 0.5, i.e.,

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

where V is the set of all possible values of x_i.

Second, we can 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

$$P(x_3 = f | x_4 = a) = 0.8$$

In other words, we see x_3 =f in 4 of the 5 cases where we see x_4 =a.[4]

Third and 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 an imaginary function g, which evaluates the implicit property. This is more rewarding still since it turns out that

$$P(x_3 = h | duplicates(X)) = 1$$

where the duplicates function is a predicate which tests whether there are duplicated values in the datum. In other words, it turns out that we always see x_3 =h when there are duplicates among the other values.

These three formulae represent the *only* ways in which a particular guess might be empirically justified.[5] In fact, there are really only two formulae to consider since we can always regard an unconditional probability as a conditional probability with an empty condition. Thus the task analysis shows that there are really just two sources inductive justification: one based on *explicitly* observed probabilities and the other based on *implicitly* observed probabilities.

If we want to make an inductive guess regarding the missing value of x_3, we therefore must exploit some combination of these two sources. In the present example we will probably guess that x_3 =h since the highest probability we have unearthed (so far) is based on the observation that this value occurs in every case where there are duplicates among the remaining values.[6]

Methods which attempt to discover and exploit such probabilities for inductive purposes — without using any other source of information — are **empirical learning** algorithms. The development of these methods is the concern of several research communities including Machine Learning and Connectionism (see [11, 12, 13]).

[4] Of course this is not the only significant conditional probability.

[5] If this seems counter-intuitive note that the third formula acts as a kind of catch-all since it covers *any* computational, mathematical or functional justification for an inductive guess.

[6] In doing so we make the unrealistic but — in the absence of background knowledge — inevitable assumption that all probability values are independent.

3 Statistical v. relational learning

The fact that inductive guesses depend on just two sources of justification implies that any inductive method must exploit either implicit or explicit justification, or some combination of the two. This allows one to make a basic complexity distinction between inductive methods. A method that attempts to exploit explicit probabilities confronts a relatively easy task. Only cases that are explicitly observed in the data need to be taken into account. There are a finite number of these. The task thus involves deriving frequency statistics (probabilities) over a *finite* dataset.

A method that attempts to exploit implicit probabilities, on the other hand, confronts a harder task since it has to first identify the appropriate evaluation function for the implicit property (i.e., it has to guess what the property is). If functions with an infinite range are to be considered, then the task is *infinitely* hard, since there are clearly an infinite number of such functions. Even if we restrict attention to functions with a finite range, the task is still *hard* since the number of functions to be considered is exponentially related to the number of observed cases. Consider the simplest case. We have n variables each of which takes m values and we consider only functions with a binary range and minimum arity n. The number of possible functions is then 2^{m^n}.

The general implication is that explicit justification is more easily exploited than implicit justification. It is no surprise, then, to find that practical learning methods tend to be predisposed towards the former approach, i.e., they tend to exploit probabilities of the explicit form rather than of the implicit form [14].

In this analysis no assumptions are made about the imaginary function g or about how it behaves. However, we can say that it must be doing something *more* than simply testing for explicit patterns of absolute variable values. In this case the function would be playing a redundant role; the relevant justification would not be based on a formula of the third form; it would really be based on a set of formulae of the second form. But if values of g cannot depend on absolutes, they must depend on non-absolutes, i.e., *relational* effects. Thus g is necessarily a relational function: it tests or measures a relationship among its inputs.

This is a satisfying connection to make since it allows us to say that 'hard' learning problems — i.e., those which involve exploitation of implicit justification — are **relational**. This reaffirms the long-standing Machine Learning heuristic that 'learning relationships is hard' (cf. [15]). Problems which merely involve exploitation of explicit probabilities can be viewed as 'statistical', since they can be solved by deriving frequency statistics over a finite dataset.

Applying this terminology to learning methods allows us to speak about 'statistical methods' (i.e., methods which depend exclusively on explicit justification sources) 'relational methods' (methods which depend on implicit justification sources) and 'hybrid methods' (methods which depend on some combination of the two types of justification). This taxonomy is, of course, purely analytic. In

practice it may be hard to allocate a particular method to a particular category.

A small number of cases *can* be conclusively classified within the scheme. The ID3 method [16], now more often used in its updated manifestation as C4.5 [17] is a case in point. ID3 takes a training set of sample input/output pairs from an input/output mapping, and constructs a decision tree (for generating outputs) by recursively partitioning the training set until every pair in a given partition has the same output value.

At each stage of the process, a new partitioning is constructed by dividing up the cases in an existing partition according to which value they have on the variable whose values are most strongly associated (within the partition) with specific output values. This has the effect of maximising the output-value uniformity of new partitions and thus minimising (subject to horizon effects) the total number of partitions required in order to achieve full uniformity. The algorithm is thus guided only by statistical effects in the training data. It is thus an *exclusively* statistical method.

Aside from ID3, learning methods which can be classified as exclusively statistical include the CART algorithms [18], the **competitive learning** regime of Rumelhart and Zipser [19], the **Kohonen net** [20] and in fact any algorithmic method which is based on the method of **clustering** [21]. There are also examples of exclusively relational learning methods. Examples include the 'BACON' methods of Langley and co-workers [22; 23; 24; 25] and related methods such as [26; 27; 28; 29]. All these systems carry out explicit searches for relational effects and in most cases ignore statistical effects altogether.

4 Transfer Revisited

One of the interesting properties of the task analysis is that it allows us to assign a clear role to knowledge transfer and to say exactly how it contributes within the overall process of induction. As we will see, it also allows us to explain why transfer has typically been treated as a separate and independent process.

The task analysis divides the inductive process into two parts: a statistical part and a relational part. We see immediately that the statistical part — the exploitation of statistical effects — appears to offer no role whatsoever to any sort of transfer process. Statistical effects exist, by definition, in the *data*; prior experience is thus essentially irrelevant. With a finite dataset there are always a finite number of statistical effects and the process of identifying them is tractable. If we *must* find a role for transfer within the statistical aspect of induction we might argue that since, in practice, statistical learners focus on a subset of the space of effects, they operate with a (statistical) bias and that this bias might be adapted and improved with experience. However, this seems contrived and tendentious. It leads, moreover, to a view of transfer which deviates markedly from the view that most researchers apply to the process.

Fortunately, the relational part of the induction process offers a rather clear

and obvious role to knowledge transfer. Arguably, it *requires* that transfer play a role. Recall that exploitation of relational effects involves the identification of relationships in the data. Since in general the space of possible relationships is infinite this identification necessarily involves a bias. A learner seeking to exploit relationships in the data must always have some particular relationships 'in mind.' Thus the learner uses assumptions regarding the relevance or salience of relationships. These assumptions constitute 'knowledge' which, if it is justified at all, must be justified in terms of prior, relevant experience. Relational learning, then, is either unjustified or based on knowledge transfer.

The implication is worth spelling out. A well justified, relational learning process applied to a sequence tasks *necessarily* engages in knowledge transfer. The process can be viewed in terms of the acquisition of a suitable bias. This bias is constituted in the set of salient relationships which are used as candidates in the effect-identification process. Thus the task analysis suggests how and why a transfer process will operate within induction.

The analysis also gives us a way of detecting the occurrence of transfer. Recall that within relational learning, relationships in the data are exploited through the application of suitable relational functions (i.e., suitable gs.) This necessarily introduces new statistical effects — it effectively reduces relational effects to statistical ones. Thus it is possible to decide whether transfer has taken place by seeing whether the actions of the learner have introduced new statistical effects.[7]

5 Comments

The task analysis presented by this paper offers a way of visualising the knowledge transfer process and of understanding its role within induction. Of course, the transfer process envisaged herein may not be the 'right' or 'only' one. It may well be far removed from the transfer process that forms the focus of investigations for other researchers. However, it is clear that viewing transfer as a knowledge-accumulation operation within relational learning does have tangible benefits. It allows us to form a more coherent view of the way in which transfer and induction interact. We see in particular that there is an important part of induction which has *nothing to do* with transfer. And we also see that transfer is an optional extra in a relational learning process — only of use in the case where the learner confronts a sequence of tasks.

The view that transfer and induction are *separate* operations now becomes explicable as a natural consequence of viewing induction in terms of statistical exploitation processes such as ID3. And the worry over whether, why or how transfer 'contributes' to induction falls away since the contribution is now fully accounted for in terms of the supporting role that transfer plays within relational

[7] This check has to be made, of course, with respect to the original data and to any internal data created by the learner.

learning. The proposed model thus offers some real benefits for the achievement of a better understanding of transfer. Whether it has any worth for those engaged in the application of transfer in practical contexts remains to be seen.

References

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

[2] Murre, J. (Forthcoming). Transfer of learning in backpropagation and in related neural network models. In J. Levy, D. Bairaktaris, J. Bullinaria and P. Cairns (Eds.), *Connectionist Models of Memory and Language*. London: UCI Press.

[3] Harvey, I. and Stone, J. (1995). Unicycling helps your french: spontaneous recovery of associations by learning unrelated tasks. CSRP 379, School of Cognitive and Computing Sciences, University of Sussex.

[4] McCloskey, M. and Cohen, N. (1989). Catastrophic interference in connectionist networks: the sequential learning problem. In G.H. Bower (Ed.), *The Psychology of Learning and Motivation*. New York: Academic Press.

[5] Schmidhuber, J. (1996). A theoretical foundation for multi-agent learning and incremental self-improvement in unrestricted environments. In X. Yao (Ed.), *Evolutionary Computation: Theory and Applications*. Singapore: Scientific Publishing Company.

[6] Caruana, R. (1995). Learning many related tasks at the same time with backpropagation. *Advances in Neural Information Processing Systems 7* (Proceedings of NIPS-94) (pp. 657-664).

[7] Martin, J. and Billman, D. (1994). Acquiring and combining overlapping concepts. *Machine Learning*, *16* (pp. 1-37).

[8] Pratt, L. (1994). Experiments on the transfer of knowledge between neural networks. In S. Hanson, G. Drastal and R. Rivest (Eds.), *Computational Learning Theory and Natural Learning Systems, Constraints and Prospects* (pp. 523-560). MIT Press.

[9] Thornton, C. (1995). Measuring the difficulty of specific learning problems. *Connection Science*, *7*, No. 1 (pp. 81-92).

[10] Kearns, M. (1990). *The Computational Complexity of Machine Learning*. The MIT Press.

[11] Shavlik, J. and Dietterich, T. (Eds.) (1990). *Readings in Machine Learning*. San Mateo, California: Morgan Kaufmann.

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

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

[14] Thornton, C. (1994). Statistical biases in backpropagation learning. *Proceedings of the International Conference on Artificial Neural Networks* (pp. 709-712). Sorrento, Italy.

[15] Dietterich, T., London, B., Clarkson, K. and Dromey, G. (1982). Learning and inductive inference. In P. Cohen and E. Feigenbaum (Eds.), *The Handbook of Artificial Intelligence: Vol III.* Los Altos: Kaufmann.

[16] Quinlan, J. (1986). Induction of decision trees. *Machine Learning, 1* (pp. 81-106).

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

[18] Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees.* Wadsworth.

[19] Rumelhart, D. and Zipser, D. (1986). Feature discovery by competitive learning. In D. Rumelhart, J. McClelland and the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Vol I* (pp. 151-193). Cambridge, Mass.: MIT Press.

[20] Kohonen, T. (1984). *Self-organization and Associative Memory.* Berlin: Springer-Verlag.

[21] Diday, E. and Simon, J. (1980). Clustering analysis. In K. Fu (Ed.), *Digital Pattern Recognition.* Communications and Cybernetics, No. 10 (pp. 47-92). Berlin: Springer-Verlag.

[22] Langley, P. (1977). Rediscovering physics with bacon-3. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence: Vol I.*

[23] Langley, P. (1978). BACON.1: a general discovery system. *Proceedings of the Second National Conference of the Canadian Society for Computational Studies in Intelligence* (pp. 173-180). Toronto.

[24] Langley, P., Bradshaw, G. and Simon, H. (1983). Rediscovering chemistry with the BACON system. In R. Michalski, J. Carbonell and T. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach* (pp. 307-329). Palo Alto: Tioga.

[25] Langley, P., Simon, H., Bradshaw, G. and Zytkow, J. (1987). *Scientific Discovery: Computational Explorations of the Creative Processes.* Cambridge, Mass.: MIT Press.

[26] Wolff, J. (1978). Grammar discovery as data compression. *Proceedings of the AISB/GI conference on Artificial Intelligence* (pp. 375-379). Hamburg.

[27] Wolff, J. (1980). Data compression, generalisation and overgeneralisation in an evolving theory of language development. *Proceedings of the AISB-80 conference on Artificial Intelligence.* Amsterdam.

[28] Lenat, D. (1982). AM: discovery in mathematics as heuristic search. In R. Davis and D.B. Lenat (Eds.), *Knowledge-Based Systems in Artificial Intelligence* (pp. 1-225). New York: McGraw-Hill.

[29] 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.