There is No Free Lunch but the Starter is Cheap: Generalisation from First Principles

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/users/cjt
Tel: (44)1273 678856

Abstract

According to Wolpert's no-free-lunch (NFL) theorems [Wolpert, 1996b, Wolpert, 1996a], generalisation in the absence of domain knowledge is necessarily a zero-sum enterprise. Good generalisation performance in one situation is always offset by bad performance in another. Wolpert notes that the theorems do *not* demonstrate that effective generalisation is a logical impossibility but merely that a learner's bias (or assumption set) is of key importance in determining its generalisation performance. However, in this paper it is argued that this may be an over-reading of the results. Situations can be identified in which a learner's assumptions are effectively *guaranteed* correct. The in-practice prevalence of these situations may account for the reliably good generalisation performance of methods such as C4.5 and Backpropagation.

Keywords: no-free-lunch, generalisation, learning complexity

Introduction

There has been lively controversy over Wolpert's no-free-lunch theorems [Wolpert, 1996b; Wolpert, 1996a; Wolpert, 1995b; Wolpert, 1992; Wolpert, 1995a; Wolpert and Macready, 1995] and Schaffer's closely related **conservation law** [Schaffer, 1994]. These results show that there is no guaranteed correct way of performing generalisation. They thus affirm Hume's claim to the effect that the observation of `the frequent conjunction of objects' does not permit the drawing of any particular inference concerning `any object beyond those of which we have had experience' [Hume, 1740].

The underlying idea behind these results is easily stated. Let's say we have a particular learning method and we would like to know how well it will generalise on the problems from a specific domain. If we have no special knowledge about the domain then all problems in the domain have to be considered uniformly likely, i.e,. the problems in the domain have to be considered to follow a uniform distribution. In this context, the problems in the domain may be organised into `opposites', such that the way the unseen (test) cases are classified in a particular problem is the reverse of the way they are classified in its opposite. A particular learning algorithm generalises cases in a specific way. Thus, if it performs slightly better than random guessing on a particular problem, it must perform slightly worse than random guessing on the problem's opposite. On a random selection of problems from the domain, a learning algorithm will therefore tend to produce above-chance performance on some problems and below-chance performance on other problems. Since the chances of it producing above-chance performance are *identical* to the chances of it producing below-chance performance, it will, on average, produce exactly the same performance as random guessing.

At first sight, the NFL result appears to demonstrate that effective (i.e., above-chance) generalisation is impossible in principle. But this is not the case. In the NFL scenario, we have the rather severe constraint that *nothing* is known about the domain. All problems have then to be considered equally likely and the process of applying a particular learner to some random selection of problems necessarily produces chance-level performance (on average). The explicit consequence of the NFL result is thus that in the situation where no domain assumptions can be made, chance-level performance is the inevitable result. But the subtext of the NFL work is that it is the assumptions a learner makes about its domain which are key.((Wolpert specifically mentions the requirement to prove that `the non-uniformity in [the problem domain] is well-matched to your ... learning algorithm.' [Wolpert, 1996b, p. 19])) As Michael Perrone has commented, `What makes NFL important is that it emphasizes in a very striking way that it is the *assumptions* that we make about our learning domains that *make all the difference*.'((From a posting to the `connectionists' mail list.))

However, interpreting the NFL theorems in this way raises a new concern. As Wolpert has noted, the biases of empirical generalisation methods are typically not made explicit and are only rarely justified in terms of the expected application domain. He notes that `for many algorithms, no one has even tried to write down that set of [problems] for which their algorithm works well.' [Wolpert, 1996b]. However, it is clear that generalisation methods *are* capable of performing well in practice across a wide variety of situations [Thrun *et al.* 1991]. In fact, in a recent article Holte [1993] has shown that even rather trivial methods may perform well on a wide variety of real-world generalisation problems. In practice, then, it seems as if generalisation methods are often able to `get away with' not being mindful of their biases. How can we reconcile this with the assumption that biases `make all the difference?'

One possibility is that these learning methods are unwittingly applying generically appropriate biases, i.e., they are exploiting a particular form of non-uniformity which turns out to be present in most or all application domains. But what might this non-uniformity be? And how could we demonstrate its existence? Wolpert points out that we cannot use any form of `prior knowledge' in order to justify any particular assumption of non-uniformity since such knowledge cannot be guaranteed to be a correct guide to future outcomes. What is required, he says, is `a proof based completely on first principles'. Anything less than this has to be regarded as inductively tainted and thus itself subject to the NFL results.

The present paper aims to respond to this challenge by producing precisely the proof the Wolpert believes is required, i.e., a proof of generic non-uniformity based exclusively on first principles. The proof will use a logical task-analysis of the process of generalisation introduced in [Clark and Thornton, 1997, Thornton and Clark, Forthcoming]. This analysis will be reviewed in section 2. Section 3 will investigate the caveats that have to be applied when the analysis is applied to realistic scenarios. Section 4 of the paper will show how the analysis justifies certain a priori assumption regarding generic non-uniformities. Section 5 is a summary.

The complexity of the learning task from first principles

Learning may involve the acquisition of new behaviour or the acquisition of new knowledge. However, for analytic purposes it is often convenient to combine the two categories into one, treating knowledge acquisition as a type of behaviour learning which produces novel `thought behaviour.' The advantage of this is that it allows us to decompose any learning task along behavioural lines. In particular, it allows us to focus attention on how learning tasks always involve acquiring a disposition to produce certain `actions' in certain `situations.'

Let us say we want a learner agent to acquire behaviour B, which involves producing actions A1 and A2. The salient information (i.e., data) available to the learner may be internally stored or derived via sensors from an external environment (or both). But for learning to be possible, this information must indicate in some way the situations in which A1 and A2 should be produced. Learning the task thus involves (a) identifying the nature of this indication and (b) consolidating it in the form of new behaviour. The difficulty of the latter operation depends entirely on the properties of the agent and should not, therefore, be considered a part of the generic complexity of the learning task. This should be estimated purely in terms of the identification operation.

Identifying the relevant indication involves identifying the connections that may exist between the learner's informational data and the actions in question. The complexity of this depends on the number of possible connections and this, in turns, depends on the *nature* of the connections. If the connections are to properties which are *relational* with respect to the learner's information resource then the search space is potentially *infinitely* large, simply because there are generally an infinite number of possible relationships that may be defined over a given set of data. Conversely, if the connections are to absolutes within the learner's information, then the search space is only finitely large, because the number of combinations of absolute values extracted from a finite data source is necessarily finite.

What this tells us is that learning tasks may be decomposed into two distinct complexity classes:

- finitely complex **non-relational** tasks involving the identification of properties which are absolute with respect to the learner's information resource, and

- infinitely complex **relational** tasks involving the identification of properties which are relational with respect to the learner's information resource.

Researchers have been familiar with this distinction for many years (cf. [Dietterich and Michalski, 1983, Clark and Thornton, 1997]). It is, in fact, common practice to refer to methods specifically intended for use on relational tasks as **relational learning methods** (cf. [Muggleton, 1992, Mitchell, 1997]). Following this practice, the present paper will make a distinction between **relational learning**, meaning learning specifically adapted for relational tasks, and **empirical** or **non-relational learning** meaning learning specifically adapted for non-relational tasks.

The complexity of the learning task in practice

Given the complexity properties noted above it is clear that, before addressing a specific learning task, a learner should ideally decide if the task is relational or non-relational. Unfortunately, this decision often cannot be made with any confidence. The distinction between relational and non-relational learning is clear in theory. But the task of classifying problems as relational or non-relational is fraught with difficulty.

At first sight, it looks as if it should be possible to classify tasks as relational or non-relational depending on the degree of clustering in evidence. Imagine that the data available to the learner agent take the form of combinations of values of variables---a very common scenario---and that each particular combination of values is treated as an n-dimensional datapoint. If the task is relational, we know that particular actions are contingent on relationships among the value combinations and that actions should therefore not be correlated with absolute values (or datapoint coordinates) in any way. Datapoints associated with the same action should not share coordinates and therefore *not* cluster together. If, on the other hand, the task is non-relational, absolute values (datapoint coordinates) are associated with particular actions; so datapoints should tend to share coordinates and should tend to cluster together.

In a relational task, then, we expect the data to show no clustering. And certainly, in extreme cases, this is precisely what is found. Consider, the so-called `parity task.' In a parity task, the action and all the data values are represented as binary digits. The nature of the task is such that the action `1' should be produced just in case there are an odd number of 1s among the data. The action `0' should be produced in all other cases. A parity task involving combinations of three data values can be written out textually as in figure-below.

```
0  0  0   -->   0
0  0  1   -->   1
0  1  0   -->   1
0  1  1   -->   0
1  0  0   -->   1
1  0  1   -->   0
1  1  0   -->   0
1  1  1   -->   1
```

3-value parity task.

Each line here shows the association between a particular action (to the right of the arrow) and a combination of three data values (to the left of the arrow). Note how the action is `0' in those cases when there is an even number of 1s among the data values, and `1' otherwise.

Parity is a *perfectly* relational task since the action depends exclusively on a relationship among the data. The net effect is that the distribution of datapoints is `perfectly unclustered.' Due to the nature of the rule underlying the parity task, a single increment or decrement of any data variable `flips' the associated action from 1 to 0, or vice versa. Thus, in the data space, datapoints associated with one action always appear *adjacent* to datapoints with the opposite action. Nearest neighbours within the space are thus guaranteed to have different actions. There is no clustering *whatsoever*: the action labels create a perfect checkerboard effect.

With respect to the parity tasks, the association between lack of clustering and relationality is thus quite clear. But when we turn attention to other sorts of relational task, the association becomes more blurred. Consider, for example, the task shown in figure-below.

```
3 4  --> 0
8 2  --> 1
7 6  --> 1
8 9  --> 0
   .
   .
   .
```

Ambiguous task.

The rule here is that the action should be 1 just in case the first data value is greater than the second. This rule is characteristically relational and we might expect that the problem---like parity---will exhibit no clustering. But if we inspect the associated distribution of datapoints we find that this is not the case. The explanation is easy to find. The task is *less* than perfectly relational. Actions do not depend exclusively on relationships among the data. Absolute values do have some significance in the determination of output. Zero, for example, cannot be greater-than any other non-negative integer. A zero as the first value thus constitutes evidence that the produced action should be `0'. The net effect is that the data for this task do show a certain degree of clustering.

Characteristically relational problems, then, may embody non-relational aspects which `show through' in the form of clustering. But this is not the only source of clustering effects we need to consider. It is also possible for a task to be characteristically *hybrid*, i.e., to exhibit different associations based on totally unrelated types of effect. For example, consider the problem in figure-below.

```
c d a b --> f
a b d b --> h
e c d e --> h
c b a e --> f
a c d e --> f
b c a e --> f
b d d e --> h
e d a c --> f
a c d c --> h
c d a c --> h
c c a e -->
```

Hybrid learning task.

Several of the cases with an `a' in the third variable have `f' as their action. There thus exists a cluster of points which share `a' in the third variable and `f' as action. A learner observing this effect might classify the problem as non-relational and attempt to find data/action associations based solely on absolute values. This might lead to the learner guessing that a value of `a' in the third variable *indicates* that the action should be `f'. But the original classification here is dubious. In addition to the association noted, there is also an effect in which cases exhibiting *duplicated* data values tend to have `h' as their action. This effect is based on a relationship among the data. A learner focussing on this effect might thus classify the problem as relational and proceed to a totally different conclusion.

We have to conclude, then, that clustering (or lack of it) is not a reliable indicator for relationality. There are a number of possible sources of clustering in (the data for) a characteristically relational problem, e.g.

- the task may have genuine, non-relational aspects and thus exhibit a degree of meaningful clustering. The `greater-than' task is a good example.
- The task may be represented to the learner in such a way as to create artificial non-relational aspects. An example of this situation is a parity task whose representation includes an extra input variable whose value always effectively records the parity status of the original inputs is an example.

In both of these situations, the exhibited clustering is useful for the purposes of learning, i.e., it can be used as the basis for generalisation. There are two further situations, however, in which the clustering is of no use whatsoever.

- The clusters may be an artifact of the way in which the learner's data have been selected or generated.
- The clusters may be the results of some sort of noise or data error.

In both of these cases, the clusters observed in the data are merely sampling artifacts and thus of no use whatsoever within the learning process.

To summarise, in a characteristically relational task we may see clustering effects arising from non-relational aspects of the task, characteristics of the task encoding, characteristics of the data selection process or noise/error. Effects due to the task encoding, data selection or noise may be termed **incidental**, on the grounds that their relationship with the underlying problem is not meaningful. Within this grouping, effects due to characteristics of the task representation may be termed **generalising** while effects due to the data selection or noise may be termed **non-generalising**. The various possibilities are tabulated in figure-below.

Origins of clusters in characteristically relational problems.

Typical scenarios

Despite the difficulties noted, it remains the case that non-relationality *does* produce clustering while relationality does tend to eliminate it. The existence or lack of clustering therefore can serve as a guide for *tentative* classification decisions. We have seen that almost all learning tasks show a certain degree of clustering. But the more clustering they exhibit the stronger the evidence in favour of a non-relational classification. The range of possibilities is illustrated in figure-below. Each task here is displayed as a 2-dimensional graph and is therefore assumed to be defined in terms of two, numeric data variables and one action variable, whose value is either `1' or `0'. The problems represent typical scenarios from the perfectly non-relational to the perfectly relational.

Clustering scenarios.

In the `perfect clustering' scenario, all the inputs whose output label is 1 are in the left half of the input space. Other the inputs whose output label is 0 are in the right half of the space. The data are thus perfectly organised into two, cleanly separated regions, definable in terms of a single, axis-aligned boundary.

Next, we have a scenario showing strong clustering. The inputs here are still cleanly separated into uniformly instantiated regions. But the organisation is less than perfect. The clusters would need to be defined in terms of, say, four circular regions.

The next scenario shows weak clustering. Now the input points are distributed in a more complex fashion. There are some uniformly instantiated regions but these do not have particularly regular shapes. The situation might correspond to a characteristically relational problem which shows some non-relational effects. Or it might simply correspond to a complex non-relational problem.

Finally, we have the `perfect checkerboard' scenario. In this situation the two types of input are perfectly mixed up. This is the extreme case of input data disorganisation, i.e., maximum `sensation entropy.' Every point has as its nearest neighbour a datapoint with a different label. Absolute input values (i.e., coordinates) therefore have no significance *whatsoever* in the determination of output.

The perfect checkerboard scenario is the logical extreme of the relational dimension. And as has already been noted, all parity problems produce perfect checkerboard distributions. But do all checkerboards arise from valid parity tasks? Recall that the parity task is defined in terms of binary data and action values. Thus each dimension of the data space has only two values. If we draw out the checkerboard for a 2-bit parity problem then it has the appearance of figure-below.

Checkerboard pattern for a parity problem.

Checkerboards whose dimensions are all 2-valued can always be viewed as n-bit parity problems---n being the number of dimensions. Problems such as the one shown in Figure 8-1, which have more than two values per dimension, but only two distinct output values, obviously cannot be interpreted as parity problems. However, they can be interpreted in terms of a modulus-addition operation, a generalisation of the parity rule.((Modulus addition behaves the same as ordinary additionexcept that the result is constrained to lie between 0 and a maximum called the **modulus** value. Applying modulus addition to some numbers involves finding their sum and then subtracting the **modulus** value M until the value lies between 0 and M-1.)) A mapping such as the one shown in Figure 8-1 can be interpreted as defining a modulus-addition function using input values which range between 0 and 7 and a modulus value of 2.

In fact, any scenario which is checkerboard-like in terms of having different output labels attached to all nearest-neighbour datapoints, can be interpreted as defining a modulus-addition mapping as long as the number of values in each dimension is a multiple of the modulus value. This is a simple consequence of the fact that, within a modulus-addition operation, incrementing or decrementing any input value *always* has the effect of changing the output value. The net effect is that within any modulus-addition mapping, nearest-neighbour datapoints always have *different* outputs. Any modulus-addition problem thus necessarily has a checkerboard-like appearance. And, by the same token, all problems with a checkerboard-like appearance can be viewed as modulus-addition problems.

The existence of universal non-uniformities

The analysis of learning-task complexity presented above involves two main steps. First we conceptualise all learning as *behaviour* learning, i.e, as involving the acquisition of a disposition to produce certain `actions' in certain `situations.' Second, we observe that this task has two distinct forms: a non-relational form, involving a finitely complex search and a relational form involving an infinitely complex search. The strength of the argument derives from the observation that any form of learning must involve identifying connections between elements of informational data and that these connections may be rendered in terms of relational or non-relational properties. There is no inductive element here whatsoever. The observations are derived using simple deduction. The argument is therefore based purely on first principles. But is it the argument we want? Does it do the job Wolpert thinks needs to be done? In particular, does it allow us to justify the assumption of generic non-uniformities?

Possible, probable and de facto non-uniformity

The analysis allows us to characterise the ways in which a generalisation problem may be solved. It thus implicitly allows us to characterise generalisation problems which *cannot* be solved. In particular we can say that a a particular generalisation task may be

- non-relationally learnable,
- hybridly learnable,
- relationally learnable,
- unlearnable.

A task falls into one of the three learnable categories if the relational (non-relational) effects exhibited in the training data are also exhibited in the testing data. All tasks in which there is no such carry over of effects fall into the unlearnable category. What does this tell us about the NFL result?

Recall that in the NFL scenario we assume nothing is known about the application domain. The result is that we cannot assume any non-uniformity in the distribution of problems. With respect to a particular classification system we must therefore assume that problems are uniformly distributed between the various categories. If we take the categories to be as specified above (three learnable and one unlearable) then we have to make the assumption that one quarter of all problems in an arbitrary domain will be unlearnable. For an effective learner, all unlearnable problems will be problems on which performance is worse than chance. Thus in this scenario the expectation must be that an average (effective) learner will perform better than chance on 50% of all problems and worse than chance on just 25% of problems.((An alternative approach would be to treat the categories as just `learnable' and `unlearnable'. In this case the situation becomes still more favourable for the effective learner. Our a priori expectation must be that it will produce better than chance performance on *all* problems from the domain. This is, of course, the expectation that we would have on intuitive grounds anyway.))

The establishment of a principled distinction between learnable and unlearnable problems thus allows us to eliminate the paradoxical NFL prediction that effective learners will perform no better than random guessing. But we can take the argument one step further. For most intents and purposes, data fed to learning algorithms is derived by careful sampling of a particular phenomenon (cf. the problems in the UCI repository of Machine Learning Databases). Such datasets are constructed according to certain rules, the most prominent of which concerns the statistical independence of the data variables. In general, `real-world' datasets (e.g., the `Breast Cancer' dataset) are generated in such a way as to ensure that the data variables are maximally independent. The net effect of this is to effectively eliminate the *possibility* of relational effects. The requirement for statistical independence among data variables tends to guarantee that the associated dataset will constitute a *non*-relational learning problem.

On a conservative estimate perhaps as many as 95% of all real-world learning problems are represented in terms of independent or approximately independent variables and must thus be considered non-relational *by design*. In practice then, most learning problems selected from, say, the UCI repository are effectively designed to be within the non-relationally-learnable category. For the machine learning practitioner there is, then, a *de facto* non-uniformity which can be safely assumed on the basis of the principles of *dataset design*. A randomly selected problem is almost certain to be contained within the subset of problems which are non-relationally learnable.

Geometric seperability of frequently-used datasets

We can demonstrate the non-relationality of typical learning problems empirically. We have already noted that the solving of a non-relationally-learnable problem involves exploitation of data clustering. Thus, we expect any such problem to exhibit reasonably well clustered data. One way to measure the degree of clustering in a particular dataset is to compute its **geometric seperability** [Thornton, 1997] which is just the proportion of datapoints whose nearest neighbours share the same output classification.

```
geometric-seperability(f) = \frac{\sum_{i=1}^{n}
      f(x_{i}) + f(x_{i}^{'}) + 1 \bmod 2} {n}
```

Here, f is a binary target function, x is the data set, x_{i}^{'} is the nearest neighbour of x_{i} and n is the total number of data. The nearest neighbour function is assumed to utilise a suitable metric, e.g., a Manhalobis metric for symbolic data or a Euclidean metric for spatial data.

Geometric seperability is a measure of the degree to which datapoints with the same action cluster together. In some sense, it is a generalisation of the linear-separability concept [Minsky and Papert, 1988]. Although not a boolean measure (i.e., a predicate), geometric seperability can be viewed, like the linear-separability concept, as differentiating tasks which are appropriate for a particular learning strategy.((A satisfying property of geometric seperability is the fact that it is zero for all parity tasks, as per expectation.)) The strategy in this case is non-relational (i.e., similarity-based) learning. Only if the geometric seperability for a particular task is high is this strategy likely to be effective.

The geometric seperability values for 16 of the most frequently used Machine Learning datasets [Holte, 1993] is tabulated in table-below. As we expect, in all cases the values are well above zero. The average geometric seperability value is, in fact, 0.85.

Dataset	BC	CH	GL	G2	HD	HE	HO	HY
GS	67.31	82.82	73.6	81.6	76.24	61.94	76.9	97.76
Dataset	IR	LA	LY	MU	SE	SO	VO	V1
GS	94.0	94.74	77.03	100.0	93.19	100.0	92.87	87.47

Interestingly, the geometric seperability values may be treated as *expected* generalisation levels for a 1-nearest-neighbour classifier. The generalisation performance of a 1-nearest-neighbour classifier depends on the degree to which data in the testing sample have nearest-neighbours in the training sample with identical target actions.((We assume that the same metric is used for the nearest-neighbour classifier as was used in computing the GS.)) The proportion of nearest neighbours in the dataset which share the same action is identical to the expected proportion of such cases in a randomly selected testing set. Thus, on average, the 1-nearest-neighbour classifier will produce a level of generalisation which is identical to the GS value.

Summary

The NFL result seems, at first glance, to be `bad news' for learning and generalisation, since it seems to suggest that it is impossible for a method to produce average performance which is any better than that achieved by random guessing. However, Wolpert and others have been gone to great lengths to emphasise that the news is not quite this bad. What NFL shows, they claim, is that it is the assumptions we make about the domain which make all the difference. Provided appropriate assumptions about non-uniformities are made, arbitrarily good performance may be obtained.

The aim of the present paper has been to suggest that even this may be an over-reading of the implications of the NFL result. The NFL work assumes, rather counter-intuitively, that there is no a priori difference between learnable and unlearnable problems and that therefore all input/output mappings definable over a given domain have to be considered plausible learning problems. The net effect of this is that there can be no a priori reason why a particular learner should encounter problems upon which it performs well and therefore no a priori reason why any learning method should perform above the chance level on average.

However, once we accept that there is an a priori distinction to be made between learnable and unlearnable problems we can eliminate a proportion of the input/output mappings from any domain on *a priori* grounds. The expected performance of an arbitrary, effective learner is then automatically raised above the chance level. But establishing the existence of an a priori distinction between learnable and unlearnable problems involves close analysis of what learning involves.

The approach taken in this paper (almost certainly not the only approach possible) involves lumping all learning tasks together under the heading of `behaviour learning.' We then decompose the learning (generalisation) task into two subtasks: (1) the identification of connections between informational data and action events and (2) the implementation of mechanisms designed to consolidate those connections. Given that the latter task is agent-specific, learning-task complexity must be measured in terms of the former subtask. And as we have seen, the complexity of this is related to the nature of the connections which need to be identified and in particular, to whether they are instantiated in terms of relational or absolute properties of the learner's information resource.

This analysis, aside from offering a detailed characterisation of what `learnability' really means, has the advantage of highlighting the fact that properly prepared learning problems tend to be non-relational by design. The net effect of this is that learning methods which aim to solve problems *in terms of* non-relational effects (i.e., in terms of data similarity((Learning methods which may be classified as similarity-based include the CART algorithms [Breiman *et al.* 1984], the **competitive learning** regime of Rumelhart and Zipser [1986], the **Kohonen net** [Kohonen, 1984] and in fact any algorithmic method which is based on the method of **clustering** [Diday and Simon, 1980]. Methods which are clearly excluded from this class include the `BACON' methods of Langley and co-workers [Langley, 1977; Langley, 1978; Langley *et al.* 1983; Langley *et al.* 1987] and related methods such as [Wolff, 1978; Wolff, 1980; Lenat, 1982; Wnek and Michalski, 1994]. These carry out explicit searches for relational effects and in many cases ignore similarity effects altogether.)) or data clustering) will tend to perform well across the board. The almost universally good (i.e., above chance) performance of methods such as C4.5 and Backpropagation may be explained in these terms.

References

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

Clark, A. and Thornton, C. (1997). Trading spaces: computation, representation and the limits of uninformed learning. *Behaviour and Brain Sciences, 20* (pp. 57-90). Cambridge University Press.

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.

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.

Holte, R. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, *3* (pp. 63-91).

Hume, D. (1740). *A Treatise of Human Nature* (second edition). Oxford University Press.

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

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

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.

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.

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

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.

Minsky, M. and Papert, S. (1988). *Perceptrons: An Introduction to Computational Geometry* (expanded edn). Cambridge, Mass.: MIT Press.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.

Muggleton, S. (Ed.) (1992). *Inductive Logic Programming*. Academic Press.

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.

Schaffer, C. (1994). Conservation law for generalization performance. *Proceedings of the International Conference on Machine Learning* (pp. 259-265). July 10th-13th, Rutgers University, New Brunswick, New Jersey.

Thornton, C. (1997). Separability is a learner's best friend. In J.A. Bullinaria, D.W. Glasspool and G. Houghton (Eds.), *Proceedings of the Fourth Neural Computation and Psychology Workshop: Connectionist Representations* (pp. 40-47). London: Springer-Verlag.

Thornton, C. and Clark, A. (Forthcoming). Reading the generalizer's mind. *Behaviour and Brain Sciences*, Cambridge University Press.

Thrun, S., Bala, J., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., De Jong, K., Dzeroski, S., Fisher, D., Fahlman, S., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R., Mitchell, T., Pachowicz, P., Reich, Y., Vafaie, H., Van de Welde, W., Wenzel, W., Wnek, J. and Zhang, J. (1991). The MONK's problems - a performance comparison of different learning algorithms. CMU-CS-91-197, School of Computer Science, Carnegie-Mellon University.

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.

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

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.

Wolpert, D. (1992). On the connection between in-sample testing and generalization error. *Complex Systems*, *6* (pp. 47-94).

Wolpert, D. (1995a). The relationship between PAC, the statistical physics framework, the bayesian framework, and the VC framework. In D. Wolpert (Ed.), *The Mathematics of Generalization*. Addison-Wesley.

Wolpert, D. (1995b). On overfitting avoidance as bias. SFI-TR-92-03-5001, Santa Fe Institute.

Wolpert, D. (1996a). The existence of a priori distinctions between learning algorithms. *Neural Computation, 8*, No. 7.

Wolpert, D. (1996b). The lack of a priori distinctions between learning algorithms. *Neural Computation*, *8,* No. 7.

Wolpert, D. and Macready, W. (1995). *No Free Lunch Theorems for Search*. Unpublished MS.

Page created on: Tue Sep 15 13:27:23 BST 1998
Feedback to *Chris.Thornton@cogs.susx.ac.uk*
hits this year