

Separability is a Learner's Best Friend

Chris Thornton
COGS/Informatics
University of Sussex
Brighton
BN1 9QH
UK

c.thornton@sussex.ac.uk
www.christhornton.eu

July 15, 2008

Abstract

Geometric separability is a generalisation of linear separability, familiar to many from Minsky and Papert's analysis of the Perceptron learning method. The concept forms a novel dimension along which to conceptualise learning methods. The present paper shows how geometric separability can be defined and demonstrates that it accurately predicts the performance of a at least one empirical learning method.

1 Separability

Given the 'spatial' way in which we tend to conceptualise the process of learning, it is not surprising to discover that many learning methods operate on a distinctively 'geometric' basis. Implicitly or explicitly, their aim is to divide the input-space up into regions such that all the points in a given region map onto the same output-space point. We expect this approach to achieve good generalisation since, once a region has been identified in which all the known points map onto the same output, it seems safe to assume that *most* points in that region will map onto the same output. We also expect it to achieve good storage performance since, once we have identified the regions and associated output-points we can throw away the original data.

But these region- or boundary-oriented methods all make a rather strong assumption about the input data and the underlying task. They assume, in effect, that the function is 'smooth' [Rendell and Seshu, 1990] and that input points with the same target output will therefore tend to cluster together in the same region of the input space. Unfortunately, this assumption turns out to be valid only in certain situations. In other situations, it is totally invalid.

The general idea that **boundary methods** have severe limitations is familiar to the Cognitive Science community via the work of Minsky and Papert [1969, 1988] on the Perceptron learning method. The Perceptron is a neural-network method which, in its simplest manifestation, attempts to acquire a target function using the simplest boundary method of all. Making the assumption that there are just two points in the output space, it tries to find a single, *linear* boundary which separates all the input points associated with one of the outputs from points associated with the other. The primitive nature of the operation paves the way for excellent performance. But it also imposes dire limitations. Only in certain types of task will we find the points mapping onto one output neatly separated by a line boundary from the points mapping onto the other. In other tasks the points will be dispersed in a more complex fashion. Thus the Perceptron method is only effective in certain cases.

Minsky and Papert described tasks which could be handled using the Perceptron method as ‘linearly separable.’ They pointed out that many tasks which we might consider to be straightforward, such as deciding on whether two binary values are the same — the so-called ‘parity’ task — do not fall into the linearly separable category. This led many researchers to turn their backs on the perceptron method and on neural network methods in general. In hindsight, however, we can see that there is no reason to ‘take it out’ on the Perceptron. The Perceptron method is a special-purpose member in a special-purpose class of methods. First and foremost, it is a boundary method and therefore assumes that the input/output function is smooth (points with similar outputs cluster together). Second, it is a method which assumes that just one linear boundary will do the trick. It makes a particularly strong assumption about the data. But when that assumption is valid it operates in a very efficient way.

Minsky and Papert focussed primarily on the Perceptron and related architectures. But their separability result is easily generalised to other learning methods. All boundary methods operate on the assumption that the target function is smooth and that the target function can therefore be acquired by placing boundaries around regions of uniform¹ input points. Thus these methods effectively assume that the data exhibit ‘regional’ or *geometric* separability and are only effective when that particular type of separability exists within the relevant data.

But this concept of *geometric separability* differs from that of linear separability in an important way. Linear separability is a boolean property. A particular task either *is* or *is not* linearly separable. Geometric separability, on the other hand, is a continuous property. Provided we are allowed to make regions arbitrarily small — small enough to enclose a single input point — then *all* tasks are geometrically separable in the limit, since we can represent any function in terms of some set of point-sized, regions. Thus the issue is not whether a task exhibits geometric separability but the degree to which it does so, i.e., the degree to which regions of uniform input points can be identified.

Geometric separability, then, is a measure of the degree to which inputs associated with the same output cluster together. As this property increases, we see an increase in the frequency with which nearest-neighbours in the input

space share the same output, and vice versa. Thus a convenient way of measuring geometric separability — and one which does not necessitate making any restrictive assumptions about the ‘shape’ of the regions that will be utilised — is in terms of the proportion of nearest-neighbour inputs in the function which share the same output.

We thus define the *Geometric Separability Index* or *GSI* for a given task f to be the proportion of nearest neighbour inputs which share the same output:

$$GSI(f) = \frac{\sum_{i=1}^n f(x_i) + f(x'_i) + 1 \bmod 2}{n}$$

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 Manhattan metric for symbolic data or a Euclidean metric for spatial data.

The GSI generalises Minsky and Papert’s linear-separability concept to the general case of boundary methods. Although it is not a boolean measure (i.e., a predicate) it can be viewed, like the linear-separability concept, as differentiating tasks which are appropriate for a particular learning strategy. The strategy in this case is boundary-making. If the GSI for a particular task is high, boundary-making methods are effective. If it is low, they are not.

2 Geometric separability of common tasks

A satisfying property of the GSI is the fact that its value is zero for all parity problems (such as the notorious ‘XOR’ problem which featured prominently in Minsky and Papert’s analysis). In a parity problem, a single increment or decrement of any input variable flips the output (classification) from positive to negative, or vice versa. Thus, in the input space, points associated with one output always appear next to points with the opposite output. Nearest neighbours always have opposite outputs and the geometric separability of a parity problem is necessarily zero.

This is, of course, exactly what we would expect. Minsky and Papert showed that Perceptrons could not learn parity functions. The Perceptron is a simple boundary method. Boundary methods produce poor performance when geometric separability is low. The geometric separability of any parity problem is zero. Thus, Perceptron’s should not be able to learn parity problems. In fact, any boundary method tends to produce poor performance on parity problems [Thornton, 1996].

But what about other common learning problems? Do frequently used learning problems such as those which reside in the UCI repository², tend to have high, low, or indifferent geometric separability? To investigate this issue, the geometric separability of all 16 ‘frequently used datasets’ (as featured in the Holte [1993] study) was tested. The results are shown in Table 1.

The GSI for each dataset is shown in the first cell of each column. The second cell shows the error rate on the same dataset as reported by Holte for the

| | | | | | | | | |
|---------|------|------|------|------|------|------|------|------|
| Dataset | BC | CH | GL | G2 | HD | HE | HO | HY |
| GSI | 0.67 | 0.91 | 0.73 | 0.82 | 0.76 | 0.62 | 0.77 | 0.98 |
| C4.5 | 0.72 | 0.99 | 0.63 | 0.74 | 0.74 | 0.81 | 0.84 | 0.99 |
| Dataset | IR | LA | LY | MU | SE | SO | VO | V1 |
| GSI | 0.94 | 0.95 | 0.77 | 1.00 | 0.93 | 1.00 | 0.93 | 0.88 |
| C4.5 | 0.94 | 0.77 | 0.77 | 1.00 | 0.98 | 0.98 | 0.96 | 0.89 |

Table 1: Comparison of GSIs with C4.5 generalisation rate on Holte datasets.

C4.5 decision tree learner, one of the best performing of all empirical learning methods. A striking feature of the results is that the GSI for half of these datasets (CH, MY, IR, LA, MU, SE, SO, VO,) is extremely high, i.e., greater than 0.9. The implication is that these datasets are rich in geometric separability and thus highly suitable for processing by boundary methods. A second striking feature is the fact that the GSI turns out to predict the performance of the C4.5 method reasonably well. The average difference between the GSI for a dataset and the corresponding mean generalisation rate reported by Holte is 0.056, i.e., slightly under 6 percentage points. The correspondence between GSI and C4.5 generalisation rates is illustrated graphically in Figure 1.

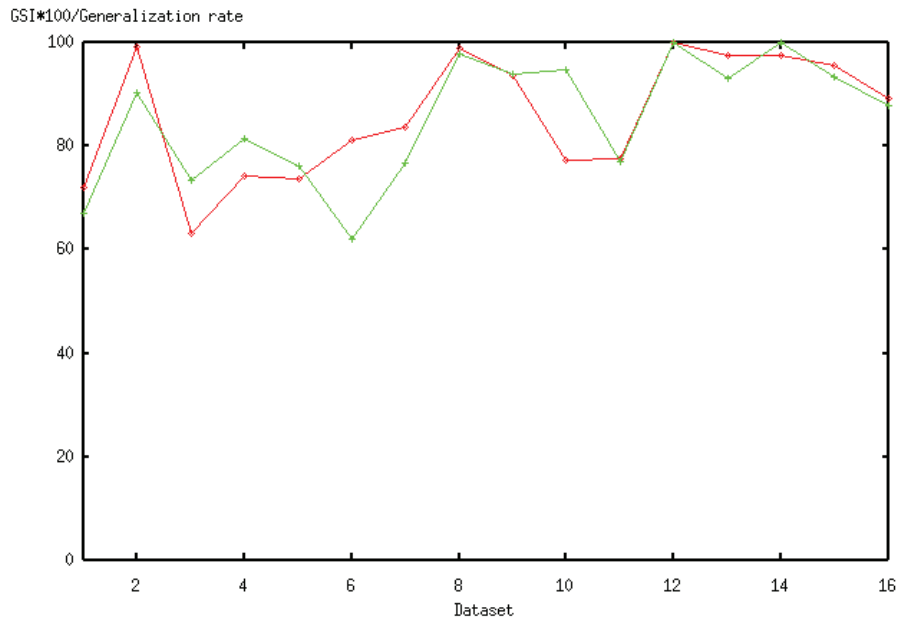


Figure 1: Graphical comparison of GSIs and C4.5 generalisation.

The broad correspondence between C4.5 generalisation rates and GSIs is, again, as per expectation. C4.5 operates by constructing hyper-rectangular boundaries in the input space and is thus unambiguously a boundary method. We therefore expect that its performance on a particular task will be broadly predicted by the GSI for that task.

Interestingly, GSI values can be viewed as identifying the *expected* performance of a 1-nearest-neighbour classifier. The generalisation performance of a 1-nearest-neighbour classifier depends on the degree to which inputs in the testing sample have nearest-neighbours in the training sample with identical target outputs.³ Clearly, the proportion of nearest neighbours in the dataset which share the same output 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 GSI. We note in passing the fact that though the 1-nearest-neighbour classifier is a somewhat primitive boundary method, it produces performance (on average) which in over half of the 16 cases is either *equal to or greater* than that of the state-of-the-art method C4.5.⁴

3 The relational origins of geometric inseparability

The empirical tests on the UCI datasets show that most of the 16 frequently used datasets tasks exhibit high geometric separability. Typical boundary methods such as C4.5 and backpropagation produce the expected levels of high performance on such problems. Is this due cause for celebration? Can we safely assume that these results suggest that learning problems are geometrically separable *in general*?

We have already seen that one important class of tasks — the parity problems — all exhibit a zero GSI. But this turns out to be just the tip of an iceberg. By reasoning backwards from what we know about geometric separability, we can demonstrate that any task which involves the recognition or testing of a *relationship* will typically exhibit negligible geometric separability.

In a task exhibiting high geometric separability, inputs with identical outputs tend to cluster together in the input space. But this clustering effect requires that inputs with identical outputs exhibit similar ‘coordinates’ or values. But if they do so, we know that absolute values (or continuous ranges of values) must be significant in the determination of the output. We can then deduce counter-factually that problems in which absolute component values are known to have *no* significance in the determination of output *must* exhibit low geometric separability.

Such problems are termed *relational* and parity is the extreme case. In a parity problem, absolute variables values have no significance whatsoever in the determination of output: it is only the relationships which count. This is reflected in the fact that the GSI value for parity problems is always zero.

In other seemingly relational problems (e.g., the greater-than relationship between two numbers), absolute values may have some significance in the determination of output: zero for example cannot be greater-than any other non-negative integer. So a zero might constitute evidence that the value of the greater-than relationship is false. In such cases the GSI of the task will be non-zero but still low. The general rule is that the more characteristically relational a problem is, the less significant are absolute values and the lower the geometric separability of the task is likely to be. Boundary methods tend to perform poorly on all relational problems. But the more characteristically relational they are (i.e., the less significant absolute values are in the determination of output) the lower the GSI and the poorer the expected performance. Widespread (if often implicit) recognition of this fact has meant that characteristically relational problems are typically addressed using special-purpose relational learning methods (e.g., ILP methods [Muggleton, 1992]). In those rare cases where a boundary method is tested on a relational task the performance tends to be poor [Thornton, 1996].

Interestingly, the argument which allows us to deduce that low geometric separability is associated with relational problems can be turned around to demonstrate that problems with high geometric separability are always efficiently processed using boundary methods. If a problem is non-relational, the absolute variable values (or continuous ranges of same) must be significant in the determination of output. If this is the case, inputs sharing the same output will tend to share coordinates and will therefore tend to cluster together in the input space. Thus boundary methods must be appropriate for all non-relational problems.

The situation comes into focus, therefore, as a *dichotomy*. On the one hand we have learning tasks exhibiting high geometric separability. These are characteristically non-relational and are efficiently processed by boundary methods. On the other hand we have learning tasks exhibiting low geometric separability. These are characteristically relational and are not efficiently processed by boundary methods. One implication of this is that we could, if we like, make the separability distinction in terms of the difference between relational and non-relational tasks. This was in fact the approach taken in the earlier paper, Trading Spaces. That paper distinguished relational tasks from non-relational or ‘statistical’ tasks.⁵

Of course in saying that geometrically separable tasks are well processed by boundary methods, we are making a broad generalisation. There are a large number of boundary methods originating in superficially distinct fields of investigation (connectionism, machine learning, genetic algorithms, pattern recognition, case-based learning etc.). Each one tends to make boundaries and thus identify regions in a slightly different way. The Perceptron favours extreme simplicity: it uses a single linear boundary. ID3 and its near relation C4.5 constructs hyper-rectangular, axis-aligned regions. The MLP, in its vanilla form employing one layer of hidden units, uses multiple linear boundaries (cf. Lipmann figure) to identify more or less arbitrarily-shaped regions. The crossover-based genetic algorithm can be viewed as manipulating hyperplanes [Holland, 1975].

4 Concluding comments

The concept of geometric separability gets close to the heart of what empirical learning is really all about. Empirical learning *is* the exploitation of geometric separability. The success of any attempted empirical learning is thus inextricably bound up with the degree of geometric separability inherent in the relevant data. In those cases where we know that geometric separability is negligible (e.g., relational problems) we need to move beyond utilisation of familiar, statistically-oriented learning methods into the domain of relational methods. The fact that such methods are not easily implemented as neural networks presents an interesting challenge for future work.

Notes

¹Uniform in the sense of having the same target output.

²<http://www.ics.uci.edu/mllearn/MLRepository.html>

³We assume that the same metric is used for the nearest-neighbour classifier as was used in computing the GSI.

⁴Of course, C4.5 and its near relation ID3 have been shown on numerous occasions to produce performance which is comparable to that of many other empirical learning methods, e.g., the connectionist MLP method [Fisher and McKusick, 1989]

⁵The terms type-1 and type-2 were also used to label the non-relational and relational cases respectively.

References

- Fisher, D. and McKusick, K. (1989). An empirical comparison of ID3 and back-propagation. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 788-793). Morgan Kaufmann.
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.
- Holte, R. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 3 (pp. 63-91).
- Minsky, M. and Papert, S. (1969). *Perceptrons*. Cambridge, Mass.: MIT Press.
- Minsky, M. and Papert, S. (1988). *Perceptrons: An Introduction to Computational Geometry* (expanded edn). Cambridge, Mass.: MIT Press.
- Muggleton, S. (Ed.) (1992). *Inductive Logic Programming*. Academic Press.
- Rendell, L. and Seshu, R. (1990). Learning hard concepts through constructive induction. *Computational Intelligence*, 6 (pp. 247-270).
- Thornton, C. (1996). Parity: the problem that won't go away. In G. McCalla (Ed.), *Proceeding of AI-96* (Toronto, Canada) (pp. 362-374). Springer.