Virtual Seens and the Frequently Used Dataset

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

January 22, 1999

Abstract

The paper considers the situation in which a learner's testing set contains close approximations of cases which appear in the training set. Such cases can be considered 'virtual seens' since they are approximately seen by the learner. Generalisation measures which do not take account of the frequency of virtual seens may be misleading. The paper shows that the 1-NN algorithm can be used to derive a normalising baseline for generalisation statistics. The normalisation process is demonstrated through application to Holte's [1993] study in which the generalisation performance of the 1R algorithm was tested against C4.5 on 16 commonly used datasets.

1 Introduction

In some cases, the training data that we present to a learner may comprise incomensurable objects. More frequently, the data exist within some space and similarity or distance measures are feasible. In this situation it is possible for a learner's testing set to contain close approximations of cases which appear in the corresponding training set. This may make the measurement of generalisation a delicate matter.

Consider the following vector of attribute values.

```
0.000 yes 3.444 left up 2 119 8.342 72 t 65.225 f
```

Imagine that this forms a single case in a learner's testing set. We consider this case to be *unseen* for the learner if it does not appear in the corresponding training set. But what if the following case appears in the training set?

```
0.000 yes 3.444 left up 2 119 8.343 72 t 65.225 f
```

Although the two cases seem identical there is a small difference: the seventh value is 8.343 rather than 8.342. Technically, then, we can still treat the original case as unseen. But in doing so we may feel a little uncomfortable. An extremely close approximation of the unseen case exists in the training set. Any learner which sees all the cases in the training set has virtually seen the unseen case.

Where such 'virtual seen' cases exist within testing data, measures of generalisation performance may be misleading. Ideally, experimenters should eliminate virtual seens from any testing data before usage. This may involve ensuring that every test case has a sufficient level of *dissimilarity* with every training case.

Where experiments have been carried out *without* any prior elimination of virtual seens, testing set error may be an unreliable guide to real generalisation performance. However, it may be possible to compensate by normalising the generalisation measures with respect to the *frequency* of occurrence of virtual unseens. This involves (a) determining the average frequency of virtual unseens in relevantly sized testing sets and (b) reexpressing the generalisation performance relative to this value.

Detecting the presence of virtual seens would appear to involve finding datapoints whose mutual distance is very small. But this begs the question what range of distance is to count as 'small.' Data-point distance vary enormously from dataset to dataset so our definition cannot be stated in terms of any absolute distance value but rather on a relative distance value. However, this still does not quite meet requirements since data-point densities can also vary widely from dataset to dataset.

A natural solution is to say that a test case should count as a virtual seen if there is a point in the training set which (a) is closer to the test case than to any other case and (b) has the same classification. This criterion factors out the variability in both data distances and densities. It also permits us to detect virtual seens through application of a simple 'nearest-neighbour' regime. In fact when we apply a 1-NN algorithm [Duda and Hart, 1973] to a training/testing set combination, the performance obtained is precisely the average frequency of virtual seens in the testing set. If the 1-NN algorithm generalises correctly on an unseen case, the training set must contain a case which is (a) closer to the unseen than any other case in the training set and (b) shows the same output (class) value as the unseen case. The unseen thus constitutes a virtual seen, and the guess generated by the algorithm can be thought of as a 'confident generalisation.'

To utilise the 1-NN algorithm for this purpose we proceed as follows.

(1) Choose a size for the testing set.

(2) Construct the testing set by randomly selecting (without replacement) the appropriate number of cases from the dataset.

(3) Form the training set out of the remaining cases.

(4) Compute the generalisation performance of the 1-NN algorithm on the given training/testing set combination.

By averaging the generalisation performance over a sufficiently large sample we can obtain an estimate of the frequency with which virtual seens will be found in testing/training sets of the given proportions. Any absolute generalisation measure which has been derived using the *same* proportions can then be converted into a relative measure simply by subtracting the 1-NN generalisation performance. The value obtained provides a measure of generalisation which discounts the possibilities for 'lookup' of virtual seens.

2 Application of NN-normalisation to the Holte Study

To get some insight into how useful the normalisation method might be in practice, it was applied to the results of Holte's 16-datasets study [Holte, 1993]. This focussed on commonly used datasets from the UCI repository of Machine Learning Databases.[1] These were BC (breast-cancer), CH (chess-end-games), GL (glass), G2 (glass with classes 1 and 3 combined and classes 4 through 7 deleted), HD (heart-disease), HE (hepatitis), HO (horse-colic), HY (hypothyroid), IR (iris), LA (labor-negotiations), LY (lymphography), MU (agaricus-lepiota), SE (sick-euthyroid), SO (soybean-small), VO (house-votes-84), V1 (VO with 'physician-pay-freeze' attribute deleted). For full details of the datasets used consult Appendix B of Holte's paper.

In Holte's study, the generalisation performance of a 1-level decision tree learner (1R) was tested and averaged over 25 runs using training sets derived by randomly selecting 2/3 of the cases from an original dataset. Holte showed that the performance of this learner was almost as good as that of C4.5 [Quinlan, 1993] even though it was restricted to the formation of simple hypotheses based on single attributes. He concluded that a 'simplicity first' methodology is appropriate in machine learning.

To apply the NN-normalisation to Holte's data it was necessary to test the generalisation of the 1-NN algorithm on training sets derived according to his protocol. This involved using training sets which were 2/3 the size of the original dataset and averaging over 25 runs for each dataset. In fact averages were taken over 50 runs for each dataset; i.e., the samples were twice as large as those used in the original study.

[1] This is accessible on the world-wide-web at URL `http://www.ics.uci.edu/AI/ML/Machine-Learning.html`

The 1-NN algorithm used for this experiment used basic 'city-block' distance measure. The distance $D(A, B)$ between two cases A and B was defined to be

$$D(A, B) = \sum_{i=1}^{n} d(A_i, B_i)$$

where $d(A_i, B_i)$ was the normalised numeric difference between A_i and B_i if both values were numeric, and the number of explicit character differences expressed as a fraction of the length of the shortest string, if both values were strings (i.e., symbolic values). In the case of one of the values being missing, the difference was defined as 1/10 of the maximum difference.

The generalisation performance achieved by the 1-NN algorithm using 2/3-sized training sets (the size Holte used) is shown in Table 1. The performance of C4.5 is also shown.

Dataset	BC	CH	GL	G2	HD	HE	HO	HY
1R	68.7	67.6	53.8	72.9	73.4	76.3	81.0	97.2
C4.5	72.0	99.2	63.2	74.3	73.6	81.2	83.6	99.1
1-NN	69.7	90.1	70.1	80.6	78.1	79.3	78.5	96.9
Dataset	IR	LA	LY	MU	SE	SO	VO	V1
1R	93.5	71.5	70.7	98.4	95.0	81.0	95.2	86.8
C4.5	93.8	77.2	77.5	99.9	97.7	97.5	95.6	89.4
1-NN	94.6	85.8	76.8	100.0	87.9	100.0	93.1	88.1

The performance data for 1-NN and C4.5 are shown in Figure 1 in graph form. Interestingly, the 1-NN algorithm produced performance which was either comparable or superior to C4.5 in seven of the 16 cases. In the remaining nine cases the performance was on average no more than 3 percentage points worse than that of C4.5. In all cases the performance was superior to that of Holte's 1R algorithm.

The measured performance of 1-NN algorithm in this study appears to be broadly compatible with its performance (or the performance of a K-NN variant) as reported in similar studies such as [Aha and Kibler, 1989] and [Henery, 1994]. However, the performance obtained in this study is in general superior to that reported by Weiss and Kapouleas [1989]. They recorded a mean generalisation level of 65.3 on the BC dataset whereas the figure obtained in the present study was 69.7. Similarly, they recorded a generalisation level of 95.3 on the HY dataset but the present figure is 96.9. On the other hand they recorded a *higher* level of performance on IR (96.0) although in this case they were employing a cross-validation method in addition to the basic algorithm. The performance of the 1-NN method on LA and VO is also markedly better than that recorded by [Bergadano, Kodratoff and Morik, 1992] and [Aha and Kibler, 1989] for K-NN variants of the method. These differences may well all be due to the fact that

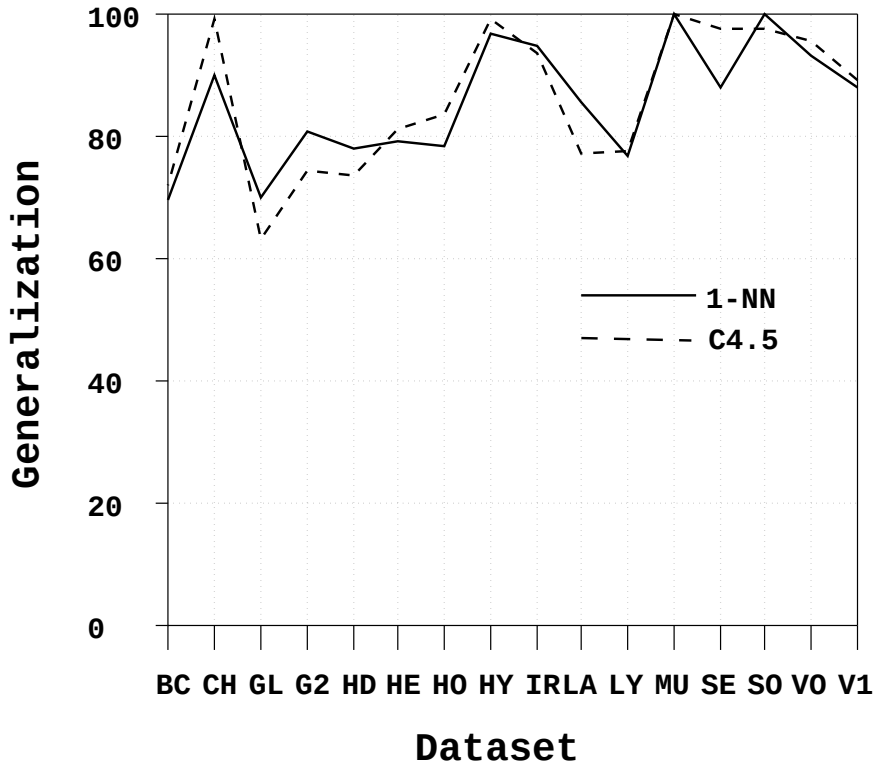

Figure 1: Performance of 1-NN and C4.5 on Holte datasets.

the present study used a distance function which took character differences into account in string comparisons.

To use the results of this study for nomalising the generalisation measures reported for C4.5, say, on these datasets we should subtract the 1-NN performance from the C4.5 performance. The fact that the 1-NN method outperforms C4.5 on seven of the datasets means that the relative generalisation performance of C4.5 on these datasets should actually be viewed as null or negative. On the remaining datasets (BC, CH, HE, HO, HY, LY, SE VO and V1) the relative generalisation is positive. On average, over all the datasets, mean relative generalisation is 0.29%. In other words, on average the method produces generalisation that is around 1/4 of a percent more than what we would expect using 'lookup' of virtual seen cases. The implication is that with these datasets, training sets containing 2/3 of the original cases do not pose a substantive test of generalisation.

This result is in agreement with Friedman's analysis [Friedman, 1994] which explains the surprising robustness of NN methods against the so-called 'curse of dimensionality' in terms of the redundant distributional properties of common datasets. It is also in agreement with the general implications of Holte's study. Holte showed that very simple learning processes can produce good performance on these problems. The present study has shown much the same thing. But of course 1-NN and 1R are 'simple' in different ways. 1R attempts to construct a rule based on observations on the *minimum* number of attributes. 1-NN on the other hand uses a rule which takes into account observations on *all* the available attributes. Thus the results of this study show that the Holte datasets are simple in at least two different senses.

3 The effect of varying the training set proportion

To try to get a better idea about the reasons for the rather small difference between the performance of 1-NN, 1R and C4.5 on the Holte datasets, experiments were carried out to determine the average performance of the 1-NN algorithm on a *range* of training set sizes. The performance of the algorithm was in fact sampled on training sets built by randomly choosing 0.5%, 2%, 33% (1/3) and 66% (2/3), 98% and 99.5% of the original cases. The generalisation performance was then averaged over 50 runs at each training set size. The results of these experiments are displayed in Figure 2.

In general, one expects the performance of the NN algorithm to increase with the size of the training set. The performance should be very poor if the training set is nearly empty and very good if the training set contains nearly all the possible cases. Thus, given the training set proportions used in this survey, we expect generalisation curves to approximate an upwards sloping diagonal. In fact, *none* of the curves shown in Figure 2have this form. The curve for the GL dataset is perhaps the best approximation. But in general the curves are remarkably *flat*.

The implications of this are worth some consideration. In order for a dataset to have a high, flat generalisation curve, it is essential that the 1-NN algorithm performs well on nearly empty training sets, i.e., training sets which include only a small proportion of the dataset. But we should only expect this to occur if the data are highly organised, i.e., if the classes in the data are very cleanly separated. In this situation any example taken from a class can serve as an exemplar for the class and thus provide a 1-NN algorithm with an effective representation of that class. Thus a very few examples may well suffice to produce excellent performance from the 1-NN algorithm.

Of course, even with clean separation of classes, a 1-NN algorithm cannot produce good performance unless the training set contains at least one exemplar

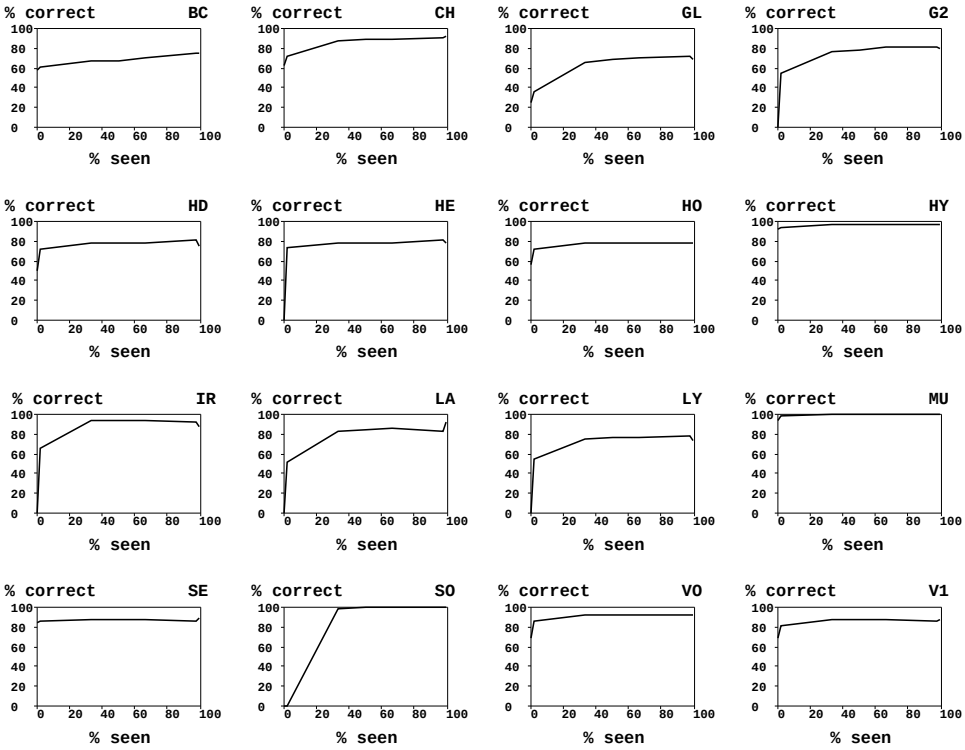

Figure 2: Performance of 1-NN on Holte training sets of different sizes.

from each class. If the training sets of a particular size are simply too small to contain at least one example from each class then good performance is impossible. This explains why the curves for LA, IR, LY, HE and G2 start at zero but then slope up very rapidly: it is just the 0.5% sized training sets that are too small to contain one example from each class. The SO dataset is an exception to the general pattern since it is both extremely small and has a relatively large number of target outputs (4). The generalisation curve for SO thus ramps up more slowly. Both the 0.5% sized training sets and the 2% sized training sets are too small to contain at least one example from each class.

One feature of the experimental data is hard to explain. The curves for problems IR, LY, HE, GL and HD curve down at the 99.5% level. This implies that the performance for the 99.5% sized training sets was actually worse than that for the 98% sized training sets. It is not clear what feature of the data might account for this. It is hoped that further work with differently sized training sets may shed some light here.

4 Summary and Conclusion

In some cases, the instances that we present to a learner may be incomensurable and thus impossible to test for similarity. More frequently, there is an explicit or implicit distance metric over instances. In this situation, a given testing set may contain very close approximations of cases from the training set. The paper has described such cases as 'virtual seens' and noted that generalisation statistics derived in the presence of virtual seens may be misleading or ambiguous.

The performance of the 1-NN algorithm can be used to derive a generalisation baseline against which true or *relative* generalisation can be measured. This approach was demonstrated though an application involving Holte's comparative study of the performance of 1R and C4.5 on 16 commonly used datasets from the UCI repository. The results of this experiment revealed that most of the datasets in the Holte selection contain data showing *extremely* clean separation between classes. For all the Holte benchmarks, the performance achievable through 'lookup' of virtual seen cases is extremely close to the performance level achieved by learning methods such as C4.5. We have to conclude therefore that these datasets do not pose a substantive tests of generalisation. If we equate learning ability with generalization ability then we have to conclude that these datasets do not effectively test anything that we can meaningfully call 'learning'.

This conclusion is a little startling given the central role that the UCI datasets have played in the evolution of Machine Learning methods. However, the wider implications are hard to trace out. Certainly, we can dispense with the oft-stated assumption that 'real-world' problems are necessarily challenging for learning methods. All of the Holte datasets are derived from the 'real-world' but none of them turn out to be challenging. The difficulty of a problem would thus appear to be quite independent of the domain from which it is derived. Real world problems may be challenging for learning methods. But there is no guarantee that they will be.

References

[1] Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis.* New York: Wiley.

[2] Aha, D. and Kibler, D. (1989). Noise-tolerant instance-based learning algorithms. *Proceedings of the Eleventh Joint Conference on Artificial Intelligence* (pp. 794-799). Morgan Kaufmann.

[3] Weiss, S. and Kapouleas, I. (1989). An empiricial comparison of pattern recognition, neural nets and machine learning classification methods. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 781-787). Morgan Kaufmann.

[4] Bergadano, F., Kodratoff, Y. and Morik, K. (1992). Machine learning and knowledge acquisition: summary of reserach contributions presented at IJCAI'91. *AI Communications*, *5*, No. 1 (pp. 19-24).

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

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

[7] Friedman, J. (1994). *Flexible Metric Nearest Neighbor Classification*. Unpublished MS.

[8] Henery, R. (1994). Review of previous empirical comparisons. In D. Michie, D. Speigelhalter and C. Taylor (Eds.), *Machine Learning, Neural and Statistical Classification*. Ellis Horwood.