A Yardstick for the Evaluation of Case-Based Classifiers

A D Griffiths and D G Bridge

Department of Computer Science, University of York, YORK YO1 5DD, UK
Email: {tony|dgb}@minster.york.ac.uk

Abstract. This paper proposes that the generalisation capabilities of a case-based reasoning system can be evaluated by comparison with a 'rote-learning' algorithm which uses a very simple generalisation strategy. Two such algorithms are defined, and expressions for their classification accuracy are derived as a function of the size of training sample. A series of experiments using artificial and 'natural' data sets is described in which the learning curve for a case-based learner is compared with those for the apparently trivial rote-learning learning algorithms. The results show that in a number of 'plausible' situations, the learning curves for a simple case-based learner and the 'majority' rote-learner can barely be distinguished, although a domain is demonstrated where favourable performance from the case-based learner is observed. This suggests that the maxim of case-based reasoning that 'similar problems have similar solutions' may be useful as the basis of a generalisation strategy only in selected domains.

1 Introduction

Case-based reasoning systems operate on the assumption that 'similar cases have similar solutions'. This is an heuristic argument which will be correct in some instances and misleading in others; evaluating the success of this heuristic is an important issue in understanding case-based reasoning properly. It is unsurprising that a case-based classifier may correctly classify problem instances which exactly match some case in the case-base. The difficult task is to correctly classify the previously unseen problem instances; this will be referred to as *generalisation*. The generalisation strategy used in case-based classification is to classify an unseen problem instance according to the classification values recorded for the problem's near neighbours in the case-base. There are, however, many other generalisation strategies that may be used in order to predict classifications for the unseen instances. This paper will consider whether 'nearest neighbour' generalisation will necessarily be more or less successful than any of the other available strategies.

The approach to the evaluation of case-based classifiers proposed in this paper compares the performance of a case-based system to that of two very simple *rote-learning* algorithms. These algorithms remember any examples presented to them in the training sample. On unseen instances however, the first algorithm simply chooses a classification at random, while the second algorithm assigns a

single fixed classification value. Thus, these learning algorithms show the level of accuracy in classification that can be achieved by a combination of rote-learning of the training sample with the simplest of generalisation strategies. Since case-based classification will also in practice give perfect recall of the training sample, subject only to very weak constraints on the similarity measure [3], this comparison allows the contribution of the nearest neighbour generalisation strategy to be isolated. As a minimum criterion for success, it is proposed that a case-based generalisation strategy should outperform the two 'naïve' strategies used by the two rote-learners.

Section 2 below introduces the definitions and notation that will be necessary to present the results of the paper, while in Section 3 the two rote-learning algorithms are defined, and equations for their learning curves are derived. Section 4 demonstrates the use of these learning curves as yardsticks for case-based classifiers, and Section 5 discusses the conclusions that can be drawn from this comparison.

2 Definitions

The paper is concerned with the problem of classification. It is assumed that there is a 'target function' which maps from a space X_1 of *problem descriptions* to a space X_2 of *classification values*, so that each possible problem description in X_1 is associated with one and only one classification value in X_2. X_1 will also be referred to as the example space and X_2 as the solution space. The task of learning a classifier involves choosing, on the basis of the correct classifications given only for some limited number of the possible problem descriptions, a function from $(X_1 \rightarrow X_2)$ which is as close as possible to the target function. A probability distribution on the example space will be referred to as μ, the probability that a randomly chosen example is a member of some set $X \subseteq X_1$ will be written μX and the *conditional* probability that an example $x \epsilon X_1$ will have some property $p(x)$ given membership of some set $X \subseteq X_1$ will be written $\mu\{x \epsilon X | p(x)\}$. A sequence of examples from $(X_1)^*$ will be written $\overline{x}$, while a training sample $\overline{x}_t \epsilon (X_1 \times X_2)^*$ is a sequence of examples from X_1 'labelled' according to the target function t so that for each pair of elements $\langle x_i^1, x_i^2 \rangle$ in $\overline{x}_t$ we have $t(x_i^1) = x_i^2$. The probability of an m-sample chosen by independent random sampling from X_1, will be given by the joint probability distribution μ^m defined on the product space $(X_1)^m$, where $\mu^m\{\overline{x}\} = \mu\{x_1\} \times \ldots \times \mu\{x_m\}$.

Systems which learn to classify may be compared in terms of their *accuracy*, meaning simply the probability with which a randomly chosen example will be correctly classified by the chosen classifier. Hence the accuracy of a learning algorithm is the probability $\mu\{x \epsilon X_1 | L(\overline{x}_t)(x) = t(x)\}$ where $L(\overline{x}_t)$ is the hypothesis produced by learning algorithm L given a training sample $\overline{x}_t$ and t is the target function. Since this will depend on both the training sample and the target function, expected accuracy is defined with respect to a particular target function by summing over all possible training samples, weighted by their respective probabilities, and with respect to a particular set of target functions,

called the target class, by summing over all admissible target functions.

$$EA^{L}_{(t,\mu)}(m) = \sum_{\overline{x}\,\epsilon\,(X_1)^m} \mu^m\{\overline{x}\} \cdot \mu\{x\,\epsilon\,X_1 | L(\overline{x}_t)(x) = t(x)\} \tag{1}$$

$$EA^{L}_{(T,\mu)}(m) = \frac{1}{|T|} \cdot \sum_{t\,\epsilon\,T} EA^{L}_{(t,\mu)}(m) \tag{2}$$

This work assumes a simple model of case-based reasoning. A case-base is a set of description-classification pairs $CB\,\epsilon\,\mathcal{P}(X_1 \times X_2)$ s.t. for each exemplar in the case-base $\langle x_1, x_2\rangle\,\epsilon\,CB$, then $t(x_1) = x_2$ for the current target function t. A similarity measure σ is a function over pairs of descriptions returning a normalised real value indicating the degree of similarity between the two descriptions. For any problem description $x\,\epsilon\,X_1$, the set of *near neighbours* of the problem description is defined as the set of exemplars from the case-base whose descriptions are maximally similar to x according to σ:

$$\begin{aligned}&\forall x\,\epsilon\,X_1 \cdot NN(x, CB, \sigma) \mathrel{\hat{=}} \\ &\quad \{(x_1, x_2) | (x_1, x_2)\,\epsilon\,CB \wedge \forall (x'_1, x'_2)\,\epsilon\,CB \cdot \sigma(x, x_1) \geq \sigma(x, x'_1)\}\end{aligned} \tag{3}$$

The decision function for a case-based classifier is defined by specifying how a single classification value can be extracted from the set of nearest neighbours $NN(x, CB, \sigma)$. In this work, we assume that a 'preference ordering' is defined over the solution space to resolve any ties that might occur, although other options are available [3]. i.e. In this paper the function defined by a case-base CB and similarity measure σ is the function f where

$$f(x) = \max_{\sqsupseteq}\{x_2 | \langle x_1, x_2\rangle\,\epsilon\,NN(x, CB, \sigma)\} \tag{4}$$

for some preference ordering $\sqsupseteq$. Having defined case-based classification, it is now necessary to define the rote-learning algorithms which will provide the yardstick for the evaluation of these classifiers.

3 Learning Curves for Rote-Learning Algorithms

The representation used by the rote learning algorithms defined here are very closely related to the decision tables used by learners such as the 'Decision Table Majority' algorithm studied by Kohavi [6]. The classification rule for Decision Table Majority is, "given an unlabelled instance, a decision table classifier searches for exact matches in the decision table [...]. If no instances are found, the majority class of the [decision table] is returned; otherwise, the majority class of all matching instances is returned" [ibid]. The two algorithms below are variants of this approach. However, the representation used here is referred to as an 'instance table' rather than a decision table since it is an important feature of operational decision tables that the table is accessed using only a subset of the features of a representation [ibid]. An instance table on the other hand requires

an exact match with the representation of a problem instance; it can be pictured as an array or table with one cell for each member of the example space, each of which contains exactly one solution value. The function represented by an instance table τ is simply the function f_τ s.t. $\forall x \epsilon X_1 \cdot f_\tau(x) = \tau[x]$. The first yardstick algorithm 'generalises' simply by assigning solution values to unseen problem instances at random.

Definition 1. L_1 **Random instance-table learner.**

`forall` $x \epsilon X_1$ `set` $\tau[x] = r$
`for` $i = 1$ `to` m `do`
    `set` $\tau[x_i^1] = x_i^2$
`set` $L_1(\overline{x}_t) = f_\tau$

where $t \epsilon (X_1 \rightarrow X_2)$ is a target function for L_1, $\overline{x}_t = \langle (x_i^1, x_i^2) \rangle_{i=1}^m$ is a training sample for t in $(X_1 \times X_2)^m$, and r is a random variable which varies over X_2 with a uniform probability distribution.

The second rote learner assigns to any unseen instances the solution value with the highest *a priori* probability, i.e. a value $x_0 \epsilon X_2$ which maximises $\mu\{x \epsilon X_1 | t(x) = x_0\}$. The algorithm presented below assumes sufficient background knowledge to choose an appropriate default value x_0. While this might seem too strong a supposition, it is similar to assuming that a preference ordering is available when defining the decision function for case-based classification. Additionally, L_2 is a useful yardstick since it represents the limit of classification accuracy that may be achieved by a naïve generalisation strategy which assigns a single value to all unseen instances.

Definition 2. L_2 **Majority instance-table learner.**

`forall` $x \epsilon X_1$ `set` $\tau[x] = x_0$
`for` $i = 1$ `to` m `do`
    `set` $\tau[x_i^1] = x_i^2$
`set` $L_2(\overline{x}_t) = f_\tau$

where $t \epsilon (X_1 \rightarrow X_2)$ is a target function for L_2, $\overline{x}_t = \langle (x_i^1, x_i^2) \rangle_{i=1}^m$ is a training sample for t in $(X_1 \times X_2)^m$ and x_0 is a member of X_2 which maximises $\mu\{x \epsilon X_1 | t(x) = x_0\}$.

The expected accuracy of L_1 and L_2 may be straightforwardly calculated as follows, as a function of the *expected coverage* of the example space by the training sample.

Definition 3. Expected Coverage of a Set. The expected coverage of a set X by a sample of size m according to a sampling distribution μ defined on X, denoted $\chi_\mu(m)$, is defined as the probability, having chosen a sample of m elements drawn independently from X according to probability distribution μ,

that a further element drawn independently from X according to μ will have appeared at least once in the sample.

$$\chi_\mu(m) = \sum_{\overline{x} \in (X)^m} \mu^m\{\overline{x}\}.\mu E_{\overline{x}}$$

where $E_{\overline{x}} = \bigcup_{1 \leq i \leq m} \{x_i\}$ is the set of distinct problem descriptions appearing in $\overline{x}$.

The equations for the learning curves require the following lemma, restating the overall expected accuracy of a consistent learning algorithm in terms of the expected coverage $\chi_\mu(m)$. Proofs below are sketched or omitted; details will be available in [2].

Lemma 4. *The expected accuracy of a consistent learning algorithm L may be re-expressed in terms of $\chi_\mu(m)$ as follows:*

$$EA^L_{(T,\mu)} = \chi_\mu(m) + \frac{1}{|T|} \cdot \sum_{t \in T} \sum_{\overline{x} \in (X_1)^m} \mu^m\{\overline{x}\} \cdot \mu(X_1 \setminus E_{\overline{x}}) \cdot \mu\{x \in (X_1 \setminus E_{\overline{x}}) | L(\overline{x}_t)(x) = t(x)\}$$

Proposition 5. Expected Accuracy of L_1. *The expected accuracy of the random instance table learner L_1 for any target class T, having been trained on a training sample of size m, is given by the following expression:*

$$EA^{L_1}_{(T,\mu)} = \frac{n\chi_\mu(m) + 1}{n+1}$$

where $n + 1 = |X_2|$ and $\chi_\mu(m)$ is the expected coverage of the example space.

Proof: The expression for the expected accuracy of a consistent learner given in Lemma 4 can be instantiated by finding a term for the probability that a previously unseen member of the example space will be correctly classified by the hypothesis of L_1. Taking any target $t \in T$ and any training sample for t, $\overline{x} \in (X_1 \times X_2)^m$, let $h = L_1(\overline{x}_t)$ be the hypothesis of L_1 on that training sample. Now $\mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid h(x) = t(x)\} = \sum_{x' \in X_2} \mu\{x \in (X_1 \setminus E_{\overline{x}}) | h(x) = t(x) = x'\}$. Given an unseen problem instance $x \in (X_1 \setminus E_{\overline{x}})$, the value of $h(x)$ is determined independently of all other considerations. Additionally, it follows immediately from the definition of the generalisation strategy that $\mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid h(x) = x'\} = \frac{1}{|X_2|}$ for any specific solution value $x' \in X_2$. Hence:

$$\begin{aligned}
&\mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid h(x) = t(x)\} \\
&= \sum_{x' \in X_2} \mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid h(x) = x'\} \cdot \mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid t(x) = x'\} \\
&= \frac{1}{|X_2|} \cdot \sum_{x' \in X_2} \mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid t(x) = x'\} = \frac{1}{|X_2|} = \frac{1}{n+1}
\end{aligned}$$

Applying lemma 4, then the overall expected accuracy $EA^{L_1}_{(T,\mu)}(m)$ is reckoned as follows:

$$\begin{aligned} EA^{L_1}_{(T,\mu)} &= \chi_\mu(m) + \frac{1}{|T|} \cdot \sum_{t \in T} \sum_{\overline{x} \in (X_1)^m} \mu^m\{\overline{x}\} \cdot \mu(X_1 \setminus E_{\overline{x}}) \cdot \frac{1}{n+1} \\ &= \chi_\mu(m) + \frac{1}{n+1} - \frac{1}{n+1} \sum_{\overline{x} \in (X_1)^m} \mu^m\{\overline{x}\} \cdot \mu E_{\overline{x}} \cdot \frac{1}{|T|} \sum_{t \in T} 1 \\ &= \chi_\mu(m) + \frac{1}{n+1} - \frac{\chi_\mu(m)}{n+1} = \frac{n\chi_\mu(m)+1}{n+1} \qquad \square \end{aligned}$$

Since L_2 is defined with knowledge of the target function, the expected accuracy must be analysed with respect to that particular target, rather than summing over some target class T.

Proposition 6. Expected Accuracy of L_2. *The expected accuracy of the majority instance table learner L_2, having been trained on a training sample of size m for a particular target function t, is given by the following expression:*

$$EA^{L_2}_{(t,\mu)} = (1-\mu_0)\chi_\mu(m) + \mu_0$$

where $\mu_0 = \max_{x' \in X_2}\{\mu\{x \in X_1 | t(x) = x'\}\}$ and $\chi_\mu(m)$ is the expected coverage of the example space.

Proof: Taking, as above, that $\mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid h(x) = t(x)\} = \sum_{x' \in X_2} \mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid h(x) = t(x) = x'\}$, consider that $h(x) = x_0$ for any unseen problem instance $x \in (X_1 \setminus E_{\overline{x}})$. Thus for solution values $x' \in X_2$ distinct from x_0, we have $h(x) = t(x) = x'$ with probability zero, while for $x' = x_0$ we have $h(x) = t(x) = x'$ iff $t(x) = x_0$. The sum above reduces to $\mu\{x \in (X_1 \setminus E_{\overline{x}}) | t(x) = x_0\}$. Since the value taken by $t(x)$ is independent of whether x has appeared in the sample (unlike the value $h(x)$), then $\mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid h(x) = t(x)\} = \mu\{x \in (X_1 \setminus E_{\overline{x}}) \mid t(x) = x_0\} = \mu\{x \in X_1 | t(x) = x_0\} = \mu_0$. The result is completed in a similar fashion to Proposition 5. $\square$

Notice that in the case of equal *a priori* probabilities for all solution values, then $\mu_0 = \frac{1}{n+1}$ and $EA^{L_1}_{(t,\mu)}(m) = EA^{L_2}_{(t,\mu)}(m)$. Additionally, in the special case of a uniform sampling distribution on the example space, then a simple closed form can be found for $\chi_\mu(m)$:

Proposition 7. *The* expected coverage *of a set X by $\overline{x}$, a sample of m elements sampled independently from X according to a* uniform *distribution on X, denoted $\chi(m)$, can be calculated as follows:*

$$\chi(m) = \sum_{i=1}^{m} \frac{(-1)^{i-1}}{|X|^i} \binom{m}{i}$$

Proof: [Sketch] In the general case it may be shown that:

$$\chi_\mu(k+1) = \chi_\mu(k) + \sum_{\overline{x}\,\epsilon\,(X)^k} \mu^k\{\overline{x}\} \sum_{x\,\epsilon\,(X\setminus E_{\overline{x}})} (\mu\{x\})^2 \tag{5}$$

In the special case of a uniform sampling distribution on X, the following recurrence relation for $\chi(k)$ is derived:

$$\chi(k+1) = \chi(k) + \frac{1}{|X|}(1 - \chi(k)) \tag{6}$$

Hence the result by the use of the weak inductive hypothesis. □

4 Comparison of Case-Based Learning and Rote Learning Algorithms

This section reports the results of some preliminary experiments intended to measure the performance of a simple case-based learning algorithm against the yardstick of the learning curves derived for L_1 and L_2 in the previous section. The case-based learning algorithm will be $CB1$ as defined in [5] [4] [3], namely a straightforward case-based classifier which collects all available cases into its case-base, which calculates similarity by counting the proportion of features of the representation on which two descriptions agree, and which resolves ties between equally near neighbours using a fixed preference ordering defined over the space of solution values [3].

The experimental method used to observe the learning curves for $CB1$ is simply to choose a function randomly from the target class and to generate a training sample of fixed maximum size by choosing elements of the example space independently at random and labelling them with the value of the target function. Each member of the training sample in turn is presented to the classifier as a test instance and the correctness of the classification noted; the instance is then added to the case-base as specified by $CB1$. This is repeated for a large number of different target functions and the proportion of correctly classified training instances is calculated for each value of m (the number of instances taken so far from the training sample).

Comparison over all possible targets

Figure 1 shows learning curves for $CB1$, L_1 and L_2, measured over the space of all possible classifications of the space of 6-dimensional binary vectors (i.e. the target class is the set of boolean functions $(\{0,1\}^6 \to \{0,1\})$. The decision function used by $CB1$ prefers negative classifications to positive ones $(0 \sqsupseteq 1)$ for those situations where both positive and negative exemplars appear in the set of near neighbours. Figure 2 illustrates the same experiment in a domain where features may take values from the set $V \mathrel{\hat{=}} \{val_0, val_1, val_2, val_3\}$, rather than

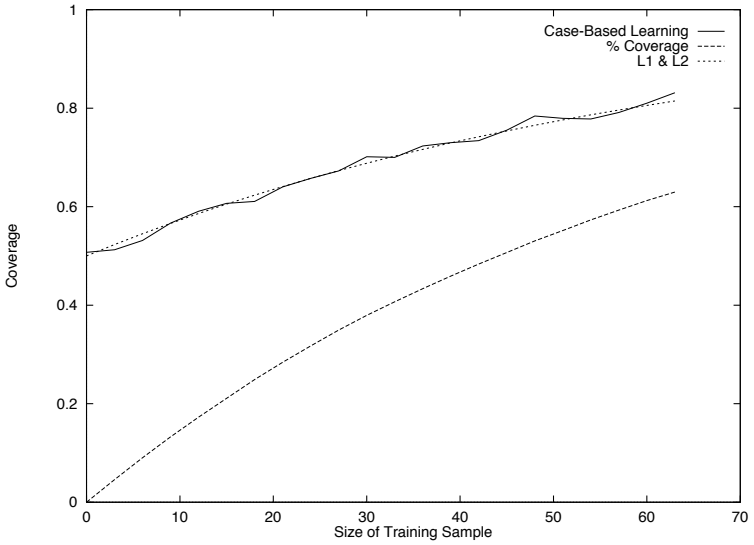

Fig. 1. Average Learning Curves For All Binary Functions

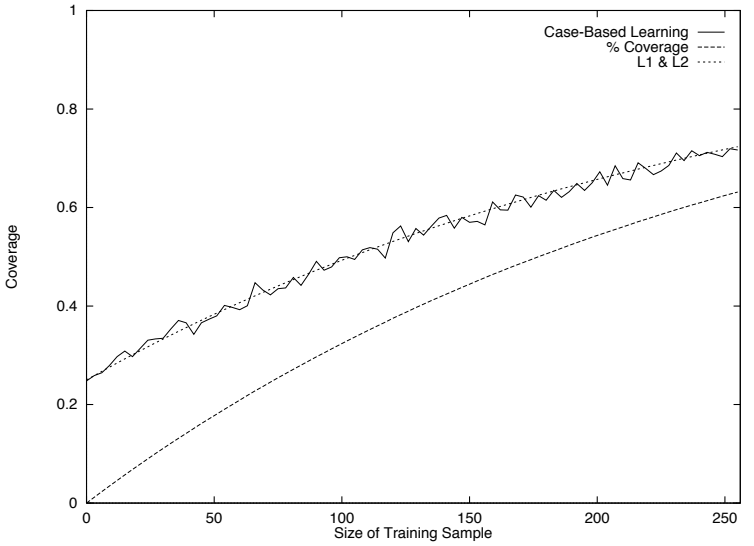

Fig. 2. Average Learning Curves For All Functions on 4 dimensional space

just the binary values $\{0, 1\}$. Specifically, the target class for the experiment graphed in Figure 2 is the space of all possible functions from the space of descriptions $(V \times V \times V \times V)$ into the space V. Additionally, ties between equally near neighbours are resolved for the case-based classifier by a preference ordering defining $val_0 \sqsupseteq val_1 \sqsupseteq val_2 \sqsupseteq val_3$. In both cases the observations are averaged over 500 experiments.

Figures 1 and 2 show the expected coverage of the example space $\chi(m)$ calculated according to the result of proposition 7, i.e. the proportion of the example space which can be expected to have been observed directly in the training sample. This is plotted alongside the observed learning curve for $CB1$, measured in the fashion described above, and the learning curves for L_1 and L_2 given by Propositions 5 and 6. As noted previously, in the case of all solution values being equally likely, then the learning curves for L_1 and L_2 coincide. For Figure 1, then $X_2 = \{0, 1\}$ and the curve plotted is $\frac{1}{2}\chi(m) + \frac{1}{2}$, while for Figure 2 $X_2 = V$ and the curve is $\frac{3}{4}\chi(m) + \frac{1}{4}$, whichever default value is assigned by L_2.

In Figures 1 and 2 the curves for case-based learning and the rote-learning algorithms coincide exactly; this is no surprise since in the maximum entropy case where all possible target mappings are equally likely, then no generalisation strategy can do better than random guessing and all are essentially equivalent [9] [8]. The success of the equations derived in the previous section in describing this case however lends confidence to the accuracy of the analysis.

Rote-learning of Monomial Target Functions

Monomial functions are those boolean functions which can be defined by a simple conjunction of (possibly negated) literals. There are 3^N such functions defined on an example space of N-dimensional binary-valued feature vectors, and $2^k \cdot \binom{N}{k}$ such functions defined by conjunctions of exactly k literals.

Previous work has considered in detail the behaviour of the case-based learning of monomial target functions [5] [4]. The performance of case-based learning was shown to be poor compared to learning algorithms which directly manipulated a representation of a monomial function, especially when the classifier uses a fixed similarity measure. Here, in addition, the behaviour of a simple case-based learning algorithm is compared with a rote-learning algorithm on this restricted target class. Since the concept defined by a monomial function is generally small, the appropriate 'default' value for the majority rote learner L_2 will be 0; most members of the example space will *not* be positive instances of a monomial function. The expected accuracy of L_2 when the value '0' is assigned to all unseen problem instances is calculated below with respect to the set of monomial functions M_N.

Proposition 8. *The expected accuracy of a rote-learning algorithm assigning the value 0 to all unseen problem instances, taken with respect to the space of*

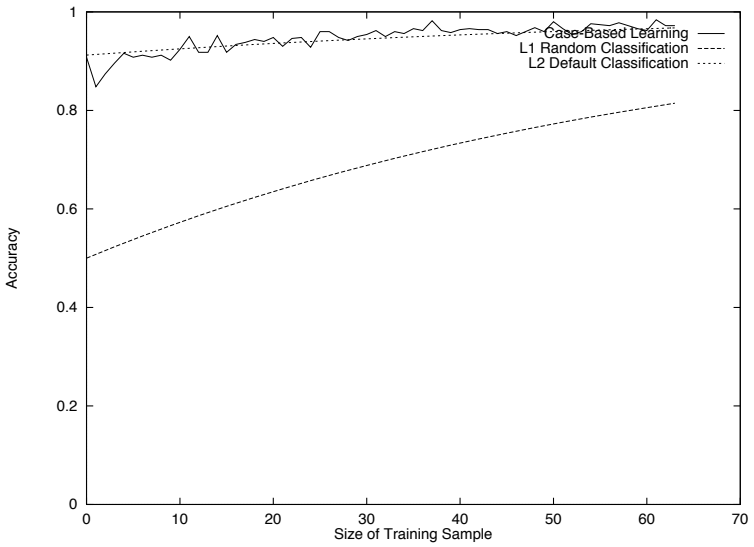

Fig. 3. Average Learning Curves For Monomial Functions

monomial functions M_N and with respect to a uniform sampling distribution on the example space, has the following value:

$$EA^{L_2}_{(M_N,U)} = \frac{2^N}{3^N}\chi(m) + (1 - \frac{2^N}{3^N})$$

where N is the size of the feature vectors in the example space, U denotes the uniform probability distribution on X_1 and $\chi(m)$ is the expected coverage of the example space with respect to a U.

Proof: In Proposition 6 the expected accuracy of L_2 with respect to a specific target function t was given as $(1 - \mu_0)\chi(m) + \mu_0$, where μ_0 is the probability of the majority classification. Hence for the space of monomial functions the expected accuracy with respect to a uniform sampling distribution is:

$$EA^{L_2}_{(M_N,U)} = \frac{1}{3^N}\sum_{t \in M_N}(1 - \mu_t)\chi(m) + \mu_t$$

where μ_t is the prior probability of the default classification with respect to the target function t. As noted above, there are $2^k \cdot \binom{N}{k}$ monomial functions defined by a conjunction of k out of N literals, and each such function will have 2^{N-k} positive instances. Hence, given a uniform sampling distribution on the example space, $\mu_t = (1 - 2^{-k})$, and

$$EA^{L_2}_{M_N} = \frac{1}{3^N}\sum_{k=0}^{N} 2^k \cdot \binom{N}{k} \cdot \left[2^{-k}\chi(m) + (1 - 2^{-k})\right]$$

$$= \frac{2^N}{3^N}\chi(m) + 1 - \frac{2^N}{3^N} \qquad \square$$

Figure 3 plots the value $\frac{2^N}{3^N}\chi(m) + (1 - \frac{2^N}{3^N})$ for the size of representation $N = 6$, for sample sizes m from 0 to 64. The graph also shows the values of $\frac{1}{2}\chi(m) + \frac{1}{2}$, the learning curve for the random instance table learner L_1, and the observed classification accuracy of the simple case-based learning algorithm using preference ordering $0 \sqsupseteq 1$. The experimental method was exactly as described before, and the observation of case-based learning of monomial functions was repeated over 500 experiments. The graph shows in contrast to Figures 1 and 2 a marked difference between the learning curves for L_1 and L_2; due to the uneven probabilities of positive and negative instances, the majority instance table learner substantially outperforms the random instance table learner. However, the observed accuracies for the case-based learner appear to be essentially similar to those for L_2. Thus while the nearest neighbour generalisation strategy is clearly better than classifying unseen instances at random, so also is the strategy of classifying all new instances as negative. It is possible also that for larger training samples, the generalisation in the case-based classifier roughly approximates that of L_2. The higher proportion of negative instances of the target function mean that the case-base will be dominated by negative exemplars. In addition, the nearest neighbour decision function described above *prefers* negative classifications and it is possible that in practice, a case-based approach to learning these particular target functions will result in nearly all unseen instances being classified negatively. It is certainly the case that any false negative classifications avoided by the case-based strategy are almost completely balanced out by additional false positives, as shown by the comparison with the all-0 generalisation strategy of L_2 in Figure 3.

Rote Learning of Natural Target Functions

The previous comparisons may be challenged because they test learning performance on 'artificial' target mappings. Figures 4 and 5 show the results of applying the yardstick proposed here to case-based classification used in 'real-world' domains. The two data sets used were chosen from the UCI Machine Learning Repository [7] on the basis of having nominal feature values, no missing values in the data sets and a solution set of size greater than two. Characteristics of the two data sets are given in Table 1. The data for the 'Contact Lens' application describes potential contact lens customers in terms of four attributes. These four attributes may take three, two, two & two values respectively, giving 24 possible customer descriptions. These descriptions are classified into three classes according to the suitability of different types of contact lens. The documentation for the data set indicates that a correct description of the target function requires 9 production rules. The distribution of the classification values is highly skewed, giving a relative frequency for the majority classification of $\frac{15}{24} = 0.625$.

The data for the 'Hayes-Roth' test database gives personal data for individuals in four attributes having three, four, four & four values respectively. The

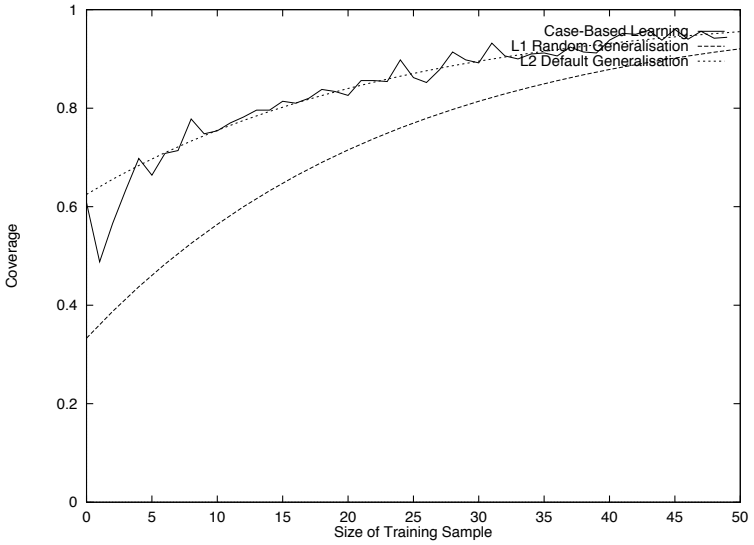

Fig. 4. Average Learning Curves For Contact Lens Data-Base

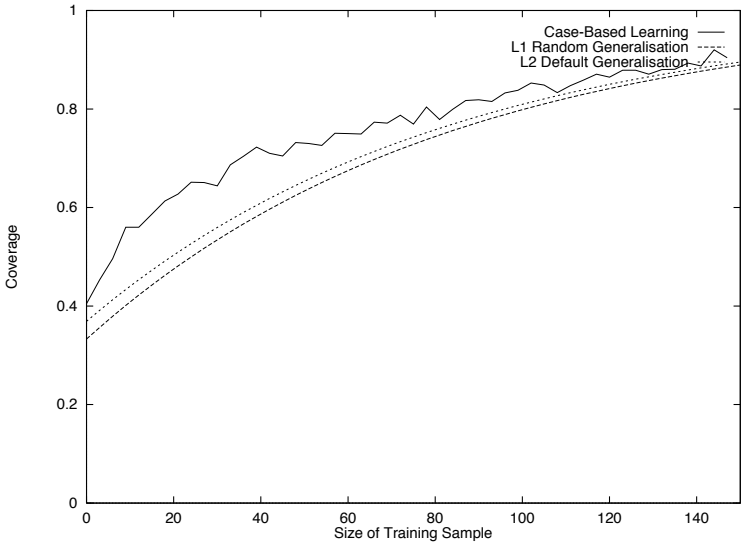

Fig. 5. Average Learning Curves For Hayes-Roth Test Data-Base

first attribute is irrelevant to the correct classification, with the target function defined in terms of the remaining attributes as follows. If a strict majority of these three attributes has value '1' then the class value is '1'. If a strict majority has value '2' then the class value is '2'. If the value '4' appears for any of the discriminating attributes then the class value '3' is assigned. If no discriminating attribute has value '3' and there are equal numbers of '1s' and '2s', the description is assigned to classes '1' and '2' with equal probability.

Characteristic	'Contact Lens' Data	'Hayes-Roth' Data
Attributes	4	4
Size of Example Space	$3 \times 2 \times 2 \times 2 = 24$	$3 \times 4 \times 4 \times 4 = 192$
Size of Data Set	24	84
No. of Classes	3	3
Freq. of Class Values	$\langle 4, 5, 15 \rangle$	$\langle 25, 28, 31 \rangle$
Freq. of Majority Class μ_0	$\frac{15}{24} = 0.625$	$\frac{31}{84} = 0.369$
Learning Curve for L_1	$\frac{2}{3}\chi(m) + \frac{1}{3}$	$\frac{2}{3}\chi(m) + \frac{1}{3}$
Learning Curve for L_2	$0.375\chi(m) + 0.625$	$0.631\chi(m) + 0.369$

Table 1. Characteristics of Data Sets used in Figures 5 and 4

The data set in the UCI repository for the 'Hayes Roth' target function described above is incomplete in that it contains instances for only some of the possible descriptions, contains duplications, and contains conflicting instances, in that there are instances of some of the 'tied' descriptions appearing with class '1' *and* '2'. The data set was processed before testing by removing all duplicate exemplars and where exemplars existed for both classifications of a particular description, one of the two was removed by the toss of a fair coin in order to ensure a functional target mapping. This process left 84 out of 192 possible descriptions represented in the data set.

In Figure 4, the learning curves for $CB1$, L_1 and L_2 are again plotted. $CB1$ is equipped with a preference ordering $3 \sqsupseteq 2 \sqsupseteq 1$, while L_2 chooses '3' as the default classification for unseen problem descriptions. The learning curves derived from Propositions 5 and 6 for L_1 and L_2 are included in Table 1. Figure 4 tells a similar story to Figure 3. The relatively skew target classification emphasises the difference between L_1 and L_2, and while the case-based classifier substantially improves on the success of the random generalisation strategy of L_1, so does the 'default value' generalisation strategy of L_2. Additionally, the learning curves for $CB1$ and L_2 fail to diverge to any significant degree. As with Figure 3, it is possible that the small example space means that in practice there will be many ties between equally similar exemplars with different classifications which must be resolved by the preference ordering. This has been chosen to prefer the majority classification, and so in practice, the nearest neighbour generalisation strategy might be very similar to that of L_2.

Finally, Figure 5 shows a result which is more encouraging for the application of case-based reasoning. Learning curves for $CB1$, L_1 and L_2 are again plotted, with $CB1$ and L_2 instantiated precisely as for Figure 4. Here, a relatively uniform distribution of class values means that the learning curves for L_1 and L_2 appear more similar. For this data set $CB1$ outperforms L_1 and L_2 despite the presence of an irrelevant attribute in the problem descriptions. Since it is known that the presence of irrelevant attributes adversely affects case-based classification, the success of $CB1$ is even more encouraging. Furthermore, there are methods for introducing appropriate weightings into the similarity measure to counteract this effect [1], and were these to be used here, even higher accuracies might be expected from the case-based classifier. This data-set would then seem a clear indication that in appropriately chosen domains, case-based classification can have significant generalisation power. Possible reasons for this will be considered in the final section.

5 Conclusions

Success in classification for a system such as a case-based classifier which is trained on a subset of the space of possible problem descriptions may be assigned to success in *recollection* for those descriptions which have been seen before, and to success in *generalisation* for previously unseen problem descriptions. The two 'rote-learning' algorithms defined in this paper, and the equations derived for their average-case learning curves, allow these two factors to be separated in the evaluation of a case-based classifier. Figures 1 and 2 give direct experimental confirmation of the results in [9] [8] that no generalisation strategy can outperform any other if all possible classifications of the example space are equally likely. Figures 3 and 4 show a formal and a natural domain in which the learning curves for $CB1$ and the Majority Instance-Table Learner L_2 appear to almost coincide, while Figure 5 demonstrates a domain in which case-based generalisation is superior to either of the two naïve generalisation strategies used here for comparison.

Earlier work [3] has shown that the 'preference ordering' plays an important part in defining the decision function for case-based classification. The results of Figures 3 and 4 here show that in addition that, especially for small, discrete example spaces, the application of the preference ordering may dominate the contribution to generalisation from the exemplars in the case-base and the similarity measure. This is presumably because ties between equally near neighbours, which must be resolved by the preference ordering, will occur more frequently in example spaces of low dimensionality. Thus the way in which ties between equally similar exemplars is resolved may be as important to classification accuracy as the way in which the similarity is calculated. However, even though the 'Hayes-Roth' database shows the same low dimensionality as the 'Lens' database, case-based classification in this domain outperforms either of the two 'yardstick' algorithms proposed here and could presumably be improved further if the similarity measure may be altered by learning. Two factors are

proposed which might explain this success. On the one hand, the more uniform distribution of solution values means that the majority classification value appears less frequently in the near neighbour sets: other classification values may then be assigned by the nearest neighbour generalisation strategy. On the other hand, it might simply be said that the decision boundaries chosen by a case-based classifier fit naturally to the 'best-of-N' target function defined for the Hayes-Roth database. Future work must identify further examples of poor *and* strong generalisation in a case-based classifier in order to better characterise the domains in which the case-based classifier will perform well.

Acknowledgements. This work was mostly carried out while the first author was supported by an EPSRC grant. Additional funding was provided by Michelin Tyres PLC. We are indebted to Messrs Murphy and Aha for the availability of the UCI Machine Learning Repository [7].

References

1. D W Aha. Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, 36:267–287, 1992.
2. A D Griffiths. *Machine Learning using a Case-Based Representation.* PhD thesis, Department of Computer Science, University of York, UK, *In Preparation.*
3. A D Griffiths and D G Bridge. Formalising the knowledge content of case memory systems. *To appear in Proceedings of the First UK Workshop on Case-Based Reasoning, Salford UK, Jan 1995.* Springer Verlag.
4. A D Griffiths and D G Bridge. Inductive bias in case-based reasoning systems. Technical Report YCS 95/259, Department of Computer Science, University of York, York YO1 5DD, UK, 1995. Available by WWW via [http://www.cs.york.ac.uk/∼tony/].
5. A D Griffiths and D G Bridge. On concept space and hypothesis space in case-based learning algorithms. In N Lavrac and S Wrobel, eds, ECML-95: Proc. 8th European Conf. on Machine Learning, 1995, LNAI Volume 914, pages 161 – 173, 1995. Springer Verlag.
6. R Kohavi. The power of decision tables. In N Lavrac and S Wrobel, eds, ECML-95: Proc. 8th European Conf. on Machine Learning, 1995, LNAI Volume 914, pages 174–189, 1995. Springer Verlag.
7. P M Murphy and D W Aha. Uci repository of machine learning databases, [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Department of Information and Computer Science, University of California, Irvine, CA, 1994.
8. C Schaffer. A conservation law for generalization performance. In *ML94: Proc. International Conference on Machine Learning, New Brunswick, New Jersey*, pages 259–265. Morgan Kaufmann, 1994.
9. D H Wolpert. On the connection between in-sample testing and generalisation error. *Complex Systems*, 6:47–94, 1992.