Optimal layered learning: a PAC approach to incremental sampling

Stephen Muggleton

Oxford University Computing Laboratory,
11 Keble Road, Oxford, OX1 3QD, United Kingdom. Email: steve@prg.oxford.ac.uk

Abstract. It is best to learn a large theory in small pieces. An approach called “layered learning” starts by learning an approximately correct theory. The errors of this approximation are then used to construct a second-order “correcting” theory, which will again be only approximately correct. The process is iterated until some desired level of overall theory accuracy is met. The main advantage of this approach is that the sizes of successive training sets (errors of the hypothesis from the last iteration) are kept low. General lower-bound PAC-learning results are used in this paper to show that optimal layered learning results in the total training set size (t) increasing linearly in the number of layers. Meanwhile the total training and test set size (m) increases exponentially and the error (ϵ) decreases exponentially. As a consequence, a model of layered learning which requires that t, rather than m, be a polynomial function of the logarithm of the concept space would make learnable many concept classes which are not learnable in Valiant’s PAC model.

1 Introduction

Since the introduction [6] of the Probably-Approximately-Correct (PAC) model of learning many theoretical results indicate that it is not possible to learn a great deal at once. Indeed it is clear that weakly constrained languages either require too many examples to describe the target concept sufficiently accurately, or require an untractably large amount of time for concept formation and testing. In this paper an incremental approach called “layered learning” is investigated and analysed using general lower-bound results for PAC learning. Layered learning proceeds by first taking a small sample from a stream of data and using it to construct an approximately correct theory. A second approximately correct theory is then constructed based on the errors of the first theory in a new sample which is a supserset of the first. Further layers of correcting theories are then added using successively larger samples until a pre-specified level of overall theory accuracy is achieved. This approach is similar to what Quinlan [5] calls “windowing”.

Clearly the minimal example requirements for layered learning will be limited by existing general lower-bound PAC results. However, it is also clear that most of these examples will simply be used for testing the present stage of the theory. Only a small number of examples, related to the present error-rate, will be used for any layer of the training set. In this paper it is shown using lower-bound PAC learning results that an optimal use of examples leads to the following.

- the total training set increases linearly with the number of layers,
- the total test set increases exponentially with the number of layers and
- the error decreases exponentially with the number of layers.

2 Lower bound PAC result

Valiant [6] introduced what has now become a widely studied stochastic model of machine learning. In this model positive and negative examples of some unknown concept, chosen from a concept class C, are presented to a learning algorithm. These examples are drawn according to a fixed but arbitrary probability distribution. From the examples drawn, the learning algorithm must, with high probability, produce a hypothesised concept that is a good approximation to the target.

Suppose the concept class C is of size 2^n. Then if a uniform prior distribution over the concept class is assumed each concept $c \in C$ can be expressed in n bits. According to [3] the following is an expression of the minimum number of examples, m, required to allow construction of an hypothesis H for which the probability of $error(H) \leq \epsilon$ is at least $1 - \delta$.

$$m = \frac{n + ln(\frac{1}{\delta})}{-ln(1-\epsilon)}$$

Existing algorithms for learning monomials, kDNF formulae, kCNF formulae and symmetric functions all use the optimal number of examples (within a constant factor).

The result can be re-expressed as follows to give the accuracy $(1-\epsilon)$ expected in terms of m, n and δ.

$$(1-\epsilon) = e^{-\frac{1}{x}}$$

where

$$x = \frac{m}{n + ln(\frac{1}{\delta})}$$

Since n is measured in bits, for a fixed value of δ, x is proportional to the number of examples required per bit of the concept learned. For fixed values of δ and n, x is simply proportional to m. Figure 1 shows the increase of accuracy with increasing x. Note the following properties of this curve.

1. $d(1-\epsilon)/dx \rightarrow 0$ as $x \rightarrow 0$.
2. $d(1-\epsilon)/dx \rightarrow 0$ as $x \rightarrow \infty$.
3. $(1-\epsilon)$ increases monotonically with x.

These observations correspond to a law of diminishing returns in machine learning. When only a small number of examples have been observed accuracy increases slowly with each example. The same occurs when a large number of

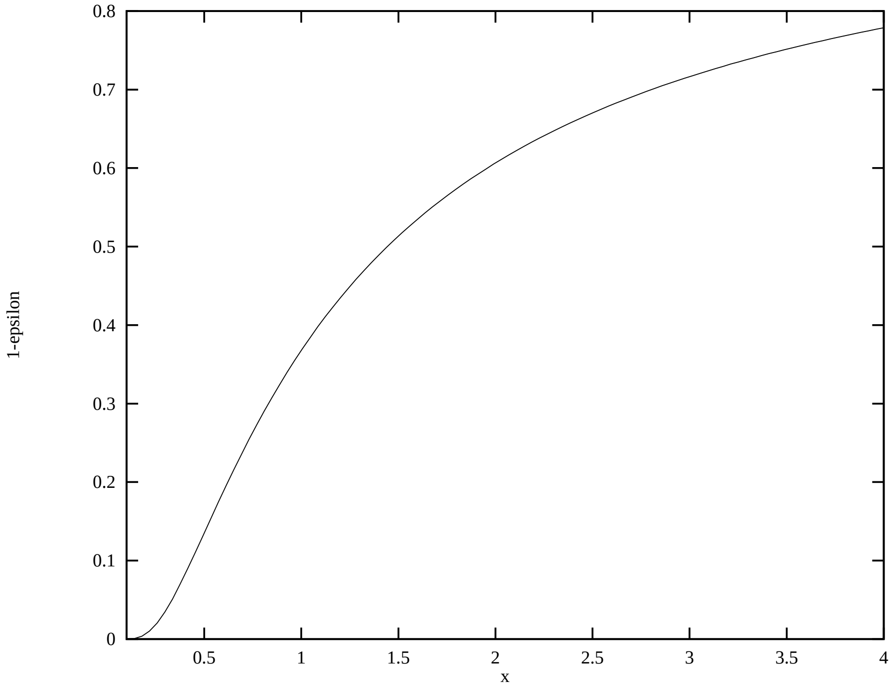

Fig. 1. Increase of $(1-\epsilon)$ with x

examples have been observed. The maximum rate of accuracy increase occurs somewhere between these two extremes. By setting the double differential to zero we find that the maximum increase of $e^{-\frac{1}{x}}$ occurs when $x = \frac{1}{2}$, i.e.

$$(1-\epsilon) = e^{-2} = 0.135$$

Note that this maximum rate of accuracy increase is independent of m, n and δ. Since the lower-bound PAC results on which this are based are also independent of both the example distribution and the concept language it can be considered that an accuracy of $(1-\epsilon) = e^{-2}$ has a fundamental significance throughout inductive learning.

3 Maximising performance increase per example

In the last section it was demonstrated that when using lower-bound PAC learning results maximum performance increase occurs when $(1-\epsilon) = e^{-2}$. However the point of maximum increase in accuracy at

$$m = \frac{n + ln(\frac{1}{\delta})}{2}$$

does not provide the optimal number of examples required for the first stage of layered learning. In order to make best use of the training set it is necessary to maximise the accuracy achieved per training example in the first stage. This can be done by solving

$$\frac{d}{dm}\frac{(1-\epsilon)}{m} = 0$$

This gives

$$m = n + ln(\frac{1}{\delta})$$

for which

$$(1-\epsilon) = e^{-1} = 0.368$$

The increase of $(1-\epsilon)/x$ with x is shown in Figure 2.

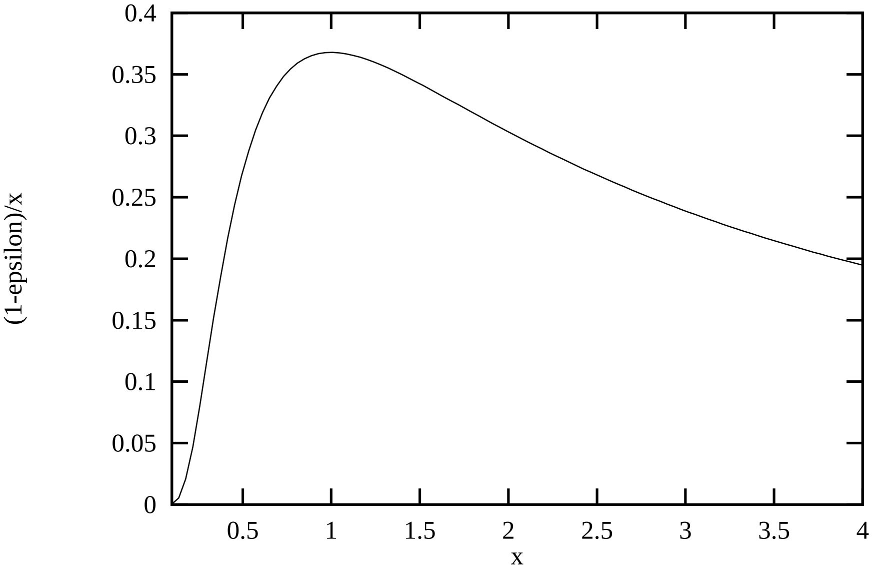

Fig. 2. Increase of $(1-\epsilon)/x$ with x

4 Two stage layered learning

In the last section we found that the optimal number of training examples for the first stage of layered learning is that which produces an accuracy of $(1-\epsilon_1) = e^{-1}$. Suppose that an accuracy of e^{-1} is produced from a training set of size m_1 in at least $(1-\delta)$ proportion of retrials. It is now possible to predict that the size, in bits, of the target concept to be learned is

$$n = m_1 - ln(\frac{1}{\delta})$$

Now suppose that the total sample for training and testing in two staged layered learning is m_2. Since only the errors in the second sample are used for training, the total training set t_2 (stage 1 + stage 2) is

$$t_2 = m_1 + \epsilon_1(m_2 - m_1)$$

Solving

$$\frac{dt_2}{dm_1} = 0$$

gives

$$m_2 = 2m_1 = 2(n + ln(\frac{1}{\delta}))$$

The error after stage two will thus be

$$(1-\epsilon_2) = e^{-\frac{1}{2}} = 0.607$$

5 Multi-stage layered learning

The analysis of the previous section can be extended to multi-stage layered learning by repeatedly partially differentiating with respect to m_i. First consider the infinite series defining the size of the training set t in terms of the partial training and test sets m_i and the corresponding error-rates ϵ_i.

$$t = m_1 + \epsilon_1(m_2 - m_1) + \ldots + \epsilon_i(m_{i+1} - m_i) + \epsilon_{i+1}(m_{i+2} - m_{i+1}) + \ldots \quad (1)$$

Setting to zero the partial differential with respect to m_{i+1} gives

$$\frac{\partial t}{\partial m_{i+1}} = \epsilon_i + m_{i+2}\frac{\partial \epsilon_{i+1}}{\partial m_{i+1}} - (m_{i+1}\frac{\partial \epsilon_{i+1}}{\partial m_{i+1}} + \epsilon_{i+1}) = 0 \quad (2)$$

To simplify the above b and x_i are defined as follows

$$b = (n + ln(\frac{1}{\delta}))$$
$$x_i = \frac{m_i}{b}$$

ϵ_i can now be approximated using series expansion as follows.

$$\begin{aligned}\epsilon_i &= 1 - e^{-\frac{b}{m_i}} \\ &= 1 - e^{-\frac{1}{x_i}} \\ &= 1 - (1 - \frac{1}{x_i} + \frac{1}{2!{x_i}^2} + \ldots) \\ &\simeq \frac{1}{x_i} \qquad \text{for large } x_i\end{aligned}$$

Simplifying equation (2) and rearranging gives

$$\begin{aligned}
m_{i+2} &= \frac{{m_{i+1}}^2}{b(1-\epsilon_{i+1})}(\epsilon_i - \epsilon_{i+1} + \frac{b}{m_{i+1}}(1-\epsilon_{i+1})) \\
&\simeq \frac{bx_{i+1}}{x_{i+1}-1}(\frac{{x_{i+1}}^2}{x_i} - 1) \\
bx_{i+2} &\simeq \frac{bx_{i+1}^2}{x_i} \qquad \text{for large } x_{i+1} \\
x_{i+2} &\simeq \frac{{x_{i+1}}^2}{x_i}
\end{aligned}$$

It can be shown that x_i is an exponential series as follows. Let $x_1 = a$ and $x_2 = ad$. Then it follows from the above that

$$\begin{aligned}
x_3 &\simeq ad^2 \\
x_4 &\simeq ad^3 \\
x_5 &\simeq ad^4 \\
&\ldots
\end{aligned}$$

and in general

$$\frac{x_{i+1}}{x_i} \simeq d$$

The general term in equation (1) can now be expressed as

$$\begin{aligned}
t_{i+1} &= \epsilon_i(m_{i+1} - m_i) \\
&= \epsilon_i b(x_{i+1} - x_i) \\
&\simeq \frac{b}{x_i}(x_{i+1} - x_i) \\
&\simeq b(d-1)
\end{aligned}$$

Thus for fixed n and δ, and large x_i the size of each successive training set remains constant at $b(d-1)$ for each value of i. The total training and test set increases exponentially in i since $x_i \simeq ad^{i-1}$. Similarly the error decreases exponentially since $\epsilon_i \simeq \frac{1}{x_i}$. Without approximations, equation (2) can be rearranged to show that

$$x_{i+2} = (1 + \frac{1}{x_{i+1}} - e^{\frac{1}{x_{i+1}} - \frac{1}{x_i}}){x_{i+1}}^2$$

This recurrence formula is used in Figure 3 to give a tabulation of t_i/b (relative training set size), x_i (relative test set size) and ϵ_i (error) for i ranging between 1 and 10. Note that although t_i/b is not a constant (it asymptotes to one), this tabulation shows that the three general trends arrived at in the analysis above hold for large x_i.

i	t_i/b	x_i	ϵ_i
1	1	1.00	.63
2	.63	2.00	.39
3	.62	3.57	.24
4	.62	6.10	.15
5	.62	10.17	.09
6	.61	16.73	.06
7	.61	27.32	.03
8	.61	44.42	.02
9	.61	72.03	.01
10	.61	116.62	.009

Fig. 3. Relative size of training set, test set and error for various values of i

6 Discussion

Layered learning offers a general approach to incremental machine induction in which successive layers of constructed knowledge decrease the error exponentially. Since successive training sets remain constant in size a learner can reach arbitrarily low values of ϵ without increasing memory requirements.

For instance, layered learning of a 10,000 bit theory requires around 6,100 training examples in each layer to produce arbitrarily low error (see Figure 3). However, the cumulative training and test set size at layer 10 would be 1,166,200 examples.

Suppose that the memory limit of the inductive learner is $l \leq 6,100$. The same general effect can be achieved as that in Figure 3 by letting $l = b(d-1)$, i.e. $d = \frac{l}{b} + 1$. In this case errors would still reduce in $O(d^{-i})$.

Layered learning provides a basis in computational complexity for what has been termed "predicate invention" within the field of Inductive Logic Programming [4]. Predicate invention involves the decomposition of predicates being learned into useful sub-concepts. Such sub-concepts can be viewed as modifiers to predicate definitions which would otherwise be both incomplete and incorrect. This approach is similar to that taken in [2, 1] and [7].

A model of layered learning which requires t, rather than m, to be a polynomial function of the number of bits in the target concept would make learnable many concept classes which are not learnable in Valiant's PAC model. The computational complexity of learning such concept classes is not explored in detail in this paper. However, it is believed that this will be a fruitful topic for future research.

Acknowledgements.

The author would like to thank Ashwin Srinivasan, Donald Michie, Tony Hoare and Brian Ripley for helpful discussions on the topics in this paper. This work was supported by the Esprit Basic Research Action ILP, project 6020.

References

1. M. Bain. Experiments in non-monotonic first-order induction. In *Proceedings of the Eighth International Machine Learning Workshop*, San Mateo, CA, 1991. Morgan-Kaufmann.
2. M. Bain and S. Muggleton. Non-monotonic learning. In D. Michie, editor, *Machine Intelligence 12*. Oxford University Press, 1991.
3. A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. In *COLT 88: Proceedings of the Conference on Learning Theory*, pages 110–120, San Mateo, CA, 1988. Morgan-Kaufmann.
4. S. Muggleton. Inductive Logic Programming. *New Generation Computing*, 8(4):295–318, 1991.
5. J.R. Quinlan. Discovering rules from large collections of examples: a case study. In D. Michie, editor, *Expert Systems in the Micro-electronic Age*, pages 168–201. Edinburgh University Press, Edinburgh, 1979.
6. L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
7. S. Wrobel. On the proper definition of minimality in specialization and theory revision. In P.Brazdil, editor, *EWSL-93*, pages 65–82, Berlin, 1993. Springer-Verlag.

This article was processed using the LaTeX macro package with LLNCS style