A Markovian Extension of Valiant's Learning Model (Extended Abstract)

David Aldous *
U.C. Berkeley

Umesh Vazirani †
U.C. Berkeley

1 Introduction

Formalizing the process of natural induction and justifying its predictive value is not only basic to the philosophy of science, but is also an essential ingredient of any formal theory of learning. A formulation of the process of induction which is intuitively compelling and reasonable from a computational point of view is arrived at by equating the process of induction with Occam Algorithms (see [BEHW]): let us suppose the phenomenon being observed (the concept being learnt) is boolean; i.e. each observed example is either a positive or a negative example of the phenomenon. An Occam Algorithm chooses from among the list of admissible hypotheses (also called the concept class) the shortest hypothesis consistent with all the examples observed so far (in general an Occam Algorithm looks for a relatively short hypothesis consistent with the data. This is formalized in [BEHW]).

It is not too hard to see that the predictive value of Occam Algorithms must, in general, rest crucially upon probabilistic assumptions about how the examples are generated. Valiant's PAC learning model [Va] gives a natural and general such set of assumptions: namely that the examples are chosen independently according to a fixed, but arbitrary, probability distribution. Under Valiant's conditions, it has been established by [Pi] and [BEHW] that a concept class is learnable if and only if it is learnable by an Occam Algorithm, thus establishing the predictive value of Occam Algorithms under this probabilistic model of the learner's environment.

In this paper, we introduce a new model of learning that expands on the Valiant model. Our basic point of departure from the Valiant model is that we place the learner in a Markovian environment. In our model, the environment of the learner is a (exponentially large) graph. The examples reside on the vertices of the graph - one example on each vertex (Valiant's model is the case where the graph is complete). The learner obtains the examples while performing a random walk on the graph. At each step, the learning algorithm guesses the classification of the example on the current vertex using its current hypothesis. If its guess is incorrect, the learning algorithm updates its current working hypothesis. The performance of the learning algorithm in a given environment is judged by the expected number of mistakes made as a function of the number of steps in the random walk. The performance of the learning algorithm is then its worst-case performance over all graphs of a given size and distribution of examples over the vertices. Our measure of performance is in the spirit of mistake bounds studied by Littlestone [Li].

We study the predictive value of Occam Algorithms under this weaker probabilistic model (and more realistic model since, for example, it expresses spatial correlations between observed examples) of the learner's environment. We reformulate this question as a question about random walks on graphs. We are able to answer affirmatively an interesting case of this question. The theorem that we prove is interesting in itself as a fact about random walks on graphs - it states that if the vertices of an n vertex graph are labeled with integers, then the expected length of the first increasing subsequence in a random walk of length t on this graph is $O(\sqrt{t}\log n)$. For the special case when the label of a vertex is its distance from the start vertex, our result is comparable to a result that can be deduced from a result of Carne [Ca] which gives a universal "large deviation" bound for the probability that a random walk

*Supported by NSF-MCS87-01426
†Supported by NSF-CCR86-58143

starting at vertex i ends up at vertex v at time t. We conjecture that the answer to the general case is also affirmative and that all Occam Algorithms are good predictors in the Markovian model. Combining this with the results of [Pi] would have the consequence that any concept class that is learnable in Valiant's model is also learnable in the more general Markovian model.

2 Problem Statement

In our Markovian model of the learner's environment, the environment is described by a Markov chain on a finite state space $[N]$ with transition matrix $P = (p_{ij})$. Let the sequence of random variables (X_t) be the Markov chain in question. Let π_i denote the stationary probability of state i. We require that the Markov chain be reversible, i.e. $\pi_i p_{ij} = \pi_j p_{ji}$. Alternately, the environment may be described as a graph on vertex set $[N]$ and with weights on the edges. The learner chooses each edge incident to its current vertex with probability proportional to its weight.

Each state of the state space houses an example (+ve or -ve) of the target concept. The learner's task is to choose a "good" hypothesis about the target concept, based upon examples (together with their classification) observed during a run of the chain of some duration. There is a difficulty in evaluating the performance of the learning algorithm. In general, the distribution from which a future example will be picked depends strongly on the current vertex - therefore the stationary distribution does not provide a good basis for judging the hypothesis. Instead, we regard learning as a never-ending process: the learning algorithm is tested at each step in its walk by having to classify the example at the current vertex. If it fails the test, it updates its working hypothesis. The performance of the learning algorithm is measured by the number of mistakes as a function of time.

We shall assume some system of representation for the concepts in the selected concept class. This induces a total ordering on the concepts (say, using the usual lexicographic ordering). Say that a learning algorithm is a strict Occam algorithm if it always selects as its working hypothesis the first concept in this ordering that is consistent with the observed data. Let us formulate the question: does the existence of an efficient strict Occam algorithm for a concept class imply that the class is learnable on a graph? Each concept in the ordering correctly classifies the examples sitting on some subset of the state space $[N]$. This yields a sequence of subsets of $[N]$: $A_1, A_2, ..., A_k, ...$ Assume that k is the index of the target concept, so that $A_k = [N]$. Now the index of the working hypothesis of the strictly Occam learning algorithm is defined by Z_t below:

Define a sequence of random variables (Z_t), where $Z_t = \min\{v : X_0, X_1, ..., X_t \in A_v\}$.

If $Z_t > Z_{t-1}$, we say that $Z_t = v$ is a record value, $X_t = j$ is a record place, t is a record time.

We are interested in the expected number of records in t steps of the Markov chain.

Examples: It is instructive to consider a few examples. First, consider a line graph - where each vertex except the two extreme ones have exactly two neighbors, one to the left and one to the right. Assume that we start at some vertex i in the middle, and that the set A_m contains the vertices at distance at most m from i. Then every time the walk encounters a previously unseen vertex, a record is created. This is not bad because on a line graph, a walk of length t is expected to stay within a distance of $\sqrt{t}$ of the starting vertex.

Another interesting example is the complete binary tree. This time if the set A_m is the set of all vertices within distance m of the root, the random walk starting at the root is expected to rush towards the leaves at a constant rate. Thus in this initial phase, a record is expected every couple of steps. However, the height of the tree is only $\log N$, and so this is an upper bound on the number of records.

Finally, in the case of the complete graph, the number of records can be shown to be $O(\log k \times \log N)$ (see section 4). By judiciously combining these graphs, all three effects can be realized simultaneously.

To summarize, these examples show that there are three different kinds of phenomena that can occur: in the complete graph, if a sequence of vertices is encountered, it is equally likely to be encountered in every other permutation. This gives a $\log k$ functional dependence for the number of records. The line graph provides a way of ensuring that the order in which new vertices are encountered is fixed in advance. Now, however, due to reversibility, the walk tends to encounter only $\sqrt{t}$ new vertices in time t. Finally, both a preset order and

rapid spreading can be achieved using fanout - as in a complete binary tree. However, the number of levels of fanout is bounded by the log of the size of the graph.

Notation: Denote by $E_i[Q]$ the expected value of Q in a random walk starting in state i, and by $P_i[Q]$ the probability that Q occurs in a random walk starting in state i.

Conjecture: E_i[Number of records in t steps] = $O(\sqrt{t} \log k \log \frac{1}{\pi_i})$.

In terms of the learning problem, k is a measure of the complexity of the concept being learned, so a dependence of the number of mistakes on $\log k$ is inevitable; similarly, $\frac{1}{\pi_i}$ is a measure of the size of the graph in the uniform case, and a dependence on its log is inevitable. In the next section, we shall prove that in the case when the A_is are increasing (i.e. $A_1 \subset A_2 \subset ... \subset A_k$), the conjecture is true.

Random walks on graphs find applications in many areas. See [Al] for a recent survey.

3 Main Theorem

Theorem: E_i[Number of records in t steps] $\leq e^2\sqrt{t}(1 + \log \frac{1}{\pi_i})$, if the $A_m's$ are increasing.

To avoid boundary effects, we first modify the problem as follows: instead of running the Markov chain for t steps, we shall run the chain for τ steps, where τ is a random variable with $E[\tau] = t$. We do this by killing the chain after each step with probability α. Then $P[\tau > n] = (1-\alpha)^n$ and $E[\tau] = \frac{1}{\alpha}$.

Let r(i) = E_i[Number of records before τ].

We shall prove:

Lemma 1: r(i) $\leq e\alpha^{-\frac{1}{2}}(1 + \log \frac{1}{\pi_i})$.

The theorem follows from Lemma 1 by setting $\alpha = \frac{1}{t}$, since

E_i[Number of records before t] $\leq \frac{r(i)}{P[\tau > t]}$
$\leq e^2\sqrt{t}(1 + \log \frac{1}{\pi_i})$.

To prove Lemma 1, we first prove a bound on the expected value of r(i) when the state i is picked at random from the state space:

Lemma 2: $\sum \pi_i r(i) \leq \alpha^{-\frac{1}{2}}$

Proof of Lemma 1: Fix $\beta > 0$ and let $B = \{i : r(i) > \beta\}$.

Clearly $r(i) \leq \beta$+ number of records in B.

Now, watching the Markov chain only on the set B gives another Markov chain (Y_t). Every record in B in the X-chain at time t corresponds to a record in the Y-chain at some time $\leq$ t. Denoting by $\hat{r}(i)$ the expected number of records before τ, stating in state i in the Y-chain, we have:

$r(i) \leq \hat{r}(i) + \beta$.

Moreover (Y_t) is a reversible Markov chain, with stationary probability $\hat{\pi}_i = \frac{\pi_i}{\pi(B)}$. Applying Lemma 2 to the Y-chain yields:

$\sum \hat{\pi}_i \hat{r}(i) \leq \alpha^{-\frac{1}{2}}$.

Thus $\hat{\pi}\{i : \hat{r}(i) > e\alpha^{-\frac{1}{2}}\} < e^{-1}$.

Thus $\hat{\pi}\{i : r(i) > e\alpha^{-\frac{1}{2}} + \beta\} < e^{-1}$.

Thus $\pi\{i : r(i) > e\alpha^{-\frac{1}{2}} + \beta\} < e^{-1}\pi(B)$.

Thus $\pi\{i : r(i) > e\alpha^{-\frac{1}{2}} + \beta\} < e^{-1}\pi\{i : r(i) > \beta\}$.

By induction on q it follows that

$\pi\{i : r(i) > qe\alpha^{-\frac{1}{2}}\} < e^{-q}$.

Choosing q to be $1 + \log \frac{1}{\pi_i}$, we see that $r(i) \leq qe\alpha^{-\frac{1}{2}}$.

To prove Lemma 2, we first need to introduce some notation, and prove a path-reversal lemma:

Definitions: Let $T_v^* = \min\{t : X_t \notin A_v\}$.

Let $\rho_i(j,v) = P_i$[v is a record value and j the corresponding record place at some time before τ]

Let $\tilde{\rho}_i(j,v) = E_i$[number of visits to j before T_v^* and before τ]

Lemma 3 (Path-Reversal Lemma): E_i[number of visits to k before τ] = $\sum_{(j,v)} \rho_i(j,v)\tilde{\rho}_k(j,v)\frac{\pi_k}{\pi_j}$

Proof: Let $R^0, R^1, ..., R^u, ...$ be the record times

Then E_i[number of visits to k before τ]

= $\sum_{u=0 to \infty}$ Number of visits to k in $[R^u, R^{u+1}]$

and before τ.
$= \sum_{u=0 to \infty} \sum_{(j,v)} P_i[X_{R^u} = j \text{ and } Z_{R^u} = v]$
E_j[number of visits to k before T_v^* and before τ].

We use the definition of τ here.

$= \sum_{(j,v)} \rho_i(j,v) E_j$ [number of visits to k before T_v^* and before τ].

But E_j[number of visits to k before T_v^* and before τ]

$= \sum_n P_j[X_n = k \text{ and } X_0, X_1, ..., X_n \in A_v \text{ and } n < \tau]$

By path reversal, this is equal to:

$= \sum_n P_k[X_n = j \text{ and } X_0, X_1, ..., X_n \in A_v$
and $n < \tau] \frac{\pi_k}{\pi_j}$

$= E_k$[number of visits to j before T_v^* and $\tau] \frac{\pi_k}{\pi_j}$

$= \tilde{\rho}_k(j,v) \frac{\pi_k}{\pi_j}$.

Proof of Lemma 2: Let f(j,v) $= \sum \pi_i \rho_i(j,v)$

i.e. f(j,v) is the probability that j is a record place and v the corresponding record value at some time $< \tau$, when the Markov chain is started in a random state at t = 0.

Then $f^2(j,v) = \sum_i \pi_i \sum_k \pi_k \rho_i(j,v) \rho_k(j,v)$
$\leq \sum_i \pi_i \sum_k \pi_k \rho_i(j,v) \tilde{\rho}_k(j,v)$

Therefore
$\sum_{(j,v)} \frac{f^2(j,v)}{\pi_j} \leq \sum_i \pi_i \sum_k \sum_{(j,v)} \rho_i(j,v) \tilde{\rho}_k(j,v) \frac{\pi_k}{\pi_j}$

By the Path-Reversal Lemma, this is equal to:
$= \sum_i \pi_i \sum_k E_i$[number of visits to k before τ]

$= \sum_i \pi_i E[\tau]$

$= \alpha^{-1}$.

Now, $\sum \pi_i r(i) = \sum_{(j,v)} f(j,v)$

By Cauchy-Schwartz this is less than or equal to:

$\sqrt{\sum_{(j,v)} \frac{f^2(j,v)}{\pi_j} \sum_{(j,v)} \pi_j}$

Summing only over the allowable (j,v), we need consider only one value of v for each j in the case where the A_vs are increasing.

So $\sum_{(j,v)} \pi_j = 1$.

Thus $\sum \pi_i r(i) \leq \sqrt{\alpha^{-1} \times 1} = \alpha^{-\frac{1}{2}}$.

Comment: The only use of the condition that A_is are increasing was made in restricting the summation above to allowable (j,v). In the general case, we lose control here; it is worth mentioning that in the case where the A_is are only approximately increasing, and there is a suitable bound on the number of allowable (j,v) pairs that each state j may participate in, we get a bound which is larger by a factor of the square root of the multiplicity than $\alpha^{-\frac{1}{2}}$.

4 The Complete Graph Case

Theorem: For the complete graph on vertex set $[N]$, E_i[number of records in time t] $= O(\log N \times \log k)$ where k is the index such that $A_k = [N]$.

Proof Sketch: Let R_j = number of records from the time that all sets A_v of size less than $N(1 - \frac{1}{2^j})$ are eliminated from contention to the time when all sets of size less than $N(1 - \frac{1}{2^{j+1}})$ are eliminated.

Now, we claim that $E[R_j]$ is bounded by $\log k$ for each j. Roughly, this is because it takes $2^j \log k$ steps to eliminate all sets of size up to $N(1 - \frac{1}{2^{j+1}})$. On the other hand only 1 in every 2^j of these steps is expected to create a record, since the current set A_{Z_t} is of size at least $N(1 - \frac{1}{2^j})$.

Now the theorem follows since the expected number of records is the summation of the expected values of the R_js, and we need only consider j's less than $\log N$.

5 Discussion

Littlestone and Warmuth [LW] introduce a learning algorithm which is not an Occam Algorithm, but takes the majority prediction among all consistent hypotheses. This algorithm is guaranteed to make polynomially many mistakes even for a worst-case draw of examples. The drawback is that taking a majority vote is as hard as approximate counting which is usually computationally intractible even when minimization (which is required to run an Occam Algorithm) is computationally

feasible.

Besides giving a more accurate model of the world, the graph model makes it possible to focus on finer questions about learnability: it is possible to model the spatial locality of the data by insisting that the graph satisfy some degree constraints or topological properties. It is worth investigating whether these added conditions explain the observed discrepancy between sample size bound predictions using Valiant's model versus empirical learning algorithms. Another interesting issue that can be expressed in our model is that of learning algorithms with limited memory. How should Occam Algorithms be modified when there is not enough memory to store all the examples encountered in the past?

6 Acknowledgements

We wish to thank Avrim Blum and Merrick Furst for several stimulating discussions.

7 References

[Al] D. Aldous, "Applications of Random Walks on Finite Graphs," to appear.

[BEHW] A. Blummer, A. Ehrenfeucht, D. Haussler, M. Warmuth, "Occam's Razor" Information Processing Letters, 24, 1987, pp. 377-380.

[Ca] T. K. Carne, "A transmutation formula for Markov chains," Bull. Sci. Math. (2), 109:399-405, 1985.

[Li] N. Littlestone, "Learning quickly when irrelevent attributes abound: a new linear-threshold algorithm", Machine learning, 2(4), pp. 285-318, 1987.

[LW] N. Littlestone, M. Warmuth, "The Weighted Majority Algorithm", FOCS '89.

[Pi] L. Pitt, "On the necessity of Occam Algorithms", STOC '90.

[Ri] J. Rissanen, "Stochastic Complexity and Modeling", The annals of Statistics, 14(3):1080-1100, 1986.

[Va] L. Valiant "A Theory of the Learnable" Communications of the ACM, Nov 1984, vol 27, No. 11.