Hidden Markov Chains: Convergence Rates and the Complexity of Inference

by

David Gillman

B.A., Mathematics
Yale University, 1986

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

September 1993

Signature of Author _____
Department of Mathematics
May 10, 1993

Certified by _____
Michael Sipser
Professor of Mathematics
Thesis Supervisor

Accepted by _____
David Vogan
Chairman, Departmental Graduate Committee
Department of Mathematics

Hidden Markov Chains: Convergence Rates and the Complexity of Inference

by

David Gillman

Submitted to the Department of Mathematics on July 15, 1993,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Mathematics

ABSTRACT

We consider a black box containing an ergodic Markov chain that has finitely many states and a labelling of the states with 0's and 1's. As the hidden Markov chain moves probabilistically through a trajectory, the black box emits a 0 or a 1 at each time step according to the label of the state just entered.

We first consider the case in which the Markov chain is a random walk on a weighted graph G. Let A be the set of vertices labelled 1; we show that the sample average of visits A converges to the stationary probability $\pi(A)$ with error probability exponentially small in the length of the random walk and the square of the size of the deviation from $\pi(A)$. The exponential bound is in terms of the expansion of G and improves previous results of Aldous, Lovász and Simonovits, and Ajtai, Komlós, and Szemerédi.

We show that the method of taking the sample average from a single trajectory is a more efficient estimator of $\pi(A)$ than the standard method of generating independent sample points from several trajectories. This more efficient sampling method is used, together with other statistical innovations, to improve the running times of the algorithms of Jerrum and Sinclair for approximating the number of perfect matchings in a wide class of graphs and for approximating the value of the partition function of a ferromagnetic Ising system. We also derive a fast estimate of the entropy of a random walk on an unweighted graph, considered as an information source.

We next consider general ergodic Markov chains. We consider the exact inference problem: construct an entire mechanism that will emit 0's and 1's and which is equivalent to the black box in the sense of having identical long-term behavior. We introduce the related discrimination problem, which we prove is no harder than inference. We show that for restricted classes of hidden Markov chains any algorithm for discrimination (and hence also for inference) must make exponentially many oracle calls. Our method is information-theoretic and does not depend on separation assumptions for any complexity classes. On the positive side we give a randomized algorithm for discrimination for a nontrivial special case of hidden Markov chains. We make partial progress toward a randomized discrimination algorithm based on the entropy estimate which follows from our exponential bound on the deviation probability of the number of visits to a set of states.

Finally, we raise several open questions and suggest possible connections between exact inference and the problem of inference from a finite sample of empirical data.

Thesis Supervisor: Michael Sipser
Title: Professor of Mathematics

Acknowledgements

As an advisor and collaborator Mike Sipser has set a lasting example for me in the way he does research and in his perspectives on mathematics. He has shown unflagging enthusiasm for my work and has at the same time been a relentlessly honest critic of my methodology and my writing. I am very grateful for all he has done.

I thank my thesis committee members, Mauricio Karchmer and Ron Rivest, for their individual contributions to my mathematical development and for commenting on the first draft of my thesis.

I am grateful to Gilbert Strang, Persi Diaconis, László Lovász, Miklos Simonovits, Alistair Sinclair, Nabil Kahale, Peter Winkler, and Greg Sorkin, for helpful comments on my thesis work. I also thank my first advisor Richard Dudley, who gave his time and ideas generously, Robin Pemantle and Kathryn Hess, who coached me for my oral examination, and David Jerison, who talked about a PhD as though it were something attainable.

MIT provided me with financial support, huge resources, and creature comforts. Phyllis Ruby, Maureen Lynch, and Be Hubbard made bureaucracy seem not to exist here.

I also received financial support through grants from the NSF (9212184 CCR), AFOSR (F49620-92-J-0215), and DARPA (N00014-92-J-1799).

Special thanks to Dan, Dimitri, and Mike H.

Contents

Chapter 1

Introduction

1.1 Overview

Consider a black box containing an ergodic Markov chain that has finitely many states and a labelling of the states with 0's and 1's. As the Markov chain moves probabilistically through a trajectory, the black box emits a 0 or a 1 at each time step according to the label of the state just entered.

Suppose an observer wants to estimate the long-term probability that a given label emitted by the black box will be 1. We investigate how efficient an estimate it is to take the average of the first n labels emitted. The efficiency depends on the structure of the hidden Markov chain. In the case of a random walk on a complete graph with self-loops, the black box is emitting each label independently at random with a fixed probability p (p is the fraction of states labelled 1 in this case). Chernoff's bound [Ch] tells us that our averaging estimate will be very close to p: it will deviate from p by any given amount γ with probability no more than $e^{-\gamma^2 pn/2}$.

We will show that for a random walk on an arbitrary graph there is a similar

exponential decay in the error probability. The complete graph is an extremely good expander and it is a regular graph. The exponent in our bound will look like the one in Chernoff's bound with extra factors to measure expansion and non-regularity.

Suppose the observer now wants to construct an entire mechanism that will emit 0's and 1's and approximate the behavior of the black box, right down to predicting the long-term frequency of any finite sequence of 0's and 1's. Clearly this is a harder problem, based only on the same sample of n labels. It is also a well-studied problem, in various interrelated forms, with applications to speech recognition [LRS], data compression [LZ], and approximation algorithms [Ald87]. We call it the problem of inference of a hidden Markov chain from empirical data.

Let us now make the problem even harder. We are not only interested in how large the sample length n must be, but also in the computational resources necessary for the observer to construct her hypothesis. Previous work on related problems has focused on heuristic techniques with limited analytic underpinnings [LRS] and asymptotically efficient methods using classical information theory which do not exploit the structure of the hidden Markov chain [LZ].

We study an idealized version of the problem and gain insights in that setting which we believe are worth bringing to bear on the empirical problem.

In the idealized version the observer has access to an oracle which can report the precise long-term frequency of any finite string of 0's and 1's. Now the observer's problem is to construct a mechanism which will mimic the black box with perfect accuracy on every string (perfect accuracy is of course impossible using empirical data). We call this the problem of inference of a hidden Markov chain from oracle queries. Gilbert [Gi] showed that if a black box contains a Markov chain with m

states, then the behavior of the black box is completely determined by some set of at most m^2 finite strings of 0's and 1's. Furthermore, the longest string in the set has length at most $2m - 1$. Gilbert's proof introduced powerful insights from linear algebra, but it did not suggest any method for finding such a set of strings other than exhaustive search.

There is a reason for this. It turns out that if a randomized algorithm with oracle queries can solve inference, then its running time must be exponential in the number of states of the hidden Markov chain. This intractibility result remains true even when the hidden Markov chain is restricted to having uniform stationary distribution, a fairly strong regularity property. It is a fairly surprising result, because there is a natural NP algorithm with oracle queries which follows from Gilbert's result.

We prove this theorem by reduction to the discrimination problem, which we introduce here and which we believe is of independent interest. In it the observer has two black boxes. She may open up one of them to inspect the hidden Markov chain, but the other is accessible only through an oracle. The problem is to determine whether they have the same behavior (i.e., the long-term probability of every finite string of 0's and 1's is the same). We prove that discrimination is no harder than inference, and we also show that discrimination is intractible. We construct a collection of black boxes which are behaviorally distinct from one another but information-theoretically indistinguishable in polynomial time.

On the positive side, we give a randomized algorithm to solve the discrimination problem on a further restriction of the class of hidden Markov chains on which discrimination is hard. We use the oracle to simulate the black box and discriminate it from an independent sequence of 0's and 1's based on the probability of a random string. The proof adapts a well-known result of Aleliunas *et al.* [AK*] on cover times

of random walks to a special case of directed graphs.

We also initiate a more general investigation into the discrimination problem on special classes of hidden Markov chains whose structure determines their convergence to long-term behavior. The Chernoff bound implies that the finite-alphabet Markov source defined by a random walk on an expander graph converges quickly to its asymptotic rate of entropy. This represents a first step toward distinguishing a hidden *time-reversible* Markov chain from an independent sequence based on the probability of a random string.

The natural question also arises whether an observer who is allowed to open both black boxes for inspection could determine whether they had the same behavior. This is the equivalence problem. By refining the algebraic techniques of [Gi], we show that it can be decided efficiently .

Our methods depend heavily on the notion of a pseudo hidden Markov chain, and we argue that this model should be imported into the empirical inference problem and explored there. Gilbert [Gi] discovered that from the point of view of linear algebra and even the behavior of black boxes, it makes no difference if some of the transitions in the hidden "Markov chain" are negative (making it a "pseudo" hidden Markov chain). The pseudo hidden Markov chain is a valid and efficient mechanism for generating strings of 0's and 1's. Thus it allows the observer freer use of linear algebra and a wider class of hypotheses, unencumbered by the requirement that all matrices are nonnegative.

Our work is the first attempt to connect two strains that exist in the literature, empirical problems and oracle problems. We introduce the idea that by looking at a class of hidden Markov chains with nice structure, one can control convergence to

limiting behavior and possibly ensure efficient inference from empirical data. We also import the insight from oracle problems that the long-term behavior of a black box is determined by the long-term probabilities of small set of short strings. Finally, we introduce the wider class of pseudo hidden Markov chains as a valid hypothesis for an observer of empirical data emitted by a black box. We identify a number of open questions on convergence rates of hidden Markov chains, on the properties of pseudo hidden Markov chains, and on inference from empirical data.

1.2 The Chernoff Bound

We first consider the case of a random walk on a weighted graph G with stationary distribution π. This is equivalent to a time-reversible Markov chain. Let A denote the set of states labelled 1. Then $\pi(A)$ is the long-term probability of the string 1. In Theorem 1 we show that the fraction of 1's in a short sequence of labels gives a good estimate of $\pi(A)$ with high probability. If the stationary distribution is nearly uniform, a sample of size logarithmic in the number of states of the Markov chain suffices, when the Markov chain starts in stationarity.

Using Theorem 1 we derive an efficient entropy estimate for a special case of random walks, and later, in Section 3.6, we give a discrimination algorithm based on this entropy estimate. For these results we restrict our attention to the special case of a *unifilar* random walk, in which no state has positive transition probability to two states with the same label. We also require the graph to be unweighted. We allow the states to be labelled by any finite alphabet, because the results go through in this case and because with a 0-1 labelling G is forced to be a cycle or a chain. One can view the random walk on a graph with labelled vertices as an information source,

and such a source has a well-defined notion of entropy. There is also a well-known notion of the *empirical entropy* of a sequence of n labels generated by such a source [Zi, Gu]: -1/n times the logarithm of the probability of the sequence of labels. With our Chernoff-type bound we show that the empirical entropy of a finite output string gives a good estimate of the actual entropy of this information source.

The discrimination algorithm which follows from the entropy estimate is able to distinguish a black box containing a hidden unifilar random walk on an unweighted expander graph from a sequence of independent labels. It distinguishes on the basis of the probability of a random finite string generated by the oracle. An extension of the entropy estimate to the non-unifilar or non-reversible case would imply the existence of a discrimination algorithm for a larger class of hidden Markov chains. The idea of using a Chernoff-type bound to choose between two hypotheses has been used by Koopmans [Koo] in a different setting. Koopmans considers Markov chains on the real line whose transition densities are positive with respect to Lebesgue measure.

We give important applications of the Chernoff bound to approximation algorithms. We show that the method of taking the sample average from a single trajectory is a more efficient estimator of $\pi(A)$ than the standard method used in approximation algorithms of generating independent sample points from several trajectories. We use this more efficient sampling method, together with other statistical innovations, to improve the running times of the algorithms of Jerrum and Sinclair for approximating the number of perfect matchings in a wide class of graphs and for approximating the value of the partition function of a ferromagnetic Ising system [JS89, JS91].

About the Bound

The classical weak law of large numbers says that the sample average of visits to A along a finite trajectory of the random walk on G will converge in probability to the stationary probability $\pi(A)$ [Fe]. Formally, this means the following: let t_n be the number of times in an n-step trajectory that the random walk enters a vertex in A. Then for arbitrarily small numbers $\gamma > 0$ and $\delta > 0$, there exists a positive integer n such that $\Pr[|\frac{t_n}{n} - \pi(A)| > \gamma] < \delta$. γ is called the *size of deviation* and δ is called the *error probability.*

Numerous authors have considered the *rate* of convergence, by bounding the error probability as a function of the size of deviation and the number of steps n. Aldous [Ald87] has bounded the rate of convergence in the L^2 norm. He shows that δ is $O(\frac{1}{\gamma^2 n})$. Lovász and Simonovits [LS] have extended Aldous' result to time-reversible Markov chains on a continuous state space. Ajtai, Komlós, and Szemerédi [AKS] have shown that the sample average will be within a constant amount of deviation from $\pi(A)$, with error probability exponentially small in the number of steps. See also [CW, IZ].

We improve these results for the finite state space case. We show that the error probability decays exponentially in the number of steps of the random walk and the square of the size of the deviation from $\pi(A)$. As with the previous results, the bound is in terms of a spectral property of G which measures its expansion. Unlike the previous results, the bound also depends on how nearly uniform the stationary distribution π is.

The corresponding bound for sums of independent random variables x_i under general assumptions is due to Cramér [Cr] and Chernoff [Ch] and is a standard tool

of statistical inference. Let $S_n = \sum_{i=1}^{n} x_i$. The crucial fact shown by these authors is that $\Pr(S_n \geq a)$ behaves roughly like m_a^n where m_a is the minimum value assumed by the moment generating function of $x_0 - a$. m_a is easy to calculate for common distributions.

Höglund [Hö] establishes the analagous fact about the behavior of $\Pr(S_n \geq a)$ for Markov chains and writes m_a in terms of the largest eigenvalue of a perturbation of the transition matrix for the Markov chain. It is generally difficult to estimate m_a. We use matrix perturbation theory to develop a useful estimate of m_a for rapidly mixing reversible Markov chains. The estimate is in terms of the gap between the largest and second largest eigenvalues of the transition matrix (this is the aforementioned spectral property which measures the expansion of G) and in terms of the nearness to uniformity of the stationary distribution of the Markov chain.

Approximation Algorithms

Estimating $\pi(A)$ is a fundamental problem for approximation algorithms in which A and G are exponentially large combinatorial sets such as sets of matchings of a graph (see, for example, [JS89]). The basic idea is to generate a number of random sample points in G and compute the fraction that are in A. Generally speaking, when G is an expander the random walk on G is rapidly mixing. The standard procedure is to use the rapid mixing property of the random walk on G to generate a single nearly random sample point from π. This process is repeated to generate the number of independent sample points Chernoff's bound requires [JS89, LS].

An alternative sampling procedure originally proposed by Aldous [Ald87] is first to generate a nearly random starting point and then to sample every point along a

single trajectory of the random walk. We combine the analysis of [Ald87] with an important statistical technique of Jerrum, Valiant, and Vazirani [JVV] and with our own Chernoff-type bound to show that the alternative sampling procedure is more efficient than the standard one. We use this more efficient sampling method together with other statistical innovations to improve the running times of the algorithms of Jerrum and Sinclair for approximating the number of perfect matchings in a wide class of graphs and for approximating the value of the partition function of a ferromagnetic Ising system (see [JS89, JS91]).

1.3 Hidden Markov Chains

We formalize a hidden Markov chain (hmc) as a stationary ergodic Markov chain together with a 0-1 labelling of the states. (Our results generalize to a finite alphabet of labels.) The Markov chain defines a probability distribution on infinite random sequences of 0's and 1's. We formalize the black box behavior of an hmc as the function which assigns a probability to each finite string. We define a pseudo hidden Markov chain (pseudo-hmc) to be a "stochastic" transition matrix (the row sums equal 1, but the entries are possibly negative) together with a 0-1 labelling of the states. We show that a pseudo-hmc formally assigns possibly negative "probabilities" to finite strings of 0's and 1's, and we extend the definition of a black box to this case. We define the *probability oracle*, which can report the value of the black box function on any finite string of 0's and 1's.

Equivalence

Two pseudo-hmc's are *equivalent* when they have the same black box behavior. We give a characterization, due to Gilbert [Gi], of the smallest pseudo-hmc equivalent to a given m-state pseudo-hmc M. A state of the minimal pseudo-hmc corresponds to the distribution reached by M after generating a string w; the behavior of the minimal pseudo-hmc starting in this state corresponds to the behavior of M conditioned on having generated w. It is thus possible to characterize the black box behavior of M by the probabilities of at most m^2 strings of length at most $2m - 1$. We also give a proof of the result, due to Gilbert [Gi] and Paz [P], that two pseudo-hmc's defined on the same set of labelled states are equivalent whenever their transition matrices are similar and the similarity transformation respects labellings in a natural sense.

We develop a polynomial-time construction of the minimal pseudo-hmc equivalent to a given M, using oracle calls. In [Gi] and [P] there is an implicit brute-force search for the strings that characterize M. We replace this with an efficient breadth-first search on the infinite binary tree, with pruning. We develop two polynomial-time algorithms in Section 3.4 to decide the equivalence of two pseudo-hmc's: one uses the minimal pseudo-hmc as a fingerprint, and the other depends on the construction of a "union" pseudo-hmc which runs the two simultaneously. After discovering these results we found that the earlier results of Tzeng [Tz92a] used the same searching and pruning method to solve the minimization and equivalence problems for probabilistic automata. Our equivalence problem is reducible to that one (we elaborate on the correspondence between the two models below). To our knowledge the union pseudo-hmc is original.

We solve a hypothesis testing problem for hmc's [Gu] by reducing it to equivalence.

The problem is to determine which of two hidden Markov chains accounts for the behavior of a black box presented by an oracle. We give an application of this, Theorem 9 to resolving the ambiguity of a Huffman code which was encoded by one of two given Huffman trees. See Gillman *et. al.* [GMR] for background on ambiguity of Huffman codes.

Hardness of Inference

We show that relative to the probability oracle the discrimination problem is reducible in polynomial time to the inference problem. This follows easily from the existence of a polynomial-time algorithm to decide equivalence. We consider the restricted class of *unifilar* hmc's in which each transition probability is 0, p, $1-p$, or 1. (A unifilar hidden Markov chain is one in which each state has nonzero transition probability to at most one state labelled 0 and one state labelled 1.) We show that even in this class there is no randomized polynomial-time algorithm to solve discrimination, and therefore the same is true for inference. This is the content of Theorem 10. For each string w of length m we construct an hmc on $2m+2$ states which behaves like a sequence of fair coin flips on every string not containing w. Each such hmc contains a certain signature state with exponentially small stationary probability and a "long memory". Tzeng [Tz92b] used a very similar construction to show that inference is intractible for a unifilar Markov chain model which does not contain notions of stationarity or ergodicity.

We show in Theorem 11 that discrimination is still hard when every state is visited frequently. We consider the class of hmc's with uniform stationary distribution. For each string w of length m we construct an hmc in this class which is "equivalent

on strings not containing w" to the corresponding hmc in the above class of unifilar hmc's. We first find an equivalent pseudo-hmc with uniform stationary distribution using a similarity transformation. Then we average the pseudo-hmc with a random walk on a complete graph and prove that the averaged hmc has the right properties.

Positive Results

Discrimination is easier on the intersection of the above two classes, when $p = 1/2$. That is, we consider the class of unifilar hmc's with uniform stationary distribution in which each transition probability is 0, 1/2, or 1. We give a randomized algorithm to distinguish a member of this class from a sequence of fair coin flips. This shows that there is a nontrivial limit to proving hardness results for discrimination. The idea behind our algorithm is that a random long string will have probability zero unless the hmc actually is equivalent to a sequence of fair coin flips. We prove in Theorem 12 that this is indeed the case, by adapting a result of Aleliunas *et al.* [AK*] on random walks.

We give some evidence that randomness is needed, by showing that hmc's from this class can appear equivalent except on long strings. Specifically, there is a family of exponentially many $O(k)$-state hmc's which behave identically on all strings of length less than k.

We also obtain some progress on discrimination of hidden Markov chains based on the entropy estimation that follows from our Chernoff bound, as mentioned previously.

Our randomized algorithms use the oracle to simulate the black box. We observe that in order to generate a random sequence it is enough to have access to the probability of an arbitrary string.

Further Work

It remains an open question whether inference is decidable efficiently with a deterministic or randomized algorithm with oracle queries on an interesting subclass of hmc's. We further propose the *approximate* inference problem with an oracle, which to our knowledge has not been investigated at all. This problem is to infer a pseudo-hmc whose behavior approximates a hidden Markov chain, from oracle queries. Approximate inference, as it turns out, is no harder than either empirical inference or the exact oracle inference problem to which our hardness results apply.

We present for further study a variant of inference in the teacher-learner setting of Angluin [Ang87]. We define a natural teacher oracle, which is able to answer a discrimination question. Presented by the learner with a hypothesis to account for black box behavior, the oracle answers either Yes (the hypothesis is correct) or No with a counterexample string on which the hypothesis differs from the black box. In other words, the teacher is an oracle for discrimination.

We propose a consistency problem: Given a set of strings and their long-term probabilities, determine whether an m-state pseudo-hmc exists which assigns the correct probabilities to those strings. We also propose the restricted version in which the set of strings must be closed under substrings. The intuition from empirical samples suggests that if the probability of a string is given then the probability of all substrings should be given as well. This is because if a sample output from a black box contains many instances of a string w, enough for a reliable estimate of the long-term probability of w, then it also contains many instances of each substring of w. Furthermore, the long-term probability of each substring of w is clearly at least as great as the long-term probability of w itself, and one would expect a finite

sample output to give the most reliable estimates for strings of high probability. The consistency problem differs from the inference problem for oracle algorithms in two ways. The algorithm is passive in that it receives a fixed set of strings that it cannot choose, a condition shared by the empirical inference problem. Also, there is not necessarily a unique correct answer up to equivalence.

Our discrimination results are a first step toward reexamining the discrimination problems considered by Ziv and others [Zi, WZ] in a more concrete setting. In [Zi], Ziv considers discrimination of probability distributions on infinite sequences of letters of a finite alphabet, using empirical data. He gives a "universal discriminant", a function of the known distribution and the training sequences which distinguishes the known distribution from any unknown one. This discriminant function is defined in terms of a universal code for a class of conditional probability distributions satisfying a fading memory condition. As Ziv points out (see discussion following [Zi, Theorem 2]), in the i.i.d. case it is possible to dispense with the universal code. Our result shows that it is possible to dispense with a universal code for a special case of discrimination of Markov chains. It would be interesting to extend our result to as large a class of Markov chains as possible.

Finally, it would be interesting to determine the power of oracles for the discrimination problem. Since an oracle can simulate a black box and generate sample strings, our model of a learning algorithm with oracle calls is *a priori* at least as powerful as the one in which an algorithm has access only to empirical samples. It would be nice to establish a case where it is provably more powerful.

1.4 Empirical Problems

The hidden Markov chains are an important subclass of hidden Markov models (hmm's), where there is an probabilistic output function attached to each state. The problem of statistically inferring these machines from examples of their output has received considerable attention in the literature in the past twenty-five years because of various practical applications, primarily in speech recognition, and also in data compression and cell biology. In many cases it is sufficient to consider hmc's because when the output functions have a finite range, an hmm may be converted to an hmc by increasing the number of states.

Previous work has focused on finding a model of fixed size that maximizes the likelihood of having generated a given set of training sequences. Baum and Petrie [BP] have shown that the maximum-likelihood hypothesis is "consistent" in the sense that its behavior converges to that of the actual unknown model as the length of a single training sequence goes to infinity. Nevertheless, there is no known efficient way to compute the maximum-likelihood hypothesis. Baum and Welch [LRS] give an iterative procedure for converging to a Markov chain which is a local maximum of the likelihood function. Their algorithm is practical and is widely used in speech recognition. The one theoretical drawback is that the method may not converge to a global maximum of the likelihood function.

Abe and Warmuth [AW] provide strong evidence that the maximum-likelihood hypothesis cannot be efficiently computed or even approximated in general. They show that the sample complexity (i.e., number of training sequences required from an information-theoretic point of view) of inferring a nearly optimal hmm is small, but the computational complexity is large in the size of the alphabet. They do not

address the question of complexity in terms of the size of the inferred model. In their setting the training sequences are all of a fixed size n and could have been generated by any probability distribution on R^n. The optimal hmm is the closest in relative entropy. Their proof is based on Angluin's proof of the NP-hardness of the consistency problem for 2-state dfa's [Ang89b]

We propose two alternatives to the maximum-likelihood hypothesis, which make use of the notion of ergodicity and the notion that an hmc is characterized by a small number of short strings. Both alternative hypotheses are based on a single training sequence and view the training sequence as an oracle for the exact probabilities of all finite strings which occur with no less than some fixed frequency in the sequence. The first hypothesis simply constructs the smallest pseudo-hmc which is compatible with the readings from the "oracle". The second hypothesis finds the pseudo-hmc of bounded size which most closely agrees with the "oracle".

We conjecture that for a certain value of the frequency threshhold, depending on the m-state hidden Markov chain generating the sequence, both of our hypotheses are consistent in the sense of [BP]. The right frequency threshhold should be the smallest probability of any string in a minimal set of strings that completely characterize the behavior of the hidden Markov chain.

We argue that the complexity of computing our hypotheses should depend partly on the complexity of the consistency problem for hmc's (with the given data restricted to being closed under substrings). The rate at which the behavior of each hypothesis converges to that of the hidden Markov chain should depend on the rate at which the hidden Markov chain converges to its limiting behavior.

1.5 Previous Work

There is a large and varied body of research on inference for models similar to hmc's. To put our work in context, it addresses (1) long-term behavior (so that ergodicity is essential in our model), (2) computational complexity of inference, and (3) structure of the hidden Markov chain, and how it affects the rate of convergence to long-term behavior and the complexity of inference. We also raise the question of (4) the power of learning with an oracle, as compared with learning from empirical data.

Related work has arisen in several disciplines. A summary of speech recognition research on hidden Markov models can be found in Levinson, *et. al.* [LRS]. Beginning with the work of Baum and Petrie [BP], this research generally emphasizes long-term behavior, but does not address complexity. The exception is the work of Abe and Warmuth [AW], which addresses complexity outside the context of long-term behavior. In automata theory, the books of Paz [P] and Rosenblatt [Ro] give good summaries of what was known about probabilistic automaton models before complexity became an issue. Rosenblatt emphasizes both the long-term behavior and the structure of hidden Markov chains, but does not consider rates of convergence. The more recent work of Tzeng [Tz92a, Tz92b] addresses complexity in a model which does not incorporate a notion of long-term behavior. In the theory of finite Markov chains, Rudich [Ru] considers asymptotic inferability of ergodic Markov chains from empirical data without regard to computational complexity. Finally, in the areas of information theory and coding, Merhav and Ziv [MZ89, ZM92] consider asymptotic inference of ergodic Markov chains from empirical data, also without regard to computational complexity.

There is some previous theoretical work on the intrinsic properties of pseudo-

hmc's, which are sometimes called "word functions" in reference to the probability assigned by the black box to each string. Paz [P] first observed that the minimal hmc equivalent to some hmc may be larger than the minimal pseudo-hmc. Cobham [Co] showed in an hmc model without the stationarity and ergodicity conditions that there are pseudo-hmc's with 5 states for which the minimal equivalent hmc exists and is arbitrarily large. It would be nice to prove that this result carries over to our model. That would show that an inference algorithm is more powerful if it is allowed to infer a pseudo-hmc to account for the behavior of an hmc.

Much of the literature on finite automata is relevant to our work. Nerode [Ne] has shown that a black box function that is realizable by a finite automaton is completely characterized by the behavior of the black box after having seen one of a finite set of strings. This is remarkably similar in spirit to [Gi]. A great deal of work has been done to adapt the insights of [Ne] to problems of inferring an automaton that accepts a given regular language. Gold [Go72] shows that inference in the limit is possible with queries to a membership oracle for the language. Angluin [Ang87] shows that with additional queries to a teacher oracle that will verify a correct hypothesis or provide a string which is a counterexample to an incorrect hypothesis, efficient inference is possible. Gold [Go78] shows that the problem of finding a dfa with m states that agrees with given data is NP-hard.

Tzeng [Tz92a] considers equivalence for a probabilistic automaton (pa) model which is like the hmc model in that it assigns a probability to each string of symbols from a finite alphabet. Instead of labelling the states with symbols, there is a different transition matrix for each symbol; certain states are designated accepting states. For any given hmc there is an easily constructed pa with the same black box behavior. Therefore, the algorithm in [Tz92a] for equivalence of pa's can be converted to an

algorithm for equivalence of hmc's. The reverse reduction from pa's to hmc's would only go through in an hmc model without the stationarity and ergodicity conditions.

Rudich [Ru] considers inference of unifilar Markov chains in the limit, but not from the point of view of computational complexity. Tzeng [Tz92b] considers inference and consistency on pa's and on a unifilar Markov chain model (mc) without the stationarity and ergodicity conditions. The probability oracle and teaching oracle for mc's he considers are essentially the same as ours. The oracle for pa's is very powerful: it reveals the distribution on the states the pa has reached after accepting the requested string. Tzeng gives efficient algorithms in both models for inference with a teacher oracle. He shows that inference is hard for pa's with a weaker oracle or for mc's with one oracle but not the other. He also shows that the consistency problem for pa's is hard. Each of these hardness results extends or reduces to a corresponding result for dfa's [Go78, Ang88, Ang89a], except for the case of mc's with the probability oracle. This is the case most similar to our first hardness argument. It remains to be seen whether the other hardness results translate to our model. We think not. Because of the ergodicity and stationarity assumptions of our model, only a trivial dfa is a special case of an hmc. By contrast, every dfa is also a pa.

Discrimination based on samples of classified objects is a problem in many statistical applications [Ha]. Many authors have studied the general problem of discriminating a known probability distribution from an unknown one, using empirical samples (see [Zi, Y]). The discrimination problem with oracle queries is apparently new, as is the question of whether the structure of the hidden Markov chain can lead one to an efficient solution.

Our learning model is not comparable to Valiant's PAC (probably approximately correct) learning model [V84] as it stands, but with certain modifications the PAC

oracle becomes comparable to ours. More precisely, it becomes weaker than ours. In the usual PAC model and its variants, samples are generated according to some probability distribution that is independent of the concept. Here it is natural to suppose that the concept itself (the hidden Markov chain) determines the conditional probability distribution from which each letter of each sample string is chosen. If we modify the PAC model to allow this; then we can simulate the PAC oracle with ours. The same remarks apply to Yamanishi's PAD (probably almost discriminative) learning model. Laird [La] has formalized the notion of an empirical learning problem as *unsupervised* learning.

1.6 Organization

Chapter 2 contains the Chernoff bound, with applications to entropy estimation and approximation algorithms. Chapter 3 contains theorems and algorithms for equivalence and minimization of pseudo-hmc's and an application to hypothesis testing for Huffman codes. It also contains hardness results for discrimination and inference, and randomized algorithms for discrimination in special cases, including the partial progress on discrimination which follows from our entropy estimation. Chapter 4 contains a discussion of further research in empirical inference, as well as a compilation of open problems from the other chapters.

Chapter 2

The Chernoff Bound

2.1 Introduction

The Main Result

Let G be a connected undirected graph with positive weights on the edges. We consider the random walk on G, which at each time step chooses an edge leaving the current vertex with probability proportional to the weight on the edge. We restrict our attention to non-bipartite graphs to ensure that the random walk will be ergodic. In this case the random walk must converge to a limiting distribution π on the vertices of G.

Let A be a subset of vertices of G. We consider the fraction of time the random walk spends in A. It is well known that for almost every trajectory of the random walk, the fraction of time spent in A converges to the limiting probability of A, $\pi(A)$ [Fe].

We will quantify this rate of convergence. We are concerned with the probability

that a given n-step trajectory of the random walk will deviate by a certain amount from $\pi(A)$. In our main theorem we show that this probability decays exponentially in the square of the amount of deviation, as a multiple of $\frac{1}{\sqrt{n}}$. This is the type of bound given by Chernoff [Ch] in the case of independent random variables (a special case of random walk).

Our bound depends on two geometric parameters of the graph G: the first parameter is a measure of the expansion of G and the second, which we denote *nonuniformity*, is $\max_{x,y \in G} \frac{\pi(x)}{\pi(y)}$, the largest ratio between the limiting probabilities of any two vertices of G. The better an expander G is, and the closer π is to being uniform, the more likely the random walk trajectory will be to visit A a fraction of time closely approximating $\pi(A)$. Our bound also depends on the starting distribution of the random walk, which may introduce a factor of $|G|$ into the bound on the deviation probability. If the random walk starts in the stationary distribution our bound does not depend on $|G|$.

Aldous [Ald87] showed that the fraction of visits to A converges in the L^2 norm to $\pi(A)$ and quantified the rate of convergence in L^2 in terms of the expansion of G. Lovász and Simonovits [LS] gave a similar result for arbitrary measure spaces. These results are second moment bounds analogous to Chebyshev's Theorem. They imply that the probability of deviation from $\pi(A)$ decays quadratically. Our main theorem strengthens that of [Ald87] for finite state spaces by establishing a Chernoff-type expontential decay in the probability of deviation, at the cost of introducing dependence on nonuniformity. Our result implies bounds on all higher moments of the fraction of visits to A, and it quantifies convergence in each L^p norm, $1 \leq p < \infty$ [Kah]. It also sharpens a theorem of Ajtai, Komlós, and Szemerédi [AKS] (see also [CW] and [IZ]), which showed that the probability of a deviation of constant size

decays exponentially in n.

We establish a more general form of the main theorem for estimating the expectation of a nonnegative function on the vertices of G. This more general form will be crucial for the application to entropy estimation and discrimination, as well as for applications to approximation algorithms. $\pi(A)$ is the expectation of χ_A, the indicator function of A, and the main theorem states that the fraction of visits to A is a good estimate of this expectation. Our more general theorem is simply the analogous result for estimating the expectation $\mathrm{E}f$ of an arbitrary nonnegative function f. In this case the bound depends on $\|f\|_\infty$ as well as on the expansion and nonuniformity of G.

Entropy Estimation

In the special case where the edges of G are not weighted, we use our result to approximate the entropy of the random walk very quickly. We view the random walk on G as an information source whose alphabet is the vertices of G. It is convenient to state the result in this form, although it applies to any labelling of the vertices such that the Markov chain is unifilar. (A unifilar Markov chain in this context is one in which each state has nonzero transition probability to at most one state of each label.) Because of the additional restriction to undirected graphs, in the case of a labelling with 0's and 1's the result is not general enough to be interesting; it includes only chains and cycles. For three-letter alphabets it is already nontrivial. If G has constant expansion and bounded degree then a good entropy estimate requires only $O(\log |G|)$ steps of the random walk.

This result quantifies the rate of convergence of the classical Shannon-McMillan

asymptotic equipartition property [Ash]. The classical result assumes an ergodic information source with entropy H. It says intuitively that for large n, roughly 2^{Hn} of the n-bit strings each have probability roughly 2^{-Hn}. For purposes of encoding n-bit blocks of output from the source into blocks of some fixed shorter length greater than Hn, we give a bound on the exponent of the error probability. This the first bound on the error exponent for fixed-length noiseless source coding based on the structure of the underlying Markov chain. Previous bounds on the error exponent based on divergence retain an asymptotic flavor. Large deviation methods for the source coding problem were used in [Na, Ana].

Approximation Algorithms

Estimating $\pi(A)$ (or in some applications, $\mathrm{E}f$) is a fundamental problem for approximation algorithms in which A and G are exponentially large combinatorial sets such as sets of matchings of a graph [JS89]. The basic idea is to generate a number of random sample points in G and compute the fraction that are in A. The standard procedure is to use the rapid mixing property of the random walk on G to generate a single nearly random sample point from π. This process is repeated to generate the number of independent sample points Chernoff's bound requires [JS89, LS]. Here we concentrate on the alternative procedure of Aldous [Ald87], which is first to generate a nearly random starting point and then to sample every point along a single trajectory of the random walk.

With our Chernoff-type analysis we are able to simplify this procedure of Aldous and to improve its performance slightly when G is uniform and the tolerated error probability δ of the procedure is small (typically $\delta = \frac{1}{n}$). The Chebyshev-type bound

on the running time implied in [Ald87] is proportional to $\frac{1}{\delta}$. A well-known statistical technique of Jerrum, Valiant, and Vazirani [JVV] (see also [LS]) shows that all that is really needed is an error probability of 1/4, since repeating the estimate $12 \log \frac{1}{\delta}$ times and taking the median of the results improves 1/4 to δ. However, using this method to boost the confidence of an algorithm requires repeating the initial step of finding a nearly random starting point, which can dominate the cost of the estimate (see [Ald87, Example 4.1]). For constant $\pi(A)$ our analysis of Aldous' procedure gives a bound on the running time proportional to $\log \frac{1}{\delta}$ directly. This eliminates the need for the statistical technique and improves the running time by as much as $\log \frac{1}{\delta}$.

In many important applications the natural random walk to use is highly nonuniform, and in these cases it becomes attractive to develop algorithms based on the asymptotically weaker Chebyshev-type bound of Aldous [Ald87]. In fact we are able to show quite generally that any algorithm that uses the standard sampling procedure to estimate $\pi(A)$ can be improved by substituting Aldous' procedure and the statistical technique of [JVV]. We do this in part by extending the Chebyshev-type bound of Aldous [Ald87] in the following way. The bound there depends on $(\min_{x \in G} \pi(x))^{-1}$; we show that in the setting of most applications the bound need only depend on $(\pi(s))^{-1}$, where s is the starting point of the random walk. This makes a difference when π is highly nonuniform. Lovász and Simonovits [LS] have incorporated similar ideas in their algorithm for approximating the volume of a convex body; however, these ideas first find systematic application here.

We use the extended Chebyshev-type bound to improve two algorithms of Jerrum and Sinclair. We improve the running time of their algorithm for approximating the number of perfect matchings in a graph [SJ], from $O(q^3 n^5 \log^2 n)$ to $O(q^2 n^4 \log n)$, where q is a known upper bound on the ratio of the number of matchings with $\frac{n}{2} - 1$

edges to the number of perfect matchings. (In the case of dense bipartite graphs it is known that $q \leq n^2$.) Our modification of their algorithm includes a new method of selecting only those sets A for which $\pi(A)$ is not too small. We also improve the running time of their algorithm for approximating the value of the partition function of a ferromagnetic Ising system [JS91], from $O(m^3n^{11})$ to $O(m^2n^{11})$, where n is the number of vertices and m the number of edges of the system. The analysis of this improved algorithm requires our extended version of the Chebyshev-type bound of Aldous [Ald87].

Our Chernoff-type bound shows that when $\pi(A)$ is large and G is uniform, for example in the case of the Dyer, Frieze, Kannan algorithm for computing the volumes of convex bodies [DFK], Aldous' procedure alone considerably outperforms the standard procedure. The improvement in running time is roughly a factor of $\frac{\log|G|}{\pi(A)}$. We also answer a point raised in [Ald87, Example 4.2] by showing that even without generating a random starting point for the sample trajectory, we still get a good estimate of $\pi(A)$ in polynomial time.

Methods

In section 2.2 we state our main mathematical result, Theorem 1, a Chernoff-type bound for random walks. Our proof strategy begins with Theorem 2, due to T. Höglund [Hö, Theorem 5.5], which follows the method of Cramér [Cr] and Chernoff [Ch] of estimating the deviation probability in terms of the moment generating function of the number of visits to A. Höglund's result shows in theory that the problem of bounding the probability of deviation can be reduced to the problem of estimating the largest eigenvalue of a perturbation of the transition matrix for the

random walk. The applicability of this reduction depends on the transition matrix. Generally, the eigenvalues of a matrix may vary wildly under small perturbations of the matrix [SS, p. 166]. In his paper, Höglund used his method to derive a bound similar to Chernoff's for the case of Bernoulli trials (in which the transition matrix has identical rows).

Theorem 1 demonstrates the applicability of Höglund's approach to random walks on weighted graphs. The transition matrix in this case is similar to a symmetric matrix. We use the properties of symmetric matrices and of the similarity transformations under consideration for our perturbation estimates. In Lemma 2 we establish a bound on the logarithm of the largest eigenvalue of a perturbation of the transition matrix by estimating its second derivative. The essential step is to notice that the eigenvalue is an *analytic* function of the perturbation parameter. It is possible to bound the second derivative of an analytic function using Cauchy's estimate [Ah], by bounding the function values on a well-chosen loop in the complex plane. This leads to equation (2.11), a bound on the probability of deviation in terms of the amount of perturbation of the transition matrix. Theorem 1 follows by choosing the optimal amount of perturbation.

The organization of this chapter is as follows. Section 2.2 contains the main theorem and the application to entropy estimation. We will discuss an application to discrimination with oracle queries in Section 3.6 of the next chapter. In Section 2.3 we describe the applications to approximation algorithms, including our extension of the Chebyshev-type bound of Aldous.

A good summary and list of references to previous work on central limit theorems for Markov chains are in Malinovskii [Mal] (and references therein). See also Koopmans [Koo], Turchin [Tu], and Jain [J]. Koopmans considers the asymptotic rate

of discrimination of two random walks on the real line that have positive transition densities. His methods and results are analogous to those of Höglund for the case of finite state-spaces. Specifically, Koopmans bounds a certain moment generating function in terms of the eigenvalues of an integral operator defined by the transition density.

2.2 The Main Theorem

Preliminaries

NOTATION

Let G be a connected, non-bipartite undirected graph on m nodes. Let each edge (x, y) of G be assigned a positive weight w_{xy}. We define the weight w_x of the vertex x by the formula $w_x = \sum_{x \sim y} w_{xy}$.

A random walk on a weighted graph is equivalent to a time-reversible finite Markov chain. The states of the Markov chain are the vertices of the graph. The Markov chain can be described by a transition matrix P whose ij^{th} entry p_{ij} is the probability (independent of time) of moving to state j after entering state i. In terms of G,

$$p_{ij} = \begin{cases} \frac{w_{ij}}{w_i}, & \text{if } (i,j) \text{ is an edge of } G \\ 0, & \text{if not.} \end{cases}$$

Let $x_0, x_1, \ldots$ be the random walk according to P on G, starting in some distribution $\mathbf{q}$ on the vertices. The connectivity and non-bipartite conditions on G guarantee that regardless of the starting distribution $\mathbf{q}$ the random walk will converge to the stationary distribution π, given by $\pi(x) = \frac{w_x}{\sum_{y \in G} w_y}$. We define the *nonuniformity* ν

of π (equivalently, the nonuniformity of G) by $\nu = \max_{x,y \in G} \frac{\pi(x)}{\pi(y)}$. Let $\frac{\mathbf{q}}{\sqrt{\pi}}$ denote the vector with entries $\frac{\mathbf{q}}{\sqrt{\pi}}(x) = \frac{\mathbf{q}(x)}{\sqrt{\pi(x)}}$, and let $N_{\mathbf{q}} = \|\frac{\mathbf{q}}{\sqrt{\pi}}\|_2$. Let $\mathbf{1} = (11\cdots1)$ be the m-vector of all 1's.

EXPANSION AND THE EIGENVALUE GAP

The eigenvalues of P are real, and the largest eigenvalue (in absolute value) is 1 (see [Se]). We denote the second largest eigenvalue by λ_2 and the second largest eigenvalue in absolute value by $\overline{\lambda}$ and we define the *eigenvalue gap* by $\epsilon = 1 - \lambda_2$. By definition $\epsilon \geq 1 - \overline{\lambda}$. The quantity $1 - \overline{\lambda}$ is directly related to the expansion of G [AM, Ta, Alo, SJ]. Therefore, if G is an expander ϵ will be large.

Let A be a set of vertices of G. Let χ_A be the indicator function of A: $\chi_A(x) = 1$, if $x \in A$, and 0 otherwise. We define the *number of visits to A in n steps* as $t_n = \chi_A(x_1) + \cdots + \chi_A(x_n)$.

Logarithms are in base e unless otherwise subscripted.

Main Theorem

We now state our main result, a large deviation bound for a random walk on a weighted graph, in terms of the eigenvalue gap and the nonuniformity of the graph.

Theorem 1 *Let G be a weighted graph with eigenvalue gap ϵ and nonuniformity ν. Consider the random walk on G starting in distribution $\mathbf{q}$ and having stationary distribution π. Let A be a set of vertices in G and let t_n be the number of visits to A in n steps. Let $\gamma \geq 0$. Then for any positive integer n,*

$$\Pr[t_n - n\pi(A) \geq \gamma] \leq 2N_{\mathbf{q}} e^{-\gamma^2 \epsilon / 20n\nu}. \quad \blacksquare \tag{2.1}$$

Remarks. (i) We may write $E_\pi \chi_A$ for $\pi(A)$ in Theorem 1. As we will see the proof of this theorem uses only that χ_A is a nonnegative function on the vertices of G such that $\|\chi_A\|_\infty \leq 1$. For an arbitrary nonnegative function f on the vertices of G, we may substitute f for χ_A in the definition of t_n. Equation (2.1) becomes

$$\Pr[t_n - nE_\pi f \geq \gamma] \leq 2N_{\mathbf{q}} e^{-(\frac{\gamma}{\|f\|_\infty})^2 \epsilon/20n\nu}. \tag{2.2}$$

(ii) Applying the theorem to $G \setminus A$ gives the same bound on $\Pr[t_n - n\pi(A) \leq -\gamma]$.

Reducing the Problem to Perturbation Theory

Before proving Theorem 1, we lay the groundwork for our proof with Theorem 2, a simplified version of a result of T. Höglund [Hö, Theorem 5.5]. This result follows the strategy of Cramér [Cr] and Chernoff [Ch], which is to estimate the deviation probability in terms of the so-called moment generating function $m(r)$, evaluated at a judiciously chosen $r > 0$. $m(r)$ is defined as the expectation $\mathrm{E}e^{rt_n}$. We will see that this strategy reduces the problem of estimating the left-hand side of (2.1) to a problem of analyzing a perturbation of the transition matrix P.

We must now introduce a perturbation $P(r)$ of the transition matrix P, where r is any complex number (we will often restrict r to the nonnegative real line). For $j \in A$ we multiply the j^{th} column vector of P by e^r to get the j^{th} column vector of $P(r)$. The remaining columns of $P(r)$ will equal the corresponding columns of P.

Note that $P(r) = PE_r$, where $E_r = \mathrm{diag}(e^{r\chi_A})$. ($E_r$ is not to be confused with the symbol for expectation, E, which is not italicized.) We now show that P and $P(r)$ are similar to symmetric matrices. Let M be the weighted adjacency matrix of G: the ij^{th} entry of M is w_{ij} if (i,j) is an edge of G and 0 otherwise. Let $D = \mathrm{diag}(\frac{1}{w_i})$.

Then

$$P = \sqrt{D}S\sqrt{D^{-1}}, \quad \text{where } S = \sqrt{D}M\sqrt{D}, \text{ and}$$
$$P(r) = \sqrt{DE_r^{-1}}S_r\sqrt{E_rD^{-1}}, \quad \text{where } S_r = \sqrt{DE_r}M\sqrt{DE_r} \tag{2.3}$$

By equation (2.3) the eigenvalues of $P(r)$ are real for $r \geq 0$; in this case let $\lambda(r)$ and $\lambda_2(r)$ denote the largest and second largest eigenvalues of $P(r)$, respectively. Note that $P(0) = P$, $\lambda(0) = 1$ (with left eigenvector π^T and right eigenvector $\mathbf{1}$), and $\lambda_2(0) = \lambda_2$. For $r \geq 0$, let the eigenvalue gap of $P(r)$ be denoted by $\epsilon_r = \lambda(r) - \lambda_2(r)$.

Theorem 2 *In the setting of Theorem 1, let $r \geq 0$. For any positive integer n,*

$$\Pr[t_n - n\pi(A) \geq \gamma] \leq e^{-r(n\pi(A)+\gamma)+n\log\lambda(r)} \frac{\mathbf{q}P(r)^n\mathbf{1}}{\lambda(r)^n}. \tag{2.4}$$

Proof. By Markov's inequality,

$$\begin{aligned}\Pr[t_n \geq n\pi(A)+\gamma] &= \Pr[e^{r(t_n)} \geq e^{r(n\pi(A)+\gamma)}] \\ &\leq e^{-r(n\pi(A)+\gamma)}\mathrm{E}_{\mathbf{q}}e^{rt_n},\end{aligned} \tag{2.5}$$

where $\mathrm{E}_{\mathbf{q}}$ denotes the expectation given that x_0 is chosen according to $\mathbf{q}$. This expectation can be evaluated by summing over all possible trajectories $x_0, x_1, \ldots, x_n$ (where t_n is understood to be a function of the trajectory):

$$\mathrm{E}_{\mathbf{q}}e^{rt_n} = \sum_{x_0,\ldots,x_n} e^{rt_n}\mathbf{q}(x_0)\prod_{i=1}^{n} p_{x_{i-1}x_i} = \mathbf{q}P(r)^n\mathbf{1}. \tag{2.6}$$

Combining Equation (2.6) with Inequality (2.5) we obtain

$$\Pr[t_n \geq n\pi(A)+\gamma] = e^{-r(n\pi(A)+\gamma)+n\log\lambda(r)} \frac{\mathbf{q}P(r)^n\mathbf{1}}{\lambda(r)^n}. \quad \blacksquare \tag{2.7}$$

Explanation. The reasons that this theorem is useful are twofold:

1. For sufficiently nice matrices $P(r)$ the fraction on the right-hand side of equation (2.4) is small. In fact, in our situation this fraction turns out to measure how close the starting distribution $\mathbf{q}$ is to the stationary distribution π (see Lemma 1 below).

2. The exponent $-r(n\pi(A)+\gamma)+n\log\lambda(r)$ in the right-hand side of equation (2.4) is negative for small r because, as we explain below, $\frac{d\log\lambda}{dr}|_{r=0} = \pi(A)$, and because $\log\lambda(0) = 0$. Our goal then (accomplished by Lemma 2) will be to bound this exponent away from zero for some r in order that a meaningful bound on $\Pr[t_n - n\pi(A) \geq \gamma]$ will follow from Theorem 2.

It is by making these remarks precise that we will prove Theorem 1. Using this approach Höglund derived a bound similar to Chernoff's for the case of Bernoulli trials. We establish our bound for random walks using estimates from matrix perturbation theory.

PROOF OF THEOREM 1

We take points 1. and 2. above in turn.

Lemma 1 *For* $0 \leq r \leq 1$,

$$\frac{\mathbf{q}P(r)^n\mathbf{1}}{\lambda(r)^n} \leq 2N_{\mathbf{q}}.$$

Proof. Using Fischer's min-max characterization of the eigenvalues of a symmetric matrix [SS],

$$\begin{aligned}\frac{\mathbf{q}P(r)^n\mathbf{1}}{\lambda(r)^n} &= \frac{\mathbf{q}\sqrt{DE_r^{-1}}S_r{}^n\sqrt{E_rD^{-1}}\mathbf{1}}{\lambda(r)^n}\\ &\leq e^{\frac{r}{2}}N_{\mathbf{q}}\\ &\leq 2N_{\mathbf{q}}. \blacksquare\end{aligned}$$

Before stating the next lemma we must define for each $r \geq 0$ the matrix $B(r) = \frac{1}{e-1}(P(r+1) - P(r))$. $B(r)$ is the result of zeroing out the j^{th} column of $P(r)$ for every $j \notin A$.

Lemma 2 *If r is a real number such that $0 \leq e^r - 1 \leq \frac{\epsilon}{4}$, then*

$$\log \lambda(r) \leq r\pi(A) + r^2 \frac{5\nu}{\epsilon}.$$

Proof. Fix r, $0 \leq e^r - 1 \leq \frac{\epsilon}{4}$. For each nonnegative real number x, we may expand the function $\log \lambda(y)$ in a Taylor series about the point $y = x$ (see [Wil]):

$$\log \lambda(y) = \log \lambda(x) + m_x(y-x) + V_x \frac{(y-x)^2}{2} + \cdots .$$

We are adopting the notations m_x and V_x from [Hö]. m_x is intended to suggest "mean". The reason for this is that $m_0 = \pi B(0)\mathbf{1} = \pi(A)$, which follows from an elementary calculation from perturbation theory of matrices (see [Wil, p. 69]). Plugging this into one version of Taylor's theorem, we see that

$$\log \lambda(r) = r\pi(A) + r^2 \int_0^1 (1-t)V_{rt}dt . \tag{2.8}$$

(The notation V_x, also from [Hö], is intended to suggest "variance." To motivate this, it is helpful to set $m = 0$ and work out the case of Bernoulli trials with mean p.)

We must now bound V_x for $0 \leq x \leq r$. The first step is to show that the eigenvalue gap of $P(x)$ is not much smaller than the eigenvalue gap of P. Note that in the next two claims we are still assuming that r is a real number such that $0 \leq e^r - 1 \leq \frac{\epsilon}{4}$.

Claim 1 *If $0 \leq x \leq r$, then $\epsilon_x \geq \frac{3\epsilon}{4}$.*

Proof. Because of the assumption on r it is enough to show that $\epsilon_x \geq \epsilon - (e^x - 1)$. Also, $P(x) \geq P$ in each entry (because $e^x > 1$), and so the Perron-Frobenius Theorem

[Se] says that $\lambda(x) \geq 1$. Let $\mu < \lambda(x)$ be any other eigenvalue of $P(x)$. It will suffice to show that $\mu \leq \lambda_2 + e^x - 1$.

The matrix S of equation (2.3) is diagonalizable; there exist a unitary matrix U and diagonal matrices D' and D_A such that

$$B(0) = PD_A\,,$$
$$D' = U^T\sqrt{D^{-1}}P\sqrt{D}U\,, \text{and}$$
$$\|D'\|_2 = \|D_A\|_2 = 1\,.$$

If $\mu \leq \lambda$ we are done. Otherwise, the following matrix is singular:

$$U^T\sqrt{D^{-1}}(P+(e^x-1)B(0)-\mu I)\sqrt{D}U = (D'-\mu I)(I+(D'-\mu I)^{-1}(e^x-1)D'U^TD_AU)\,.$$

Therefore,

$$1 \leq \|(D'-\mu I)^{-1}(e^x-1)D'U^TD_AU\|_2 \leq \tfrac{1}{\mu-\lambda_2}(e^x-1)\,.$$

(The inequality on the left uses the continuity of the function $\lambda_2(y)$.) This proves Claim 1. ∎

Claim 2 *If* $0 \leq x \leq r$, *then* $V_x \leq \frac{10\nu}{\epsilon}$.

Proof. Fix x, $0 \leq x \leq r$. Our strategy is to use Cauchy's estimate from complex analysis (see [Ah]), which can be used to bound V_x in terms of the maximum value attained by $\lambda(z)$ in a complex neighborhood of x. We bound this maximum value indirectly: the convergence of a certain loop integral will imply that $\lambda(z)$ lies inside the loop.

For z in a small complex neighborhood of x we may write $P(z) = P(x) + (e^{z-x} - 1)B(x)$. A fundamental theorem of perturbation theory [Kat, §II.1.4] says that the

projection matrix for $\lambda(z)$ is given by the operator-valued complex integral

$$-\tfrac{1}{2\pi i}\int_\Gamma (P(z)-\zeta I)^{-1}\,d\zeta\,,$$

where Γ is any circle with $\lambda(x)$, and no other eigenvalues of $P(x)$, in its interior. The main fact we use is that if $\lambda(z)\in\Gamma$ then the integrand will have a singularity at $\lambda(z)$. To avoid this we choose Γ to have center $\lambda(x)$ and radius $\frac{\epsilon_x}{2}$. The norm of the integrand is uniformly bounded on Γ as long as the following holds [Kat, §II.3.1]:

$$|e^{z-x}-1| \;<\; \|B(x)(P(x)-(\lambda(x)-\tfrac{\epsilon_x}{2})I)^{-1}\|_2^{-1} \tag{2.9}$$

The matrix S_x of equation (2.3) is diagonalizable; by an expansion like the one used to prove Claim 1, $\|B(x)(P(x)-(\lambda(x)-\frac{\epsilon_x}{2})I)^{-1}\|_2 \le \frac{2\sqrt{\nu}}{\epsilon_x}$. Some steps of algebra show that in order for (2.9) to hold it is enough that

$$|z-x| \;<\; \tfrac{3\epsilon_x}{8\sqrt{\nu}}\,. \tag{2.10}$$

Whenever (2.10) holds $\lambda(z)$ does not lie on Γ. But by continuity of $\lambda(z)$, (2.10) must imply that $\lambda(z)$ lies *inside* Γ, and therefore $|\lambda(z)-\lambda(x)| \le \frac{\epsilon_x}{2}$. Comparing Taylor series for $\lambda(z)$ and $\log\lambda(z)$, we see that $V_x \le 2\,\frac{d^2\lambda}{dz^2}|_{z=x}$. Cauchy's estimate for the Taylor coefficients of $\lambda(z)$ (see [Ah]) and Claim 1 show that

$$V_x \le 2\tfrac{\epsilon_x}{2}/(\tfrac{3\epsilon_x}{8\sqrt{\nu}})^2 = \tfrac{64\nu}{9\epsilon_x} \le \tfrac{10\nu}{\epsilon}\,.$$

This proves Claim 2. ∎

Equation (2.8) and Claim 2 imply that

$$\log\lambda(r) \;\le\; r\pi(A) + r^2\tfrac{10\nu}{\epsilon}\int_0^1 (1-t)dt \;=\; r\pi(A) + r^2\tfrac{5\nu}{\epsilon}\,,$$

which proves Lemma 2. ∎

To complete the proof of Theorem 1, we combine equation (2.4), Lemma 1, and Lemma 2 to get

$$\Pr[t_n - n\pi(A) \geq \gamma] \leq 2N_{\mathbf{q}}\, e^{-n(r\frac{\gamma}{n} - r^2\frac{5\nu}{\epsilon})}. \tag{2.11}$$

The expression $E(r) = r\frac{\gamma}{n} - r^2\frac{5\nu}{\epsilon}$ is quadratic in r and is maximized when $r = \frac{\gamma\epsilon}{10n\nu}$, which satisfies the condition of Lemma 2 that $e^r - 1 \leq \frac{\epsilon}{4}$ (we can assume $\gamma < n$, because otherwise Theorem 1 is trivially true). For this value of r,

$$E(r) = \frac{\gamma^2\epsilon}{20n^2\nu}.$$

Substituting into equation (2.11),

$$\Pr[t_n - n\pi(A) \geq \gamma] \leq 2N_{\mathbf{q}}\, e^{-\gamma^2\epsilon/20n\nu}.$$

This completes the proof of Theorem 1. ∎

Entropy of Markov Sources

We now consider the special case of a random walk on a non-weighted undirected graph G. We view the random walk as an information source whose alphabet is the vertices of G. Theorem 1 enables us to quantify the rate of convergence of the classical Shannon-McMillan asymptotic equipartition property for this information source [Ash]. We will then be able to estimate the entropy of the source very quickly, in $O(\log |G|)$ steps when G has constant expansion and constant nonuniformity. For purposes of fixed-length noiseless coding of the source, this result will imply a bound on the error exponent in terms of the expansion of G.

Let G have eigenvalue gap ϵ and nonuniformity ν. Let M be twice the number of edges of G. Let $P = (p_{ij})$ be the transition matrix for the random walk on G.

For all i, j, $p_{ij} = \frac{1}{d_i}$, where d_i is the degree of vertex i. $\pi(i) = \frac{d_i}{\sum_j d_j} = \frac{d_i}{M}$; therefore $M \geq \max_i \frac{1}{\pi(i)} \geq \nu$. Let $x_0, x_1, \ldots$ be a random walk on G starting from stationary. Consider the random sequence $X = x_1, x_2, \ldots$ an information source.

THE RANDOM WALK AS AN INFORMATION SOURCE

In general, the *entropy* $H(Y)$ of an information source $Y = y_1, y_2, \ldots$, is defined in terms of the ordinary Shannon entropy: $H(Y) = \lim_{n\to\infty} E\, H[y_n | y_1, \ldots, y_{n-1}]$. In our case there is a simple formula:

$$H(X) = E\, H(x_1|x_0) = -E \log_2 p_{x_0 x_1} = E_\pi \log_2 d_{x_0}\,. \tag{2.12}$$

The Shannon-McMillan Theorem [Ash, Theorem 6.6.1] states that under an ergodicity assumption which is satisfied here, an information source Y has the *asymptotic equipartition property* (AEP): for a fixed length n of source sequences, there are approximately $2^{nH(Y)}$ source sequences each of approximate probability $2^{-nH(Y)}$, and the probability of the "bad" set of remaining sequences tends to zero as n tends to infinity. The next theorem establishes an upper bound on the probability of the "bad" set.

Now consider the particular information source X and a finite sequence $x_1, \ldots, x_n$, generated by X. Define the *empirical entropy* V_n of this finite sequence by $V_n = -\frac{1}{n} \log_2 \Pr[x_1, \ldots, x_n]$.

Theorem 3 *Let X be the information source generated by the random walk on G. Let $x_1, \ldots, x_n$ be a finite sequence generated by X, with empirical entropy V_n. Then*

$$\Pr[|V_n - H(X)| \geq \gamma] \leq 4e^{-(\gamma - \frac{\log_2 M}{n})^2 n\epsilon / 20\nu \log_2^2 \nu}\,. \tag{2.13}$$

Proof. Define a nonnegative function g on the vertices of G by $g(x) = \log_2 d_x$. Set $f(x) = g(x) - \min_y g(y)$. Then $\|f\|_\infty = \log_2 \nu$ and by equation (2.12), $E_\pi f =$

$H(Y) - \min_y g(y)$. Let $t_n = f(x_1) + \cdots + f(x_n)$. Then $V_n - \min_y g(y) = \frac{t_n}{n} + \frac{1}{n}\log_2 \frac{1}{\pi(x_1)} - \frac{1}{n}\log_2 d_{x_n}$. Using equation (2.2) and remark (i) following Theorem 1,

$$\begin{aligned}
\Pr[V_n - H(X) \geq \gamma] &= \Pr[\tfrac{t_n}{n} - (H(X) - \min_y g(y)) \geq \gamma - \tfrac{1}{n}\log_2 \tfrac{1}{\pi(x_1)} + \tfrac{1}{n}\log_2 d_{x_n}] \\
&\leq 2\exp[-(\tfrac{\gamma - \frac{1}{n}\log_2 M}{\frac{1}{n}\log_2 \nu})^2 \epsilon/20n\nu] \\
&\leq 2e^{-(\gamma - \frac{\log_2 M}{n})^2 n\epsilon/20\nu\log_2{}^2\nu}
\end{aligned}$$

The inequality in the other direction is similar. ∎

UNIFILAR SOURCES

Let $\chi : V(G) \to \{0, 1, \ldots, k-1\}$ be any labelling of the vertices of G such that the random walk on G is unifilar; that is, no vertex G is adjacent to two or more vertices of the same label. Let $Y = y_1, y_2, \ldots$ be the sequence of labels: $y_i = \chi(x_i)$. Then the entropy of the information source Y still satisfies equation (2.12): $H(Y) = E_\pi \log_2 d_{x_0}$. The definition of empirical entropy for Y is also the same, and we immediately have the following generalization:

Corollary 1 *Let V_n denote the empirical entropy of $y_1, \ldots, y_n$. Then*

$$\Pr[|V_n - H(Y)| \geq \gamma] \leq 4e^{-(\gamma - \frac{\log_2 k|G|}{n})^2 n\epsilon/20k\log_2{}^2 k}. \quad \blacksquare \tag{2.14}$$

Remarks. (i) The logarithm of the right-hand side of Equation (2.14) gives an upper bound on the error exponent for fixed-length coding of the source Y. [Should see if this is comparable to previous bounds.]

(ii) To estimate $H(Y)$ to within an additive error γ with probability at least $1 - \delta$ using Corollary 1, the length of random walk required is $O(\frac{k\log^2 k}{\epsilon\gamma^2} \log\frac{1}{\delta} \log k|G|)$. When the number of labels k is constant and G has constant expansion, this simplifies to $O(\frac{1}{\gamma^2} \log\frac{1}{\delta} \log|G|)$.

(iii) When $k = 2$, G must be either a chain or a cycle. In order to generate a larger class of hidden Markov chains, either Theorem 3 and Corollary 1 must be extended to nonunifilar Markov sources, or Theorem 1 must be extended to non-reversible Markov chains.

2.3 Approximation Algorithms

General Comparison of Procedures

We now discuss the cost of estimating $\pi(A)$ to within some fraction $\beta\pi(A)$ with probability $1 - \delta$, using random walks to generate sample points from G. The cost of an algorithm will be the total number of random walk steps taken (see the discussion of measures of cost at the end of this subsection). Theorem 4 establishes the cost of Aldous' procedure. We use Theorem 5 to show that Aldous' procedure together with the statistical technique of [JVV] is better than the standard procedure. We also show that for constant $\pi(A)$ and uniform G Aldous' procedure does as well or slightly better without the statistical technique. Finally, we remark that Aldous' procedure gives good estimates in polynomial time even if we deterministically choose the starting point of the sample trajectory.

We assume we can efficiently find one point s in A from which to start a random walk. The procedures are

Standard procedure (SP) Start the random walk at s and simulate it for k' steps, so that the final state is distributed according to **q'**. Take the final state as a sample point. Repeat this l' times by choosing the same starting point s and

taking a walk of length k' each time. This procedure generates l' independent sample points from the same distribution **q'**.

Aldous' procedure (AP) Start the random walk at s and simulate it for k steps (the "delay"), so that the final state x_0 is distributed according to $\mathbf{q}$. Starting from x_0, continue the random walk l more steps taking each subsequent point as a sample point.

Aldous' procedure with statistical technique (AP+ST) Choose k and l so that Aldous' procedure estimates $\pi(A)$ to within $\beta\pi(A)$ with probability 3/4. Repeat $12 \log \frac{1}{\delta}$ times and take the median of the answers.

For a distribution $\mathbf{d}$ on the vertices of G, let the *chi-square distance from* π be defined by $\chi_d^2 = \sum_x \pi(x)(\frac{\mathbf{d}(x)}{\pi(x)} - 1)^2$. Let χ_s^2 denote the chi-square distance from π of the initial distribution concentrated at s.

Theorem 4 *The cost of estimating $\pi(A)$ to within $\beta\pi(A)$ with probability $1-\delta$ using Aldous' procedure (AP) is*

$$\frac{1}{\epsilon} \log \frac{1}{\pi(s)} + \frac{20\nu}{\epsilon} \frac{1}{\beta^2 \pi(A)^2} \log \frac{8}{\delta}.$$

Proof. Let $k = \frac{1}{\epsilon} \log \frac{1}{\pi(s)}$ [1] and $l = \frac{20\nu}{\epsilon} \frac{1}{\beta^2 \pi(A)^2} \log \frac{8}{\delta}$. In the notation of Theorem 1, $N_{\mathbf{q}}^2 = 1 + \chi_{\mathbf{q}}^2$. By [Fi, equation (2.11)], $N_{\mathbf{q}} \leq 1 + \sqrt{\chi_s^2}(1-\epsilon)^k \leq 1 + \frac{1}{\pi(s)} e^{-\epsilon k} \leq 2$. By Theorem 1 and the ensuing Remark (ii),

$$\Pr[|\tfrac{i_l}{l} - \pi(A)| \geq \beta\pi(A)] \leq 4 N_{\mathbf{q}} e^{-\beta^2 \pi(A)^2 \epsilon l / 20\nu} \leq \delta. \quad \blacksquare$$

[1] To eliminate problems of near-periodicity, it may be necessary to choose a random Poisson delay time with mean k. See [Ald87] for more details.

Comparison to other procedures. Let $\pi_{\min} = \min_x \pi(x)$. Sinclair and Jerrum showed in [SJ] that for the standard procedure (SP) to estimate $\pi(A)$ to within $\beta\pi(A)$ with probability $1 - \delta$ requires choosing $k' \geq \frac{1}{\epsilon}(\log \frac{1}{\pi_{\min}} + \log \frac{1}{\delta})$. Chernoff's bound then requires that $l' \geq \frac{1}{\beta^2\pi(A)} \log \frac{1}{\delta}$. The method used in Lemma 3 of [JS91] is easily seen to be generally applicable; this improves the $\pi_{\min}$ to $\pi(s)$ in the expression for k'.

The analysis of [Ald87, Proposition 4.2] implies that the cost of Aldous' procedure with the statistical technique (AP+ST) of [JVV] is $O[\frac{1}{\epsilon} \log \frac{1}{\delta}(\log \frac{1}{\pi_{\min}} + \frac{1}{\beta^2\pi(A)})]$. We are able to improve the picture with the following theorem which serves to replace $\pi_{\min}$ by $\pi(s)$. This will come in handy in the applications of the next section to particular algorithms.

Theorem 5 *The cost of estimating $\pi(A)$ to within $\beta\pi(A)$ with probability $1-\delta$ using AP+ST is*

$$12 \log \tfrac{1}{\delta}[\tfrac{1}{\epsilon} \log(\tfrac{12}{\pi(s)\beta^2\pi(A)^2}) + \tfrac{1}{\epsilon}\tfrac{12}{\beta^2\pi(A)}] .$$

Proof. Let $k = \frac{1}{\epsilon} \log(\frac{12}{\pi(s)\beta^2\pi(A)^2})$ Let $\|\mathbf{q} - \pi\|$ denote the total variation distance between $\mathbf{q}$ and π. Citing, for example, [JS91, Theorem 6], we have that $\|\mathbf{q} - \pi\| < \frac{1}{12}\beta^2\pi(A)^2$. [2] Let $l = \frac{1}{\epsilon}\frac{12}{\beta^2\pi(A)}$. Define a vector $\mathbf{b}$ by letting $\mathbf{b}_j = \mathrm{E}[(\frac{t_l}{l} - \pi(A))^2 \,|\, x_0 = j]$. Observe that $\|\mathbf{b}\|_\infty \leq 1$, and therefore by Chebyshev's Inequality and Proposition 4.1 of [Ald87],

$$\Pr[|\tfrac{t_l}{l} - \pi(A)| \geq \beta\pi(A)] \leq \frac{\mathrm{E}_{\mathbf{q}}\mathbf{b}}{\beta^2\pi(A)^2}$$

[2] The requirement in [JS91, Theorem 6] that $p_{ii} \geq 1/2$ for all i is a minor technical condition satisfied in the applications we consider here. Even where it is not satisfied the problems of periodicity that may arise can be circumvented by replacing k with a random Poisson delay time as mentioned in the previous footnote.

$$\leq \frac{1}{\beta^2\pi(A)^2}[\mathrm{E}_\pi \mathbf{b} + |\mathrm{E}_\mathbf{q}\mathbf{b} - \mathrm{E}_\pi \mathbf{b}|]$$

$$\leq \frac{1}{\beta^2\pi(A)^2}[\frac{2\pi(A)}{cl} + \frac{1}{12}\beta^2\pi(A)^2]$$

$$= \frac{1}{4}.$$

The theorem follows from Lemma 6.1 of [JVV]. ∎

Disregarding constants, the table of costs is this:

Algorithm	Cost
SP	$\frac{1}{\epsilon}\log\frac{1}{\delta}\log\frac{1}{\pi(s)}\frac{1}{\beta^2\pi(A)}$
AP+ST	$\frac{1}{\epsilon}\log\frac{1}{\delta}(\log\frac{1}{\pi(s)} + \frac{1}{\beta^2\pi(A)})$
AP	$\frac{1}{\epsilon}(\log\frac{1}{\pi(s)} + \log\frac{1}{\delta}\frac{\nu}{\beta^2\pi(A)^2})$

Table of Costs

Note that AP+ST is always better than SP. Now consider the case of uniform G. Aldous' procedure alone is at least as good as the standard procedure in the range $1 \geq \pi(A) \geq 1/\log\frac{1}{\pi(s)}$ and better than the standard procedure when $\pi(A) >> 1/\log\frac{1}{\pi(s)}$. $\pi(A) = \Omega(1)$ holds in the Dyer, Frieze, Kannan algorithm for computing the volumes of convex bodies [DFK]. $\pi(A) \approx 1/\log\frac{1}{\pi(s)}$ holds in Broder's algorithm for counting perfect matching in a dense bipartite graph [Br]. Finally, for $\pi(A) = \Omega(1)$, AP alone is as good as or slightly better (up to a factor of $\log\frac{1}{\delta}$) than AP+ST.

Remarks. (i) The cost of Aldous' procedure "should" depend inversely on $\pi(A)$ instead of $\pi(A)^2$. This in fact is what Chernoff's bound gives in the independent case

and it is not contradicted by the Chebyshev-type theorems [Ald87, LS]. If this could be proven, then for uniform G it would make AP alone better than AP+ST. With regard to the relevance of nonuniformity, it is possible that the cost of AP in reality depends only on $\frac{1}{\epsilon}$, and not on ν. With present techniques we have been unable to make the necessary improvements to Theorem 1, but we do not rule them out.

(ii) The results of this section all have analogues for estimating the expectation $\mathrm{E}f$ of a nonnegative function f on the vertices of G. The running times for all procedures depend on $\frac{\|f\|_\infty}{\mathrm{E}f}$ instead of $\frac{1}{\pi(A)}$.

Non-delayed samples. Suppose instead of generating a random initial point we had begun sampling immediately from s. Setting $k = 0$ in Theorem 4 yields $l = \frac{20\nu}{\epsilon}\frac{1}{\beta^2\pi(A)^2}(\log\frac{1}{\pi(s)} + \log\frac{1}{\delta})$. An upper bound on $\log\frac{1}{\pi(s)}$ is $\log|G| + \log\nu$, which must be polynomial in the data. This shows that non-delayed samples give good estimates in polynomial time, as long as $\pi(A)$ is not too small and ν is not too large. This issue was raised in an example [Ald87, Example 4.2] of estimating the expectation of a function with exponentially small support, which amounts to the same thing as $\pi(A)$ being too small.

Measures of cost. It can be argued that SP has the advantage of taking only a small fraction (around ϵ) of the number of sample points of either AP or AP+ST. We have not considered the number of sample points in our measure of cost because in the cases we know of the cost of sampling a point is dominated by the cost of taking one step of the random walk. For example, in the case of the random walk on matchings treated below, to determine the transition probability from one matching to another it is necessary to know whether the number of edges is going up or down by one, or staying the same. Therefore, it adds only a constant cost to keep track of the size of the current matching (see [SJ]). The same sort of justification holds for the case of

the Ising model considered below. Computing the value of f at a vertex is no more difficult than computing the next transition probability (see [JS91]). Situations may yet arise of having to estimate Ef for complicated f, where the number of sample points will be an important part of the cost of an algorithm.

Improving Two Algorithms

In this section we discuss two random walks used in combinatorial applications for which the current algorithms use the standard sampling procedure (SP). In both cases we provide algorithms that are faster than the current ones using Aldous' procedure together with the statistical technique of [JVV] (AP+ST), and we compare the running times. In broad outline, each of the current algorithms uses a complex random walk several times with different edge weights each time. Our algorithms do not alter the basic random walks, but in the case of the all-matchings random walk, our algorithm introduces a variation in the method of choosing edge weights. In each case we summarize the problem and briefly outline the current algorithm. We then indicate the differences between our algorithm and the existing one. We urge the reader to consult the stated references for full details.

The all-matchings random walk for counting perfect matchings

The problem is to approximate the number of perfect matchings in a graph H on $2n$ vertices. A *matching* is a subset of $E(H)$ such that no two edges share a common endpoint. A *perfect matching* is a matching that contains n edges. Let M_k denote the set of matchings of size k in H, and let $m_k = |M_k|$. A upper bound q on the ratio $\frac{m_{n-1}}{m_n}$ is assumed to be known. In the random walk of Sinclair and Jerrum [SJ] the vertex set of G is the set of all matchings of H. Edges in G correspond to the addition, deletion, or exchange of an edge for another in a given matching of H. The weights

of these transitions are set according to a parameter c so that for every matching M, $\pi(M) \propto c^{|M|}$. The algorithm of [SJ] contains $n-1$ stages, each using a different value of c; stage k approximates the ratio $\frac{m_{k+1}}{m_k}$. The algorithm multiplies the ratios together to obtain an estimate of m_n.

We now reproduce Figure 2 of [SJ], which gives the algorithm of Jerrum and Sinclair for approximating the number of perfect matchings of H to within a ratio of $1+\alpha$ with probability at least 3/4:

(1) if $m_n = 0$ **then halt** with output 0

else begin

(2) $c := |E(H)|^{-1}$; $\Pi := |E(H)|$;

for $k := 1$ **to** $n-1$ **do begin**

(3) if $c > 2q$ or $c < (2|E(H)|)^{-1}$ **then halt** with output 0

else begin

(4) Make a call to SP with $\beta = O(\alpha/n)$ and $\delta = O(1/n)$, using the random walk on G with weight parameter c;

(5) Let $\tilde{p}_k$ and $\tilde{p}_{k+1}$ be the estimates of $\pi(M_k)$ and $\pi(M_{k+1})$ obtained by the call to SP.

(6) if $\tilde{p}_k = 0$ or $\tilde{p}_{k+1} = 0$ **then halt** with output 0

(7) else begin $c := c\tilde{p}_k/\tilde{p}_{k+1}$; $\Pi := \Pi/c$ **end**

end

end;

(8) halt with output Π

end

Our modifications improve the running time from $O(q^3n^5 \log^2 n)$ (implicit in [SJ]) to $O(q^2n^4 \log n)$ by means of the following:

1. We eliminate step (3).

2. For the call to SP in step (4) of the algorithm of [SJ] we substitute a call to AP+ST with the same error tolerances.

3. Between steps (6) and (7) of their algorithm, we add this step: **else if** $\tilde{p}_{k+1} \leq \frac{1}{3n}$, **then** $c := 2c$; **go to (4)**.

4. We attach a clock to the algorithm with the instruction that **if** the time ever exceeds $O(q^2n^4 \log n)$, **then halt** with output 0.

Analysis. Change 1. does not improve the running time. We justify change 4. by analyzing changes 1. and 2. in turn. An upper bound on $\log \frac{1}{\pi_{\min}}$ is $4n \log n$. Referring to the Table of Costs, this is a multiplicative factor in the running time of the algorithm of [SJ]. In the running time of the modified algorithm this factor is dominated by $\frac{1}{\beta^2} \geq \frac{n^2}{\alpha^2}$, since $\tilde{p}_k$ is within $(1 + \frac{\alpha}{4n})$ of the actual value. Therefore, change 1. saves a factor of $4n \log n$.

We now show that change 2. saves a factor of $\Omega(q)$. The method of setting the parameter c in stage k of the algorithm of [SJ] ensures that $\pi(M_k) > \frac{1}{n}$ and that $\frac{\pi(M_{k+1})}{\pi(M_k)} \geq \frac{1}{q}$. We leave the $\frac{1}{n}$ intact and at least double $\frac{\pi(M_{k+1})}{\pi(M_k)}$ each time our algorithm doubles c. This eventually improves the lower bound from $\frac{1}{q}$ to $1/3$. The running time of the random walk of [SJ] is directly proportional to this lower bound, since it is playing the role of $\pi(A)$ in the Table of Costs. Since c never decreases, this doubling of its value in our algorithm can happen only $O(\log q)$ times, so the process at most doubles the running time of the whole algorithm.

The total savings of our algorithm is therefore $\Omega(qn \log n)$.

THE SUBGRAPHS RANDOM WALK FOR THE ISING MODEL

This example will necessarily be sketchier than the last. However, our improvement of the algorithm of Jerrum and Sinclair [JS91] involves a straighforward substitution of AP+ST for SP in each random walk stage of their algorithm. Let H be a graph on n vertices and m edges with weight λ_{ij} on edge (i,j). A certain function $Z'(\mu)$ must be estimated which is a sum over all subgraphs X of H of a function of X involving the λ_{ij} and an external parameter μ. Another function $f = f_{\mu'}(X)$ is defined for which $\mathrm{E}f = \frac{Z'(\mu')}{Z'(\mu)}$. It turns out that $Z'(1)$ is polynomial-time computable, so that approximating $Z'(\mu)$ is reduced to approximating $\mathrm{E}f$ for n successive values of μ and μ' and multiplying together the ratios so obtained.

Theorem 2 and Lemmas 3 and 4 of [JS91] give a bound of $\mathrm{O}(m^3 n^{11})$ random walk steps to estimate n successive values of $\mathrm{E}f$ using SP. Referring again to our Table of Costs, the upper bound on $\log \frac{1}{\pi(s)}$ given by the proof of Theorem 2 of [JS91] is m. We are able to eliminate this factor by a straightforward conversion from SP to AP+ST. The modified algorithm therefore has running time $\mathrm{O}(m^2 n^{11})$.

Remark. Another factor of $O(n)$ can probably be removed from the running times of both of these algorithms by adapting an idea of Dyer and Frieze [DF, Section 4.1]. Both of these algorithms work in n stages. We have been ensuring a multiplicative error of $1 + O(\frac{\epsilon}{n})$ at each stage. The running time is proportional to $O(\frac{n^2}{\epsilon})$. Since the results of each stage are multiplied together and are independent, it should suffice to ensure an error of only $e^{\frac{\epsilon}{\sqrt{n}}}$ and apply the central limit theorem to the cumulative error exponent.

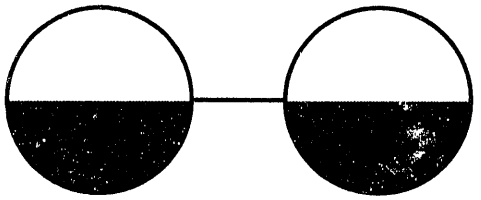

Figure 2.1: A barbell graph with high entropy (A is shaded).

2.4 Further Work

Philosophical Remarks on Sampling

In [Ald90], Aldous makes the following remarks, translated to our terminology:

> If τ is sufficiently large then x_τ has approximately the stationary distribution π independent of x_0. The sequence $x_\tau, x_{2\tau}, \ldots, x_{l\tau}$ is approximately independent, and so the mean-square error of the average $\frac{1}{l}(\chi_A(x_\tau)+\cdots+\chi_A(x_{l\tau}))$ is about $\frac{\pi(A)}{l}$. The average itself is roughly the standard sampling procedure. More observations cannot hurt, so for $n = l\tau$ the mean-square error of t_n is at most $\frac{\pi(A)}{l}$.

More observations cannot hurt, but our business has been quantifying how much they help.

An interesting theoretical question is how much they help as a function of τ. Intuitively, when τ is small, barring periodicities in the Markov chain, taking every τ^{th} sample point should not be much less efficient than taking every sample point, since $x_{j\tau+1}, x_{j\tau+2}, \ldots, x_{(j+1)\tau}$ are very dependent. Figure 2.1 shows a "barbell" graph. The two circles represent large cliques connected by an edge. The shaded region represents the set A. Clearly sampling every other point is not much less efficient than sampling every point, but sampling every $\frac{1}{\epsilon}^{th}$ point is much less efficient.

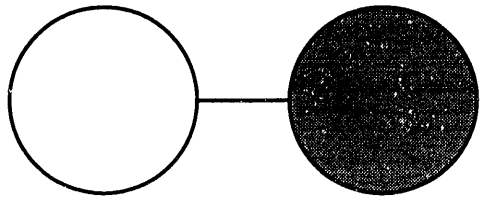

Figure 2.2: A barbell graph with low entropy (A is shaded).

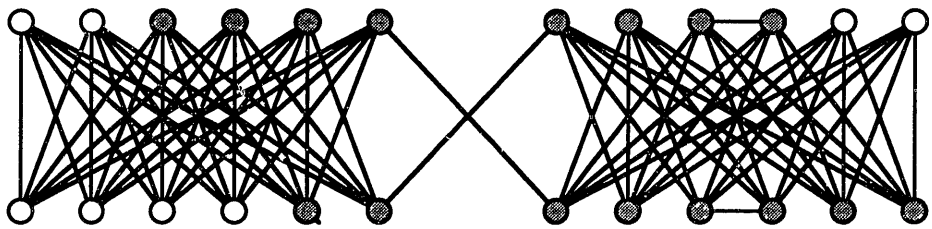

Figure 2.3: An example where the sample average is a bad estimate of $\pi(A)$ (A is shaded).

The accuracy gained by sampling every point seems to depend on the entropy of the random walk (as a 0-1 source), as well as the eigenvalue gap ϵ. It would be interesting to clarify this relationship. Figure 2.2 shows the same barbell graph, but the set A has all been moved into one clique. Now it is not much more efficient to sample every point than to sample every $\frac{1}{\epsilon}^{th}$ point.

Although it cannot hurt to sample every point, the sample average may not be the best estimate of $\pi(A)$. This odd phenomenon was observed by Winkler [Win], who gave the counterexample shown in Figure 2.3. The graph G is regular and nearly bipartite. The set A is represented in the proportion $\pi(A)$ in both the top and bottom nodes. After a short random walk, the sampled average from the top points (most likely every other point) will give a better estimate than the sample average of every point.

It would be interesting to characterize those graphs for which the sample average of every point is the best estimate of $\pi(A)$, or those classes of graphs for which prior knowledge about the class would not make it advantageous to use any estimate other than the sample average. We conjecture that an important property of such graphs would be that the second largest eigenvalue is also the second largest in absolute value. The graph G in the figure has the property that the random walk on G^2 is not an expander, because there is a negative eigenvalue very close to -1.

Open Questions

There are several ways to improve our Chernoff-type bound which seem plausible:

1. Remove the dependence of the bound on nonuniformity.

2. Introduce a factor of $\frac{1}{\pi(A)}$ into the exponent of the bound, by analogy to the independent case. The idea here is to show that the cost of Aldous' procedure for estimating $\pi(A)$ depends inversely on $\pi(A)$ instead of $\pi(A)^2$.

3. Generalize to some class of non-reversible Markov chains, perhaps using the notion of μ-conductance [LS] or Fill's notion of reversiblization [Fi]. As Aldous points out in [Ald88], it won't do simply to look at the eigenvalues in the nonreversible case. He presents a large class of Markov chains with no obvious pathologies, for which ϵ is $\Omega(1)$ but for which variance of t_n is $\omega(\frac{1}{n})$.

   There is also a perturbation theory for matrices which relies on the separation of the invariant spaces of the matrix. Even in the non-reversible case if π is uniform then it is orthogonal to the other invariant spaces. This fact may prove fruitful.

4. Give a lower bound on the probability of deviation (see Deuschel and Stroock [DS] for the standard approaches to large deviation lower bounds.)

It would be nice to be able to estimate the entropy of a random walk on a weighted graph. We have mentioned the potential application to discrimination of hidden Markov chains. The immediate difficulty is not the lack of a nice formula for entropy in this case, but the problem of estimating the expected value of a function on the edges of G. The random walk on G induces a Markov chain on the edges of G. The induced Markov chain is not time-reversible, so the Chernoff bound does not apply as it stands.

Ultimately we would like identify those structural properties of a hidden Markov chain that make it possible to estimate the entropy of the black box efficiently.

Chapter 3

Hidden Markov Chains

3.1 Introduction

In this chapter we present some old and new results on oracle problems for hidden Markov chains. We also introduce some new problems motivated by similar research in the theory of automata, and we raise several open questions.

The basic notion of this chapter is that of a black box emitting 0's and 1's from which we would like to reconstruct the internal workings of the black box or construct our own model whose behavior mimics that of the black box. This motivates the following definition.

Definition 1 *An m-state* hidden Markov chain (hmc) *is a stationary ergodic discrete-time Markov chain defined by an $m \times m$ transition matrix M on a state space $Q = \{q_1, \ldots, q_m\}$, together with a two-coloring $\chi : Q \to \{0, 1\}$ which partitions Q into* 0-states *and* 1-states.

We will identify an hmc by a triple (Q, χ, M), or, where no confusion is possible,

simply M (or L or N). The next section contains a careful exposition of the terms "Markov chain", "stationary" and "ergodic" in their present context. For now we let it suffice to say that an hmc generates a random sequence $y_1, y_2, \ldots$ of 0's and 1's, which we denote collectively $\{y_i\}^M$; and that it makes sense to speak of the long-term behavior of $\{y_i\}^M$. Thus $\{y_i\}^M$ is our formalization of the black box.

The hmc model is a simplification of the hidden Markov model (hmm) found in research on speech recognition. For example, in Rabiner [LRS], χ is replaced by a probabilistic function onto some alphabet. In the case of a finite alphabet, we lose no generality by considering hmc's, since an hmm can be converted to an hmc by increasing the number of states.

The hmc model also appears as one form of the *Markov source* model (see Ziv and Merhav [ZM92] for various forms of this model) in the literature on data compression and information theory. There various empirical problems are considered, including hypothesis testing [Gu], discrimination [Zi, WZ], and inference [MZ89].

The ergodicity and stationarity properties of the hmc are essential, since we are interested in long-term behavior. In applications of hmm's to speech recognition, the problem is often to infer from short-term behavior of a Markov chain started in a particular state. Some of our results will apply to this setting as well. Tzeng [Tz92b] has used the term "hidden Markov chain" for a unifilar nonergodic Markov chain model which starts in a particular state, a model which is motivated by the theory of learning finite automata.

Rudich [Ru] studied unifilar ergodic Markov chains in the context of asymptotic inference of a Markov chain from an infinite stream of data. Evans *et al.* [ERV] consider asymptotic inference of countable classes of arbitrary conditional probability

distributions on infinite sequences.

Oracle Problems

Inference of hmc's divides into two kinds of problems according to the kind of information received by the algorithm. In *empirical* problems, the hmc generates a finite sequence s of 0's and 1's and presents this sequence to the algorithm. In *oracle* problems we idealize the notion of empirically estimated probabilities by considering algorithms that query a *probability oracle* which gives the exact long-term probability of any requested binary string w.

We now introduce the main oracle problems of inference, discrimination, and equivalence. We discuss their relative difficulty, describe our results, and discuss related problems. The input to an algorithm may include an *unknown* hmc M, which means that the algorithm has access only to the process $\{y_i\}^M$. If the input includes a *known* hmc, then the algorithm knows the transition matrix.

We also refer to *pseudo hidden Markov chains (pseudo-hmc's).* Informally, pseudo-hmc's are like hmc's except that negative transition probabilities are allowed in the transition matrix. For the moment we let it suffice to say that a pseudo-hmc also defines a process $\{y_i\}$ with a well-defined notion of the long-term probabilities of finite strings. Also, an hmc is a special case of a pseudo-hmc. Precise definitions and justifications of pseudo-hmc's and the probability oracle appear in Sections 3.3 and 3.4.

For the following problems, let us assume a fixed unknown hmc M which generates the random 0-1 process $\{y_i\}^M$:

Inference Given an upper bound m on the size of M, construct a pseudo-hmc N

for which $\{y_i\}^N$ is equal to $\{y_i\}^M$ in distribution.

Discrimination Given a known pseudo-hmc N and an upper bound m on the sizes of N and M, decide whether $\{y_i\}^N$ is equal to $\{y_i\}^M$ in distribution.

Equivalence Given two known hmc's L and N of size at most m, decide whether $\{y_i\}^L$ is equal to $\{y_i\}^N$ in distribution (L is *equivalent* to N).

We first remark that two unequal hmc's can be equivalent. Theorems 7 and 8 in Section 3.3 completely characterize equivalence of two pseudo-hmc's. These results are due to Gilbert [Gi] and Paz [P]. It will follow easily from this characterization that generally there are many hmc's and pseudo-hmc's equivalent to a given hmc or pseudo-hmc. An example of an hmc and a pseudo-hmc that are equivalent is shown in Section 3.3, Figures 3.2 and 3.3.

In Section 3.4 we show that equivalence is decidable in polynomial time. We give two algorithms for this. One will be based on an algorithm to find the smallest pseudo-hmc equivalent to a given hmc. This minimal pseudo-hmc can be used as a fingerprint to test equivalence. There is evidence that finding the minimal hmc is hard: in a closely related hmc model which does not include the notions of stationarity and ergodicity, Cobham [Co] has shown the existence of constant size pseudo-hmc's whose smallest equivalent hmc is arbitrarily large. We also mention that minimization and equivalence algorithms for hmc's can be derived from the earlier work of Tzeng [Tz92a] on *probabilistic automata* (pa's), since a known hmc can be converted to a pa with the same behavior.

In Section 3.4 we also solve a hypothesis testing problem for hmc's [Gu] by reducing it to equivalence. The problem is to determine which of two hidden Markov chains accounts for the behavior of a black box presented by an oracle. We give an application

of this to resolving the ambiguity of a Huffman code which was encoded by one of two given Huffman trees [GMR].

Hierarchy and Hardness

The following theorem is an immediate consequence of the existence of a polynomial-time algorithm to decide equivalence of two pseudo-hmc's.

Theorem 6 *Discrimination is reducible to inference in polynomial-time.* ∎

In Section 3.5 we show for each of two restricted classes of hmc's there is no randomized polynomial-time probability oracle algorithm to solve discrimination. By Theorem 6, these results imply the intractibility of inference on each of these classes as well. This is spite of the existence of an NP probability oracle algorithm which follows immediately from Theorem 8. The proofs of intractibility are information-theoretic and constructive, and they do not depend on any assumptions about separation of complexity classes.

We first consider the restricted class of unifilar hmc's in which each transition probability is 0, p, $1-p$, or 1, for some p. For each string w of length m we construct an hmc on $2m+2$ states which behaves like a sequence of fair coin flips on every string not containing w. Each such hmc contains a certain signature state with exponentially small stationary probability and a "long memory". Tzeng [Tz92b] used a very similar construction to show that inference is intractible for a unifilar Markov chain model which does not contain notions of stationarity or ergodicity.

We show that discrimination is still hard when every state is visited frequently. We consider the class of hmc's with uniform stationary distribution. The proof of

intractibility here rests on the results on equivalence and an interesting technique of avergaing two pseudo-hmc's. For each string w of length m we construct an hmc in this class which is "equivalent on strings not containing w" to the corresponding hmc in the above class of unifilar hmc's. To do this, we first find an equivalent pseudo-hmc with uniform stationary distribution using a similarity transformation on the transition matrix. Then we average the pseudo-hmc with a random walk on a complete graph and prove that the averaged hmc has the right properties.

Two Randomized Algorithms

Discrimination is easier on the intersection of these two classes, when $p = 1/2$. In Section 3.6 we give a randomized probability oracle algorithm with one-sided error to distinguish a member of this smaller class from a sequence of fair coin flips. Our analysis of the algorithm borrows heavily from a result of Aleliunas *et al.* [AK*] on random walks. We show that a random long string will have probability zero unless the hmc actually is equivalent to a sequence of fair coin flips. We give some evidence that randomness is needed, by showing that an hmc from this class can be indistinguishable from a sequence of coin flips except on long strings. Specificaly, there is a family of 2^m $(4m + 2)$-state hmc's which behave identically on all strings of length less than m.

We also present partial results on randomized discrimination that follow from Theorem 3 on entropy estimation. We consider unifilar random walks on graphs whose vertices are labelled by a finite alphabet. We give a randomized probability oracle algorithm with one-sided error for discriminating such a model from a sequence of independent labels.

These randomized algorithms use the oracle to simulate the black box. We observe that in order to generate a random sequence it is enough to have access to the probability of an arbitrary string.

Three Problems

There remains the problem of finding a large class of hmc's on which efficient inference is possible. A natural question when is whether a stronger oracle can help. In Section 3.7 we define inference in a teacher-learner setting, where in addition to a probability oracle the algorithm has access to an oracle which can essentially solve discrimination problems. This is a natural extension of the work of Angluin [Ang87] on learning dfa's from membership and equivalence queries. The discrimination oracle is essentially the same as the equivalence oracle of Tzeng [Tz92b], who gives an algorithm with probability and equivalence queries for inference of unifilar nonergodic hidden Markov chains which start in a particular state.

We also define an oracle inference problem in which an algorithm has access to a probabilty oracle but is only required to come up with an approximation to M. It does not follow from the intractiblity of inference for a certain class that this approximate inference problem is also intractible. In fact the hardness results rely on the construction of exponentially many hmc's whose behavior is close to a sequence of independent labels (see Wyner and Ziv [WZ] for similar results on the minimum memory required for classification of Markov sources). If this approximate oracle inference problem is tractible, then it is reasonable to expect empirical inference to be tractible on a class of hmc's which converge quickly to their asymptotic behavior on most output sequences. As we mentioned at the end of Chapter 2, it is still an

open problem to quantify this convergence for a large class of hmc's.

Finally, we define the consistency problem for pseudo-hmc's: Given a set of strings and their long-term probabilities, determine whether an m-state pseudo-hmc exists which assigns the correct probabilities to those strings. This problem is similar to the consistency problem for dfa's, which Gold [Go78] has shown to be NP-complete, and the consistency problems for pa's and Markov chains, which Tzeng [Tz92b] has shown to be NP-complete by reduction from the dfa problem. Because of the ergodicity and stationarity condition in our model, the restriction of our model to dfa's appears to be too impoverished to prove NP-completeness of consistency for hmc's by reduction.

We also propose the restricted version of the consistency problem in which the set of strings must be closed under substrings. The intuition from empirical samples suggests that if the probability of a string is given then the probability of all substrings should be given as well. This is because if a sample output contains many instances of a string w, enough for a reliable estimate of the long-term probability of w, then it also contains many instances of each substring of w. Furthermore, the long-term probability of each substring of w is clearly at least as great as the long-term probability of w itself, and one would expect a finite sample output to give the most reliable estimates for strings of high probability.

We note that the consistency problem differs from the inference problem for oracle algorithms in two ways. The algorithm is passive in that it receives a fixed set of strings that it cannot choose, a condition shared by the empirical inference problem. Also, there is not necessarily a unique correct answer up to equivalence.

3.2 Hmc's

In this section we elaborate on the definition of an hmc given at the start of the chapter. Technically, most of the material of this section is subsumed by the definitions for pseudo-hmc's in the next section; but we risk repetition for ease of presentation.

We start by explaining the terms "stationary" and "ergodic" in the present context. The Markov chain itself is a sequence of random variables $x_0, x_1, \ldots x_n$ taking values in the state set Q. The Markov chain is determined by an $m \times m$ stochastic transition matrix $M = (p_{ij})$ which gives for each pair i and j the probability p_{ij} of moving to state q_j after having entered state q_i, together with a distribution on the state set Q from which x_0 is chosen. Therefore, one can think of x_i as the state entered at time i in a random trajectory. We will identify an hmc by a triple (Q, χ, M), or, where no confusion is possible, simply M (or L or N). Unless specifically stated, in this chapter we will make no assumption about reversibility of our Markov chains.

To ensure that it is meaningful to speak of the long-term behavior of an hmc, we require that the Markov chain be ergodic. In the non-reversible setting an *ergodic* Markov chain is such that for some postive k every entry of M^k is positive. The ij entry of M^k is the probability that a particle which starts in state q_i will find itself in state q_j after k steps. Ergodicity implies that every row of M^k converges to the stationary distribution π, which is the unique distribution satisfying $\pi M = \pi$ [Fe].

The condition that the Markov chain is *stationary* means that x_0 is chosen from the distribution π. Consider any fixed finite sequence of states $q_{j_0}, \ldots, q_{j_k}$. The probability of the event $\{x_i = q_{j_0}, \ldots, x_{i+k} = q_{j_k}\}$, the event that the Markov chain visits these states consecutively in order at time i, is independent of i. This probability

is denoted $\Pr[q_{j_0} \cdots q_{j_k}]$, and it satisfies

$$\Pr[q_{j_0} \cdots q_{j_k}] = \pi(q_{j_0}) \prod_{l=1}^{k} p_{j_{l-1} j_l}. \tag{3.1}$$

Let $x_0, \ldots, x_n$, be the first $n+1$ states entered by the Markov chain. Ergodicity implies that with probability 1, the fraction of numbers i in the range $1 \leq i \leq n-k$ for which the event $\{x_i = q_{j_0}, \ldots, x_{i+k} = q_{j_k}\}$ occurs in $x_1, \ldots, x_n$ approaches $\Pr[q_{j_0} \cdots q_{j_k}]$ as $n \to \infty$. The last chapter dealt with the rate of this convergence in the reversible case. We also discuss the rate of convergence in Chapter 4, where we are concerned with how accurately a finite output sequence represents the long-term behavior of the black box.

Let $y_i = \chi(x_i)$, $i = 1, 2, \ldots$; then $y_1, y_2, \ldots$, is a random sequence of 0's and 1's. This random sequence is not generally a Markov chain; that is, the distribution of y_{i+1} does not only depend on the value of y_i but may depend on the values of earlier y_j. [1] It still makes sense to speak of the long-term behavior of $\{y_i\}$. Because of the ergodicity of the $\{x_i\}$ process, the $\{y_i\}$ process is also ergodic in this sense: as $k \to \infty$ the fraction of occurrences in $y_1, \ldots, y_k$, of any event $\{y_i = \sigma_0, \ldots, y_{i+k} = \sigma_k\}$, where $\sigma_j \in \{0, 1\}$, approaches a limiting value denoted $\mathrm{P}[\sigma_0 \cdots \sigma_k]$. This value depends on the events in the $\{x_i\}$ process making up this event, according to the formula

$$\mathrm{P}[\sigma_0 \cdots \sigma_k] = \sum_{\chi(q_j) = \sigma_j} \Pr[q_{j_0} \cdots q_{j_k}]. \tag{3.2}$$

We call $\mathrm{P}[\sigma_0 \cdots \sigma_k]$ the *probability of seeing* $\sigma_0 \cdots \sigma_k$ *in stationarity*, or simply the *long-term probability of* $\sigma_0 \cdots \sigma_k$. Given two strings $w, v \in \{0, 1\}^*$, we call $\mathrm{P}[v|w] = \frac{\mathrm{P}[wv]}{\mathrm{P}[w]}$ the probability of seeing v after having seen w. Once again the

[1] The $\{y_i\}$ process is a Markov chain if and only if its black box behavior is that of a two-state Markov chain.

rate of convergence to $\mathrm{P}[\sigma_0 \cdots \sigma_k]$ will be important when we consider how accurately a finite sample "represents" the black box, in Chapter 4. We will write $\mathrm{P}^M[w]$ when we wish to emphasize the role of the hmc M.

3.3 Pseudo-Hmc's

Definitions and Notation

Our model of computation is a machine which can perform a precise arithmetic operation on rational numbers in one time step. All numbers are assumed to be rational.

With a slight abuse of terminology we call a matrix $M = p_{ij}$ that satisfies, for all i, $\sum_{j=1}^{m} p_{ij} = 1$, "stochastic" even if it contains negative entries. We will also speak of a "distribution" on Q as any vector $d = (d_1, \ldots, d_m)$ of rational "probabilities" (possibly negative) satisfying $\sum_{i=1}^{m} d_i = 1$.

Definition 2 *An m-state* pseudo hidden Markov chain (pseudo-hmc) *consists of a state space $Q = \{q_1, \ldots, q_m\}$, a two-coloring $\chi : Q \to \{0, 1\}$ which partitions Q into* 0-states *and* 1-states, *an $m \times m$ stochastic matrix $M = (p_{ij})$* (the transition matrix) *with rational entries, and a* stationary distribution π *which satisfies $\pi M = \pi$.*

As with hmc's we identify a pseudo-hmc by the triple (Q, χ, M), or simply M.

We would like to carry the notion of ergodicity over to pseudo-hmc's in a useful way. In the nonnegative case M is ergodic if and only if 1 is a simple eigenvalue and every other eigenvalue is smaller than 1 in absolute value [Se]. But there is also a geometric characterization, namely that the underlying directed graph of nonzero transitions is strongly connected and aperiodic (the l.c.d. of the lengths of all cycles is

1). This geometric characterization of ergodicity does not carry over to pseudo-hmc's. For the present we allow an arbitrary choice of π when there is more than one vector to choose, rather than insisting in the definition that 1 be a simple eigenvalue.

Open Question 1 *Find a useful geometric definition of ergodicity for pseudo-hmc's.*

We return to this question in the discussion following Example 1.

The Probabilities of Strings

Having left behind nonnegative matrices, we cannot describe a trial run of a pseudo-hmc as a physical process of visiting a sequence of states with a certain probability. Still, formally a pseudo-hmc $M = (p_{ij})$ starting "in stationarity" assigns a possibly negative probability to each sequence of states by the same formula as before:

$$\Pr[q_{j_0} \cdots q_{j_k}] = \pi(q_{j_0}) \prod_{l=1}^{k} p_{j_{l-1} j_l}. \tag{3.3}$$

Let I be the $m \times m$ identity matrix. Let $I_1 = \mathrm{diag}[\chi(q_i)]$ and let $I_0 = I - I_1$. Given a pseudo-hmc with transition matrix M, let $M_0 = MI_0$ be the result of zeroing out the columns of M corresponding to 0-states. Likewise, let $M_1 = MI_1$, $\pi_0 = \pi I_0$, and $\pi_1 = \pi I_1$.

Let $w = w_1 w_2 \cdots w_k$ be a color sequence, and let $\mathbf{d}$ be a distribution. Define the function $\mathrm{P}_{\mathbf{d}}[\cdot]$ on strings by

$$\mathrm{P}_{\mathbf{d}}[w] = \mathbf{d} M_{w_1} M_{w_2} \cdots M_{w_k} \mathbf{1}, \tag{3.4}$$

where $\mathbf{1}$ is the m-vector $(11 \cdots 1)$. In particular, let δ_i be the point mass at state q_i. We call P_{δ_i} the *state function* for q_i and usually denote it simply P_{q_i}. To stay

consistent with the notation for nonnegative hmc's we write $\mathrm{P}[w]$ for $\mathrm{P}_\pi[w]$, the "long-term probability" of w. For $\sigma \in \{0,1\}$ we let $\pi(\sigma)$ denote $\pi(\chi^{-1}(\sigma))$, which is also equal to $\mathrm{P}^M[\sigma]$.

Whenever $\mathrm{P}^M[w] \neq 0$, we define the *distribution reached after seeing* w, $\mathbf{d}_w$, by

$$\mathbf{d}_w = \frac{\pi M_{w_1} \cdots M_{w_k}}{\mathrm{P}^M[w]}, \tag{3.5}$$

and we say that $\mathbf{d}_w$ is *reachable*. We call $\mathrm{P}_w[\cdot] = \mathrm{P}^M_{\mathbf{d}_w}[\cdot]$ the *string function* at w. The usual formula for conditional probability holds: $\mathrm{P}_w[v] = \frac{\mathrm{P}[wv]}{\mathrm{P}[w]}$. When $\mathrm{P}^M[w] = 0$ we set $\mathbf{d}_w = \mathbf{0}$, where $\mathbf{0} = (00\cdots0)$.

We define the *shifted* functions as follows: Let $\sigma \in \{0,1\}$ and let $\mathbf{d}$ be a distribution supported on $\chi^{-1}(\sigma)$. Given a string w and a label $\tau \in \{0,1\}^*$, put

$$\mathrm{R}_{\mathbf{d}}[\tau w] = \begin{cases} \mathrm{P}_{\mathbf{d}}[w] & \text{if } \tau = \sigma \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

When $\mathbf{d} = \delta_q$ for some state q we call $\mathrm{R}_{\mathbf{d}}[\cdot]$ a *shifted state function*, and when $\mathbf{d} = \mathbf{d}_v$ for some v we call $\mathrm{R}_{\mathbf{d}}[\cdot]$ a *shifted string function.*

By defining the way in which pseudo-hmc's assign probabilities to strings, we have made it possible to allow an inference algorithm to construct a black box containing a pseudo-hmc to mimic the behavior of a black box containing an hmc.

Certain pathologies occur for pseudo-hmc's which defy our intuition about probabilities. For a pseudo-hmc it can happen that $\mathrm{P}[w] = 0$ but $\mathbf{d}_w \neq \mathbf{0}$; furthermore, it is possible to construct a pseudo-hmc M and to choose strings w and v for which $\mathrm{P}^M[w] = 0$ but $\mathrm{P}^M[wv] \neq 0$. Figure 3.1 gives one such example. Note that in this example there are certain strings, for example 11000, which have negative probability.

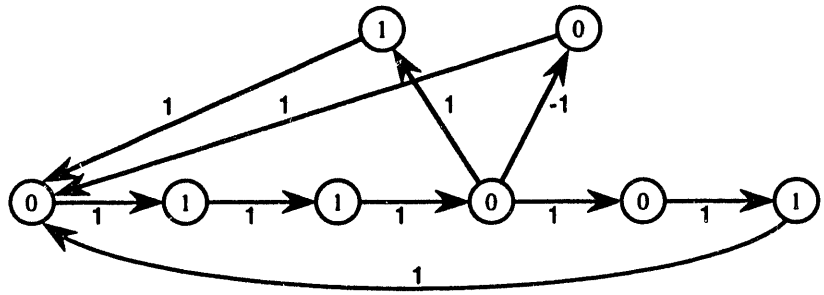

Figure 3.1: A pseudo-hmc for which P[1100] = 0 but P[11001] = 1/6.

Open Question 2 *Characterize the class of pseudo-hmc's M for which $P^M[\cdot]$ is a nonnegative function. [For all such M, $P^M[\cdot]$ defines a (nonnegative) probability measure on infinite binary sequences.]*

Let us give the name *Phmc* to an element of the class of Open Question 2. For any Phmc M it is always the case that $\mathrm{P}^M[w] = 0$ implies $\mathrm{P}^M[wv] = 0$ for any two strings w and v. The class of Phmc's contains all hmc's, but it also contains pseudo-hmc's which are not hmc's, as the following example shows.

Example 1 Let M be the pseudo-hmc with two 0-states and one 1-state shown in Figure 3.2. There are only four reachable distributions on the 0-states; they are $\mathbf{d}_0, \mathbf{d}_{00}, \mathbf{d}_{10}$, and $\mathbf{d}_{100}$. In order to show that $\mathrm{P}[w] \geq 0$ for all w, it suffices to check that $\mathrm{P}_{10}[1] = 0$ and $\mathrm{P}_{100}[1] = 1/2$.

It is not hard to see that M is equivalent to the hmc shown in Figure 3.3, which generates symbols by flipping a fair coin: 00 for heads and 1 for tails.

Since a Phmc can generate a bona fide random process $\{y_i\}$ it is reasonable to ask when this process is ergodic as an information source. We may reiterate the theme of Open Question 1: what structural conditions on the Phmc are sufficient to ensure that $\{y_i\}$ is an ergodic process? See also the remarks following Theorems 7 and 8.

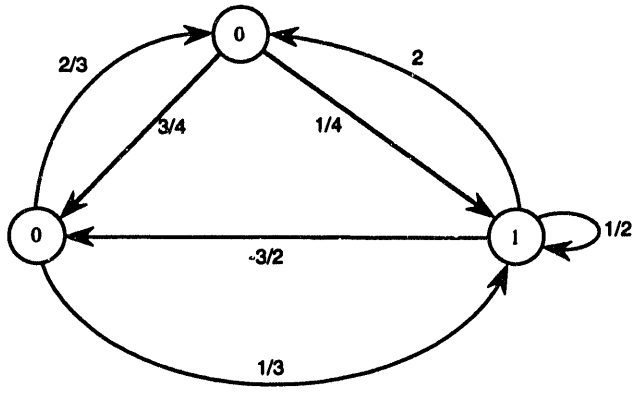

Figure 3.2: A pseudo-hmc which assigns nonnegative probability to every string.

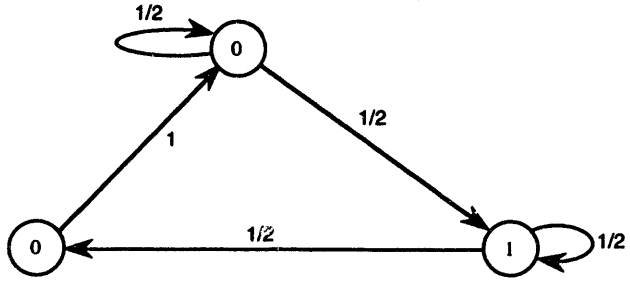

Figure 3.3: An hmc which flips a coin and stutters on 0.

Equivalence of Pseudo-Hmc's

In this subsection we characterize the equivalence of two pseudo-hmc's of possibly different sizes or colorings. This is via reduction to the minimal pseudo-hmc, which is unique up to similarity of the transition matrix. We also give a sufficient condition for the equivalence of two pseudo-hmc's with the same state set S and coloring χ. The characterization of equivalence trivially gives a lower bound on the size of the smallest hmc equivalent to a given hmc M. These results are originally due to Gilbert [Gi]. We give our own proofs to prepare for the next section, in which we develop efficient algorithms for equivalence and minimization with oracle queries.

Definition 3 *Let (Q, χ, M) and (Q', χ', M') be pseudo-hmc's with stationary distributions π and π', respectively. We write $(Q, \chi, M) \equiv (Q', \chi', M')$, or simply $M \equiv M'$,*

if for all color sequences w, $P^M[w] = P^M[w]$. If this is the case we say that M and M' are equivalent,

Definition 4 *Let M and M' be the transition matrices for two pseudo-hmc's on the same state space and coloring. We say that M and M' are* block similar (with respect to χ) *if there is an invertible matrix $S = (s_{ij})$ satisfying $M' = S^{-1}MS$ and $s_{ij} = 0$ whenever $\chi(i) \neq \chi(j)$. We write $M \sim_\chi M'$. S is called a* block similarity matrix.

Theorem 7 *Let M and M' be the transition matrices for two pseudo-hmc's on the same state space and coloring. Suppose that M and M' are block similar, satisfying $M' = S^{-1}MS$. Suppose also that π and π' are the stationary distributions of M and M' and that they satisfy $\pi' = \pi S$. Then $M \equiv M'$.*

In particular, if M and M' are hmc's and $M \sim_\chi M'$, then $M \equiv M'$.

First a technical result.

Lemma 3 *Let A and B be $m \times m$ matrices. Suppose A is stochastic and B is invertible. Let A_i and B_i denote the i^{th} row vectors of A and B, respectively. Let $b_i = B_i \cdot \vec{1}$ be the i^{th} row sum of B. Then $B^{-1}AB$ is stochastic if and only if $A\vec{b} = \vec{b}$. In particular, if A is ergodic then B and B^{-1} have constant row sums.*

Proof. Let a_{ij} denote the typical entry of A. Then

$$\begin{aligned} B^{-1}AB \text{ is stochastic} &\Leftrightarrow \forall i \; B_i B^{-1} AB\vec{1} = B_i \cdot 1 = b_i \\ &\Leftrightarrow \forall i \; b_i = A_i B\vec{1} = \sum_{j=1}^{m} a_{ij} B_j \cdot \vec{1} \\ &\Leftrightarrow A\vec{b} = \vec{b}. \end{aligned}$$

If A is ergodic then $\vec{b} = c\vec{1}$ for some constant c. Hence, $\vec{1}$ is a right eigenvector for both B and B^{-1} and they both have constant row sums. ∎

Proof of theorem 7. By Lemma 3 we can take S to be stochastic. Let $w = w_1 \cdots w_n$ be any color sequence. We have

$$\begin{aligned} \mathrm{P}^{M'}[w] = \pi' M'_{w_1} \cdots M'_{w_n} \vec{1} &= \pi S S^{-1} M (S I_{w_1} S^{-1}) \cdots M (S I_{w_n} S^{-1}) S \vec{1} \\ &= \pi M I_{w_1} \cdots M I_{w_n} \vec{1} \\ &= \mathrm{P}^{M}[w]. \end{aligned}$$

If M and M' are hmc's then by ergodicity we automatically have $\pi' = \pi S$, and the theorem follows. ∎

Example 2 *We can use Theorem 7 to prove that the pseudo-hmc M in Figure 3.2, is equivalent to the hmc N in Figure 3.3. The transition matrices are*

$$N = \begin{bmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \end{bmatrix} \quad \textit{and} \quad M = \begin{bmatrix} 0 & 3/4 & 1/4 \\ 2/3 & 0 & 1/3 \\ 2 & -3/2 & 1/2 \end{bmatrix}, \tag{3.7}$$

and the block similarity matrix S is

$$S = \begin{bmatrix} -2 & 3 & 0 \\ 4 & -3 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \tag{3.8}$$

from which we see that $M = S^{-1} N S$.

We now pursue a converse to Theorem 7. The criterion for equivalence of two pseudo-hmc's will be that they both reduce to the same minimal pseudo-hmc.

Definition 5 *A pseudo-hmc is* minimal *if there is no equivalent pseudo-hmc with fewer states.*

The key to finding the minimal equivalent pseudo-hmc is to observe that the dimension of the space of state functions is finite (actually, at most m).

Lemma 4 *Fix an m-state pseudo-hmc M with stationary distribution π. For every string w such that $P(w) \neq 0$ the string function P_w is in the affine hyperplane $H = \{\sum_{i=1}^{m} \alpha_i P_{q_i} : \sum_{i=1}^{m} \alpha_i = 1\}$.*

Proof. Let $\alpha_i = \frac{(\mathbf{d}_w)_i}{P(w)}$. ∎

Note that $(\mathbf{d}_{x0})_i = 0$ whenever $\chi(i) = 1$ and $(\mathbf{d}_{x1})_i = 0$ whenever $\chi(i) = 0$. Therefore, the 0-string functions P_{x0} are all contained in

$$H_0 = \{\sum_{i=1}^{m} \alpha_i P_{q_i} : \chi(q_i) = 0, \sum_{i=1}^{m} \alpha_i = 1\} \tag{3.9}$$

and the 1-string functions P_{x1} are all contained in

$$H_1 = \{\sum_{i=1}^{m} \beta_i P_{q_i} : \chi(q_i) = 1, \sum_{i=1}^{m} \beta_i = 1\}. \tag{3.10}$$

Let m_0 and m_1 be the dimensions of the vector spaces spanned by the 0-string functions and 1-string functions, respectively. Note that $m_0, m_1 \leq \dim(H_i) + 1 \leq m$.

Theorem 8 *Suppose M is a pseudo-hmc. Let m_0 and m_1 be defined as above. Then there is a minimal pseudo-hmc M' with m_0 0-states and m_1 1-states such that $M' \equiv M$.*

Proof. The minimality of any pseudo-hmc with m_0 0-states and m_1 1-states follows from lemma 4 and the discussion following it. To construct such a pseudo-hmc find

a basis $P_{v_1}, \ldots, P_{v_{m_0}}$ of 0-string functions and a basis $P_{w_1}, \ldots, P_{w_{m_1}}$ of 1-string functions, such that $v_1 = 0$ and $w_1 = 1$. We will construct an equivalent pseudo-hmc with transition matrix M' whose states are $q_1, \ldots, q_{m_0}$, $r_1, \ldots, r_{m_1}$. We will require that its state functions $\mathrm{P}^{M'}_{q_i}[\cdot]$ and $\mathrm{P}^{M'}_{r_i}[\cdot]$ be equal to the corresponding string functions of M, $\mathrm{P}^{M}_{v_i}[\cdot]$ and $\mathrm{P}^{M}_{w_i}[\cdot]$, for each i.

Recall the definitions of the shifted state functions. Let $\mathbf{P}$ and $\mathbf{R}$ be matrices whose columns are indexed by $\{0,1\}^*$ such that the w^{th} column of $\mathbf{P}$ is $(\mathrm{P}^{M'}_{q_1}[w], \ldots, \mathrm{P}^{M'}_{q_{m_0}}[w], \mathrm{P}^{M'}_{r_1}[w], \ldots, \mathrm{P}^{M'}_{r_{m_1}}[w])^T$, and the w^{th} column of $\mathbf{R}$ is $(\mathrm{R}^{M'}_{q_1}[w], \ldots, \mathrm{R}^{M'}_{q_{m_0}}[w], \mathrm{R}^{M'}_{r_1}[w], \ldots, \mathrm{R}^{M'}_{r_{m_1}}[w])^T$. Let $\mathbf{P}(w)$ and $\mathbf{R}(w)$ denote the w^{th} columns of $\mathbf{P}$ and $\mathbf{R}$. The matrix M' must be a solution to the equation

$$\mathbf{P} = M'\mathbf{R} \tag{3.11}$$

By the linear independence of the $\mathrm{P}^{M}_{v_i}[\cdot]$ and the $\mathrm{P}^{M}_{w_i}[\cdot]$ the matrix $\mathbf{R}$ has rank m_0+m_1. Therefore, there is a unique solution M' to Equation (3.11). By inspection, $M'\mathbf{1} = M'(\mathbf{R}(0) + \mathbf{R}(1)) = \mathbf{P}(0) + \mathbf{P}(1)) = \mathbf{1}$; therefore, M' is stochastic, and M' is a pseudo-hmc.

It remains to show that $M' \equiv M$. Since $\mathrm{P}^{M}[\cdot] = \pi(0)\mathrm{P}^{M}_{0}[\cdot] + \pi(1)\mathrm{P}^{M}_{1}[\cdot] = \pi(0)\mathrm{R}^{M}_{0}[\cdot] + \pi(1)\mathrm{R}^{M}_{1}[\cdot]$, the vector $\pi' = \pi(1)\delta_{q_1} + \pi(0)\delta_{r_1}$ satisfies $\pi'M'\mathbf{R} = \pi'\mathbf{R}$. But $\mathbf{R}$ has full row rank. Therefore, $\pi'M' = \pi'$ and we may take π' to be the stationary distribution of M'. ∎

Remark. Recall that a Phmc is a pseudo-hmc for which $\mathrm{P}[\cdot]$ is a nonnegative function. A consequence of Theorems 7 and 8 is that two minimal pseudo hmc's are equivalent if and only if the transition matrices are block similar. Using this characterization, it is easy to construct a minimal Phmc whose minimal equivalent hmc is strictly larger than itself (see Paz [P]). In a Markov chain model without the

notions of stationarity and ergodicity, Cobham [Co] gives a very elegant construction of a 5-state pseudo hidden Markov chain which is equivalent to an m-state hidden Markov chain but no smaller one, for arbitrarily large m. This is evidence that finding the minimal pseudo-hmc equivalent to a given hmc is easier than finding the minimal hmc.

Open Question 3 *Does the construction of Cobham extend to our model? Does there exist a Phmc which is not equivalent to any nonnegative hmc? Note that a negative answer to the second question would also answer Open Question 1.*

3.4 The Probability Oracle

In this section we define the probability oracle. We give a polynomial-time algorithm with oracle queries for finding the minimal pseudo-hmc equivalent to a given pseudo-hmc. We give two polynomial-time algorithms with oracle queries for determining whether two pseudo-hmc's are equivalent. These algorithms are based on efficient tree-pruning methods for constructing the bases of 0-string functions and 1-string functions introduced in proof of Theorem 8 and for finding a finite set of arguments to these functions over which they are linearly independent. We also derive an oracle algorithm for hypothesis testing for pseudo-hmc's and we give an application to hypothesis testing for Huffman codes.

The probability oracle is an adaptation of the oracle considered by Tzeng [Tz92a, Tz92b] for a unifilar, nonergodic, nonstationary hidden Markov chain model. The idea of a probability oracle is a natural extension of the idea of a membership oracle for dfa's considered by Gold [Go72] and Angluin [Ang87, Ang88].

Definition 6 (Probability Oracle) *Suppose an algorithm takes as part of its input a bound m on the size of a given unknown pseudo-hmc M. The* probability oracle *receives as a query a string $w \in \{0,1\}^*$ and returns $P^M[w]$.*

Algorithms for Equivalence and Minimization

Let M be an unknown pseudo-hmc with m states. The first obstacle for both algorithms is that the state functions form a finite dimensional space but they take values on an infinite set of strings. The subroutine described in the first subsection extracts a finite set of strings that witness all of the structure of the state functions.

The Witness-strings Subroutine

Let (Q, χ, M) be a pseudo hmc. For each string $w = w_1 w_2 \cdots w_n$ let $\mathbf{P}(w)$ denote the column vector $(\mathrm{P}_{q_1}[w], \ldots, \mathrm{P}_{q_m}[w])^T$. This vector satisfies

$$\mathbf{P}(w) = M_{w_1} M_{w_2} \cdots M_{w_n} \mathbf{1}.$$

Therefore, for $\sigma \in \{0,1\}$ the following relation holds:

$$\mathbf{P}(\sigma w) = M_\sigma \mathbf{P}(w). \tag{3.12}$$

A set of *witness strings* is defined as a maximal set of strings $W = \{w_1, \ldots, w_k\}$ so that $\mathbf{P}(w_1), \ldots, \mathbf{P}(w_k)$ are linearly independent. Note that k can be at most m. The subroutine constructs W by picking the strings one at a time in a specified order, and adding w to W if $\mathbf{P}(w)$ is linearly independent of the previous ones.

WITNESS-STRINGS SUBROUTINE.

Input: (Q, χ, M).

Output: W.

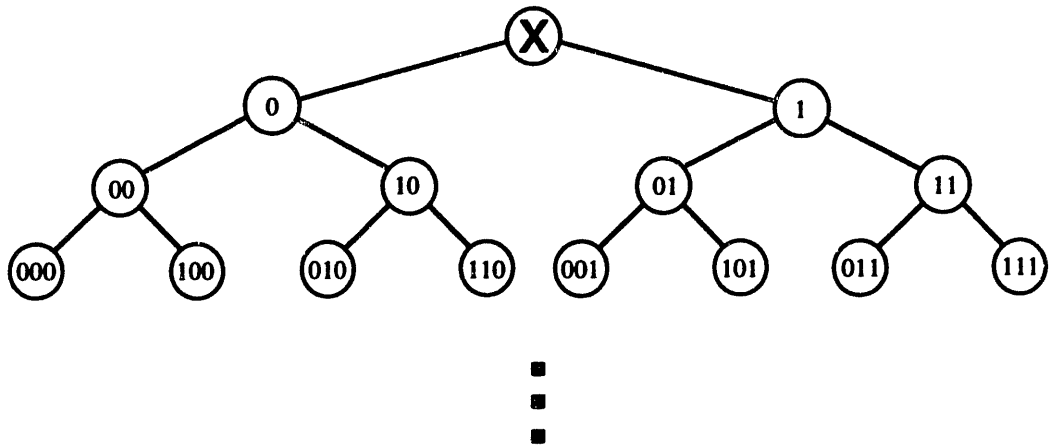

Figure 3.4: The tree of binary strings w.

1. Label the nodes of an infinite binary tree with binary strings so that the root is the null string, and for every string w the left child of w is $0w$ and the right child is $1w$ (see Figure 3.4). The subroutine will proceed by traversing the tree breadth first and pruning the tree below certain nodes, until what is left is a finite tree whose nodes are labelled by exactly the strings in W.

2. Initially, set $W = \emptyset$. Start traversing the tree at the node 0, followed by 1, 00, 10, 01, etc. At each node w,

   (a) test whether the vector $\mathbf{P}(w)$ is linearly independent of $\{\mathbf{P}(v) : v \in W\}$. If it is, add w to W. If it is not, prune the subtree rooted at w (including w itself).

   (b) Continue to the next unpruned node.

3. Stop when the unpruned portion is finite.

Claim 3 *The Witness-strings Subroutine halts after examining at most* $2m+1$ *nodes, and the set* W *generated by it is a set of witness strings. Every string in* W *has length*

at most $m-1$.

Proof. The subroutine clearly halts, and the label of every node in the finite tree that remains belongs to W. Let T denote that finite tree. T contains at most m nodes. Therefore, its depth is at most $m-1$, and the length of the longest string in W is also at most $m-1$. The subroutine examines the nodes of W and their children, a total of at most $2m+1$ nodes.

Let $\mathbf{P}(W)$ denote the set of column vectors indexed by W: $\mathbf{P}(W) = \{\mathbf{P}(w) : w \in W\}$. Impose a lexicographic ordering on W so that $\mathbf{P}(W)$ can be considered a matrix of column vectors. The column vectors of $\mathbf{P}(W)$ are linearly independent by the construction of W. We show by induction on the distance to T that for every string v not in W the vector $\mathbf{P}(v)$ is linearly dependent on $\mathbf{P}(W)$. Let v denote both the string and the node it labels. If $\mathrm{dist}(v,T) \leq 1$, then either $v \in W$ or the subroutine discovered that $\mathbf{P}(v)$ is linearly dependent on some subset of $\mathbf{P}(W)$ and pruned the subtree rooted at v. Now suppose that if $\mathrm{dist}(v',T) = j$ then $\mathbf{P}(v')$ satisfies some linear dependence of the form

$$\mathbf{P}(v') = \sum_{w \in W} \alpha_w \mathbf{P}(w). \tag{3.13}$$

and let v be such that $\mathrm{dist}(v,T) = j+1$. We can write $v = \sigma v'$, where $\mathbf{P}(v')$ satisfies Equation (3.13). By Equation (3.12),

$$\begin{aligned} \mathbf{P}(v) &= M_\sigma \mathbf{P}(v') \\ &= M_\sigma \sum_{w \in W} \alpha_w \mathbf{P}(w) \\ &= \sum_{w \in W} \alpha_w \mathbf{P}(\sigma w). \end{aligned} \tag{3.14}$$

For each $w \in W$, $\mathrm{dist}(\sigma w, T) \leq 1$; therefore the right-hand side of (3.14) can be rewritten as a linear combination of the $\mathbf{P}(w)$, $w \in W$. This proves the claim. ∎

An Algorithm For Finding a Minimal Equivalent Pseudo-Hmc

We give an algorithm that takes as input a known m-state pseudo-hmc M and in time polyomial in m constructs the minimal pseudo-hmc M' equivalent to M of Theorem 8. The heart of the algorithm is a subroutine which efficiently constructs the basis $\mathbf{P}$ of string functions in the proof of Theorem 8. We follow the notation of that theorem.

MINIMAL PSEUDO-HMC ALGORITHM.

Input: (Q, χ, M).

Output: (Q', χ', M').

1. Generate the basis of string functions $\mathbf{P}(\cdot) = \{\mathrm{P}_{v_1}[\cdot], \ldots, \mathrm{P}_{v_{m_0}}[\cdot], \mathrm{P}_{w_1}[\cdot], \ldots, \mathrm{P}_{w_{m_1}}[\cdot]\}$ by calling the following STRING FUNCTIONS SUBROUTINE.

   Input: (Q, χ, M).

   Output: $\mathbf{P}$

   The subroutine will actually generate, in parallel, the two sets $\mathbf{P}_0$ and $\mathbf{P}_1$ defined by $\mathbf{P}_0 = \{\mathrm{P}_{v_1}[\cdot], \ldots, \mathrm{P}_{v_{m_0}}[\cdot]\}$ and $\mathbf{P}_1 = \{\mathrm{P}_{w_1}[\cdot], \ldots, \mathrm{P}_{w_{m_1}}[\cdot]\}$. The union of these two sets is $\mathbf{P}$.

   (a) Call the Witness-strings Subroutine on (Q, χ, M). The output is a set of strings W.

   (b) Label the nodes of a new infinite binary tree with binary strings so that the root is the null string, and for every string w the left child of w is $w0$ and the right child is $w1$ (see Figure 3.5). This subroutine proceeds at this point exactly as did the last one: it traverses the tree breadth first and prunes the tree below certain nodes, until what is left is a finite tree whose nodes are labelled by exactly the subscripts of the string functions in $\mathbf{P}$.

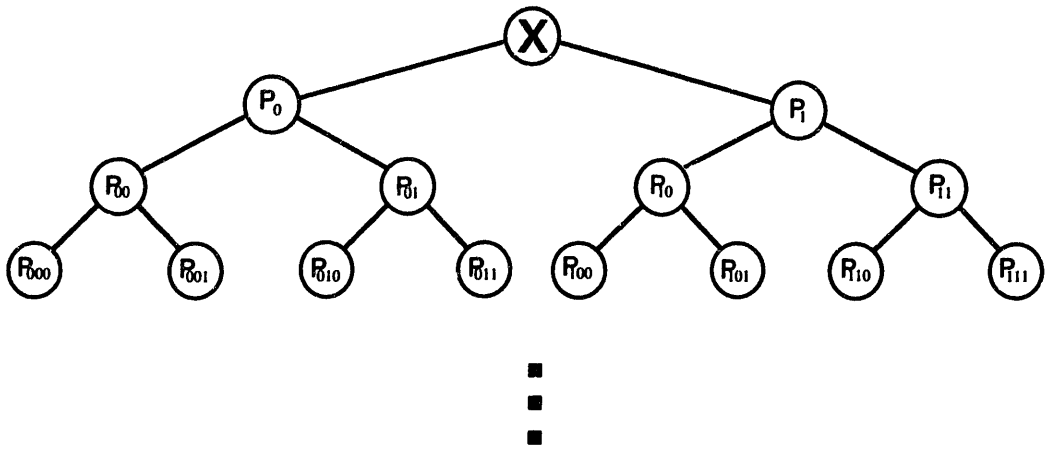

Figure 3.5: The tree of string functions $\mathrm{P}_w[\cdot]$.

(c) Initially, set $\mathbf{P}_0 = \mathbf{P}_1 = \emptyset$. Start traversing the tree at the node 0, followed by 1, 00, 01, 10, etc. At each node v, find $\sigma \in \{0, 1\}$ such that $v = v'\sigma$.

i. Test whether the row vector $\{\mathrm{P}_v[w] : w \in W\}$ is linearly independent of $\{\mathbf{P}_\sigma(v) : v \in W\}$ (again, using the lexicographic ordering on W, consider $\{\mathbf{P}_\sigma(v) : v \in W\}$ a matrix of row vectors). If it is, add $\mathrm{P}_v[\cdot]$ to $\mathbf{P}_\sigma$ If it is not, prune the subtree rooted at w (including w itself). (We don't care about any descendents that end in "$1 - \sigma$"!)

ii. Continue to the next unpruned node.

(d) Stop when the unpruned portion is finite. Set $\mathbf{P} = \mathbf{P}_0 \cup \mathbf{P}_1$.

2. For the values of m_0 and m_1 obtained by the subroutine, set $Q' = \{q_1, \ldots, q_{m_0}, r_1, \ldots, r_{m_1}\}$ as in the proof of Theorem 8. Set $\chi(q_i) = 0$ and $\chi(r_i) = 1$ for all i.

3. Calculate the shifted string function values $\mathbf{R}(w)$ for each $w \in W$ according to Equation (3.6). Solve the equation $\mathbf{P} = M'\mathbf{R}$.

4. Solve for $\pi(0)$ and $\pi(1)$ using the set of equations $\mathrm{P}^M[w] = \pi(0)\mathrm{P}_0^M[w] + \pi(1)\mathrm{P}_1^M[w]$, $w \in W$. Set $\pi' = \pi(1)\delta_{q_1} + \pi(0)\delta_{r_1}$.

Claim 4 *The String Functions Subroutine halts after examining at most* $2m + 1$ *nodes, and the sets* $\mathbf{P}_0$ *and* $\mathbf{P}_1$ *generated by it are bases for the* 0*-string functions and* 1*- string functions of* M*. The pseudo-hmc* M' *generated by the algorithm is the minimal pseudo-hmc equivalent to* M*.*

Proof. The eqivalence and minimality of M' follow from Theorem 8 and the rest of the claim. By an argument identical to that of the proof of Claim 3, with $\mathbf{P}$ standing in for W, the subroutine clearly halts after examining at most $2m + 1$ nodes. The proof that $\mathbf{P}_0$ and $\mathbf{P}_1$ are bases is identical to the proof of Claim 3, with $\mathbf{P}_0$ and $\mathbf{P}_1$ each standing in for W, and the row vectors of $\mathbf{P}_0(W)$ and $\mathbf{P}_1(W)$ each standing in for the column vectors of $\mathbf{P}(W)$. ∎

Two Equivalence Algorithms

We give two algorithms to test equivalence of two pseudo-hmc's (Q, χ, M) and (Q', χ', N), each of which has at most m states. The first algorithm implements the natural approach of reducing both pseudo-hmc's to their minimal equivalent pseudo-hmc's, which can act as signatures. The second algorithm can be generalized more easily to a hidden Markov chain model which allows arbitrary initial distributions. It also introduces the interesting construction of the "union" pseudo-hmc, a mechanism which runs the two pseudo-hmc's simultaneously.

EQUIVALENCE ALGORITHM 1: REDUCTION TO MINIMAL PSEUDO-HMC.

Input: (Q, χ, L) and (Q', χ', N).

Output: YES, if $(Q, \chi, L) \equiv (Q', \chi', N)$, and NO otherwise.

1. Call Minimal Pseudo-Hmc Algorithm to generate minimal pseudo-hmc's L' and N' such that $L' \equiv L$ and $N' \equiv N$.

2. If $L' = N'$ (as matrices), output YES. Otherwise, output NO.

EQUIVALENCE ALGORITHM 2: CONSTRUCTION OF UNION PSEUDO-HMC.

Input: (Q, χ, L) and (Q', χ', N).

Output: YES, if $(Q, \chi, L) \equiv (Q', \chi', N)$, and NO otherwise.

1. Construct the matrix for the "union" pseudo-hmc consisting of the state set $Q \cup Q'$ and the labelling $\chi \cup \chi'$, as follows: The transition probabilities within Q remain as they were, as do the transition probabilities within Q'. The transition probabilities between Q and Q' are all 0. Set the stationary distribution of the union pseudo-hmc to be the concatenation of the stationary distributions π and π' of L and N.

2. Call the Witness-string Subroutine on the union pseudo-hmc; this generates a set of strings W.

3. If for all $w \in W$, $\mathrm{P}^L[w] = \mathrm{P}^N[w]$, output YES. Otherwise, output NO.

Claim 5 *Both Equivalence Algorithms are correct and run in time polynomial in m.*

Proof. The running time of each algorithm is dominated by O(1) calls to the Witness-string Subroutine and the Minimal Pseudo-Hmc Algorithm. The correctness of Algorithm 1 follows immediately from the fact that the minimal pseudo-hmc equivalent to M given by the Minimal Pseudo-Hmc Algorithm depends only on $\mathrm{P}^M[\cdot]$, and therefore if $L \equiv N$ it must be the case that $L' = N'$.

For Algorithm 2, let m_L and m_N be the sizes of L and N. The vector $\mathbf{P}$ used by the Witness-string Subroutine is a concatenation $\mathbf{P} = \mathbf{P}_L|\mathbf{P}_N$, where $\mathbf{P}_L$ and $\mathbf{P}_N$ are the vectors the Subroutine would use on inputs L and N individually. For each

string $v \notin W$, $\mathbf{P}(v)$ there exist coefficients $\alpha_w, w \in W$ such that both $\mathbf{P}_L(v) = \sum_{w \in W} \alpha_w \mathbf{P}_L(w)$ and $\mathbf{P}_N(v) = \sum_{w \in W} \alpha_w \mathbf{P}_N(w)$. Therefore if the Algorithm answers YES, then for all v, $\mathrm{P}^L[v] = \mathrm{P}^N[v]$. This proves the claim. ∎

Remark: Other Initial Distributions. The results of this section and the previous one hold for nonstationary initial distributions. We may define equivalence of two pseudo-hmc's M_0 and M_1 with respect to initial distributions $\mathbf{d}_0$ and $\mathbf{d}_1$ by

$$(M_0, \mathbf{d}_0) \equiv (M_1, \mathbf{d}_1) \text{ iff } \mathrm{P}^{M_0}_{\mathbf{d}_0}[\cdot] = \mathrm{P}^{M_1}_{\mathbf{d}_1}[\cdot].$$

Equivalence Algorithm 2 clearly works for equivalence with respect to any initial pair of initial distributions. Equivalence Algorithm 1 and the Minimal Pseudo-Hmc Algorithm use the relation $\pi M = \pi$ from the proof of Theorem 8. This relation isn't actually necessary in the proof; the important fact is that the initial distribution is recoverable from the string functions.

Hypothesis Testing for Huffman Codes

Gillman *et al.* [GMR] have considered ambiguity of a binary files encoded by a Huffman tree from a source file [Hu], under various assumptions about how the source file was produced. We refer to [GMR] for background. We are interested in the setting in which a source is producing letters independently at random from an m-letter alphabet A. Each letter $a \in A$ has a probability p_a of being produced each time. The probabilities define a Huffman tree (for details see [Hu]) with which the source file is encoded into a binary file. The letter a becomes a binary string $w(a)$.

Given an already encoded binary file the Huffman tree will parse it in a unique way into a sequence of separate $w(a)$'s, each with its own frequency of occurrence, denoted $\mathrm{P}[w(a)]$, in the binary file. We shall assume that the binary file encoded

from the source file can be treated as a precise oracle; that is, the frequency of each $w(a)$ is precisely p_a.

Under these assumptions the question is whether the binary file is ambiguous. Is it possible for there to have been a separate source producing letters from the same alphabet A with different probabilities $q_a : a \in A$, and for the Huffman tree for these new probabilities to produce the same file?

To check the consistency of a Huffman tree with the binary file, it is certainly sufficient to evaluate P[w] for each w of length at most m in the file. This exhaustive procedure becomes expensive as m becomes large. The complexity of checking consistency remains an open question, as does the complexity of determining whether a given file is ambiguous.

Toward answering these questions we consider the *hypothesis testing problem*: given two different trees, one of which was used to produce the binary file, either decide which tree produced the file or show that the file is essentially ambiguous. This problem is easier than (i.e., can be reduced to) checking the consistency of a single tree, but *a priori* it would seem to require the same exhaustive evaluation of P[w] for all w of length n. Nevertheless, we give an algorithm to solve the hypothesis testing problem by a reduction to equivalence for hidden Markov chains, in a model where we do not insist on ergodicity and where we allow an arbitrary initial distribution. Not surprisingly, the algorithm will end up evaluating P[w] for $O(m^2)$ different w each of length at most $2m$.

This suggests that the problems of checking consistency of a tree or determining whether a file is ambiguous may admit efficient solutions, where the measure of cost is the number of binary strings w for which the frequency in the file is evaluated.

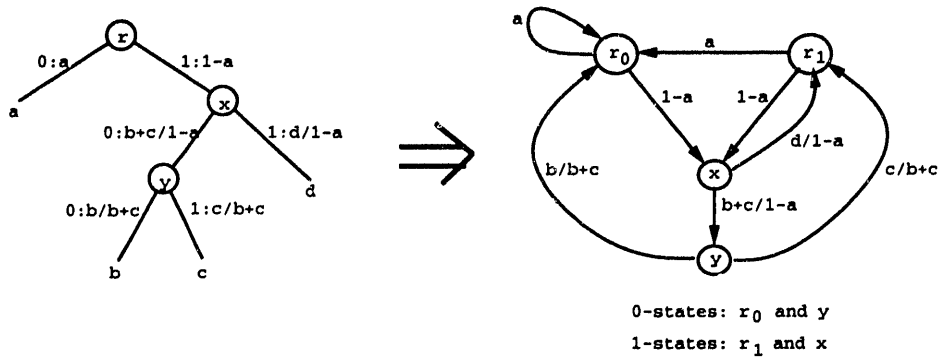

Figure 3.6: Converting a Huffman tree to a hidden Markov chain.

The independent source combined with the Huffman encoding scheme actually make up a hidden Markov chain, by a conversion shown in Figure 3.6. Under this conversion, there is a state of the Markov chain corresponding to each internal node in the tree, and the color of that state is the color of the tree branch leading down to the node. The root corresponds to two states (r_0 and r_1 in the figure), since branches to leaves implicitly lead back to the root. The initial distribution π is the point mass at either of the two root states.

Theorem 9 *Suppose the tree T used to encode the source file into the binary file is given, along with another m-letter Huffman tree S. Then there is an algorithm to determine either that the file was produced by T or that the file is ambiguous, at a cost of evaluating $P[w]$ for at most $O(m^2)$ different w each of length at most $2m$.*

Proof (ALGORITHM).

1. Parse the binary file according to both T and S. (This is equivalent to evaluating $\frac{\Pr[w0]}{\Pr[w]}$ and $\frac{\Pr[w1]}{\Pr[w]}$ for $O(m)$ different w.) This will yield the transition probabilities of the hidden Markov sources M_T and M_S corresponding to T and S, respectively, according to the conversion shown in Figure 3.6.

2. Using Equivalence Algorithm 2, determine whether M_T and M_S are equivalent (at a cost of $O(m^3)$ $m \times m$ matrix multiplications).

   (a) If they are, output AMBIGUOUS.

   (b) If not, then for some witness w_S for M_S it is the case that $\Pr[w_S] \neq \Pr_{M_S}[w_S]$. To find w_S, compare $\Pr[w]$ to $\Pr_{M_T}[w]$ for each witness w for M_T and compare $\Pr[w]$ to $\Pr_{M_S}[w]$ for each witness w for M_S, at a cost of $O(m^2)$ evaluations. Output w_S.

This proves the theorem. ∎

Remark. The algorithm given by Theorem 9 does not immediately solve the problem of checking consistency of a single Huffman tree S, except in the case where M_S is minimal (in the sense of this nonergodic, nonstationary model). The problem is that the witness strings for the tree T which was used to produce the file remain unknown, and even if T is not equivalent to S, the file may mimic S on all of the witnesses of S (we speak loosely of trees being equivalent instead of the corresponding hidden Markov chains). This problem can arise in theory, and there are simple examples of pairs of nonequivalent hidden Markov chains M and N for which $\Pr_M[w] = \Pr_N[w]$ for every witness w of N. However, we know of no example of such M and N arising from two m-letter Huffman trees.

It remains an open question to determine efficiently whether a given m-letter Huffman tree will produce an ambiguous binary file, when independence is known to the cryptographer. It would be nice to find two equivalent m-letter Huffman trees which do not both collapse to two-letter trees. We have shown that no such pairs exist for $m \leq 5$ [GMR].

3.5 Hardness of Inference

In this section we show that even for restricted classes of pseudo-hmc's, there is no polynomial-time randomized probability oracle algorithm for discrimination. This is in spite of the existence of an NP probability oracle algorithm which follows immediately from Theorem 8. The proof of intractibility is information-theoretic and constructive, and it does not depend on any assumptions about separation of complexity classes.

Definition 7 (Restricted Classes of Pseudo-Hmc's) *A* coinflip machine *with* bias p *is any pseudo-hmc whose output is a sequence of Bernoulli trials for which* $\Pr[1] = p = 1 - \Pr[0]$. *A coinflip machine is called* fair *if* $p = 1/2$.

A p-machine *is a hidden Markov chain in which every transition probability is 0,* p, $1 - p$, *or 1.*

A uniform machine *is an* m*-state hmc with transition matrix* $M = (p_{ij})$ *such that for* $1 \leq i, j \leq m$, $p_{ij} = \frac{1}{m}$. *This is a special case of a coinflip machine, in which the stationary distribution* π *is uniform and the bias satisfies* $p = \pi(1)$. *It is also equivalent to a random walk on a complete graph with self-loops.*

A pseudo-hmc is balanced *if its stationary distribution is the uniform vector. Every uniform machine is balanced.*

A pseudo-hmc is unifilar *if no state has nonzero transition probability to two or more states with the same label.*

Let $\hat{U}_p$ denote the class of unifilar p-machines. Let $\hat{U} = \bigcup_p \hat{U}_p$. Let $\overline{U}$ be the class of balanced hmc's.

For each p we define the following subclass of $\hat{U}_p$ to be used in the next theorem.

Definition 8 (Scorpions) *Let $w = w_1 w_2 \cdots w_k$ be a string of* 0*'s and* 1*'s. Fix p, $0 < p \leq 1/2$. Define the* scorpion $S_{w,p}$*, an element of $\hat{U}_p$, as follows: $S_{w,p}$ has $m = 2k+6$ states $q_0, q_1, \ldots, q_{k+2}, r_0, r_1, \ldots, r_{k+2}$. Set $\chi(q_0) = \chi(r_0) = 1 - w_1$. For $i = 1, \ldots k$, set $\chi(q_i) = \chi(r_i) = w_i$. Set $\chi(q_{k+1}) = \chi(r_{k+1}) = 1$ and $\chi(q_{k+2}) = \chi(r_{k+2}) = 0$.*

The transition probabilities are as follows:

Case 1: $i \leq k-1$.

$q \to q$ **transitions:** *If $\chi(q_{i+1}) = \chi(q_1)$ then $p_{q_i q_0} = 1-p$ and $p_{q_i q_{i+1}} = p$. Otherwise, $p_{q_i q_1} = p$ and $p_{q_i q_{i+1}} = 1-p$.*

$r \to r$ **transitions:** *If $\chi(r_{i+1}) = \chi(r_1)$ then $p_{r_i r_0} = 1-p$ and $p_{r_i r_{i+1}} = p$. Otherwise, $p_{r_i r_1} = p$ and $p_{r_i r_{i+1}} = 1-p$.*

Case 2: $i = k$.

$q \to q$ **transitions:** $p_{q_i q_{i+1}} = p/2$ *and* $p_{q_i q_{i+2}} = 1 - p/2$.

$r \to r$ **transitions:** $p_{r_i r_{i+1}} = 3p/2$ *and* $p_{r_i r_{i+2}} = 1 - 3p/2$.

Case 3: $i = k+1$ *or* $k+2$.

$q \to r$ **transitions:** $p_{q_i r_1} = p$ *and* $p_{q_i r_0} = 1-p$.

$r \to q$ **transitions:** $p_{r_i q_1} = p$ *and* $p_{r_i q_0} = 1-p$.

See Figure 3.7 for the example of S_{010}.

The scorpions are hmc's. $S_{w,p}$ is ergodic; let π_w denote its stationary distribution. For any binary string w let $0(w)$ denote the number of 0's in w and let $1(w)$ denote the number of 1's.

Let F be a coinflip machine with bias p. Then $\mathrm{P}^F[1w1] = p^{1(w)+2}(1-p)^{0(w)}$. The next lemma shows that $S_{w,p}$ is not a coinflip machine.

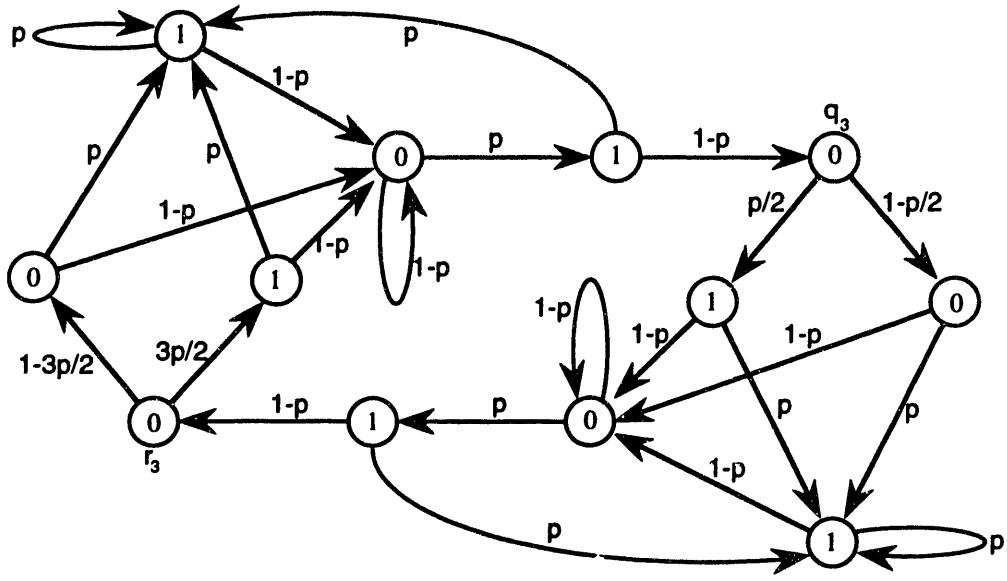

Figure 3.7: The scorpion $S_{010,p}$.

Lemma 5 *For every w, $P^{S_{w,p}}[1w1] < P^{F}[1w1]$.*

Proof. Let π_w be the statonary distribution of $S_{w,p}$. By symmetry, for $i \leq k$ $\pi_w(q_i) = \pi_w(r_i)$; also $\pi_w(q_{k+1})+\pi_w(q_{k+2}) = \pi_w(r_{k+1})+\pi_w(r_{k+2})$. The probability that $S_{w,p}$ will generate a particular string starting in a particular state is easy to calculate, since the string determines a single path through the hmc (we are using the unifilar property). Specifically, for each $i \neq k$, $\mathrm{P}_{q_i}[1w1] = \mathrm{P}_{r_i}[1w1] = p^{1(w)+2}(1-p)^{0(w)}$ and $\mathrm{P}_{q_k}[1w1] = \mathrm{P}_{r_k}[1w1] = 3p^{1(w)+2}(1-p)^{0(w)}/4$. Therefore,

$$\mathrm{P}^{S_{w,p}}[1w1] = (1-\pi_w(q_k)/2)p^{1(w)+2}(1-p)^{0(w)}. \quad \blacksquare \tag{3.15}$$

Hardness Theorems

Theorem 10 *Let k be a positive integer and let $m = 2k+6$. Fix p, $0 < p \leq 1/2$. Suppose a randomized probability oracle algorithm can discriminate a known coinflip machine with bias p from an unknown element of $\hat{U}$ of at most m states. Such an*

algorithm must take at least $2^k + k - 1$ *steps to run.*

Proof. Let F denote the coinflip machine. By hypothesis the algorithm can distinguish F from $S_{w,p}$ for each w of length k. We will show that for any string v not containing w as a substring, $\mathrm{P}^{S_{w,p}}[v] = \mathrm{P}^{F}[v] = p^{1(v)}(1-p)^{0(v)}$, Assuming this fact for now, let V be thc set of strings v for which the algorithm queries $\mathrm{P}[v]$. To discriminate every $S_{w,p}$ from F it is necessary that every w be a substring of some $v \in V$. Let W_v be the set of k bit substrings of v. Then $|v| \geq |W_v| + k - 1$. Therefore,

$$\sum_{v \in V} |v| \geq \sum_{v \in V} (|W_v| + k - 1) \geq |\{0,1\}^k| + k - 1 = 2^k + k - 1,$$

and the algorithm must write down at least that many bits.

Fix w and consider $S_{w,p}$. Choose a string v which does not have w as a substring. We will show that for each i $\mathrm{P}_{q_i}[v] + \mathrm{P}_{r_i}[v] = 2p^{1(v)}(1-p)^{0(v)}$, and in view of Lemma 5 this will prove the theorem.

A token leaving any state q will subsequently take exactly one path with the label sequence v. Denote this path $\alpha(q, v)$, and write $q' \in \alpha(q, v)$ if the path passes through q'. We adopt the convention that $q \in \alpha(q, v)$. Three facts hold:

1. $q_{k+1} \in \alpha(q_i, v)$ if and only if $r_{k+1} \in \alpha(r_i, v)$, and $q_{k+2} \in \alpha(q_i, v)$ if and only if $r_{k+2} \in \alpha(r_i, v)$.

2. If $\alpha(q_i, v) \cap \{q_{k+1}, q_{k+2}\} \neq \emptyset$ then $\alpha(q_i, v) \cap \{r_{k+1}, r_{k+2}\} = \emptyset$.

3. If $\alpha(r_i, v) \cap \{r_{k+1}, r_{k+2}\} \neq \emptyset$ then $\alpha(r_i, v) \cap \{q_{k+1}, q_{k+2}\} = \emptyset$.

There are three cases to consider:

Case 1: $q_{k+1} \in \alpha(q_i, v)$. Then

$$\mathrm{P}_{q_i}[v] = p^{1(v)}(1-p)^{0(v)}/2 \text{ and } \mathrm{P}_{r_i}[v] = 3p^{1(v)}(1-p)^{0(v)}/2.$$

Case 2: $q_{k+2} \in \alpha(q_i, v)$. Then

$$P_{q_i}[v] = (1 - p/2)p^{1(v)}(1-p)^{0(v)-1} \text{ and } P_{r_i}[v] = (1 - 3p/2)p^{1(v)}(1-p)^{0(v)-1}.$$

Case 3: $\alpha(q_i, v) \cap \{q_{k+1}, q_{k+2}\} = \emptyset$. Then

$$P_{q_i}[v] = p^{1(v)}(1-p)^{0(v)} \text{ and } P_{r_i}[v] = p^{1(v)}(1-p)^{0(v)}.$$

In all three cases, $P_{q_i}[v] + P_{r_i}[v] = 2p^{1(v)}(1-p)^{0(v)}$. ∎

Theorem 10 shows that $S_{w,p}$ contains two states, q_k and r_k, which have long memories: to go from one of these states to the other, $S_{w,p}$ must generate w. Tracing the transition probabilities backward shows that these states are very infrequently visited: $\pi_w(q_k) = \pi_w(r_k) = O(p^{1(v)}(1-p)^{0(v)})$. It is only after passing through both of these states that the behavior of $S_{w,p}$ differs from the coinflip machine F. Therefore, one would expect to have trouble discriminating between $S_{w,p}$ and F.

Ron Rivest has pointed out to us an alternative construction of $S_{w,p}$ based on the finite automaton which is used to implement the pattern matching algorithm of Knuth, Morris, and Pratt, for detecting instances of w in a string (see Cormen et. al. [CLR]). The resulting hmc would be very similar to ours. (The only affected transitions in the definition of $S_{w,p}$ would be those of probability $1 - p$ in case 1, which in Rivest's alternative construction would sometimes go to q_i, $i > 1$, for $q \to q$ transitions, or to r_i, $i > 1$, for $r \to r$ transitions, depending on w. We omit the details.) It may, however, be slightly easier to characterize the behavior of this alternative hmc for the purposes of Theorem 10.

We will construct a new class of hmc's, based on the scorpions, that visit every state frequently. The next theorem will show somewhat surprisingly that this class is still hard to distinguish from a coinflip machine.

Lemma 6 *Let j, k be positive integers such that $j \leq (k+3)/2$. Set $p = j/(k+3)$. Fix a binary string $w = w_1 w_2 \cdots w_k$ such that if we set $w_0 \neq w_1$ then $1(w_0 w)+1 = j$. Consider the scorpion $S_{w,p}$. There is a pseudo-hmc equivalent to $S_{w,p}$ with uniform stationary distribution, which we will denote $\Psi_{w,p}$.*

Proof. Let π_w be the statonary distribution of $S_{w,p}$. Note that $\pi_w(1) = \mathrm{P}^{S_{w,p}}[1] = p$. Therefore, we can find a block similarity matrix S such that the vector π'_w defined by $\pi'_w = \pi_w S$ is uniform. Identifying pseudo-hmc's with their transition matrices, let $\Psi_{w,p} = S^{-1} S_{w,p} S$. By Theorem 7, $\Psi_{w,p} \equiv S_{w,p}$. ∎

There are many $\Psi_{w,p}$'s, depending on the choice of S, but it won't matter which S we choose.

Definition 9 (Averaged Scorpions) *Let j, k be positive integers such that $j \leq (k+3)/2$. Let $m = 2k+6$ and $p = j/(k+3)$. Fix a binary string $w = w_0 w_1 \cdots w_k$ such that if we set $w_0 \neq w_1$ then $1(w)+1 = j$. Define the* averaged scorpion *$T_{w,p}$, an element of $\overline{U}$, as follows: $T_{w,p}$ has the same set Q of $m = 2k+6$ states as $S_{w,p}$; $Q = \{q_0, q_1, \ldots, q_{k+2}, r_0, r_1, \ldots, r_{k+2}\}$. The coloring χ is also the same. Let U be a uniform machine on state set Q with coloring χ. Choose ρ, $0 < \rho < 1$, small enough that $\rho\Psi_{w,p} + (1-\rho)U$ is a positive matrix. Put $T_{w,p} = \rho\Psi_{w,p} + (1-\rho)U$.*

Note that $T_{w,p}$ is a balanced hmc. With this definition we have introduced the technique of averaging a pseudo-hmc with a uniform machine. The following lemma demonstrates the power of this technique.

Lemma 7 *For all strings v not containing w as a substring, $P^{T_{w,p}}[v] = P^U[v]$. Also, $P^{T_{w,p}}[1w1] \neq P^U[1w1]$.*

Proof. Let π be the uniform distribution on Q. $T_{w,p}$ is equivalent to the following random process which starts in π:

1. Flip a coin with Pr[heads] $= \rho$. If heads, put $R = \Psi_{w,p}$; if tails, put $R = U$.
2. Run R for one step.
3. Repeat steps 1. and 2. *ad infinitum.*

Let X denote this process. Let v be a binary string of length n. Let R^n denote a given n-tuple $R^n = \{R_1, \ldots, R_n\}$ of matrices, $R_i \in \{U, \Psi_{w,p}\}$. Define $\mathrm{P}^{X|R^n}[v]$ to be the probability R^n assigns v, in a manner analogous to Equation (3.4). In particular, let $(R_i)_0 = R_i I_0$ and $(R_i)_1 = R_i I_1$, and put

$$\mathrm{P}^{X|R^n}[v] = \pi(R_i)_{v_1} \cdots (R_n)_{v_n} \mathbf{1}.$$

Then X defines a probability function $\mathrm{P}^X[\cdot]$ according to the formula

$$\mathrm{P}^X[v] = \sum_{R^n} \Pr[R^n] \mathrm{P}^{X|R^n}[v]. \tag{3.16}$$

By expanding $T_{w,p}$ in terms of U and $\Psi_{w,p}$ and using Equation (3.4), we see that for all v, $\mathrm{P}^X[v] = \mathrm{P}^{T_{w,p}}[v]$.

We first prove that if v does not contain w as a substring, $\mathrm{P}^{X|R^n}[v] = \mathrm{P}^U[v]$, for all R^n. The proof is by induction on n. The base case $n = 1$ is trivial, so suppose $n > 1$ and fix R^n. Recall that $\pi_0 = \pi I_0$ and $\pi_1 = \pi I_1$. There are three cases to consider:

Case 1: $R_n = U$. By induction, $\pi(R_i)_{v_1} \cdots (R_{n-1})_{v_{n-1}} \mathbf{1} = \mathrm{P}^U[v_1 \cdots v_{n-1}]$. Therefore, $\pi(R_i)_{v_1} \cdots (R_n)_{v_n} = \mathrm{P}^U[v_1 \cdots v_{n-1}]\pi_{v_n}$ and $\pi(R_i)_{v_1} \cdots (R_n)_{v_n} \mathbf{1} = \mathrm{P}^U[v]$.

Case 2: For some $i < n$, $R_i = U$ and $R_{i+1} = R_{i+2} = \cdots = R_n = \Psi_{w,p}$. By induction, $\pi(R_i)_{v_1} \cdots (R_i)_{v_i} = \mathrm{P}^U[v_1 \cdots v_{i-1}]\pi_{v_i}$. Therefore, by Theorem 10 and Lemma 6,

$$\begin{aligned} \pi(R_i)_{v_1} \cdots (R_n)_{v_n}\mathbf{1} &= \mathrm{P}^U[v_1 \cdots v_{i-1}]\pi(\Psi_{w,p})_{v_i} \cdots (\Psi_{w,p})_{v_n}\mathbf{1} \\ &= \mathrm{P}^U[v_1 \cdots v_{i-1}]\mathrm{P}^{\Psi_{w,p}}[v_i \cdots v_n] \\ &= \mathrm{P}^U[v]. \end{aligned} \tag{3.17}$$

Case 3: $R_1 = R_2 = \cdots = R_n = \Psi_{w,p}$. In this case,

$\mathrm{P}^{X|R^n}[v] = \mathrm{P}^{\Psi_{w,p}}[v] = \mathrm{P}^U[v]$.

This proves the first part of the lemma.

For the second part, let $n = k + 3$. A closer inspection of $S_{w,p}$ reveals that for any proper substring v of $1w1$, $\mathrm{P}^{S_{w,p}}[v] = \mathrm{P}^U[v]$. By Lemma 6 and the argument of case 2 above, unless $R_1 = R_2 = \cdots = R_n = \Psi_{w,p}$, $\mathrm{P}^{X|R^n}[1w1] = \mathrm{P}^U[1w1]$. By Lemmas 5 and 6, $\mathrm{P}^X[1w1] = \mathrm{P}^U[1w1] - \pi_w(q_k)p^{1(w)+2}(1-p)^{0(w)}/2$, where π_w is the stationary distribution of $S_{w,p}$. This proves the lemma. ∎

Theorem 11 *Let j, k be positive integers such that $j \leq (k+3)/2$. Let $m = 2k+6$ and $p = j/(k+3)$. Suppose a randomized probability oracle algorithm can discriminate a known coinflip machine with bias p from an unknown element of $\overline{U}$ of at most m states. Such an algorithm must take at least $\binom{k-1}{j-2} + k$ steps to run. In particular, if $p = 1/2$, the algorithm takes $\Omega(2^k/\sqrt{k})$ steps.*

Proof. Let V be the set of strings v for which the algorithm queries $\mathrm{P}[v]$. To discriminate every $T_{w,p}$ from F it is necessary that every w be a substring of some $v \in V$. There are at least $\binom{k-1}{j-2}$ different strings $w = w_1 \cdots w_k$ such that if we set

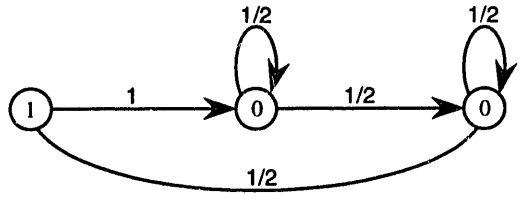

Figure 3.8: An hmc which is not equivalent to any unifilar hmc.

$w_0 \neq w_1$ then $1(w_0 w) + 1 = j$. Let W_v be the set of $k+1$-bit substrings of v. Then $|v| \geq |W_v| + k$. Therefore,

$$\sum_{v \in V} |v| \geq \sum_{v \in V} (|W_v| + k) \geq \binom{k-1}{j-2} + k,$$

and the algorithm must write down at least that many bits. ∎

Notes

Rudich [Ru] and Tzeng [Tz92b] both have considered inference of unifilar Markov chains. Rudich shows that from an infinite stream of empirical data, a Markov chain is inferrable in the limit. Tzeng considers a hidden Markov chain model motivated by automata theory which has no notion of stationarity or ergodicity. He establishes the intractibility of inference in this model by a construction very similar to that of Theorem 10.

The unifilar hidden Markov chains are a very restricted class. Figure 3.8 shows an example of a simple hmc which is not equivalent to any unifilar hmc.

3.6 Randomized Algorithms for Discrimination

In this section we give a randomized probability oracle algorithm with one-sided error for discriminating a known fair coinflip machine from an unknown balanced unifilar

1/2-machine. This is a first step toward establishing some form of tightness for the theorems of the last section. We give an example of a familiy of balanced unifilar 1/2-machines on $O(k)$ states which behave like coinflip machines on all strings of length k; this example gives some evidence that randomness is needed for discrimination.

We also present partial results on randomized discrimination that follow from Theorem 3 on entropy estimation. We consider unifilar random walks on graphs whose vertices are labelled by a finite alphabet. We give a randomized probability oracle algorithm with one-sided error for discriminating an unknown machine of this class from a known sequence of independent labels. Koopmans [Koo] has previously considered discrimination based on large deviations, for continuous Markov chains on the real line.

The technique introduced by the second algorithm is to let the oracle simulate the black box and to discriminate based on the conditional probability of each label in the sequence. We believe this technique could potentially generalize to discriminating an arbitrary known machine using relative entropy.

Logarithms are in base 2.

A Randomized Algorithm

DISCRIMINATION ALGORITHM FOR $\hat{U}_{1/2} \cap \overline{U}$

Input: One known fair coinflip machine F, plus one unknown element M of $\hat{U}_{1/2} \cap \overline{U}$ with at most m states.

Output: If $F \equiv M$, then output YES. If $F \not\equiv M$, then with probability at least 1/2, output NO, along with a string w such that $\mathrm{P}^F[w] \neq \mathrm{P}^M[w]$.

1. Let $n = 32m^2 \log m$. Choose a binary string w of length $n+1$ uniformly at random.

2. Query the oracle for $\mathrm{P}^M[w]$.

3. If $\mathrm{P}^M[w] = 2^{-n-1}$, output YES. If not, output NO, along with w.

Theorem 12 *The Discrimination Algorithm for $\hat{U}_{1/2} \cap \overline{U}$ works correctly as specified. Its running time is $O(m^2 \log m)$.*

Proof. We may assume $m > 2$. The algorithm is clearly correct whenever $M \equiv F$. Suppose $M \not\equiv F$. We will show that with probability at least $1/2$ a random w satisfies $\mathrm{P}^M[w] = 0$. It will suffice to show that for each state q of M, $\mathrm{P}_q^M[w] \neq 0$ with probability at most $1/2m$.

Call a state "bad" if it has transition probability 1 to some other state; call it "good" otherwise. Because $M \in \hat{U}$ there must be some bad state, call it q_b. Because of the unifilar property, the random string $w = w_1 w_2 \cdots w_n$ defines a trajectory of the Markov chain M as follows: suppose after step k the Markov chain is in state q. If q is good, choose the next state to be the outgoing neighbor of q labelled w_{k+1}. If q is bad, ignore w_{k+1} and simply go to the unique outgoing neighbor of q.

The next lemma borrows heavily from a result of Aleliunas et al. [AK*] concerning random walks on Eulerian directed graphs.

Definition 10 *Let L be an ergodic Markov chain. Let q, q' be two states of L. The* hitting time $ET_{q,q'}$ *is the expected number of steps to reach q' starting from q.*

Lemma 8 *For any initial state q_0, $ET_{q_0 q_b} \leq 2m^2$.*

Proof. Since M is ergodic there is a path of length k from q_0 to q_b, where $k \leq m$ (we allow $q_0 = q_b$). Denote this path by

$$q_0 \to q_1 \to \cdots \to q_{k-1} \to q_k = q_b.$$

It will suffice to show that for each i, $0 \leq i \leq k-1$, $ET_{q_i q_{i+1}} \leq 2m$. Each time the Markov chain leaves q_i the probability is at least $1/2$ that it will transit to q_{i+1}. By the expectation of a geometric random variable, therefore, $ET_{q_i q_{i+1}} \leq 2ET_{q_i q_i}$. But by the Ergodic Theorem [Fe] $ET_{q_i q_i} = 1/\pi(q_i) = m$. Therefore, $ET_{q_i q_{i+1}} \leq 2m$. ∎

Let $\alpha = \alpha(q, w)$ denote the trajectory of M defined by w with initial state $\alpha_0 = q$. Let $\alpha_1 \alpha_2 \cdots \alpha_n$, be the sequence of states in this trajectory. Partition α into $8 \log m$ blocks of length $4m^2$ each, plus one bit at the end. Let α^i denote the i^{th} block. By Lemma 8 and Markov's inequality, for each i, $1 \leq i \leq 8 \log m$, $\Pr[q_b \in \alpha^i] \geq 1/2$, independent of the other α^j. The expected number of $i \leq n$ such that $q_b \in \alpha^i$ is at least $4 \log m$, and Chernoff's bound [Ch] implies that

$$\Pr[|\{i \leq n : \ q_b \in \alpha^i\}| < \log m] \ < \ e^{-(6 \log m)^2/16 \log m} \ < \ 1/4m, \qquad (3.18)$$

for $m > 2$.

Observe that $\mathrm{P}_q^M[w] = 0$ if and only if, for some $i \leq n+1$, $\chi(\alpha_i) \neq w_i$. Whenever $\alpha_i = q_b$ the probability is $1/2$ that $\chi(\alpha_{i+1}) \neq w_i$. By Equation (3.18),

$$\begin{aligned} \Pr[\forall i \leq n+1, \chi(\alpha_i) = w_i] \ &\leq \ \Pr[|\{i \leq n : \ \alpha_i = q_b\}| < \log m] \ + \\ &\qquad \Pr[|\{i \leq n : \ \alpha_i = q_b\}| \geq \log m \text{ and} \\ &\qquad\qquad \forall i \leq n+1, \chi(\alpha_i) = w_i] \\ &\leq \ 1/4m \ + \ 1/4m \\ &= \ 1/2m \,. \end{aligned} \qquad (3.19)$$

This completes the proof of Theorem 12. ∎

Remark. The condition that M have uniform stationary distribution is flexible. We used it only in Lemma 8 to show that for every state q, $1/\pi(q) \leq m$. Any polynomial bound $p(m)$ on $\max_q 1/\pi(q)$ will do. The value of n would then be $32mp(m)\log m$.

An Example: Leapfrogs

In this subsection we describe a collection of balanced unifilar 1/2-machines. Each of these hmc's has $O(k)$ states and behaves like a coinflip machine on all strings of length k. This family of examples gives some evidence that randomness is needed for discrimination.

Definition 11 (Leapfrogs) *Let $w = w_1 w_2 \cdots w_k$ be a string of* 0*'s and* 1*'s. Define the* leapfrog $L_{w,p}$*, an element of $\hat{U}_{1/2} \cap \overline{U}$, as follows: L_w has state set Q containing $m = 4k + 2$ states; $Q = \{q_0, \bar{q}_0, q_1, \bar{q}_1, \ldots, q_k, \bar{q}_k, s_0, r_0, \bar{r}_0, r_1, \bar{r}_1, \ldots, r_k, \bar{r}_k, s_1\}$. For $i = 1, \ldots k$, set $\chi(q_i) = \chi(r_i) = w_i$, and set $\chi(\bar{q}_i) = \chi(\bar{r}_i) = 1 - w_i$. Set $\chi(s_1) = w_1$ and $\chi(s_0) = 1 - w_1$.*

The transition probabilities are as follows:

Case 1: $i \leq k - 1$.

$q \to q$ **transitions:** $p_{q_i q_{i+1}} = p_{q_i \bar{q}_{i+1}} = 1/2$, *and* $p_{\bar{q}_i \bar{q}_i} = p_{\bar{q}_i q_i} = 1/2$.

$r \to r$ **transitions:** $p_{r_i r_{i+1}} = p_{r_i \bar{r}_{i+1}} = 1/2$, *and* $p_{\bar{r}_i \bar{r}_i} = p_{\bar{r}_i r_i} = 1/2$.

Case 2: $i = k$.

$q \to q$ **transitions:** $p_{q_i s_1} = 1$ *and* $p_{\bar{q}_i \bar{q}_i} = p_{\bar{q}_i q_i} = 1/2$.

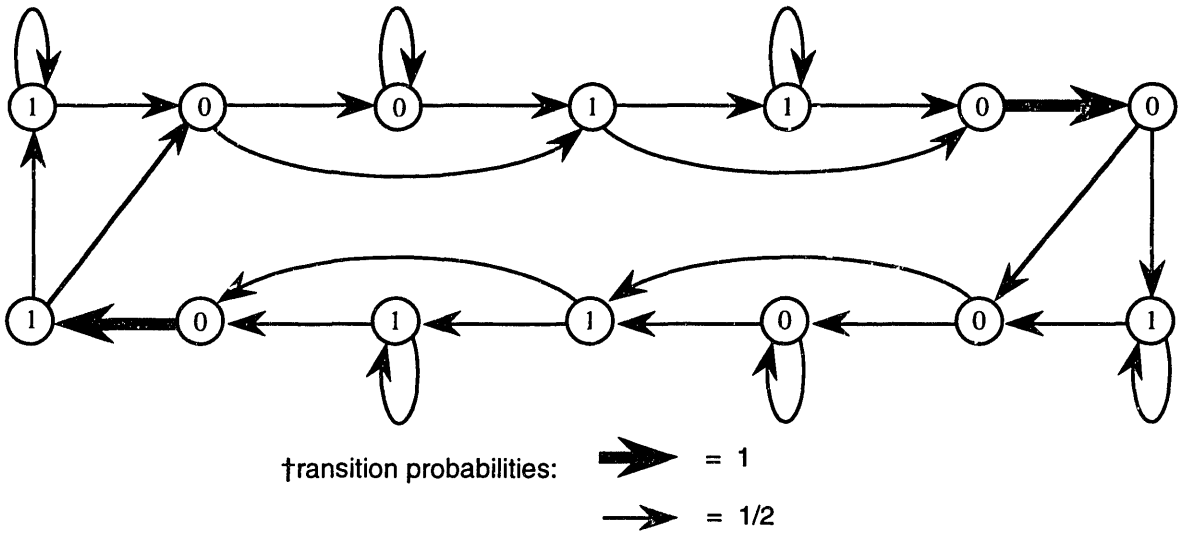

Figure 3.9: The leapfrog L_{010}.

$r \to r$ **transitions:** $p_{r_i s_0} = 1$ *and* $p_{\bar{r}_i \bar{r}_i} = p_{\bar{r}_i r_i} = 1/2$.

Case 3: s_0, s_1.

$$p_{s_0 \bar{q}_1} = p_{s_0 q_1} = 1/2, \text{ and } p_{s_1 \bar{r}_1} = p_{s_1 r_1} = 1/2.$$

See Figure 3.9 for the example of L_{010}.

Let π denote the uniform distribution on Q. Observe that π is the stationary distribution for L_w, for any w of length k. Recall that $\mathbf{d}_w$ denotes the distribution reached after seeing the string w. Let $(\mathbf{d}_w)_q$ denote the q^{th} coordinate of this vector. As was the case with the scorpions, the two crucial states in L_w are q_k and r_k.

Lemma 9 *Let v be a binary string. Suppose $P^{L_w}[v] \neq 2^{-|v|}$. Then for some proper prefix u of v, $(\mathbf{d}_{v'})_{q_k} \neq (\mathbf{d}_{v'})_{r_k}$.*

Proof. Suppose that $(\mathbf{d}_{v'})_{q_k} = (\mathbf{d}_{v'})_{r_k}$ for every proper prefix v'. Then we claim that for *every* prefix u, including v itself, $\mathrm{P}^{L_w}[u] = 2^{-|u|}$. We prove this by induction on $|u|$. The case $|u| = 1$ is true because $\mathrm{P}^{L_w}[0] = \pi(0) = 1/2$. Suppose now that

$u = u'\sigma$, for $\sigma \in \{0,1\}$ and $|u'| \geq 1$, and that $\mathrm{P}^{L_w}[u'] = 2^{-|u'|}$. Since for every $q \in Q \setminus \{q_k, r_k\}$, $\mathrm{P}_q^{L_w}[\sigma] = 1/2$, and since $\mathrm{P}^{L_w}_{1/2(\delta_{q_k}+\delta_{r_k})}[\sigma] = 1/2$, we have $\mathrm{P}^{L_w}[u] = 1/2\mathrm{P}^{L_w}[u'] = 2^{-|u|}$. ∎

The next theorem shows that L_w is indistinguishable from a coinflip machine on all short strings.

Theorem 13 *The shortest word v for which $P^{L_w}[v] \neq 2^{-|v|}$ has length $k+2$. An example is $v = w_1 w w_1$.*

Proof. Fix v such that $|v| \leq k$. In view of Lemma 9, it will suffice to show that $(\mathbf{d}_v)_{q_k} = (\mathbf{d}_v)_{r_k}$. Then $(\mathbf{d}_v)_{q_k}$ is a sum of contributions from paths starting in s_0, the q_i's, and the $\bar{q}_i$'s, and ending in q_k. For each such path there is a corresponding path with the same probability from s_1, an r_i, or an $\bar{r}_i$, and ending in r_k. Therefore, $(\mathbf{d}_v)_{q_k} = (\mathbf{d}_v)_{r_k}$.

The same argument holds for $v = w_1 w$ except that there is a path $\alpha(q_k, w_1 w)$, labelled $w_1 w$, starting from q_k and ending in r_k; and there is no corresponding path from r_k to q_k. Therefore, $(\mathbf{d}_{w_1 w})_{q_k} < (\mathbf{d}_{w_1 w})_{r_k}$, and so $\mathrm{P}^{L_w}[w_1 w w_1] < 2^{-k-2}$. ∎

Entropy Estimation and Discrimination

In this subsection we establish a method of randomized discrimination on an hmc-like model, as a consequence of Corollary 1. Our algorithm uses the probability oracle to mimic the black box behavior of the unknown machine, by checking conditional probabilties at each step.

Let a *hidden random walk (hrw)* be an ergodic random walk on an unweighted graph G, together with a labelling χ of the vertices of G by some finite alphabet Σ

such that no vertex of G is adjacent to two vertices of the same label. Except for the size of the alphabet, in all respects an hrw is a special case of a unifilar hmc; and all of the notation of hmc's will carry over to hrw's.

Define a *Bernoulli hrw* to be an hrw which is equivalent to a sequence of independent letters chosen uniformly at random. In the following assume an underlying k-letter alphabet Σ.

DISCRIMINATION ALGORITHM FOR HRW'S

Input: hrw's B and N; positive parameters $\epsilon \leq 1$ and $\gamma \leq 1$. B is a known Bernoulli hrw, and N is an unknown hrw on a graph G on m vertices, which generates an information source Y. ϵ is a lower bound on the eigenvalue gap of G and $\log k - \gamma$ is an upper bound on $H(Y)$.

Output: If $B \equiv N$, then output YES. If $B \not\equiv N$, then with probability at least $1/2$ output NO, along with a string w such that $\mathrm{P}^B[w] \neq \mathrm{P}^N[w]$.

1. **For** each $\sigma \in \Sigma$,
   (a) Query the oracle for $\mathrm{P}^N[\sigma]$.
   (b) **If** $\mathrm{P}^N[\sigma] \neq 1/k$, output NO, along with σ.
   (c) **Else** continue.
2. Set $w_1 = \sigma$ with probability $\mathrm{P}^N[\sigma]$.
3. Let $n = \lceil 50 \frac{k \log^2 k \log m}{\epsilon \gamma^2} \rceil$.
4. **For** $i = 1, 2, \ldots, n-1$,
   (a) Let w^i denote the string $w_1 \cdots w_i$.
   (b) **For** each $\sigma \in \Sigma$,

i. Query the oracle for $\mathrm{P}^N[w^i\sigma|w^i]$.

ii. **If** $\mathrm{P}^N[w^i\sigma|w^i] \neq 1/k$, output NO, along with $w^i\sigma$.

iii. **Else** continue.

(c) Set $w_{i+1} = w^i\sigma$ with probability $\mathrm{P}^N[w^i\sigma|w^i]$.

5. Output YES.

Theorem 14 *The Discrimination Algorithm for Hrw's works correctly as specified. Its running time is* $O(n^2k)$, *where* $n = \lceil 50\frac{k\log^2 k\log m}{\epsilon\gamma^2}\rceil$.

Proof. The algorithm clearly outputs YES if $N \equiv B$. Suppose that $N \not\equiv B$. Observe that the algorithm will output YES only if for all $i \geq 1$, $\mathrm{P}^N[w^i] = k^{-i}$ Recall that the empirical entropy V_n of Y is given by $V_n = -\frac{1}{n}\log_2 \mathrm{P}^N[w]$. By Equation (2.14) and Remark (ii) following Corollary 1,

$$\Pr_w[|V_n - H(Y)| \geq \gamma] \leq 4e^{-2.1} \leq 1/2.$$

But the left-hand side dominates $\Pr_w[\forall i, \mathrm{P}^N[w^i] = k^{-i}]$.

The running time is dominated by nk oracle queries for strings of length $O(n)$ each. The theorem is proved. ∎

Further Work

1. There is a gap to be closed between the positive results of this section and the negative results of the last section. In particular, we ask whether the Discrimination Algorithm for $\hat{U}_{1/2} \cap \overline{U}$ can be extended to

   (a) Balanced 1/2-machines, or

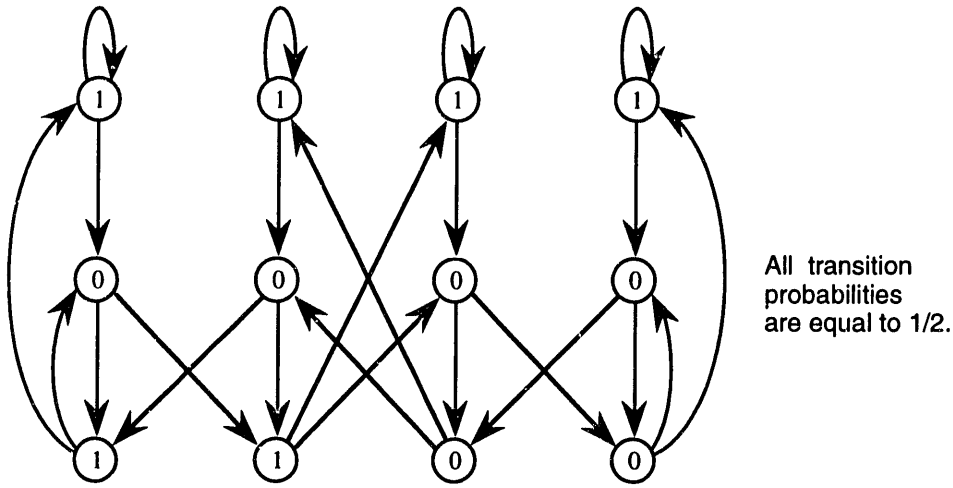

Figure 3.10: A balanced coinflip machine with bad states.

(b) Balanced unifilar hmc's.

Regarding extending the proof of Theorem 12 to the first of these classes: there do exist coinflip machines which have bad states (see Figure 3.10), so the proof cannot rely on a guarantee of visiting a bad state often. Regarding the second of these classes, every example we know of which is not a coinflip machine actually assigns probability zero to sufficiently long random strings. It would be nice to find a counterexample.

2. The Discrimination Algorithm for Hrw's checks conditional probabilities instead of just checking probabilities. This idea could have been used in the Discrimination Algorithm for $\hat{U} \cap \overline{U}$, but it wasn't necessary. We believe that this idea should generalize. It should be possible to discriminate an unknown hmc from *any* known hmc from one of these classes, given a lower bound on the *divergence* of the two hmc's (divergence is the limit as $n \to \infty$ of the relative entropy of the distributions on $\{0,1\}^n$ assigned by the two machines).

3. We would also like to extend the Discrimination Algorithm for Hrw's to be

dependent on only the *structural* parameters m and ϵ of the graph G, and not on γ. As Figure 2.1 shows in the binary case, it is possible for ϵ to be small even when N is a Bernoulli hrw. Still, if N is *not* a Bernoulli hrw, an upper bound on m and a lower bound on ϵ should be equivalent to an upper bound on $H(Y)$. These comments should extend to relative entropy and the concerns of Question 2 as well.

Our focus on the structure of the underlying Markov chains complements previous discrimination techniques based on relative entropy and universal codes. Ziv [Zi] and Gutman [Gu] have defined functions involving relative entropy and universal codes to discriminate Markov sources using empirical data in the limit.

4. We would like to find an interesting class of hmc's for which there is an efficient probability oracle algorithm for inference.

5. Since an oracle can simulate a black box and generate sample strings, our learning algorithm is *a priori* more powerful than one that relies on empirical data. It would be nice to establish a case where it is provably more powerful.

3.7 Three Problems

In this section we define three new problems which have not been addressed in our research.

Approximate Inference

Let M and N be Phmc's. Recall that this means that $Y^M = \{y_i\}^M$ and $Y^N = \{y_i\}^N$ are random processes. Define the *divergence* $D(M\|N)$ by

$$D(M\|N) = \lim_{n\to\infty} \frac{1}{n} \sum_{\{0,1\}^n} \mathrm{P}^M[w] \log_2 \frac{\mathrm{P}^M[w]}{\mathrm{P}^N[w]}.$$

Consider the following problem:

Problem: Approximate Inference. Given an unknown m-state hmc M and a parameter ϵ, find a Phmc N such that $D(M\|N) \leq \epsilon$, using probability oracle queries.

Remarks. Approximate inference is at most as hard as oracle inference in complexity, since oracle inference is just approximate inference with $\epsilon = 0$. Approximate inference is also no harder than empirical inference, assuming the empirical inference algorithm is required to converge according to divergence or some equivalent measure (see [MZ89, AW]). This is because the oracle can be used to simulate the black box, as we saw in the previous section.

It follows from Theorem 8 that there is an exponential-time probability oracle algorithm (alternatively: an NP probability oracle algorithm) to solve oracle inference (and therefore also approximate inference). This is the best known algorithm.

Inference with a Teacher

In [Ang87], Angluin gives an algorithm for learning regular sets in a *teacher-learner* setting where the examples being presented to the learning algorithm are chosen to be helpful (as opposed to Valiant's PAC learning model [V84], in which the examples

are chosen at random). Tzeng [Tz92b] defines a natural extension of Angluin's equivalence oracle for a unifilar hidden Markov chain model with no notions of stationarity or ergodicity. He extends the methods of [Ang87] to show that inference is possible in this model. The following definition of a teacher oracle is an adaptation Tzeng's equivalence oracle to hmc's.

Definition 12 (Teacher Oracle) *The* teacher oracle *receives as a query a known pseudo-hmc N and returns either (a) the statement "$N \equiv M$", if that is the case, or (b) a string $w \in \{0,1\}^*$ for which $P^N[w] \neq P^M[w]$, otherwise.*

The teacher oracle is thus able to solve discrimination. Theorem 6 showed that discrimination is reducible to inference. The following problem is the converse.

Problem: Inference with a Teacher. Given an unknown m-state hmc M, find a pseudo-hmc N such that $N \equiv M$, using probability oracle queries and teacher oracle queries.

We call an algorithm a *teacher oracle algorithm* if it can query both the probability oracle and the teacher oracle. The algorithm of [Tz92b] can be used to infer a unifilar hmc, but the hypothesis may not be an hmc. It would be nice to find a teacher oracle algorithm for general hmc's.

Consistency

Gold [Go78] has observed that inference from *given* data is potentially more difficult than inference from *requested* data. He shows that the problem of determining whether a given set of examples is consistent with any dfa of bounded size is NP-complete.

This result has been extended for dfa's by Pitt and Warmuth [PW] and Angluin [Ang88, Ang89b]. Tzeng [Tz92b] extended the methods of Gold and Angluin to show that consistency for the Markov chain model mentioned previously is also NP-complete.

Because of ergodicity and stationarity conditions, the restriction of hmc's to dfa's is trivial, so the aforementioned hardness results do not immediately translate to hmc's. Therefore we propose the following problem.

Problem: Consistency. Given a set $W \subseteq \{0,1\}^*$ and, for each $w \in W$, a probability $\mathrm{P}[w]$, determine whether there is an m-state hmc M such that for all $w \in W$, $\mathrm{P}^M[w] = \mathrm{P}[w]$.

Remarks. (i) The consistency problem differs from oracle inference in two ways. The algorithm is not an oracle algorithm; it receives a fixed set of strings that it cannot choose. In this sense consistency is more like empirical inference. Also, there is not necessarily a unique correct answer up to equivalence.

(ii) It is also interesting to consider a *restricted consistency problem*, in which W is closed under taking substrings or closed under taking prefixes. The intuition here is from empirical samples. If a given string is frequent enough in the sample that its empirical probability is reliable, then the empirical probabilities of all substrings should be reliable as well. We discuss this idea further in Section 4.1.

Chapter 4

Further Work

4.1 Empirical Problems

In this section we discuss empirical inference of hidden Markov chains. We begin by reviewing work on the maximum-likelihood estimate. We also mention alternative estimates that have been used. In the light of the results of Chapters 2 and 3 we propose two new estimates. We state the major open problems about the convergence of these estimates and the complexity of computing them.

Background

Consider the following problem:

Problem 1: Bounded Size Maximum Likelihood Estimate. Suppose an observer (an algorithm) is given a sequence $\mathbf{s} = \sigma_1, \ldots, \sigma_n$ of n 0's and 1's generated by an m-state hmc M. The problem is to construct an hmc N with at most m states which maximizes the quantity $\mathrm{P}^N[\mathbf{s}]$.

The solution N to Problem 1 is called the *maximum-likelihood estimate* of M. (Technically there will in general be more than one solution.) Let K denote the subset of m^2-dimensional Euclidean space $\mathbb{R}^{m^2}$ consisting of all pseudo-hmc's equivalent to M. The following result of Baum and Petrie [BP, Theorems 3.2 and 3.3] justifies the search for a solution to Problem 1 by showing that the maximum likelihood estimate eventually converges to the right answer.

Theorem 15 *Suppose an m-state hmc M generates a n-bit sequence* **s**. *Let N be the m-state maximum-likelihood estimate, and let K be as defined above. Then*

(1) the Euclidean distance from N to K converges to 0, and

(2) the process $\{y_i\}^N$ converges to $\{y_i\}^M$ in divergence, with probability 1,

as $n \to \infty$.

An estimate of an hmc that satisfies the convergence conditions (1) and (2) is said to be *consistent.* To avoid confusion with the consistency problem for hmc's, we will use the term *convergent* instead.

The algorithms for finding the maximum-likelihood estimate for the hidden Markov model (hmm) use a heuristic iterative procedure [LRS]. The parameters of the Markov chain are re-estimated at each step to increase the likelihood of a training sequence. There is no guarantee that such a procedure will not get stuck at a local maximum. The convergence rate appears difficult to bound analytically, and in practice is often poor [LRS].

Abe and Warmuth [AW] have given evidence of the computational difficulty of approximating the maximum likelihood estimate. They study a probabilistic automaton

(pa) model which is equivalent to the hmc model except that it does not include ergodicity or stationarity conditions. Training sequences of length n are generated by an unknown probability distribution on $\mathbb{R}^n$. Abe and Warmuth give a training algorithm which requires only a polynomial number of training sequences, but whose running time is exponential in m, the size of the pa being constructed. Furthermore, they show that unless P = NP, there is no algorithm running in time polynomial in n and the size of the alphabet which takes a single training sequence and finds a solution for which the likelihood of the training sequence is within 2^n of optimal. This is true even in the case of a two-state pa. They adapt the method of Angluin's proof in [Ang89b] that consistency is NP-complete for two-state dfa's, by extending the notion of a truth assignment for dfa's to that of an apporoximate truth assignment for pa's. According to the authors, this hardness result does not immediately translate to hmm's. It is also not clear that the methods of [AW] extend to hmc's, either, where ergodicity and stationarity may play an important role in eliminating pathological behavior.

Given the potential difficulty of computing the maximum likelihood estimate, it is natural to look for another estimate which is convergent in the sense of Theorem 15.

We propose two alternative estimates that are motivated by the results of Chapter 3. An observer is given an n-bit sequence $\mathbf{s}$ produced by an m-state hmc M, and for each string w of length less than $2m$ he measures the *empricial probability* $\mathrm{P}^M_{emp}[w]$ by taking its fraction of occurrences in $\mathbf{s}$. For a fixed parameter $\epsilon > 0$, let $A_\epsilon = \{w : |w| \leq 2m, \mathrm{P}^M_{emp}[w] \geq \epsilon\}$. Let $n_\epsilon = |A_\epsilon|$. The problems are,

Problem 2: Minimal Consistent Pseudo-Hmc. Find the smallest pseudo-hmc which agrees with all empirical probabilities $\mathrm{P}^M[w]$, $w \in A_\epsilon$.

Problem 3: Least Squares Pseudo-Hmc. Find an m-state pseudo-hmc N such that the Euclidean distance between the n_ϵ-tuples $\{\mathrm{P}^N[w] : \ w \in A_\epsilon\}$ and $\{\mathrm{P}^M_{emp}[w] : \ w \in A_\epsilon\}$ is minimized.

Remarks.

Convergence. The motivation for both problems is as follows. By Theorem 8 the probabilities of strings of length less than $2m$ characterize the behavior of M. Set $\epsilon = \min_{|w|<2m} \mathrm{P}^M[w]$. If somehow the values of $\{\mathrm{P}^M_{emp}[w] : \ w \in A_\epsilon\}$ were exactly correct, then the solution to both problems would be a pseudo-hmc of size at most m that is equivalent to M. This observation leads to the following conjecture.

Conjecture 1 *For every hmc M there exists $\epsilon > 0$ such that the solutions to Problems 2 and 3 are convergent in the sense of Theorem 15.*

The results of Chapter 2 open the possiblity of quantifying the rate of this convergence for a class of hmc's with sufficiently nice structure. To our knowledge this has not been done for the rate of convergence of maximum-likelihood estimate.

Complexity. First, it would be nice to characterize those hmc's for which n_ϵ is polynomially large in $1/\epsilon$. A rapid mixing condition on M is sufficient, but Figure 2.1 of Section 2.4 shows that it is not necessary. Having done this it makes sense to talk about the complexity of Problems 2 and 3 in terms of m

and $1/\epsilon$. Furthermore, from the above remarks on convergence it would be nice to characterize the class of hmc's M for which the "right" value of ϵ is such that $1/\epsilon$ is polynomial in m (that is, the behavior of M is determined by a set of strings of high probability).

The complexity of Problem 2 should be related to the complexity of the consistency problem restricted to a set of input strings closed under taking substrings. In fact, if v is a *prefix* of w then $\mathrm{P}^M[v] \geq \mathrm{P}^M[w]$. If v is any substring of w and $|w| < 2m$, then $\mathrm{P}^M_{emp}[v] \geq \frac{\mathrm{P}^M_{emp}[w]}{m}$. When we form the closure of A_ϵ under substrings we get a subset of $A_{\epsilon/m}$, which under the regularity assumptions just discussed should not be too much larger. The extra strings in A_ϵ may still make Problem 2 more difficult.

It would be interesting to compare the complexity of Problem 3 to that of Problem 1, since they both involve optimization over complicated geometric objects in Euclidean space.

Notes

In [MZ89, ZM92], Merhav and Ziv investigate other asymptotically good estimates of the parameters of a Markov source, based on universal data compression and the uniform cost function. Ziv [Zi], Ziv and Merhav [ZM93], and Gutman [Gu] have considered empirical discrimination based on universal data compression as well as relative entropy. Merhav and Ziv [MZ91] consider discrimination using a Bayesian estimator.

The size bound m that is imposed on the hypothesis in Problems 1 and 3 is related to Occam's Razor by way of entropy. If we allow a hypothesis of size $n = |\mathbf{s}|$, then it

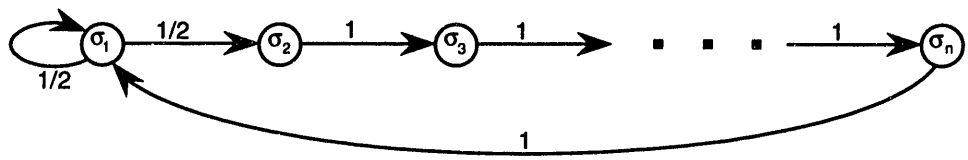

Figure 4.1: A very high likelihood hmc for $\mathbf{s} = \sigma_1, \ldots, \sigma_n$

becomes easy to construct N for which $\mathrm{P}^N[\mathbf{s}] \geq \frac{1}{n+1}$, as shown in Figure 4.1.

The problem with N as a hypothesis to account for $\mathbf{s}$ is that the entropy of the process $Y = \{y_i\}^N$ is too low: $H(Y) = \frac{2}{n+1}$, as long as $\sigma_1 \neq \sigma_2$. Occam's Razor states that until compelled, one should not impute too much structural complexity to a natural phenomenon. One interpretation of low entropy is high structural complexity. It would be interesting (though possibly impractical) to impose a lower bound on entropy instead of an upper bound on the number of states, as a limitation on the complexity of the hypothesis.

4.2 Open Questions

In this section we compile a list of open problems, arranged by subject. We refer back to the indicated section for more details.

The Chernoff Bound

1. Improving the bound (Section 2.2).

   (a) Remove the dependence of the bound on nonuniformity.

   (b) Introduce a factor of $\frac{1}{\pi(A)}$ into the exponent of the bound, by analogy to the independent case. The idea here is to show that the cost of Aldous'

procedure for estimating $\pi(A)$ depends inversely on $\pi(A)$ instead of $\pi(A)^2$.

(c) Generalize to some class of non-reversible Markov chains.

(d) Give a lower bound on the probability of deviation.

2. Improving the entropy estimate (Section 2.2).

(a) Extend the estimate to a larger class of 0-1 sources.

(b) Quantify the rate of convergence to asymptotic black box behavior for a large class of hmc's.

3. Sampling methods (Section 2.4).

(a) Quantify the advantage of sampling every point versus every τ^{th} point, in terms of entropy and expansion.

(b) Characterize those graphs for which the sample average of every point is the best estimate of $\pi(A)$.

Pseudo Hidden Markov Chains (Section 3.3)

1. Find a useful definition of ergodicity for pseudo-hmc's.

2. Characterize the class Phmc of pseudo-hmc's M for which $\mathrm{P}^M[\cdot]$ is a nonnegative function.

3. Characterize those Phmc's N for which $\{y_i\}^N$ is an ergodic random process.

4. Characterize those hmc's for which the minimal equivalent hmc is within some given bound in size of the minimal equivalent pseudo-hmc.

Oracle Problems

1. Discrimination (Sections 3.5–3.6).

   (a) Close the gap between the negative results of Section 3.5 and the positive results of Section 3.6. Find a probability oracle algorithm for discrimination on a larger class of unknown hmc's. In particular, investigate these two classes:

      i. Balanced 1/2-machines,

      ii. Balanced unifilar hmc's. (Find an hmc in this class which is not a coinflip machine but does not assign probability zero to sufficiently long random strings.

   (b) Find a probability oracle algorithm for discrimination where the known hmc does not behave like a sequence of independent labels, possibly using relative entropy.

   (c) Extend the Discrimination Algorithm for HRW's to be dependent on only the structural parameters m and ϵ of the graph G.

   (d) Find a case where discrimination with a probability oracle algorithm is more powerful than with one that relies on empirical data.

2. Inference (Sections 3.5–3.7).

   (a) Find an interesting class of hmc's for which there is an efficient probability oracle algorithm for inference.

   (b) Find a probability oracle algorithm for the approximate inference problem. Find a case where approximate inference with a probability oracle algorithm is more powerful than with one that relies on empirical data.

(c) Find a teacher oracle algorithm for inference.

3. Find a probability oracle algorithm for the consistency problem for hmc's (Section 3.7).

Empirical Problems (Section 4.1)

1. Determine the complexity of computing or approximating the maximum-likelihood estimate for hmc's.

2. Prove that the minimal consistent pseudo-hmc estimate and the least-squares pseudo-hmc estimate are convergent in the sense of Theorem 15.

3. Characterize the class of hmc's M for which the set A_ϵ of Section 4.1 is polynomially large in ϵ.

4. Determine the complexity of computing the minimal consistent pseudo-hmc and the least-squares pseudo-hmc, for the class of the previous problem.

Bibliography

[AW] N. Abe and M. K. Warmuth, "On the Computational Complexity of Approximating Distributions by Probabilistic Automata," *Machine Learning* **9** (1992), 205-260.

[Ah] L. Ahlfors, *Complex Analysis*, 2^{nd} ed., McGraw-Hill, 1966.

[AKS] M. Ajtai, J. Komlós, and E. Szemerédi, "Deterministic simulation of logspace," *Proceedings of the 19th ACM Symposium on the Theory of Computing* (1987).

[Ald87] D. Aldous, "On the Markov chain simulation method for uniform combinatorial distributions and simulated annealing," *Probability in the Engineering and Information Sciences* **1** (1987), 33-46.

[Ald88] D. Aldous, "Finite-time implications of relaxation times for stochastically monotone processes," *Probability Theory and Related Fields* **77** (1988), 137-145.

[Ald90] D. Aldous, "Bibliography: random walks on graphs," Electronic mail from aldous@stat.berkeley.edu (1990).

[AK*] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff, "Random walks, universal traversal sequences, and the complexity of maze problems," *Proceedings of the 20th Annual IEEE Symposium on the Foundations of Computer Science* (1979), 218-233.

[Alo] N. Alon, "Eigenvalues and expanders," *Combinatorica* **6** (1986), 83-96.

[AM] N. Alon and V. D. Milman, "λ_1, isoperimetric inequalities for graphs, and superconcentrators," *J. Combinatorial Theory B* **38** (1985), 73-88.

[Ana] V. Anantharam, "A large deviation approach to error exponents in source coding and hypothesis testing," *IEEE Transactions on Information Theory* **36** (1990), 938-943.

[Ang87] D. Angluin, "Learning regular sets from queries and counterexamples," *Information and Computation* **75** (1987), 87-106.

[Ang88] D. Angluin, "Queries and concept learning," *Machine Learning* **2** (1988), 319-342.

[Ang89a] D. Angluin, "Equivalence queries and approximate fingerprints," *Proceedings of the Second ACM Workshop on Computational Learning Theory* (1989), 134-145.

[Ang89b] D. Angluin, "Minimum consistent DFA problem is NP-complete," unpublished manuscript.

[AV] D. Angluin and L. G. Valiant, "Fast probabilistic algorithms for Hamiltonian circuits and matchings," *Journal of Computer and System Sciences* **18** (1979), 155-193.

[AK] D. Applegate and R. Kannan, "Sampling and integration of near log-concave functions," *Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing* (1991), 156-163.

[ALM] S. Arora, F. T. Leighton and B. Maggs, "On-line algorithms for path selection in a nonblocking network," *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing* (1990), 149-158.

[Ash] R. Ash, *Information Theory*, Interscience, 1965.

[BP] L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *Annals of athematical Statistics* **37** (1966), 1554 1563.

[Bo] K. A. Borovkov, "On a new variant of the Monte Carlo method for multidimensional integration," *Theory of Probability and Its Applications* **36** No. 2 (1991), 355-360.

[Br] A. Z. Broder, "How hard is it to marry at random? (on the approximation of the permanent)," *Proceedings of the 18th ACM Symposium on the Theory of Computing* (1986), 50-58.

[Ch] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Annals of Mathematical Statistics* **23** (1952), 493-507.

[Co] A. Cobham, "Stochastic automata with large state spaces and low rank," *Linear Algebra and Its Applications* **75** (1986), 57-66.

[CW] A. Cohen and A. Wigderson, "Dispersers, deterministic amplification, and weak random sources," *Proceedings of the Thirtieth IEEE Symposium on Foundations of Computer Science* (1989).

[CLR] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, MIT Press, 1990.

[Cr] H. Cramér, "Sur un nouveau théorème de la théorie des probabilités," *Actualités Scientifiques et Industrielles*, No. 736, Paris 1938.

[DLS] L. D. Davisson, G. Longo, and A. Sgarro, "The error exponent for the noisless encoding of finite ergodic Markov sources," *IEEE Transactions on Information Theory* **27** No. 4 (1981), 431-438.

[DS] J. Deuschel and D. Stroock, *Large Deviations*, Academic Press, Boston, 1989.

[Di] G. A. Dirac, "Some theorems on abstract graphs, *Proceedings of the London Mathematical Society* **2** (1952), 69-81.

[Du] R. M. Dudley, *Real Analysis and Probability*, Brooks/Cole, Pacific Grove, California, 1990.

[DF] M. Dyer and A. Frieze, "Computing the volume of convex bodies: as case where randomness provably helps," (1991). Preprint.

[DFK] M. Dyer, A. Frieze and R. Kannan, "A random polynomial time algorithm for approximating the volume of convex bodies," *Proceedings of the 21st Annual ACM Symposium on the Theory of Computing* (1989), 375-381.

[E] J. Edmonds, "Paths, trees, and flowers," *Canadian Journal of Mathematics* **17** (1965), 449-467.

[ERV] W. Evans, S. Rajagopalan, and U. Vazirani, "Learning an unknown randomized algorithm from its behavior," to appear in *Proceedings of the Sixth ACM Workshop on Computational Learning Theory*, 1993.

[Fe] W. Feller, *An Introduction to Probability Theory and Its Applications*, 2 volumes, 2d ed., Wiley, 1957.

[Fi] J. A. Fill, "Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the exclusion process," *The Annals of Applied Probability* **1** (1991), 62-87.

[Ge] P.-G. de Gennes, *Scaling Concepts in Polymer Physics*, Cornell University Press, Ithaca, 1979.

[Gi] E. J. Gilbert, "On the identifiability problem for functions of finite Markov chains," *Annals of Mathematical Statistics* **30** (1959), 688-697.

[Go72] E. M. Gold, "System identification via state characterizaton," *Automatica* **8** (1972), 621-636.

[Go78] E. M. Gold, "Complexity of automaton identification from given data," *Information and Control* **37** (1978), 302-320.

[GMR] D. Gillman, M. Mohtashemi, and R. Rivest, "On breaking Huffman codes," in preparation.

[GJ] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, Wiley, 1983.

[Gu] M. Gutman, "Asymptotically optimal classification for multiple tests with empirically observed statistics," *IEEE Transactions on Information Theory* **35** (1989), 401-408.

[GS] D. Gillman and M. Sipser, "Colored Markov Chains," unpublished manuscript, 1991.

[Ha] D. J. Hand, *Discrimination and Classification*, Wiley, 1981.

[Hö] T. Höglund, "Central limit theorems and statistical inference for finite Markov chains," *Z. Wahrscheinlichkeitstheorie verw. Gebiete* **29** (1974), 123-151.

[Hu] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE* **40** No. 9 (1952), 1098-1101.

[IZ] R. Impagliazzo and D. Zuckerman, "How to Recycle Random Bits," *Proceedings of the Thirtieth IEEE Symposium on Foundations of Computer Science* (1989).

[IAK] H. Ito, S.-I. Amari, and K. Kobayashi, "Identifiability of hidden Markov information sources and their minimum degrees of freedom," *IEEE Transaction on Information Theory* **38** (1992), 324-333.

[J] N. C. Jain, "Large deviation lower bounds for additive functionals of Markov processes," *Annals of Probability* **18** (1990), 1071-1098.

[JS89] M. Jerrum and A. Sinclair, "Approximating the permanent," *SIAM Journal on Computing* **18** (1989), 1149-1178.

[JS91] M. Jerrum and A. Sinclair, "Polynomial-time approximation algorithms for the Ising model," preprint. An extended abstract appeared in the "Proceedings of the International Colloquium on Automata, Languages, and Programming," Warwick, UK, July 1990; published as *Lecture Notes in Computer Science* **443**, Springer.

[JVV] M. Jerrum, L. Valiant, and V. Vazirani, "Random generation of combinatorial structures from a uniform distribution," *Theoretical Computer Science* **43** (1986), 169-188.

[Kah] N. Kahale, *Personal communication.*

[KK] N. Karzanoff and L. Khachiyan, On the conductance of order Markov chains," *DIMACS Technical Report* 90-60, 1990.

[Kat] T. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*, Springer, 1982.

[Kom] J. Komlós, *Personal communication.*

[Koo] L. H. Koopmans, "Asymptotic rate of discrimination for Markov processes," *Annals of Mathematical Statistics* **31** (1960), 982-994.

[KL] R. Karp and M. Luby, "Monte Carlo algorithms for enumeration and reliability problems," *Proceedings of the 15th ACM Symposium on the Theory of Computing* (1983).

[La] P. D. Laird, "Efficient Unsupervised Learning," *Proceedings of the First ACM workshop on Computational Learning Theory* (1988), 297-311.

[LZ] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory* **24** No. 5 (1978), 530-536.

[LRS] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *The Bell System Technical Journal* **62** (1983), 1035-1074.

[Lo] L. Lovász, *Personal communication.*

[LS] L. Lovász and M. Simonovits, "Random walks in a convex body and an improved volume algorithm," to appear in *Random Structures & Algorithms*, 1993.

[Mal] V. K. Malinovskii, "Limit theorems for Harris Markov chains," *Theory of Probability and its Applications*, Vol. 31, no. 2 (1987), 269-285.

[Mat] P. Matthews, Generating a random linear extension of a partial order, University of Maryland (Baltimore County) Technical Report, 1989.

[MZ89] N. Merhav and J. Ziv, "Estimating with partial statistics the parameters of ergodic finite Markov sources," *IEEE Transactions on Information Theory* **35** No. 2 (1989), 326-333.

[MZ91] N. Merhav and J. Ziv, "A Bayesian approach for classification of Markov sources," *IEEE Transactions on Information Theory* **37** No. 4 (1991), 1067-1071.

[Mi] H. Minc, *Permanents*, Addison-Wesley, 1978.

[Na] S. Natarajan, "Large deviations, hypothesis testing, and source coding for finite Markov chains," *IEEE Transactions on Information Theory* **31** (1985), 360-365.

[Ne] A. Nerode, "Linear automaton transformations," *Proceeding of the American Mathematical Society* **9** (1958), 541-544.

[P] A. Paz, *Introduction to Probabilistic Automata*, 1971.

[PW] L. Pitt and M. K. Warmuth, "The minimum consistent dfa problem cannot be approximated within any polynomial," *Proceedings of the 21st ACM Symposium on the Theory of Computing* (1989), 421-432.

[Ro] M. Rosenblatt, *Markov Processes. Structure and Asymptotic Behavior*, Springer, New York, 1971.

[Ru] S. Rudich, "Inferring the Structure of a Markov Chain From Its Output," *Proceedings of the Twenty-Sixth IEEE Symposium on Foundations of Computer Science* (1985), 321-326.

[Se] E. Seneta, *Nonnegative Matrices*, Wiley, 1973.

[Si] A. Sinclair, "Improved bounds for mixing rates of Markov chains and multicommodity flow," Technical Report ECS-LFCS-91-178, Dept. of Computer Science, University of Edinburgh, October 1991.

[SJ] A. Sinclair and M. Jerrum, "Approximate counting, uniform generation, and rapidly mixing Markov chains," *Information and Computation* **82** (1989), 93-113.

[SS] G. W. Stewart and J.-g. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.

[St] L. Stockmeyer, "On approximation algorithms for #P," *Proceedings of the 15th ACM Symposium on the Theory of Computing* (1983).

[Ta] R. M. Tanner, "Explicit construction of concentrators from generalized n-gons," *SIAM Journal on Algebraic Discrete Methods* **5** (1984), 287-294.

[Tu] V. F. Turchin, "On the computation of multidimensional integrals by the Monte-Carlo method," *Theory of Probability and Its Applications* **16** No. 3 (1971), 720-724.

[Tz92a] W.-G. Tzeng, "A polynomial-time algorithm for the equivalence of probabilistic automata, " *SIAM Journal on Computing* **21** No. 2 (1992), 216-227.

[Tz92b] W.-G. Tzeng, "Learning probabilistic automata and Markov chains via queries," *Machine Learning* **8** (1992), 151-166.

[V79] L. Valiant, "The Complexity of computing the permanent," *Theoretical Computer Science* **8** (1979), 189-201.

[V84] L. Valiant, "A theory of the learnable," *Communications of the ACM* **27** No. 11 (1984), 1134-1142.

[Wil] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, 1965.

[Win] P. Winkler, *Personal communication.*

[WZ] A. D. Wyner and J. Ziv, "Classification with finite-memory," preprint – in preparation.

[Y] K. Yamanishi, "Probably almost discriminative learning," *Proceedings of the Fifth ACM workshop on Computational Learning Theory* (1992), 164-171.

[Zi] J. Ziv, "On classification with empirically observed statistics and universal data compression," *IEEE Transactions on Information Theory* **34** No. 2 (1988), 278-286.

[ZM92] J. Ziv and N. Merhav, "Estimating the number of states of a finite-state source," *IEEE Transactions on Information Theory* **38** No. 1 (1992), 61-65.

[ZM93] J. Ziv and N. Merhav, "A Measure of Relative Entropy between Individual Sequences with Application to Universal Classification," preprint.

[Zu] D. Zuckerman, "A technique for lower bounding the cover time," *Proceedings of the 22nd Annual ACM Sympcsium on the Theory of Computing* (1990), 254-259.