Finite Memory Algorithms for Testing Bernoulli Random Variables

PATRICK ROBERT HIRSCHLER

Department of Electrical Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332

This paper solves the basic multiple hypothesis-testing problem with a time-varying finite-state automaton. Let $X_1, X_2, \ldots$, be a sequence of iid Bernoulli random variables with unknown parameter $p = \Pr(X_i = 1)$. The $K$-hypothesis testing problem is investigated under the following assumptions: the $X_i$'s are observed sequentially, and summarized after each new observation by an $m$-valued statistic $T_n \in \{1, \ldots, m\}$ which is updated by an algorithm of the form $T_n = f_n(T_{n-1}, X_n)$. Two automata are exhibited which make only a finite number of errors with probability one: $\mathcal{A}_1$, a $2K$-state machine resolving perfectly the $K$ simple hypotheses $H_k : p = p_k (k = 1, \ldots, K)$; and $\mathcal{A}_2$, a 4-state machine solving the difficult testing problem $H_0 : p = p_0$ versus $H_1 : p \neq p_0$. The algorithms do not require artificial randomization. The rate of convergence is related to the Kullback discrimination information between the hypotheses.

I. INTRODUCTION

Let $X_1, X_2, \ldots$, be a sequence of iid Bernoulli random variables with unknown parameter $p = \Pr(X_i = 1)$. In this paper we are interested in the following testing problems:

I. $H_k : p = p_k \quad (k = 1, \ldots, K)$,

$(0 < p_1 < p_2 \cdots < p_K < 1)$.

II. $H_0 : p = p_0$ versus,

$H_1 : p \neq p_0 \quad (0 < p_0 < 1)$.

Our intent is to specify for each of these problems an automaton which makes only a finite number of errors with probability one, under a finite memory constraint.
Discussion of the Finite Memory Algorithm

Different types have been discussed in detail by Cover (1969). Throughout this paper, we shall adopt the following terminology: a decision rule has a finite-state memory of size $m$ if it can be implemented by an $m$-state automaton. Thus when $X_i$ takes on a continuum of values a rule based on the last $m$ observations requires an infinite-state memory. Considerations in Cover (1969) have led to the following formulation: Let $T_n \in \{1, 2, ..., m\}$ represent the state of the memory at time $n$, and $d(.) : \{1, ..., m\} \rightarrow \{H_1, ..., H_K\}$ be a decision function that takes decision $d(T_n)$ at that time. An error is made if $d(T_n) \neq H_t$, where $H_t$ denotes the true hypothesis. Let

$$P^* = \lim_{n \to \infty} \Pr(d(T_n) \neq H_t)$$

be the limiting probability of error. We make the following definition: a $K$-hypothesis testing problem is $m$-state perfectly achievable if there exists an $m$-state automaton making only a finite number of errors with probability one. Time-invariant rules (i.e., $f_n \equiv f$) with a finite memory constraint do not even achieve a zero limiting probability of error (Hellman and Cover, 1970). Therefore the following family of learning algorithms is considered:

$$T_n = f_n(T_{n-1}, X_n)$$

where $X_n$ is the $n$-th observation, and $f_n$ a time-varying function: $f_n : \{1, 2, ..., m\} \times \{0, 1\} \rightarrow \{1, 2, ..., m\}$ that does not involve randomization.

History of the Problem

The formulation of the finite-memory constraint as given previously was introduced by Cover (1969). In that paper the two-hypothesis testing problem $H_0 : \rho = p_0$ versus $H_1 : \rho = p_1$ is shown to be 4-state achievable. In addition the problem of testing if the bias of a coin $\rho$ is less than or greater than a fixed value $p_0$ is solved using a time-varying rule and a 4-state memory. Following the same technique and using randomization, Sengupta (1969) exhibits a rule that gives $\max(p_1, ..., p_k)$ as the limiting frequency of heads, with probability one, where the values $p_1, ..., p_k$ are known but it is not known which $\rho$ corresponds to which coin. This latter result is an extension to the case of $k$ coins of Cover's solution to the "Two-Armed Bandit" problem with finite memory (Cover, 1968).

Results

In this paper two finite-memory algorithms are exhibited which require no artificial randomization, and make a finite number of errors wp1 under
any hypothesis. \( K \) simple hypotheses \( H_k : p = p_k (k = 1, \ldots, K) \) on a Bernoulli random variable are considered first. Section 2 gives one algorithm (Theorem 1) that can resolve perfectly these hypotheses, and relates the rate of convergence of the procedure to the discrimination information between the hypotheses. In Section 3, Theorem 2 proving that the testing problem \( p = p_0 \) versus \( p \neq p_0 \) is 4-state achievable constitutes a result of primary importance. Finally the testing technique is interpreted, and the results are discussed. We now conclude this introduction by describing a comparison procedure used in the sequel.

**Comparison Procedure**

Let \( B = (B_1, \ldots, B_n) \) and \( P = (P_1, \ldots, P_n) \) be two finite sequences of binary digits \( B_i, P_i \in \{0, 1\} \). This procedure compares the block \( B \) to the pattern \( P \) using only 1 bit of memory \( Q \in \{0, 1\} \) as follows: \( Q \) is set automatically to 1 when the procedure is started, so that \( Q_0 = 1 \). If \( B_1 = P_1 \) then \( Q_1 = Q_0 \), and subsequently if \( B_i = P_i \), then \( Q_i = Q_{i-1} \) otherwise \( Q_i = 0 \) \( (i = 1, \ldots, n) \). Let \( Q(B, P) \) be the last value \( Q_n \) of \( Q \). It is clear that \( Q(B, P) = 1 \) iff \( B \) and \( P \) are identical.

In the following sections, the \( B_i \)'s correspond to the incoming observations whereas \( P \) represents a preselected pattern.

**Notation.**  
\([a] \) is the largest integer less than or equal to \( a \).  
\([a] \) is the smallest integer greater than or equal to \( a \).  
\( wp \) stands for "with probability"

**II. Test of \( K \) Simple Hypotheses**

Consider a sequence of coin tosses with unknown bias \( p = \Pr(\text{Heads}) \), and \( K \) distinct numbers \( p_1, \ldots, p_K \) satisfying \( 0 < p_1 < p_2 < \cdots < p_K < 1 \).

**Theorem 1.** Let \( X_1, X_2, \ldots \), be a sequence of iid Bernoulli random variables with \( \Pr(X_i = 1) = p \). The \( K \)-hypothesis testing problem

\[
H_k : p = p_k \quad (k = 1, 2, \ldots, K)
\]

is \( 2K \)-state perfectly achievable.

**Proof.** Let the memory consist of the pair \((T, Q)\) where \( T \in \{1, \ldots, K\} \) and \( Q \in \{0, 1\} \). Consider \( K \) sequences \( \{t_k^i\}_{i=1}^{\infty} (k = 1, \ldots, K) \) of positive integers, and \( P_k^{t_k} \) a pattern of length \( t_k \) defined as a sequence of \( \lceil p_k t_k \rceil \)'s
followed by \([q_k^{i} t_k^{i}]\) 0's. Now divide the sequence of observations into successive blocks:

\[
\begin{array}{cccccccc}
B_1^1 & B_2^1 & \cdots & B_K^1 & B_1^2 & \cdots & B_K^2 & \cdots & B_k^i & \cdots \\
\end{array}
\]

\textbf{FIGURE 1}

where \(B_k^i\) has length \(t_k^i\). Let \(\mathcal{M}_1\) be the automaton described by the program:

- Start: \(i := 2;\)
- Cycle: \(i := i + 1;\)
  - \(k := 0;\)
- Test: \(k := k + 1;\)
  - If \(Q(B_k^i, P_k^i) = 1\), set \(T_n = k;\)
  - If \(Q(B_k^i, P_k^i) = 0\), \(T_n\) stays unchanged;
  - If \(k < K\), go to Test;
  - Go to Cycle; End.

Observe that \(B_k^i\) checks for the pattern \(P_k^i\). At the end of the block, \(Q(B_k^i, P_k^i) = 1\) iff the test is successful. If the pattern \(P_k^i\) has occurred then the memory \(T_n\) is updated to state \(k\), index of the currently favored hypothesis. At each cycle \(i\), this test is performed for all successive values of \(k\) after which a new cycle \(i + 1\) is started. Thus the updating of the statistics \(T_n\) occurs only at the end of a test block.

\textit{Asymptotic Behavior of} \(\mathcal{M}_1\)

Let \(\alpha_k^i \triangleq \Pr (\text{Transit to state } k \text{ at cycle } i)\), and \(A_k \triangleq \sum_{i=1}^{\infty} \alpha_k^i (k = 1, \ldots, K)\).

Suppose that the following is true for \(j = 1, \ldots, K\).

\[
\text{Under } H_j \left\{ \begin{array}{c}
A_j = \infty, \\
A_k < \infty, \quad \forall k \neq j.
\end{array} \right.
\]

(2.1)

By use of the Borel–Cantelli lemma (Loève, \textit{Probability Theory}, p. 228) we conclude that the automaton transits infinitely often wp1 to state \(j\), and only finitely often wp1 to any other state \(k\). That is \(T_n \to j\) wp1, and the procedure makes wp1 only a finite number of mistakes under any hypothesis. Thus the \(K\)-simple hypothesis testing problem is solved if we can demonstrate the existence of \(K\) sequences \(\{t_k^i\}_{i=1}^{\infty} (k = 1, \ldots, K)\) satisfying (2.1).

Let \(t_k^i\) be the integer such that

\[
\log(H_k) \left( \frac{1}{i} \right) \leq t_k^i < 1 + \log(H_k) \left( \frac{1}{i} \right), \quad \text{where } H_k \defeq p_k^{q_k} d_k^{q_k}. \quad (2.2)
\]
A transition from state $k'$ to a state $k \neq k'$ is performed at cycle $i$ iff $Q(B_k^i, P_k^i) = 1$. Thus, $\alpha_k^i = p_k, t_k^{i-1} q_k, t_k^{i-1}$. Let

$$r_k(p) \triangleq (p_k \log p + q_k \log q)/(p_k \log p + q_k \log q).$$

The two inequalities

$$p_k t_k^i - 1 \leq [p_k t_k^i] \leq p_k t_k^i,$$  \hfill (2.3)

$$q_k t_k^i \leq [q_k t_k^i] \leq q_k t_k^i + 1,$$  \hfill (2.4)

imply

$$q(p^n q_k t_k^i) \leq \alpha_k^i \leq \frac{1}{p} (p^n q_k t_k^i),$$  \hfill (2.5)

and by (2.2) we have

$$qp^n q_k \left(\frac{1}{i}\right)^{r_k(p)} \leq \alpha_k^i \leq \frac{1}{p} \left(\frac{1}{i}\right)^{r_k(p)}.$$  \hfill (2.6)

Therefore $A_k$ has same convergence characteristics as $\sum_{i=1}^{\infty} (1/i)^{r_k(p)}$. Under $H_j : p = p_j \Rightarrow r_j(p_j) = 1 \Rightarrow A_j = \infty$, whereas

$$r_k(p_j) > 1 \quad \text{for} \quad k \neq j \Rightarrow A_k < \infty.$$

This completes the proof of theorem 1.

**Rate of Convergence**

In the steady state the probability $\mu_k^j(i)$ of being under $H_j$ in state $k$ at cycle $i$ satisfies the difference equation:

$$\mu_k^j(i + 1) = \mu_k^j(i) \left[1 - \sum_{h=1}^{K} \alpha_h^j(i)\right] + \sum_{h=1}^{K} \mu_h^j(i) \alpha_k^j(i);$$  \hfill (2.7)

or equivalently,

$$\mu_k^j(i + 1) = \mu_k^j(i) \left[1 - \sum_{h=1}^{K} \alpha_h^j(i)\right] + \alpha_k^j(i).$$  \hfill (2.8)

From Eq. 2.8, it can be shown that $\mu_k^j(i) (k \neq j)$ converges to 0 like

$$1/i^{\rho_j-1} \quad \text{if} \quad \rho_j < 2,$$

$$\log i/i \quad \text{if} \quad \rho_j = 2,$$

$$1/i \quad \text{if} \quad \rho_j > 2,$$

(\text{where} \quad j = 1, \ldots, K)
where we define
\[ \rho_j \triangleq \min_{h \in F_j} \{ r_h(p_j) \}, \quad (j = 1, \ldots, K). \] (2.10)

Introducing the self entropy function \( s(p) = -p \log p - q \log q \), and Kullback discrimination information
\[ J(p : p') = p \log(p/p') + q \log(q/q'), \]
we can state the following (let \( K = 2 \) for simplicity): Under \( H_1, P(e_i | H_1) \) converges to zero like
\[ \frac{1}{i^{r_p(p_0)^{-1}}}, \quad \text{if } J(p_2 : p_1) < s(p_2); \]
\[ \log \frac{i}{i}, \quad \text{if } J(p_2 : p_1) = s(p_2); \]
\[ \frac{1}{i}, \quad \text{if } J(p_2 : p_1) > s(p_2). \] (2.11)

In other words, the larger the discrimination information in favor of the alternative hypothesis \( H_2 \) against the true hypothesis \( H_1 \), the faster will the automaton converge to the right decision. However, the rate of convergence does not exceed \( 1/i \) in any case.

III. THE POINT TEST

We assume that the parameter \( p \) can take any value between 0 and 1. We investigate in this section the following testing problem \( H_0: p = p_0 \) versus \( H_1: p \neq p_0 \) assuming \( 0 < p_0 < 1 \), and prove the more difficult result.

**Theorem 2.** Let \( X_1, X_2, \ldots \) be a sequence of iid Bernoulli random variables with \( \Pr(X_i = 1) = p \). The two-hypothesis testing problem \( p = p_0 \) versus \( p \neq p_0 \) is 4-state perfectly achievable.

**Proof.** Let the memory consist of \( (T, Q) \) where \( T, Q \in \{0, 1\} \). Let \( \{r_i\}^\infty, \{s_i\}^\infty, \{t_i\}^\infty \) be three sequences of positive integers, and let \( R_i, S_i, T_i \) be three patterns consisting of, respectively, \( r_i \) consecutive zeroes, \( s_i \) consecutive ones and \( |p_0 t_i| \) zeros followed by \( |p_0 t_i| \) zeroes. Divide the sequence of observations into blocks \( B_i^1, B_i^2, B_i^3 \) of same lengths as \( R_i, S_i, T_i \) as follows:

\[
\begin{array}{ccc}
| & B_i^1 & | B_i^2 & | B_i^3 |
\end{array}
\]
Let $\mathcal{M}_2$ be the automaton described by the program $P_2$:

\begin{align*}
\text{Start} & \quad i := 1 \\
\text{Cycle} & \quad i := i + 1 \\
& \quad \text{If } Q(B_i, R_i) = 1, \quad \text{set } T = 1; \\
& \quad \text{If } Q(B_i, S_i) = 1, \quad \text{set } T = 1; \\
& \quad \text{If } Q(B_i, T_i) = 1, \quad \text{set } T = 0; \\
& \quad \text{Go to cycle;}
\end{align*}

End.

In other words, each cycle consists of three tests: for $p < p_0$, for $p > p_0$ and for $p = p_0$. A success in the first or the second test results in the updating of the memory $T$ which is then set to 1. A success of the third test results in the updating of $T$ to 0.

Suppose that under $H_j(j = 0, 1)$, the probability of error satisfies

$$
\sum_{i=1}^{\infty} \Pr(e_i/H_j) < \infty.
$$

(3.1)

Then by use of the Borel–Cantelli Lemma the automaton makes only a finite number of errors wp1. We proceed to show that the sequences $r_i, s_i, t_i$ can be chosen so as to satisfy (3.1). Define

$$
\rho_i \triangleq 1 + (1 + \epsilon) \log \log i/\log i \quad \text{for some } \epsilon > 0,
$$

(3.2)

and let

$$
\rho_i \triangleq r_i^{1/\rho_i}, \quad q_i \triangleq q_i^{1/\rho_i}, \quad H_0 \triangleq p_0^{q_0/q_0}.
$$

(3.3)

We make the following choice of the three sequences:

$$
r_i \triangleq \log_{\rho_i}(1/i),
$$

(3.4)

$$
s_i \triangleq \log_{q_i}(1/i),
$$

(3.5)

$$
t_i \triangleq \log_{H_0}(1/i).
$$

(3.6)

Under $H_1$, $p \not= p_0$ and $\log_{H_0}(p_0^{q_0/q_0}) > 1$. Hence the third test will be passed finitely often wp1. Whereas one of the two first tests will be successful infinitely often wp1 since $\sum_{i=1}^{\infty} q_i^{1/\rho_i}$ or $\sum_{i=1}^{\infty} p_i^{1/\rho_i}$ diverges. Under $H_0$, $p = p_0$. Since $\log_{H_0}(p_0^{q_0/q_0}) = 1$, the third test is passed infinitely often wp1. On the other hand

$$
q_i^{1/\rho_i} = p_i^{1/\rho_i} = 1/i(\log i)^{1+\epsilon}.
$$

(3.7)
This implies
\[ \sum_{i=1}^{\infty} (q_0^i + p_0^i - p_0^i q_0^i) < \infty, \quad (3.8) \]
and consequently neither of the first two tests if passed infinitely often wp. 1. This completes the proof of Theorem 2.

**EXAMPLE.** Borel defined a normal number in the base \( b \) as a number such that the limiting relative frequency of each digit in the base \( b \) expansion is equal to \( 1/b \). Let \( b = 2 \). Theorem 2 shows that 2 bits of memory are sufficient to test if a number is normal or not wp.1.

### IV. Conclusions

Now that the two problems have been solved, let us underline the main ideas involved in the testing procedure. If one desires to resolve the hypotheses with a finite number of errors wp1, it is necessary to remember an infinite number of observations. This appears impossible under a finite memory constraint. A single observation contains only a finite amount of information. In order to obtain events of arbitrarily high information, the first idea is to compound experiments; i.e., to base the decisions taken on blocks rather than individual observations. The events considered are the appearance of specific patterns of 1's and 0's. Each pattern is matched to a hypothesis in the sense that it possesses the right proportion of 1's. The patterns adopted are sequences of 1's followed by 0's; it is clear that such a choice is arbitrary; any sequence with the right proportion of 1's is satisfactory as long as its structure is preassigned. However, the length of each pattern is critical; patterns too long would occur too infrequently to ensure convergence, whereas patterns too short would not be meaningful enough to guarantee achievability.

In other words, at time \( n \) different models of the series are advanced and confronted with experience. If the prediction of the event is correct, agreement is recorded in the immediate memory \( Q \), while the decision to retain this hypothesis updates the permanent memory \( T \).

The most significant result is Theorem 2; it shows the somewhat surprising fact that infinite precision problems and finite memory scheme are totally compatible.

These techniques have been successful in solving a much broader class of testing problems. This will be the object of a forthcoming publication.
ACKNOWLEDGMENTS

I wish to thank Tom Cover for many ideas and the stimulating discussions he provided. I also wish to thank the referee for his detailed comments and suggestions.

RECEIVED: August 2, 1972; REVISED: March 27, 1973

REFERENCES


