ECONOMETRICA

JOURNAL OF THE ECONOMETRIC SOCIETY

An International Society for the Advancement of Economic Theory in its Relation to Statistics and Mathematics

http://www.econometricsociety.org/

Econometrica, Vol. 77, No. 1 (January, 2009), 93–105

THE COMPLEXITY OF FORECAST TESTING

LANCE FORTNOW
Northwestern University, Evanston, IL 60208, U.S.A.

RAKESH V. VOHRA
Kellogg Graduate School of Management, Northwestern University, Evanston, IL 60208, U.S.A.

Econometrica, Vol. 77, No. 1 (January, 2009), 93–105

THE COMPLEXITY OF FORECAST TESTING

BY LANCE FORTNOW AND RAKESH V. VOHRA[1]

Consider a weather forecaster predicting a probability of rain for the next day. We consider tests that, given a finite sequence of forecast predictions and outcomes, will either pass or fail the forecaster. Sandroni showed that any test which passes a forecaster who knows the distribution of nature can also be probabilistically passed by a forecaster with no knowledge of future events. We look at the computational complexity of such forecasters and exhibit a linear-time test and distribution of nature such that any forecaster without knowledge of the future who can fool the test must be able to solve computationally difficult problems. Thus, unlike Sandroni's work, a computationally efficient forecaster cannot always fool this test independently of nature.

KEYWORDS: Forecast testing, prediction, bounded rationality.

1. INTRODUCTION

SUPPOSE ONE IS ASKED to forecast the probability of rain on successive days. Sans knowledge of the distribution that governs the change in weather, how should one measure the accuracy of the forecast? One criterion for judging the effectiveness of a probability forecast, called calibration, has been an object of interest. Dawid (1982) offered the following intuitive definition of calibration:

> Suppose that, in a long (conceptually infinite) sequence of weather forecasts, we look at all those days for which the forecast probability of precipitation was, say, close to some given value ω and (assuming these form an infinite sequence) determine the long run proportion p of such days on which the forecast event (rain) in fact occurred. The plot of p against ω is termed the forecaster's *empirical calibration curve*. If the curve is the diagonal $p = \omega$, the forecaster may be termed (empirically) *well calibrated*.

Foster and Vohra (1993) exhibited a randomized forecasting algorithm that with high probability will be calibrated on all sequences of wet–dry days. Thus, a forecaster with no meteorological knowledge would be indistinguishable from one who knew the distribution that governs the change in weather.[2] This has inspired the search for tests of probability forecasts that can distinguish between a forecaster who knows the underlying distribution of the process being forecast and one who simply "games" the test.

A test takes as input a forecasting algorithm and a sequence of outcomes, and after some period accepts the forecast (PASS) or rejects it (FAIL). Sandroni (2003) proposed two properties that such a test should have. The first is that the test should declare PASS/FAIL after a finite number of periods. This seems unavoidable for a practical test. Second, suppose the forecast is indeed

[1]Research supported in part by NSF grant ITR IIS-0121678. We thank A. Sandroni and W. Olszewski and the referees for useful comments.

[2]Lehrer (2001), Sandroni, Smorodinsky, and Vohra (2003), and Vovk and Shafer (2005) gave generalizations of this result.

 DOI: 10.3982/ECTA7163

correct, that is, accurately gives the probability of nature in each round. Then the test should declare PASS with high probability. We call this second condition *passing the truth*. Call a test that satisfies these two conditions a *good* test. A test based on calibration is an example of a good test. A forecaster with no knowledge of the underlying distribution who can pass a good test with high probability on all sequences of data is said to have *ignorantly* passed the test. For every good test, Sandroni (2003) showed that there exists a randomized forecasting algorithm that will ignorantly pass the test. Therefore, no good test can distinguish between a forecaster who knows the underlying distribution of the process being forecast from one who simply games the test. Since this randomized forecast can pass the test for all distributions, it must be independent of the underlying (if any) distribution, being forecasted. Hence, in some sense these forecasts provide no information at all about the process being forecasted.

Dekel and Feinberg (2006) as well as Olszewski and Sandroni (2007) got around the impossibility of result of Sandroni (2003) by relaxing the first property of a good test; for example, allowing the test to declare PASS/FAIL at infinity, allowing the test to declare FAIL in a finite number of periods but PASS at infinity, or relaxing the condition that the test always passes the truth. These tests can often be made efficient in the sense that they can run in time linear in the length of the current sequence, but the number of forecasts before a bad forecaster is failed could be extremely large as a function of the forecaster.

Olszewski and Sandroni (2008) have noted that the tests considered by Dekel and Feinberg (2006) and Olszewski and Sandroni (2007) rely on counterfactual information. Specifically, the test can use the predictions the forecast would have made along sequences that did not materialize because the test has access to the forecasting algorithm itself. As noted by Olszewski and Sandroni (2008), this is at variance with practice. For this reason they considered tests that are not permitted to make use of counterfactual predictions on the part of the forecaster, but relaxed the condition that the test must decide in finite time. Formally, two different forecasting algorithms that produce the same forecast on a realization must be treated in the same way. If such tests pass the truth with high probability, they show that for each such test, there is a forecasting algorithm that can ignorantly pass the test.

It is natural to ask if a test, using a proper scoring rule[3] like *log-loss*, can circumvent these difficulties. Here one penalizes the forecaster $\log p$ if the forecaster predicts a probability p of rain and it rains, and a penalty of $\log(1-p)$ if it does not rain. The lowest possible score that can be obtained is the long-run average entropy of the distribution. One could imagine the test passing the forecaster if the log-loss matches the entropy. However, such a test would need

[3]Assuming the forecaster is compensated on the basis of the scores associated with the rule, a proper scoring rule gives the forecaster the incentive to reveal his/her true beliefs. See Good (1952).

to know the entropy of the distribution. As noted in the Introduction, we are concerned with tests which operate without any prior knowledge of the distribution. Proper scoring rules are good methods to compare two forecasters, but are not useful for testing the validity of a single forecaster against an unknown distribution of nature.

1.1. *Computationally Bounded Forecasters*

This paper will examine the consequences of imposing computational limits on both the forecaster and the test. We measure the complexity as a function of the length of the history so far.

Most practical tests have a complexity that is polynomial in the length of the history, so it seems reasonable to restrict attention to good tests that have a complexity that is polynomial in the length of the history. Restricting the test in this way should make it easier to be ignorantly passed. It seems natural to conjecture that for every polynomial-time good test, there exists a polynomial-time randomized forecasting algorithm that will ignorantly pass the test. Remarkably, this is not the case. We exhibit a good linear-time test that would require the forecaster to factor numbers under a specific distribution or fail the test. The existence of an efficient (i.e., probabilistic polynomial-time) algorithm for factoring composite numbers is considered unlikely. Indeed, many commercially available cryptographic schemes are based on just this premise. This result suggests that the "ignorant" forecaster of Sandroni (2003) must have a complexity at least exponential in n. Hence, the ignorant forecaster must be significantly more complex than the test. In particular, its complexity may depend on the complexity of nature's distribution.

To prove this result, we interpret the observed sequence of 0–1's as encoding a number followed by a list of its possible factors. A sequence that correctly encodes a list of factors is called correct. The test fails any forecaster who does not assign high probability to these correct sequences when they are realized. Consider now the distribution that puts most of its weight on correct sequences. If the forecaster can ignorantly pass the test, he/she must be able to identify sequences that correspond to correct answers to our computational question.

The factoring proof does not generalize to all NP problems, because we need a unique witness so as to guarantee that the test always passes the truth. Witness reduction techniques as in Valiant and Vazirani (1986) do not appear to help.

Our second result strengthens the previous one by exhibiting a good test that requires the forecaster to solve PSPACE-hard problems[4] by building on the structure of specific interactive proof systems. While the latter result is strictly stronger than the factoring result, we present both since the factoring proof

[4]See Section 5 of this paper for a definition. This class contains the more well known class of NP-hard problems.

is much simpler and illustrates some of the techniques used in the PSPACE proof. In both cases the tests are deterministic. Furthermore, they use only the realized outcomes and forecasts to render judgement.

We can modify the factoring proof to use efficiently sampleable distributions of nature. Whether we can do the same for the more general PSPACE result remains open.

In addition we also consider the possibility that the test may have more computational power than the forecaster. If we restrict ourselves to forecasters using time $O(t(n))$, there is a test T using time $O(n^{O(1)}t(n))$ with the following properties.

- For all distributions of nature μ, T will pass, with high probability, a forecaster forecasting μ.
- For some distribution τ of nature, for every forecaster F running in time $O(t(n))$, T will fail F with high probability.

If one takes a highly noncomputable distribution τ, a forecaster would not be able to forecast τ well, but T could not test this in general if T is also required to always pass the truth. In a nutshell, no forecasting algorithm can ignorantly pass a good test that is more complex than itself.

In Section 2 we give formal definitions of forecasters and tests. Section 3 shows that the forecaster needs to be nearly as powerful as the test. Section 4 gives a simple test where a successful ignorant forecasters must be able to factor. In Section 5 we sketch a more complicated test that checks if the forecaster successfully acts like a prover in an interactive proof system. This section will also provide a brief description of an interactive proof system and its implications.

2. DEFINITIONS

Let N be the set of natural numbers. Let $S = \{0, 1\}$ be the state space.[5] An element of S is called an *outcome*. Let S^n, for $n \in N$, be the n-Cartesian product of S. An n-sequence of outcomes is denoted $s = (s_1, s_2, \ldots, s_n) \in S^n$, where s_i denotes the state realized in period i. Given $s \in S^n$ and $r < n$, let $s^r = (s_1, s_2, \ldots, s_r) \in S^r$ be the prefix of length r of s.

An element of $[0, 1]$ is called a *forecast* of the event 1. A forecast made at period r refers to outcomes that will be observed in period $r + 1$. Let Δ^* be the set of probability distributions over $[0, 1]$. A forecasting algorithm is a function

$$F: \bigcup_{r=0}^{n-1} (S^r \times [0, 1]^r) \to \Delta^*.$$

At the end of each stage $r < n$, an r-history $(s^r, f_0, f_1, \ldots, f_{r-1}) \in S^r \times [0, 1]^r$ is observed. Here $f_j \in [0, 1]$ is the forecast made by F in period j. Let

[5]The results can easily be extended to more than two states.

$f^r = (f_0, \ldots, f_r)$. Based on this r-history, the forecaster must decide which forecast $f_r \in [0, 1]$ to make in period r. The forecaster is allowed to randomize, so $f_r \in [0, 1]$ can be selected (possibly) at random, using a probability distribution in Δ^*.

An n-outcome sequence $s \in S^n$ and a forecasting algorithm F determine a probability measure $\bar{F}^s$ on $[0, 1]^n$, where, conditional on (s^r, f^{r-1}), the probabilities of forecasts next period are given by $F(s^r, f^{r-1})$. The vector of realized forecasts associated with F on a sequence s will be denoted $f(s)$.

Denote the unknown data generating process by P. Given P and $s^r \in [0, 1]^r$, let $P_{s^r} \in [0, 1]$ be the probability that $s_{r+1} = 1$ conditional on s^r. Given P, let $F^P(s) \in [0, 1]^n$ be the forecast sequence such that $f_r^P(s) = P_{s^r}$.

A *test* is a function $T : S^n \times [0, 1]^n \to \{0, 1\}$. After a history of n forecasts and outcomes are observed, a test must either accept (PASS) or reject (FAIL) the forecast. When the test returns a 0, the test is said to FAIL the forecast based on the outcome sequence. When the test returns a 1, the test is said to PASS the forecast based on the outcome sequence.

A test is said to pass the truth with probability $1 - \varepsilon$ if

$$\Pr_P\big(\{s : T(s, F^P(s)) = 1\}\big) \geq 1 - \varepsilon$$

for all P.

A test T can be *ignorantly* passed by a forecasting algorithm F with probability $1 - \varepsilon$ if for all P,

$$\Pr_P\big(\{s : T(s, f(s)) = 1\}\big) \geq 1 - \varepsilon.$$

Equivalently, for all $s \in S^n$,

$$\Pr\big(T(s, f(s)) = 1\big) \geq 1 - \varepsilon,$$

where the probability is with respect to the forecaster's randomness. Therefore, F can ignorantly pass T if on any sequence of outcomes, the realized forecast sequence will not be failed with probability at least $1 - \varepsilon$ (under the distribution induced by the forecasting scheme). A test T is said to fail the forecasting algorithm F on the distribution Q with probability $1 - \varepsilon$ if

$$\Pr_Q\Big(\Big\{s : \Pr_{\bar{F}^s}\big(\{T(s, f(s)) = 1\}\big) \geq 1 - \varepsilon\Big\}\Big) \leq \varepsilon.$$

THEOREM 1—Sandroni's Theorem: *Suppose test T passes the truth with probability $1 - \varepsilon$. Then there is a forecasting algorithm F that can ignorantly pass T with probability $1 - \varepsilon$.*

3. TESTS MORE COMPLEX THAN THE FORECAST

We show that no forecasting algorithm can ignorantly pass a test with probability $1-\varepsilon$ that is more complex than itself. The basic idea of the proof can be found in Dawid (1985). For notational simplicity our exposition will be limited to deterministic forecasts. The extension to randomized forecasts is straightforward and we outline it at the end of the section.

Let $t(\cdot)$ be a time-constructible function and let $p(\cdot)$ be a polynomial.[6] The results here and later in the paper hold only for sufficiently large n, that is, $n > n_0$, where n_0 can depend on the forecaster and ε. For small n, a forecaster could be hard-wired with enough information to fool any test that passes the truth.

LEMMA 1: *For every deterministic forecaster F using time $t(n)$ there is a distribution P, polynomial p, and test T, using time $p(n)t(n)$, so that for all $\varepsilon > 0$ and n sufficiently large*:

(i) *T passes the truth with probability at least $1-\varepsilon$.*

(ii) *T fails F on P with probability $1-\varepsilon$.*

PROOF: Let n be sufficiently large and let a sequence $s^* = (s_1^*, s_2^*, s_3^*, \ldots, s_n^*)$ of outcomes be such that the probability F assigns to seeing s_j^* given $(s_1^*, \ldots, s_{j-1}^*)$ and f^{j-2} is at most $1/2$ for all $j \le n$.

To describe the test, let G be any forecasting algorithm and let $(s, g(s))$ be an n-history.

- If $s \neq s^*$, then $T(s, g(s)) = 1$.
- If $s = s^*$ and the probability that G assigns to s^* is less than ε, then $T(s, g(s)) = 1$.
- If $s = s^*$ and the probability that G assigns to s^* is at most ε, then $T(s, g(s)) = 0$.

Observe that the truth is failed if and only if s^* is realized and the probability assigned by the truth to this event is less than ε. Since this can only happen with probability less than ε, it follows that the test passes the truth with probability $1-\varepsilon$.

Now we exhibit a distribution P such that $\Pr_P(\{s : T(s, f) = 1\}) = 0$. Let P be the distribution that puts measure 1 on s^*. The probability that F assigns to $s = s^*$ is less than $(1/2)^n \le \varepsilon$. Hence, the test fails F on P with probability $1-\varepsilon$. *Q.E.D.*

LEMMA 2: *Suppose m deterministic forecasting algorithms, $F_1, F_2, \ldots, F_m$, that each use time $t(n)$. Then there is a distribution P, polynomial p, and test T that uses $mp(n)t(n)$ time so that for all $\varepsilon > 0$ and n sufficiently large*:

[6] A time-constructible function $t(n)$ is one which can be constructed from n by a Turing machine in time order $t(n)$. Formally, there exists a Turing machine M which, given a string of n 1's, returns the binary representation of $t(n)$ in $O(t(n))$ steps. All natural functions are time constructible.

(i) *T passes the truth with probability* $1-\varepsilon$.
(ii) *T fails all* $F_1, F_2, \ldots, F_m$ *on P with probability* $1-\varepsilon$.

PROOF: Consider the sequence of integers $1, 1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5, \ldots$. Let $h(j)$ be the jth element of the sequence. Let n be sufficiently large and let a sequence $s^* = (s_1^*, s_2^*, s_3^*, \ldots, s_n^*)$ of outcomes be such that the probability $F_{h(j)}$ assigns to seeing s_j^* given $(s_1^*, \ldots, s_{j-1}^*)$ and $f_{h(j)}^{j-2}$ is $f_{h(j)}(s_j^* | s_1^*, \ldots, s_{j-1}^*, f_{h(j)}^{j-2}) \leq 1/2$ for all $j \leq n$. If $h(j) > m$, let $s_j^* = 0$.

To describe the test, let G be any forecasting algorithm and let $(s, g(s))$ be an n-history.

- If $s \neq s^*$, then $T(s, g(s)) = 1$.
- If $s = s^*$ and the probability that G assigns to s^* is at least ε, then $T(s, g(s)) = 1$.
- If $s = s^*$ and the probability that G assigns to s^* is at most ε, then $T(s, g(s)) = 0$.

Observe that the truth is failed if and only if s^* is realized and the probability assigned by the truth to this event is less than ε. Since this can only happen with probability less than ε, it follows that the test passes the truth with probability $1-\varepsilon$.

Let P be the distribution that puts measure 1 on s^*. Choose any F_k. Consider the subsequence $\{s_j^*\}_{h(j)=k}$ of s^*. The probability that F_k assigns to this subsequence is at most $(1/2)^{|\{j : h(j)=k\}|} < \varepsilon$. Hence, the failure condition in the part third of the test holds. *Q.E.D.*

THEOREM 2: *For any* $t(n)$ *there is a polynomial* p *and a test* T *of complexity* $p(n)t(n)$ *with the following properties*.

(i) *For all n sufficiently large the test passes the truth with probability* $1-\varepsilon$.
(ii) *There is a distribution P on infinite* 0–1 *sequences such that each forecaster F of complexity* $t(n)$ *fails the test on P with probability* $1-\varepsilon$ *for all* $n \geq n_F$, *where* n_F *depends on the forecaster F*.
(iii) *P is independent of the forecaster and n*.

PROOF: Let $F_1, F_2, \ldots$ be an enumeration of all $t(n)$-computable forecasts. Note that the sequence s^* defined in Lemma 2 does not depend on n. Let P be the distribution that puts all its weight on the sequence s^*. For each n the test we require coincides with the test as defined in Lemma 2. This test will pass the truth with high probability for every n. For any $t(n)$ computable test F_k, by Lemma 2, for sufficiently large n, the test will fail F_k on P with probability $1-\varepsilon$. *Q.E.D.*

To extend the results to randomized forecasts it suffices to show how the proof of Lemma 1 can be modified. Recall that the sequence s^* was chosen so that the forecasted probability assigned by the forecast to s_j^* (given the previous history) is less than $1/2$. For a randomized forecast, we choose s_j^* so as to

maximize the probability of choosing a forecast of less than $1/2$ of observing event s_j^*.

4. FORECASTS THAT FACTOR

In this section we describe a test that always passes the truth and any forecaster who could ignorantly pass this test must be able to factor any number. We prove our results for deterministic forecasters, but one can extend, in a manner similar to Section 3, the results to randomized forecasters.

Given an integer k and a 0–1 sequence s, we can interpret the sequence as encoding an arbitrary tuple of numbers. The first number in the tuple we shall call the prefix of the sequence. The remaining numbers (the suffix) will be interpreted as a potential factorization of the prefix. A sequence that encodes the tuple $(m, \pi_1, e_1, \pi_2, e_2, \ldots, \pi_k, e_k)$ is said to encode the unique factorization of m if (i) $\pi_1, \pi_2, \ldots, \pi_k$ are primes, (ii) $\pi_1 > \pi_2 > \cdots > \pi_k > 1$, and (iii) $m = \pi_1^{e_1}\pi_2^{e_2}\cdots\pi_k^{e_k}$.

To describe the test, let G be any forecasting algorithm and let $(s, g(s))$ be an n-history.

- Let S^* be the set of all sequences that encode the factorization of a number.
- If $s \notin S^*$, then $T(s, g(s)) = 1$.
- If $s \in S^*$, then the prefix of s corresponds to some number m. Determine the probability, p, that G assigns to s conditional on the prefix being m.
- If $p \geq \varepsilon$, then $T(s, g(s)) = 1$; otherwise $T(s, g(s)) = 0$.

Call this the *factoring* test. It can be implemented in polynomial time since we have efficient algorithms for multiplication and primality testing (see Agrawal, Kayal, and Saxena (2004)). We can make a linear-time test by appropriately padding the sequence.

THEOREM 3: *For any $\varepsilon > 0$, for sufficiently large m, the factoring test passes the truth with probability $1 - \varepsilon$ and any forecasting algorithm that can ignorantly pass the test with probability $1 - \varepsilon$ can be used to factor m.*

PROOF: Consider any distribution of nature P and the forecaster F^P. Let p_m be the probability that P assigns a sequence whose prefix encodes m. Let q_m be the probability that a sequence encodes the factorization of m conditional on its prefix encoding m. The factoring test will fail F^P if for some m a sequence $s \in S^*$ with prefix m is realized and $q_m < \varepsilon$. The probability of this happening is

$$\sum_{m\,:\,q_m<\varepsilon} p_m q_m < \sum_m p_m \varepsilon = \varepsilon.$$

Hence, the factoring test will pass the truth with high probability.

Now, fix an m and consider the distribution that puts its full weight on the sequence that encodes the unique factorization of m. If the forecaster passes the

test for this distribution, the conditional probability that suffix of the sequence reveals the prime factors of m is at least ε. Use the forecasted probabilities of the suffix as a distribution to generate a sequence. With probability at least ε, we generate the factors of m. If we repeat the process $O(1/\varepsilon)$ times, we determine the factors with high probability. *Q.E.D.*

4.1. *Efficiently Sampleable Distributions*

Theorem 3 makes use of the fact that the test must be passed for all distributions and the distribution given in the proof is one where nature must know how to factor. We can easily modify the proof to nicer distributions, particularly polynomial-time sampleable distributions. A distribution μ is *polynomial-time sampleable* (see Ben-David, Chor, Goldreich, and Luby (1992)) if there is a polynomial-time computable function f and a polynomial q such that h maps strings of length $q(n)$ to strings of length n, and $\mu(x) = \Pr_r(h(r) = x)$, where r is chosen uniformly from strings of length $q(|x|)$.

Specifically, we sample the distribution μ by randomly choosing two large primes p and q and output the sequence corresponding to the tuple (pq, p, q). Using exactly the same test as in Theorem 3, we get the following result.

THEOREM 4: *For any $\varepsilon > 0$, for sufficiently large n, if we use a distribution that creates factors of length n, the factoring test passes the truth with probability at least $1 - \varepsilon$, but any forecast that passes the test with distribution μ described above will be able to factor numbers on average by those generated by μ projected on the first coordinate (the value pq).*

Most cryptographic work based on factoring assumes that factoring numbers even on average is very difficult. This proof can be generalized in a straightforward way to invert any cryptographic one-way function with unique inverses.

5. PSPACE HARDNESS

It is natural to suspect that the proof of Theorem 3 should generalize to all NP problems. The obstacle to such a generalization is the absence of unique witnesses. This means we cannot create a test that will pass the truth for all distributions of nature and still force the forecaster to put heavy weight on a witness. Witness reducing techniques as in Valiant and Vazirani (1986) do not appear to help.

In this section we take a different approach to create a linear-time test that can be ignorantly passed by a forecaster only if he/she can solve NP-hard and even PSPACE-hard problems by using the theory of interactive proof systems. Once again we assume deterministic forecasters, though the results extend to probabilistic forecasters as well.

In complexity theory the class PSPACE is the set of decision problems that can be solved by a deterministic or nondeterministic Turing machine using a

polynomial amount of memory and unlimited time. They contain the more well known class of NP problems. A logical characterization of PSPACE is that it is the set of problems expressible in second-order logic. A major result of complexity theory is that PSPACE can be characterized as all the languages recognizable by a particular interactive proof system. The hardest problems within the class PSPACE are called PSPACE-complete.

An interactive proof system is an abstraction that models computation as an exchange of messages between a prover (P) and a verifier (V). P would like to convince V that a given x belongs to a given set L.[7] P has unlimited computational resources, while V's computational resources are bounded. Messages are passed, in rounds, between P and V until V reaches a conclusion about the correctness of the statement $x \in L$. The parties take turns sending a message to the other party. A strategy for P (or V) is a function that specifies in each round what message is to be sent as a function of the history of messages exchanged in prior rounds. A pair of strategies, s_P for P and s_V for V, is called a protocol.

Membership in L admits an interactive proof system if there is a protocol (s_P, s_V) with the following two properties:

Completeness: If $x \in L$, and both P and V follow the protocol (s_P, s_V), then V will be convinced that $x \in L$.

Soundness: Suppose $x \notin L$. If V follows s_V but P is allowed to deviate from s_P, V can be convinced that $x \in L$ with some small probability.

Shamir (1992), building on the work of Lund, Fortnow, Karloff, and Nisan (1992), exhibited an interactive proof system for any PSPACE language L. The proof system has the following properties for an input x of length n.

- Let $\mathcal{F}$ be a field GF(q) for a prime q exponential in n.
- The protocol alternates between the prover (P) giving a polynomial of degree m over $\mathcal{F}$ and the verifier (V) choosing a random element of $\mathcal{F}$. Both m and the number of rounds r of the protocol are bounded by a polynomial in n.
- V takes all of the messages and decides whether to accept or reject with a deterministic polynomial-time algorithm.
- In each round for V there are at most m *bad* choices from the possible q elements of $\mathcal{F}$ that depend only on the previous messages. The other choices we call *good*. It can be computationally difficult to decide whether a choice is good or bad, but since $m \ll q$ the verifier will pick good choices with very high probability.
- If $x \notin L$, then for any strategy of P, V will reject if all of its choices are good.
- If $x \in L$, there is a strategy for P that will cause V to accept always. Moreover, if all of V's choices are good, the messages of P that cause V to accept are unique.

[7] More formally, that a given string x belongs to a certain language L.

Fix a PSPACE-complete language L. Consider a sequence of outcomes interpreted as the tuple $(x, \rho_1, v_1, \ldots, \rho_r, v_r)$, where the ρ_i's are P's messages and the v_i's are V's messages in an interactive proof system for showing $x \in L$. We call the tuple *accepting* if the verifier would accept if the protocol played out with these messages.

Let p_i be the probabilities that the realized forecaster predicts for each ρ_i and let b_i be the probabilities for each v_i.

Given a sequence interpreted as such a tuple and the realized forecasts, our test will declare FAIL if all of the following occur:

(i) $(x, \rho_1, v_1, \ldots, \rho_r, v_r)$ is an accepting tuple.
(ii) Each b_i is at most $\frac{1}{\sqrt{q}}$.
(iii) The product of the ρ_i's is at most $\frac{1}{n}$.

Otherwise the test declares PASS. The test runs in polynomial time and we can pad the sequence so that the test can run in linear time. Call this the PSPACE test.

THEOREM 5: *For any $\varepsilon > 0$, for sufficiently large n, the PSPACE test passes the truth on inputs of length n with probability at least $1 - \varepsilon$. A forecaster who can ignorantly pass the PSPACE test with probability at least $1 - \varepsilon$ can solve all inputs of PSPACE problems of length at least n.*

PROOF: Consider any distribution of nature P and the associated forecaster F^P. Let p_i's and b_i's be as described above for F^P. Let d_x be the probability that P puts on choosing a sequence starting with x. By the properties of the proof system, we have that the probability that the test fails is bounded by

$$\sum_x p_x(u_x + w_x),$$

where u_x is the probability that a sequence beginning with x fails the test when all the elements are good and w_x is the probability that a sequence beginning with x fails the test when some element is bad.

The probability w_x is bounded by the probability that some element is bad, which is bounded by

$$\sum_i mb_i \leq \frac{mr}{\sqrt{q}} = o(1).$$

If the v_i's are good, then there is at most one choice of the ρ_i's that leads to an accepting tuple. So

$$u_x \leq \prod_i p_i \leq \frac{1}{n} = o(1).$$

So the probability that the test fails is

$$\sum_x p_x(u_x + w_x) = o(1) \sum_x p_x = o(1).$$

For any x in L we define the following distribution. Fix a strategy for the prover in the interactive proof system for L that guarantees acceptance for all verifier's messages. For each ρ_i play that strategy; for each v_i pick an element from F uniformly. Output $(x, \rho_1, v_1, \ldots, \rho_r, v_r)$.

Suppose we have a forecaster who passes the test on this distribution with high probability. The first condition of the test is always true. Fix an i. There are q possible v_i of which the forecaster can only put weight greater than $\frac{1}{\sqrt{q}}$ on $\sqrt{q}$ of those possibilities. So with probability at least $1 - r\sqrt{q}/q = 1 - o(1/n)$, all of the b_i's will be less than $\frac{1}{\sqrt{q}}$ and the second condition of the test will be satisfied. By a similar argument, with high probability all of the v_i's are good.

Since the forecaster passes the test with high probability, then with high probability the product of the p_i's must be at least $1/n$. So if we use the forecaster as a distribution to generate the prover's responses, the verifier will accept with probability close to $1/n$. If we repeat the process $O(n)$ times at least once, the verifier will accept with high probability, whereas if x were not in L, repeating the protocol a polynomial number of times will rarely cause the verifier to accept, no matter what the prover's strategy. Thus the forecaster gives us a probabilistic algorithm for determining whether x is in L. *Q.E.D.*

6. CONCLUSION

Sandroni (2003) showed that any reasonable forecast tester can be passed by an ignorant forecaster. However his proof requires the forecaster to run in exponential time. We show this exponential blowup is necessary by exhibiting efficient forecast testers that any forecaster ignorant of nature requires to solve PSPACE-complete problems, generally believed to require exponential time.

We showed that for efficiently sampleable distributions of nature, an ignorant forecaster would still be required to factor random numbers. It remains open whether one can prove a similar result for NP-hard or PSPACE-hard problems.

In the future we would also like to consider what happens when nature is drawn from some smaller set of simpler distributions (see, for example, Al-Najjar, Sandroni, Smorodinsky, and Weinstein (2008)). Note that this may change results in two directions: the forecaster need only succeed on these distributions of nature, but also the tester need only pass the truth on that set of distributions as well.

REFERENCES

AGRAWAL, M., N. KAYAL, AND N. SAXENA (2004): "PRIMES Is in P," *Annals of Mathematics*, 160, 781–793. [100]

AL-NAJJAR, N., A. SANDRONI, R. SMORODINSKY, AND J. WEINSTEIN (2008): "Testing Theories With Learnable and Predictive Representations," Manuscript, Northwestern University. [104]

BABAI, L., L. FORTNOW, AND C. LUND (1991): "Nondeterministic Exponential Time Has Two-Prover Interactive Protocols," *Computational Complexity*, 1, 3–40.

BEN-DAVID, S., B. CHOR, O. GOLDREICH, AND M. LUBY (1992): "On the Theory of Average Case Complexity," *Journal of Computer and System Sciences*, 44, 193–219. [101]

DAWID, A. P. (1982): "The Well Calibrated Bayesian," *Journal of the American Statistical Association*, 77, 605–613. [93]

——— (1985): "The Impossibility of Inductive Inference," *Journal of the American Statistical Association*, 80, 340–341. [98]

DEKEL, E., AND Y. FEINBERG (2006): "Non-Bayesian Testing of a Stochastic Prediction," *Review of Economic Studies*, 73, 893–906. [94]

FOSTER, D. P., AND R. V. VOHRA (1993): "Asymptotic Calibration," *Biometrika*, 85, 379–390. [93]

GOOD, I. J. (1952): "Rational Decisions," *Journal of the Royal Statistical Society, Series B*, 14, 107–114. [94]

LEHRER, E. (2001): "Any Inspection Rule Is Manipulable," *Econometrica*, 69, 1333–1347. [93]

LUND, C., L. FORTNOW, H. KARLOFF, AND N. NISAN (1992): "Algebraic Methods for Interactive Proof Systems," *Journal of the Association for Computing Machinery*, 39, 859–868. [102]

OLSZEWSKI, W., AND A. SANDRONI (2009): "A Non-Manipulable Test," *Annals of Statistics* (forthcoming). [94]

——— (2008): "Manipulability of Future-Independent Tests," *Econometrica*, 76, 1437–1480. [94]

SANDRONI, A. (2003): "The Reproducible Properties of Correct Forecasts," *International Journal of Game Theory*, 32, 151–159. [93-95,104]

SANDRONI, A., R. SMORODINSKY, AND R. VOHRA (2003): "Calibration With Many Checking Rules," *Mathematics of Operations Research*, 28, 141–153. [93]

SHAMIR, A. (1992): "IP = PSPACE," *Journal of the Association for Computing Machinery*, 39, 869–877. [102]

VALIANT, L., AND V. VAZIRANI (1986): "NP Is as Easy as Detecting Unique Solutions," *Theoretical Computer Science*, 47, 85–93. [95,101]

VOVK, V., AND G. SHAFER (2005): "Good Sequential Probability Forecasting Is Always Possible," *Journal of the Royal Statistical Society, Series B*, 67, 747–763. [93]

Dept. of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, U.S.A.

and

Dept. of Managerial Economics and Decision Sciences, Kellogg Graduate School of Management, Northwestern University, Evanston, IL 60208, U.S.A.

Manuscript received May, 2007; final revision received June, 2008.