Can Theories be Tested? A Cryptographic Treatment of Forecast Testing

Kai-Min Chung*
Department of Computer Science
Cornell University
chung@cs.cornell.edu

Edward Lui
Department of Computer Science
Cornell University
luied@cs.cornell.edu

Rafael Pass†
Department of Computer Science
Cornell University
rafael@cs.cornell.edu

ABSTRACT

How do we test if a weather forecaster actually knows something about whether it will rain or not? Intuitively, a "good" forecast test should be *complete*—namely, a forecaster knowing the distribution of Nature should be able to pass the test with high probability, and *sound*—an uninformed forecaster should only be able to pass the test with small probability.

We provide a comprehensive cryptographic study of the feasibility of complete and sound forecast testing, introducing various notions of both completeness and soundness, inspired by the literature on interactive proofs. Our main technical result is an incompleteness theorem for our most basic notion of computationally sound and complete forecast testing: If Nature is implemented by a polynomial-time algorithm, then every complete polynomial-time test can be passed by a completely uninformed polynomial-time forecaster (i.e., a computationally-bounded "charlatan") with high probability. We additionally study alternative notions of soundness and completeness and present both positive and negative results for these notions.

Categories and Subject Descriptors

F.1.2 [**Theory of Computation**]: Modes of Computation—*Interactive and reactive computation*

General Terms

Theory, Economics

Keywords

Forecast testing; incompleteness; multiplicative weights

1. INTRODUCTION

Forecasting plays an important role in many parts of society and in many fields, such as economics and the sciences. One example is weather forecasting, where a weather forecaster tries to predict the weather in future days (e.g., the forecaster may predict that the next day is sunny with 80% probability and rainy with 20% probability) over a sequence of days. How can we evaluate if such a forecaster actually knows something about the weather?

More generally, forecasting is used by many firms, investors, and governments to make decisions based on the forecasts of variables such as demand, stock prices, unemployment rates, and energy consumption. Also, entire industries, such as management consulting and financial advising, are dedicated to the business of supplying forecasts. In the sciences, theories and models also supply forecasts (i.e., predictions) of the outcome of experiments and natural processes. One way of evaluating forecasts is the use of a scoring rule, which assigns a numerical score to a (probabilistic) forecast based on the forecast and the observed outcome (e.g., see [Bri50, Bic07, GR07]), and the score of a sequence of forecasts is typically the average score of the forecasts. Strictly proper scoring rules (see [Bic07, GR07]) incentivize forecasters to report forecasts equal to their personal beliefs (instead of gaming the system). Another way of evaluating a sequence of forecasts is seeing how well "calibrated" the forecasts are; roughly speaking, the forecasts are well-calibrated if when we consider the periods where the forecasts predict an outcome o with probability close to p, we have that the empirical frequency of outcome o during these periods is close to p (e.g., see [Daw82]). For example, if we consider the days where a weather forecaster predicts something close to "30% rainy" and compute the proportion of these days that were actually rainy, then the proportion should be close to 30%.

For concreteness, let us consider weather forecasting specifically, and suppose Nature generates weather sequences according to some distribution. Can we use scoring rules, calibration, or some other method to determine whether a forecaster actually knows (or even just approximately knows)

*Chung is supported in part by a Simons Foundation postdoctoral fellowship.

†Pass is supported in part by an Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, NSF Award CCF-1214844, NSF Award CNS-1217821, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2-0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US government.

ITCS'13, January 9–12, 2013, Berkeley, California, USA.

Nature's distribution for generating weather? More specifically, can one design a *forecast test* that passes forecasters that have accurate knowledge about Nature's distribution, but fails ignorant forecasters that have no knowledge about Nature's distribution? Such a test is an algorithm that takes a sequence of n forecasts and n outcomes generated by a forecaster and Nature (respectively) over n days, and either passes or fails the forecasting; the forecasts made by the forecaster each day are distributions over outcomes (e.g., 80% sunny and 20% rainy).

A good test should at least be passable (with high probability) by an informed forecaster that knows Nature's distribution and forecasts the correct conditional probabilities each day (otherwise, the test is not very useful, since it does not distinguish good forecasters from bad/ignorant forecasters). We call such a property *completeness*.

Definition 1 (Completeness of a test (Informal)). *A test is said to be* complete *if an informed forecaster that knows Nature's distribution can pass the test with high probability.*

A good test should at the very least also fail (with decent probability) every ignorant forecaster (i.e., a "charlatan") that has no knowledge about Nature's distribution. We call such a property *soundness*.

Definition 2 (Soundness of a test (Informal)). *A test is said to be* sound *if for every forecaster P that has no (a priori) knowledge of Nature's distribution, P fails the test with decent probability.*

One of the central contributions of this paper is to provide formal definitions of completeness and soundness for forecast testing (inspired by the literature on interactive proofs initiated by Goldwasser, Micali and Rackoff [GMR89] and Babai and Moran [BM88]) and to study the feasibility of such tests. But before turning to our own treatment, let us briefly review the intriguing literature on this topic.

In [San03], Sandroni showed a surprising incompleteness theorem: Any test that is complete (in a particular well-defined way) cannot even satisfy very weak notions of soundness. That is, for any test that passes an informed forecaster that forecasts the correct conditional probabilities each day, there exists an *ignorant* cheating forecaster that passes the test with the same probability as the informed forecaster (who *knows* Nature's distribution). (See [DF06, OS09, OS06, FV98, Leh01, VS03] for related work.) Let us briefly review the proof of this elegant result (which follows a proof technique by Hart). Sandroni's proof uses the minimax theorem and roughly works as follows. Consider a two-player zero-sum game between a forecaster and Nature. The forecaster chooses a forecasting scheme, while Nature chooses a weather generating distribution. The forecaster's payoff is the probability that the test passes the chosen forecasting scheme. Since the test is complete, if the forecaster knows Nature's distribution, then the forecaster can choose a forecasting scheme that has high payoff. Thus, for every mixed strategy of Nature, there exists a forecaster strategy that yields high payoff. By the minimax theorem, there exists a mixed forecaster strategy that yields high payoff no matter what strategy Nature uses; that is, the forecaster passes the test with high probability no matter what Nature's distribution is. Thus, a completely ignorant forecaster (i.e., a charlatan) can pass the test with high probability.

Forecast testing with computational restrictions. Even though Sandroni's proof ([San03]) exhibits an ignorant cheating forecaster that can pass the test, it is not clear whether the cheating forecaster can be found efficiently. Since the game is exponentially large in the sense that both the forecaster and Nature have (at least) exponentially many pure strategies, there is little hope in finding the cheating forecaster efficiently. This leads to the question of whether computational restrictions on the forecaster can help us design a test that is both complete and sound for reasonable notions of completeness and soundness.

Some initial positive results along these lines appear in the work of Fortnow and Vohra [FV09]: it shows the existence of a particular (polynomial-time computable) test and a particular polynomial-time algorithm for Nature such that a forecaster that forecasts the correct conditional probabilities will pass the test, but an uninformed *efficient* (i.e., polynomial-time) forecaster can only pass the test with negligible probability. This result thus suggests that at the very least, Sandroni's universal cheating strategy does not hold for *every* Nature distribution, as long as we restrict to computationally bounded forecasters. The idea is as follows: the algorithm for Nature generates large random primes p and q and outputs one bit of (pq, p, q) per day. Assuming that one cannot factor the product of two large random primes efficiently, it follows that an *efficient* forecaster cannot make decent forecasts and thus cannot pass the test with decent probability.

1.1 Our Treatment of Forecast Testing

The result of Fortnow and Vohra gives some hope that meaningful forecast testing may actually be possible. However, observe that an *efficient* forecaster that *knows* the algorithm for Nature cannot even pass the test of Fortnow and Vohra. This is because their algorithm for Nature keeps *hidden state* in the sense that after generating p and q, the algorithm remembers p and q for use in future days. A forecaster that knows the algorithm for Nature would still not know the hidden state (p, q), so the forecaster would be unable to efficiently make good forecasts. Thus, the test of Fortnow and Vohra does not satisfy completeness with respect to informed *efficient* forecasters. If we aim to provide a computational treatment of forecast testing, it is imperative that the test satisfies a notion of completeness also with respect to *efficient* forecasters (knowing Nature); towards this, we thus consider Natures with *publicly observable state*.

Natures with publicly observable state. In general, if Nature's algorithm keeps hidden state, then even if a forecaster knows Nature's algorithm, it might be computationally intractable for the forecaster to learn Nature's distribution for the next day. Thus, to ensure that an informed forecaster who knows Nature's algorithm can efficiently pass the test, we consider algorithms for Nature as efficient Turing machines that keep no hidden state, and on input a history of outcomes, outputs a distribution over outcomes for the next day. Note that in this model, an informed forecaster can simply run Nature's algorithm to learn the next day's distribution. (One can also consider a slightly more complicated model where we allow Nature to also output state information that is publicly observable and of polynomial length; our results still hold in this model, but for simplicity, we will mainly work with the simpler model.)

Our computational incompleteness theorem. One of our main results is to restore the strong impossibility result of Sandroni ([San03]) in the computationally bounded setting; that is, we show the following:

THEOREM 1 (INFORMALLY STATED). *Any efficient test that is complete cannot even satisfy very weak notions of soundness with respect to efficient forecasters. That is, for any efficient test that is complete, and for any polynomial bound $t(\cdot)$, there exists a universal uniform* efficient *cheating forecaster that passes the test with high probability for every uniform Nature algorithm that runs in time $t(\cdot)$.*

Recall that Sandroni ([San03]) used the minimax theorem to show the existence of a (possibly inefficient) universal cheating forecaster in the corresponding zero-sum game. A natural approach to showing the above theorem is to show that such a universal cheating forecaster is actually efficient and that we can find it efficiently. Note that when Nature algorithms are polynomial-time Turing machines, by a standard enumeration argument, it suffices to consider polynomially many such machines as pure strategies of Nature in the game.[1]

Thus, it seems that the game is now of polynomial size, and one can find a universal mixed strategy for the forecaster efficiently using linear programming. However, Nature still has exponentially many mixed strategies, each of which may have a different "good response", resulting in super-polynomially many strategies for the forecaster. Thus, the linear programming approach may yield a complex mixed strategy over super-polynomially many pure strategies, and so the universal cheating forecaster may not be efficient.

Another possible strategy is for the forecaster to run an experts algorithm that roughly guarantees that the forecaster's payoff is almost as good as the payoff of the best forecaster. However, these algorithms and results consider a scenario where the forecaster receives some payoff on each day, and it is only guaranteed that the forecaster's payoff on average (over the days) is almost as good as that of the best forecaster. This does not fit our scenario where we consider *general* tests that may not necessarily assign some payoff/score to the forecaster on each day; in particular, the test may fail the forecaster if the forecaster does not do well near the beginning. One can also consider forecasters that try to learn Nature's algorithm over time; however, it may take many days to learn Nature's algorithm, and by then, the forecaster may have already failed the test.

Our approach. In general, our result does not follow from a straight-forward application of known results on experts/learning algorithms. However, we do use a multiplicative weights algorithm, which can be used to approximately find the minimax strategies of two-player zero-sum games (e.g., see [FS99]). Unlike the linear programming approach, a multiplicative weights algorithm can efficiently yield an approximate minimax strategy for the forecaster that only has polynomial support. The multiplicative weights algorithm roughly works as follows. A zero-sum game between a row player and a column player is played for a certain number of rounds. In the first round, the column player chooses the uniform mixed strategy, and then the row player chooses a "good response" (a pure strategy) that yields high payoff. In the next round, the column player updates it's mixed strategy using a multiplicative update rule that depends on the row player's strategy in the previous round. The row player again chooses a good response that yields high payoff. By repeating this procedure for polynomially many rounds, the row player obtains a sequence of good responses such that the uniform distribution over the multiset of good responses yields high payoff no matter what strategy the column player chooses. In our setting, the row player is the forecaster and the column player is Nature.

There are some issues that need to be resolved. Firstly, finding a good response for the forecaster involves running Nature's mixed strategy, but Nature's mixed strategy depends on the forecaster's strategy in the previous round. This recursion may potentially lead to a blow up in the algorithm's running time. However, using the fact that Nature only has polynomially many strategies, we can approximate Nature's current mixed strategy by just keeping track of the weights for each of the possible Turing machines. This guarantees that the time complexity needed to implement any mixture over Nature strategies is always a priori bounded, which prevents a blow-up in running time. Another issue is that the forecaster needs to efficiently find a good response to Nature's mixed strategy. This is done by using the informed forecaster P from the completeness property of the test; however, P only needs to work for Nature algorithms that have no hidden state, but a mixture over Nature strategies is normally implemented as a Nature algorithm that keeps hidden state. We overcome this problem by considering a non-uniform stateless Nature algorithm that uses Bayesian updating to essentially simulate Nature's mixed strategy, and we then use the informed forecaster P for this stateless Nature algorithm instead. After resolving these issues, we get a universal uniform efficient cheating forecaster.

Let us also point out that it is crucial for our proof that Nature is modeled as a uniform polynomial-time algorithm, as opposed to a non-uniform algorithm. Our technique easily extends to the case where Nature is a polynomial-time machine with $\log n$ bits of advice, but as we show in our full paper, the ideas of Fortnow and Vohra [FV09] can be used to show the existence of a complete test for polynomial-time Natures with polylog n bits of advice that no universal efficient cheater can pass (except with negligible probability).

1.2 Restricted Natures

In light of our computational incompleteness theorem for forecast testing, we investigate other alternative approaches (other than computational restrictions) to get complete and sound forecast testing. A crucial component of both Sandroni's and our impossibility result is that the strategy set of Nature is convex. Thus, a natural approach to circumvent these results would be to only require completeness with respect to a non-convex subclass of Nature distributions. Towards this, we consider Nature distributions that are slightly "biased", in the sense that for at least e.g., 10% of days, the weather is $> 60\%$ likely to be rainy or $> 60\%$ likely to be sunny, and completeness only needs to hold with respect to biased Nature distributions. In this setting, we are able to design a simple complete test that satisfies the

[1]Recall that a polynomial-time Turing machine M is a single (fixed-sized, independent of the input length) Turing machine that, on inputs of length n, takes $poly(n)$ computational steps. Thus, if we enumerate the first n Turing machines in a list, M will always be in the list as long as n is sufficiently large.

following notion of soundness[2] (inspired by the classic notion of soundness of an interactive proof [GMR89, BM88]):

DEFINITION 3 (SOUNDNESS OF A TEST (INFORMAL)). *A test is said to be* sound (resp., computationally sound) with respect to a class $\mathcal{N}$ of Nature algorithms *if every uninformed forecaster (resp., uninformed efficient forecaster) will fail the test with high probability over a uniformly random Nature algorithm in $\mathcal{N}$.*

We show the following:

THEOREM 2 (INFORMALLY STATED). *There exists a linear-time computable test that is complete with respect to biased Nature algorithms, and is sound with respect to a class of biased polynomial-size Nature algorithms.*

Our notion of a "biased" Nature, as well as the above theorem, are closely related to a recent result by Olszewski and Sandroni [OS09].[3] The main differences are that our test is simpler and more efficient, our notion of soundness is stronger, and our analysis is significantly simpler and can provide a concrete bound on the soundness error as a function of the number of days. Our test T computes a score s_i for each day i with forecast p_i as follows: if the outcome is 1, $s_i = p_i - 1/2$; otherwise, $s_i = 1/2 - p_i$. The test T accepts if and only if the total score is greater than a certain threshold. We show that T satisfies the restricted completeness and soundness by a martingale analysis using Azuma's inequality.

1.3 Towards Stronger Notions of Soundness

In our opinion, the above notion of soundness is still relatively weak. Ideally, we would want a notion of soundness similar to the cryptographic notion of a proof of knowledge [FS89, BG92, MP07]: roughly speaking, we only want a forecaster to be able to pass the test if the forecaster's predictions are close to the actual distribution of Nature, on average over the days. We refer to this notion of soundness as *strong soundness*.

DEFINITION 4 (STRONG SOUNDNESS (INFORMAL)). *A test is said to be* strongly sound (resp., computationally strongly sound) with respect to a class $\mathcal{N}$ of Nature algorithms *if for every Nature $N \in \mathcal{N}$, with high probability, if a forecaster (resp., efficient forecaster) manages to pass the test, then the forecasts made by the forecaster are close to the Nature N's actual distribution, on average over the days.*

Unfortunately, we demonstrate that achieving this strong notion of soundness cannot be done, even in the setting of computationally-bounded forecasters, and even when only requiring completeness to hold for biased computationally-efficient Natures.

THEOREM 3 (INFORMALLY STATED). *Any test that is complete with respect to biased Nature algorithms is not computationally strongly sound with respect to some class of biased polynomial-time Nature algorithms with n bits of non-uniform advice. If we additionally assume the existence of a problem in the complexity class* E *with exponential circuit complexity[4], then the same holds even if Nature algorithms are uniform and efficient.*

The idea behind Theorem 3 is quite straight-forward. Consider a Nature that outputs rain with probability 0.7 on each day independently. By completeness, a forecaster outputting 0.7 on each day will pass the test with high probability. Now, if we select a "deterministic" Nature (that on each day either outputs rain with probability 1, or sun with probability 1) at random such that the probability of rain on each day is 0.7 independently, then the same forecaster will still pass the test with high probability, although it outputs an incorrect prediction on every day (which contradicts strong soundness). Such a "deterministic" Nature requires n bits of non-uniform advice (namely, the rain/sun sequence); nevertheless, by relying on derandomization techniques, these n bits can be generated using an appropriate Nisan-Wigderson type pseudorandom generator ([NW94]) using only $O(\log n)$ bits of random seed. By enumerating over all polynomially many possible seeds, we can make Nature uniform and efficient.

The above argument has other implications. Consider the following commonly used method for checking the accuracy of a theory of some physical phenomenon. The theory outputs a single prediction and next this prediction is evaluated on multiple supposedly independent samples (using e.g., some statistical test such as a t-test or a χ^2-test). For example, a theory may predict the movement of a particle, and we can evaluate the prediction on multiple supposedly independent samples. When the samples are indeed independent, the soundness of the evaluation/test follows from elementary statistical theory. However, when independence cannot be safely assumed, in order for this method to be actually sound, we need to check that the samples are actually independent[5]. It is not hard to see that the above argument can be used to show that checking independence is impossible.

The above discussion motivates our last relaxation of the problem of forecast testing. We further restrict both the completeness and the soundness conditions to only hold with respect to Natures that already satisfy some notion of independence over each consecutive (non-overlapping) segment of k days. More specifically, for each segment i, conditioned on the outcomes before segment i, the outcomes in segment i are pairwise independent and have the same marginal distribution. We refer to such Natures as *k-segment Natures*. These Natures can capture the testing of scientific theories, where predictions are evaluated on multiple independent or pairwise independent trials/samples. In this setting, we present a simple test that achieves both completeness and strong soundness.

[2] As far as we can tell, this notion of soundness is significantly stronger than previous notions of soundness considered in the literature on forecast testing; there, the typical notion of soundness only requires that for each uninformed forecaster, there exists some Nature for which the test will fail the forecaster with high probability.

[3] At the time of writing the first version of this paper, we were unaware of this result. We thank Lance Fortnow for pointing us to it.

[4] This is a standard complexity-theoretic assumption from the literature on derandomization (e.g., see [IW97]).

[5] There are various method used in practice, such as permuting the data, using sliding windows, etc.

THEOREM 4 (INFORMALLY STATED). *There exists a linear-time computable test that is both complete and strongly sound with respect to k-segment Natures.*

The test is straight-forward: On each segment of k days, compute the distance between the empirical probability of outcome 1 and the forecaster's prediction on the first day of the segment. If the average distance (over the segments) is sufficiently small, the test accepts; otherwise, the test rejects. To analyze the test, we use Chebyshev's inequality to analyze the distance in each segment, and then we use Azuma's inequality to analyze the average distance over the segments.

1.4 Outline of Our Results

Our model for forecast testing can be found in Section 2, and our definitions of soundness can be found in Section 2.1. Our computational incompleteness theorem (Theorem 1) can be found in Section 3. Our feasibility result for biased Natures (Theorem 2) can be found in Section 4. Our computational strong soundness impossibility result for biased Natures (Theorem 3) can be found in Section 5. Lastly, our strong soundness feasibility result for k-segment Natures (Theorem 4) can be found in Section 6. All missing proofs, as well as our other results, can be found in the full version of this paper.

2. PRELIMINARIES AND DEFINITIONS

Let $\mathcal{O}$ be any finite set representing possible *outcomes*, and let $\Delta(\mathcal{O})$ be the set of all probability distributions over $\mathcal{O}$. A *forecast* is a distribution over $\mathcal{O}$, i.e., an element of $\Delta(\mathcal{O})$. Given a vector $\vec{v} = (v_1, \ldots, v_n)$ and a positive integer $i \leq n+1$, let $\vec{v}_{<i}$ be the vector $(v_1, \ldots, v_{i-1})$. Given a positive integer $k \in \mathbb{N}$, let $[k] = \{1, \ldots, k\}$.

A *nature* is a deterministic Turing machine N such that on input 1^n and a history $\vec{o}_{<i}$ of outcomes for days 1 through $i-1$, N outputs a distribution over $\mathcal{O}$ representing the likelihood of each outcome for day i. For generality, we allow natures to be non-uniform and take an advice string as auxiliary input. A class of natures $\mathcal{N} = \{\mathcal{N}_n\}_{n\in\mathbb{N}}$ is a sequence of sets of natures $\mathcal{N}_n$; we often omit the index n when the context is clear. A *forecaster* is a probabilistic interactive Turing machine P such that on input 1^n and a history $\vec{o}_{<i}$ of outcomes for days 1 through $i-1$, P outputs a forecast for day i.

Given a forecaster P, a nature N, and a positive integer n, let $(P, N)(1^n)$ be the output of the following experiment: For $i = 1, \ldots n$ do: $p_i \leftarrow P(1^n, \vec{o}_{<i})$; $O_i \leftarrow N(1^n, \vec{o}_{<i})$; $o_i \leftarrow O_i$; End for; Output $((p_1, \ldots, p_n), (o_1, \ldots, o_n))$. The forecaster P can be *stateful* throughout the experiment; in particular, P can keep track of the forecasts that it has made so far. Nature is always given the history of outcomes as well as any (non-uniform) advice it had at the beginning of the experiment, but nature is not allowed to keep any hidden (unobservable) state for use in future days. $(P, N)(1^n)$ represents an n-day experiment where the forecaster P tries to make good forecasts of the outcomes generated by the nature N.

We can also consider an alternative model where nature can keep track of state, but the state is publicly observable (since otherwise even a good forecaster that knows the algorithm of nature would not be able to make good forecasts). In our full paper, we describe such a model where nature also outputs state information on each day, and a history now includes the outcome as well as the state for each day in the past. We refer to this alternative model as "the model of Nature with publicly observable state", and we refer to the earlier simpler model as "the simple Nature model". In this paper, we will mainly work with the simple Nature model for simplicity. However, all of our technical results still hold in the other model.

A (forecast) *test* is a (possibly probabilistic) Turing machine T such that on input $(\vec{p}, \vec{o}) \in (\Delta(\mathcal{O}))^n \times \mathcal{O}^n$ representing a sequence of forecasts and outcomes, T outputs a bit in $\{0, 1\}$ to indicate whether the test accepts or rejects, with 1 being accept. A test is only meaningful if it passes (with high probability) a forecaster that knows the distribution of nature (otherwise, the test is not very useful, since it does not distinguish good forecasters from bad/ignorant forecasters). Borrowing terminology from interactive proofs [GMR89, BM88], we call this property *completeness*.

Our definition of completeness considers a "canonical forecaster" that knows the nature algorithm. Given a nature N, the *canonical forecaster for* N is the forecaster P_N that, on input $(1^n, \vec{o}_{<i})$, runs N on input $(1^n, \vec{o}_{<i})$ to get a distribution O_i, and then outputs O_i as its forecast. Given a nature N, we will use P_N to denote the canonical forecaster for N. We now formally define what it means for a test T to be *complete*.

DEFINITION 5 (COMPLETENESS OF A TEST). *A test T is said to be* complete with completeness error $\epsilon(\cdot)$ *if for every nature N and every $n \in \mathbb{N}$, we have*

$$\Pr[T((P_N, N)(1^n)) = 1] \geq 1 - \epsilon(n),$$

where P_N is the canonical forecaster for N, and the probability is over the random coins of T and the experiment $(P_N, N)(1^n)$.

2.1 Definitions of Soundness

In this section, we present several definitions of soundness with varying levels of strength. Let T be any test. We would like the test T to satisfy some notion of *soundness*, i.e., a forecaster without knowledge of nature should not be able to pass the test with decent probability. A very weak notion of soundness is that there is no universal forecaster that passes the test with decent probability no matter what nature is.

DEFINITION 6 (WEAK SOUNDNESS OF A TEST). *Let $\mathcal{N}$ be any set of natures. A test T is said to be* weakly sound (resp., computationally weakly sound) with soundness error $\epsilon(\cdot)$ with respect to $\mathcal{N}$ *if for every forecaster (resp., efficient forecaster) P, there exists an $N \in \mathcal{N}$ such that for sufficiently large $n \in \mathbb{N}$, we have*

$$\Pr[T((P, N)(1^n)) = 1] \leq \epsilon(n),$$

where the probability is over the random coins of T and the experiment $(P, N)(1^n)$.

Weak soundness has been the typical notion of soundness used in the literature. However, this notion of soundness is very weak, and it does not even prevent the existence of a cheating forecaster that actually passes the test for most natures in $\mathcal{N}$. Ideally, if a forecaster does not know any information about Nature beyond being in a certain class $\mathcal{N}$, then the test should reject the forecaster with high probability for most natures in $\mathcal{N}$. We formalize this stronger soundness property in the following definition.

DEFINITION 7 (SOUNDNESS OF A TEST). *Let $\mathcal{N}$ be any class of natures. A test T is said to be* sound (resp., computationally sound) with respect to $\mathcal{N}$ *if for every forecaster (resp., efficient forecaster) P, there exists a negligible function*[6] *$negl(\cdot)$ such that for every $n \in \mathbb{N}$, we have that at least a $(1 - negl(n))$-fraction of the natures $N \in \mathcal{N}$ satisfy*

$$\Pr[T((P, N)(1^n)) = 1] \leq negl(n),$$

where the probability is over the random coins of T and the experiment $(P, N)(1^n)$.

We note that the above soundness definition is still weak in the following sense. It is possible that a sound test T passes some $(P, N)(1^n)$ with high probability, but the forecast $\vec{p}$ produced by P may be still poor in the sense that for most of the days $i \in [n]$, p_i is actually statistically far from the true distribution $N(1^n, \vec{o}_{<i})$ of the nature N. Ideally, we would like a good test T to prevent such possibilities, which is formalized by the following definition of strong soundness similar to the cryptographic notion of a proof of knowledge [FS89, BG92, MP07].

DEFINITION 8 (STRONG SOUNDNESS OF A TEST). *Let $\mathcal{N}$ be any class of natures. A test T is said to be* (ϵ, δ)-strongly sound (resp., computationally (ϵ, δ)-strongly sound) with respect to $\mathcal{N}$ *if for every forecaster (resp., efficient forecaster) P, every $n \in \mathbb{N}$, and every nature $N \in \mathcal{N}$, the following event B occurs with probability at most $\epsilon(n)$ over the randomness of the experiment $(P, N)(1^n)$ and the test T: Let $(\vec{p}, \vec{o})$ be the output of $(P, N)(1^n)$; then, the event B is defined to be the event where both of the following conditions hold:*

- $T(\vec{p}, \vec{o}) = 1$
- $\frac{1}{n} \sum_{i=1}^{n} ||p_i - N(1^n, \vec{o}_{<i})||_1 > \delta(n)$.

In other words, the event B says that the test T accepts but the average statistical distance between the forecasts and the true distributions is greater than $\delta(n)$.

3. A UNIVERSAL EFFICIENT CHEATING FORECASTER

In this section, we formalize and prove our computational incompleteness theorem (Theorem 1) for forecast testing: For any efficient complete test, there exists a universal uniform efficient cheating forecaster that passes the test without any knowledge of nature's distribution.

THEOREM 5. *Let $t(\cdot)$ be any polynomial, and let $\mathcal{N}$ be the collection of all uniform natures whose running time is bounded by $t(\cdot)$. Let $p(\cdot)$ be any positive polynomial.*

Let T be any efficient (PPT) test that is complete with completeness error $\epsilon(\cdot)$. Then, there exists a universal uniform efficient cheating forecaster C for T that runs in time $poly(n, t(n), p(n))$ such that for every nature $N \in \mathcal{N}$ and every sufficiently large $n \in \mathbb{N}$, we have

$$\Pr[T((C, N)(1^n)) = 1] \geq 1 - \epsilon(n) - \frac{1}{p(n)},$$

where the probability is over the random coins of T and the experiment $(C, N)(1^n)$.

[6] A function $f : \mathbb{N} \to \mathbb{R}^+$ is *negligible* if it decays faster than any inverse polynomial function, i.e., for every $c \in \mathbb{N}$, we have $f(n) \leq \frac{1}{n^c}$ for sufficiently large $n \in \mathbb{N}$.

We now prove Theorem 5. We will construct a uniform forecaster C running in time $poly(n, t(n), p(n))$ such that for every uniform nature N running in time $t(n)$, we have that for sufficiently large $n \in \mathbb{N}$,

$$\Pr[T((C, N)(1^n)) = 1] \geq 1 - \epsilon(n) - O\left(\frac{1}{p(n)}\right).$$

Since $p(\cdot)$ is an arbitrary positive polynomial, this is sufficient for proving Theorem 5.

High-level description of the universal cheating forecaster C. Let $M_1, M_2, M_3, \ldots$ be any enumeration of the set of all uniform natures, and let M'_1, M'_2, $M'_3, \ldots$ be the corresponding sequence where M'_i is the same as M_i except that after $t(n)$ steps, M'_i stops and deterministically outputs any fixed distribution over $\mathcal{O}$.

At a high level, the forecaster C finds a "good set" of forecasting machines $\{P_1, \ldots, P_L\}$ using a multiplicative weights algorithm ([FS99]), and then chooses one of them uniformly at random and uses the chosen machine to make forecasts in the interactive game *for all n days*.

In the multiplicative weights algorithm of the forecaster C, C simulates L rounds (repetitions) of a zero-sum game between a forecaster F and "Nature", where the payoff for the forecaster is the probability that the forecaster passes the test T. In each round i, Nature chooses a mixed strategy $N^{(i)}$ over its pure strategies (nature machines) $\{M'_1, \ldots, M'_n\}$, and then F chooses a forecasting machine $P_i := P_i(N^{(i)})$ that hopefully "does well" against Nature's mixed strategy $N^{(i)}$, i.e., F's expected payoff $\mathbb{E}[\Pr[T((P_i, N^{(i)})(1^n)) = 1]]$ is high. In the first round, Nature chooses the uniform distribution $N^{(1)}$ over $\{M'_1, \ldots, M'_n\}$, and after each round i, Nature updates its mixed strategy to get $N^{(i+1)}$ in a manner similar to the multiplicative weights algorithm described in [FS99].

We later show that by using an analysis similar to that in [FS99], we can show that the forecaster C does well against all nature machines in $\{M'_1, \ldots, M'_n\}$ if in every round i, the chosen forecasting machine P_i does well against Nature's mixed strategy $N^{(i)}$. To achieve the latter condition, the forecaster C makes use of the completeness of the test T, which guarantees that the canonical forecaster P_N for a nature N does well against N. However, the forecaster C cannot simply choose P_i to be the canonical forecaster $P_{N^{(i)}}$ for $N^{(i)}$, since $N^{(i)}$ is a *mixture* of nature machines, not a single nature machine N. (A mixture of nature machines may be captured by a single nature machine that keeps hidden state, but nature machines are stateless by definition and the completeness of T only holds for stateless nature machines.)

The forecaster C overcomes this problem by letting P_i be the canonical forecaster $P_{B(N^{(i)})}$ for $B(N^{(i)})$, where $B(N^{(i)})$ is essentially a "stateless version" of $N^{(i)}$ that uses Bayesian updating. Using the fact that completeness holds for all natures[7], we have that $P_i = P_{B(N^{(i)})}$ does well against the nature $B(N^{(i)})$. We show that the output distribution of the nature $B(N^{(i)})$ and the mixed-strategy nature $N^{(i)}$ are essentially the same in the (single-round) forecasting experiment, so P_i also does well against the mixed-strategy nature $N^{(i)}$, as required. (Due to the fact that $B(N^{(i)})$ uses

[7] In fact, it suffices for completeness to hold for non-uniform polynomial-time natures.

Bayesian updating, $B(N^{(i)})$ might actually only be an approximation of the mixed-strategy nature $N^{(i)}$; however, by making the error polynomially small, the result still holds.)

The universal cheating forecaster C. Let $M_1, M_2, M_3, \ldots$ be any enumeration of the set of all uniform natures, and let $M'_1, M'_2, M'_3, \ldots$ be the corresponding sequence where M'_i is the same as M_i except that after $t(n)$ steps, M'_i stops and deterministically outputs any fixed distribution over $\mathcal{O}$.

On input 1^n, C proceeds as follows:

1. Let $L = \Theta(p(n)^2 \ln n)$ and $\beta = \frac{1}{1+\sqrt{(2\ln n)/L}}$.

2. **Multiplicative weights algorithm:**

   Let $N^{(1)}$ be the uniform distribution over $\{M'_1, \ldots, M'_n\}$.

   For $i = 1, \ldots, L$ **do:**

   (a) **Choosing a forecaster P_i that does well against $N^{(i)}$:**

   Let $P_i = P_{B(N^{(i)})}$ be the canonical forecaster for the nature $B(N^{(i)})$, where B is the (non-uniform) nature that takes the weights of a distribution $N^{(i)}$ over $\{M'_1, \ldots, M'_n\}$ as advice and, on input a history $\vec{o}_{<k}$, updates the distribution $N^{(i)}$ using Bayesian updating (by conditioning on the history $\vec{o}_{<k}$), samples a nature M'_j from the updated distribution, and then outputs $M'_j(1^n, \vec{o}_{<k})$.

   (b) **Weight update:**

   Let $N^{(i+1)}(M'_j) \sim \beta^{\widehat{\mu}(P_i, M'_j)} \cdot N^{(i)}(M'_j)$, where $\widehat{\mu}(P_i, M'_j)$ is an approximation of

$$\begin{aligned}\mu(P_i, M'_j) :=\ & \Pr[T((P_i, M'_j)(1^n)) = 1] \\ =\ & \mathbb{E}[T((P_i, M'_j)(1^n))]\end{aligned}$$

   by taking the average of $\Theta(p(n)^2 \ln(Lnp(n)))$ samples of $T((P_i, M'_j)(1^n))$.

   End for

3. Choose $i \leftarrow \{1, \ldots, L\}$ uniformly at random.

4. Use the machine P_i to make forecasts in the interactive game for all n days.

We now continue with the formal proof of Theorem 5. It can be easily verified that C runs in time $poly(n, t(n), p(n)) = poly(n, t(n))$. Let N be any uniform nature with running time bounded by $t(n)$.

Claim 1. *For every sufficiently large $n \in \mathbb{N}$, we have*

$$\Pr[T((C, N)(1^n)) = 1] \geq 1 - \epsilon(n) - O\left(\frac{1}{p(n)}\right).$$

To prove the theorem, it suffices to prove the above claim. Fix a sufficiently large integer n such that the nature N appears in $\{M_1, \ldots, M_n\}$. We note that the nature N also appears in $\{M'_1, \ldots, M'_n\}$, since the running time of N is bounded by $t(n)$. Given a forecaster F and a nature M, let $\mu(F, M)$ be defined by

$$\mu(F, M) = \Pr[T((F, M)(1^n)) = 1].$$

Given a forecaster F and a distribution $N^{(i)}$ over natures, let

$$\begin{aligned}\mu(F, N^{(i)}) &:= \mathbb{E}_{M \sim N^{(i)}}[\mu(F, M)] \\ &= \sum_{M \in Supp(N^{(i)})} N^{(i)}(M) \cdot \mu(F, M).\end{aligned}$$

We note that $\widehat{\mu}(P_i, M'_j)$ in the forecasting algorithm C above is an approximation of $\mu(P_i, M'_j)$. Let

$$\widehat{\mu}(P_i, N^{(i)}) = \sum_{k=1}^{n} N^{(i)}(M'_k) \cdot \widehat{\mu}(P_i, M'_k).$$

Then $\widehat{\mu}(P_i, N^{(i)})$ is an approximation of $\mu(P_i, N^{(i)})$. We first prove a lemma that gives a lower bound on $\frac{1}{L}\sum_{i=1}^{L} \widehat{\mu}(P_i, M'_j)$ for a fixed nature M'_j in $\{M'_1, \ldots, M'_n\}$.

Lemma 1. *For every nature $M'_j \in \{M'_1, \ldots, M'_n\}$, if we run the forecaster $C(1^n)$, then (with probability 1) we have*

$$\frac{1}{L}\sum_{i=1}^{L} \widehat{\mu}(P_i, M'_j) \geq \frac{1}{L}\sum_{i=1}^{L} \widehat{\mu}(P_i, N^{(i)}) - O\left(\frac{1}{p(n)}\right)$$

The proof of Lemma 1 is very similar to the analysis of the multiplicative weights algorithm found in [FS99]. In [FS99], the multiplicative weights algorithm updates the weights in $N^{(i)}$ using the exact value of $\mu(P_i, M'_j)$; here, we only have an approximation $\widehat{\mu}(P_i, M'_j)$ of $\mu(P_i, M'_j)$, but with minor changes, the analysis in [FS99] can still be used to show Lemma 1. We now prove a lemma that gives a lower bound on $\frac{1}{L}\sum_{i=1}^{L} \mu(P_i, M'_j)$ for every nature M'_j in $\{M'_1, \ldots, M'_n\}$.

Lemma 2. *For every nature $M'_j \in \{M'_1, \ldots, M'_n\}$, if we run the forecaster $C(1^n)$, then with probability $1 - O(\frac{1}{p(n)})$ over the random coins of C, $C(1^n)$ internally generates $N^{(1)}$, ..., $N^{(L)}$ and $P_1, \ldots, P_L$ such that*

$$\frac{1}{L}\sum_{i=1}^{L} \mu(P_i, M'_j) \geq \frac{1}{L}\sum_{i=1}^{L} \mu(P_i, N^{(i)}) - O\left(\frac{1}{p(n)}\right).$$

The proof of Lemma 2 follows from Lemma 1, the Chernoff bound, and a union bound. We will now show that $\mu(C, N) = \Pr[T((C, N)(1^n)) = 1] \geq 1 - \epsilon(n) - O(\frac{1}{p(n)})$. Since $N \in \{M'_1, \ldots, M'_n\}$, we have by Lemma 2 that with probability $1 - O(\frac{1}{p(n)})$ over the random coins of C, $C(1^n)$ internally generates $N^{(1)}, \ldots, N^{(L)}$ and $P_1, \ldots, P_L$ such that $\frac{1}{L}\sum_{i=1}^{L} \mu(P_i, N) \geq \frac{1}{L}\sum_{i=1}^{L} \mu(P_i, N^{(i)}) - O(\frac{1}{p(n)})$. Since C chooses a machine in $\{P_1, \ldots, P_L\}$ uniformly at random to use in the n-day experiment, to show that $\Pr[T((C, N)(1^n)) = 1] \geq 1 - \epsilon(n) - O(\frac{1}{p(n)})$, it suffices to show that we always have $\mu(P_i, N^{(i)}) \geq 1 - \epsilon(n) - O(\frac{1}{p(n)})$ for every $i \in [L]$.

Fix $i \in [L]$. Let B be the (non-uniform) nature described in Step 2a of the description of C: $B(N^{(i)})$ takes the weights of the distribution $N^{(i)}$ over $\{M'_1, \ldots, M'_n\}$ as advice and, on input a history $\vec{o}_{<k}$, updates the distribution $N^{(i)}$ using Bayesian updating (by conditioning on the history $\vec{o}_{<k}$), samples a nature M'_j from the updated distribution, and then outputs $M'_j(\vec{o}_{<k})$. We note that $P_i = P_{B(N^{(i)})}$ by definition, so it suffices to show

$$\mu(P_{B(N^{(i)})}, N^{(i)}) \geq 1 - \epsilon(n) - O\left(\frac{1}{p(n)}\right). \qquad (1)$$

By definition of completeness of T, we have

$$\mu(P_{B(N^{(i)})}, B(N^{(i)})) \geq 1 - \epsilon(n). \quad (2)$$

Let us first assume for simplicity that $B(N^{(i)})$ can do the Bayesian updating exactly without any precision error. Then, we have $\mu(P_{B(N^{(i)})}, N^{(i)}) = \mu(P_{B(N^{(i)})}, B(N^{(i)}))$. To see this, we first view the "nature" $N^{(i)}$ in $\mu(P_{B(N^{(i)})}, N^{(i)})$ as a stateful nature that first chooses a machine M'_j according to the distribution $N^{(i)}$, and then uses M'_j to generate (distributions over) outcomes for all n days. Then, we note that for the first day, the output distribution of $B(N^{(i)})$ is the same as that of $N^{(i)}$. It is easy to verify by induction that for every $k \in [n]$, the output distribution of $B(N^{(i)})$ for the first k days is the same as that of $N^{(i)}$. This shows that $\mu(P_{B(N^{(i)})}, N^{(i)}) = \mu(P_{B(N^{(i)})}, B(N^{(i)}))$, so by (2) we have $\mu(P_{B(N^{(i)})}, N^{(i)}) = \mu(P_{B(N^{(i)})}, B(N^{(i)})) \geq 1 - \epsilon(n)$. Thus, (1) holds, as required.

Let us finally show that even if $B(N^{(i)})$ cannot do the Bayesian updating exactly, as long as $B(N^{(i)})$ uses at least $O(n)$ (significant) bits of precision in its computation of probabilities, B can ensure that the updated distribution obtained from Bayesian updating has at most $O(\frac{1}{2^n}) \leq O(\frac{1}{n^2 p(n)})$ relative error pointwise. This ensures that the updated distribution has an L_1-error of at most $O(\frac{1}{np(n)})$. Then, the output distribution of $B(N^{(i)})$ for all n days has an L_1-error of at most $O(\frac{1}{p(n)})$. Thus, we have

$$\left|\mu(P_{B(N^{(i)})}, N^{(i)}) - \mu(P_{B(N^{(i)})}, B(N^{(i)}))\right| \leq O\left(\frac{1}{p(n)}\right),$$

so

$$\begin{aligned}\mu(P_{B(N^{(i)})}, N^{(i)}) &\geq \mu(P_{B(N^{(i)})}, B(N^{(i)})) - O\left(\frac{1}{p(n)}\right) \\ &\geq 1 - \epsilon(n) - O\left(\frac{1}{p(n)}\right),\end{aligned}$$

where the second inequality follows from (2). Thus, (1) still holds, as required.

This completes the proof of Theorem 5. The proof of Theorem 5 can be easily extended so that the theorem holds even if Nature is modeled as a non-uniform Turing machine with $O(\log n)$ bits of non-uniform advice; in our full paper, we show that the ideas from Fortnow and Vohra [FV09] can be used to show that this is tight.

4. FEASIBILITY FOR BIASED NATURES

In this section, we investigate the possibility of achieving a positive result for forecast testing by imposing restrictions on the completeness condition. We will show that if we only require the completeness property to hold for a slightly restricted class of natures (as opposed to all possible distributions over outcomes), then we can obtain a meaningful forecast test that satisfies this restricted completeness and the soundness property as defined in Definition 7 of Section 2.1. Furthermore, our forecast test is simple and efficient. For simplicity, we shall focus on the case of two possible outcomes, i.e. $\mathcal{O} = \{0, 1\}$, and hence a forecast in $\Delta(\mathcal{O})$ can be denoted by a probability $p \in [0, 1]$ of the outcome being 1.

Informally, we say that a nature N is (α, β)-biased if for at least an α-fraction of the days, the probability p of outcome 1 is β-biased, i.e., $|p - 1/2| \geq \beta$.

Definition 9. *A nature N is (α, β)-biased if on input 1^n, for every possible n-day outcome $\vec{o} \in \{0, 1\}^n$, the following holds: For at least an α-fraction of $i \in [n]$, we have $|N(1^n, \vec{o}_{<i}) - 1/2| \geq \beta$.*

For example, suppose outcomes 1 and 0 denote raining and sunny respectively; a nature N being $(.1, .1)$-biased means that at least 10% of days are either 60% likely to be raining or 60% likely to be sunny, which seems to be a very reasonable assumption on the nature. (See [OS09] for a class of natures that is similar to our class of (α, β)-biased natures.)

For every constants $\alpha, \beta \in [0, 1]$, we will present a simple (deterministic) forecast test T that satisfies completeness with respect to all (α, β)-biased natures (including inefficient and non-uniform ones) and the soundness property (with respect to a class of non-uniform natures). We now define completeness with respect to biased natures.

Definition 10. *A test T is (α, β)-*restricted complete *if for every (α, β)-biased nature N and every $n \in \mathbb{N}$, we have*

$$\Pr[T((P_N, N)(1^n)) = 1] \geq 1 - negl(n),$$

where P_N is the canonical forecaster for N.

As a sanity check, we note that by considering (α, β)-restricted completeness, we get around the impossibility result in Section 3 and in [San03]. The reason is that a mixture of (α, β)-biased natures may not be a (α, β)-biased nature. Therefore, the (α, β)-restricted completeness does not imply that the zero-sum game defined in Section 3 and [San03] has value close to 1, and hence the constructed universal cheating forecaster C may only pass the test with poor probability. We proceed to present our formal theorem.

Theorem 6. *For every constants $\alpha, \beta \in (0, 1]$, there exists a deterministic and efficient forecast test that satisfies (α, β)-restricted completeness, and soundness with respect to the class $\mathcal{N}_{det}$ of natures with deterministic outcomes.*

We now present the desired forecast test T. On input $(\vec{p}, \vec{o}) \in (\Delta(\{0, 1\}))^n \times \{0, 1\}^n$, T does the following.

- For every $i \in [n]$, let
$$s_i = \begin{cases} p_i - 1/2 & \text{if } o_i = 1, \\ 1/2 - p_i & \text{if } o_i = 0. \end{cases}$$
- Compute $s = \sum_i s_i$, and accept iff $s \geq \alpha\beta^2 n$.

In other words, T computes some score s_i for each day, and accepts if the total score is sufficiently large. In our full paper, we show that the test T satisfies (α, β)-restricted completeness, as well as soundness with respect to the class $\mathcal{N}_{det}$ of natures with deterministic outcomes.

5. IMPOSSIBILITY OF ACHIEVING STRONG SOUNDNESS

In this section, we demonstrate that achieving strong soundness (as formally defined in Definition 8 of Section 2.1) is impossible, even in the setting of computationally-bounded forecasters, and even when only requiring completeness to hold for (α, β)-biased and computationally-efficient natures. Our result for the setting of computationally-efficient natures uses derandomization techniques from complexity theory and is based on mild complexity-theoretic assumptions.

Towards formally stating our result, we first present the necessary preliminaries as follows.

Preliminaries on derandomization. The main tool we will use is a complexity-theoretic *pseudorandom generator* (PRG) that "fools" bounded size circuits [NW94, IW97]. Let U_n denote the uniform distribution over $\{0,1\}^n$. Informally, a pseudorandom generator $G : \{0,1\}^d \to \{0,1\}^m$ is a function that takes a short d-bit uniform seed $z \leftarrow U_d$ and stretches it to a long m-bit pseudorandom string $G(z)$ that "looks like" an m-bit uniformly random string. Here, "looks like" means that for any bounded size circuit C, the probability that C outputs 1 on uniformly random input U_m is roughly the same as the probability that C outputs 1 on pseudorandom input $G(U_d)$.

DEFINITION 11. *A function $G : \{0,1\}^d \to \{0,1\}^m$ is a (s,ϵ)-pseudorandom generator if for every circuit $C : \{0,1\}^m \to \{0,1\}$ of size at most s,*

$$|\Pr[C(U_m) = 1] - \Pr[C(G(U_d)) = 1| \leq \epsilon.$$

The existing construction of a pseudorandom generator that we use is (necessarily) based on the (unproven) complexity-theoretic assumption of the existence of problems that are solvable in exponential time but has exponential circuit complexity. We first provide the necessary definitions in order to state the pseudorandom generator that we will use later.

DEFINITION 12. *The complexity class* E *consists of all languages L that can be decided by a deterministic Turing machine M in time $O(2^{cn})$ for some constant $c \in \mathbb{N}$; that is, for every $x \in \{0,1\}^*$, M on input x terminates in time $O(2^{c|x|})$ and $M(x) = 1$ iff $x \in L$.*

DEFINITION 13. *Let L be a language. The circuit complexity of L on input length n is the size of the smallest circuit C_n that decides L on input length n, i.e., for every $x \in \{0,1\}^n$, $C_n(x) = 1$ iff $x \in L$.*

The following pseudorandom generator was first achieved by Impagliazzo and Wigderson [IW97] based on a construction from Nisan and Wigderson [NW94].

THEOREM 7. *[IW97] Assume the existence of a language $L \in$ E with circuit complexity $2^{\gamma n}$ for every input length $n \in \mathbb{N}$ and some constant $\gamma > 0$. Then for every $m \in \mathbb{N}$, there exists an $(m, 1/m)$-pseudorandom generator $G : \{0,1\}^{O(\log m)} \to \{0,1\}^m$. Furthermore, G is computable in uniform poly(m) time.*

Our impossibility result for strong soundness. We are now ready to formally state our impossibility result.

THEOREM 8. *Let $\alpha \in (0,1], \beta \in (0,1/2]$ be constants. For every efficient forecast test T that satisfies (α,β)-restricted completeness, T is not computationally $(1-negl(n), (1/2) - \alpha\beta)$-strongly sound with respect to (α,β)-biased natures.*

Furthermore, if we assume the existence of a language $L \in$ E with circuit complexity $2^{\gamma n}$ for every input length $n \in \mathbb{N}$ and some constant $\gamma > 0$, then T is not computationally $(1 - 1/n, (1/2) - \alpha\beta)$-strongly sound with respect to uniform and efficient *(α,β)-biased natures.*

6. FEASIBILITY OF STRONG SOUNDNESS FOR K-SEGMENT NATURES

In this section, we consider natures that output pairwise independent samples with the same marginal distribution in each segment of k consecutive non-overlapping days. We refer to such natures as *k-segment natures*. We show that for such natures, we can achieve complete and strongly sound forecast testing (see Definition 8 in Section 2.1 for the definition of strong soundness). For simplicity, we consider the case of two possible outcomes, i.e. $\mathcal{O} = \{0,1\}$, and thus a forecast in $\Delta(\mathcal{O})$ can be represented by the probability $p \in [0,1]$ of the outcome being 1.

We first give the definition of a k-segment nature. Roughly speaking, a k-segment nature is a nature that groups the days into consecutive non-overlapping segments of k days, and within each segment, the k samples obtained from the nature are pairwise independent and have the same marginal distribution (when conditioned on the outcomes in the previous segments). Such natures can capture the testing of scientific theories, where predictions are evaluated on multiple independent or pairwise independent trials/samples.

DEFINITION 14 (k-SEGMENT NATURE). *Let $k \in \mathbb{N}$. A nature N is said to be a* k-segment nature *if on input 1^{nk}, if we group the days into n consecutive non-overlapping segments each containing k days, and if we let $o_{<i}$ denote the outcomes that occur before segment i, and if we let $q_{i,j}$ denote the output of N on day j of segment i, then for each segment $i \in [n]$ and every $o_{<i}$, the following holds:*

- *The random variables $\{q_{i,1}|_{o_{<i}}, \ldots, q_{i,k}|_{o_{<i}}\}$ are pairwise independent, where $q_{i,j}|_{o_{<i}}$ is the random variable $q_{i,j}$ conditioned on $o_{<i}$.*
- *There exists a $q_i \in [0,1]$ such that $\mathbb{E}[q_{i,j} \mid o_{<i}] = q_i$ for every $j \in [k]$.*

We now define completeness with respect to k-segment natures.

DEFINITION 15. *Let $k \in \mathbb{N}$. A test T is said to be* complete with respect to k-segment natures *if there exists a negligible function $\epsilon(\cdot)$ such that for every k-segment nature N and every $n \in \mathbb{N}$, we have*

$$\Pr[T((P_N, N)(1^{nk})) = 1] \geq 1 - \epsilon(nk),$$

where P_N is the canonical forecaster for N, and the probability is over the random coins of T and the experiment $(P_N, N)(1^{nk})$.

The definition of strong soundness can be found in Definition 8 of Section 2.1. We now formally state our feasibility result.

THEOREM 9. *Let $k \in \mathbb{N}$. Then, there exists a deterministic and efficient forecast test T that satisfies the following properties:*

- *T is complete with respect to k-segment natures.*
- *T is (ϵ,δ)-strongly sound with respect to k-segment natures, where ϵ is a negligible function, and $\delta(n) = O(\frac{1}{k^{1/3}} + \frac{1}{n^{1/3}})$.*

We now present the desired forecast test T. On input $(\vec{p}, \vec{o}) \in (\Delta(\{0,1\}))^{nk} \times \{0,1\}^{nk}$, T does the following:

- For each segment $i \in [n]$, compute the average $\widehat{q}_i$ of the outcomes in segment i.
- Compute $\frac{1}{n}\sum_{i=1}^{n} |p_i - \widehat{q}_i|$, where p_i is the forecast in $\vec{p}$ for the first day of segment i.
- Output 1 (accept) if $\frac{1}{n}\sum_{i=1}^{n} |p_i - \widehat{q}_i| \leq \frac{2}{k^{1/3}} + \frac{1}{n^{1/3}}$; otherwise, output 0 (reject).

In our full paper, we show that the test T satisfies the properties described in Theorem 9.

7. ACKNOWLEDGMENTS

We thank Bobby Kleinberg and Lance Fortnow for helpful discussions.

8. REFERENCES

[BG92] Mihir Bellare and Oded Goldreich, *On defining proofs of knowledge*, CRYPTO '92, 1992, pp. 390–420.

[Bic07] J. Eric Bickel, *Some comparisons among quadratic, spherical, and logarithmic scoring rules*, Decision Analysis **4** (2007), no. 2, 49–65.

[BM88] László Babai and Shlomo Moran, *Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity class*, J. Comput. Syst. Sci. **36** (1988), no. 2, 254–276.

[Bri50] Glenn W. Brier, *Verification of forecasts expressed in terms of probability*, Mon. Wea. Rev. **78** (1950), no. 1, 1–3.

[Daw82] A. P. Dawid, *The well-calibrated bayesian*, Journal of the American Statistical Association **77** (1982), no. 379, pp. 605–610.

[DF06] Eddie Dekel and Yossi Feinberg, *Non-bayesian testing of a stochastic prediction*, Review of Economic Studies **73** (2006), no. 4, 893–906.

[FS89] Uriel Feige and Adi Shamir, *Zero knowledge proofs of knowledge in two rounds*, CRYPTO, 1989, pp. 526–544.

[FS99] Yoav Freund and Robert E. Schapire, *Adaptive game playing using multiplicative weights*, Games and Economic Behavior **29** (1999), no. 1, 79–103.

[FV98] Dean P. Foster and Rakesh V. Vohra, *Asymptotic calibration*, Biometrika **85** (1998), no. 2, 379–390.

[FV09] Lance Fortnow and Rakesh V. Vohra, *The complexity of forecast testing*, Econometrica **77** (2009), no. 1, 93–105.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff, *The knowledge complexity of interactive proof systems*, SIAM Journal on Computing **18** (1989), no. 1, 186–208.

[GR07] Tilmann Gneiting and Adrian E. Raftery, *Strictly proper scoring rules, prediction, and estimation*, Journal of the American Statistical Association **102** (2007), 359–378.

[IW97] Russell Impagliazzo and Avi Wigderson, *P = BPP if E requires exponential circuits: derandomizing the XOR lemma*, Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, STOC '97, ACM, 1997, pp. 220–229.

[Leh01] Ehud Lehrer, *Any inspection is manipulable*, Econometrica **69** (2001), no. 5, pp. 1333–1347 (English).

[MP07] Silvio Micali and Rafael Pass, *Precise zero knowledge*, 2007.

[NW94] Noam Nisan and Avi Wigderson, *Hardness vs randomness*, J. Comput. Syst. Sci. **49** (1994), no. 2, 149–167.

[OS06] W Olszewski and A Sandroni, *Counterfactual predictions*, 2006.

[OS09] Wojciech Olszewski and Alvaro Sandroni, *Strategic manipulation of empirical tests*, Math. Oper. Res. **34** (2009), no. 1, 57–70.

[San03] Alvaro Sandroni, *The reproducible properties of correct forecasts*, International Journal of Game Theory **32** (2003), no. 1, 151–159.

[VS03] Vladimir Vovk and Glenn Shafer, *Good randomized sequential probability forecasting is always possible, the game-theoretic probability and finance project, http://probabilityandfinance.com, working paper*, the Journal of the Royal Statistical Society B **67** (2003), 747–763.