

Bits per Spike as a Betting Game

A simple way to interpret heldout log-likelihood scores

Alex H Williams
New York University
alex.h.williams@nyu.edu

Revision history
v1 (2026-05-27): Initial version.
v2 (2026-05-30): Added time to significance.

doi: [10.5281/zenodo.20418560](https://doi.org/10.5281/zenodo.20418560)

Whenever we fit a model to neural or behavioral data, we need to benchmark it against simpler or well-known baselines. Typically this is done by reporting the difference in log-likelihoods on heldout data. For example, the popular “bits per spike” performance metric (see e.g. [Pillow et al. 2008](#)) is simply the log (base 2) likelihood of the model minus the log (base 2) likelihood of a homogeneous Poisson process (or another appropriate baseline model), divided by the total number of spikes in the dataset.

This post offers some thoughts on how we can interpret this performance measure. For example, if my model gives a 0.34 bits per spike improvement over the baseline, should I interpret that as very good? Marginal? Completely inconsequential?

I have struggled to answer these questions satisfactorily and this post is part of my attempt to rectify this gap in my understanding. I’ll focus on a particular interpretation that imagines the model playing a betting game against the “market” defined by the baseline model. This game-theoretic framing has gained traction in a certain corner of statistics (see [Further Reading](#) for a brief list of introductory references).

Basic Setup

Suppose that we have fit a model Q and a baseline B on training data and that we’re now ready to compare them head-to-head on heldout test data. Let X_1, X_2, X_3, \dots denote a (potentially infinite) sequence of heldout data samples. We assume these samples are independent and identically distributed according to an unknown distribution P . Our hope is that Q is in some sense “closer” to the true distribution P than the baseline model B .

For simplicity, we’ll assume that Q and B have strictly positive probability density or mass functions $q(x) > 0$ and $b(x) > 0$. This allows us to compute *likelihood ratios*, $q(X)/b(X)$ for $X \sim P$, without having to worry about divide-by-zero errors.¹

¹ It is possible to make the math work without making this assumption. Generally, this would let us turn “> relations” into “ \geq relations” and likewise turn “ \leq relations” into “< relations”. However, for simplicity we just stick to strict inequalities for the purpose of this post.

A nice measure of performance is the expected log-likelihood ratio:

$$\mathcal{L} = \mathbb{E}_{X \sim P} \left[\log q(X)/b(X) \right] = \mathbb{E}_{X \sim P} \left[\log q(X) - \log b(X) \right] \quad (1)$$

Note that the expectation is computed under P . Since P is unknown in real world situations, we can estimate the expression above by holding out a test set with T data samples and approximating the expectation with an empirical average:

$$\mathcal{L} \approx \hat{\mathcal{L}} = \frac{1}{T} \sum_{t=1}^T \log q(X_t)/b(X_t) = \frac{1}{T} \sum_{t=1}^T \left[\log q(X_t) - \log b(X_t) \right] \quad (2)$$

We have thus far used natural logarithms, but substituting base-2 logarithms can aid interpretation. By the change of base formula, $\mathcal{L}_2 = \mathcal{L}/\log(2)$ is the expected base-2 log likelihood. The base-2 log likelihood has units of “bits”, and normalizing by the number of spikes gives the popular “bits per spike” metric.

Even more intriguingly, we will be able to draw a connection to null hypothesis testing at significance threshold α (for example, $\alpha = 0.05$). It turns out that the reciprocal of the expected base- $(1/\alpha)$ log likelihood, $\mathcal{L}_{1/\alpha}^{-1} = -\log(\alpha)/\mathcal{L}$, will have a very satisfying interpretation as the number of heldout data samples needed to reject the baseline model as a null hypothesis.

You don't need to understand the betting game to use this result! If you wish to skip the derivation, you can jump straight to the [Take Home Message](#) at the end of this post.

A Demo of the Betting Game

Our goal is to come up with intuitive interpretations of \mathcal{L} as a measure of model performance. One way to approach this is to imagine model Q as a “player” in a betting game. On each round of the game, we sample a heldout datapoint $X \sim P$ and pay the player based on how well they predicted the outcome. Each betting contract comes at a price, which is determined by the baseline model B . Intuitively, the more that Q is able to “beat the market” by accumulating wealth, the stronger the evidence we have in favor of Q over B .

Before we get to the precise details of the game, let's look at a few examples. Figure 1 shows tuning curves fit to three example head-direction cells from the mouse anterior thalamus (Peyrache et al., 2015), distributed through the [nemos](#) library. Each cell's spiking is modulated by the animal's current head direction, and binning spike counts by head-direction angle gives *empirical firing rates* (grey dots). We then discretized spike trains using $\Delta = 0.05$ second time bins and used [nemos](#) to fit a GLM model with cyclic B-spline basis functions (solid line, Q), and a homogeneous Poisson process as a baseline model (dashed grey line).

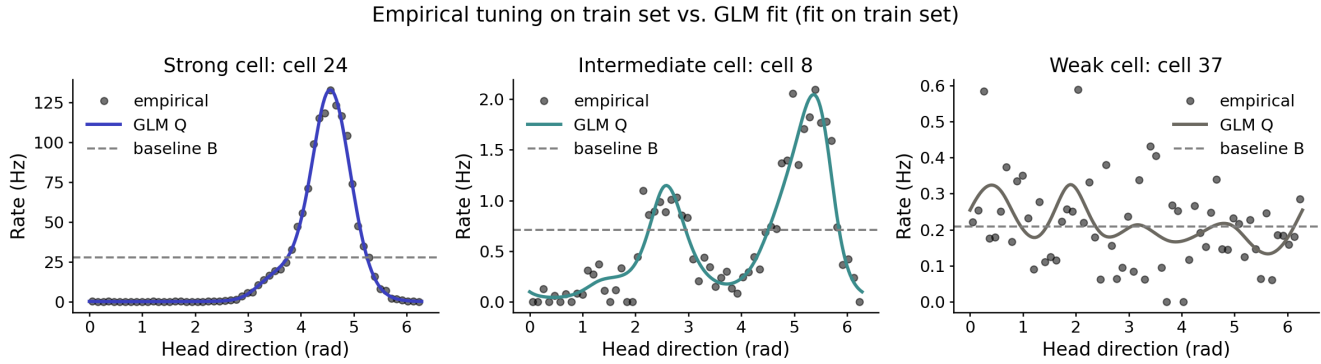


Figure 1. Three example cells with strong, intermediate, and weak head direction modulation. For each cell we fit a GLM model, Q , and a flat baseline model, B .

The plots above were generated using 50% of the observations as “training data”. The remaining 50% of the observations serve as heldout “test data” which we use below to play the forecasting game.

In the game, player Q gambles their wealth over discrete rounds of betting. In each round, the player forecasts and places a bet on the number of spikes they’ll see in the next time bin. Intuitively, if Q is a much better model of the data than B then the player has an “edge” that they can exploit and generate returns very quickly—in fact, it turns out to be exponentially fast.

Throughout, we will use W_t to denote the wealth of the player at round t of the game. In Figure 2, we plot how the player’s wealth W_t evolves across the test set for each of the three cells. Every player starts with $W_0 = 1$ and the y-axis is on a \log_2 scale, so a value of $\log_2 W_t = 10$ means the player has doubled their wealth ten times (a $2^{10} = 1024\times$ return). For each cell we ran ten independent random train/test splits of the recording and overlaid all ten trajectories.²

Notice that the x-axis range differs across panels — the strong cell’s wealth game unfolds over 100 time bins (5 seconds), while the weak cell’s takes the full ~15 minutes of test data.

² It is worth pointing out an important approximation. We use ten independently generated train and test splits to generate each random wealth trajectory. Thus, we fit ten different GLM models and have them “play the betting game” on ten different heldout test sets. Thus, some of the variability in the trajectories is due to differences in the training data. Our exposition of the betting game does not account for this. Another approximation worth noting is that we do not account for autocorrelation in the observations—we shuffle the order of time bins on each split. Extending the betting game to time series data with autocorrelation could be important for comparing certain models of neural data.

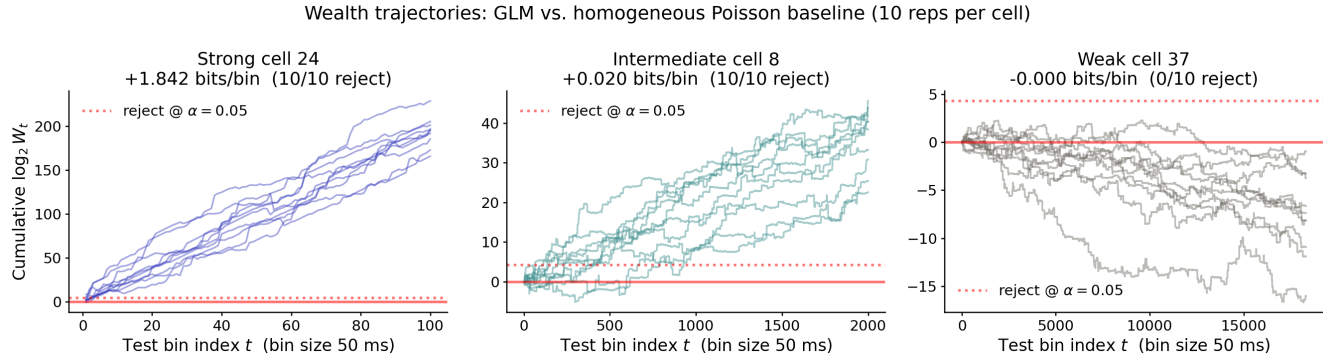


Figure 2. Wealth trajectories for the three example cells, ten random train/test splits each. The y-axis is on a \log_2 scale (number of doublings of the starting wealth). The dashed red horizontal line at $\log_2 W_t \approx 4.3$ marks a threshold we will motivate later. Each panel uses its own x-axis range so the dynamics of each cell are visible at an appropriate scale.

For the **strong cell** (left panel), the player’s wealth grows very fast—roughly linearly on the \log_2 scale at a rate of ~ 1.8 units of log wealth per round. Each round, the player nearly quadruples their wealth (on average)! The ten trajectories cluster tightly because the underlying signal is strong enough that essentially any random split of the recording produces nearly the same model and the same betting edge.

For the **intermediate cell** (middle panel) we see the same qualitative pattern — linear growth on the \log_2 scale — but at a much shallower slope of about 0.02 units of log wealth per round. This means that it takes the player $1/0.020 = 50$ rounds to double their wealth, on average.

For the **weak cell** (right panel), the player has effectively no edge at all. Wealth wanders around the starting value of $W_0 = 1$ and drifts slightly *downward* on average; across all ten splits, the trajectories stay well below the dashed threshold line at the top of the panel. This is the behavior we hope to see whenever a model offers no genuine improvement over the baseline: betting on a useless predictor is, in the long run, a losing strategy.

By the end of this post, we’ll see that expected log likelihood ratios describe the rate of wealth growth. For example, $1/\mathcal{L}_2$ describes the number of rounds of betting needed (on average) to double the player’s wealth. Similarly, if we normalize by time bin size, computing Δ/\mathcal{L}_2 , we obtain an estimate of how much heldout data we need (in recording duration) to double the player’s wealth.

We can also connect this to a null hypothesis test with significance level α . It turns out that if the player ever generates more than $1/\alpha$ units of wealth, we can reject the null hypothesis that the true distribution, P , is equal to the baseline model, B , with type-I error rate less than α . For example, if we set $\alpha = 0.05$, then we can reject the null hypothesis if we ever obtain $W_t > 1/\alpha = 20$.

In light of this, the expected base- $(1/\alpha)$ log likelihood ratio, denoted $\mathcal{L}_{1/\alpha}$, tells us the rate at which we accumulate evidence to reject the null hypothesis that $P = B$. Similarly, $1/\mathcal{L}_{1/\alpha}$ and $\Delta/\mathcal{L}_{1/\alpha}$ respectively tell us how much data we need (on average) to reject the null in units of time bins and seconds. Figure 3 below re-plots the data from Figure 2 on this log scale. Further, the x-axis is plotted in units of recorded time (by dividing by time bin size Δ). The units of the y-axis now correspond to the number of times the player has accumulated enough wealth to reject the null hypothesis at significance level α . For the strong cell (*left*) that timescale is around 120 ms; for the intermediate cell it is around 11 seconds; for the weak cell the trajectories never cross the threshold, and we therefore fail to reject the null.

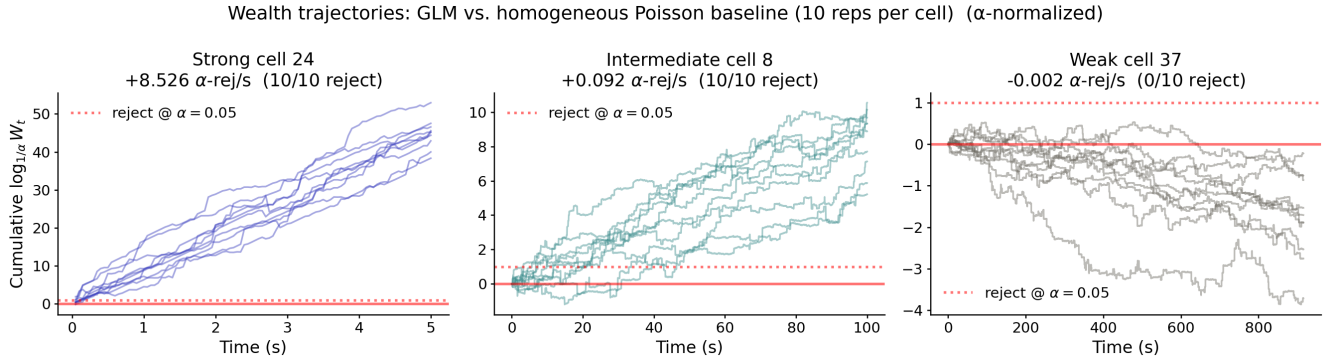


Figure 3. The same wealth trajectories as Figure 2, now plotted in units of $\log_{1/\alpha} W_t$ with $\alpha = 0.05$. The dashed red line at $y = 1$ marks the wealth required to reject the null at level α . The slope of each trajectory is the rate at which the player accumulates rejection-worth of evidence.

A common alternative normalization is *bits per spike*, the metric mentioned at the top of the post. If $\bar{\lambda}_b$ is the baseline’s expected spike count per bin, then $\text{bits/spike} = \mathcal{L}_2/\bar{\lambda}_b$ coincides with the player’s wealth growth per spike observed, rather than per bin elapsed. Normalizing by the number of spikes is potentially useful when comparing models fit to neurons with very different firing rates. However, since the betting game is naturally indexed by time bins of the recording, the rest of the post will stick to normalizing the log likelihood ratio in units of time.

Now that we’ve sketched some results, it’s time to make this picture rigorous. Specifically, we want to (1) define exactly how each round of betting works, (2) explain the optimal strategy that player Q can implement, and (3) understand what the significance threshold line in the figure means in terms of a formal hypothesis test.

Introducing the Game

The game starts by giving player Q one unit of wealth:

$$W_0 = 1$$

At each round of the game, player Q uses all of their wealth to purchase *prediction contracts*, specified by a function $C(x) > 0$. On each round, indexed by t , we sample an observation $X_t \sim P$ and pay the player $C(X_t)$ units of wealth. Thus, the player receives a random sequence of returns $C(X_1), C(X_2), C(X_3), \dots$ over discrete rounds of the game. The wealth updates according to:

$$W_t = (W_{t-1}/\pi(C)) \cdot C(X_t). \tag{3}$$

where $\pi(C)$ denotes the *price* of the contract.

Equation (3) is simple. The first term, $W_{t-1}/\pi(C)$, is the *number of contracts* the player is able to purchase given their previous wealth at round $t - 1$. The second term, $C(X_t)$, is the payoff of each contract. Note that we allow the wealth and the number of purchased contracts to be infinitely divisible into fractions.

Intuitively, the price of a contract is set by what people are willing to buy and sell it for, which reflects their expectations about the underlying distribution P . For the purposes of our game we’ll assume that the market consensus—or the “wisdom of the crowd”—coincides with the baseline model B . Formally, it turns

out that the fair price of a contract is given by its expected value under the market consensus distribution. That is,

$$\pi(C) = \mathbb{E}_{X \sim B} [C(X)]. \quad (4)$$

See [Supplementary Note 1](#) for a quick derivation of this equation.

The pricing constraint in equation (4) allows us to simplify the structure of the game by assuming that the contracts have unit price. Indeed, for any contract $C(\cdot)$, we can define a new contract $S(x) = C(x)/\pi(C)$ which has unit price, $\pi(S) = 1$. The wealth update for C and S is equivalent since

$$(W_{t-1}/\pi(C)) \cdot C(X_t) = W_{t-1} \cdot S(X_t),$$

by the definition of S . Therefore, for the rest of this post we will focus on the simplified wealth process

Simplified wealth process. Assuming that the player purchases positive contracts $S(x) > 0$ of unit price, i.e.

$$\mathbb{E}_{X \sim B} [S(X)] = 1 \quad (5)$$

then the player's wealth evolves according to

$$W_t = W_{t-1} \cdot S(X_t). \quad (6)$$

Choosing the optimal contract function

Player Q is allowed to choose the function $S(\cdot)$ however they like, so long as it satisfies (5). Out of this space of feasible contracts, which one should Q choose to play?

After T rounds of betting according to equation (6), the player will accumulate

$$W_T = \prod_{t=1}^T S(X_t) = \exp \log \prod_{t=1}^T S(X_t) \quad (7)$$

$$= \exp \sum_{t=1}^T \log S(X_t) \quad (8)$$

$$= \exp \left(T \cdot \left(\frac{1}{T} \sum_{t=1}^T \log S(X_t) \right) \right) \quad (9)$$

$$\approx \exp \left(T \cdot \mathbb{E}_{X \sim P} \log S(X) \right) \quad (10)$$

units of wealth. The approximation in the final line comes from replacing the empirical expectation $\frac{1}{T} \sum_{t=1}^T \log S(X_t)$ with the true expected value under P .

Since the player believes that $P = Q$, they anticipate that their wealth can grow exponentially over time according to:

$$W_T \approx \exp \left(T \cdot \mathbb{E}_{X \sim Q} \log S(X) \right) \quad (11)$$

To maximize their rate of wealth growth, a reasonable strategy is to choose $S(\cdot)$ in order to

$$\text{maximize } \mathbb{E}_{X \sim Q} \log S(X) \quad \text{subject to (5)} \quad (12)$$

Quite pleasingly, as shown in [Supplementary Note 2](#), the solution to this optimization problem, denoted S^* , turns out to be the likelihood ratio!

$$S^*(x) = \frac{q(x)}{b(x)} \quad (13)$$

Combining equations (13) and (10) with the definition of \mathcal{L} in (1), we see that the long-run wealth of the player is approximated by

$$W_T \approx \exp(\mathcal{L} \cdot T) \quad (14)$$

for large T . In other words, the wealth accumulated by player Q in the game will, over the long run, grow or decay exponentially fast at a rate given by the expected log-likelihood ratio.

Some interpretations

We are now in a position to rigorously reinterpret Figure 2. Taking logarithms on both sides of (14) and changing to base-2 yields a linear relationship between the log-wealth and time bin index:

$$\log_2(W_T) \approx \mathcal{L}_2 \cdot T. \quad (15)$$

The reciprocal $1/\mathcal{L}_2$ is therefore the average *doubling time* of the wealth process in units of time bins (or equivalently, rounds of betting). Multiplying by the time bin size, Δ/\mathcal{L}_2 , changes the units of the doubling time to seconds of recording. For example, in Figure 2, the strong cell (*left*) doubles its wealth every ~25 ms while the intermediate cell (*middle*) doubles its wealth every ~2.5 seconds.

It is also interesting to note a connection to the [KL divergence](#). While \mathcal{L} determines the *actual* rate of wealth growth, the player’s anticipated rate of wealth growth is given by replacing P in (10) with Q , yielding:

$$\mathbb{E}_{X \sim Q} \left[\log q(X)/b(X) \right] \quad (16)$$

which is precisely the KL divergence, $D_{\text{KL}}(Q \parallel B)$.³

Figure 4 shows the same wealth trajectories as in Figure 3 with the expected rate of wealth growth, given by $D_{\text{KL}}(Q \parallel B)$, overlaid as a dark black line. For the strong and intermediate cells (*left* and *middle* panels), the anticipated and realized growth rates agree, indicating that the GLM model is well-calibrated on this dataset. For the weak cell, the anticipated line climbs gradually above zero while the realized trajectories drift slightly below. This is suggestive of overfitting: Q believes it has a tiny edge that doesn’t actually exist on heldout data.

³ This interpretation of KL divergence is due to JL Kelly Jr. in a [tech report from 1956](#). The principle that the player should choose their bet to maximize the term in the exponent appearing in (10) is named after him—it is known as the [Kelly criterion](#) in quantitative finance.

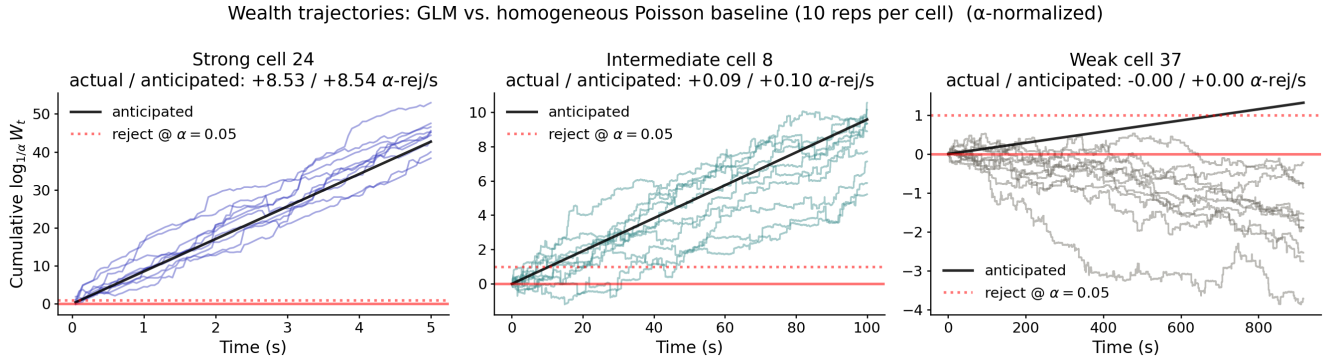


Figure 4. Same wealth trajectories as Figure 3, now with the player’s anticipated rate of wealth growth overlaid as a black line. The slope of the line is $D_{\text{KL}}(Q \parallel B)$ per round of betting, averaged across the ten train/test splits. For the strong and intermediate cells, the anticipated and realized trajectories agree closely. For the weak cell, the anticipated line drifts slightly above zero while the realized trajectories drift slightly below — a small but real signature of overfitting on the training data.

Connection to Hypothesis Testing

We have not yet drawn a rigorous connection to null hypothesis testing, as promised by the plots in Figures 3 and 4. The connection is simple and follows concretely from a result called [Ville’s inequality](#), summarized below.

Ville’s Inequality (informal). Let $(M_t)_{t \geq 0}$ be a sequence of random variables such that $M_t \geq 0$ almost surely for all t and $\mathbb{E}[M_t \mid M_0, \dots, M_{t-1}] \leq M_{t-1}$ for all t . Then for any $\alpha > 0$,

$$\Pr\left(\sup_{t \geq 0} M_t \geq 1/\alpha\right) \leq \alpha \cdot \mathbb{E}[M_0].$$

Intuitively, this result states that if each step of the random sequence $(M_t)_{t \geq 0}$ is not increasing in expectation,⁴ then the probability of $(M_t)_{t \geq 0}$ ever crossing above a high threshold is small, even if we run the simulation infinitely far into the future. [Supplementary Note 3](#) sketches a quick proof of Ville’s inequality.

To see why this matters in our setting, suppose that the baseline B is in fact the true data-generating distribution—i.e. $P = B$. Then we would have

$$\mathbb{E}_{X_t \sim B}[W_t \mid W_0 \dots W_{t-1}] = W_{t-1} \cdot \mathbb{E}_{X_t \sim B}\left[\frac{q(X_t)}{b(X_t)}\right] = W_{t-1}.$$

since the expectation of the likelihood ratio equals one.⁵ This means that we can apply Ville’s inequality, which tells us that the probability of player Q ’s wealth ever exceeding the threshold $1/\alpha$ —at any round of

⁴ For those who appreciate jargon, we call $(M_t)_{t \geq 0}$ a *supermartingale* if it satisfies $\mathbb{E}[M_t \mid M_0, \dots, M_{t-1}] \leq M_{t-1}$. In the stricter case where the inequality is saturated, i.e. $\mathbb{E}[M_t \mid M_0, \dots, M_{t-1}] = M_{t-1}$, we call $(M_t)_{t \geq 0}$ a *martingale*. Ville’s inequality says that all nonnegative supermartingales (and martingales) are upper bounded for all time with high probability.

⁵ Concretely, $\mathbb{E}_{X \sim B}\left[\frac{q(X)}{b(X)}\right] = \int \frac{q(x)b(x)}{b(x)} dx = \int q(x) dx = 1$.

the game, even infinitely far into the future—is at most α :

$$P\left(\sup_{t \geq 0} W_t \geq 1/\alpha\right) \leq \alpha.$$

This furnishes an *anytime-valid hypothesis test* of the null $H_0 : P = B$. We may reject H_0 at level α as soon as W_t crosses $1/\alpha$, regardless of how many rounds have been played. Unlike classical fixed-sample tests, we are free to peek at the data, stop early, or keep collecting more samples, all without inflating the type I error rate. For example, if we use $\alpha = 0.05$ (as is customary), then we can reject the null hypothesis that $P = B$ if player Q 's wealth *ever* exceeds 20.

Take Home Message

A central message of this post is that the expected log-likelihood ratio, \mathcal{L} , can be interpreted as the exponential rate at which a player generates wealth in a betting game based on forecasting future events. For example, $\log(2)/\mathcal{L}$ defines the time it takes for the player to double their wealth (in units of time bins).

The betting game interpretation is satisfying, but it involves some effort to conceptualize and formally derive. In an effort to simplify the take home message, I propose two main summary statistics: the **samples to significance**, τ , and the **time to significance**, given by $\tau \Delta$.

Time to Significance. Let $\mathcal{L} > 0$ denote the expected log-likelihood ratio of a model Q relative to baseline B , as in equation (1). The samples to significance,

$$\tau = \frac{-\log(\alpha)}{\mathcal{L}},$$

reflects the number of heldout samples needed on average to reject the null hypothesis $H_0 : P = B$ at significance level α . If each sample is a time bin of Δ seconds, the corresponding time to significance is given by $\tau \Delta = -\Delta \log(\alpha)/\mathcal{L}$.

In practice, we estimate these quantities by first fitting Q and B on training data and using heldout test data to compute an estimate of the expected log-likelihood ratio, $\hat{\mathcal{L}}$ in (2). The time to significance is a summary statistic of model performance that can be used in place of, or complementary to, the conventional “bits per spike” metric. For the three example cells shown, this gives $\tau \Delta \approx 120$ ms (strong) and $\tau \Delta \approx 11$ s (intermediate); for the weak cell $\hat{\mathcal{L}} < 0$, so the threshold is never crossed and τ is undefined.

Because τ is a strictly decreasing function of \mathcal{L} , it ranks models identically to the heldout log-likelihood (and hence to bits per spike). Thus, it is not a fundamentally new statistic, but a more interpretable *unit* for an existing one, answering the question: “how much data do I need to confirm that Q beats B ?” Quantities similar to the time to significance appear in prior literature cited below, as well as in the context of [Wald’s sequential probability ratio test](#).

Further Reading

Review paper by Ramdas, Grünwald, Vovk, and Shafer (2023). [“Game-Theoretic Statistics and Safe Anytime-Valid Inference.”](#) *Statist. Sci.* 38 (4) 576-601.

YouTube Tutorial Lectures by Ramdas, “A Martingale Theory of Evidence” ([Part I](#)) ([Part II](#)) ([Part III](#))

Textbook by Ramdas and Wang (2025). “[Hypothesis Testing With E-Values.](#)”

Supplementary Note 1

Here we sketch how the market price constraint (4) arises in more detail. For simplicity, let’s consider a case where the random outcomes $X \sim P$ take on one of n discrete values. That is, $X \in 1, \dots, n$ almost surely. The same argument can be extended to continuous-valued random variables with sufficient care.

By assuming there are only n discrete outcomes, then we can express any potential contract function $C(\cdot)$ as a finite linear combination of elementary basis functions:

$$C(x) = \sum_{i=1}^n r_i \delta_i(x) \quad (17)$$

where r_1, \dots, r_n are scalar coefficients denoting the return of outcome $x = i$,

$$r_i = C(i) , \quad (18)$$

and $\delta_1, \dots, \delta_n$ are contracts that pay off one unit of wealth if the outcome is $x = i$,

$$\delta_i(x) = \begin{cases} 1 & x = i \\ 0 & x \neq i \end{cases} .$$

Recall that our goal is to show $\pi(C) = \mathbb{E}[C(X)]$, with the expectation taken with respect to some appropriate choice of distribution B . We will show that this is true if the market is organized such that no player can receive “free money” without taking any risk (in other words, there are no [arbitrage](#) opportunities). We assume that any player can either buy or sell contracts, and that [fractional contracts](#) are supported in the market (e.g. by sharing contracts with other players).

Observation 1 – The pricing function π must be linear. For any scalar $a > 0$, we must have $\pi(a \cdot C) = a\pi(C)$. If $\pi(a \cdot C) > a \cdot \pi(C)$ a player would get free money by selling the contract $a \cdot C$ and simultaneously buying a contracts of C . For any outcome $X \sim P$, the player neither gains nor loses any wealth and they pocket $\pi(a \cdot C) - a \cdot \pi(C) > 0$ units of wealth. Conversely, if $\pi(a \cdot C) < a \cdot \pi(C)$ a player would get free money by buying the contract $a \cdot C$ and simultaneously selling a contracts of C . An arbitrage-free pricing scheme therefore must satisfy $\pi(a \cdot C) = a\pi(C)$.

Next, for any two contracts C_1 and C_2 we must have that $\pi(C_1 + C_2) = \pi(C_1) + \pi(C_2)$. The argument is quite similar to above. If $\pi(C_1 + C_2) > \pi(C_1) + \pi(C_2)$ then a player would get free money by selling the contract $C_3(x) = C_1(x) + C_2(x)$ and simultaneously buying C_1 and C_2 . Since $C_1(x) + C_2(x) - C_3(x) = 0$ for all x , the player neither gains nor loses any wealth based on the random outcome and they pocket $\pi(C_3) - \pi(C_1 + C_2) > 0$ units of wealth. Conversely, if $\pi(C_1 + C_2) < \pi(C_1) + \pi(C_2)$ then run the same trade in reverse to get free money.

Taken together, we conclude that pricing must be linear. Applying this to (17) we can conclude that the price of any contract can be written down as:

$$\pi(C) = \sum_{i=1}^n r_i \pi(\delta_i) \quad (19)$$

Thus, we can determine the price of any contract by determining the prices of the elementary basis contracts $\delta_1, \dots, \delta_n$.

Observation 2 – the prices $\pi(\delta_1), \dots, \pi(\delta_n)$ are nonnegative and sum to one. First we prove nonnegativity. The random payout for each contract, $\delta_i(X)$ for $X \sim P$, is greater than or equal to zero almost surely. Thus, the price of each contract must be nonnegative—if it were negative, it would imply that the player is *paid* to accept a contract that never results in a loss (i.e. receive free money).

Next we prove the normalization condition that $\sum_i \pi(\delta_i) = 1$. Note that we can construct a contract with constant payoff $C(X) = 1$, almost surely, by setting $r_1 = r_2 = \dots = r_n = 1$ in equation (17). If the price of this contract were less than one, a player would get free money by purchasing it. Likewise, if the price were greater than one, a player would get free money by selling it.

Putting it together. From observation 2, it is clear that the prices $\pi(\delta_1), \dots, \pi(\delta_n)$ define a probability measure over outcomes $x \in \{1, \dots, n\}$. Call this probability measure B . Then, recalling that $r_i = C(i)$ denotes the return of outcome $x = i$ under the contract, we deduce from equation (19):

$$\pi(C) = \sum_{i=1}^n r_i \pi(\delta_i) = \mathbb{E}_{X \sim B} [C(X)]$$

confirming our claim that the price of a contract is given by the expected payoff of the contract under an appropriate distribution B .

Supplementary Note 2

We prove that the optimization problem stated in (12) is solved by the likelihood ratio $S^*(x) = q(x)/b(x)$ given in (13).

The argument relies on a useful reparameterization. Let $r(x) = b(x) S(x)$ and note that $r(x)$ is a probability density. Indeed, $r(x) > 0$ since both b and S are strictly positive, and

$$\int r(x) dx = \int b(x) S(x) dx = \mathbb{E}_{X \sim B} [S(X)] = \pi(S) = 1$$

by the unit-price constraint (5). Conversely, since we assumed $b(x) > 0$ everywhere on the support of P , any probability density r defines a feasible contract function via $S(x) = r(x)/b(x)$.

Now substitute $S(x) = r(x)/b(x)$ into the objective in (12), then add and subtract $\mathbb{E}_{X \sim Q} \log q(X)$:

$$\mathbb{E}_{X \sim Q} \log S(X) = \mathbb{E}_{X \sim Q} \log \frac{r(X)}{b(X)} \tag{20}$$

$$= \mathbb{E}_{X \sim Q} \log \frac{q(X)}{b(X)} - \mathbb{E}_{X \sim Q} \log \frac{q(X)}{r(X)} \tag{21}$$

$$= D_{\text{KL}}(Q \parallel B) - D_{\text{KL}}(Q \parallel R). \tag{22}$$

The first term, $D_{\text{KL}}(Q \parallel B)$, does not depend on the player's choice of contract. The second term, $D_{\text{KL}}(Q \parallel R)$, is non-negative by [Gibbs' inequality](#), with equality if and only if $R = Q$. To maximize the objective, the

player should therefore choose $R = Q$, that is, $r(x) = q(x)$. Translating back to a contract function via $S = r/b$ yields

$$S^*(x) = \frac{q(x)}{b(x)},$$

as claimed in (13). The maximum achievable value of the objective is $D_{\text{KL}}(Q \| B)$ — the exponential rate at which the player anticipates their wealth will grow, recovering the Kelly-criterion interpretation discussed in the main text.

Supplementary Note 3

Here we give an informal proof sketch of Ville's inequality. The remarkable part of this result is that the inequality holds *uniformly over all time t* . So if we simulated a very large number of wealth trajectories (i.e. player Q gets to restart and play the game many times), then only a small fraction, α , of the trajectories would ever cross above $1/\alpha$, *even if each individual trajectory were simulated infinitely long*.

Recall that we are given a discrete-time sequence $(M_t)_{t \geq 0}$, satisfying $\mathbb{E}[M_t | M_0, \dots, M_{t-1}] \leq M_{t-1}$ and $M_t \geq 0$ for all t . A sequence with these two properties is called a *nonnegative supermartingale*.

Let c denote the first time that the trajectory crosses above a threshold $\lambda > 0$. That is, c is the smallest natural number such that $M_c \geq \lambda$. If the trajectory *never* crosses above λ , we take this as meaning $c = \infty$. Our goal is to prove

$$\Pr(c \text{ is finite}) \leq \frac{\mathbb{E}[M_0]}{\lambda} \quad (23)$$

which is more-or-less identical to the result we want. In the main post, we chose the threshold to be $\lambda = 1/\alpha$.

The trick is to define a new *nonnegative supermartingale* $(Z_t)_{t \geq 0}$ as

$$Z_t = \begin{cases} M_t & \text{if } t < c \\ \lambda & \text{if } t \geq c \end{cases} \quad (24)$$

Note that if $c = \infty$, then M_t never crosses above the threshold and Z_t is just a copy of M_t for all t . It is easy to show that if $(M_t)_{t \geq 0}$ is a nonnegative supermartingale, then so is $(Z_t)_{t \geq 0}$ (exercise to the reader).

Now for any fixed value of t , we can apply [Markov's inequality](#) to conclude that

$$\Pr(c \leq t) = \Pr(Z_t \geq \lambda) \leq \frac{\mathbb{E}[Z_t]}{\lambda} \leq \frac{\mathbb{E}[Z_0]}{\lambda} \quad (25)$$

The final inequality follows from the fact that $(Z_t)_{t \geq 0}$ is a nonnegative supermartingale, which implies $\mathbb{E}[Z_t] \leq \mathbb{E}[Z_0]$.

Further, $\mathbb{E}[Z_0] \leq \mathbb{E}[M_0]$, since Z_t is a clipped version of M_t . Thus, we have:

$$\Pr(c \leq t) \leq \frac{\mathbb{E}[M_0]}{\lambda}. \quad (26)$$

Notice that the right hand side is not a function of t . There is a way to rigorously take the limit of $t \rightarrow \infty$ on both sides of the inequality to yield our desired result, equation (23).