entropy

MDPI

Article

Entropy-Based Strategies for Multi-Bracket Pools

Ryan S. Brill [1,*], Abraham J. Wyner [2] and Ian J. Barnett [3]

[1] Graduate Group in Applied Mathematics and Computational Science, University of Pennsylvania, Philadelphia, PA 19104, USA
[2] Department of Statistics and Data Science, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104, USA
[3] Department of Biostatistics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA
* Correspondence: ryguy123@sas.upenn.edu

Abstract: Much work in the parimutuel betting literature has discussed estimating event outcome probabilities or developing optimal wagering strategies, particularly for horse race betting. Some betting pools, however, involve betting not just on a single event, but on a tuple of events. For example, pick six betting in horse racing, March Madness bracket challenges, and predicting a randomly drawn bitstring each involve making a series of individual forecasts. Although traditional optimal wagering strategies work well when the size of the tuple is very small (e.g., betting on the winner of a horse race), they are intractable for more general betting pools in higher dimensions (e.g., March Madness bracket challenges). Hence we pose the multi-brackets problem: supposing we wish to predict a tuple of events and that we know the true probabilities of each potential outcome of each event, what is the best way to tractably generate a set of n predicted tuples? The most general version of this problem is extremely difficult, so we begin with a simpler setting. In particular, we generate n independent predicted tuples according to a distribution having optimal entropy. This entropy-based approach is tractable, scalable, and performs well.

Keywords: Entropy; optimization; parimutuel betting; NCAA March Madness; bracket challenges

check for updates

Citation: Brill, R.S.; Wyner, A.J.; Barnett, I.J. Entropy-Based Strategies for Multi-Bracket Pools. *Entropy* **2024**, *26*, 615. https://doi.org/10.3390/e26080615

Academic Editor: Nikolai Leonenko

Received: 5 June 2024
Revised: 19 July 2024
Accepted: 20 July 2024
Published: 23 July 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Parimutuel betting or pool betting involves pooling together all bets of a particular type on a given event, deducting a track take or vigorish, and splitting the pot among all winning bets. Prime examples are horse race betting and the March Madness bracket challenge (which involves predicting the winner of each game in the NCAA Division I men's basketball March Madness tournament). Profitable parimutuel wagering systems have two components: a probability model of the event outcome and a bet allocation strategy. The latter uses the outcome probabilities as inputs to a betting algorithm that determines the amount to wager on each potential outcome. There is a large body of literature on estimating outcome probabilities for pool betting events. For instance, we provide an overview of estimating outcome probabilities for horse races and college basketball matchups in Appendix B.1. There is also a large body of literature on developing optimal wagering strategies, particularly for betting on horse race outcomes. Notably, assuming the outcome probabilities are known, Isaacs [1] and Kelly [2] derive the amount to wager on each horse so as to maximize expected profit and expected log wealth, respectively. Rosner [3] derives a wagering strategy for a risk-averse decision maker, and Willis [4] and Hausch et al. [5] derive other wagering strategies. On the other hand, there has been very limited work, and no literature to our knowledge, on deriving optimal strategies for generating multiple predicted March Madness brackets. Existing work focuses on generating a single predicted bracket (see Appendix B.2 for details).

Existing wagering strategies for pools that involve betting on the outcome of a single event (e.g., the winner of a horse race) have been successful. For instance, Benter [6]

reported that his horse race gambling syndicate made *significant* profits during its five-year gambling operation. However, many betting pools in the real world involve betting not just on a single event but on a *tuple of events*. For example, the pick six bet in horse racing involves predicting each winner of six horse races. Also, the March Madness bracket challenge involves predicting the winner of each game in the NCAA Division I men's basketball tournament. Another compelling example to the pure mathematician is predicting each of the bits in a randomly drawn bitstring. In each of these three prediction contests, the goal is to predict as best as possible a tuple of events, which we call a *bracket*. We suppose it is permissible to generate multiple predicted brackets, so we call these contests *multi-bracket pools*. In developing wagering strategies for multi-bracket pools, the literature on estimating outcome probabilities for each event in the bracket still applies. However, given these probabilities, the wagering strategy the literature developed for betting on single events does not extend to general multi-bracket pools. Although these methods work well in low-dimensional examples such as betting on the winner of a horse race, they are intractable for general multi-bracket pools having larger dimension (e.g., March Madness bracket challenges; see Appendix C for details).

Hence, we pose the *multi-brackets problem*. Suppose we wish to predict a bracket (a tuple of events) and suppose we know the "true" probabilities of each potential outcome of each event. Then, what is the best way to tractably generate a set of n predicted brackets? More concretely, how can we construct a set of n brackets that maximize an objective function such as expected score, win probability, or expected profit? The most general version of the multi-brackets problem, which finds the optimal set of n brackets across all such possible sets, is extremely difficult. To make the problem tractable, possible, and/or able to be visualized, depending on the particular specification of the multi-bracket pool, we make simplifying assumptions. First, we assume we (and optionally a field of opponents) predict i.i.d. brackets generated according to a bracket distribution. The task becomes to find the optimal generating bracket distribution. For higher-dimensional examples (e.g., March Madness bracket challenges), we make another simplifying assumption, optimizing over an intelligently chosen low-dimensional subspace of generating bracket distributions. In particular, we optimize over brackets of varying levels of entropy. We find that this entropy-based approach is sufficient to generate well-performing sets of bracket predictions. We also learn the following high-level lessons from this strategy: we should increase the entropy of our bracket predictions as n increases and as our opponents increase entropy.

The remainder of this paper is organized as follows. In Section 2, we formally introduce the multi-brackets problem. Then, in Section 3, we propose an entropy-based solution to what we consider a canonical example of a multi-bracket pool: guessing a randomly drawn bitstring. Using this example to aid our understanding of multi-bracket pools, in Section 4, we connect the multi-brackets problem to Information Theory, particularly via the Asymptotic Equipartition Property. Then, in Section 5, we propose entropy-based solutions to real-world examples of multi-bracket pools, including the pick six bet in horse racing in Section 5.1 and March Madness bracket challenges in Section 5.2. We conclude in Section 6.

2. The Multi-Brackets Problem

In this section, we formally introduce the multi-brackets problem. The goal of a multi-bracket pool is to predict a tuple of m outcomes $\tau = (\tau_1, ..., \tau_m)$, which we call the "true" observed reference *bracket*. We judge how "close" a bracket prediction $x = (x_1, ..., x_m)$ is to τ by a *bracket scoring function* $f(x, \tau)$. One natural form for the scoring function is

$$f(x, \tau) = \sum_{i=1}^{m} w_i \cdot \mathbb{1}\{x_i = \tau_i\}, \tag{1}$$

which is the number of outcomes predicted correctly weighted by $\{w_i\}_{i=1}^{m}$. Another is

$$f(x, \tau) = \mathbb{1}\{x = \tau\}, \tag{2}$$

which is one if and only if the predicted bracket is exactly correct. The contestants who submit the highest-scoring brackets win the pool.

The multi-brackets problem asks the general question: if we could submit n brackets to the pool, how should we choose which brackets to submit? This question takes on various forms depending on the information available to us and the structure of a particular multi-bracket pool. In the absence of information about opponents' predicted brackets, how should we craft our submitted set $\mathscr{B}_n$ of n bracket predictions in order to maximize the expected maximum score? Formally, we solve

$$\mathscr{B}_n^* := \underset{\{\mathscr{B}_n \subset \mathscr{X} : |B_n| = n\}}{\arg\max} \mathbb{E}_\tau \left[\max_{x \in \mathscr{B}_n} f(x, \tau) \right]. \tag{3}$$

Or, assuming a field of opponents submits a set $\mathscr{O}_k$ of k bracket predictions to the pool according to some strategy, how should we craft our submitted set $\mathscr{B}_n$ of n brackets in order to maximize our probability of having the best bracket? Formally, we solve

$$\mathscr{B}_n^* := \underset{\{\mathscr{B}_n \subset \mathscr{X} : |B_n| = n\}}{\arg\max} \mathbb{P}_{\tau, \mathscr{O}_k} \left[\max_{x \in \mathscr{B}_n} f(x, \tau) \geq \max_{y \in \mathscr{O}_k} f(y, \tau) \right]. \tag{4}$$

Another version of a multi-bracket pool offers a *carryover* C of initial money in the pot, charges b dollars per submitted bracket, and removes a fraction α from the pot as a track take or vigorish. The total pool of money entered into the pot is thus

$$T = C + b(n + k)(1 - \alpha), \tag{5}$$

which is split among the entrants with the highest-scoring brackets. The question becomes the following: how should we craft our submitted set $\mathscr{B}_n$ of n brackets in order to maximize expected profit? Formally, we solve

$$\mathscr{B}_n^* := \underset{\{\mathscr{B}_n \subset \mathscr{X} : |B_n| = n\}}{\arg\max} T \cdot \mathbb{P}_{\tau, \mathscr{O}_k} \left[\max_{x \in \mathscr{B}_n} f(x, \tau) > \max_{y \in \mathscr{O}_k} f(y, \tau) \right] - b \cdot n. \tag{6}$$

This variant assumes no ties but is easily extended to incorporate ties (see Section 5.1). The optimization problems in Equations (3), (4) and (6) and related variants define the multi-brackets problem.

In upcoming sections, we explore specific examples of the multi-brackets problem. In guessing a randomly drawn bitstring (Section 3) and the March Madness bracket challenge (Section 5.2), we explore the multi-brackets problem via scoring function (1) and objective functions (3) and (4). In pick six betting in horse racing (Section 5.1), we explore the multi-brackets problem via scoring function (2) and objective function (6).

The most general version of the multi-brackets problem, which finds the optimal set of n brackets across all such possible sets, is extremely difficult. To make the problem tractable, possible, and/or able to be visualized, depending on the particular specification of the multi-bracket pool, we make simplifying assumptions. We assume we (and the field of opponents) submit i.i.d. brackets generated from some bracket distribution. As the size of a bracket increases, solving the multi-brackets problem under this assumption quickly becomes intractable, so we optimize over intelligently chosen low-dimensional subspaces of bracket distributions. We find this entropy-based strategy is sufficient to generate well-performing sets of submitted brackets.

3. Canonical Example: Guessing a Randomly Drawn Bitstring

In this section, we delve into what we consider a canonical example of a multi-bracket pool: guessing a randomly drawn bitstring. In this contest, we want to predict the sequence of bits in a reference bitstring, which we assume is generated according to some known probability distribution. We submit n guesses of the reference bitstring with the goal of

being as "close" to it as possible or of being "closer" to it than a field of k opponents' guesses, according to some distance function. With some assumptions on the distribution **P** from which the reference bitstring is generated, the distribution **Q** from which we generate bitstring guesses, and the distribution **R** from which opponents generate bitstring guesses, the expected maximum score and win probability are analytically computable and tractable. By visualizing these formulas, we discern high-level lessons relevant to all multi-bracket pools. To maximize the expected maximum score of a set of n submitted randomly drawn brackets, we should increase the entropy of our submitted brackets as n increases. To maximize the probability that the maximum score of n submitted randomly drawn brackets exceeds that of k opposing brackets, we should increase the entropy of our brackets as our opponents increase entropy.

The objective of this multi-bracket pool is to predict a randomly drawn bitstring, which is to predict a sequence of bits. Here, a bracket is a bitstring consisting of m bits divided into R rounds with m_{rd} bits in each round $\mathsf{rd} \in \{1, ..., R\}$. For concreteness, we let there be $m_{\mathsf{rd}} = 2^{R-\mathsf{rd}}$ bits in each of the $R = 6$ rounds (i.e., 32 bits in round 1, 16 bits in round 2, 8 bits in round 3, ..., 1 bit in round 6, totaling 63 bits), but the analysis in this section holds for other choices of m_{rd} and R. The "true" reference bracket that we are trying to predict is a bitstring $\tau = (\tau_{\mathsf{rd},i} : 1 \le \mathsf{rd} \le R, 1 \le i \le m_{\mathsf{rd}})$. A field of opponents submits k guesses of τ, the brackets $(y^{(1)}, ..., y^{(k)})$, where each bracket is a bitstring $y^{(\ell)} = (y^{(\ell)}_{\mathsf{rd},i} : 1 \le \mathsf{rd} \le R, 1 \le i \le m_{\mathsf{rd}})$. We submit n guesses of τ, the brackets $(x^{(1)}, ..., x^{(n)})$, where each bracket is a bitstring $x^{(j)} = (x^{(j)}_{\mathsf{rd},i} : 1 \le \mathsf{rd} \le R, 1 \le i \le m_{\mathsf{rd}})$. The winning submitted bracket among $\{x^{(j)}\}_{j=1}^{n} \cup \{y^{(\ell)}\}_{\ell=1}^{k}$ is "closest" to the reference bracket τ according to a scoring function $f(x, \tau)$ measuring how "close" x is to τ. Here, we consider

$$f(x, \tau) = \sum_{\mathsf{rd}=1}^{R} \sum_{i=1}^{m_{\mathsf{rd}}} w_{\mathsf{rd},i} \cdot \mathbb{1}\{x_{\mathsf{rd},i} = \tau_{\mathsf{rd},i}\}, \tag{7}$$

which is the weighted number of bits guessed correctly. This scoring function encompasses both the *Hamming score* and *ESPN score*. The Hamming score measures the number of bits guessed correctly, weighing each bit equally ($w_{\mathsf{rd},i} \equiv 1$). The ESPN score weighs each bit by $w_{\mathsf{rd},i} = 10 \cdot 2^{\mathsf{rd}-1}$ so that the maximum accruable score in each round is the same ($10 \cdot 2^{R-1}$).

Suppose the "true" reference bitstring τ is generated according to some known distribution **P** and opponents' bitstrings are generated according to some known distribution **R**. Our task is to submit n predicted bitstrings so as to maximize the expected maximum score

$$\mathbb{E}\left[\max_{j=1,...,n} f(x^{(j)}, \tau)\right] \tag{8}$$

or the probability that we do not lose the bracket challenge

$$\mathbb{P}\left[\max_{j=1,...,n} f(x^{(j)}, \tau) \ge \max_{\ell=1,...,k} f(y^{(\ell)}, \tau)\right]. \tag{9}$$

In particular, we wish to submit n bitstrings generated according to some distribution **Q**, and it is our task to find suitable **Q**. For tractability, we consider the special case that bits are drawn independently with probabilities varying by round. We suppose that each bit $\tau_{\mathsf{rd},i}$ in the reference bitstring is an independently drawn Bernoulli(p_{rd}) coin flip. The parameter $p_{\mathsf{rd}} \in [0.5, 1]$ controls the entropy of the contest: lower values correspond to a higher-entropy (more variable) reference bitstring that is harder to predict. By symmetry, our strategy just needs to vary by round. So, we assume that each of our submitted bits $x^{(j)}_{\mathsf{rd},i}$ is an independently drawn Bernoulli(q_{rd}) coin flip and each of our opponents' submitted bits $y^{(\ell)}_{\mathsf{rd},i}$ is an independently drawn Bernoulli(r_{rd}) coin flip. The parameters q_{rd} and r_{rd} control the entropy of our submitted bitstrings and our opponents' submitted bitstrings, respectively. Our task is to find the optimal strategy or entropy level $(q_{\mathsf{rd}})_{\mathsf{rd}=1}^{R}$. In this

setting, expected maximum score and win probability are analytically computable and tractable (see Appendix E).

We first visualize the case where the entropy of the reference bitstring, our submitted bitstrings, and our opponents' submitted bitstrings do not vary by round: $p \equiv p_{\text{rd}}$, $q \equiv q_{\text{rd}}$, and $r \equiv r_{\text{rd}}$. In Figure 1, we visualize the expected maximum Hamming score of n submitted bitstrings as a function of p, q, and n. We find that we should increase the entropy of our submitted brackets (decrease q) as n increases, transitioning from pure chalk ($q = 1$) for $n = 1$ bracket to the true amount of randomness ($q = p$) for large n. Specifically, for small n, the green line $q = p$ lies below the blue lines (large q), and for large n, the green line lies above all the other lines.

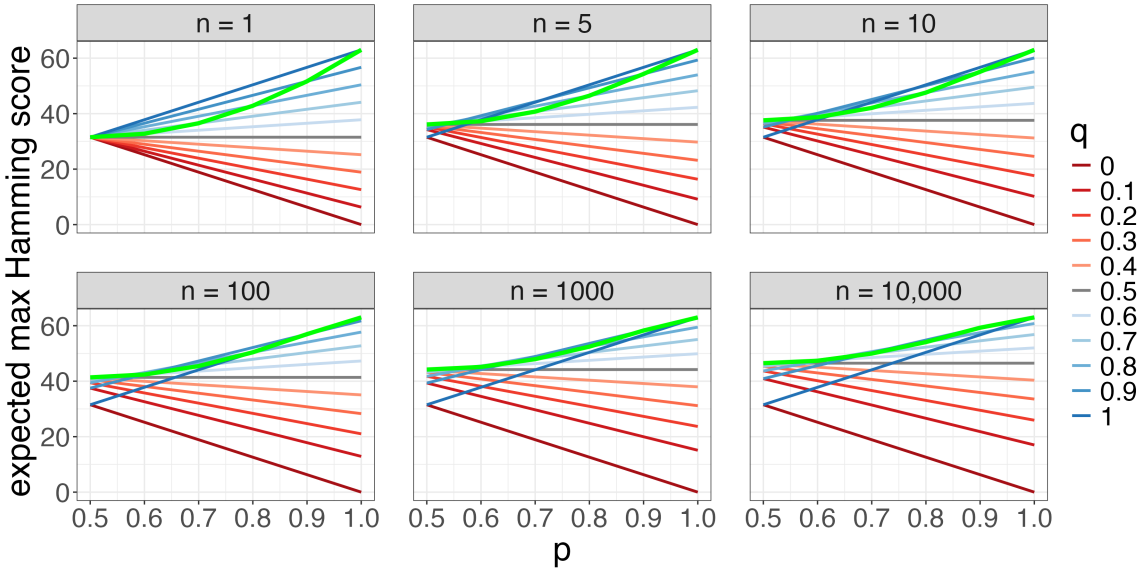


Figure 1. The expected maximum Hamming score (y-axis) of n submitted Bernoulli(q) bitstrings relative to a reference Bernoulli(p) bitstring as a function of p (x-axis), q (color), and n (facet) in the "guessing a randomly drawn bitstring" contest with $p \equiv p_{\text{rd}}$, $q \equiv q_{\text{rd}}$, $r \equiv r_{\text{rd}}$, and $R = 6$ rounds. As n increases, we want to increase the entropy of our submitted brackets.

In Figure 2, we visualize the win probability as a function of q, r, and n for $k = 100$ and $p = 0.75$. The horizontal gray dashed line $q = p = 0.75$ represents that we match the entropy of the reference bitstring, the vertical gray dashed line $r = p = 0.75$ represents that our opponents match the entropy of the reference bitstring, and the diagonal gray dashed line $q = r$ represents that we match our opponents' entropy. We should increase entropy (decrease q) as n increases, visualized by the green region moving downwards as n increases. Further, to maximize win probability, we should increase entropy (decrease q) as our opponents' entropy increases (as r decreases), visualized by the triangular form of the green region. In other words, we should tailor the entropy of our brackets to the entropy of our opponents' brackets. These trends are similar for other values of k and n (see Figure A3 of Appendix E).

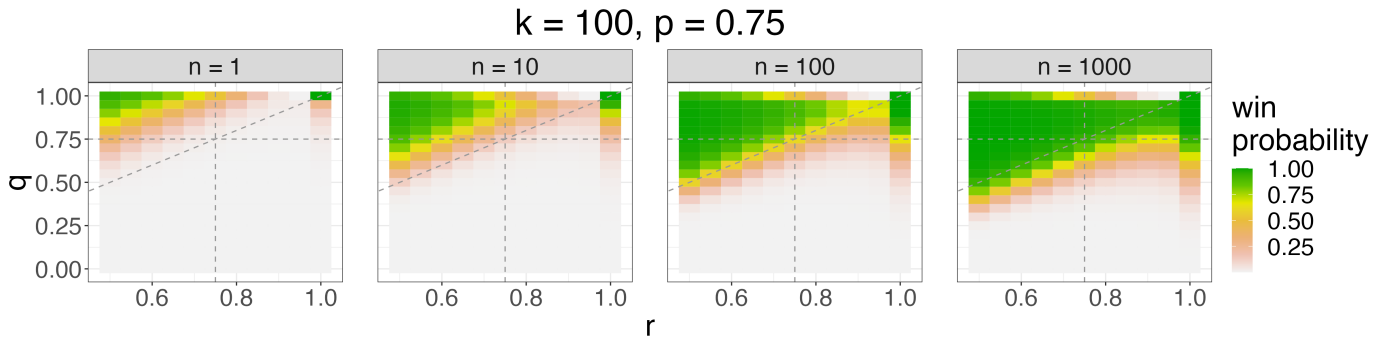


Figure 2. The probability (color) that the maximum Hamming score of n submitted Bernoulli(q) brackets relative to a reference Bernoulli(p) bracket exceeds that of k opposing Bernoulli(r) brackets as a function of q (y-axis), r (x-axis), and n (facet) for $p = 0.75$ and $k = 100$ in the "guessing a randomly drawn bitstring" contest with $p \equiv p_{\text{rd}}$, $q \equiv q_{\text{rd}}$, $r \equiv r_{\text{rd}}$, and $R = 6$ rounds. We should increase entropy as n increases and as our opponents' entropy increases.

These trends generalize to the case where the entropy of each bitstring varies by round (i.e., general q_{rd}, r_{rd}, and p_{rd}). It is difficult to visualize the entire $R = 6$ dimensional space of $p = (p_1, ..., p_6)$, $q = (q_1, ..., q_6)$, and $r = (r_1, ..., r_6)$, so we instead consider a lower-dimensional subspace. Specifically, we visualize a two-dimensional subspace of q parameterized by (q_E, q_L), where q_E denotes q in early rounds and q_L denotes q in later rounds. For example, $q_E = q_1 = q_2 = q_3$ and $q_L = q_4 = q_5 = q_6$ is one of the five possible partitions of (q_E, q_L). We similarly visualize a two-dimensional subspace of r parameterized by (r_E, r_L). Finally, we let the reference bitstring have a constant entropy across each round, $p_{\text{rd}} \equiv p$.

In Figure 3, we visualize the expected maximum ESPN score of n bitstrings as a function of q_E, q_L, and n for $p = 0.75$. The three columns display the results for $n = 1$, $n = 10$, and $n = 100$, respectively. The five rows display the results for the five partitions of (q_E, q_L). For instance, the first row shows one partition $q_E = q_1$ and $q_L = q_2 = q_3 = q_4 = q_5 = q_6$. As n increases, the expected maximum ESPN score increases. We visualize this as the lines moving upwards as we move right across the grid of plots. As E increases (i.e., as q_E encompasses a larger number of early rounds), the impactfulness of the late round strategy q_L decreases. We visualize this as the lines become more clumped together as we move down the grid of plots in Figure 3. For $n = 1$, the best strategy is pure chalk ($q_E = 1$, $q_L = 1$), and as n increases, the optimal values of q_E and q_L decrease. In other words, as before, we want to increase the entropy of our submitted brackets as n increases. We visualize this as the circle (i.e., the best strategy in each plot) moving leftward and having a more reddish color as n increases.

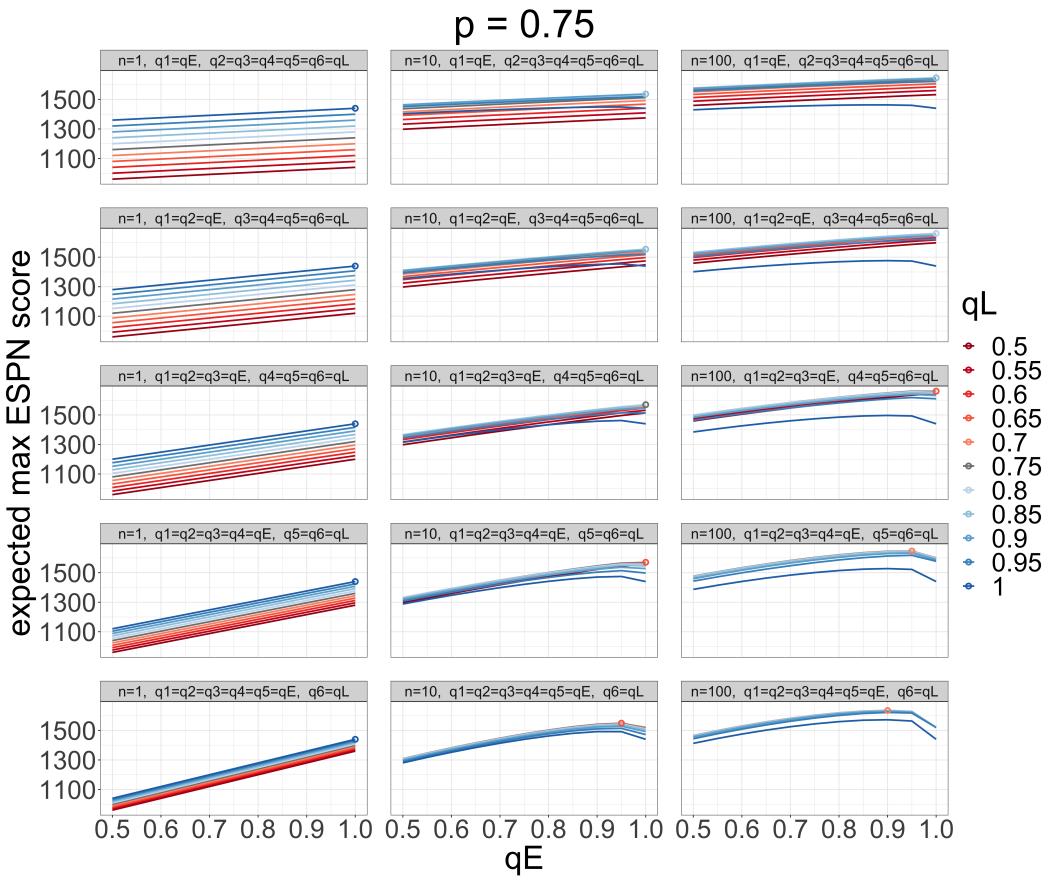
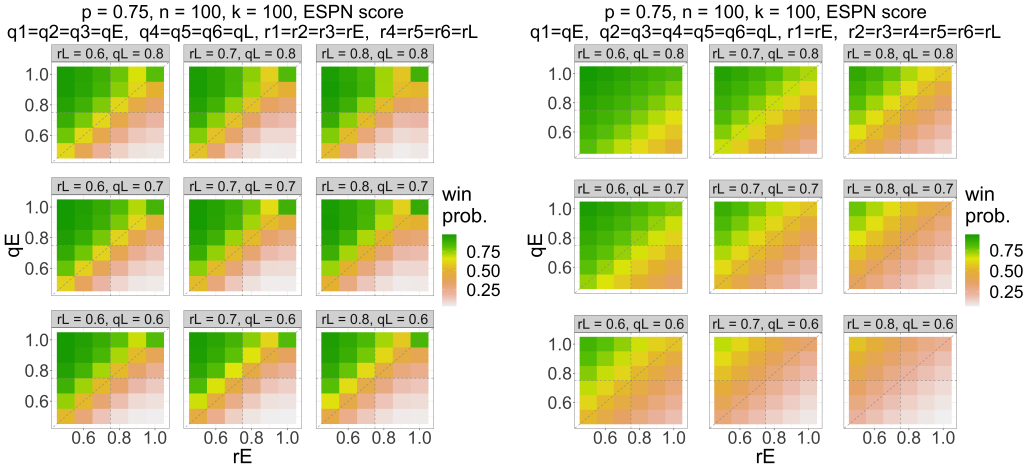


Figure 3. The expected maximum ESPN score (y-axis) of n submitted bitstrings, with Bernoulli(q_E) bits in early rounds and Bernoulli(q_L) bits in later rounds, relative to a reference Bernoulli(p) bitstring as a function of q_E (x-axis), q_L (color), n (columns), and the partition (q_E, q_L) (rows) in the "guessing a randomly drawn bitstring" contest with $R = 6$ rounds and $p = 0.75$. The circles indicate the best strategy in each setting. As n increases, we want to increase the entropy of our bracket predictions in both the early and late rounds.

In Figure 4, we visualize the win probability as a function of q_E, q_L, r_E, and r_L, for $n = k = 100$, $p = 0.75$, and ESPN score. Figure 4a uses the partition where the first three rounds are the early rounds (e.g., $q_E = q_1 = q_2 = q_3$ and $r_E = r_1 = r_2 = r_3$). In this scenario, early round strategy q_E and r_E are much more impactful than late round strategy q_L and r_L. We visualize this as each subplot looking the same. The green triangle within each subplot illustrates that we should increase early round entropy (decrease q_E) as our opponents' early round entropy increases (i.e., as r_E decreases). Figure 4b uses the partition where just the first round is an early round (e.g., $q_E = q_1$ and $r_E = r_1$). In this scenario, both early round strategy q_E and r_E and late round strategy q_L and r_L are impactful. The green triangle appears again in each subplot, illustrating that we should increase early round entropy as our opponents' early round entropy increases. But the green triangle grows as r_L decreases, indicating that we should increase late round entropy (decrease q_E) as our opponent's entropy increases.



(**a**) (**b**)

Figure 4. The probability (color) that the maximum ESPN score of n bitstrings, with Bernoulli(q_E) bits in early rounds and Bernoulli(q_L) bits in later rounds, relative to a reference Bernoulli(p) bitstring exceeds that of k opposing bitstrings, with Bernoulli(r_E) bits in early rounds and Bernoulli(r_L) bits in later rounds, as a function of q_E (y-axis), r_E (x-axis), q_L (rows), and r_L (columns) for $p = 0.75$, $k = 100$, and $n = 100$ in the "guessing a randomly drawn bitstring" contest with $R = 6$ rounds. (**a**) uses the partition where the first three rounds are the early rounds (e.g., $q_E = q_1 = q_2 = q_3$ and $r_E = r_1 = r_2 = r_3$) and (**b**) uses the partition where just the first round is an early round (e.g., $q_E = q_1$ and $r_E = r_1$). We should still increase the entropy of our bracket predictions as our opponents increase entropy.

4. An Information Theoretic View of the Multi-Brackets Problem

The multi-brackets problem is intimately connected to Information Theory. Viewing the multi-brackets problem under an information theoretic lens provides a deeper understanding of the problem and elucidates why certain entropy-based strategies work. In particular, the Asymptotic Equipartition Property from Information Theory helps us understand why it makes sense to increase entropy as the number of brackets increases and as our opponents' entropy increases. In this section, we give an intuitive explanation of the Equipartition Property and discuss implications, relegating the formal mathematical details to Appendix F.

To begin, we partition the set of all brackets $\mathcal{X}$ into three subsets,

$$\begin{cases} \text{low entropy "chalky" brackets} & \mathscr{C} \subset \mathcal{X}, \\ \text{"typical" brackets} & \mathscr{T} \subset \mathcal{X}, \\ \text{high entropy "rare" brackets} & \mathscr{R} \subset \mathcal{X}. \end{cases} \tag{10}$$

We visualize this partition of $\mathcal{X}$ under three lenses in Figure 5.

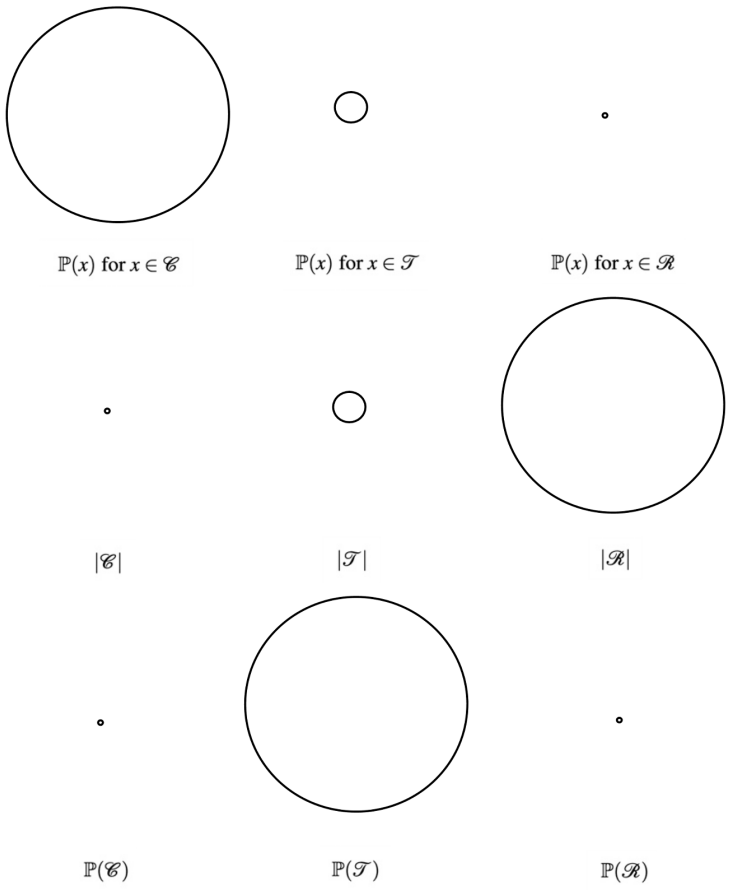


Figure 5. Note that these figures are not drawn to scale. First line: the probability mass of an individual low-entropy (chalky) bracket is much larger than the probability mass of an individual typical bracket, which is much larger than the probability mass of an individual high-entropy (rare) bracket. Second line: there are exponentially more rare brackets than typical brackets, and there are exponentially more typical brackets than chalky brackets. Third line: the typical brackets occupy most of the probability mass on aggregate.

First, the probability mass of an individual low-entropy or chalky bracket is much larger than the probability mass of an individual typical bracket, which is much larger than the probability mass of an individual high-entropy or rare bracket. In symbols, if $x_1 \in \mathscr{C}, x_2 \in \mathscr{T}$, and $x_3 \in \mathscr{R}$, then $\mathbb{P}(x_1) >> \mathbb{P}(x_2) >> \mathbb{P}(x_3)$. "Rare" is a good name for high-entropy brackets because they are highly unlikely. "Chalk", a term from sports betting, is a good name for low-entropy brackets because it refers to betting on the heavy favorite (i.e., the outcome with the highest individual likelihood). Most of the individual forecasts within a low-entropy bracket must consist of the most probable outcomes. For example, in the "guessing a bitstring" contest, assuming the reference bitstring consists of independent Bernoulli(p) bits where $p > 0.5$, low-entropy brackets are bitstrings consisting mostly of ones. In real-world examples of multi-bracket pools, people are drawn to these low-entropy chalky brackets because they have high individual likelihoods.

Second, there are exponentially more rare brackets than typical brackets, and there are exponentially more typical brackets than chalky brackets. In symbols, $|\mathscr{R}| >> |\mathscr{T}| >> |\mathscr{C}|$. In the "guessing a bitstring" contest with $p > 0.5$, the overwhelming majority of possible brackets are high-entropy brackets having too many zeros, and very few possible brackets are low-entropy brackets consisting almost entirely of ones. Typical brackets tow the line, having the "right" amount of ones. March Madness is analogous: the overwhelming majority of possible brackets are rare brackets with too may upsets (e.g., a seed above 8

winning the tournament) and relatively few possible brackets are chalky brackets with few upsets (there are only so many distinct brackets with favorites winning nearly all the games). Typical brackets tow the line, having the "right" number of upsets.

Lastly, the typical set of brackets contains most of the probability mass. In symbols, $\mathbb{P}(\mathscr{T}) >> \mathbb{P}(\mathscr{C})$ and $\mathbb{P}(\mathscr{T}) >> \mathbb{P}(\mathscr{R})$. This is a consequence of the previous two inequalities. Although $|\mathscr{R}|$ is massive, $\mathbb{P}(x)$ for $x \in \mathscr{R}$ is so small that $\mathbb{P}(\mathscr{R})$ is small. Also, although $\mathbb{P}(x)$ for $x \in \mathscr{C}$ is relatively large, $|\mathscr{C}|$ is so small that $\mathbb{P}(\mathscr{C})$ is small. Hence, the remainder of the probability mass, $\mathbb{P}(\mathscr{T})$, is large. "Typical" is thus a good name for brackets whose entropy is not too high or too low because a randomly drawn bracket typically has this "right" amount of entropy. For example, the observed March Madness tournament is almost always a typical bracket featuring a typical number of upsets.

Drilled down to its essence, the Equipartition Property tells us that, as the number of forecasts m within each bracket grows, the probability mass of the set of brackets becomes increasingly more concentrated in an exponentially small set, the "typical set." See Appendix F for a more formal treatment of the Equipartition Property.

This information theoretic view of the multi-brackets problem sets up a tradeoff between chalky and typical brackets. Typical brackets have the "right" entropy but consist of less likely individual outcomes, whereas chalky low-entropy brackets have the "wrong" entropy but consist of more likely individual outcomes. The former excels when n is large, the latter excels when n is small, and for moderate n we interpolate between these two regimes; so, we should increase the entropy of our set of predicted brackets as the number of brackets n increases. We justify this below using the Equipartition Property.

As the typical set contains most of the probability mass, the reference bracket is highly likely to be a typical bracket. So, when n is large, we should generate typical brackets as guesses since it is likely that at least one of these guesses is "close" to the reference bracket. When n is small, generating typical brackets as guesses does not produce as high an expected maximum score as chalky brackets. To understand, recall that a bracket consists of m individual forecasts. A single randomly drawn typical bracket has the same entropy as the reference bracket but is not likely to correctly predict each individual forecast. For instance, in our "guessing a bitstring" example, a single randomly drawn bitstring has, on average, a similar number of ones as the reference bitstring, but not the "*right*" ones in the "right" locations. A chalky bracket, on the other hand, predicts highly likely outcomes in most of the individual forecasts. The chalkiest bracket, which predicts the most likely outcome in each individual forecast, matches the reference bracket for each forecast in which the reference bracket realizes its most likely outcome. This, on average, yields more matches than that of a typical bracket because more forecasts realize their most likely outcome than any other single outcome. For instance, in our "guessing a bitstring" example, a chalky bracket consists mostly of ones (assuming $p > 0.5$) and so correctly guesses the locations of ones in the reference bitstring. This is better, on average, than guessing a typical bracket, which, on average, has the "right" number of ones but in the wrong locations.

5. Real-World Examples

Now, we discuss real world examples of multi-bracket pools: pick six betting in horse racing and March Madness bracket challenges. Both contests involve predicting a tuple of outcomes. An individual pick six bet (ticket) involves predicting each winner of six horse races, and an individual March Madness bet (bracket) involves predicting the winner of each game in the NCAA Division I Men's Basketball March Madness tournament. In both contests, it is allowed, but not necessarily commonplace (outside of horse racing betting syndicates), to submit many tickets or brackets. We demonstrate that the entropy-based strategies introduced in the previous sections are particularly well-suited for these problems. In particular, optimizing over strategies of varying levels of entropy is tractable and yields well-performing solutions.

5.1. Pick Six Horse Race Betting

Horse race betting is replete with examples of multi-bracket pools. A prime example is the *pick six* bet, which involves correctly picking the winner of six horse races. Similar pick three, pick four, and pick five bets, which involve correctly picking the winner of three, four, or five horse races, respectively, also exist. Due to the immense difficulty of picking six consecutive horse race winners coupled with a large number of bettors in these pools, payoffs for successful pick six bets can be massive (e.g., in the millions of dollars). In this section, we apply our entropy-based strategies to pick six betting, demonstrating the massive profit potential of these bets.

To begin, let $s \in \{3, 4, 5, 6\}$ denote the number of races comprising the pick-s bet (for the pick three, four, five, and six contests, respectively). Suppose, for simplicity, that one pick-s ticket, consisting of s predicted horse race winners, costs USD 1 each (typically a pick-s bet costs USD 1 or USD 2). Indexing each race by $j = 1, ..., s$, suppose there are m_j horses in race j, and let $\mathbf{m} = (m_1, ..., m_s)$. There is a fixed carryover C, an amount of money leftover from previous betting pools in which no one won, that is added to the total prize pool for the pick-s contest. As assumed throughout this paper, assume the "true" win probability $\mathbf{P}_{ij}$ that horse i wins race j is known for each i and j. As our operating example in this section, we set $\mathbf{P}$ to be the win probabilities implied by the Vegas odds from the pick six contest from Belmont Park on 21 May 2023 (https://entries.horseracingnation.com/entries-results/belmont-park/2023-05-21, accessed on 1 June 2023), which we visualize in Figure 6.

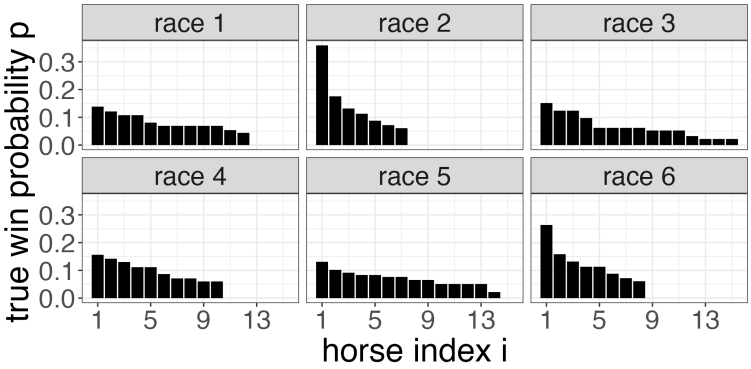


Figure 6. The "true" win probability $\mathbf{P}_{ij}$ (y-axis) that horse i (x-axis) wins race j (facet) for each i, j. These probabilities are implied by the Vegas odds for the pick six contest at Belmont Park on 21 May 2023.

Suppose the public purchases k entries according to some strategy. In particular, we assume the public submits k independent tickets according to $\mathbf{R}$, where $\mathbf{R}_{ij}$ is the probability an opponent selects horse i to win race j. We purchase n entries according to strategy $\mathbf{Q}$. Specifically, we submit n independent tickets according to $\mathbf{Q}$, where $\mathbf{Q}_{ij}$ is the probability we select horse i to win race j. The total prize money is thus

$$T = C + (n + k)(1 - \alpha), \tag{11}$$

where α is the track take (vigorish). Let W be our number of winning tickets and let W_{opp} be our opponents' number of winning tickets. Under our model, both W and W_{opp} are random variables. Formally, denote the "true" observed s winning horses by $\tau = (\tau_1, ..., \tau_s)$, our n tickets by $(x^{(1)}, ..., x^{(n)})$, where each $x^{(\ell)} = (x_1^{(\ell)}, ..., x_s^{(\ell)})$, and the publics' k tickets by $(y^{(1)}, ..., y^{(k)})$, where each $y^{(\ell)} = (y_1^{(\ell)}, ..., y_s^{(\ell)})$. Then,

$$W = \sum_{\ell=1}^{n} \mathbb{1}\{x^{(\ell)} = \tau\} = \sum_{\ell=1}^{n} \mathbb{1}\{x_1^{(\ell)} = \tau_1, ..., x_s^{(\ell)} = \tau_s\} \tag{12}$$

and

$$W_{\text{opp}} = \sum_{\ell=1}^{k} \mathbb{1}\{y^{(\ell)} = \tau\} = \sum_{\ell=1}^{k} \mathbb{1}\{y_1^{(\ell)} = \tau_1, ..., y_s^{(\ell)} = \tau_s\}. \tag{13}$$

Then, the amount we profit is also a random variable,

$$\text{Profit} = \left(\frac{W}{W + W_{\text{opp}}}\right) T - n, \tag{14}$$

where we treat $\frac{0}{0}$ to be 0 (i.e., if both $W = 0$ and $W_{\text{opp}} = 0$, the fraction $W/(W + W_{\text{opp}})$ is 0). Here, the randomness is over $\tau \sim \mathbf{P}$, $x \sim \mathbf{Q}$, and $y \sim \mathbf{R}$.

Our task is to solve for the optimal investment strategy $\mathbf{Q}$ given all the other variables n, k, $\mathbf{P}$, $\mathbf{R}$, C, and α. Formally, we wish to maximize expected profit,

$$\mathbb{E}[\text{Profit}] = -n + T \cdot \mathbb{E}\left(\frac{W}{W + W_{\text{opp}}}\right). \tag{15}$$

In Appendix G, we compute a tractable lower bound for the expected profit.

We are unable to analytically optimize the expected profit to find an optimal strategy $\mathbf{Q}^*$ given the other variables, and we are unable to search over the entire high-dimensional $\mathbf{Q}$-space for an optimal strategy. Instead, we apply the entropy-based strategies described in the previous sections. The idea is to search over a subspace of $\mathbf{Q}$ that explores strategies of varying entropies, finding the optimal entropy given the other variables. To generate n pick six tickets at varying levels of entropy, we let $\mathbf{Q} = \mathbf{Q}(\lambda, \phi)$ vary according to parameters λ and ϕ that control the entropy. Assuming without loss of generality that in each race j the "true" win probabilities are sorted in decreasing order, $\mathbf{P}_{1j} \geq \mathbf{P}_{2j} \geq ... \geq \mathbf{P}_{m_j j}$, we define $\mathbf{Q}(\lambda, \phi)$ for $\lambda > 0$ and $\phi \in [0, 1]$ by

$$\widetilde{\mathbf{Q}}_{ij}(\lambda, \phi) = \begin{cases} \left(\mathbf{P}_{ij} / \mathbf{P}_{round(\phi \cdot m_j), j}\right)^{\lambda} & \text{if } \lambda < 1, \\ \mathbf{P}_{ij}\left(\lambda \cdot \mathbb{1}\{i \leq round(\phi \cdot m_j)\} + \frac{1}{\lambda} \cdot \mathbb{1}\{i \leq round(\phi \cdot m_j)\}\right) & \text{if } \lambda \geq 1, \end{cases} \tag{16}$$

$$\mathbf{Q}_{ij}(\lambda, \phi) = \widetilde{\mathbf{Q}}_{ij}(\lambda, \phi) / \sum_{i=1}^{m_j} \widetilde{\mathbf{Q}}_{ij}(\lambda, \phi), \tag{17}$$

recalling that there are m_j horses in race j. We visualize these probabilities for race $j = 6$ in Figure 7. For fixed ϕ, smaller values of λ push the distribution $\mathbf{Q}_{*j}$ closer towards a uniform distribution, increasing its entropy. Conversely, increasing λ lowers its entropy. In lowering its entropy, we shift the probability from some horses onto other horses in a way that makes the distribution less uniform. The parameter ϕ controls the number of horses to which we transfer probability as λ increases. For instance, there are $m_j = 8$ horses in race $j = 6$, so when $\phi = 3/8$, we transfer successively more probability to the top $3 = round(\phi \cdot m_j)$ horses as λ increases.

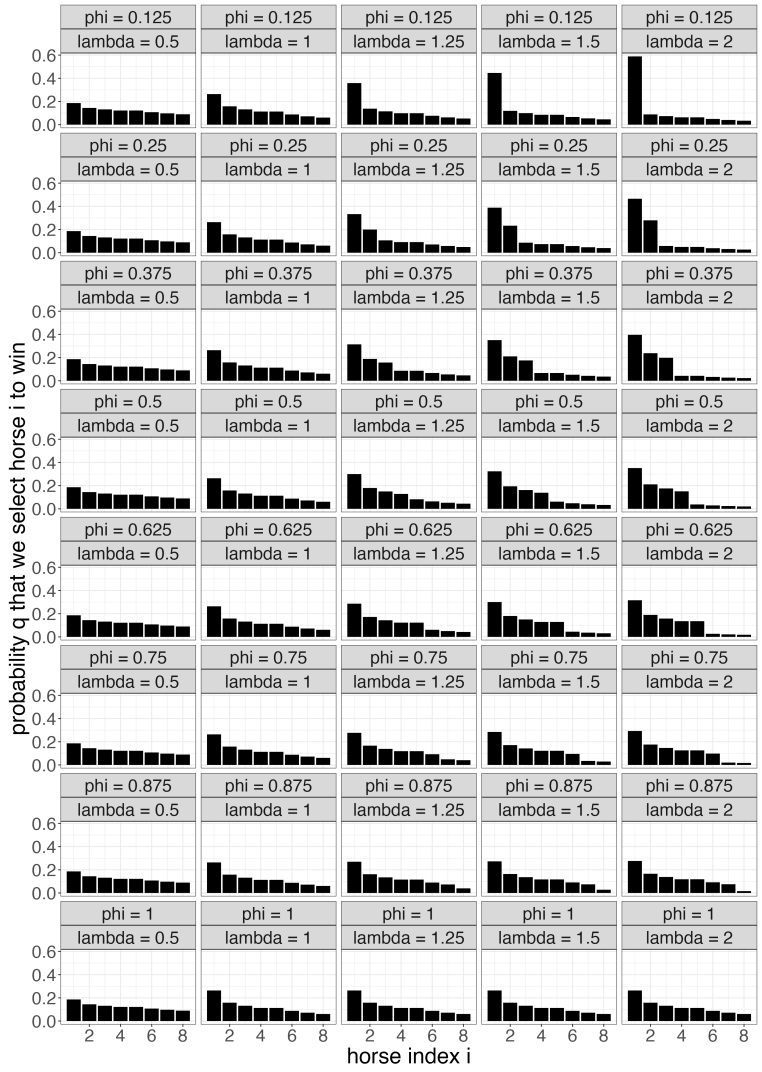


Figure 7. The probability $\mathbf{Q}_{ij} = \mathbf{Q}_{ij}(\lambda, \phi)$ (y-axis) that we select horse i (x-axis) to win race $j = 6$ for various values of λ (column) and ϕ (row). For fixed ϕ, entropy increases as λ decreases. For fixed λ, the probabilities of successively fewer horses are upweighted as ϕ decreases.

Further, we assume we play against opponents who generate brackets according to the strategy $\mathbf{R}_{ij}(\lambda_{opp}) = \mathbf{P}(\lambda = \lambda_{opp}, \phi = 1/8)$. In other words, low-entropy opponents bet mostly on the one or two favorite horses (depending on m_j), high-entropy opponents are close to a uniform distribution, and moderate-entropy opponents lie somewhere in the middle. The exact specification of the opponents' distribution is not important, as we use it to illustrate a general point. In future work, one can try to model the distribution of the public's ticket submissions to obtain more precise results.

In Figure 8, we visualize the expected profit for a pick six horse racing betting pool in which we submit n tickets according to strategy $\mathbf{Q}(\lambda, \phi)$ against a field of k = 25,000 opponents who use strategy $\mathbf{R}(\lambda_{opp})$, assuming a track take of $\alpha = 0.05$ and carryover C = 500,000, as a function of λ_{opp} and n. Given these variables, we use the strategy (λ, ϕ) that maximizes expected profit over a grid of values. We see that the entropy of the optimal strategy increases as n increases (i.e., λ decreases and ϕ increases as n increases). Further, we see that submitting many brackets at a smart entropy level is hugely profitable. This holds true particularly when the carryover is large enough, which occurs fairly regularly.

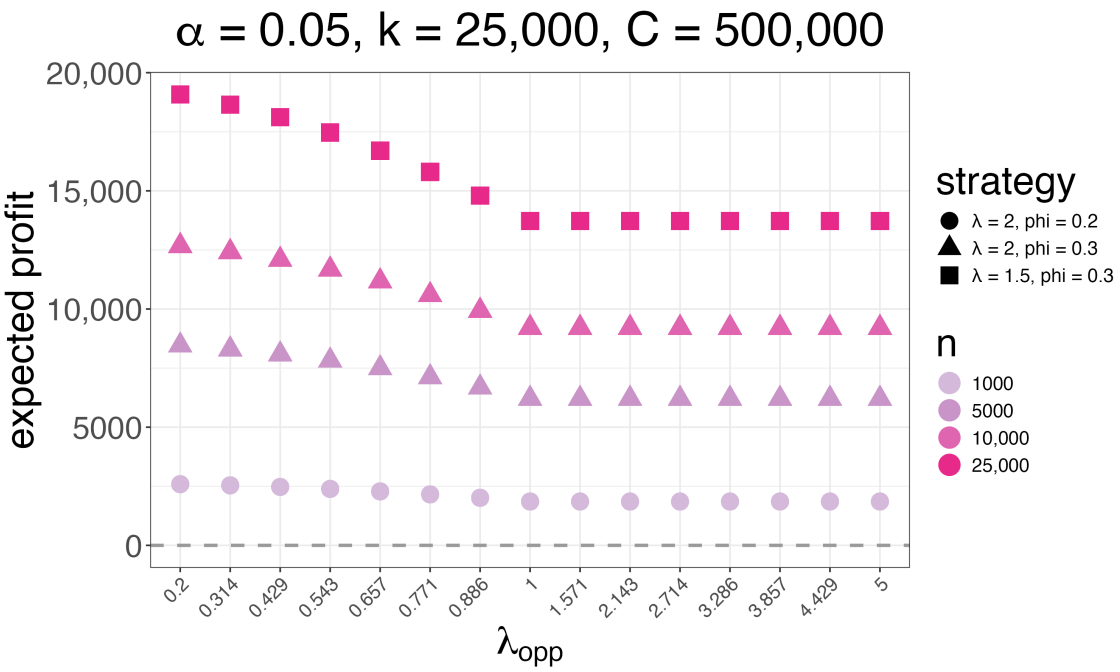


Figure 8. A lower bound of our expected profit (y-axis) for a pick six horse racing betting pool in which we submit n tickets according to strategy $\mathbf{Q}(\lambda, \phi)$ against a field of k = 25,000 opponents who use strategy $\mathbf{R}(\lambda_{opp})$, assuming a track take of $\alpha = 0.05$ and carryover C = 500,000, as a function of λ_{opp} (x-axis) and n (color). Given these variables, we use the strategy (λ, ϕ) that maximizes expected profit over a grid of values.

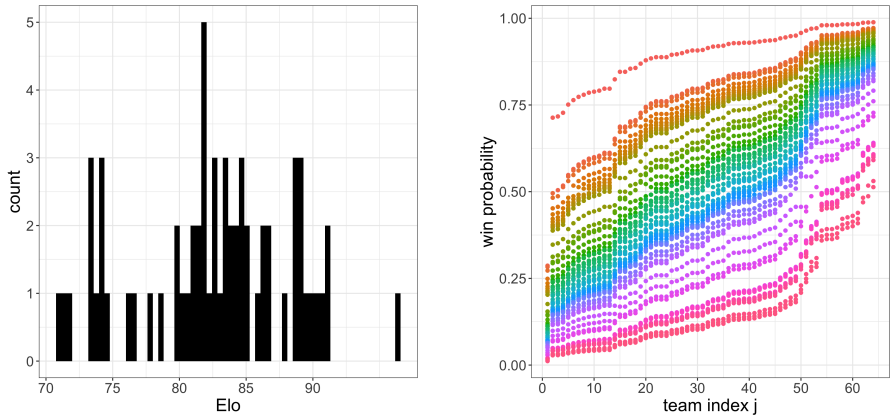
5.2. March Madness Bracket Challenge

March Madness bracket challenges are prime examples of multi-bracket pools. In a bracket challenge, contestants submit an entire *bracket*, or a complete specification of the game winners of each of the games in the NCAA Division I Men's Basketball March Madness tournament. The winning bracket is closest to the observed NCAA tournament according to some metric. Popular March Madness bracket challenges from ESPN, BetMGM, and DraftKings, for instance, offer large cash prizes—BetMGM offered USD 10 million to a perfect bracket or USD 100,000 to the closest bracket, DraftKings sent USD 60,000 in cash prizes spread across the best 5096 brackets last year, and ESPN offered USD 100,000 to the winner of a lottery among the entrants who scored the most points in each round of the tournament (https://www.thelines.com/best-march-madness-bracket-contests/, accessed on 1 June 2023). To illustrate the difficulty of perfectly guessing the observed NCAA tournament, Warren Buffett famously offered USD 1 billion to anyone who filled out a flawless bracket (https://bleacherreport.com/articles/1931210-warren-buffet-will-pay-1-billion-to-fan-with-perfect-march-madness-bracket, accessed on 1 June 2023). In this section, we apply our entropy-based strategies to March Madness bracket challenges, demonstrating the impressive efficacy of this strategy.

To begin, we denote the set of all brackets by $\mathcal{X}$, which consists of $N = 2^{63} = 2^{2^6-1}$ brackets since there are 63 games through six rounds in the NCAA tournament (excluding the four-game play-in tournament). We define an atomic probability measure $\mathbb{P}$ on $\mathcal{X}$, where $\mathbb{P}(x)$ is the probability that bracket $x \in \mathcal{X}$ is the "true" observed NCAA tournament, as follows. Given that match $m \in \{1, ..., 63\}$ involves teams i and j, we model the outcome of this match by $i \cdot b_m + j \cdot (1 - b_m)$, where $b_m \overset{ind}{\sim} \text{Bernoulli}(\mathbf{P}_{ij})$. In other words, with probability $\mathbf{P}_{ij}$, team i wins the match; else, team j wins the match. Prior to the first round (games 1 through 32), the first 32 matchups are set. Given these matchups, the 32 winning teams in round one are determined by Bernoulli coin flips according to $\mathbf{P}$. These 32 winning teams from round one then uniquely determine the 16 matchups for the second round

of the tournament. Given these matchups, the 16 winning teams in round two are also determined by Bernoulli coin flips according to $\mathbf{P}$. These winners then uniquely determine the matchups for round three. This process continues until the end of round six, when one winning team remains.

In this work, we assume we know the "true" win probabilities $\mathbf{P}$. As our operating example in this section, we set $\mathbf{P}$ to be the win probabilities implied by FiveThirtyEight's Elo ratings from the 2021 March Madness tournament (https://projects.fivethirtyeight.com/2021-march-madness-predictions/, accessed on 5 June 2024). We scrape FiveThirtyEight's pre-round-one 2021 Elo ratings $\{\beta_i\}_{i=1}^{64}$ and index the teams by $i \in \{1, ..., 64\}$ in decreasing order of Elo rating (e.g., the best team Gonzaga is 1 and the worst team Texas Southern is 64). Then, we define $\mathbf{P}$ by $\mathbf{P}_{ij} = 1/(1 + 10^{-(\beta_i - \beta_j) * 30.464/400})$. In Figure 9a, we visualize $\{\beta_i\}_{i=1}^{64}$. The Elo ratings range from 71.1 (Texas Southern) to 96.5 (Gonzaga), who is rated particularly highly. In Figure 9b, we visualize $\mathbf{P}$ via the functions $j \mapsto \mathbf{P}_{ij}$ for each team i. For instance, Gonzaga's win probability function is the uppermost orange line, which is considerably higher than the other teams' lines. See Appendix D for a discussion of the robustness of our results to this choice of $\mathbf{P}$.



(**a**) Histogram of Elo ratings (**b**) $j \mapsto \mathbf{P}_{ij}$ for each i

Figure 9. (**a**): A histogram of FiveThirtyEight's pre-round-one Elo ratings for the 2021 March Madness tournament. (**b**): The function $j \mapsto \mathbf{P}_{ij}$ for each team i (color) implied by these Elo ratings.

Suppose a field of opponents submits k brackets $(y^{(1)}, ..., y^{(k)}) \subset \mathcal{X}$ to the bracket challenge according to some strategy $\mathbf{R}$. In particular, we assume the public submits k independent brackets according to $\mathbf{R}$, where $\mathbf{R}_{ij}$ is the probability an opponent selects team i to beat team j in the event that they play. We submit n brackets $(x^{(1)}, ..., x^{(n)}) \subset \mathcal{X}$ to the bracket challenge according to strategy $\mathbf{Q}$. Specifically, we submit n independent brackets according to $\mathbf{Q}$, where $\mathbf{Q}_{ij}$ is the probability we select team i to beat team j in the event that they play. The goal is to become as "close" to the "true" reference bracket $\tau \in \mathcal{X}$, or the observed NCAA tournament, as possible according to a bracket scoring function. The most common such scoring function in these bracket challenges is what we call *ESPN score*, which credits $10 \cdot 2^{\mathsf{rd}-1}$ points to correctly predicting the winner of a match in round $\mathsf{rd} \in \{1, ..., 6\}$. Since there are $2^{6-\mathsf{rd}}$ matches in each round rd, ESPN score ensures that the maximum accruable points in each round is the same (320). Formally, our task is to submit n brackets so as to maximize the probability we do not lose the bracket challenge,

$$\mathbb{P}\left[\max_{j=1,...,n} f(x^{(j)}, \tau) \geq \max_{\ell=1,...,k} f(y^{(\ell)}, \tau)\right]. \tag{18}$$

Alternatively, in the absence of information about our opponents, our task is to submit n brackets so as to maximize expected maximum score,

$$\mathbb{E}\left[\max_{j=1,\ldots,n} f(x^{(j)}, \tau)\right]. \tag{19}$$

Under this model, it is intractable to explicitly *evaluate* these formulas for expected maximum score or win probability for general **P**, **Q**, and **R**, even when we independently draw brackets from these distributions. This is because the scores $f(x^{(1)}, \tau)$ and $f(x^{(2)}, \tau)$ of two submitted brackets $x^{(1)}$ and $x^{(2)}$ relative to τ are both dependent on τ, and integrating over τ yields a sum over all $2^m = 2^{63}$ possible true brackets for τ, which is intractable. Hence, we use Monte Carlo simulation to approximate expected maximum score and win probability. We approximate expected maximum score via

$$\mathbb{E}\left[\max_{j=1,\ldots,n} f(x^{(j)}, \tau)\right] \approx \frac{1}{B_1}\sum_{b_1=1}^{B_1}\frac{1}{B_2}\sum_{b_2=1}^{B_2}\max_{j=1,\ldots,n} f(x^{(j,b_2)}, \tau^{(b_1)}), \tag{20}$$

where the $\tau^{(b_1)}$ are independent samples from **P** and the $x^{(j,b_2)}$ are independent samples from **Q**. We use a double Monte Carlo sum, with $B_1 = 250$ draws of τ and $B_2 = 100$ draws of $(x^{(1)}, \ldots, x^{(n)})$, because it provides a smoother and stabler approximation than a single Monte Carlo sum. Similarly, we approximate win probability via

$$\mathbb{P}\left[\max_{j=1,\ldots,n} f(x^{(j)}, \tau) \geq \max_{\ell=1,\ldots,k} f(y^{(\ell)}, \tau)\right] \tag{21}$$

$$\approx \frac{1}{B_1}\sum_{b_1=1}^{B_1}\frac{1}{B_2}\sum_{b_2=1}^{B_2}\mathbb{1}\left\{\max_{j=1,\ldots,n} f(x^{(j,b_2)}, \tau^{(b_1)}) \geq \max_{\ell=1,\ldots,k} f(y^{(\ell,b_2)}, \tau^{(b_1)})\right\}, \tag{22}$$

where the $\tau^{(b_1)}$ are independent samples from **P**, the $x^{(j,b_2)}$ are independent samples from **Q**, and the $y^{(\ell,b_2)}$ are independent samples from **R**. We again use a double Monte Carlo sum, with $B_1 = 250$ draws of τ and $B_2 = 100$ draws of $(x^{(1)}, \ldots, x^{(n)})$ and $(y^{(1)}, \ldots, y^{(k)})$, because it provides a smooth and stable approximation.

We are unable to analytically optimize these objective functions to find an optimal strategy $\mathbf{Q}^*$ given the other variables, and we are unable to search over the entire high-dimensional **Q**-space for an optimal strategy. These problems are even more difficult than simply evaluating these objective functions, which itself is intractable. Thus, we apply the entropy-based strategies from the previous sections, which involve generating successively higher entropy brackets as n increases. The idea is to search over a subspace of **Q** that explores strategies of varying entropies, finding the optimal entropy given the other variables. To generate n brackets at varying levels of entropy, we let $\mathbf{Q} = \mathbf{Q}(\lambda)$ vary according to the parameter λ that controls the entropy. In a game in which team i is favored against team j (so $i < j$, since we indexed the teams in decreasing order of team strength, and $\mathbf{P}_{ij} \in [0.5, 1]$), the lowest entropy (chalkiest) strategy features $\mathbf{Q}_{ij} = 1$, the "true" entropy strategy features $\mathbf{Q}_{ij} = \mathbf{P}_{ij}$, and the highest entropy strategy features $\mathbf{Q}_{ij} = 1/2$. We construct a family for **Q** that interpolates between these three poles,

$$\mathbf{Q}_{ij}(\lambda) := \begin{cases} (1-2\lambda)\frac{1}{2} + (2\lambda)\mathbf{P}_{ij} & \text{if } \lambda \in [0, \frac{1}{2}] \text{ and } i < j, \\ (1-2(\lambda-\frac{1}{2}))\mathbf{P}_{ij} + 2(\lambda-\frac{1}{2})1 & \text{if } \lambda \in [\frac{1}{2}, 1] \text{ and } i < j, \end{cases} \tag{23}$$

where $\lambda \in [0, 1]$. The entropy of $\mathbf{Q}(\lambda)$ increases as λ decreases.

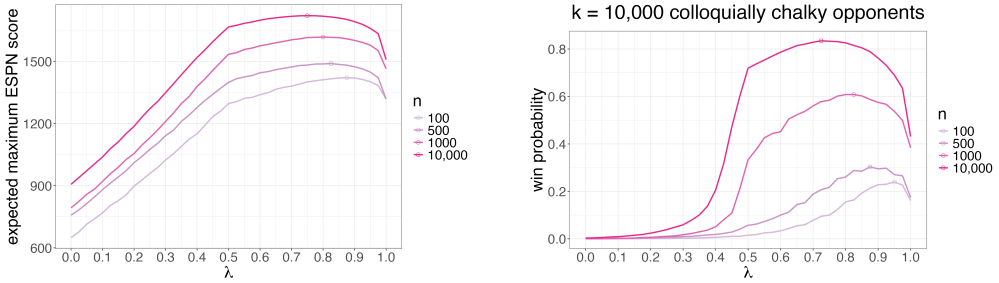
Further, we assume we play against *colloquially chalky* opponents, who usually bet on the higher-seeded team. Each team in the March Madness tournament is assigned a numerical ranking from 1 to 16, their *seed*, prior to the start of the tournament by the NCAA Division I Men's Basketball committee. The seeds determine the matchups in round one and are a measure of team strength (i.e., lower-seeded teams are considered better by the

committee). We suppose colloquially chalky opponents generate brackets according to a distribution $\mathbf{R}^{(\mathrm{cc})}$ based on the seeds s_i and s_j of teams i and j,

$$\mathbf{R}_{ij}^{(\mathrm{cc})} = \begin{cases} 0.9 \text{ if } s_i - s_j < -1, \\ 0.5 \text{ if } |s_i - s_j| \leq 1, \\ 0.1 \text{ if } s_i - s_j > 1, \end{cases} \tag{24}$$

so they usually bet on the higher-seeded team. The exact specification of the colloquially chalky distribution is not important, as we use $\mathbf{R}^{(\mathrm{cc})}$ to illustrate a general point. In future work, one can try to model the distribution of the public's bracket submissions to obtain more precise results.

In Figure 10a, we visualize the expected max score of n brackets generated according to $\mathbf{Q}(\lambda)$ as a function of n and λ. In Figure 10b, we visualize the probability that the max score of n brackets generated according to $\mathbf{Q}(\lambda)$ exceeds that of k = 10,000 colloquially chalky brackets generated according to $\mathbf{R}^{(\mathrm{cc})}$ as a function of n and λ. In both, we again see that we should increase entropy (decrease λ) as n increases. In particular, the small circle (indicating the best strategy given n and k) moves leftward as n increases. Further, we see that tuning the entropy of our submitted bracket set given the other variables yields an excellent win probability, even when n is much smaller than k.



(**a**) (**b**)

Figure 10. (**a**): The expected max ESPN score (y-axis) of n brackets generated according to $\mathbf{Q}(\lambda)$ as a function of n (color) and λ (x-axis). (**b**): The probability (y-axis) that the max ESPN score of n brackets generated according to $\mathbf{Q}(\lambda)$ exceeds that of $k = 10,000$ colloquially chalky brackets generated according to $\mathbf{R}^{(\mathrm{cc})}$ as a function of n (color) and λ (x-axis). The small circle indicates the best strategy given n and k. We want to increase entropy (decrease λ) as n increases.

6. Discussion

In this work, we pose and explore the multi-brackets problem: how should we submit n predictions of a randomly drawn reference bracket (tuple)? The most general version of this question, which finds the optimal set of n brackets across all such possible sets, is extremely difficult. To make the problem tractable, possible, and/or able to be visualized, depending on the particular specification of the multi-bracket pool, we make simplifying assumptions. First, we assume we (and optionally a field of opponents) submit i.i.d. brackets generated according to a bracket distribution. The task becomes to find the optimal generating bracket distribution. For some multi-bracket pools, this is tractable, and for others, it is not. For those pools, we make another simplifying assumption, searching over an intelligently chosen low-dimensional subspace of generating bracket distributions covering distributions of various levels of entropy. We find this approach is sufficient to generate well-performing sets of submitted brackets. We also learn the following high-level lessons from this strategy: we should increase the entropy of our bracket predictions as n increases and as our opponents increase entropy.

We leave much room for future work on the multi-brackets problem. First, it is still an open and difficult problem to find the optimal set of n bracket predictions across *all* such possible subsets, where optimal could mean maximizing expected maximum score, win

probability, or expected profit. Second, in this work, we assume the "true" probabilities **P** and our opponents' generating bracket strategy **R** exist and are known. A fruitful extension of this work would revisit the problems posed in this work under the lens that, in practice, these distributions are either estimated from data or are unknown (e.g., as in Metel [7]). Finally, we suggest exploring more problem-specific approaches to particular multi-bracket pools. For instance, in March Madness bracket challenges, we suggest exploring strategies of varying levels of entropy within each round. Perhaps the public's entropy is too low in early rounds and too high in later rounds, suggesting we should counter by increasing our entropy in earlier rounds and decreasing our entropy in later rounds.

Author Contributions: Conceptualization, R.S.B.; methodology, R.S.B., A.J.W. and I.J.B.; software, R.S.B.; validation, R.S.B.; formal analysis, R.S.B.; investigation, R.S.B.; data curation, R.S.B.; writing—original draft preparation, R.S.B.; writing—review and editing, R.S.B.; visualization, R.S.B.; supervision, A.J.W. and I.J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: See Appendix A.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Our Code

The code and data for this project is publicly available at https://github.com/snoopryan123/entropy_ncaa (accessed on 22 July 2024).

Appendix B. Previous Work Details

Appendix B.1. Estimating Outcome Probabilities in Horse Racing and March Madness

There has been a plethora of research on estimating horse race outcome probabilities. A line of research beginning with Harville [8] estimates the probabilities of the various possible orders of finish of a horse race, assuming knowledge of just the win probabilities of each individual horse. Henery [9], Stern [10], Bacon-Shone et al. [11], Lo and Bacon-Shone [12], and Lo and Bacon-Shone [13] extend this work, developing better and more tractable models. There has also been extensive research on the favorite-longshot bias, the phenomenon that the public typically underbets favored horses and overbets longshot horses, skewing the win probabilities implied by the odds that each horse wins the race. For instance, Asch et al. [14], Ali [15], Rosenbloom [16], Brown and Lin [17], and Hausch et al. [18] illustrate and explain the favorite-longshot bias, and some of these works attempt to adjust for this bias in estimating horse racing outcome probabilities. Yet another line of research focuses on comprehensively estimating these horse finishing probabilities. Bolton and Chapman [19] model outcome probabilities using a multinomial logistic regression model, forming the basis for most modern prediction methods. Chapman [20], Benter [6], Edelman [21], Lessmann et al. [22], and Lessmann et al. [23] extend this work (Benter in particular reported that his team has made significant profits during their five-year gambling operation). Dayes [24] searches for covariates that are predictive of horse race outcome probabilities even after adjusting for the odds, Silverman [25] estimates a hierarchical Bayesian model of these probabilities, and Gulum [26] uses machine learning and graph-based features to estimate these probabilities. For a more comprehensive review of this literature, see Hausch et al. [18].

Similarly, there has been a plethora of research on estimating win probabilities for March Madness matchups. Publicly available NCAA basketball team ratings have been around for decades, for instance, from Massey [27], Pomeroy [28], Sagarin [29], Moore [30], and Georgia Institute of Technology [31]. Other early approaches from Schwertman et al. [32] and Kvam and Sokol [33] use simple logistic regression models to rate teams. Since then,

modelers have aggregated existing team rating systems into ensemble models. For instance, Carlin [34] uses a prediction model that merges Vegas point spreads with other team strength models, Lopez and Matthews [35] merge point spreads with possession-based team efficiency metrics, and FiveThirtyEight [36] combines some of these publicly available ratings systems with their own ELO ratings. Today, people use machine learning or other more elaborate modeling techniques to build team ratings systems. For instance, ESPN's BPI uses a Bayesian hierarchical model to predict each team's projected point differential against an average Division I team on a neutral court [37]. Further, Ji et al. [38] use a matrix completion approach, Goto [39] uses gradient boosting, reHOOPerate [40] uses a neural network, Forsyth and Wilde [41] use a random forest, and Gumm et al. [42], Deshpande [43], and Kim et al. [44] compare various machine learning models. Finally, each year there is a popular Kaggle competition in which contestants submit win probabilities for each game and are evaluated on the log-loss of their probabilities [45].

Appendix B.2. Previous Approaches to Submitting Multiple Brackets to a March Madness Bracket Challenge

For March Madness bracket challenges, there has been limited research on what we should do with team ratings and win probability estimates once we obtain them. Most existing research has focused on filling out one optimal bracket after obtaining win probabilities. For instance, Kaplan and Garstka [46] find the bracket which maximizes expected score and Clair and Letscher [47] find the bracket which maximizes expected return conditional on the behavior of other entrants' submitted brackets. There has also been some work on filling out multiple brackets in the sports analytics community. For instance, Scott Powers and Eli Shayer created an `R` package `mRchmadness` (https://github.com/elishayer/mRchmadness, accessed on 1 June 2023) which uses simulation methods to generate an optimal set of brackets. Also, Tauhid Zaman at the 2019 Sports Analytics Conference (https://www.youtube.com/watch?v=mAgb8A2GDAQ, accessed on 1 June 2023) used integer programming to greedily generate a sequence of "optimal" brackets subject to "diversity" constraints, which force the next bracket in the sequence to be meaningfully different from prior brackets. Nonetheless, we are not aware of any papers which focus on filling out multiple brackets so as to optimize maximum score or win probability.

Appendix C. The Difficulty of Generating Multiple Predicted Brackets

In this section, we discuss the limitations of existing single-event wagering strategies when extended to multi-bracket pools. To start, there are a few key differences between betting on single events and multi-bracket pools. First, in betting on single events, there is no need to submit multiple predictions. In betting on the winner of a basketball game, either one or zero of the teams will have positive expected profit (abbreviated as EV, or expected value), depending on the probabilities and odds. Similarly, to maximize EV in betting on a horse race, bet on the highest EV horse. Second, in betting on single events, there is usually no notion of win probability or a field of opponents because these bets are against the house. Third, whereas multi-bracket pools are one-time events, betting on single events can take a sequential nature in which bettors place a sequence of many bets over a long time horizon. To maximize expected log bankroll in the long run, Kelly [2] devised an algorithm that spreads the bankroll across multiple horses. Only betting on plus EV horses can be suboptimal if there are many other horses with nontrivial win probability. The existence of other horses can yield the need to hedge in order to not go bankrupt. The optimization problem that Kelly solved boils down to calculus.

Directly extending these strategies to betting on the outcome of a random tuple (or bracket) in a bracket pool involves optimally generating a single predicted bracket. Generating a bracket in order to maximize expected score or expected profit, ignoring a field of opponents, is straightforward. First, we consider tournaments featuring a smaller number of possible brackets. For instance, in a pick six contest with m_j horses in race j,

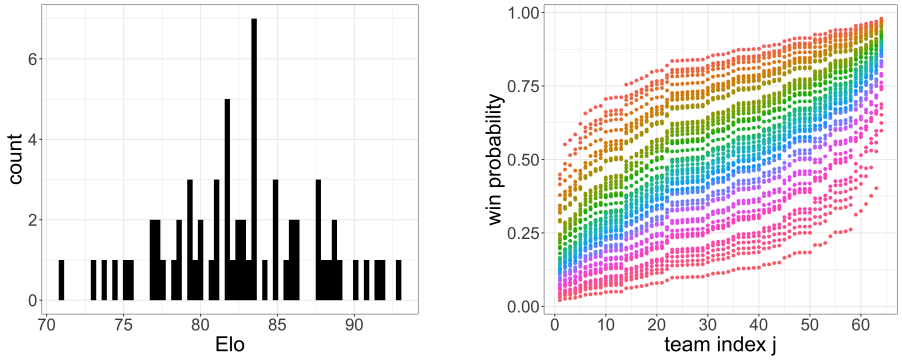
there are $\prod_j m_j$ tickets (e.g., 1 million tickets for $m_j \equiv 10$). To maximize expected score, we simply enumerate tickets by their probability and select the highest one. We can then maximize expected profit, a function of the probabilities and the odds, similarly. For tournaments consisting of a larger number of possible brackets, such as a March Madness bracket challenge, we cannot enumerate the brackets by their likelihood. But we can find the bracket that maximizes expected score: it is the most probable bracket, which features zero upsets.

Generating a bracket in order to maximize win probability or expected profit in a way that accounts for a field of opponents takes more work. To account for opponents, existing work (e.g., Clair and Letscher [47]) assumes opponents pick according to some probability distribution. Then, it is intractable to even evaluate the expected return or win probability of any given bracket since that involves integrating over 2^m realizations of the observed bracket, where m is the size of a bracket (e.g., $m = 63$). Existing work instead approximates these quantities. Clair et al. use a normal approximation to do so. Then, it is still intractable to enumerate the 2^m brackets by approximated expected score and select the best one. Existing work instead approximates this search. Clair et al. use a hill climbing greedy algorithm to do so. Beginning with a subset of brackets, they examine all "neighboring brackets" (brackets who have one differing match outcome), choose the most optimal neighbor, and repeat until they converge to a local optimum.

These strategies do not extend to general multi-bracket pools. For instance, consider March Madness. It is intractable to enumerate the set of $\binom{2^m}{n}$ combinations of size-n bracket subsets, let alone to list them by likelihood or in order of expected score. Further, approximations such as the greedy hill climbing algorithm from Clair and Letscher [47] quickly become ineffectual as n increases, as any tractable number of starting seeds for the algorithm is so substantially smaller than the search space. The set of size-n subsets of brackets is just too large.

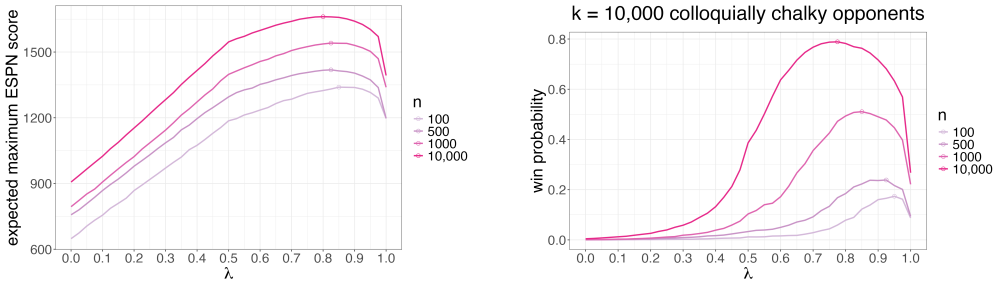
Appendix D. Robustness of Our Results to P

We would like an idea of how our entropy-based strategies are affected by variations in the "true" win probabilities $\mathbf{P}$. Due to the high-dimensional nature of the problem, this is a difficult question. Considering our March Madness tournament example from Section 5.2, to show that our results are not entirely dependent on the idiosyncrasies of the 2021 ELO ratings from FiveThirtyEight, such as Gonzaga having a very high ELO score, we reconduct our simulations using alternative win probabilities $\mathbf{P}$, denoted $\mathbf{P}'$. In particular, we generate new ELO scores $\{\beta_i'\}_{i=1}^{64}$ from a Gaussian using the sample mean and standard deviation of the 64 FiveThirtyEight ELO ratings. We visualize the new ELO scores and their implied win probabilities $\mathbf{P}'$ in Figure A1. Although specific numbers that depend on $\mathbf{P}'$ change accordingly, such as the optimal entropy parameter λ, we find that the qualitative results of our study do not change. In particular, the overall shapes of the optimal entropy curves for $\mathbf{P}'$ in Figure A2 mimic those that use the original $\mathbf{P}$ from Figure 10 in Section 5.2.



(**a**) Histogram of Elo ratings (**b**) $j \mapsto \mathbf{P}'_{ij}$ for each i

Figure A1. On the left, we plot the distribution of Elo ratings. On the right, for each team i, we plot the function $j \mapsto \mathbf{P}'_{ij}$.



(**a**) (**b**)

Figure A2. An analagous plot to Figure 10 in Section 5.2 that uses $\mathbf{P}'$ rather than $\mathbf{P}$. (**a**): The expected max ESPN score (y-axis) of n brackets generated according to $\mathbf{Q}(\lambda)$ as a function of n (color) and λ (x-axis). (**b**): The probability (y-axis) that the max ESPN score of n brackets generated according to $\mathbf{Q}(\lambda)$ exceeds that of $k = 10,000$ colloquially chalky brackets generated according to $\mathbf{R}^{(\mathrm{cc})}$ as a function of n (color) and λ (x-axis). The small circle indicates the best strategy given n and k.

Appendix E. Guessing a Randomly Drawn Bitstring's Details

Appendix E.1. Expected Maximum Score

The expected maximum score of n submitted brackets is

$$\mathbb{E}\left[\max_{j=1,\ldots,n} f(x^{(j)}, \tau)\right] \tag{A1}$$

$$= \sum_{a=0}^{m} \mathbb{P}\left(\max_{j=1,\ldots,n} f(x^{(j)}, \tau) > a\right) \quad \text{by tail sum} \tag{A2}$$

$$= \sum_{a=0}^{m} \left\{1 - \mathbb{P}\left(\max_{j=1,\ldots,n} f(x^{(j)}, \tau) \leq a\right)\right\} \tag{A3}$$

$$= \sum_{a=0}^{m} \left\{1 - \sum_{u=0}^{m} \mathbb{P}\left(\max_{j=1,\ldots,n} f(x^{(j)}, \tau) \leq a \middle| u\right) \mathbb{P}(u)\right\}, \tag{A4}$$

where $u = (u_1, \ldots, u_R)$ and u_{rd} is the number of zeros in τ in round rd. With this definition of u, $\{f(x^{(j)}, \tau)\}_{j=1}^{n}$ are conditionally i.i.d. given u and

$$\mathbb{P}(u) = \prod_{\mathsf{rd}=1}^{R} \mathbb{P}(u_{\mathsf{rd}}) = \prod_{\mathsf{rd}=1}^{R} \mathsf{dbinom}(u_{\mathsf{rd}}, m_{\mathsf{rd}}, 1 - p_{\mathsf{rd}}). \tag{A5}$$

Thus,

$$\mathbb{E}\left[\max_{j=1,\dots,n} f(x^{(j)},\tau)\right] \tag{A6}$$

$$=\sum_{a=0}^{m}\left\{1-\sum_{u=0}^{m}\mathbb{P}\left(f(x^{(j)},\tau)\le a \text{ for all } j\middle|u\right)\mathbb{P}(u)\right\} \tag{A7}$$

$$=\sum_{a=0}^{m}\left\{1-\sum_{u=0}^{m}\mathbb{P}\left(f(x^{(1)},\tau)\le a\middle|u\right)^{n}\mathbb{P}(u)\right\}. \tag{A8}$$

The CDF of the score given u is

$$\mathbb{P}\left(f(x^{(1)},\tau)\le a\middle|u\right) \tag{A9}$$

$$=\mathbb{P}\left(\sum_{\mathsf{rd}=1}^{R}\sum_{i=1}^{m_{\mathsf{rd}}} w_{\mathsf{rd},i}\cdot\mathbb{1}\{x^{(1)}_{\mathsf{rd},i}=\tau_{\mathsf{rd},i}\}\le a\middle|u\right) \tag{A10}$$

$$=\mathbb{P}\left(\sum_{\mathsf{rd}=1}^{R} w_{\mathsf{rd}}\cdot\left(\mathsf{Binom}(u_{\mathsf{rd}},1-q_{\mathsf{rd}})+\mathsf{Binom}(m_{\mathsf{rd}}-u_{\mathsf{rd}},q_{\mathsf{rd}})\right)\le a\right). \tag{A11}$$

This is the CDF of a generalized Poisson Binomial distribution, which we compute in R using the PoissonBinomial package [48].

Appendix E.2. Win Probability

The probability that the maximum score of our n submitted brackets exceeds or ties that of k opposing brackets is

$$\mathbb{P}\left[\max_{j=1,\dots,n} f(x^{(j)},\tau)\ge\max_{\ell=1,\dots,k} f(y^{(k)},\tau)\right] \tag{A12}$$

$$=1-\mathbb{P}\left[\max_{j=1,\dots,n} f(x^{(j)},\tau)<\max_{\ell=1,\dots,k} f(y^{(k)},\tau)\right] \tag{A13}$$

$$=1-\mathbb{P}\left[f(x^{(j)},\tau)<\max_{\ell=1,\dots,k} f(y^{(k)},\tau)\ \forall j\right] \tag{A14}$$

$$=1-\sum_{u=0}^{m}\mathbb{P}\left[f(x^{(j)},\tau)<\max_{\ell=1,\dots,k} f(y^{(k)},\tau)\ \forall j\middle|u\right]\mathbb{P}(u) \tag{A15}$$

$$=1-\sum_{u,a=0}^{m}\mathbb{P}\left[f(x^{(j)},\tau)<a\ \forall j\middle|u\right]\mathbb{P}\left(\max_{\ell=1,\dots,k} f(y^{(k)},\tau)=a\middle|u\right)\mathbb{P}(u) \tag{A16}$$

$$=1-\sum_{u,a=0}^{m}\mathbb{P}\left[f(x^{(1)},\tau)<a\middle|u\right]^{n}\left\{\mathbb{P}\left(\max_{\ell=1,\dots,k} f(y^{(k)},\tau)\le a\middle|u\right)-\mathbb{P}\left(\max_{\ell=1,\dots,k} f(y^{(k)},\tau)\le a-1\middle|u\right)\right\}\mathbb{P}(u) \tag{A17}$$

$$=1-\sum_{u,a=0}^{m}\mathbb{P}\left[f(x^{(1)},\tau)\le a-1\middle|u\right]^{n}\left\{\mathbb{P}\left(f(y^{(1)},\tau)\le a\middle|u\right)^{k}-\mathbb{P}\left(f(y^{(1)},\tau)\le a-1\middle|u\right)^{k}\right\}\mathbb{P}(u). \tag{A18}$$

Here, we condition on $u=(u_1,\dots,u_R)$ where u_{rd} is the number of zeros in τ in round rd. With this definition of u, both $\{f(x^{(j)},\tau)\}_{j=1}^{n}$ and $\{f(y^{(\ell)},\tau)\}_{\ell=1}^{k}$ are conditionally i.i.d. given u. We compute the Generalized Poisson Binomial CDFs of the scores $f(x^{(1)},\tau)$ and $f(y^{(1)},\tau)$ given u as described in Appendix E.1.

In Figure A3, we visualize this win probability as a function of q and r for $p=0.75$ and various values of k and n.

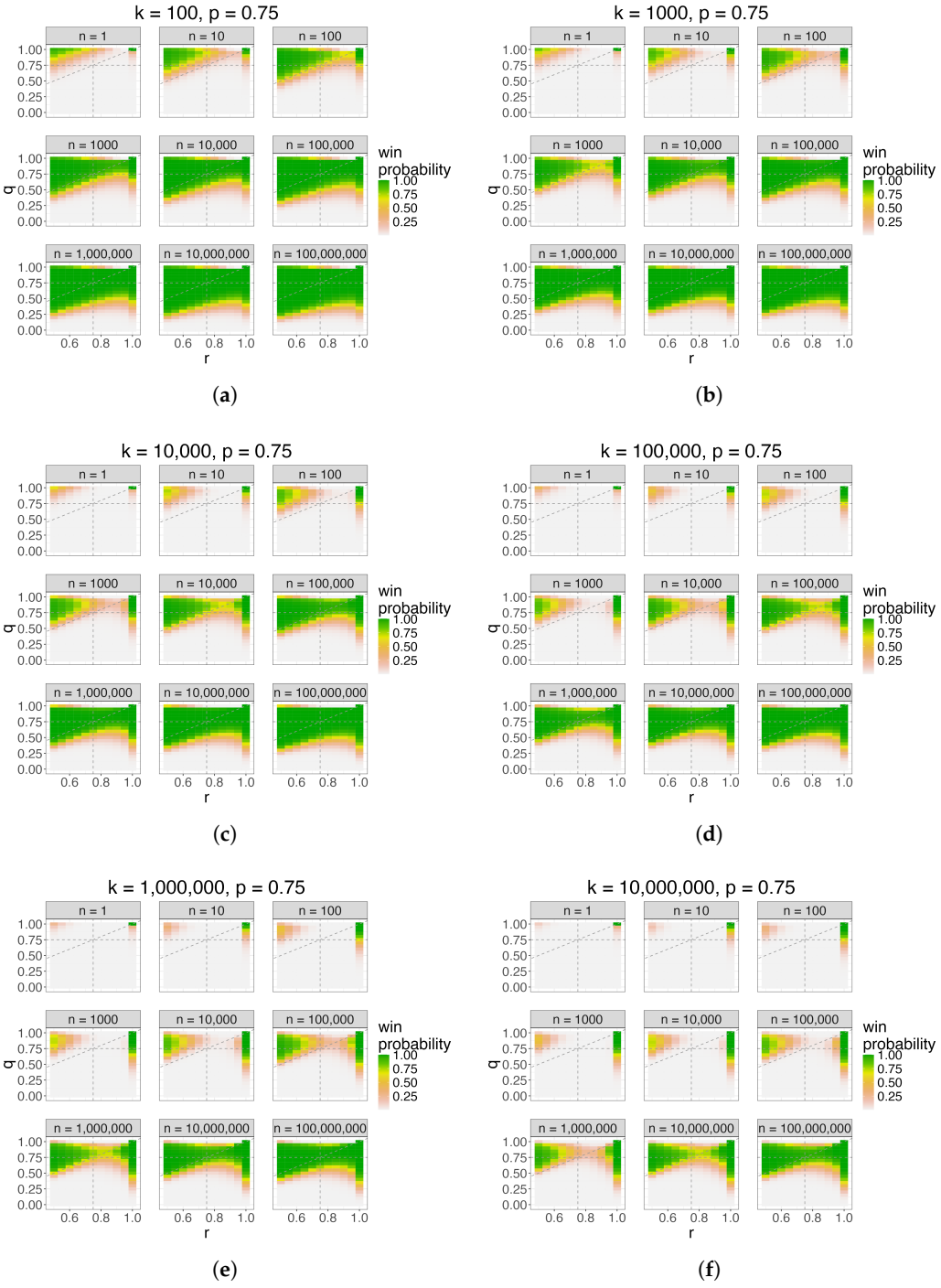


Figure A3. The probability (color) that the maximum Hamming score of n submitted Bernoulli(q) brackets relative to a reference Bernoulli(p) bracket exceeds that of k opposing Bernoulli(r) brackets as a function of q (y-axis), r (x-axis), n (facet), and k (letter) for $p = 0.75$ in the "guessing a randomly drawn bitstring" contest with $p \equiv p_{\text{rd}}$, $q \equiv q_{\text{rd}}$, $r \equiv r_{\text{rd}}$, and $R = 6$ rounds. (**a**) features k = 100, (**b**) features k = 1000, (**c**) features k = 10,000, (**d**) features k = 100,000, (**e**) features k = 1,000,000, (**f**) features k = 10,000,000.

Appendix F. The Asymptotic Equipartition Property

Let $\mathcal{X}$ denote the set of all brackets of length m. Each bracket $x \in \mathcal{X}$ consists of m individual forecasts $x = (x_1, ..., x_m)$. Each forecast x_i has $o \geq 2$ possible outcomes. Let $\mathbb{P}$ be a probability measure on $\mathcal{X}$. The *entropy* of $(\mathcal{X}, \mathbb{P})$ is $H := \mathbb{E}[-\frac{1}{m} \log_2 \mathbb{P}(X)]$, where X is

a bracket randomly drawn from $\mathscr{X}$ according to $\mathbb{P}$, and the *entropy* of a bracket $x \in \mathscr{X}$ is $H(x) := -\frac{1}{m}\log_2 \mathbb{P}(x)$. For instance, in our "guessing a bitstring" example from Section 3, $\mathscr{X}$ is the set of all bitstrings of length m, each individual forecast is a bit, and supposing a bitstring is randomly drawn by m independent Bernoulli(p) coin flips, the entropy is $H = -(p\log_2(p) + (1-p)\log_2(1-p))$.

Letting $\epsilon > 0$, we partition the set of all brackets $\mathscr{X}$ into three subsets,

$$\begin{cases} \epsilon\text{-low entropy "chalky" brackets} & \mathscr{C}_\epsilon := \{x \in \mathscr{X} : \mathbb{P}(x) \geq 2^{-m(H-\epsilon)}\}, \\ \epsilon\text{-"typical" brackets} & \mathscr{T}_\epsilon := \{x \in \mathscr{X} : 2^{-m(H+\epsilon)} < \mathbb{P}(x) < 2^{-m(H-\epsilon)}\}, \\ \epsilon\text{-high entropy "rare" brackets} & \mathscr{R}_\epsilon := \{x \in \mathscr{X} : \mathbb{P}(x) \leq 2^{-m(H+\epsilon)}\}. \end{cases} \tag{A19}$$

Under this definition, an individual chalky bracket is more probable than an individual typical bracket, which is more probable than an individual rare bracket.

The *Asymptotic Equipartition Property* (A.E.P.) from Information Theory, Theorem A1, quantifies our intuition about chalky, typical, and rare brackets from Section 4: as m tends to infinity, the probability mass of the set of brackets becomes increasingly more concentrated in an exponentially small set, the typical set. The proof of Theorem A1 is adapted from Cover and Thomas [49].

The primary mathematical takeaway from Theorem A1 is as follows. The set $\mathscr{X}$ of all possible length m brackets has exponential size $\mathscr{o}^m$, recalling that $\mathscr{o}$ is the number of possible outcomes of each individual forecast (e.g., $\mathscr{o} = 2$ in "guessing a bitstring"). The ϵ-typical set $\mathscr{T}_\epsilon$ comprises a tiny fraction of $\mathscr{X}$, having size $|\mathscr{T}_\epsilon| \approx 2^{mH}$ by part (b) of the theorem. Therefore, $\mathscr{T}_\epsilon$ is exponentially smaller than $\mathscr{X}$, $|\mathscr{T}_\epsilon|/|\mathscr{X}| \approx 2^{mH}/\mathscr{o}^m = 2^{-m(\log_2 \mathscr{o} - H)}$. In the "guessing a bitstring" example with $p = 0.75$ in which the reference bitstring consists of m independent Bernoulli(p) bits, $\mathscr{o} = 2$ and $H \approx 0.81$. Thus, $|\mathscr{T}_\epsilon|/|\mathscr{X}| \approx 2^{-m(0.19)} \approx 0.88^m$. When $m = 63$ (as in March Madness), $|\mathscr{T}_\epsilon|/|\mathscr{X}| \approx 0.00026$, so the typical set of brackets is about 4000 times as small as the full set. This factor increases exponentially as m increases.

Theorem A1 (Asymptotic Equipartition Property)**.** *Let $\epsilon > 0$.*

(a) The typical set asymptotically contains most of the probability mass:

$$\mathbb{P}(\mathscr{T}_\epsilon) \to 1 \text{ in probability as } m \to \infty. \tag{A20}$$

In other words, the reference bitstring is likely a typical bracket.

(b) For m sufficiently large, we can bound the sizes of sets of chalky, typical, and rare brackets in terms of the entropy,

$$\begin{cases} |\mathscr{C}_\epsilon| < 2^{m(H-\epsilon)}, \\ (1-\epsilon)\cdot 2^{m(H-\epsilon)} < |\mathscr{T}_\epsilon| < 2^{m(H+\epsilon)}, \\ |\mathscr{R}| > \mathscr{o}^m - 2^{m(H+\epsilon)} - 2^{m(H-\epsilon)}. \end{cases} \tag{A21}$$

In other words, most brackets are rare, exponentially fewer brackets are typical, and exponentially fewer of those are chalky.

(c) For m sufficiently large, the typical set is essentially the smallest high-probability set: letting $\delta > 0$ and $B_\delta \subset \mathscr{X}$ be any high-probability set with $\mathbb{P}(B_\delta) \geq 1-\delta$, B_δ and $\mathscr{T}_\epsilon$ have similar sizes, $|B_\delta| \geq (1-\epsilon-\delta)2^{m(H-\epsilon)}$. B_δ is a high-probability set when δ is small, and in that case, both $|B_\delta|$ and $|T|$ are essentially bounded below by $(1-\epsilon)2^{m(H-\epsilon)}$.

Proof of Theorem A1.

$$\mathbb{P}(\mathscr{T}_\epsilon) = \mathbb{P}_{X\sim\mathbb{P}}(X \in \mathscr{T}_\epsilon) = \mathbb{P}\big(2^{-m(H+\epsilon)} < \mathbb{P}(X) < 2^{-m(H-\epsilon)}\big) = \mathbb{P}\big(\big|-\frac{1}{m}\log_2 \mathbb{P}(x) - H\big| \geq \epsilon\big), \tag{A22}$$

which converges to 1 by the law of large numbers since $H = \mathbb{E}[-\frac{1}{m}\log_2 \mathbb{P}(X)]$. This proves part (a).

Now,

$$1 = \sum_{x\in\mathcal{X}} \mathbb{P}(x) \geq \sum_{x\in\mathcal{T}_\epsilon} \mathbb{P}(x) > \sum_{x\in\mathcal{T}_\epsilon} 2^{-m(H+\epsilon)} = 2^{-m(H+\epsilon)} \cdot |\mathcal{T}_\epsilon|, \tag{A23}$$

so $|\mathcal{T}_\epsilon| < 2^{m(H+\epsilon)}$. By part (a), for m sufficiently large,

$$1 - \epsilon \leq \mathbb{P}(\mathcal{T}_\epsilon) = \sum_{x\in\mathcal{T}_\epsilon} \mathbb{P}(x) < \sum_{x\in\mathcal{T}_\epsilon} 2^{-m(H-\epsilon)} = 2^{-m(H-\epsilon)} \cdot |\mathcal{T}_\epsilon|, \tag{A24}$$

so $|\mathcal{T}_\epsilon| > (1-\epsilon)\cdot 2^{m(H-\epsilon)}$. Similarly,

$$1 = \sum_{x\in\mathcal{X}} \mathbb{P}(x) \geq \sum_{x\in\mathcal{C}_\epsilon} \mathbb{P}(x) > \sum_{x\in\mathcal{C}_\epsilon} 2^{-m(H-\epsilon)} = 2^{-m(H-\epsilon)} \cdot |\mathcal{C}_\epsilon|, \tag{A25}$$

so $|\mathcal{C}_\epsilon| < 2^{m(H-\epsilon)}$. Therefore,

$$|\mathcal{R}_\epsilon| = |\mathcal{X} \setminus (\mathcal{T}_\epsilon \cup \mathcal{C}_\epsilon)| = \mathcal{o}^m - |\mathcal{T}_\epsilon| - |\mathcal{C}_\epsilon| > \mathcal{o}^m - 2^{m(H+\epsilon)} - 2^{m(H-\epsilon)}. \tag{A26}$$

This proves part (b).

Finally, by part (a), for m sufficiently large,

$$1 - \delta - \epsilon = (1-\epsilon) + (1-\delta) - 1 \leq \mathbb{P}(\mathcal{T}_\epsilon) + \mathbb{P}(B_\delta) - \mathbb{P}(\mathcal{T}_\epsilon \cup B_\delta). \tag{A27}$$

Thus,

$$\begin{aligned} 1 - \delta - \epsilon \leq \mathbb{P}(\mathcal{T}_\epsilon \cap B_\delta) &= \sum_{x\in\mathcal{T}_\epsilon\cap B_\delta} \mathbb{P}(x) \leq \sum_{x\in\mathcal{T}_\epsilon\cap B_\delta} 2^{-m(H-\epsilon)} \\ &= |\mathcal{T}_\epsilon \cap B_\delta| \cdot 2^{-m(H-\epsilon)} \leq |B_\delta| \cdot 2^{-m(H-\epsilon)}, \end{aligned} \tag{A28}$$

so $|B_\delta| \geq \big(1-\epsilon-\delta\big)2^{m(H-\epsilon)}$. This proves part (c). □

Appendix G. Pick Six Details

We can explicitly and quickly compute a tractable lower bound for the expected profit (Formula (15)) under our pick six model from Section 5.1. We begin with

$$\mathbb{E}\left(\frac{W}{W+W_{\text{opp}}}\right) = \mathbb{E}_{\tau\sim\mathbf{P},x\sim\mathbf{Q},y\sim\mathbf{R}}\left(\frac{W}{W+W_{\text{opp}}}\right) \tag{A29}$$

$$= \sum_\tau \mathbb{P}(\tau)\mathbb{E}\left(\frac{W}{W+W_{\text{opp}}}\middle|\tau\right) \tag{A30}$$

$$= \sum_\tau \mathbb{P}(\tau) \sum_{w,w'} \left(\frac{w}{w+w'}\right)\mathbb{P}(W=w, W_{\text{opp}}=w'|\tau) \tag{A31}$$

$$= \sum_\tau \mathbb{P}(\tau) \sum_{w,w'} \left(\frac{w}{w+w'}\right)\mathbb{P}(W=w|\tau)\mathbb{P}(W_{\text{opp}}=w'|\tau) \tag{A32}$$

Since W is conditionally independent of W_{opp} given τ,

$$= \sum_\tau \mathbb{P}(\tau) \sum_{w'} \sum_{w\geq 1} \left(\frac{w}{w+w'}\right)\mathbb{P}(W=w|\tau)\mathbb{P}(W_{\text{opp}}=w'|\tau) \tag{A33}$$

Since if $w = 0$, $w/(w + w') = 0$,

$$\geq \sum_{\tau} \mathbb{P}(\tau) \sum_{w'} \sum_{w \geq 1} \left(\frac{1}{1+w'}\right) \mathbb{P}(W = w|\tau) \mathbb{P}(W_{\text{opp}} = w'|\tau) \tag{A34}$$

Since $w/(w + w') \geq 1/(1 + w')$, which is essentially to say that we will not submit duplicate tickets,

$$= \sum_{\tau} \mathbb{P}(\tau) \mathbb{P}(W \neq 0|\tau) \sum_{w'} \left(\frac{1}{1+w'}\right) \mathbb{P}(W_{\text{opp}} = w'|\tau) \tag{A35}$$

$$= \sum_{\tau} \mathbb{P}(\tau) \mathbb{P}(W \neq 0|\tau) \mathbb{E}\left[\frac{1}{1+W_{\text{opp}}} \middle| \tau\right] \tag{A36}$$

$$\geq \sum_{\tau} \mathbb{P}(\tau) \mathbb{P}(W \neq 0|\tau) \frac{1}{1+\mathbb{E}[W_{\text{opp}}|\tau]} \tag{A37}$$

by Jensen's inequality, since $x \mapsto 1/(1 + x)$ is convex when $x > 0$.

Now,

$$\mathbb{P}(\tau) = \mathbb{P}_{\tau \sim \mathbf{P}}(\tau) = \mathbb{P}(\tau_1, ..., \tau_s) = \prod_{j=1}^{s} \mathbb{P}(\tau_j) = \prod_{j=1}^{s} \mathbf{P}_{\tau_j j}. \tag{A38}$$

Also,

$$\mathbb{P}(W \neq 0'|\tau) = \mathbb{P}_{\tau \sim \mathbf{P}, x \sim \mathbf{Q}}(W \neq 0'|\tau) \tag{A39}$$

$$= \mathbb{P}(\exists \ell \in \{1, ..., n\} \text{ such that } x^{(\ell)} = \tau|\tau) \tag{A40}$$

$$= 1 - \mathbb{P}(\forall \ell \in \{1, ..., n\},\ x^{(\ell)} \neq \tau|\tau) \tag{A41}$$

$$= 1 - \mathbb{P}(x^{(1)} \neq \tau|\tau)^n \tag{A42}$$

Since the $\{x^{(\ell)}\}$ are i.i.d.,

$$= 1 - \mathbb{P}(\exists j \in \{1, ..., s\} \text{ such that } x_j^{(1)} \neq \tau_j|\tau)^n \tag{A43}$$

$$= 1 - \left(1 - \mathbb{P}(\forall j \in \{1, ..., s\},\ x_j^{(1)} = \tau_j|\tau)\right)^n \tag{A44}$$

$$= 1 - \left(1 - \prod_{j=1}^{s} \mathbb{P}(x_j^{(1)} = \tau_j|\tau)\right)^n \tag{A45}$$

Since each of the s races are independent,

$$= 1 - \left(1 - \prod_{j=1}^{s} \mathbf{Q}_{\tau_j j}\right)^n. \tag{A46}$$

Then, by similar logic,

$$\mathbb{E}[W_{\text{opp}}|\tau] = \mathbb{E}_{\tau\sim\mathbf{P},y\sim\mathbf{R}}[W_{\text{opp}}|\tau] \tag{A47}$$

$$= \mathbb{E}\left[\sum_{\ell=1}^{k} \mathbb{1}\{y^{(\ell)} = \tau\}\middle|\tau\right] \tag{A48}$$

$$= \sum_{\ell=1}^{k} \mathbb{P}(y^{(\ell)} = \tau|\tau) \tag{A49}$$

$$= k \cdot \mathbb{P}(y^{(1)} = \tau|\tau) \tag{A50}$$

$$= k \cdot \prod_{j=1}^{s} \mathbb{P}(y_j^{(1)} = \tau_j|\tau) \tag{A51}$$

$$= k \cdot \prod_{j=1}^{s} \mathbf{R}_{\tau_j j}. \tag{A52}$$

Combining all these formulas, we can explicitly and quickly evaluate a lower bound for the expected profit,

$$\mathbb{E}[\text{Profit}] = -n + T \cdot \mathbb{E}\left(\frac{W}{W + W_{\text{opp}}}\right) \tag{A53}$$

$$\geq -n + T \cdot \sum_{\tau} \mathbb{P}(\tau)\mathbb{P}(W \neq 0|\tau)\frac{1}{1 + \mathbb{E}[W_{\text{opp}}|\tau]} \tag{A54}$$

$$= -n + T \cdot \sum_{\tau} \left(\prod_{j=1}^{s} \mathbf{P}_{\tau_j j}\right)\left(1 - (1 - \prod_{j=1}^{s} \mathbf{Q}_{\tau_j j})^{n}\right)\left(\frac{1}{1 + k \cdot \prod_{j=1}^{s} \mathbf{R}_{\tau_j j}}\right). \tag{A55}$$

References

1. Isaacs, R. Optimal Horse Race Bets. *Am. Math. Mon.* **1953**, *60*, 310–315. [CrossRef]
2. Kelly, J.L. A new interpretation of information rate. *Bell Syst. Tech. J.* **1956**, *35*, 917–926. [CrossRef]
3. Rosner, B. Optimal Allocation of Resources in a Pari-Mutuel Setting. *Manag. Sci.* **1975**, *21*, 997–1006. [CrossRef]
4. Willis, K.E. Optimum No-Risk Strategy for Win-Place Pari-Mutuel Betting. *Manag. Sci.* **1964**, *10*, 574–577. [CrossRef]
5. Hausch, D.B.; Ziemba, W.T.; Rubinstein, M. Efficiency of the Market for Racetrack Betting. *Manag. Sci.* **1981**, *27*, 1435–1452. [CrossRef]
6. Benter, W. Computer Based Horse Race Handicapping and Wagering Systems: A Report. In *Efficiency of Racetrack Betting Markets*; World Scientific Publishing Co. Pte. Ltd.: Singapore, 2008; Chapter 19, pp. 183–198.
7. Metel, M. Kelly Betting on Horse Races with Uncertainty in Probability Estimates. *Decis. Anal.* **2017**, *15*, 47–52. [CrossRef]
8. Harville, D.A. Assigning Probabilities to the Outcomes of Multi-Entry Competitions. *J. Am. Stat. Assoc.* **1973**, *68*, 312–316. [CrossRef]
9. Henery, R.J. Permutation Probabilities as Models for Horse Races. *J. R. Stat. Soc. Ser. (Methodol.)* **1981**, *43*, 86–91. [CrossRef]
10. Stern, H. Models for Distributions on Permutations. *J. Am. Stat. Assoc.* **1990**, *85*, 558–564. [CrossRef]
11. Bacon-Shone, J.; Lo, V.S.Y.; Busche, K. *Logistics Analyses of Complicated Bets*; Research Report 11; Department of Statistics, The University of Hong Kong: Hong Kong, China, 1992.
12. Lo, V.S.Y.; Bacon-Shone, J. A Comparison between Two Models for Predicting Ordering Probabilities in Multiple-Entry Competitions. *J. R. Stat. Soc. Ser. Stat.* **1994**, *43*, 317–327. [CrossRef]
13. Lo, V.S.Y.; Bacon-Shone, J. Chapter 4—Approximating the Ordering Probabilities of Multi-Entry Competitions by a Simple Method. In *Handbook of Sports and Lottery Markets*; Hausch, D.B., Ziemba, W.T., Eds.; Handbooks in Finance; Elsevier: San Diego, CA, USA, 2008; pp. 51–65. [CrossRef]
14. Asch, P.; Malkiel, B.G.; Quandt, R.E. Market Efficiency in Racetrack Betting. *J. Bus.* **1984**, *57*, 165–175. [CrossRef]
15. Ali, M.M. Probability models on horse-race outcomes. *J. Appl. Stat.* **1998**, *25*, 221–229. [CrossRef]
16. Rosenbloom, E. A better probability model for the racetrack using Beyer speed numbers. *Omega* **2003**, *31*, 339–348. [CrossRef]
17. Brown, L.D.; Lin, Y. Racetrack betting and consensus of subjective probabilities. *Stat. Probab. Lett.* **2003**, *62*, 175–187. [CrossRef]
18. Hausch, D.B.; Lo, V.S.; Ziemba, W.T. (Eds.) *Efficiency of Racetrack Betting Markets*; World Scientific Publishing Co. Pte. Ltd.: Singapore, 2008.
19. Bolton, R.; Chapman, R. Searching for Positive Returns at the Track. *Manag. Sci.* **1986**, *32*, 1040–1060. [CrossRef]
20. Chapman, R.G. Still searching for positive returns at the track: Empirical results from 2000 Hong Kong races. In *Efficiency of Racetrack Betting Markets*; World Scientific Publishing Co. Pte. Ltd.: Singapore, 2008; Chapter 18, pp. 173–181.

21. Edelman, D. Adapting support vector machine methods for horserace odds prediction. *Ann. Oper. Res.* **2007**, *151*, 325–336. [CrossRef]
22. Lessmann, S.; Sung, M.C.; Johnson, J. Adapting Least-Square Support Vector Regression Models to Forecast the Outcome of Horseraces. *J. Predict. Mark.* **2007**, *1*, 169–187. [CrossRef]
23. Lessmann, S.; Sung, M.C.; Johnson, J. Identifying winners of competitive events: A SVM-based classification model for horserace prediction. *Eur. J. Oper. Res.* **2009**, *196*, 569–577. [CrossRef]
24. Dayes, V.S. Model Considerations for Multi-Entry Competitions. Ph.D. Thesis, San Diego State University, San Diego, CA, USA, 2010.
25. Silverman, N. A Hierarchical Bayesian Analysis of Horse Racing. *J. Predict. Mark.* **2012**, *6*, 1–13. [CrossRef]
26. Gulum, M.A. Horse Racing Prediction Using Graph-Based Features. Ph.D. Thesis, University of Louisville, Louisville, KY, USA, 2018.
27. Massey, K. Massey Ratings: Frequently Asked Questions. Available online: https://masseyratings.com/faq.php (accessed on 5 June 2024).
28. Pomeroy, K. Ratings Explanation. Available online: https://kenpom.com/blog/ratings-explanation/ (accessed on 5 June 2024).
29. Sagarin, J. Jeff Sagarin's College Basketball Ratings. Available online: http://sagarin.com/sports/cbsend.htm (accessed on 5 June 2024).
30. Moore, S. Sonny Moore's Computer Power Ratings. Available online: http://sonnymoorepowerratings.com/m-basket.htm (accessed on 5 June 2024).
31. Georgia Institute of Technology. *LRMC (Classic) Results through Games of 3/5/2023*; Georgia Institute of Technology: Atlanta, GA, USA, 2023. Available online: https://www2.isye.gatech.edu/~jsokol/lrmc/ (accessed on 5 June 2024).
32. Schwertman, N.C.; Schenk, K.L.; Holbrook, B.C. More Probability Models for the NCAA Regional Basketball Tournaments. *Am. Stat.* **1996**, *50*, 34–38. [CrossRef]
33. Kvam, P.; Sokol, J. A logistic regression/Markov chain model for NCAA basketball. *Nav. Res. Logist.* **2006**, *53*, 788–803. [CrossRef]
34. Carlin, B.P. Improved NCAA basketball tournament modeling via point spread and team strength information. In *Anthology of Statistics in Sports*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2005; pp. 149–153.
35. Lopez, M.J.; Matthews, G.J. Building an NCAA men's basketball predictive model and quantifying its success. *J. Quant. Anal. Sport.* **2015**, *11*, 5–12. [CrossRef]
36. FiveThirtyEight. 2022 March Madness Predictions. Available online: https://projects.fivethirtyeight.com/2022-march-madness-predictions/ (accessed on 5 June 2024).
37. ESPN Sports Analytics Team. BPI and Strength of Record: What Are They and How Are They Derived? Available online: https://www.espn.com/blog/statsinfo/post/_/id/125994/bpi-and-strength-of-record-what-are-they-and-how-are-they-derived (accessed on 5 June 2024).
38. Ji, H.; O'Saben, E.; Boudion, A.; Li, Y. March Madness Prediction: A Matrix Completion Approach. In *Modeling, Simulation, and Visualization Student Capstone Conference*; Department of Computer Science, Old Dominion University: Norfolk, VA, USA, 2015.
39. Goto, K. Predicting March Madness Using Machine Learning. Available online: https://towardsdatascience.com/kaggle-march-madness-silver-medal-for-two-consecutive-years-6207ff63b86c (accessed on 5 June 2024).
40. reHOOPerate. Training a Neural Network to Fill Out My March Madness Bracket. Available online: https://medium.com/re-hoop-per-rate/training-a-neural-network-to-fill-out-my-march-madness-bracket-2e5ee562eab1 (accessed on 5 June 2024).
41. Forsyth, J.; Wilde, A. A Machine Learning Approach to March Madness. 2014. Available online: https://axon.cs.byu.edu/~martinez/classes/478/stuff/Sample_Group_Project3.pdf (accessed on 5 June 2024).
42. Gumm, J.; Barrett, A.; Hu, G. A machine learning strategy for predicting march madness winners. In Proceedings of the 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Takamatsu, Japan, 1–3 June 2015; pp. 1–6. [CrossRef]
43. Deshpande, A. Applying Machine Learning To March Madness. Available online: https://adeshpande3.github.io/Applying-Machine-Learning-to-March-Madness (accessed on 5 June 2024).
44. Kim, J.W.; Magnusen, M.; Jeong, S. March Madness prediction: Different machine learning approaches with non-box score statistics. *Manag. Decis. Econ.* **2023**, *44*, 2223–2236. [CrossRef]
45. Sonas, J.; Maggie, W.C. March Machine Learning Mania 2023. 2023. Available online: https://www.kaggle.com/competitions/march-machine-learning-mania-2023/discussion/475046 (accessed on 5 June 2024).
46. Kaplan, E.H.; Garstka, S.J. March Madness and the Office Pool. *Manag. Sci.* **2001**, *47*, 369–382. [CrossRef]
47. Clair, B.; Letscher, D. Optimal Strategies for Sports Betting Pools. *Oper. Res.* **2007**, *55*, 1163–1177. [CrossRef]
48. Junge, F. R Package, Version 1.2.5; PoissonBinomial: Efficient Computation of Ordinary and Generalized Poisson Binomial Distributions. 2022. Available online: https://rdrr.io/cran/PoissonBinomial/ (accessed on 5 June 2024).
49. Cover, T.M.; Thomas, J.A. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*; Wiley-Interscience: Hoboken, NJ, USA, 2006.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.