An Application of Machine Learning To Protect Against Sure Loss in Games of Chance

Matthew Doty Varun Mazumdar

May 28th, 2019

About Me

- Sr. software engineer
- Co-designed the ASIC-resistant mining algorithm for Ethereum
- Worked on the Transiting Exoplanet Survey Sattelite (TESS) for NASA/MIT
- Amateur mathematician

What this Talk is About

Gamblers can make bets where they lose money no matter the outcome.

How can we apply machine learning techniques to avoid this?

Kentucky Derby Prop Bets Arbitrage (1)

On PredictIt.org bets cost $X¢$ and pay out $\$1$.

Using this format (rather than odds), I offer to make three prop bets with you:

- $90¢$ for *Country House* to finish before *Tacitus*
- $90¢$ for *Tacitus* to finish before *Improbable*
- $90¢$ for *Improbable* to finish before *Country House*

How much money are you guaranteed to make?

Kentucky Derby Prop Bets Arbitrage (2)

When I make all these bets I will give you $\$2.70$.

It will never be true that

- "A before B"
- "B before C"
- "C before A"

...all at once!

You have to pay $\$2.00$ *at most*.

That is at least $70¢$ profit, *always*.

Kentucky Derby Prop Bets Regression (1)

What is the nearest triple of bets $\langle X¢, X¢, X¢ \rangle$ to $\langle 90¢, 90¢, 90¢ \rangle$ such that you cannot guarantee a profit?

- $X¢$ for *Country House* to finish before *Tacitus*
- $X¢$ for *Tacitus* to finish before *Improbable*
- $X¢$ for *Improbable* to finish before *Country House's*

Kentucky Derby Prop Bets Regression (2)

You have to pay $\$2.00$ at most.

So if I bet $\langle 66.\bar{6}¢, 66.\bar{6}¢, 66.\bar{6}¢ \rangle$ you are not *guaranteed* a profit.

(I can break even.)

By finding the nearest non-strictly-losing set of bets we have managed a kind of *least squares regression*.

Reservation Price

I can at best break even with the price of $66.\bar{6}¢$ for each bet.

$66.\bar{6}¢$ *reflects the most I should ever pay.*

In auction theory this called my *reservation price* (*1*).

Goal of This Talk

We have computed my *least squares regressed reservation price* from a collection of losing bets in a simple example.

The rest of this talk looks at a general algorithm for doing this using machine learning techniques.

Mathematical Background: Dutch Books

My poor gambling strategy has fallen prey to what probability theorists call a *Dutch Book*.

A theorem from the *subjectivist* school of probability provides sufficient conditions to avoid this (*2*).

Probability theory states my bets should follow a *probability measure* or be vulnerable to arbitrage.

Mathematical Background: Probabilistic Satisfaction

Determining if my bets follow a probability measure is called *Probilistic Satisfaction* (PSAT) in computer science (*3*).

PSAT is about finding a feasible P to matrix equation (*4*):

$$\Pi = VP$$

- Π is a vector of my initial bet prices
- V is a binary design matrix reflecting the outcomes where I recieve a payout or not
- P is a vector reflecting the *weights* of each possible outcome obeying $\forall p \in P . 0 \leq p \leq 1$ and $\sum_{p \in P} p = 1$

Mathematical Background: Least Squares Optimization

I care about a related problem to PSAT.

I want to compute the least squares optimization:

$$\text{argmin}_P \|\Pi - VP\|^2$$

...where $0 \leq P \leq 1$ and $\sum_{p \in P} p = 1$ as before.

Here $\|\cdot\|^2$ is the $L2$ norm.

My regressed bets are $A\hat{P}$ using the least squares minimizing $\hat{P}$.

Mathematical Background: Machine Learning

Convex Optimization is a common approach to machine learning.

One algorithm for solving $\operatorname{argmin}_P \|\Pi - VP\|^2$ is *projected gradient descent*.

In projected gradient descent successive approximations $P_1, P_2, ...$ of the solution are computed with:

$$P_{n+1} = \pi\left(P_n - t_n \nabla_P \|\Pi - VP_n\|^2\right)$$

- π is a projection that finds the nearest vector X where $0 \leq X \leq 1$ and $\sum_{x \in X} x = 1$ (*5*).
- If $t_n = \frac{1}{n}$ this is guaranteed to converge (*6*).

Kentucky Derby Simulation

To test the regression, I simulated random exotic horse bets in the Kentucky Derby (20 horses).

Simulate collections of 2 to 50 bets, 200 samples each.

Random bets include

- Win X, where horse X comes in first
- Place X, where a horse X comes in first or second
- Exacta $X\ Y$, where horse X comes in first and horse Y comes in second
- Quinella $X\ Y$, where both horse X and Y place
- Lay bets against horses winning, placing, exacta bets and quinella bets

Regression Algorithm In Action (1)

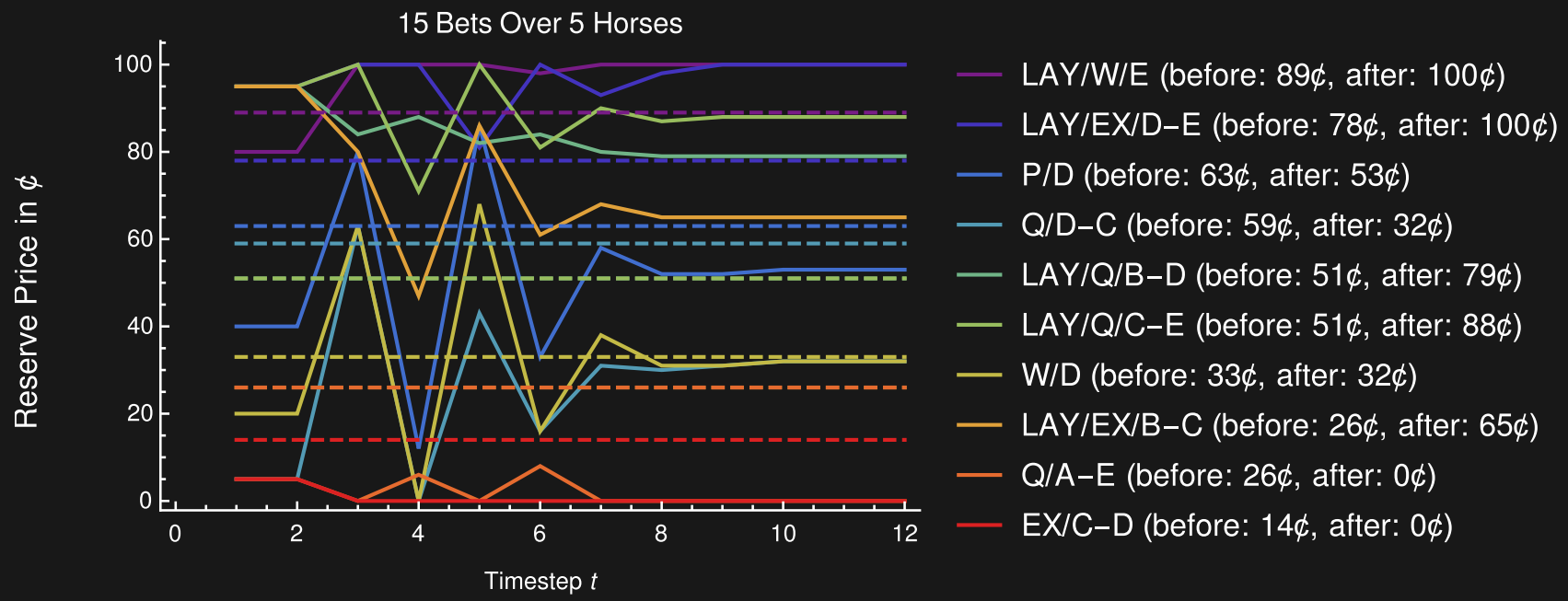

Figure 1: Convergence of Fitted Reserve Prices

Regression Algorithm In Action (2)

Observations:

- The projected subgradient descent is turbulent at the start as it sorts out inconsistency.
- Lay bets tend to increase while other bets tend to decrease.
- *Regression cautions restraint.* According to the algorithm I should avoid many of the bets I initially entertained.

Avoided Bets (1)

As I make more bets, the algorithm suggests I avoid more bets too.

This is because as I make more bets I am more likely to miscalculate.

Avoided Bets (2)

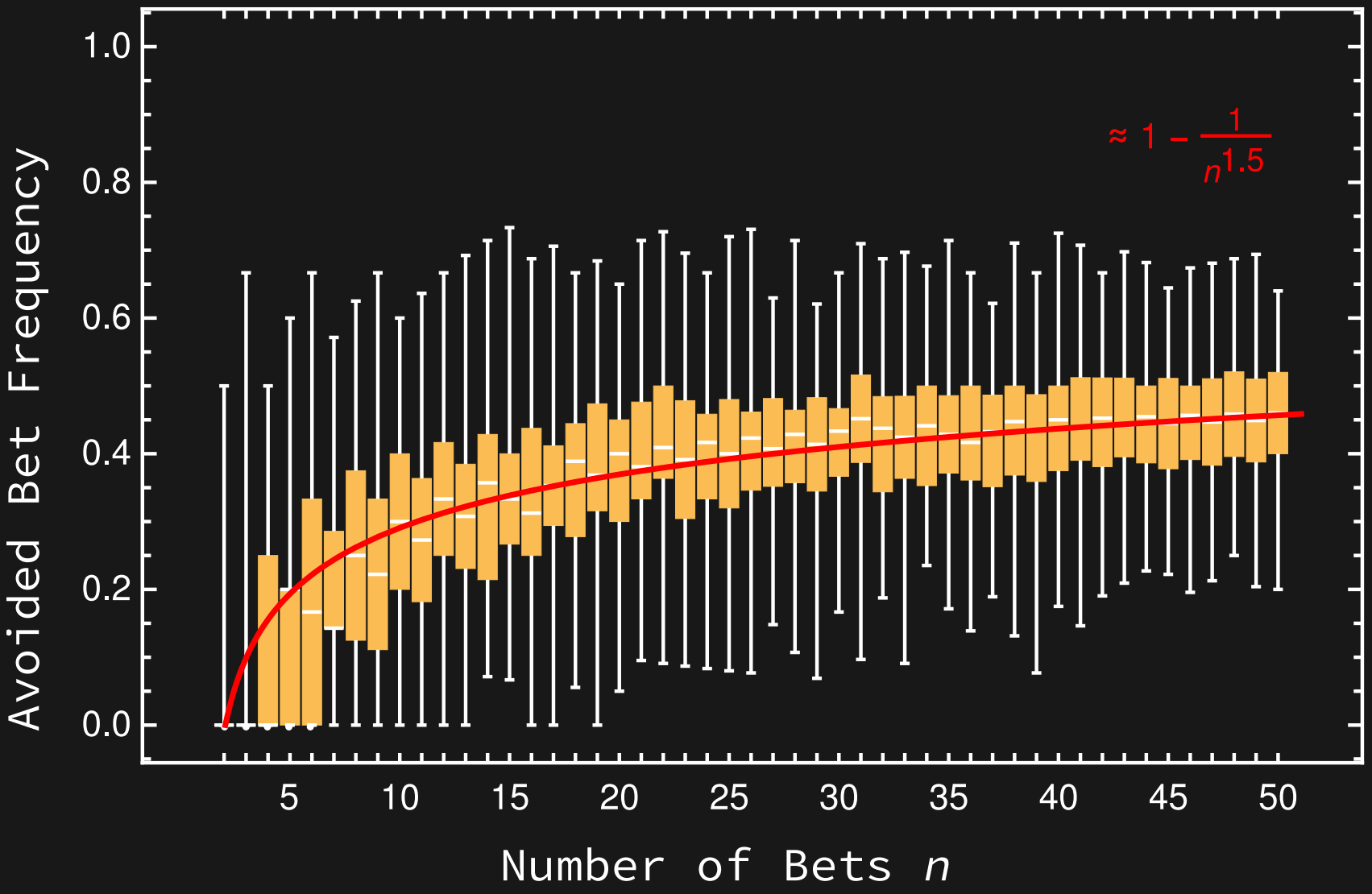

Figure 2: Bets are more likely to be withdrawn as more bets are made

Avoided Bets (3)

If I make n bets the frequency of bets I should avoid grows like $\approx 1 - \frac{1}{n^{1.5}}$

As $n \to \infty$, $1 - \frac{1}{n^{0.15}}$ tends to 1.

In the limit, I should not take any bets at all!

Loss Per Bet (1)

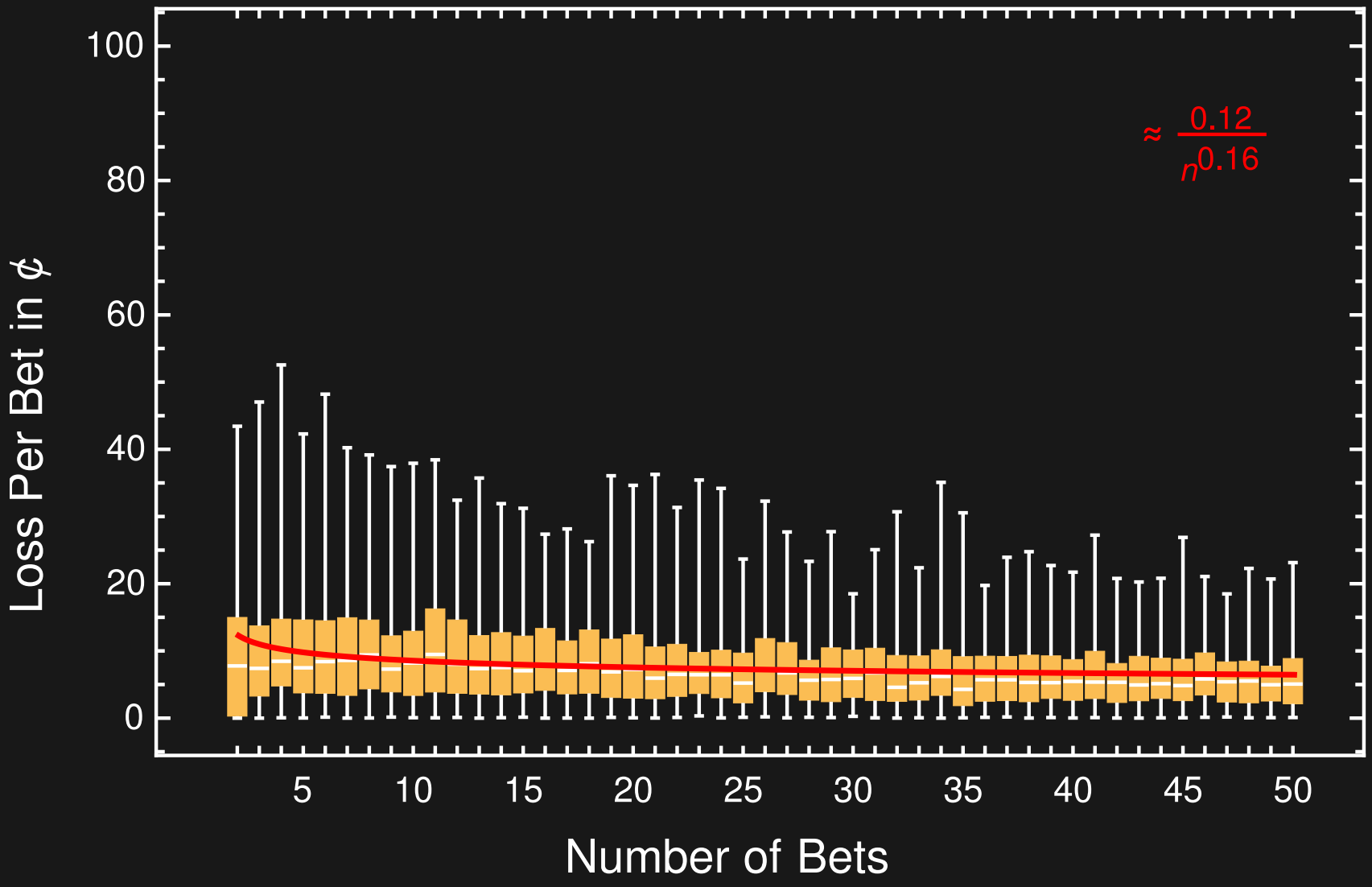

Figure 3: Loss per bet declines as more bets are made

Loss Per Bet (2)

As I make more bets, my loss per bet declines as $\frac{0.12}{n^{0.16}}$.

In the limit, my loss per bet converges to zero.

Paradoxically, in the limit I should take no bets just as my loss per bet goes to 0!

Estimated Loss to Arbitrage

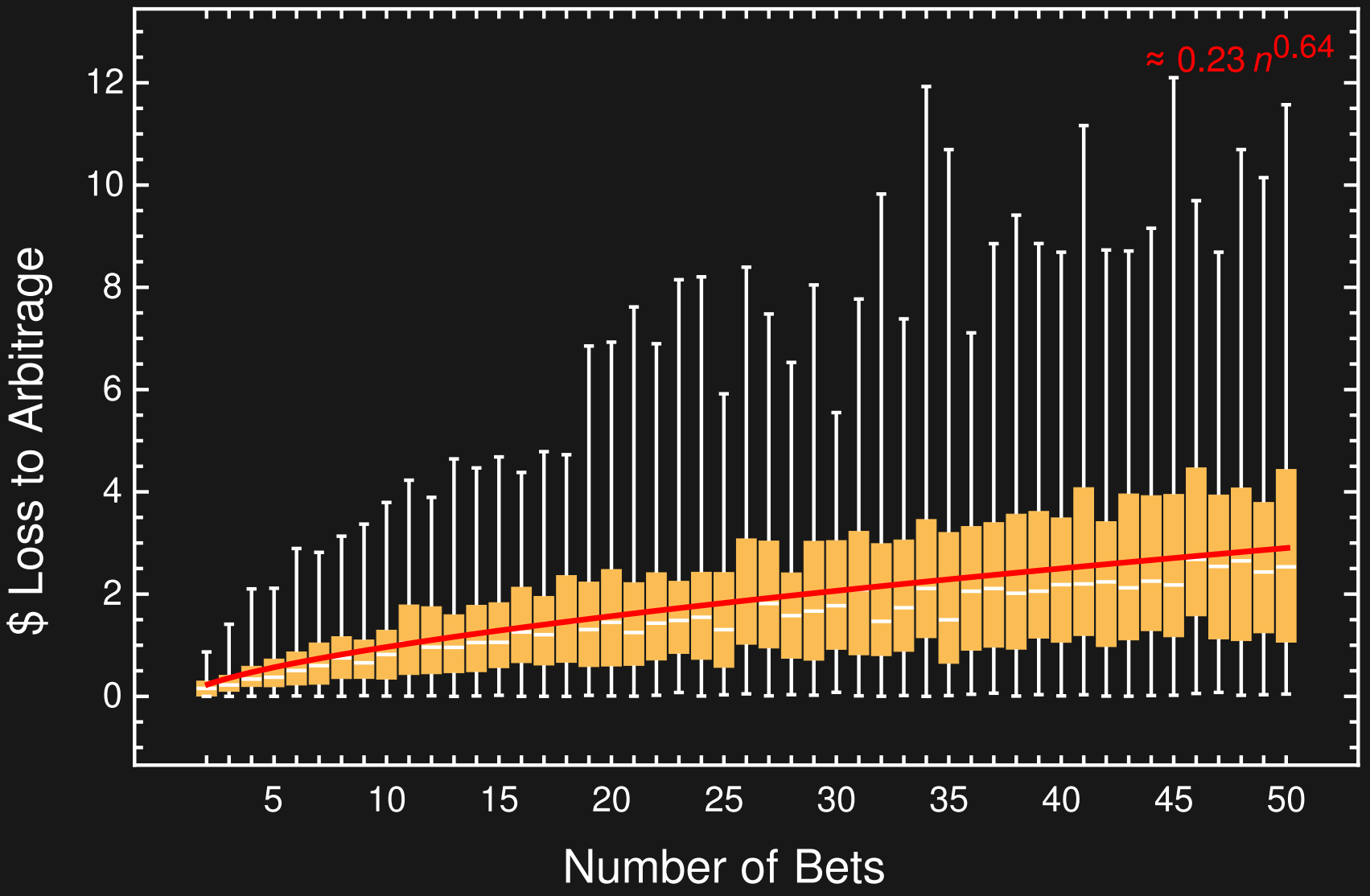

Figure 4: Paradox resolved; loss diverges as more bets are made

Conclusion

- A new machine learning technique for finding vulnerabilities in portfolios has been introduced.
- The algorithm is currently limited and somewhat slow. It needs to be sped up and made more general.
- Hopefully the algorithm is applicable to finance and other games of chance.

References

1. R. B. Myerson, Optimal Auction Design. *Mathematics of Operations Research*. **6**, 58–73 (1981).

2. F. P. Ramsey, "Truth and Probability" (Histoy of Economic Thought Chapters, McMaster University Archive for the History of Economic Thought, 1926), pp. 156–198.

3. G. Georgakopoulos, D. Kavvadias, C. H. Papadimitriou, Probabilistic satisfiability. *Journal of Complexity*. **4**, 1–11 (1988).

4. N. J. Nilsson, Probabilistic logic. *Artificial Intelligence*. **28**, 71–87 (1986).

5. W. Wang, M. Á. Carreira-Perpiñán, Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application (2013) (available at http://arxiv.org/abs/1309.1541).

6. S. Boyd, L. Xiao, A. Mutapcic, Subgradient Methods (2003).