MONEYGANS: GENERATING SYNTHETIC MONEYLINE DATA FOR SPORTS BETTING MARKETS

Jake Birnbach

What is a Moneyline Bet?

- Captures outcome (winner) of sporting event (i.e., NBA, MLB, NFL, etc.)
- Favorite in matchup represented with negative number
  - -150: bettor must stake $150 to win $100 in profit if favorite wins
- Underdog in matchup has positive number
  - +125: an initial bet of $100 bet yields $125 profit if the underdog wins
- Influenced by team performance, injuries, weather conditions, and public betting sentiments.
- Adjusts in real time to reflect current state of game

Need for Synthetic Moneyline Data

- Data Scarcity
  - Limited availability of accurate sports betting data.
  - Sportsbooks don’t allow access to proprietary market data
  - Publicly available data from third parties often expensive and imprecise
- Prevents research into key aspects of sports betting markets such as identifying market inefficiencies or testing new betting strategies
- Sports betting exploding in popularity due to legalization across U.S. states, need further exploration into market dynamics

Applications of Synthetic Data

- Researchers can bypass data availability limitations to gain valuable insights into sports betting market dynamics
- Can be used to develop and train machine learning models which need large amounts of training data
- Beyond sports betting
  - Can generate realistic patient data to develop sophisticated diagnostic models.
  - In the autonomous vehicle field, it can create diverse, realistic scenarios for training self-driving vehicles, enabling advancements in areas with limited access to real-world data.

Problem Formulation

Consider discrete measurements of moneyline data $\boldsymbol{x} = \binom{+\,moneyline}{-\,moneyline}$ at the start of a game S_0. We have our dataset $D = \{\boldsymbol{x_i}\}_{i=1}^{n}$ representing moneyline observations at S_0 for different NBA games. Let $\boldsymbol{X}$ be the space of all moneylines $\boldsymbol{x}$ at S_0. We want to construct a probability distribution $\mathbb{P}_\theta(\boldsymbol{X})$ such that

$$\mathbb{P}_r(\boldsymbol{X}) \approx \mathbb{P}_\theta(\boldsymbol{X})$$

where $\mathbb{P}_r$ is the observed data distribution

Problem Formulation cont.

We want to find parameters θ that minimize the distance between our constructed distribution $\mathbb{P}_\theta$ and our observed data distribution $\mathbb{P}_r$

$$\min_{\theta} D\left(\mathbb{P}_r || \mathbb{P}_\theta\right)$$

We can create $\mathbb{P}_\theta$ using a Wasserstein GAN

Wasserstein GAN (W-GAN)

Minimizes Wasserstein-1 (or EM) distance metric

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_\theta)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||]$$

Intuitively $\gamma(x, y)$ indicates how much “mass” must be transported from x to y to transform the distribution $\mathbb{P}_r$ into $\mathbb{P}_\theta$. Essentially, this distance metric is the “cost” of the optimal transport plan

W-GAN cont.

By the Kantorovich-Rubinstein duality, we can calculate our Wasserstein-1 distance metric using

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{||f||_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)]$$

where $||f||_L \leq 1$ means that $f(x)$ must be 1-Lipschitz (f bounded by linear function with slope of 1.

W-GAN Training

Consider two networks $C_\phi(x)$ and $G_\theta(z)$. First, we sample random vectors $\boldsymbol{z} \sim N(0, I)$ and pass them through the generator network G_θ. We then pass the outputs $G_\theta(\boldsymbol{z})$ and real data observations to the critic network C_ϕ which attempts to classify samples as real or generated. Optimizing both networks

$$\min_{\theta} \max_{\phi} \mathbb{E}_{\boldsymbol{x} \sim \mathbb{P}_{data}}\left[C_\phi(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{z} \sim N(0,I)}\left[C_\phi\left(G_\theta(\boldsymbol{z})\right)\right]$$

results in a generator network that outputs synthetic data which approximates the real data distribution.

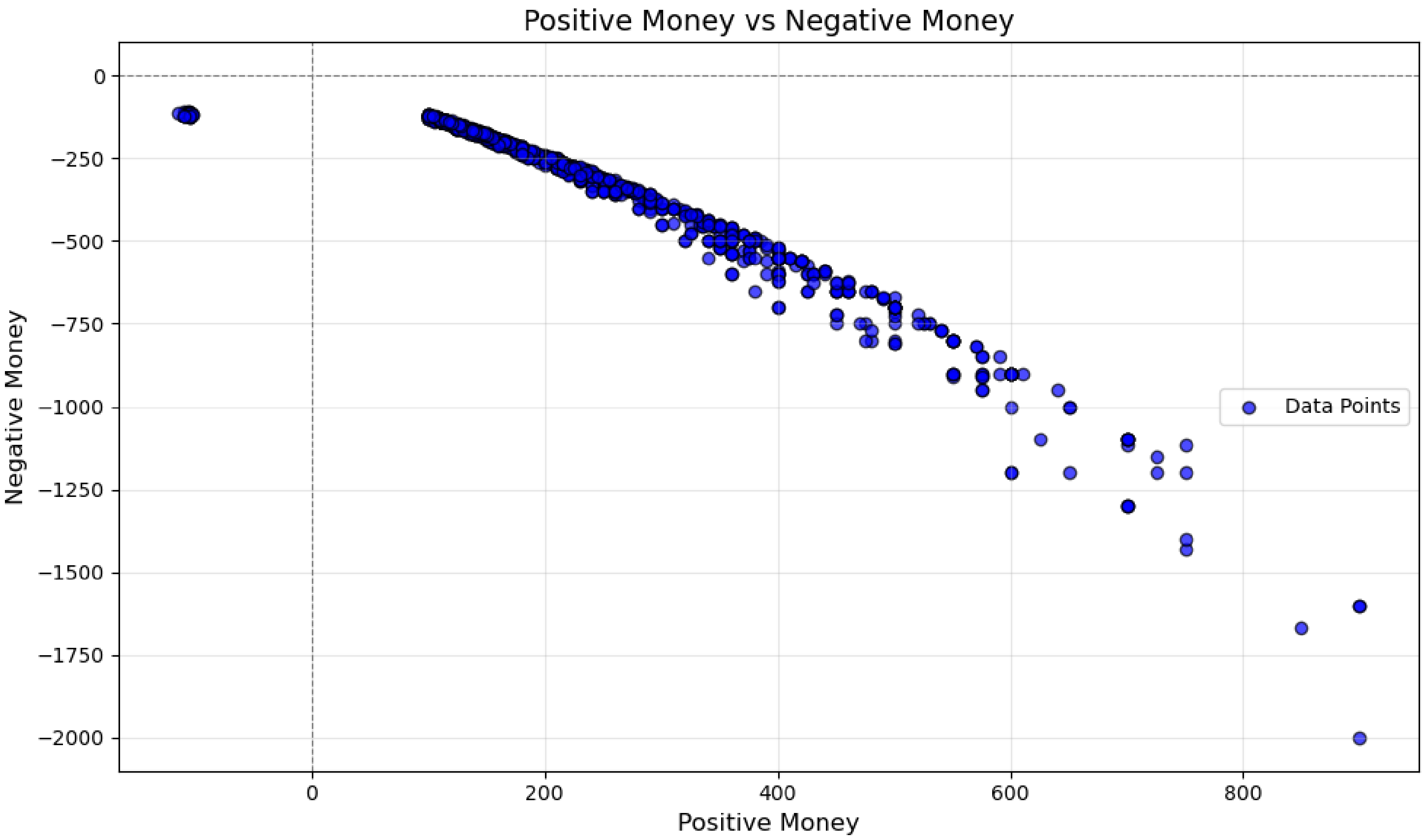
W-GAN Training cont.

Algorithm 1 Wasserstein GAN Training Algorithm

Require: Learning rate α, clipping parameter c, batch size m, number of critic iterations n_{critic}, initial critic parameters w_0, initial generator parameters θ_0

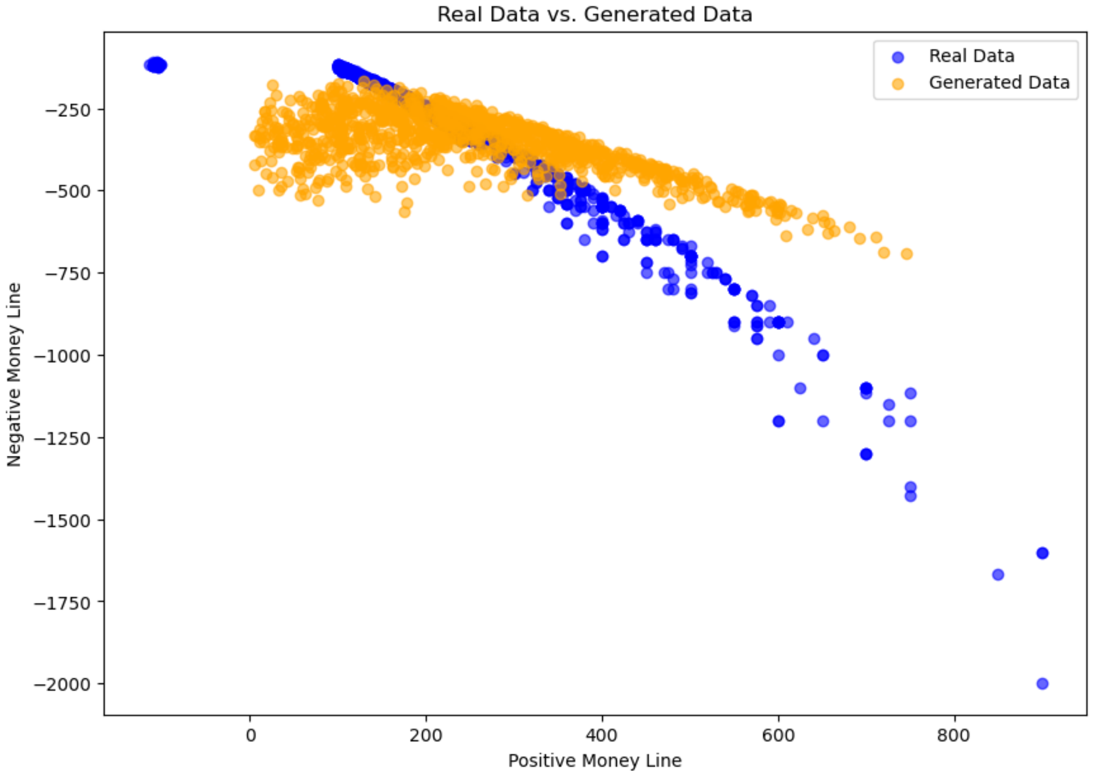
while generator parameters θ have not converged **do**
  for $t = 1$ to n_{critic} **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim P_r$, a batch from the real data
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$, a batch of prior samples
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$, a batch of prior samples
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

Empirical Data Distribution



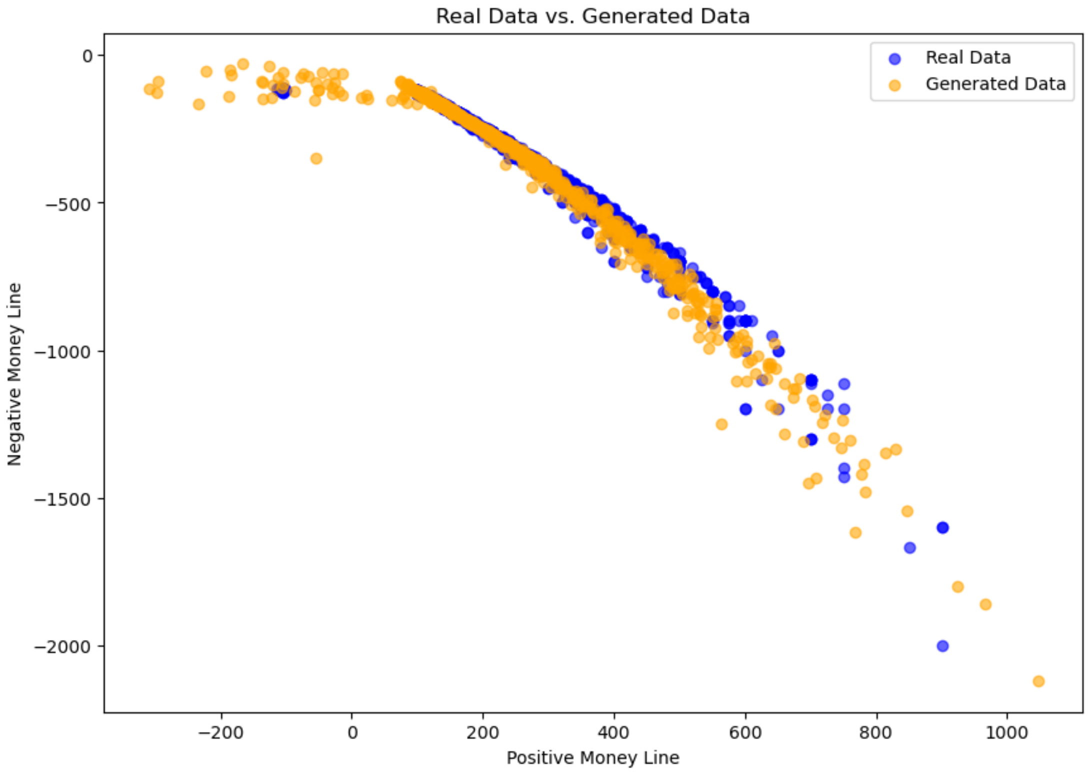
Initial Model

- W-GAN Architecture from 2017 foundational paper (Arjovsky et al. 2017)
- Critic & Generator
  - 1 hidden layer with 128 units
  - ReLU activation functions (leaky ReLU used in critic)
- Enforces Lipschitz in Critic via weight clipping
- Trained for 5000 epochs using RMSProp with $\eta = 0.0001$

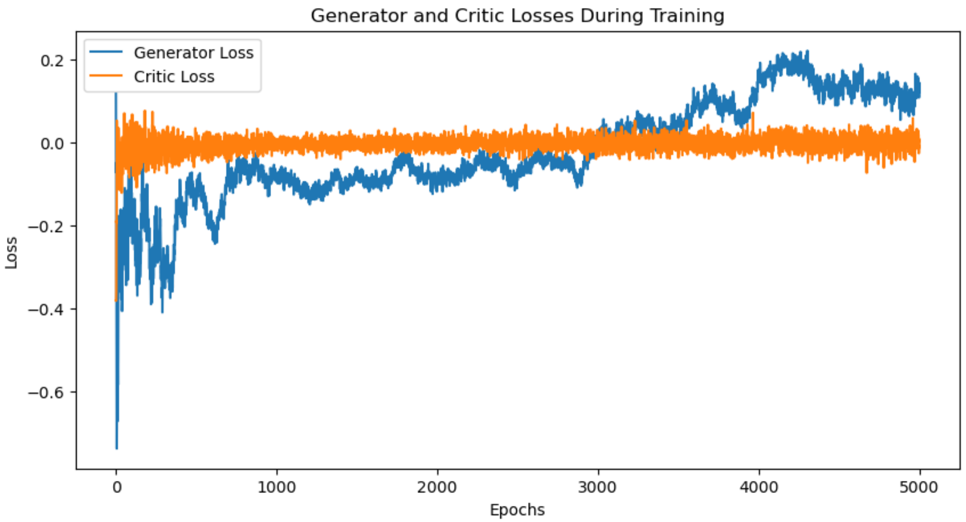


Improved Model with Regularization

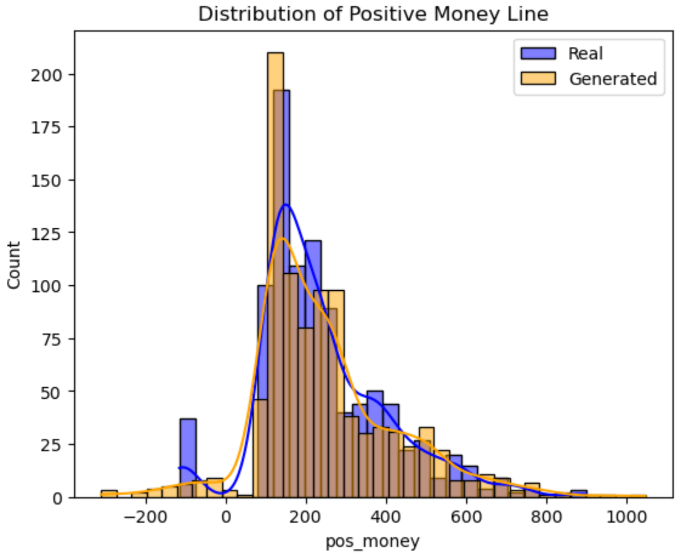
- Larger network, uses Dropout and Batch Normalization
- Critic & Generator
  - 2 hidden layer with 512 and 256 units respectively
  - ReLU activation functions (leaky ReLU used in critic)
- Enforces Lipschitz in Critic via Spectral Normalization (dividing weights matrix by largest singular value)
- Trained for 5000 epochs using ADAM with $\eta = 0.0001$

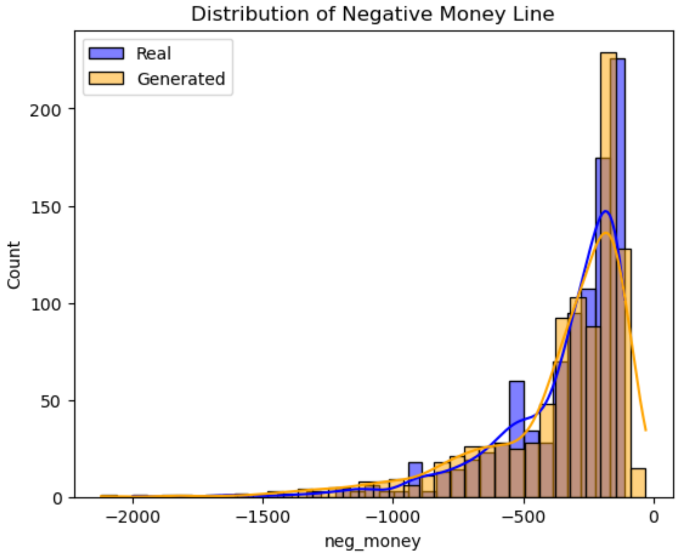


Additional Insights



Appears to be unstable during training





Fails to capture cluster when pos money is negative

Future Work

- Experiment with larger networks on distributed cluster
- Model using traditional mixture modeling
- Incorporate Integral Probability Metrics (IPMs) for better training stability
- Scale to generate synthetic timeseries data (instead of one point in time)

Acknowledgements

Markos Katsoulakis PhD
University of Massachusetts Amherst

Jonathan Larson PhD
University of Massachusetts Amherst

References

[1] J. Drape. 2018. Supreme Court Ruling Favors Sports Betting. The New York Times. Retrieved from https://www.nytimes.com/2018/05/14/sports/sports-betting-supreme-court.html.

[2] C. Green. 2023. Massachusetts Sports Betting Sites: Best Legal MA Sportsbooks. Forbes. Retrieved from https://www.forbes.com/betting/legal/massachusetts-sports-betting-sites/.

[3] M. Rouse. 2023. Money line bet. Investopedia. Retrieved from https://www.investopedia.com/money-line-bet-5217219#:~:text=Money%20line%20bets%20are%20wagers,a%20couple%20of%20possible%20outcomes.

[4] D. A. Harville. 2023. "Modern and post-modern portfolio theory as applied to moneyline betting." Journal of Quantitative Analysis in Sports, vol. 19, no. 2, pp. 73-89. De Gruyter. DOI: https://doi.org/10.1515/jqas-2021-0107.

[5] Forbes Betting Guide. 2023. "How Sports Betting Odds Work." Forbes. Retrieved from https://www.forbes.com/betting/guide/how-sports-betting-odds-work/#where_do_sports_betting_odds_come_from_section.

.

6] Levitt, S. D. 2004. Why are Gambling Markets Organised so Differently from Financial Markets? The Economic Journal, 114(495), 223–246. https://doi.org/10.1111/j.1468-0297.2004.00207

[7] Goodfellow et al. 2014. Generative Adversarial Networks. arXiv. http://arxiv.org/abs/1406.2661.

[8] R. J. Paul and A. P. Weinbach. 2012. Sportsbook behavior in the NCAA football betting market: Tests of the traditional and Levitt models of sportsbook behavior. Journal of Prediction Markets

[9] M. Wiese, R. Knobloch, R. Korn, and M. Kretschmer. 2020. Quant GANs: Deep Generation of Financial Time Series. arXiv. https://doi.org/10.48550/arXiv.1907.06673

[10] Arjovsky et al. 2017. Wasserstein GAN. arXiv. https://doi.org/10.48550/arXiv.1701.07875

[11] C. Villani. 2009. Optimal Transport: Old and New. Springer-Verlag Berlin Heidelberg. DOI: https://doi.org/10.1007/978-3-540-71050-9.

[12] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. 2018. "Spectral Normalization for Generative Adversarial Networks." arXiv. Retrieved from http://arxiv.org/abs/1802.05957. Accessed 6 November 2024

[13] GPT o1-preview. 2024. OpenAi