If (A)I were a Betting Man: Profitable Sports Betting with Deep Reinforcement Learning

Stanford CS 229 Final Project

Anthony Weng
Department of Computer Science
Stanford University
ad2weng@stanford.edu

Abstract

Sports betting is a popular pastime for sport spectators as it creates a personal financial stake in the outcome of a given sporting event. Professional bettors develop complex systems often employing obscure, sport-specific knowledge in an attempt to make profitable bets. We explore the efficacy of using a deep reinforcement learning approach to learn a profitable betting policy using a freely available, sport-agnostic source of information: an outcome's moneyline itself.

For this project milestone, we implement and assess the performance of two baseline strategies: (1) always betting the favorite; and (2) random betting. We find that always betting the favorite outperforms random betting significantly in terms of overall profit despite netting less earnings per successful bet (or "hit"). Next steps include implementing a logistic regression baseline model and the deep reinforcement learning architecture.

0 Key Information

General logistics:

- Category: Theory & Reinforcement Learning.
- External collaborators or mentor(s): None.
- Sharing project with another class: No.
- **Contributions statement:** all work completed by sole author.

Locations of required elements:

- Motivation → Section 1, **Introduction**.
- Methods → Section 2, **Approach**.
- Preliminary experiments → Sections 2 and 3, **Approach** and **Experiments**.
- Next steps → Section 5, **Future work**.

1 Introduction

Sports betting has long been a favored pastime for sport spectators, capable of increasing their perceived level of engagement with a given sporting event by creating a personal financial stake in its outcome. Digital technologies like sportsbook apps (e.g., *FanDuel*, *DraftKings*) have democratized sportsbook access and simplified the betting process – and with the bettor influx necessarily comes those who aim to make a profession out of beating the odds. These dedicated odds analysts employ expert prediction models, arbitrage strategies, and other complex systems in attempt to regularly place profitable bets [1].

While these approaches are capable of generating consistent, long-term returns, their execution can rely on sport-specific knowledge, proprietary assessments, and/or insider information which further obscures the anatomy of such methods. In response to these guarded systems, we aim to develop a profitable betting model using a source of information available to all – an outcome's moneyline itself – by training a betting agent via deep reinforcement learning. In doing so, we hope to find that machine learning technologies can convert even imperfect, public signals of the wisdom of the crowd (i.e., the moneyline history) into a long-term, profitable strategy.

2 Approach

We train a betting agent via deep reinforcement learning (DRL) using a five-layer multilayer perceptron (MLP) to define our agent's decision-making neural architecture. Key elements of the reinforcement learning environment (RLE) experience tuple $(s_t, a_t, r_t, s_{t+1}, d_t)$ are defined as follows:

- State (s_t): a vector $\mathbf{v} \in \mathbb{R}^{144}$ which contains the concatenated 72-hour histories of by-hour moneylines for the home and away teams, respectively.
- Action (a_t): a 0/1 binary output corresponding to betting on the home or away team for a given event, respectively.
- Reward (r_t): the payout from placing a "standard moneyline bet" (as defined in Section 3.2, **Evaluation methods**) on the selected team.
- State terminality (d_t): all states are regarded as terminal states.

Given the technically infinite number of possible 72-hour moneyline histories, we elect to use a model-free algorithm to train our agent. Specifically, we employ deep Q-learning to circumvent the need to design quality exploration policies (which we lack the requisite expert knowledge to do so). Also, two key RL/DRL training techniques are employed: (1) an ϵ-greedy action selection regime; and (2) use of training and target networks. These techniques help to limit early convergence to locally optimal solutions and to encourage stability in the network's Q-value estimations, respectively. Section 3.3, **Experimental details**, provides additional model configuration details (network layer sizes, weight initializations, etc.).

We design and implement three baseline strategies to compare the DRL-trained agent against:

1. The policy represented by a logistic regression model, demonstrating how much traction a simpler ML technique can gain on the problem domain prior to introducing more sophisticated, compute-intensive model architectures.
2. The heuristic strategy of always betting the favorite, helping to separate DRL model success from dataset artifacts related to how accurate the underlying odds data are.
3. The strategy of randomly betting, enabling comparisons against a minimal baseline to ensure we are making meaningful traction on the problem in an absolute sense.

3 Experiments

3.1 Data

The data employed for this work are a subset of those collected by Kaunitz et al. [1] in the relevant work. The original dataset includes 14 months' worth of per-match time series odds, with each time series reporting hourly sampled odds of winning for both the home and away teams from up to 32 bookmakers for 72 hours till the start of each game. The games in question are English football matches (which will also be referred to as soccer).

For this work, we construct an simplified, aggregated dataset $\mathcal{D}$ as follows:

1. Subset the original dataset to entries corresponding to matches played in the 2015 calendar year;
2. Filter entries which concluded in a draw;
3. For both the home and away teams, consider the *median* odds figure presented by the up to 32 bookmakers to be the representative odds of that team winning;
4. Remove entries which do not possess complete, 72-hour odds histories for both the home and away teams; and
5. Convert aggregated odds figures to moneyline figures.

We apply the no-draws filter to enable problem framing as a binary classification task (i.e., an output of 0 can correspond to the home team winning, and a 1 the away team winning). The revised dataset $\mathcal{D}$ contains 9995 aggregated entries and is randomly shuffled and divided into training, validation, and test splits each containing 60/30/10 percent of $\mathcal{D}$, respectively.

3.2 Evaluation methods

For a given model or strategy and for each entry in the test set, we generate a 0/1 output corresponding to placing a bet on the home or away team, respectively. Model/strategy bets are evaluated using the final (i.e., hour 72) moneyline figure offered for the bet-upon team using a "standard moneyline bet". We define a "standard moneyline bet" as follows:

- If the final moneyline m offered for the bet-upon team is positive (i.e., $+400$), our model/strategy agent will bet $\$100$, resulting in a net earning of $\$m$ if the bet-upon team wins (i.e., the bet "hits") and a net loss of $\$100$ otherwise.
- If the final moneyline m is negative (i.e., -2000), our model/strategy agent will bet $\$m$, resulting in a net earning of $\$100$ if the bet-upon team wins and a net loss of $\$m$ otherwise.

Given a series of bets for the test set matches, various metrics including (but not limited to) profitable bet rate, overall profit, and average amounts won and loss are computed and presented in Table 1.

3.3 Experimental details

When the deep reinforcement learning model is created and trained, key information including details on the model configuration, hyperparameter values, training and evaluation times, etc. will be included in this section. For a description of the models/strategies evaluated for the present milestone, please refer to final portion of the **Approach** section.

3.4 Results

Table 1: Preliminary test set metrics for baseline strategies.

	Metric					
Strategy	Profitable bet rate (%)	Overall profit ($)	Avg. amount won ($)	Avg. amount lost ($)	Underdog bet hit rate (%)	Avg. underdog hit winnings ($)
Always favorite	68	43040	110.03	100.00	NA	NA
Random	52	1780	158.87	165.03	33	267.75

To date, we have implemented and evaluated the performance of two of the three projected baseline strategies/models, the heuristic strategy of always betting the favorite and random betting. We observe that always betting the favorite significantly outperforms random betting, despite earning less net winnings per successful bet. The high profitable bet rate and overall profit figures associated with the always-favorite strategy suggest that the draws-filtering step of constructing dataset $\mathcal{D}$ may have exaggerated the proportion of instances in which team moneylines accurately forecast match outcomes (as compared to a real-world setting), especially since the sporting event for which the moneylines are constructed (English football matches) regularly end in draws.

We also note that, for the random betting strategy, the average earnings associated with correctly betting on an underdog[1] upset (i.e., average underdog hit winnings) is noticeably higher than the average amount won on any type of correct bet. This reflects the real-world observation that correctly forecasting underdog upsets can lead to extreme payoffs – suggesting that maximizing DRL system performance on this type of match (and others types such as those with close moneylines for both teams) may be instrumental to producing results better than that of the always-favorite strategy.

4 Future work

- Develop and assess the logistic regression and DRL models (and update Sections 3.3 and 3.4 as necessary).
- Expand error analysis in Section 3.4 **Results** to compare system performance across key match types of interest; e.g., underdog upsets, matches with a clear favorite, matches with close odds, etc.
- Add a section, **Background** for a literature review after **Introduction**.
- Add a section **Future work and limitations** to discuss the following topics (and others):
  - Implications of reducing the problem (with the no-draws filter) to a binary classification task, especially in the context of soccer.
  - How to adapt the problem framework and decision-making agent to generate bet portfolios rather than single-bet decisions.

[1] In betting terminology, the term "underdog" refers to the team/player less favored to win the game as reflected by a less negative/more positive moneyline.

References

[1] Lisandro Kaunitz, Shenjun Zhong, and Javier Kreiner. Beating the bookies with their own numbers - and how the online sports betting market is rigged, 2017.