Bandit Algorithm, Reinforcement Learning, and Horse Racing Result Prediction

LYU2103
Wong Tin Wang David (1155127053)
Sze Muk Hei (1155127477)
Supervised by Prof. Michael R. Lyu
Review

Achievements in Last Semester
Recall that…
Recall that...

Simulated horse betting on the time prediction result from the random forest model
Recall that…

Explored different Bandit algorithms in many constructs (e.g. action sets, reward functions)
On Horse betting
Recall that... 

Applied a tricky technique to attempt to let bandit algorithm decide how much to bet.
Agenda

1. Introduction
2. Data
3. Horse Racing Prediction
4. Betting Strategies
5. Conclusion
6. Q&A
Introduction

Objectives, Contribution
Objectives (2nd Semester)

- Improve **accuracy** and **interpretation** of time prediction model
- Explore new horse betting strategies using **new bandit algorithms** and **other** types of reinforcement learning algorithms
- Enable the agent **bet with different amount of money**
- Enhance the stability of horse betting strategies using **model selecting with EXP3**
WHAT’S NEW?
WHAT’s NEW?

1. Improved Random Forest Model
2. Explored New Bandit Algorithms
3. Applied More RL Algorithms on Horse Betting
4. Model Selection using Bandit Algorithm
Contribution

1. Reduced loss of random forest betting
   - WIN bet
     i. Reduced 87.162% loss
   - PLACE bet
     i. Reduced 46.008% loss
2. Explored possible horse betting strategies generation (PLACE)
   - Neural Bandit / Neural UCB
   - Other reinforcement learning algorithms
3. Enhanced stability of horse betting strategies using model selection
Data

Descriptions, Analysis & Pre-processing
Sources & Descriptions

● **Data Sources**
  a. The Hong Kong Jockey Club
  b. Data Guru
  c. hkHorse

● **Datasets**
  ○ Ranged from 1979 to 2021
  ○ Tables:
    ■ Races data
    ■ Horses data
    ■ Horse-race data
    ■ Betting odds data
## Input Data for Training

<table>
<thead>
<tr>
<th>Features</th>
<th>Types</th>
<th>Encoding Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>raceclass</td>
<td>Categorical</td>
<td>Ordinal</td>
</tr>
<tr>
<td>tracktype</td>
<td>Categorical</td>
<td>One-hot</td>
</tr>
<tr>
<td>racktrack</td>
<td>Categorical</td>
<td>One-hot</td>
</tr>
<tr>
<td>course</td>
<td>Categorical</td>
<td>One-hot</td>
</tr>
<tr>
<td>country</td>
<td>Categorical</td>
<td>One-hot</td>
</tr>
<tr>
<td>importtype</td>
<td>Categorical</td>
<td>One-hot</td>
</tr>
<tr>
<td>sex</td>
<td>Categorical</td>
<td>One-hot</td>
</tr>
<tr>
<td>colour</td>
<td>Categorical</td>
<td>One-hot</td>
</tr>
<tr>
<td>going</td>
<td>Categorical</td>
<td>One-hot</td>
</tr>
<tr>
<td>jockey</td>
<td>Categorical</td>
<td>Ordinal</td>
</tr>
<tr>
<td>trainer</td>
<td>Categorical</td>
<td>Ordinal</td>
</tr>
<tr>
<td>horseid</td>
<td>Categorical</td>
<td>Ordinal</td>
</tr>
<tr>
<td>dam</td>
<td>Categorical</td>
<td>Ordinal</td>
</tr>
<tr>
<td>sire</td>
<td>Categorical</td>
<td>Ordinal</td>
</tr>
<tr>
<td>dam sire</td>
<td>Categorical</td>
<td>Ordinal</td>
</tr>
<tr>
<td>distance</td>
<td>Categorical</td>
<td>Ordinal</td>
</tr>
<tr>
<td>draw</td>
<td>Categorical</td>
<td>Ordinal</td>
</tr>
<tr>
<td>rating</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>rating_rank</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>last_rating</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>avg_rating</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>last_place</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>win odds</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>win odds rank</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>actual weight</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>declared weight</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>gear</td>
<td>Categorical</td>
<td>Customized Encoding</td>
</tr>
<tr>
<td>racetime</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>count [1-3]</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>weight_diff</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>avg_finishtime</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>avg_pos [1-6]</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>avg_pos [1-6]</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>avg_pos [1-6]</td>
<td>Real Value</td>
<td>/</td>
</tr>
<tr>
<td>last_pos [1-6]</td>
<td>Real Value</td>
<td>/</td>
</tr>
</tbody>
</table>

- **Features included**
  - Races data
  - Horses data
  - Horse-race data
  - Additional features
- **Drop unnecessary, irrelevant features**
- **Split train and test data according to race season**
  - **Training** data: 2008 – 2019
  - **Testing** data: 2019 – 2021
Horse Racing Prediction

Procedure, Evaluation & Performance
A New Random Forest
Why Need This?

- Some features are not influential
- Some features are correlated
- Improve accuracy of the model
- Investigate the importance of features among the selected features
Features of New Model

- Features included
  - Horses data
  - Horse-race data
- Extract features with **7 highest importance**
- Split train and test data according to race season
  - **Training** data: 2008 – 2019
  - **Testing** data: 2019 – 2021
Results and Analysis
Evaluation Metrics

- **Mean Squared Error (MSE)**
  - Accuracy of the prediction
  - Closer to 0, the better performance
  - MSE of model: 1.7177 seconds
    - reduced by 24% with value of 0.5472

- **Explained Variance Score**
  - Discrepancy between the model and data
  - The closer to 1, the stronger association
  - Explained Variance Score of model: 0.99547
    - increased by 0.00159
Betting Accuracy

- **WIN Betting**
  - Correctly predicted 24.331% of races
    - Decreased by 0.206% from old model
- **PLACE Betting**
  - Correctly predicted 47.108% of races
    - Decreased by 0.499% from old model
Partial Dependence Plot (Rating)

- Rating has highest feature importance
- Race classes determined by rating
- Inversely proportional to finishing time
- Clear intervals in PDP
  - Matches race classes
- Race class 2 has the most varied results

Table 1.1 Rating for Race classes [57]

<table>
<thead>
<tr>
<th>Rating upper bound</th>
<th>Race classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>80</td>
<td>3</td>
</tr>
<tr>
<td>60</td>
<td>4</td>
</tr>
<tr>
<td>40</td>
<td>5</td>
</tr>
</tbody>
</table>
Partial Dependence Plot (Odds)

- Win odd regards as public intelligence
- **Greatly dropped at low win odds**
  - Horses with low odd may not always win
- Win odd rank shows clear intervals
  - Rank 5 - 11 has a large step up
Error Range of Predictions

- Variance of predictions better than old model
- Average range of predictions: \(0.851\text{ s}\)
  - Reduced by 48.188% from old model
- 22.397% of predictions range > mean
  - Reduced by 4.08% from old model
Error Range of Predictions

- Bet only predictions with small variance
  - Range < mean
  - Reduce loss
- Change of number of correct predictions
  - WIN: Unchanged
  - PLACE: -5.2%
Betting Simulation

1. Group all the horses by the race
2. Order the horses by the predicted finishing time in ascending order
3. Assign a **predicted place** to each horse according to the ranking
4. Start Betting!

![Horse Race Data](image-url)
Betting Simulation

1. Assume $10 would be used for each bet
2. Gain $10 * odds - 10 if correctly picked the horses
3. Lose $10 otherwise
4. **PLACE** betting would be simulated
5. Compare with different strategies
   - Based on lowest odds
   - **Based on highest odds**
   - Based on error range
   - Based on highest rating
   - Random
Betting Simulation

- Betting **PLACE**
- Based on **Highest odds** is the worst
- **New model performs better** (No error checking)
- Based on **Error range** is the best
  - **Old** Model has 39.873% accuracy
  - **New** Model has 41.914% accuracy
Horse Betting Strategies
Bandit part
Improvements
Use better models

Last Sem: Linear models
Possible Problem: lower accuracy

Attempt:
Use more complex models: neural networks
- **Neural UCB** (Single neural network & UCB exploration)
- **Neural Bandit** (neural network committee & epsilon greedy exploration)
Use better models

Doesn’t show significant Improvement in terms of Cash balance

However, the earn rate is 8% higher than that of linUCB
Bets on fewer options

As top 5 horses occupy most out of all horses bet

Bet on only top 5
  • Slight improvements but still losing
Redoing previous approach with these improvement

- Maintain its balance
- But doesn’t earn
- Earn rate still grow overtime
- Lose rate reduces over time
Problems of directly using bandit algorithms on horse betting:

- **Not flexible**
  - unable to consider state information like remaining balance
  - Not easy to make variable amount of bet (Directly set as actions: Failed, always fall to safest option which is $10 bet)

- **Not work for very low odds and insufficient accuracy**
  - Low expected return

Might be better to use more common RL algorithms
Other Algorithms
Why using other algorithms?

- Explore the possibility of finding horse betting strategies using different algorithms
  - Enhance the profitability
  - Able to bet with different amounts of money
- Evaluate the performance of multi-armed bandit by comparing all results
Algorithms used

- Selected from previous projects
  - Deep Q Network
  - Proximal Policy Optimization
- Other model-free, policy-based algorithms
  - Augmented Random Search
  - Cross Entropy Method
Environments

Type 1: only bet with $10
Type 2: bet with different amount of money ($10 - $50)

Data to Use
- Split into train and test set with 707 records each

Features (for each horse):
- Last moment place odds
- Last 10 minutes EMA of odds
- Rankings (odds, predicted finishing time)
- Ratio of finishing time between each horse with the horse ranked 1 place ahead (finishing time)
- Confidence level related (error range, upper and lower bound)
Environments (Type 1)

**Action Set**
- 14 horses (at most) ordered by predicted finishing time
  - + not to bet

**Terminating State**
- No more races
- Cash balance < 9000
Reward Functions (Type 1 & 2)

- $R(\text{Bet any of top 3 horses correctly and error range < mean}) = (\text{dollar bet} \times \text{betting odd}) \times \left(\left(\frac{\text{dollar bet}}{10}\right) + 0.5\right)$
- $R(\text{Bet any of top 3 horses correctly}) = \text{dollar bet} \times \text{betting odd of betted horse}$
- $R(\text{Bet wrong and error range > mean}) = -\text{dollar bet} \times \left(\left(\frac{\text{dollar bet}}{10}\right) + 0.5\right)$
- $R(\text{Bet wrong}) = -\text{dollar bet}$
- $R(\text{Not bet}) = -3$
Reward Convergence (Type 1)
Win Rate (Type 1)

- **DQN** has **highest** win rate
- **ARS** has **highest** loss rate
- ** Majority of** selection are **betting**
• Only **ARS** bet on single option
• **DQN & PPO** has a **safer** strategy
• **CEM**’s strategy involves different risks
Profitability (Type 1)

- All losing money
- DQN perform significantly better than others
Environments (Type 2)

**Action Set**
- 14 horses (at most) ordered by predicted finishing time
  + not bet
- 5 different amount of dollar bets ($10, $20, $30, $40, $50)
- Total actions: $15 \times 5 = 75$

**Terminating State**
- No more races
- Cash balance < 8000
Reward Convergence (Type 2)
Win Rate (Type 2)

- CEM has the highest win rate
- ARS has the highest loss rate
- Majority of selection are betting
Actions Selected (Type 2)

- Only ARS bet on single option
- DQN, PPO & CEM bet safer than before
Money Actions Selected (Type 2)

- PPO & ARS only bet $50
- CEM mostly bet with $10
- DQN bets with different amount
Profitability (Type 2)

- All losing money
- CEM perform better than others
- PPO & ARS has great loss
Overall Comparison
### Optimal Actions Counts

<table>
<thead>
<tr>
<th>Method</th>
<th>Optimal</th>
<th>Sub-optimal</th>
<th>Non-optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural Epsilon</td>
<td>53 (7.50%)</td>
<td>243 (34.37%)</td>
<td>411 (58.13%)</td>
</tr>
<tr>
<td>Neural UCB</td>
<td>65 (9.19%)</td>
<td>117 (16.55%)</td>
<td>525 (74.26%)</td>
</tr>
<tr>
<td>DQN</td>
<td>59 (8.35%)</td>
<td>205 (29.00%)</td>
<td>443 (62.66%)</td>
</tr>
<tr>
<td>PPO</td>
<td>63 (8.91%)</td>
<td>197 (27.86%)</td>
<td>447 (63.22%)</td>
</tr>
<tr>
<td>ARS</td>
<td>89 (12.59%)</td>
<td>127 (17.96%)</td>
<td>491 (69.45%)</td>
</tr>
<tr>
<td>CEM</td>
<td>59 (8.35%)</td>
<td>84 (11.88%)</td>
<td>564 (79.77%)</td>
</tr>
</tbody>
</table>

- **Optimal:**
  - Place: **top 3**
  - Reward: **top 3**
- **Sub-optimal:**
  - Place: **top 3**
- **Non-optimal:**
  - otherwise
Overall Comparison (Bet 1 option)

- Lowest odd $> 1.5$ outperform other
- **DQN** perform the best among all algorithms
Model Selection
Why Model Selection

**Model selection**: selecting best suitable model at each time step
- No guarantee that a particular algorithm consistently performs well
- The best performing algorithm might be different over time
- Combining power of different algorithms

**How?**
We again use bandit algorithm (**EXP3**)
Why EXP3

**EXP3** (Exponential-weight algorithm for Exploration and Exploitation)
- Adversarial Bandit (no assumption to make it work)
- Only update belief by reward (we don’t use contextual since it would be just trying to approximate other algorithms)
- Sensitive to reward changes (exponential)

Suitable when the behavior of algorithms might constantly changing
Procedure

EXP3 picks one algorithm at a time and bet according to the decision of the chosen algorithm

Action Set

● Algorithms include DQN, PPO, ARS, CEM, neural bandit, neural UCB.
● All run on the simplest setting (bet on 1 horse at a time with $10 bet)

Reward

● Reward of selected algorithm by betting on its decided horse
Result

Using EXP3 to do model selection outperforms any single algorithm!
Comparing algorithms by EXP3

- DQN most selected overtime and follow by ARS and CEM
- Bandit is less selected which shows its weaker performance compared to others
Observing betting strategy from EXP3

- Horse 5 is selected the most. And followed by 7, 11, 4, 2
- Not bet is seldomly chosen
- Almost all are not safe options but EXP3 doesn’t lose much at the end
Conclusion
Conclusion

- Horse racing prediction model
  - Enhanced interpretability of random forest
    - Showed how features affect the results
  - Acceptable betting strategy
    - Based on error range
    - Reduced loss without missing a lot profits

- Horse betting strategies
  - Bandit algorithms
    - Not flexible (variable bet, unaware of state like cash balance, hardly profit for negative expected return)
    \rightarrow better use other algorithms
    - But can be used in other scenarios
    - Shown good performance in model selection
  - Other algorithms
    - Comparable accuracy and profits to the bandits
6
Q&A Section
The End

Thanks!