Predicting Horse Racing Results with TensorFlow

LYU 1703

LIU YIDE
WANG ZUOYANG
CUHK Professor, Gu Mingao, wins 50 MILLIONS dividend using his “sure-win” statistical strategy.
AlphaGO defeats human world champions at the Chinese ancient game of GO.
Motivation

Can we predict the horse racing results, using

- Machine Learning (specifically, Neural Network) only
- instead of statistical inference*

* Professor Gu’s work on this topic is NOT PUBLISHED by the time of the presentation.
Related Work

Few work on related topic is published.

◉ Williams and Li (2008)
  ○ Reviewed neural network algorithms. (BP, Quasi Newton, etc.)
  ○ Predicted horse finishing time of individual horses.
  ○ Claimed to have great performance (little result data).

◉ LYU1603
  ○ Predicted horse finishing time of all horses.
  ○ Obtained actual net gain with a threshold (>95%)
  ○ Problem: too high threshold (bet <10 times in a season)
Introduction

Outline

- Background
- Two Approaches
  - Additional information - Weather
  - Divide and Conquer
- Model Architecture
- Results & Discussion
- Conclusion & Future Work
- Q&A
1 Background
**Horse Racing Background**

- **Professional sport to run horse in time**
  - Horses are competing in a game for speed.

- **Professional & National entertainment events for Hong Kong citizens**
  - Over 45% of citizens have betting account.
  - Advanced Pari-mutuel betting.
  - >20 bet types.
Objective

<table>
<thead>
<tr>
<th>Bets</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win</td>
<td>1st in a race</td>
</tr>
<tr>
<td>Place</td>
<td>1st, 2nd, 3rd in a race</td>
</tr>
</tbody>
</table>

Table 1: Bets of focus in this project

Objective: Build a prediction model to obtain positive net gain.
Possible ways to model results

Horse racing result is very difficult to model.

- Horse win
  - Predict whether a horse will win
  - Binary classification of win or not
  - Problems:
    - Unevenly distributed dataset (1 win and 13 losses, normally)
    - Cannot model a race
    - Repetitive wins in a race
Possible ways to model results

Horse racing result is very difficult to model.

- Horse ranks
  - Predict ranks of horses in a race
  - Multi-class classification
  - Problems:
    - Races of different horses
    - Ambiguous
    - Repetitive

<table>
<thead>
<tr>
<th>Horse\Place</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>60%</td>
<td>40%</td>
<td>20%</td>
</tr>
<tr>
<td>#2</td>
<td>30%</td>
<td>60%</td>
<td>50%</td>
</tr>
<tr>
<td>#3</td>
<td>50%</td>
<td>40%</td>
<td>60%</td>
</tr>
</tbody>
</table>
Possible ways to model results

Horse racing result is very difficult to model.

- Horse finishing time
  - Predict horse finishing time in a race
  - Regression problem
  - Reflect recent horse strength to some extent
  - Problems:
    - Predict finishing time individually
    - But then grouped into a race
2 Approach
Approach

- Additional Information
  - Weather
  - Extract horse racing features
    - Weight difference/ Previous Place

- Divide and Conquer
  - Divide on location
  - Shatin (ST) and Happy Valley (HV)
  - (Extract horse racing features)
Weather Features

- Horse Performance is influenced by the weather
  - Average performance
  - Individual performance

- Collected Features:
  - Moon phase
  - Wind direction and speed
  - Humidity and weather condition
  - Temperature
Average Performance

- Average horse finishing time can be influenced by weather features
  - Temperature $\uparrow$
  - Finishing time $\downarrow$

*Finishtime is averaged and normalized by distance to represent horse performances.*
Individual Performance

- Individual horse has different performances in different weather

- Weather is closely correlated to both average and individual performances.

*Finishtime normalized by distance to represent horse performances.*
Why Divide and Conquer

- Two racecourses: Sha Tin and Happy Valley;
- Previous studies show some patterns;
- Tuning sub-models to optimize in the future.
Divide and Conquer By Location

- Split the data set into two subsets;
- Build and train NN models based on both subsets;
- Predict separately on both models and combine.
Win odds?

- Odds is closely related to the prediction by intuition.
- However LYU 1603 chose to exclude this feature.
- Compare models with odds and without odds to figure it out.
3 Configuration

Structures and settings of the models
Layer and batch size

- Commonly used structures are used for this semester;
- Number of layers: 2;
- Batch size: 128;
- We assume this network configuration is representative.
Train & Test data set

- Need to be comparable to LYU 1603 and 1604;
- Train data: 2011 - 2014;
- Test data: 2015 - 2016.
Number of training steps

To search for a best number of training steps, a simple experiment is conducted.

<table>
<thead>
<tr>
<th>Number of Steps</th>
<th>noodds_noweather</th>
<th>noodds_weather</th>
<th>odds_noweather</th>
<th>odds_weather</th>
</tr>
</thead>
<tbody>
<tr>
<td>10k</td>
<td>4.025</td>
<td>3.603</td>
<td>4.347</td>
<td>3.263</td>
</tr>
<tr>
<td>100k</td>
<td>4.291</td>
<td>4.697</td>
<td>4.819</td>
<td>3.668</td>
</tr>
<tr>
<td>1m</td>
<td>5.192</td>
<td>5.221</td>
<td>5.088</td>
<td>4.281</td>
</tr>
</tbody>
</table>

*Table 3.1: Experiments on the number of training steps*
Evaluation Standard

- **Loss:** Mean-square-error between predicted and actual finishing time
- **Accuracy_win:** Accuracy of correct win bets
- **Accuracy_place:** Accuracy of correct place bets
- **Net gain:** Overall profits of all bets
Results & Discussion
## Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Model 000</th>
<th>Model 001</th>
<th>Model 010</th>
<th>Model 011</th>
<th>Model 100</th>
<th>Model 101</th>
<th>Model 110</th>
<th>Model 111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>515.2</td>
<td>461.2</td>
<td>556.4</td>
<td>417.7</td>
<td>583/575</td>
<td>527/536</td>
<td>629/577</td>
<td>652/589</td>
</tr>
<tr>
<td>Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>win</td>
<td>0.08367</td>
<td>0.07029</td>
<td>0.08090</td>
<td>0.10742</td>
<td>0.08798/0.08014</td>
<td>0.07725/0.07292</td>
<td>0.08155/0.09028</td>
</tr>
<tr>
<td>Accuracy</td>
<td>place</td>
<td>0.42926</td>
<td>0.41954</td>
<td>0.47547</td>
<td>0.47789</td>
<td>0.44277/0.43902</td>
<td>0.43419/0.46766</td>
<td>0.44778/0.47052</td>
</tr>
<tr>
<td>Net gain</td>
<td>-1087</td>
<td>-991</td>
<td>-1378</td>
<td>-568</td>
<td>37/-1005</td>
<td>-1088/-1579</td>
<td>655/-917</td>
<td>339/-1724</td>
</tr>
</tbody>
</table>

- Notation: three binary digits representing divided/undivided, odds/no odds and weather/no weather.
- For the divided models, the first values refer to Sha Tin and the second refer to Happy Valley.
## Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Model 000</th>
<th>Model 001</th>
<th>Model 010</th>
<th>Model 011</th>
<th>Model 100</th>
<th>Model 101</th>
<th>Model 110</th>
<th>Model 111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>515.2</td>
<td>461.2</td>
<td>556.4</td>
<td>417.7</td>
<td>583/575</td>
<td>527/536</td>
<td>629/577</td>
<td>652/589</td>
</tr>
<tr>
<td>Accuracy _win</td>
<td>0.08367</td>
<td>0.07029</td>
<td>0.08090</td>
<td>0.10742</td>
<td>0.08798/0.08014</td>
<td>0.07725/0.07292</td>
<td>0.08155/0.09028</td>
<td>0.07940/0.06944</td>
</tr>
<tr>
<td>Accuracy _place</td>
<td>0.42926</td>
<td>0.41954</td>
<td>0.47547</td>
<td>0.47789</td>
<td>0.44277/0.43902</td>
<td>0.43419/0.46766</td>
<td>0.44778/0.47052</td>
<td>0.4542/0.47685</td>
</tr>
<tr>
<td>Net gain</td>
<td>-1087</td>
<td>-991</td>
<td>-1378</td>
<td>-568</td>
<td>37/-1005</td>
<td>-1088/-1579</td>
<td>655/-917</td>
<td>339/-1724</td>
</tr>
</tbody>
</table>

- **Loss:**
  - Weather features reduce prediction loss.
  - Win odds increases prediction loss.
  - Dividing the dataset will increase prediction loss.
Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Model 000</th>
<th>Model 001</th>
<th>Model 010</th>
<th>Model 011</th>
<th>Model 100</th>
<th>Model 101</th>
<th>Model 110</th>
<th>Model 111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>515.2</td>
<td>461.2</td>
<td>556.4</td>
<td>417.7</td>
<td>583/575</td>
<td>527/536</td>
<td>629/577</td>
<td>652/589</td>
</tr>
<tr>
<td>Accuracy_A</td>
<td>0.08367</td>
<td>0.07029</td>
<td>0.08090</td>
<td>0.10742</td>
<td>0.08798/0.08014</td>
<td>0.07725/0.07292</td>
<td>0.08155/0.09028</td>
<td>0.07940/0.06944</td>
</tr>
<tr>
<td>Accuracy_A</td>
<td>0.42926</td>
<td>0.41954</td>
<td>0.47547</td>
<td>0.47789</td>
<td>0.44277/0.43902</td>
<td>0.43419/0.46766</td>
<td>0.44778/0.47052</td>
<td>0.4542/0.47685</td>
</tr>
<tr>
<td>Net gain</td>
<td>-1087</td>
<td>-991</td>
<td>-1378</td>
<td>-568</td>
<td>37/-1005</td>
<td>-1088/-1579</td>
<td>655/-917</td>
<td>339/-1724</td>
</tr>
</tbody>
</table>

- Accuracy:
  - Weather features reduce prediction accuracy.
  - Win odds affects prediction accuracy unclearly.
  - Dividing the dataset does not affect prediction accuracy significantly.
Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Model 000</th>
<th>Model 001</th>
<th>Model 010</th>
<th>Model 011</th>
<th>Model 100</th>
<th>Model 101</th>
<th>Model 110</th>
<th>Model 111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>515.2</td>
<td>461.2</td>
<td>556.4</td>
<td>417.7</td>
<td>583/575</td>
<td>527/536</td>
<td>629/577</td>
<td>652/589</td>
</tr>
<tr>
<td>Accuracy _win</td>
<td>0.08367</td>
<td>0.07029</td>
<td>0.08090</td>
<td>0.10742</td>
<td>0.08798/0.08014</td>
<td>0.07725/0.07292</td>
<td>0.08155/0.09028</td>
<td>0.07940/0.06944</td>
</tr>
<tr>
<td>Accuracy _place</td>
<td>0.42926</td>
<td>0.41954</td>
<td>0.47547</td>
<td>0.47789</td>
<td>0.44277/0.43902</td>
<td>0.43419/0.46766</td>
<td>0.44778/0.47052</td>
<td>0.4542/0.47685</td>
</tr>
<tr>
<td>Net gain</td>
<td>-1087</td>
<td>-991</td>
<td>-1378</td>
<td>-568</td>
<td>37/1005</td>
<td>-1088/-1579</td>
<td>655/-917</td>
<td>339/-1724</td>
</tr>
</tbody>
</table>

- **Net gain:**
  - Weather features increase net gain this time.
  - No obvious patterns shown for win odds or dividing the data.
  - Races in Sha Tin are much more predictable than those in Happy Valley.
## Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Model 000</th>
<th>Model 001</th>
<th>Model 010</th>
<th>Model 011</th>
<th>Model 100</th>
<th>Model 101</th>
<th>Model 110</th>
<th>Model 111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>515.2</td>
<td>461.2</td>
<td>556.4</td>
<td>417.7</td>
<td>583/575</td>
<td>527/536</td>
<td>629/577</td>
<td>652/589</td>
</tr>
<tr>
<td>Accuracy_win</td>
<td>0.08367</td>
<td>0.07029</td>
<td>0.08090</td>
<td>0.10742</td>
<td>0.08798/0.08014</td>
<td>0.07725/0.07292</td>
<td>0.08155/0.09028</td>
<td>0.07940/0.06944</td>
</tr>
<tr>
<td>Accuracy_place</td>
<td>0.42926</td>
<td>0.41954</td>
<td>0.47547</td>
<td>0.47789</td>
<td>0.44277/0.43902</td>
<td>0.43419/0.46766</td>
<td>0.44778/0.47052</td>
<td>0.4542/0.47685</td>
</tr>
<tr>
<td>Net gain</td>
<td>-1087</td>
<td>-991</td>
<td>-1378</td>
<td>-568</td>
<td>37/-1005</td>
<td>-1088/-1579</td>
<td>655/-917</td>
<td>339/-1724</td>
</tr>
</tbody>
</table>

- Decrease in loss ≠ Increase in accuracy.
- Higher accuracy ≠ higher net gain (because of win odds).
- Net gain is low because we bet on all the horses the predictions suggest.
- To increase net gain, more strategies need to be applied.
Results

Figures

- Average Net gain:
  - Unvied: -1006
  - Shatin: -1306
  - Happy Valley: -14.25
Why?

- Using Loss to evaluate a model hardly works
  - Finishing time is predicted individually
  - yet grouped together in a race
  - Loss is too simple to model the prediction results

- Confidence/Trend matters
  - imply the relative horse performance
  - Help lessen being influenced by randomness
Bet on best predicted races
Bet on best predicted races

Net gain & Accuracy in different time intervals on training set (undivided)
Bet on best predicted races

Using strategy on test set (undivided)
Confidence

Combination of 2 models

- Average Net gain: -530.8
- Average Net gain (Previously): -1006

- Average Net gain: 325.32
- Average Net gain (Previously): -1306
Future outlook

- Explore the best way to predict the results
- Build a more solid regressor in use
Future Outlook
Directions In Progress

◉ Investigate in depth on the relations between Loss(MSE) and our goal.
  ○ Models trained with 1m steps. (Overfit, increasing loss)
  ○ Models with regularizations (e.g. dropout) to minimize MSE

◉ Use average finishing time to regularize finishing time in a race
  ○ Combine our understandings on horse racing and model design
  ○ Test error (MSE) ≈ 0.59
Future Outlook

Goal

- Build a more solid system
  - Maybe Shatin racecourse
  - Maybe average finishing time

- Deploy models to train on individual horse records
  - Similar to markov chain
  - Where future state depends on current state (& past in this case)
  - Inspired by Prof. Gu wengao in STAT department

- Try other bets
Summary
Horse racing prediction is not a traditional machine learning problem;
Loss, accuracy and net gain are less related to each other than we expected;
However, divide-and-conquer and apply the idea of confidence help improve the prediction.
Q & A