Basketball Analytics Pipeline---From Raw Video to Dynamic Visualization

Lukengu Tshiteya, Wenge Xie, Joe Zuo
Project Manager: Heather Mathews
Faculty Lead: Alexander Volfovsky

RHODES INFORMATION INITIATIVE AT DUKE UNIVERSITY

Abstract

Currently, data science is widely introduced in both NBA and NCAA as a new way to analyze tactics. The vast majority of the game's analytics evaluate the general offensive performance and defensive performance while individual behavior remains somehow entirely overlooked. Our project aims to analyze ball-holder's movements during an NCAA basketball game.
The dataset comes from SportVu (former NBA video tracking technology provider), including all of the game frames of 2014-2015 Duke Men's Basketball games (24 games in total). For initial data cleaning, we deleted several variables which are irrelevant to event prediction, including player names, jersey numbers, and game ID, from the original dataset. Each row in the dataset includes the absolute player and ball position information (in x-y coordinates), game clock, shot clock, and current event label. There are 7 different labels: dribble, pass, shot, touch, free throw, turnover, and rebound.

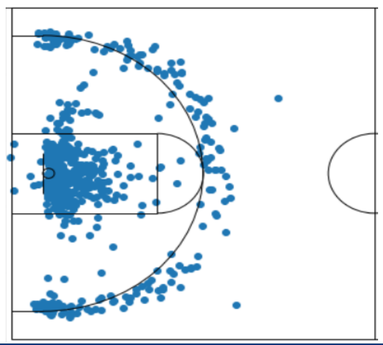

Fig.1 Distribution of Made Shots by Duke players in 2014-2015 season

Objectives

Our primary goal of this project is to establish a well-rounded prediction model of basketball player movements. To be specific, our task is divided into 4 parts:

1. Data cleaning and visualization of SportVu data.
2. Apply different classification methods for prediction
3. Use Sequential Neural Network and Recurrent Neural Network for the prediction model and evaluate their performances
4. Build an R Shiny App for interactive prediction

Methods

1. Data Cleaning ---- Deal with imbalanced data

- Class Setting

Set the label (class) of each row (moment) in the dataset to be the action the ball holder made in the next row.

- Normalization

Organize data into a related table; Also eliminate redundancy and increases the integrity which improves performance of the query

- Class merging

Merge 12 classes (in the origin dataset) into 6 classes to make different classes more distinguishable

- SMOTE (Synthetic Minority Over-sampling Technique)

An approach to over sample the minority classes (pass, touch) by creating "synthetic" examples.

2. Model building ---- Typical sequential NN and Recurrent NN

- 4-layer sequential Neural Network

The 49-sized input vector (including the action and position information of the players) will go through each layer in the neural network by inner production. All the parameters in this model will be adjusted well by back propagation during training.

Methods

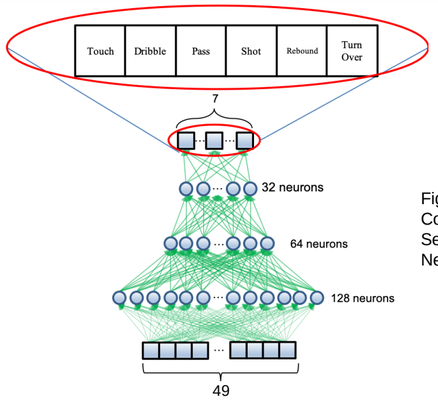

Fig.2 Construction of Sequential Neural Network

- 4-layer Recurrent Neural Network (Works better on Sequential data)

Connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs

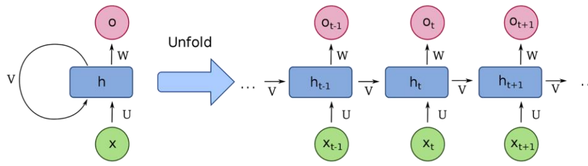

Fig.3 Construction of Recurrent Neural Network cell

3. Cross Validation ---- choose the best model

- Iteratively training the model on 19 datasets (games) and validate the model on 2 datasets. Test it on the left one

4. Visualization ---- Real time prediction video & Shiny App

- Real time prediction video

A video shows all the players position and ball holder's current action on the field and all the probability of different actions the ball holder may make as his next move

- Shiny App

The shiny app below visualizes a database of predictions from the model. The main takeaway is how fluctuating the game of basketball is and can be.
For instance, the frame displayed shows a varied distribution of probability among the event descriptions. We can deduce that the ball handler was in a triple threat position at the top of the key because that's an area where the most possibilities are available. Other frames may give us a glimpse into the offensive identity of the Duke 14-15 team at a particular moment in a game. Whether the offensive was free flowing and equal or stagnant and isolation heavy - typical of a half-court setting and older styles of play. The model with a bar for only touch, is likely an example of more traditional style of play.

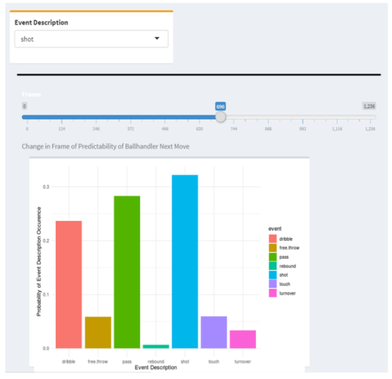

Fig.4 Frame screenshot from our Shiny App

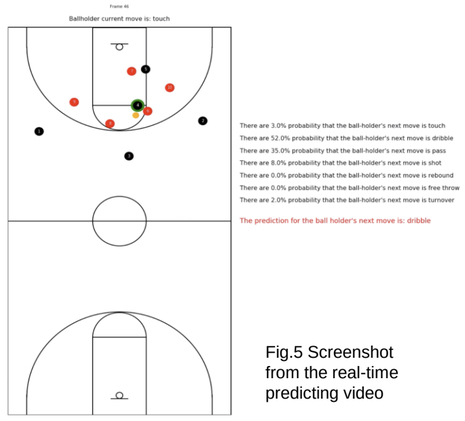

Fig.5 Screenshot from the real-time predicting video

Results

1. Testing Accuracy:

Testing accuracy of the typical sequential neural network: 75.85%

Testing accuracy of the Recurrent neural network: 76.51%

2. Error Analysis ---- See which class is the most difficult to predict

Error rate of different classes		
Action	Typical Sequential NN	Recurrent NN
Touch	0.05	0.07
Dribble	0.1	0.13
Pass	0.77	0.71
Shot	0.05	0.06
Rebound	0.01	0
Turn Over	0.02	0.02

Fig.6 Error analysis for both neural networks

3. Confusion Matrix ---- Judge the prediction performance

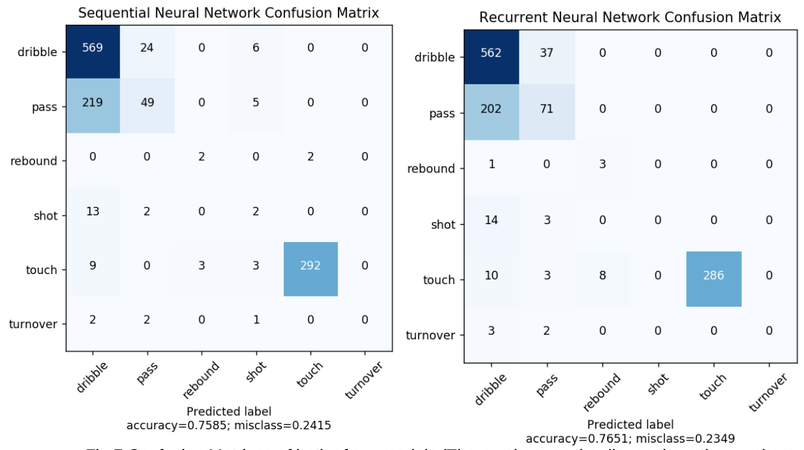

Fig.7 Confusion Matrices of both of our models (The numbers on the diagonal are the numbers of the correct predictions of the corresponding class)

Conclusion

1. Imbalanced data can make the prediction result very biased. In our model, more than 40% of the actions made by the ball holder are dribble. So, our model learned has the trend to predict other classes to be dribble more likely. Using normalization and SMOTE can decrease the influence brought by the imbalanced data.
2. RNN has better performance on sequential data. It improved the testing accuracy of our model from 75.85% to 76.51%. More importantly, RNN improved our model's performance on imbalanced dataset. It decreased the error rate of "pass" (the most imbalanced class in our dataset) from 77% to 71%.
3. The reason RNN did a better performance on the minority classes (pass) is because it sacrificed some of the prediction accuracy on the majority class (dribble). In the confusion matrix, the correct labelled "dribble" decreased from 569 to 562 while the correct labelled "pass" increased from 49 to 71.

References

1. Graves, Alex. Generating Sequences With Recurrent Neural Networks. arXiv:1308.0850

2. A Beginner's Guide to LSTMs and Recurrent Neural Networks, 2019, https://skymind.ai/wiki/lstm.

Acknowledgements

Special thanks to Dr. Alexander Volfovsky and Heather Mathews for the support you gave us. Also thanks to Dr. James Moody, Katherine Heller, Fan Bu, Heather Matthews, Harsh Parikh.