Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for an advanced degree from Emory University, I agree that the Library of the University shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this thesis may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate School when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this thesis which involves potential financial gain will not be allowed without written permission.

Chenxi Xu

December 5nd, 2020

Domain-Independent Sports Match Prediction using Dynamically Updated Database

By

Chenxi Xu

Jinho D. Choi

Advisor

Department of Computer Science

Jinho D. Choi

Advisor

Michelangelo Grigni

Committee Member

Yuanzhe Xi

Committee Member

2020

Domain-Independent Sports Match Prediction using Dynamically Updated Database

by

Chenxi Xu

Jinho D. Choi

Advisor

Abstract of

a thesis submitted to the Faculty of the Graduate School

of Emory University in partial fulfillment

of the requirements of the degree of

Bachelor of Science with Honors

Department of Computer Science

2020

Abstract

This thesis presents the design of sports match predictions with various approaches, including linear models based on different evaluation metrics of players' performance and neural networks. Our main goal is to incorporate this prediction model into the modern conversational AI and help the bot to make more interesting and human-like conversations. Therefore, our models have high interpretability to convey information about various aspects of a sports match. While our model can talk about players' performance of past matches, predictions of further matches and players' impact, it can still achieve an accuracy of 68.7% on the NBA and 67.5% on the MLB match predictions. Moreover, our model has a high generality and applicable to new sports. Hence, it will be easy for further development and expansion after being incorporated into the conversation system.

Domain-Independent Sports Match Prediction using Dynamically Updated Database

by

Chenxi Xu

Jinho D. Choi

Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences

of Emory University in partial fulfillment

of the requirements of the degree of

Bachelor of Science with Honors

Department of Computer Science

2020

Acknowledgements

First, I would like to give my thanks to my advisor Dr. Choi. I met with Dr. Choi when I was a sophomore and still a beginner in computer science. In the past three years, he taught me how and why to do research. Being the advisor of this honor program is also one of the epitomes of his continuous support for me.

In addition, I want to give my thanks to my committee members, not only for their suggestions for the honor program, but also their consistent support during my college time. Dr. Grigni was my coach for VTRMC and Putnam mathematical competition, and these competitions improved my problem solving skills. Dr. Xi was the instructor for the numerical analysis course and he brought me into the field of machine learning.

I also want to give my thanks to Ruixiang Qi, a student of class 2020. We developed the basic structure of the sports component and this became the fundamental of the sports prediction models.

Last but not least, I want to give my thanks to my parents for their support during my growth.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As one of the applications of the growing artificial intelligence technology, the conversational AI has become part of life for many people. Given the high popularity, many top tech companies are also designing their own chatbots, including Siri from Apple, Alexa from Amazon, Assistant from Google, etc. Chatbot is the name used by the most people, which suggests that the ultimate goal of such services is to be capable of chatting like a human. Generally speaking, a human-like conversation can be separated into several topics, and in each topic, people can share their known information, exchange their ideas, express their opinions, etc. However, given certain topics, it is always one of the challenging parts in the conversational AI to find interesting contents and continue the topic.

In this thesis, we will focus on sports related topics and use machine learning algorithms to analyze the matches and explore interesting contents behind them. Unlike other topics, sports matches are usually bound with large amounts of data such as players' box scores and teams' statistics, which provides the possibility of deeper analysis. Therefore, we will gather all the necessary information of players and matches in at least the past 5 seasons. These data will be stored in the database and updated daily automatically. Then by using appropriate algorithms to analyze these data, we want to set up a model to predict the results of future matches. More importantly, since our goal is to create an interesting conversation, we want the model to be highly interpretable, which includes giving out predictions of future matches, evaluating past matches, judging players' performance, etc. In addition, the prediction model will be integrated into a conversation system, which means all analysis or prediction requests are coming in real-time. It requires the model to be capable of handling various cases in a very short time. At last, we want our model to be general enough and applicable to multiple kinds of sports. In this thesis, we will mostly focus on the NBA and the MLB for more clear illustration purposes.

To begin with, Chapter 2 introduces some former work that this thesis is based on. We will show how our ideas are generated from these previous works. Chapter 3 explains the methods we use to set up the model and multiple evaluation metrics we use to analyze a player's performance. Chapter 4 shows the result of different sports returned by our model. In addition, we separate the dataset into different time periods for testing. Chapter 5 gives a more comprehensive analysis on how we interpret the results. In addition, we will explain how we will incorporate the result into a conversation and make it interesting. Chapter 6 is the conclusion and proposes some possible future works.

After all the work is done, we expect our models can adapt well to any conversation system and help the bot lead a more engaging and accurate conversation.

Chapter 2

Background

2.1 Sports Match Outcome Prediction

Sports match prediction is a traditional classification problem. Models will analyze the performance of players, two teams' statistics and past matches [1]. Although it seems straightforward to let the model learn how each feature will impact the match outcome, the number of categories of players' box scores is huge and diverse, and interactions between players in a match are complicated. In addition, a player's performance can be highly unpredictable, which adds more complexity for predicting the future matches.

2.1.1 Sports Performance Indicators

Given so many categories of box scores, understanding how much impact each category has on the outcome is meaningful. Mikolajec et al. [13] uses a regression model to analyze the weight of each box score category on the outcome of the match. They selected game statistics in multiple seasons and analyzed the correlation between match outcomes and box scores. They found that a match result is mostly determined by a limited number of factors. These factors, including percentage of win, offensive efficiency and 3rd quarter points, are usually on the offensive side. Similarly, Thabtah et al. [20] also analyzes which factor will influence the outcome of a match. However, they find that categories on the defensive side, such as defensive rebounds, have the most impact on the results.

However, the model evaluates the weight of each category based on the whole team, while in this thesis, we are focusing on the performance of individual players. In addition, we will use various methods to evaluate the performance of a player in each match. Then based on the performance we will make a prediction of a player's future performance. We will elaborate the details in section 3.3.1.

2.1.2 Statistical Model

Most approaches focus on setting up a traditional statistical model, such as logistic regression, SVM, etc. for match prediction. Haghighat et al. [4] did a comprehensive review on how these approaches perform on different sports.

Similarly, Cao [2] focuses on analyzing the performance of various models on the NBA match predictions. Among all the models, logistic regression can achieve the highest accuracy. However, Maszczyk et al. [10] conclude that a neural network approach can do better than a linear model on javelin throwers predictions. Given that javelin is very different from ball games, this might indicate that the performances of neural networks and simple statistical models are highly dependent on the type of sports.

All these approaches use one single prediction model. However, in this thesis, we want to retrieve more information about various aspects of the matches from the models. Therefore, we will separate the model into several components. An ensemble framework is also used in Min et al. [14]. However, while Min et al. want to gain a higher accuracy of using a compound framework, we hope while our model has better interpretability, it can still achieve at least the same prediction accuracy as previous work.

2.1.3 Neural Network Approach

Neural network approaches is also used for match outcome predictions. For example, multilayer perceptrons are used to evaluate the quality of a match [11, 12]. Rahman [19] passes players' and teams' information to a LSTM and uses it to predict football match results. As a result, the model can achieve 63.3% accuracy on FIFA world cup 2018 matches. Similarly, Nyquist and Pettersson [16] also propose a Recurrent Neural Networks (RNNs) approach to predict football matches.

In this thesis, we will focus on NBA and MLB matches. Instead of combining teams' and players' information together as the initial input, we will use CNN to process the players' box scores and then concatenate with teams' statistics. In addition, Rahman [19] suggests that with more prior match information of players and teams, it is highly possible to further boost the accuracy. Therefore, we will use past six matches in total to predict the result of a match.

2.2 Conversational AI

Conversational AI is a group of technologies including speech recognition, natural language understanding and generation. To enable computers to better interact with humans in a conversation, various kinds of conversational AI applications are available now. Each of them provides services on different fields, such as supporting more functions in smart phones and delivering information. Making a conversation human-like and interesting is their ultimate goal.

2.2.1 Alexa Prize

Alexa, developed and maintained by Amazon, is a conversational AI that powers the Amazon Echo. The Alexa Prize is a competition held by Amazon to further push the field of conversational AI by using Alexa as the medium[1]. The aim of the Alexa Prize is to design a social bot that can cover popular topics and chat about them with users. Each year, around ten universities are selected to participate in this competition and each university will implement their own version of conversational bots. The performance of a social bot is evaluated by the ratings given by the users after the conversations.

Fortunately, I was able to participate in the Alexa Prize Grand Challenge 3 as one member of the Emory University team, which was the champion of the competition. I helped to design Emora, the name of our social bot. Emora understands multiple topics including music, movies, holidays, etc. And sports, one of the major components in Emora, is the part that I mostly focused on. Together with my teammates, we scraped the sports data, stored the data into the database, designed sports related conversation flows and wrote dialog templates.

[1] https://developer.amazon.com/alexaprize/about

Limitation

In Alexa Prize Grand Challenge 3, Emora is based on the state machine approach [3]. Every turn of a conversation is represented by moving from one state to another. Therefore, the conversation flows are all pre-defined, and if one user talks about something not handled by a connected state, Emora will not know how to answer. In addition, all the responses are produced from manually generated templates, which requires much effort to develop.

Another limitation of template based conversations is that Emora can only provide some basic information which requires no further understanding of the data. In sports topic, when a user talks about a past match, Emora can give the fact of which team won or what were the final match scores, because these data can be directly retrieved from the database. However, it is impossible for Emora to analyze further and talk more beyond the existing data. For example, Emora cannot evaluate the players' performance or predict the outcomes of further matches. Here we list some examples of conversations with Emora about sports.

- Conversation 1: This is a typical conversation between Emora and a user about the NBA. The information of LeBron James has been manually made into the template, and all the information can be directly retrieved from the database. As we can see from the conversation, no further analysis is made when Emora is talking about LeBron James. After Emora introduces LeBron James, it will jump to the next topic about Emora's favorite player.

Emora: *I like watching NBA, do you watch NBA?*

User: *Yes.*

Emora: *I am glad we share the same interest. By the way, who is your favorite player?*

User: *LeBron James.*

Emora: *Oh, the King. LeBron James has won three championships and he is one the best players in the world. In this season, he has won 49 matches and his team Los Angeles Lakers ranks first. If NBA is not shut down due to coronavirus, I think he has a good chance to win. Guess who is my favorite player?*

- Conversation 2: Emora recognizes the user as the return user. However, Jamal Murray is not a prepared player and does not have a corresponding template. In this case, Emora just skips the discussion part about players and directly introduces its favorite player. In addition, when talking about Kawhi Leonard, Emora does not include any analysis but only the fact that Kawhi won the championship last year.

Emora: *Is LeBron James still your favorite player?*

User: *No.*

Emora: *Ok that is fine. Who is your favorite player now.*

User: *Jamal Murray.*

Emora: *My favorite player is Kawhi Leonard. I like him since he won the championship last year.*

Therefore, we want to incorporate a highly interpretable prediction model into the social bot. Then with the model, the bot can talk about something happening in real time, such as the matches yesterday. Everything should be also based on analysis instead of simple facts. This will make the bot have his own thoughts on the games or the players.

Chapter 3

Approach

3.1 Database Structure

Motivated by the database design of the previous work [18], we will continue to use the database to store the data. However, we will reduce the number of tables and ensure that the design can support more complex data searching. For example, the previous design separated the teams' scores and players' box scores into two tables. Then it will be hard to select the performance of one player in the consecutive matches played by the same two teams. In addition, we will store data in multiple seasons, and the database should be easy to be maintained in the future. A detailed description of columns is available in table 3.1. Note that all the approaches and models will only rely on this table.

The difference between the column *game ID* and the column *match*

Columns	Description	Type
Game ID	Unique ID for each game	Str
Date	The date of the match	Date
Match Number	Number that represents two teams	Int
Player Name	Name of player appeared in this match	Str
Team Info	Multiple columns includes name, scores, rank, number of matches won, etc	Str & Int
Opponent Info	Same as above except for the opponent	Str & Int
Player Box Score	Multiple columns includes number of assists, points, minutes played, etc	Int
Win/Lose	1 for Win, 0 for Lose	Int

Table 3.1: Table structure to store match data

number is that *game ID* is the combination of the date and the home team abbreviation, while *match number* is an unique number that represents two teams in this match. To be more specific, each NBA team is assigned with a prime number, and then each match number is calculated by multiplying two prime numbers of the teams. The reason is that during the data process stage, we need to look up the last match played by the same two teams as the current match. However, it does not work if we only compare teams' names since each match will have a home team and an away team. Therefore, we want to have this additional column to help us track those matches that are played by the same two teams.

3.1.1 Dynamically Update

Since all the sports data is stored in the database, it is easy to use a script to automatically maintain and update the database daily. During the NBA season, there can be up to around 10 matches every day. We represent each player of a match as one row. For each match, each team will have at least 8 players in its roster. At least 160 rows will be updated daily. While in the MLB season, around 15 matches are available every day. The number of participating players in MLB is larger than the NBA's, and usually over 350 rows are daily updated.

3.2 Baseline

The baseline approach is to simply compare the number of wins of each team. We predict that the team with more winning matches will win. This is the same as predicting the team with a higher rank will win. Though this is a naive approach, people may intuitively believe that this prediction is reasonable and promising in the real life.

3.3 Ensemble Learning Method

In this approach, we will separate our model into three parts. In the first part, we will use various methods to evaluate the players' performance in a match, which we refer as the performance score. In the second part, we will use the performance scores of past matches and the last match with the same two teams to predict the performance scores of future matches. After this step is done, for each future match, we can have an estimated performance score for each participating player. In the third part, we will use the predicted performance scores of each player to predict the match result.

When a linear method is required, we will use a square loss with regularization terms. To be more specific, the loss function is define as follow:

$$L = ||y - X\omega||_2^2 + \alpha \cdot ||\omega||_2^2 \tag{3.1}$$

This is well known as the Ridge regression [5]. In different cases where the targets and features are different, the value of α also varies. We will determine the optimal value of α according to the number of samples and the accuracy gap between the training set and the testing set.

3.3.1 Performance Score

Evaluation Metrics for Players' Performance

Various evaluation metrics of players' performance have already existed for different sports. For basketball, common evaluation metrics include Box Plus_Minus (BPM) [15], offensive rating, defensive rating [17], etc. In this thesis, we are going to use the Player Impact Estimate (PIE), invented by the NBA, to evaluate a player's performance in one match.

The reason why we choose PIE is that it includes most of the statistics of a player's box score and is able to comprehensively evaluate the player. In addition, built upon Player Efficiency Rating (PER) [6], another evaluation metric which focuses on the offensive categories of the box scores, PIE also incorporates the defensive part of the box scores with PER. Therefore, any player of either an offensive or a defensive role can receive a reasonable performance score.

For baseball, we use the Win Probability Added (WAP), which measures the change of percentage of win with each play made by the player, as the performance evaluation metrics. Since the goal of the performance score is to evaluate how much contribution players make to the teams, we believe that WAP provides a direct measurement for this.

The advantage of using an existing evaluation metric of player performance is that we can easily apply them on the box scores and get the performance scores. However, each sports has their own evaluation metric and the box scores of different sports usually vary a lot. In addition, there is no guarantee that available evaluation metrics of performance are suitable for prediction purposes. Therefore, it will be challenging to find a suitable evaluation metric for each kind of sports that we want to cover.

Ideal Cooperation Based Evaluation

Cooperation plays an important role in most sports. In this approach, we assume that when a team wins a match, all the players in the team perform well during the match. However, it might not be true in the reality. For example, in some matches in the NBA, maybe an exceptionally well played player who scores more than 50 points can directly lead the team to win. We want to compare the results under this assumption with other evaluation metrics of performance. Hence, we will use the match results as the target and the box scores as the features.

In addition, not only in basketball but also in many other sports, players have different roles and their own playing styles. Therefore, each of them is unique and needs to be treated separately. In the model, we will also divide

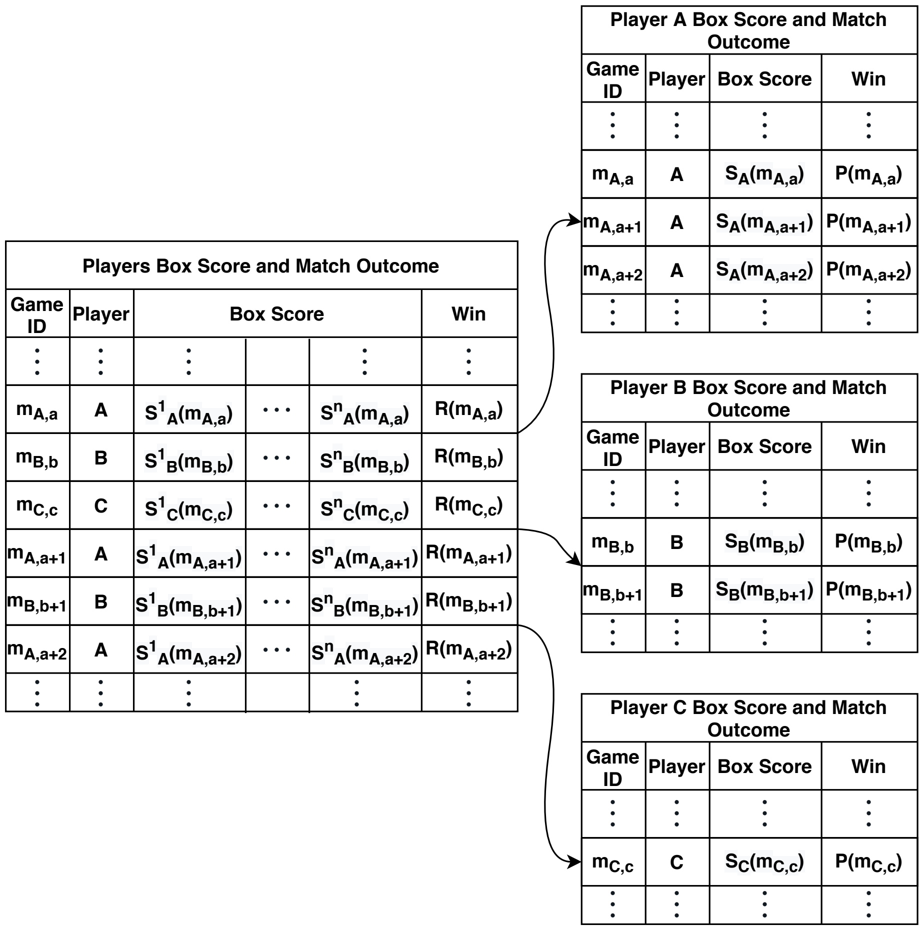

Figure 3.1: Row separation of ideal cooperation based evaluation

the box scores with respect to the player and train them separately. Then each player will only be compared to his own history but not to the others.

Suppose for a player i, we represent the series of matches he participates in as M_i. Then in the match $m_{i,j} \in M_i$ where j denotes the jth match in M_i, the player has his box score $S_i(m_{i,j})$. Every sports' box score contains various categories. For example, players' box score in the NBA includes assists, blocks, points, rebounds, etc. Therefore, $S_i(m_{i,j})$ is a vector and we denote $S_i^k(m_{i,j})$ as the kth entry in the vector, which is equivalent to the kth category in the box score. The game result is $R(m_{i,j})$ which is 1 if the team represented by the player i wins or 0 if the team loses. Our dataset contains all the matches in the past five seasons. Suppose player i has participated in n matches. Then the input for player i will be his box scores in all the participated matches, $S_i(m_{i,1}), S_i(m_{i,2}), S_i(m_{i,3}), \ldots S_i(m_{i,n})$ and the target will be the outcome of these matches, $R(m_{i,1}), R(m_{i,2}), R(m_{i,3}), \ldots R(m_{i,n})$. Then given with any future box score of this player $S_i(m_{i,j*})$, we can predict for the result of the match, $R(m_{i,j*})$, which is equivalent to predicting the player's performance, denoted as $P(m_{i,j*})$.

Group-wise Evaluation

In this approach, we also use the match outcomes as the target. However, instead of believing that every player of a team performs well in a winning match, we will group the players' box scores in one team and train them together. To be more specific, for the NBA, for each team in each match, we will select n players ordered by their points. If there is a tie, the number of assists will be used. For the MLB, we will order the players according to on-base plus slugging percentage, which evaluates the offensive skill of a player. Suppose box score contains d number of categories. Then, we can represent everything in a $n \times d$ matrix. We will use a 1 dimensional convolution and a max pooling layer to reduce the size of the matrix to $1 \times d$, and this is the representation of one team's box score. By adding another fully connected layer, we can get the weight of each category of the box score, which we will use to give a performance score for the individual player.

3.3.2 Future Performance Prediction

In section 3.3.1, we show various metrics based on the box scores to give a representation for each player in each game, and we refer it as the performance score. Then, given these scores from the past, we also want to predict a player's performance in the future.

A player's future performance score will be predicted by the player's performance scores of five past matches and one match whose teams are the same as the current match. To be more specific, suppose player i has participated in a series of matches represented by M_i. We want to predict the performance score of this player in one match $m_{i,j} \in M_i$ where j means the jth match in M_i. Meanwhile, $T(m_{i,j})$ means two teams of the match. Then

$P_i(m_{i,j})$ can be predicted by $P_i(m_{i,j-1})$, $P_i(m_{i,j-2})$, $P_i(m_{i,j-3})$, $P_i(m_{i,j-4})$, $P_i(m_{i,j-5})$, $P_i(m_{i,k})$ where $T(m_{i,k}) = T(m_{i,j})$.

3.3.3 Match Result Prediction

	Player Performance Scores										Team Statistics				
Game ID	$pl_{A,1}$	$pl_{A,2}$	$pl_{A,3}$	$pl_{B,1}$	$pl_{B,2}$	$pl_{B,3}$	$pl_{C,1}$	$pl_{C,2}$	$pl_{C,3}$		team rank	opponent rank	is home team		win
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
m^1	$P_{pl_{A,1}}(m^1)$	$P_{pl_{A,2}}(m^1)$	$P_{pl_{A,3}}(m^1)$	0	0	0	0	0	0	$\cdots$	$Rank(T_A)$	$Rank(T_B)$	1	$\cdots$	1
m^1	0	0	0	$P_{pl_{B,1}}(m^1)$	$P_{pl_{B,2}}(m^1)$	$P_{pl_{B,3}}(m^1)$	0	0	0	$\cdots$	$Rank(T_B)$	$Rank(T_A)$	0	$\cdots$	0
m^2	0	0	0	0	0	0	$P_{pl_{A,3}}(m^2)$	$P_{pl_{A,3}}(m^2)$	$P_{pl_{A,3}}(m^2)$	$\cdots$	$Rank(T_C)$	$Rank(T_A)$	1	$\cdots$	0
m^2	$P_{pl_{A,1}}(m^2)$	$P_{pl_{A,2}}(m^2)$	$P_{pl_{A,3}}(m^2)$	0	0	0	0	0	0	$\cdots$	$Rank(T_A)$	$Rank(T_C)$	0	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$

Figure 3.2: Matrix used to predict match result

The match result prediction is based on the players' performance prediction and two teams' statistics, such as whether the team is the home team or the away team, team's rank, etc. As shown in the figure 3.2, each game corresponds to two rows, with one row representing the home team and the other row representing the away team. Suppose match m^1 is played by team A and team B, match m^2 is played by team A and team C. $pl_{T,i}$ represents the ith player in team T. $P_i(j)$ represents the performance score of player i in match j. $Rank(T)$ represents the rank of team T. Team statistics includes the team's information as well as its opponent's. Note that the player performance columns contain all the players in the league. While each player coming from the team will be assigned with a performance score, the players in other teams including the opponent team will be assigned with 0. Therefore, the matrix will be very sparse.

The reason for putting all the players together is that we want to globally evaluate the players' impact throughout multiple seasons. In any kind of sports, players are frequently transferred to different teams in each season. Dividing all the players by the team and training them separately can only give the impact of a certain player to a certain team. For the next season, the player may go to another team, and we will lose all the information. In addition, the coefficient we get represents the overall impact of a player(C_i) to any team he has played in.

3.4 Deep Learning

We will also try to use the convolutional neural network (CNN) to make match outcomes predictions. CNN has been proven to be successful in multiple fields such as image classification [8], face recognition [9], natural language processing [7], etc. Therefore, we want to see whether CNN can recognize and distinguish winning team's box scores from losing one's. Different from a

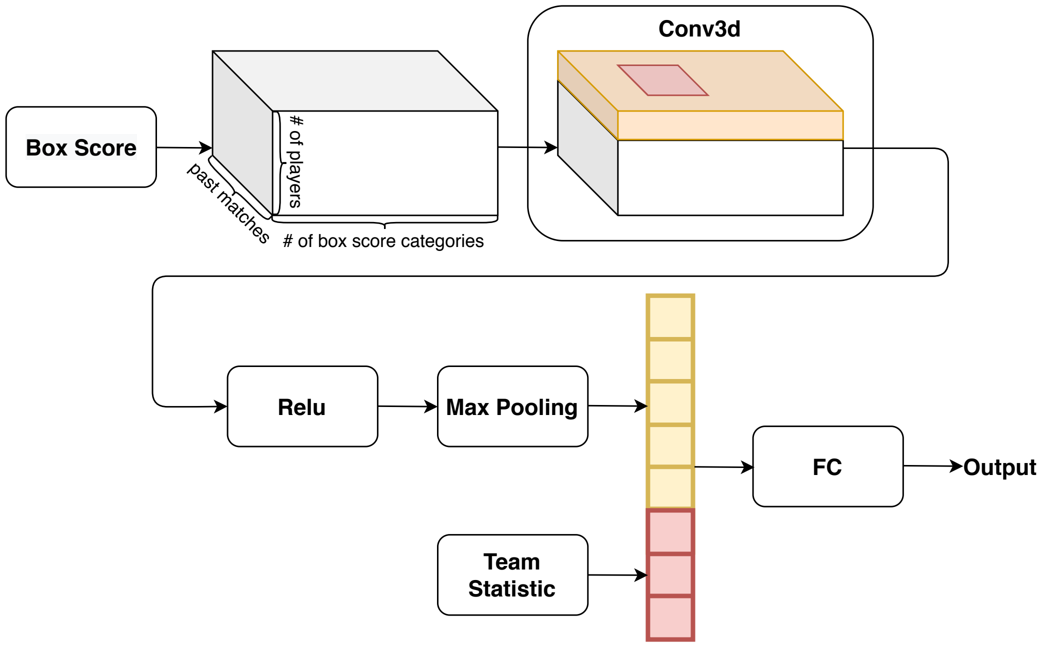

Figure 3.3: Architecture of NN model

linear model which consists of several steps before making the final prediction, a deep learning approach will take the players' box scores and teams' statistics as the input, and then return the prediction directly.

As shown in the figure 3.3, the first input contains one team's box scores for one match, and it is a three dimensional matrix. The height represents the number of players we choose to represent the team. It is ordered by the number of points and assists of the last match for the NBA or on-base plus slugging percentage for the MLB. The width represents the number of categories in box scores. In other words, each team in each match can be represented by a 2 dimensional matrix. Then, similar to the approach we use in section 3.3.2, the depth represents the past five matches and one match with the same teams as the current match.

We will use a 3 dimensional convolutional layer with the filter height set to 1. In other words, we apply filters on each horizontal piece of an input data. Then we will apply max pooling. Then, we will concatenate it with the team's statistics, such as team's rank, opponent's rank, is home team, etc. At last, we will use a fully connected layer to do the classification.

Our goal is to incorporate the prediction models into a conversational AI. Hence, a highly interpretable model is required. However, neural network models are sometimes harder to be interpreted than linear models, and this will be a factor when we decide which type of model is more suitable for conversation systems.

Chapter 4

Experiments

4.1 Dataset

NBA	Rows	Games	Players
Prior 2020 Season	107436	5249	583
Include 2020 Season	129581	6392	609
MLB	Rows	Games	Players
Prior 2020 Season	178209	7433	790
Include 2020 Season	198286	8394	790

Table 4.1: Dataset basic information

The data we use comes from a Python package *sportsreference*[1]. The package contains the statistics of matches, teams and players from various kinds of sports. Among them, we focus on the NBA and the MLB packages most.

The data retrieved from the package will be placed into the structure shown in the section 3.1. While the update happens once every day and new

[1]https://pypi.org/project/sportsreference/

matches' statistics will be inserted to the database, we also backfill the data from 2015-2016 season for the NBA and data from 2017 season for the MLB. However, a player's box score sometimes contains missing values. We will fill the values by the mean of the player's statistics from all time.

Table 4.1 shows the basic information of our dataset. We separate each kind of sports into two parts: prior 2020 season and include 2020 season due to the concerns of impact brought by Covid-19. To be specific, all matches in the 2020 season in NBA and MLB are removed in the prior 2020 season. Given that usually the total number of players is huge, we only select players that participate in at least 50 matches in NBA and 100 matches in MLB into the dataset. This explains why the number of players is the same for MLB because due to Covid-19, the 2020 season in MLB is shortened, and each team will only play 60 games instead of 162 against the other teams. Therefore, no new player in 2020 season will be selected. For each dataset, 80% of data is the training set while 20% of data is the test set.

4.2 Regularization Term

We add a regularization term to increase the generality and decrease the probability of overfitting in linear models. In addition, we find that the coeffi-

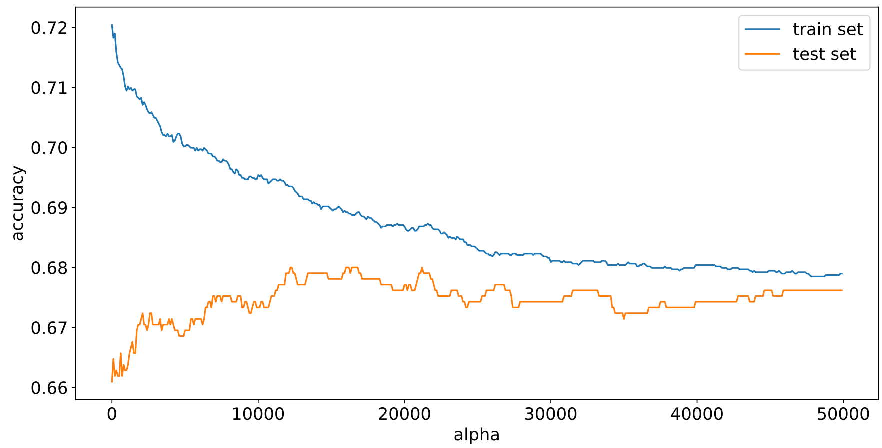

Figure 4.1: Accuracy of train and test set with respect to α

cients of teams' statistics are much larger than those of players' performance, which suggests that the former one are strong features for the match outcome prediction. Therefore, we also use the regularization term to balance the coefficients of teams and players. Shown in the figure 4.1, which is an example of using the PIE evaluation method for the NBA players' performance, we adjust the constant until the training set's accuracy is closed to the testing set's.

4.3 Experimental Result

We test various of our methods against the NBA and the MLB datasets in different period, which is shown in the table 4.2. Include '20 means the dataset contains the matches after the outbreak of Covid-19, while prior '20 means the matches during Covid-19 are ignored.

NBA	Team Rank	PIE	Ideal Cooperation	Group-wise	CNN
Prior'20	0.648	0.681	0.665	0.662	**0.687**
Include'20	0.635	0.651	0.650	0.636	**0.660**
% Change	-0.013	-0.03	-0.015	-0.026	-0.027
MLB	Team Rank	WAP	Ideal Cooperation	Group-wise	CNN
Prior'20	0.596	0.653	**0.675**	0.592	0.555
Include'20	0.576	0.638	**0.652**	0.582	0.542
% Change	-0.020	-0.015	-0.023	-0.010	-0.013

Table 4.2: Match outcome prediction accuracy using different methods

4.3.1 Impact of Covid-19

Covid-19 broke out at the end of 2019 which greatly influenced many sports games due to safety concerns. 2019-2020 season of the NBA is suspended half way and rest of the matches are delayed to July. Full 162-game season of the MLB is reduced to 60 games. This makes match outcomes become more unpredictable. In table 4.2, the first row shows the result after we remove the matches after the outbreak of Covid-19, while the second row includes all the matches. For all the prediction approaches we use, the accuracy of the NBA and the MLB drops when we include the matches in the pandemic period. Since the data in 2020 cannot represent the normal times, for the rest of the thesis, we will only focus on the matches prior to the outbreak of Covid-19.

4.3.2 Approaches Comparison

For the NBA, among the baseline, linear models and the neural network approach, CNN can achieve the highest accuracy. In three linear models based on different evaluation methods of players' performance, the linear model based on the official metrics PIE outperforms the other two, which suggests that PIE can best represent a player's performance according to his box score.

However, as another official metric for baseball, WAP cannot perform as well as PIE in basketball. The performance based on WAP is lower than ideal cooperation. Therefore, it suggests that not all the official evaluation metrics have the same capability for prediction. Group-wise and CNN approach have a much worse performance in baseball. While former one is close to the baseline, the latter method has a much lower accuracy than the baseline. We think the reason is that in baseball, players in each team is clearly divided into two groups, batters and pitchers. Players from different groups have completely different goals. Therefore, the categories of box scores for batters and pitchers are also different. Although in basketball, players also have different roles, they still share the same categories of box scores. In the group-wise approach, players from a team are grouped together to get the weight of each box score category, and in CNN approach, the input is also the concatenation of players' box scores of a team. Both methods do not have specific step to distinguish different roles in a team. However, the official evaluation metrics and ideal cooperation deal with each player's box score individually. Therefore, the problem of multiple roles will not have a huge impact on these models.

Chapter 5

Analysis and Interpretation

5.1 Player Performance Analysis

5.1.1 Box Score Category Weight

NBA Categories	Weight	MLB Categories	Weight
blocks	0.428	at bats	0.513
assist percentage	0.297	runs	0.374
two pointers	0.292	hits	0.292
field goals	0.291	on base plus slugging percentage	0.341
effective field goal percentage	0.287	earned runs allowed	0.326
...	...	...	...
personal fouls	-0.377	times struck out	-0.328
three point attempt rate	-0.379	bases on balls given	-0.353
steals	-0.395	putouts	-0.439
defensive rebounds	-0.406	assists	-0.465
free throw attempt rate	-0.419	runs batted in	-0.596

Table 5.1: Weight of NBA and MLB box score categories

In section 3.3.1, we discuss how to use 1 dimensional convolution layer to evaluate the weight of each category in the box score. Table 5.1 shows the top 5 categories that positively or negatively indicate the performance of a player. The group-wise method will use these weights to calculate the performance score.

From table 5.1, for the NBA, we can see that active actions such as blocks and those related to goals will be classified as the positive indicators for a player's performance, while those related to attempts and passive actions, such as defensive rebounds, will be classified as the negative indicators of the performance. However, it is interesting that steals have a large negative weight. It is possible that a large number of steals indicates a high ball possession of the opponent.

For the MLB, at bats, runs, hits, on base plus slugging percentage are related to offensive actions, while putouts and runs batted in are related to defensive actions. Therefore, categories of the offensive side are more likely to receive positive weights.

Note that the assist percentage in the NBA receives a large positive weight, but assists has a negative weight in the MLB matches. This is not a contradiction because in the NBA there is also a category in box score called assists. Although this is not shown in the table, assists in the NBA also receive a negative weight of -0.216. It indicates that the percentage of assists is a positive indicator of a good performance while the number of assists is a negative one.

5.1.2 Practical Usage

With the capability to give a performance score according to the box score, we can either judge whether a player is playing well in a past match, or predict the player's future performance with the past records.

For the past matches, we can compare the player's performance score with the mean value of the selective players and seasons. For example, we can compare the score with all players in all matches. Then we can know the performance standings in the whole league. Or we can compare the score with only the player himself in that particular season. Then we can know whether the player is making a progress. In addition, if we analyze the performance scores with the match outcomes, a high performance score with a losing match indicates that the player tries his best but his teammates might not play well. In contrast, a low performance score with a winning match indicates that his teammates are carrying the match.

5.2 Player and Team Impact

In table 5.2, we show the coefficients when predicting the outcome of a match in the linear model based on three evaluation metrics of performance in the NBA. The features include all the players in the league and teams' statistics. The weights of player features indicates the overall impact of the player to the match. If a player has a large positive weight, it means this player's performance has a large positive influence on the outcome of the match. In contrast, a negative weight indicates the player usually negatively affects the match outcome.

Models based on PIE and ideal cooperation have similar weights of players. Both models agree that Chirs Paul has the largest positive impact while Hollis Thompson has the largest negative impact. For the other outstanding players, both models include Stephen Curry, LeBron James and Klay Thompson. However, the weights returned from the model based on group-wise are very different from the other two. Chris Paul and Stephen Curry receive large negative weights.

In all the three models, the weights of teams' statistics are at least 4 times larger than the players', which indicates teams' statistics are strong features when predicting the outcomes of future matches. Therefore, it is reasonable

PIE		Ideal Cooperation		Group-wise	
Player Performance					
Player	Weight	Player	Weight	Player	Weight
Chris Paul	9.619e-3	Chris Paul	9.110e-3	David West	3.336e-3
Stephen Curry	9.116e-3	LeBron James	9.010e-3	Shaun Livingston	3.121e-3
LeBron James	8.650e-3	Stephen Curry	8.495e-3	Ian Clark	3.036e-3
Kevin Durant	6.519e-3	Klay Thompson	7.245e-3	Marquese Chriss	2.601e-3
Klay Thompson	6.473e-3	Draymond Green	5.853e-3	James Michael McAdoo	2.507e-3
...	...	...	...	...	...
Dillon Brooks	-5.899e-3	Dillon Brooks	-5.820e-3	Chris Paul	-3.705e-3
Ryan Kelly	-5.934e-3	Marcelo Huertas	-5.894e-3	Kevin Love	-3.812e-3
Michael Beasley	-6.353e-3	Mario Hezonja	-6.031e-3	Stephen Curry	-3.936e-3
Mario Hezonja	-6.611e-3	Jahlil Okafor	-6.335e-3	T.J. Warren	-4.311e-3
Hollis Thompson	-7.632e-3	Hollis Thompson	-7.915e-3	Draymond Green	-5.108e-3
Team Statistics					
Category	Weight	Category	Weight	Category	Weight
Team Rank	3.674e-02	Team Rank	3.666e-02	Team Rank	1.284e-02
Opponent Rank	-7.175e-03	Opponent Rank	-6.723e-03	Opponent Rank	-1.108e-02
Is Home Team	4.420e-02	Is Home Team	4.419e-02	Is Home Team	1.510e-02

Table 5.2: Player and team statistics impact in NBA

that many people use teams' ranks to make some simple predictions in the real life.

5.3 Possible Conversation

With all the information given by interpreting the models, it is now possible for the conversational AI to handle more interesting conversations. Below we list some sample conversations in different scenarios. Although we only use the NBA for a more clear illustration purpose, all the conversations can be easily applied to the MLB as well.

- Scenario 1: The user and the bot talk about a NBA match yesterday. LeBron James is one player of a team. According to our analysis, his performance score is higher than average performance score of all the players in the league but lower than the average of his own performance score in the season. Before the match began, we also made predictions about LeBron James' performance score and the match outcome. Since the match has already ended, we can make our predictions with the real results.

  ***User**: Did you watch the NBA game played by Los Angeles Lakers against Chicago Bulls?*

Bot: *Yes, Los Angeles Lakers beat Chicago Bulls and got their sixth win.*

User: *Do you think LeBron James did a great job in the match?*

Bot: *It depends. His performance yesterday was better than the average of players in the league. However, compared with his own performance in this season, he did not do well.*

User: *I agree with you. He was not doing as well as in previous matches.*

Bot: *Actually, before the match began, I predicted the performance of LeBron James. I thought he would do much better than he did. But I did predict that the Los Angeles Lakers would win correctly.*

User: *Then do you think he will still play bad next time?*

Bot: *According to my prediction, he will do much better next time.*

- Scenario 2: The user and the bot talk about players in the NBA in general. For each player the user talks about, we have the impact score. In addition, we set the player with the largest impact as the favorite player of the bot.

  User: *Do you like watching NBA games?*

  Bot: *Yes, I like watching it. What is your favorite player?*

User: *I like Stephen Curry.*

Bot: *He is a great player. From my analysis of the past five seasons, Stephen Curry has the second largest impact to the team on average. By the way, the player with the largest impact is Chris Paul and he is my favorite player.*

- Scenario 3: This is a typical conversation involved a match prediction. For each prediction, we have a confidence score which is the difference of the likelihood of each team winning the match. In addition, similar to the conversation above, we can evaluate a player by his impact in general.

  User: *Which team do you think will win the NBA match tomorrow, Orlando Magic or Oklahoma City Thunder?*

  Bot: *Oklahoma City Thunder, but I am not very sure about the prediction. Which team do you think will win?*

  User: *I think Orlando Magic will win. Aaron Gordon played really well in the last few matches.*

  Bot: *Yes, he did a good job. However, from the matches in the past seasons, on average his impact to the team is not very large.*

Chapter 6

Conclusion

In this thesis, we design prediction models for the NBA and the MLB sports matches using linear models and the neural network approach. Linear models have high interpretability and each component can convey some information which can be easily used by the bot to talk with users. The neural network approach for the NBA and the linear model based on ideal cooperation for the MLB can achieve highest prediction accuracy. In addition, the models also have a high generality and can be applied to any other sports as long as the players' box scores and teams' statistic can be put into the designed structure of the database. Compared with the previous social bot Emora based on the state machine approach, this sport prediction model can adapt to a more intelligent conversation system. In the future, while we will incorporate more sports into the models, we will also further boost the accuracy of the prediction without sacrificing interpretability.

Bibliography

[1] Rory P Bunker and Fadi Thabtah. A machine learning framework for sport result prediction. *Applied computing and informatics*, 15(1):27–33, 2019.

[2] Chenjie Cao. Sports data mining technology used in basketball outcome prediction. Master's thesis, Technological University Dublin, 2012.

[3] Sarah E Finch, James D Finch, Ali Ahmadvand, Xiangjue Dong, Ruixiang Qi, Harshita Sahijwani, Sergey Volokhin, Zihan Wang, Zihao Wang, Jinho D Choi, et al. Emora: An inquisitive social chatbot who cares for you. *arXiv preprint arXiv:2009.04617*, 2020.

[4] Maral Haghighat, Hamid Rastegari, and Nasim Nourafza. A review of data mining techniques for result prediction in sports. *Advances in Computer Science: an International Journal*, 2(5):7–12, 2013.

[5] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[6] John Hollinger. The player efficiency rating, 2009.

[7] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[9] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[10] Adam Maszczyk, Artur Golas, Przemyslaw Pietraszewski, Robert Roczniok, Adam Zajac, and Arkadiusz Stanula. Application of neural and regression models in sports results prediction. *Procedia-Soci Behavio Sci*, 117:482–487, 2014.

[11] Alan McCabe. An artificially intelligent sports tipper. In *Australian Joint Conference on Artificial Intelligence*, pages 718–718. Springer, 2002.

[12] Alan McCabe and Jarrod Trevathan. Artificial intelligence in sports prediction. In *Fifth International Conference on Information Technology: New Generations (itng 2008)*, pages 1194–1197. IEEE, 2008.

[13] Kazimierz Mikolajec, Adam Maszczyk, and Tomasz Zajac. Game indicators determining sports performance in the NBA. *Journal of human kinetics*, 37(1):145–151, 2013.

[14] Byungho Min, Jinhyuck Kim, Chongyoun Choe, Hyeonsang Eom, and RI Bob McKay. A compound framework for sports results prediction: A football case study. *Knowledge-Based Systems*, 21(7):551–562, 2008.

[15] Daniel Myers. About box plus/minus (bpm). `https://www.basketball-reference.com/about/bpm2.html`, February 2020.

[16] Robert Nyquist and Daniel Pettersson. Football match prediction using deep learning. Master's thesis, Chalmers University of Technology, 2017.

[17] Dean Oliver. *Basketball on paper: rules and tools for performance analysis*. Potomac Books, Inc., 2004.

[18] Ruixiang Qi. Analysis of a State Machine-based Interactive Dialogue Management System. In *Emory Theses and Dissertations*, Atlanta, GA, 2020.

[19] Md Ashiqur Rahman. A deep learning framework for football match prediction. *SN Applied Sciences*, 2(2):165, 2020.

[20] Fadi Thabtah, Li Zhang, and Neda Abdelhamid. NBA game result prediction using feature analysis and machine learning. *Annals of Data Science*, 6(1):103–116, 2019.