Using Machine Learning to Derive Insights from Sports Location Data

by

Joel Brooks

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

© Massachusetts Institute of Technology 2018. All rights reserved.

Signature redacted

Author .................................................................
Department of Electrical Engineering and Computer Science
April 2, 2018

Signature redacted

Certified by ..........................................................
John Guttag
Dugald C. Jackson Professor
Thesis Supervisor

Signature redacted

Accepted by .........................................................
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUN 18 2018
LIBRARIES
ARCHIVES
DISCLAIMER NOTICE

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available.

Thank you.

The images contained in this document are of the best quality available.
Using Machine Learning to Derive Insights from Sports Location Data

by

Joel Brooks

Submitted to the Department of Electrical Engineering and Computer Science on April 2, 2018, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

Historically, much of sports analytics has aimed to find relationships between discrete events and outcomes. The availability of high-resolution event location and tracking data has led to many new opportunities in sports research. However, it is often challenging to apply machine learning to understand a particular aspect of a sport. These tasks typically require learning on high-dimensional data, scarce labels, and multiple interacting agents.

In this thesis, we present applications of machine learning to derive insights from location data in soccer and basketball. In each case, we chose a data representation that allows the models to discover the importance of particular features and patterns.

We demonstrate how new quantitative metrics can be derived from predictive models. We built a model that uses the location of passes in soccer to predict shots with an AUROC of 0.79. From this model we defined a novel metric that can evaluate the value of any pass. We also trained a model for predicting shot quality using non-shooting player trajectories in basketball. This allows us to calculate offensive contributions by player movement alone.

We also developed an encoder-decoder architecture for learning a low-dimensional encoding of player and ball trajectories. When trained in an unsupervised setting, the model learned a representation that lends itself well to possession querying and clustering. We show that these clusters characterize different team styles on offense. Additionally, we trained this same architecture in a semi-supervised setting for set play classification. Compared to only training on labeled data, this framework improved six-way classification accuracy from 69% to 78%.

While we chose specific applications of machine learning to sports data for this thesis, the methods described could generalize to other sports-related tasks or other real-world domains with tracking or multi-agent data.

Thesis Supervisor: John Guttag
Title: Dugald C. Jackson Professor
Acknowledgments

Throughout my time at MIT I have had the privilege of meeting, collaborating, and befriending many brilliant individuals. It is with their help and support that this thesis has become possible.

I could not have asked for a better advisor in Professor John Guttag. John never ceases to amaze me with his breadth and depth of knowledge in the field, and his ability to shape difficult problems into tangible solutions. He is a true advocate for each of his students, and I am extremely grateful for his patient guidance throughout my PhD experience.

I would also like to thank the other members of my committee. Professor Jenna Wiens was a crucial mentor during my first few years as a graduate student as a labmate, and now as a committee member she continues to be a helpful ally. Professor Fredo Durand has graciously offered his expertise and unique perspective, giving invaluable feedback and advice.

Many others have helped me along my path. Zeeshan Syed was an inspiration during my early foray into research. My fellow labmates have been a constant source of feedback and friendship. Garthee, Anima, Yun, Guha, Jen, Amy, Davis, Maggie, Harini, Jose, Adrian, Wayne, Divya, and others have always been the first source of advice I have sought out when I have been stuck. I would like to thank Matthew Kerr and Akhil Nistala for collaborating closely with me on a number of projects. Sheila Marian, Dorothy Curtis, and others on G9 have provided invaluable support throughout the years. I would also like to thank the National Basketball Association for providing access to their player tracking data.

MIT is truly a one-of-a-kind institution, and I believe that the community is what really sets it apart. I feel extremely lucky to have met so many wonderful friends during my time here. Some I’ve met through department, others through ultimate frisbee and climbing, all who I deeply admire and respect. I would like to thank Alex, Parker, and Michael in particular for their frequent lunchtime companionship.

I am forever grateful for the love and support from my family. Throughout my entire life, my parents and my sister have been there to encourage me. I could not have made it this far without them. Finally, I would like to thank my best friend, pillar of strength, and wonderful wife, Alisha. Finishing this journey is so much more fulfilling knowing that you will be there for whatever the future holds.
## Contents

1 Introduction 17

2 Developing a Data-Driven Pass Metric in Soccer 23
   2.1 Introduction ........................................ 23
   2.2 Related Work ........................................ 24
   2.3 The Data ............................................ 26
      2.3.1 Data Overview .................................... 26
      2.3.2 Pass Location Representation .................... 27
   2.4 Methods ............................................. 28
      2.4.1 Feature Extraction ................................ 28
      2.4.2 Experimental Design and Testing ................. 30
   2.5 Results ............................................. 30
      2.5.1 Classification Results ........................... 30
      2.5.2 Player Rankings by Shot Prediction Models ...... 32
   2.6 Conclusion ......................................... 36

3 Non-shooting Player Movement Contributions on Offense in Basketball 39
   3.1 Introduction ......................................... 39
   3.2 Related Work ........................................ 40
   3.3 Description of the NBA Player Tracking Data ........ 41
      3.3.1 Data Overview .................................... 41
   3.4 Methods ............................................. 43
3.4.1 Possession Quality ............................................ 43
3.4.2 Data Representation ........................................... 46
3.4.3 Model Architecture and Training .......................... 48
3.4.4 Experimental Design and Testing ......................... 50
3.5 Classification Results ......................................... 50
3.6 Player Contribution Metrics ................................. 51
3.7 Conclusion ....................................................... 55

4 Unsupervised Exploration of Basketball Possessions 57
4.1 Introduction ....................................................... 57
4.2 Related Work ..................................................... 58
4.3 Methods ......................................................... 59
  4.3.1 Data Representation ........................................ 59
  4.3.2 Auto-encoder Architecture ................................. 60
  4.3.3 Experimental Design and Testing ....................... 64
4.4 Reconstruction Results ......................................... 64
4.5 Possession Encoding Analysis .............................. 64
  4.5.1 Possession Querying ....................................... 66
  4.5.2 Possession Clustering ..................................... 67
  4.5.3 Team Profiles .............................................. 70
4.6 Conclusion ....................................................... 72

5 A Semi-supervised Framework for Basketball Play Recognition 77
5.1 Introduction ....................................................... 77
5.2 Related Work ..................................................... 78
5.3 Data Overview ................................................... 79
5.4 Methods ........................................................ 80
  5.4.1 Semi-supervised Model Architecture ................... 80
  5.4.2 Supervised Model .......................................... 83
  5.4.3 Auto-encoder Model ....................................... 83
  5.4.4 Experimental Design and Testing ....................... 83
5.5 Classification Results ........................................... 84
5.6 Discussion ......................................................... 85
  5.6.1 Classification .................................................. 85
  5.6.2 Play Querying ................................................. 86
5.7 Conclusion ......................................................... 88

6 Summary and Conclusion ........................................ 91
List of Figures

2-1 The playing area split up into 18 zones. The left side of the field (zones 1-3) is the defensive side of the field, and the right side (zones 16-18) is the offensive area. ................................. 27

2-2 ROC Curve for the Shot Prediction Model. Our model predicts whether a possession will end in a shot with an AUROC of 0.79. ....................... 31

2-3 Top 10 most influential features for predicting when a possession will end in a shot. All features are normalized by the total sum of absolute weights. Each number corresponds to a zone. “DT” designates a pass destination feature, and features labeled with two zones, “Zone₁ - Zone₂,” designate an origin-destination feature from Zone₁ to Zone₂. 32

2-4 APSV of passes for all players with more than 200 passes in the 2012-13 La Liga season. Each item in the plot represents the APSV of a single player. Players have an associated color and shape chosen by their position. The red line represents the APSV across the entire collection of players. ................................. 34

3-1 An illustration of the camera-based player tracking system. Three cameras mounted above each end of the basketball court track the locations of the players on offense (blue), players on defense (green), and the ball (orange). ................................. 42
3-2 An example player's trajectory during a possession after converting to our $25 \times 25 \times 2$ pixel image representation. We generate an image for each non-shooting player on offense during the possession. The trajectory is blue when the player was moving without the ball, and red when they had the ball. The "fading" employed by our image representation provides information about directionality of the player's movement as well as their speed at different points along the trajectory. The court lines are displayed for visualization purposes, and are not included as part of the input to the model.

3-3 The general layout of the shared-weight model architecture. Player images are fed through subnetworks with shared weights. Each network produces a single output value, which is summed before sigmoid activation, producing the final output.

3-4 The architecture of the player subnetworks. The input is passed through 3 successive layers of 2D convolutions and max-pooling, before a dense layer to produce a single output.

3-5 AUROC for predicting shot quality vs. various input time windows. Performance increases if more than just the of the end of the possession is used, suggesting that it is important to account for non-shooting player movements and location that happen well before the shot.

4-1 A single frame of a possession in our video-based representation. The frame consists of a three channel image, corresponding to the players on offense, the players on defense, and the ball. It indicates the trajectories during a half second segment of the possession. The court lines are shown for visualization purposes and are not included in the actual representation.

4-2 Overview of encoder/decoder auto-encoder model architecture.
4-3 Overview of the architecture of the encoder layers. The encoder consists of convolutional layers (Conv-E) followed by an LSTM (LSTM-E) to convert the input sequence into a low-dimensional encoding.

4-4 Breakdown of the convolutional layers. Each arrow represents a convolutional layer, followed by ReLu activation and 2D max-pooling. Each convolutional layer consists of 64 size 3 x 3 filters.

4-5 Overview of the architecture of the decoder layers. The decoder consists of an LSTM (LSTM-D) followed by convolutional layers (Conv-D) to reconstruct the input.

4-6 Two examples of input frames and their corresponding reconstructions by the auto-encoder. The reconstructions are blurry in comparison to the original trajectories of the players and the ball at each frame. This blurriness indicates that the encoding does not exactly memorize the original locations, but preserves the general areas that are reached in a given time frame. The court lines are shown for visualization purposes.

4-7 Two examples of query possessions and their corresponding closest neighbor by both encoding distance and distance in the raw data space. Both methods find possessions that are similar to the query possession. Only the players on offense (blue) and the ball (red) trajectories are shown.

4-8 Three examples of possessions that were close to the center of cluster 24. In all possessions the offense uses an off ball screen in the post to get a player open on the perimeter. Only the players on offense (blue) and the ball (red) trajectories are visualized for clarity.

4-9 Cluster profiles for the Cleveland Cavaliers and the San Antonio Spurs during the 2015-2016 season. The charts highlight the difference in distribution of offensive possessions executed by each team.
4-10 Cluster profiles for the Oklahoma City Thunder and the Golden State Warriors during the 2014-2015 and the 2015-2016 seasons. The charts display how OKCs injuries and coaching change led to differences in their offensive style across seasons. Golden State was much more consistent between the two years.

4-11 Cluster profiles for the Cleveland Cavaliers for the 2015-2016 during the each quarter. The profiles indicate a noticeable change in strategy in different quarters.

5-1 Diagram of an example set play. Solid lines indicate player movement. Dashed lines indicate a pass. The \( \perp \) symbol represents a screen.

5-2 Overview of encoder/decoder auto-encoder model architecture. This framework is nearly identical to the auto-encoder used in Chapter 4, but it also includes a pathway for predicting a play class from the encoding layer.

5-3 Overview of the architecture of the encoder layers. The encoder consists of convolutional layers (Conv-E) followed by an LSTM (LSTM-E) to convert the input sequence into a low-dimensional encoding. The pathway for the supervised learning objective (represented by the dashed lines) is only considered when training with labeled batches.

5-4 A visualization of a ground truth example of plays 1 and 5, and similarly run plays in the unlabeled data found by our model. The black circles denote the location of each player on offense.
List of Tables

2.1 Teams in the La Liga 2012-13 season and their number of shots attempted. Teams are ranked by their final standing in the league at the season end. .................................................. 26

2.2 Top 10 players in La Liga 2012-13 by APSV. We separate the positions into three categories: Offense, Midfield, and Defense. ................. 35

2.3 Top 10 players in La Liga 2012-13 by Goals and Assists ................. 35

3.1 A short sample of the EVENTS data. Events are recorded sporadically as they occur throughout the game. We primarily use them to mark the beginning and end of possessions, and for identifying possession results such as shots. ........................................... 43

3.2 A short sample of the POSITIONS data. Positions for all players and the ball are recorded at a sampling rate of 25Hz. The ball’s z coordinate is tracked in addition to its x and y coordinates. ...................... 44

3.3 The shooter history for Kevin Durant. This lookup table indicates Durant’s likelihood of making a shot at different shot ranges, with and without having a defender within 3 feet. ............................... 45

3.4 AUROC for our model compared to a two baselines. 95% confidence intervals were found through bootstrapping the test predictions. The first baseline is the same model, but only contains trajectories from the last second before the shot. The second contains the same inputs, but uses linear subnetworks. .............................................. 50

3.5 Top 25 players by PCV in the 2015-2016 NBA season ................. 53
3.6 Leaders in points and assists per game during the 2015-2016 NBA season. There is some overlap with the leaders in PCV and QCV, but some of the most dominant scorers, such as James Harden, are not rated highly by PCV or QCV.

3.7 Top 25 players by QCV in the 2015-2016 NBA season

4.1 Description of the offensive motifs seen within each cluster

5.1 The number of examples for each play class. Each of these examples was labeled by an expert to be an instance of one of six plays.

5.2 Accuracy for the models tested. The semi-supervised model significantly outperforms the supervised and auto-encoder models on our 6-way play classification task.

5.3 Confusion matrices for the supervised and semi-supervised models. A large part of the semi-supervised models increase in performance over the supervised model comes from a greater accuracy classifying Play 6.
Chapter 1

Introduction

Sports analytics is a rapidly growing industry, accelerating from a market size of $125 million in 2014, to a projected $4.7 billion in 2021 [61]. While much of this increased interest has been sparked by success stories like the 2003 Oakland Athletics [33], the concept of using data to drive sports strategy and decision making is not new. Sports enthusiasts and teams have been analyzing simple statistics, such as the counts of various events that occurred during the game, for well over a century.

In the 1850’s, Henry Chadwick developed the “box score,” the first united system of keeping baseball statistics [58]. Similar systems would soon spread to other sports, leading to media, teams, and fans alike being fascinated with summarizing what occurred during a game. However, it wasn’t until the 20th century that people began thinking more critically about how to utilize the statistics to provide insights about sports. Beginning in the 1950’s, Charles Reep started recording passes that occurred during a soccer game. He analyzed how the number of passes in a soccer possession related to goals, concluding that an “optimal” possession had three or fewer passes [56]. This investigation is commonly credited with being the statistical basis for the popular “English Longball” strategy. Others in this era tried to formally describe how the numbers had an effect on the game [58, 18].

Later, the sports community began to combine statistics and expert knowledge to move beyond direct relationships between counts of events and outcomes. With the advent of computers, a new trend of computing more advanced statistics arose,
especially in baseball’s “sabermetrics” community. More complicated player metrics were developed that combined a number of box-score statistics to quantify a player’s overall contribution to a team’s success. Two examples of such metrics are Wins Above Replacement (WAR) in baseball, and the Player Efficiency Rating (PER) in basketball [70, 9]. Both of these metrics take into account a number of statistics about players, and combine them using expert knowledge. By using a combination of domain knowledge and experimentation, statisticians were able to create metrics that took many player and/or team statistics into account all at once. These statistics gave a more complete picture of an individual or team than box score statistics alone.

More recently, researchers have focused on building data-driven methods for understanding relationships among various elements of a game. Much of the early work investigated how various statistics related to concrete outcomes such as wins or points [17, 16]. For soccer and hockey, researchers have worked to understand how characteristics of possessions are important for scoring goals [4, 39, 55]. Instead of expert-driven metrics like WAR and PER, the data-driven tools allow the data to “speak for themselves.” Instead of experts balancing the various aspects that go into the metrics, models try to find the optimal balance of the various statistics for predicting an outcome of interest.

As the data that was being collected evolved, so did the ideas of how to utilize them. Now there is rich location data being collected for nearly every popular professional team sport. The National Basketball Association has utilized camera-based tracking systems since 2013 [64]. Major League Baseball added Statcast in 2015, a system that not only tracks every player but detailed information about the ball as well [63]. The National Football League added a system in 2016 that uses sensors to track the players and the ball [73].

These data allow for quantitative analysis of previously incalculable elements of sports. Many researchers began to focus on the movements of players instead of the events they were involved in [6, 52, 35]. For example, models in both the soccer and basketball domains extend previous expected goal models to include context based on how players were moving during an offensive attack [38, 10, 37]. In addition,
researchers could focus more on defensive contributions of players, looking at metrics based on the space they occupy instead of box-score statistics like steals or blocks [22]. Others focused on more subtle analysis such as movement characteristics of team styles in soccer [5, 24] and effects of player movement on rebounding in basketball [40, 71, 11].

Such analyses help teams, researchers, and media better understand what occurred over the course of a short period of play, a game, or a season. Some provide quantitative evidence for intuition. The work in this thesis is primarily focused on discovery of new knowledge: patterns of play, player contributions, and other subtleties that were not obvious even to expert observers. Finding these insights could help drive strategy for teams in ways that will provide an “edge” over others. They could also help better evaluate players for more quantitative based scouting.

While the location-based data provides new opportunities for finding insights in sports, there are many challenges involved in using these data:

• **Missing information:** The locations alone do not provide a complete picture of what occurred. Some datasets only record the locations of events (passes, shots, goals, etc.). Any actions outside these recorded events taken by the players either need to be inferred or cannot be taken into consideration. Even the player tracking systems have limitations on what they can capture. Most systems give a single x and y coordinate for each player. Information about the location of limbs or orientation of players is often not included. This lack of information could make some tasks of interest difficult or infeasible.

• **Non-intuitive data representations:** Understanding how to represent the raw data in a useful format is non-obvious. The tracking systems are often precise to within a few inches. This leads to millions of possible locations for each player. The playing area is usually discretized to reduce this number, but the appropriate granularity is highly dependent on sport, area of the field, or game context. Features can be engineered and extracted from the raw location data, but it is not clear if those features will provide all of the information
necessary for the task of interest. Additionally, extracting features may introduce unwanted bias in the type of patterns that may be discovered. Many have discarded feature engineering for feature discovery using the power of neural networks. However, just getting the high-dimensional, multi-agent data into an appropriate input format itself is a challenge.

- **Defining learning objective:** It is not straightforward to determine which outcomes or relationships are most interesting or useful to discover. For example, one might be interested in using the data to evaluate possessions on offense. One way to approach this problem is to build a predictive model that takes in location data about the possession, and tries to predict possession success. But what is your measure of possession success? Goals or points may not be good measures, since they could be dependent on many factors that are independent of the actions taken by the players on offense. One could use shots instead, but for sports like basketball, a shot happens with nearly every possession. Properly defining objective for any models using the data can have a profound effect on how models can be useful or actionable.

- **Label scarcity:** Data is often plentiful, but labels may not be. Using supervised learning often requires not only a large amount of data, but associated labels with those data. However, labels for certain tasks of interest can often be difficult to acquire. The tracking systems have automatically recorded hundreds of thousands of demonstrations of offensive possessions. Annotating these possessions with abstract concepts such as set plays requires manual review of each possession by an expert for that particular sport. The need for expert knowledge or tedious hand-labeling can make it infeasible to obtain more than a small set of labeled examples for a given task.

- **Actionable analyses:** Even if models can successfully discovery patterns and understand the relationship between the data and aspects of the game, these models must be utilized in actionable ways. Teams may not get much use out of predictive models alone. There must also be a pathway for using the information
found in the model to insights that are useful to teams and other stakeholders.

In this thesis, we explore different methodologies for using machine learning for knowledge discovery in sports data. The central focus of our work is using the locations of players and events instead of box score statistics. Additionally, do not engineer features or use models that require an extensive amount of expert domain knowledge. Our analyses and models are largely data-driven; we try to let the models decide what features, patterns, and relationships in the data are the most important.

We also focus on applications and tasks that are actionable. In addition to training models for understanding the relationship between patterns of locations in sports and various outcomes, we demonstrate how these models can be utilized. This lays the framework for future work on using models to both understand the data being used and to use that understanding as a guide for strategy and decision making.

Specifically, the contributions of this thesis are:

- **A location-based quantitative measurement for the value of soccer passes:** We train a model to understand the relationship between pass locations and shots in possessions. This model predicts pass locations with an AUROC of 0.79. Based on the model weights, we develop a metric for assessing the offensive value of any given pass based on the passes origin and destination.

- **A data-driven player ranking based on passes:** We use our pass metric to rank soccer players. We show that this ranking corresponds well with traditional notions of players that contribute positively on the offensive end for the premier Spanish professional soccer league.

- **Evaluating possession quality in basketball:** We relate the non-shooting player paths taken on offense to the resulting shot quality. We build a model that measures each non-shooting player’s contribution to overall possession success. We use this measurement to build two new player metrics based on player movements.

- **An unsupervised analysis of offensive possessions in basketball.** We train an
auto-encoder to learn a low-dimensional encoding of basketball possessions. We cluster the possessions using our learned encodings. This clustering leads to the discovery of a number of different offensive motifs. We show how teams can be profiled by their tendencies to utilize these different motifs, and how those profiles change based on personnel, coaching, seasons, and different game situations.

- *A semi-supervised neural network approach to play classification in basketball:* We train a semi-supervised neural network model that jointly optimizes unsupervised and supervised learning objectives. This model leads to better performance on play classification than models trained with expert-engineered features, classifying plays with 78% accuracy on a 6-way classification task.

The contributions laid out in this thesis are all applied in the sports domain. However, many of the results have potential applications outside of the sports world. The datasets used in this thesis are high-dimensional, sparse, and noisy, like many real-world datasets. Many of the methods used to build machine learning models using these data could be applicable to other datasets as well. This is especially true for other data taken from multi-agent systems, such as GPS data from a fleet of autonomous vehicles. Additionally, the demonstration of semi-supervised learning could help drive other work in domains where data is plentiful, but labels are not.

The rest of thesis is organized as follows. Chapter 2 describes our work on building a data-driven pass metric in soccer. In Chapter 3, we detail our work on building a model for qualitatively evaluating basketball possessions from player movement. In Chapter 4, we demonstrate an unsupervised exploration of offensive possession motifs in basketball. In Chapter 5, we describe the semi-supervised methods for utilizing unlabeled data for improving play prediction in basketball. Finally, in Chapter 6, we summarize the contributions of this thesis, and discuss possible avenues of future work in using machine learning to discover interesting patterns in sports location data.
Chapter 2

Developing a Data-Driven Pass Metric in Soccer

2.1 Introduction

Although soccer is by far the world’s most popular sport [53], published work in soccer analytics has yet to achieve the same level of sophistication as analytics being performed in other professional sports. Crude summary statistics such as goals, shots, and assists are still the most common way to compare player performance analytically. More work is emerging [1] that leverages the rich datasets available to make discoveries about soccer, but there has not been much focus on quantitative metrics for evaluating player performance.

We developed a novel way of quantitatively measuring a player’s passing performance using existing data. We chose to focus on passing because it is one of the more strategic elements of soccer. Currently, players are often considered good passers if they accumulate many assists. Assists identify when a player makes a pass that directly leads to a goal, but this measure alone is quite limited. For example, assists do not capture passes that would have been assists except for an errant shot, or an excellent save by the opposing team’s goalkeeper. Opta [50] extends the idea of assists to include all passes that lead to shots (whether or not they lead to a goal) in their “key passes” metric, but both this metric and assists are only applicable to passes
immediately preceding a shot. There may be players who are excellent passers that create many opportunities for their team, but rarely make the last pass before a shot or goal.

Instead, we present a method for quantitatively measuring the importance of any pass. We accomplish this by first training a classifier that uses information about the locations of a set of passes to identify when that group of passes results in a shot. Since we use a linear classifier, we can directly utilize the model weights to understand the relative importance of pass origins and destinations for generating shot opportunities. These weights allow us to compute an estimated value of any pass for creating shots. We can then rank players by the value of the passes they complete over the course of a season.

In this chapter, we use data from the 2012-2013 La Liga season. The contributions of this chapter are:

1. A model relating pass origins and destinations during a possession to the probability of a shot. This model accurately identifies whether a possession ends in a shot from the pass locations alone.

2. A demonstration how the resulting model weights offer insights into the offensive utility of passes.

3. A ranking of players by the frequency with which their passes are highly valued by the model.

2.2 Related Work

Much of soccer analytics has focused on building probabilistic models to simulate game actions and predict the outcomes of matches or goals scored. Reep and Benjamin developed models for the success rates of different length passing sequences [56]. More recently, there has been work on predicting the outcomes of matches by using possession rates of different teams and other historical statistics to develop probabilistic models [16, 17]. Other work has identified a relationship between a goal
being scored and the frequency of passes in the 5 minutes preceding that goal [55]. In contrast, our work focuses on predicting shots taken in possessions as opposed to the outcome of games or goals. Additionally, we focus on how these predictions can be useful for ranking players in a quantitative fashion, as opposed to the performance of the predictors themselves.

An increasing amount of spatiotemporal data for soccer is allowing analysts to study the underlying mechanics of the game in a manner that would not be possible with box score statistics alone. Bloomfield et al. used player tracking information to investigate the physical demands of soccer and the work rates of different players [6]. Gyarmati et al. leveraged ball-event data and passing sequences to cluster the playing styles of different teams [24]. Lucey et al. used ball-event data to infer the location of the ball throughout a game. Using this information, they constructed “entropy-maps” to characterize how different teams move the ball during a match [35]. In more recent work, the authors combine match statistics, event data and player tracking data to identify the teams playing in a given game with 70% accuracy [5].

Our feature representation is an extension of previous work for using pass locations to identify teams by their passing styles [30]. We also showed that these feature representations are promising for building quantitative player-rankings [8]. However, in this work, we extend these representations to have smarter ways of relating field locations, and a better breakdown of the understanding of offensive performance by different types of positional players.

Finally, we draw upon work that trains interpretable discriminative models, and then utilizes those model weights for developing quantitative metrics for stratifying populations. This is commonly done in the medical literature for developing risk scores. Specifically, both the TIMI risk score [3] for risk of adverse cardiac events and the SAPS III score [49] for evaluation of patient status in intensive care units utilize logistic regression models to understand which patient variables have the strongest relationship with adverse outcomes. They then utilize their trained model weights for constructing simple metrics that are both easy to compute manually, and can accurately separate high risk patients from low risk patients. This is analogous to
Table 2.1: Teams in the La Liga 2012-13 season and their number of shots attempted. Teams are ranked by their final standing in the league at the season end.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team</th>
<th>Shots</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Barcelona</td>
<td>528</td>
</tr>
<tr>
<td>2</td>
<td>Real Madrid</td>
<td>710</td>
</tr>
<tr>
<td>3</td>
<td>Atletico Madrid</td>
<td>513</td>
</tr>
<tr>
<td>4</td>
<td>Real Sociedad</td>
<td>557</td>
</tr>
<tr>
<td>5</td>
<td>Valencia</td>
<td>567</td>
</tr>
<tr>
<td>6</td>
<td>Malaga</td>
<td>443</td>
</tr>
<tr>
<td>7</td>
<td>Betis</td>
<td>456</td>
</tr>
<tr>
<td>8</td>
<td>Rayo Vallecano</td>
<td>567</td>
</tr>
<tr>
<td>9</td>
<td>Sevilla</td>
<td>553</td>
</tr>
<tr>
<td>10</td>
<td>Getafe</td>
<td>473</td>
</tr>
<tr>
<td>11</td>
<td>Levante</td>
<td>402</td>
</tr>
<tr>
<td>12</td>
<td>Atletico Bilbao</td>
<td>482</td>
</tr>
<tr>
<td>13</td>
<td>Espanyol</td>
<td>413</td>
</tr>
<tr>
<td>14</td>
<td>Valladolid</td>
<td>393</td>
</tr>
<tr>
<td>15</td>
<td>Granada</td>
<td>470</td>
</tr>
<tr>
<td>16</td>
<td>Osasuna</td>
<td>514</td>
</tr>
<tr>
<td>17</td>
<td>Celta de Vigo</td>
<td>460</td>
</tr>
<tr>
<td>18</td>
<td>Mallorca</td>
<td>485</td>
</tr>
<tr>
<td>19</td>
<td>Deportiva La Coruna</td>
<td>527</td>
</tr>
<tr>
<td>20</td>
<td>Zaragoza</td>
<td>461</td>
</tr>
</tbody>
</table>

how we use a model trained on predicting shots for identifying important origin and destinations of passes, and then utilize these locations to rank players.

2.3 The Data

2.3.1 Data Overview

The data we used are hand-labeled annotations of each ball-event that took place during the course of a match (e.g., each pass, tackle, shot, etc). A ball-event is recorded any time a player makes a play on the ball, apart from dribbling. The dataset also includes additional information for each ball-event such as the location, the players involved, and the outcome.

We focused specifically on the locations of pass attempt origins and destinations
Figure 2-1: The playing area split up into 18 zones. The left side of the field (zones 1-3) is the defensive side of the field, and the right side (zones 16-18) is the offensive area.

because we hypothesized that pass location is a strong indicator of team strategy and personnel. We use data collected from the 2012-13 La Liga season. La Liga is the premier league in Spain and is comprised of 20 teams. On average, each team in La Liga attempted around 18,000 passes during the entire season, with Barcelona making the most passes by a wide margin at 30,283, and Levante attempting the smallest number of passes at 13,094. We present a complete list of teams ranked by their final standing in La Liga with the number of shots they took in the 2012-13 season in Table 2.1.

2.3.2 Pass Location Representation

Each pass event in the dataset has an origin and destination location. To aggregate pass origin or destination locations that are near each other, we discretized the field into 18 zones, as shown in Figure 2-1. This representation has previously been shown to identify critical zones on the field associated with offensive outcomes such as shots and goals [15]. Additionally, in previous work we showed that by discretizing the field into these 18 zones, we can build representations of pass locations that are highly
indicative of team passing styles [8]. However, we did not want to represent each pass location as belonging to a single zone, since we do not believe the hard cutoffs between zones are representative of anything fundamental to the game of soccer.

Instead, we extended our previous representation of each pass location (either a location of a pass origin or a destination) to be a continuous-valued vector of length 18, where each element in the vector represents the closeness of a given pass location to the corresponding zone. We then compute the vector representation for a pass location \( l \) as \( r' = [r'_1 \ldots r'_18] \), where each element is

\[
r'_i = \frac{c_i}{\max(d(l, z_i), 1)}
\]

(2.1)

where \( d(l, z_i) \) is the Euclidean distance between the coordinates of \( l \) and the center of zone \( i \) \((z_i)\), and \( c_i \) is an indicator variable that takes the value 1 if zone \( i \) is one of the \( N \) closest zones (by Euclidean distance to the center of the zone) to \( l \), and 0 otherwise. We utilize the \( \max(d(l, z_i), 1) \) to prevent passes located very near the center of a zone from leading to extremely large values. Finally, we normalize \( r' \) by its L1-norm so that the \( \sum_{i=1}^{18} r'_i = 1 \).

The resulting vector \( r' \) has \( N \) non-zero values. This vector describes how close a pass location is to the \( N^{th} \) closest zones. For our task, \( N = 2 \) seemed to provide the best results. Intuitively, this results in a location representation that emphasizes the zone in which the pass occurred, but accounts for cases where a pass location is near the border of two zones. This representation allows aggregation or averaging of multiple passes in a sequence of passes without restricting any one pass in that sequence to have an origin or destination in just a single zone.

2.4 Methods

2.4.1 Feature Extraction

To extract the features that we used to build our predictive models, we first segment each game into a discrete sequence of observations. We segment the game at the
level of possessions. A possession in soccer is defined as a period of time when a single team retains the ball without an interruption in play or loss of the ball to the opposing team. Possessions contain a sequence of passes between players on the same team. We only considered possessions that had a minimum number of passes in order to remove epochs of play where a team only had the possession of the ball for a short period of time. We set this minimum to be three complete passes. Using a minimum of four or greater led to our models having a steep drop in predictive performance.

We extract features from each possession to construct the feature vectors for our learning algorithms. We construct the feature vector for a single pass with origin and destination locations \((l_o, l_d)\) in a possession by:

1. Computing \(r^{l_o}\) and \(r^{l_d}\), the representations of the pass origin and destination, respectively, using the method described in Section 2.3.2.

2. Constructing the matrix \(R^{l_{od}} = r^{l_o} \otimes r^{l_d}\), the outer product of the origin and destination representations. These origin-destination-pair features provide information about which pairs of origins and destinations are more likely to lead to shots (e.g. knowing that the ball was passed into the middle from the corner as opposed to just knowing that the ball reached the middle).

3. Constructing the feature vector as the concatenation of \(r^{l_o}\), \(r^{l_d}\), and a flattened \(R^{l_{od}}\).

The origin and destination representations have one value per zone, so each one accounts for 18 features. The flattened origin-destination-pair representation is of length 324. As a result, each pass is converted into a feature vector of length 360. To get a feature vector for a possession, we average the feature vectors across all completed passes in that possession. This feature vector represents a summary of the locations of pass origins and destinations of a possession.

Each feature vector is labeled with \(y \in \{1, -1\}\) according to how the possession ended. Possessions that ended in a shot taken by the offensive team were assigned a label of 1, and all others were assigned a label of −1.
2.4.2 Experimental Design and Testing

Upon converting each possession in a game to a feature vector, we then used these feature vectors to train models to relate passing strategy in a possession to shots taken. We first split the data into a training and holdout set. We split the data by using the first 80% of games chronologically as the training set, and setting aside the final 20% as the holdout set. We did this split to simulate the scenario of applying our models to newly generated data.

Using the training set, we trained an L2-regularized Support Vector Machine (SVM) model using the LIBLINEAR package [20]. We used LIBLINEAR’s asymmetric cost parameter option in order to account for the extreme class imbalance between positive and negative examples in the training set. We utilized 5-fold cross validation to find the optimal class-specific cost parameters on the training set. The folds were constructed at the game level so possessions in a single game were not split across multiple folds. We chose the cost parameters that had the maximum average Area Under the Receiver Operating Characteristic Curve (AUROC) on the five test folds, and used those parameters to train the final model. The final model was then tested on the holdout set.

2.5 Results

2.5.1 Classification Results

Our model that predicts whether a possession will end with a shot has an AUROC of 0.79 and an F-score of 0.31 (using a probability threshold of 0.5). We plot the ROC curve in Figure 2-2. We used this model to investigate the relationship between our features and shots by looking at the relative importance of each zone. We show the feature weights for the top 10 features by magnitude of weight in Figure 2-3. The weights presented in the figure are normalized by the total sum of the absolute value of all the weights to reflect a feature’s relative importance. We can see that no single feature dominates and that the top features have a mixture of positive and negative
weights. This provides further evidence that our model captures a trade-off between where passes are likely to lead to shot opportunities in the future and is not based solely on simple rules such as "shots happen when there are passes near the opposing team’s goal." In fact, zone 17, the area directly in front of the opposing team’s goal, does not appear in the top 10 features.

Looking at the relative feature importance provides insight into which pass origins and destinations generally lead to shot opportunities. Many of these top features involve zone 14, the "critical zone" centered in the field in front of the penalty box. Possessing the ball in this zone has been strongly identified to be associated with positive offensive outcomes [15]. This is congruent with our model, where 6 of the top 10 features are positively associated features that involve zone 14. However, not all of our top features involving zone 14 have positive weights. Passes to zone 14 from either zone 5 or zone 9 both have negative weights. Thus long completed passes to zone 14 are not positively associated with shot opportunities by our model. So while getting the ball into zone 14 can often lead to shot opportunities, it depends a great deal on how you get it there.
Many of the top features also involve the corner areas. The top two features overall are both passes between the corner areas and zone 14. This suggests that many passes that lead to shot opportunities later in the possession involve getting the ball into the corner of the field. However, there is also a strongly negatively associated feature weight with a ball sent from one corner to another (zone 18 to zone 16). This suggests that crosses across the field are harmful for generating shots if they result in the ball being played to the other corner.

2.5.2 Player Rankings by Shot Prediction Models

In the previous section, we described how we trained a model relating passes during a possession to the probability of the possession ending in a shot. The resulting model is a vector $\mathbf{w} = [\mathbf{w}^o, \mathbf{w}^d, \mathbf{w}^{od}]$, where $\mathbf{w}^o, \mathbf{w}^d, \mathbf{w}^{od}$ are the vectors of feature weights corresponding to the different zones for pass origins, destinations, and origin-destination pairs, respectively. These weights provide a conceptual map of the field.
that suggests which pass locations are most likely to lead to a shot opportunity later in the possession. We can use this map to rate a given pass by its association with shot opportunities in our model.

We took every completed pass in the La Liga 2012-13, and using our model computed an estimate of its relative importance for generating a shot. This importance, called Pass Shot Value (PSV), is computed for a pass with an origin in zone $i$ and a destination in zone $j$ as:

$$PSV(i,j) = w_i^o + w^d_j + w_{ij}^{od}$$

Thus, the PSV for a pass is the sum of the feature weights for its corresponding origin, destination and origin-destination pair. For example, a pass from zone 3 to zone 4 would have a PSV of the sum of the model weight for an origin in zone 3, the weight for the destination in zone 4, and the weight of the pair of having an origin of 3 and a destination of 4. We then computed the Average Pass Shot Value (APSV) for all players in La Liga who had over 200 completed passes in the 2012-13 season. We found that 200 passes filtered out everyone besides the top 15-20 players per team by number of completed passes, leaving the main contributors over that season. We again excluded passes that did not occur in a possession of 3 or more completed passes. Note that we only use the closest zone to the origin and destination for computing PSV. This is because we assume the predictive model has accounted for relationships between zones that are near each other.

We plotted the APSV for these players in Figure 2-4, which shows how our model would rank each player by their average tendency to complete passes that lead to a shot. Unsurprisingly, the APSV metric is biased towards offensive oriented players. In fact, the separation of the different positions by APSV indicates that APSV alone would be a fairly strong predictor of position. The goalkeepers are completely separate from the field position players. As such, APSV is most useful when comparing players within a position.

We ranked the top ten players by position category in Table 2.2. We also listed
Figure 2-4: APSV of passes for all players with more than 200 passes in the 2012-13 La Liga season. Each item in the plot represents the APSV of a single player. Players have an associated color and shape chosen by their position. The red line represents the APSV across the entire collection of players.

the top players by goals and assists in Table 2.3. Note that APSV is almost always negative. Most possessions do not end in a shot, and thus, most of the model’s features are negatively associated with a shot opportunity being generated. Therefore, players make passes with a negative model value the vast majority of the time. In spite of this, some offensive players and one midfielder (Mesut Ozil) have a positive APSV. This suggests that generally their passes were rated by the model to be positively associated with shots at the end of a possession.

We grouped forwards, strikers, and wingers into an “offense” category. The top players by APSV in the offense category include renowned players such as Ronaldo and Messi. They were the top two scorers in La Liga that season and they both finished the season in the top 10 for assists as well. In fact, we find that a significant ($\rho = 0.27, p < 0.05$) correlation between APSV and goals scored for the season for players in the “offense” category. However, neither goals nor assists are used in compute APSV, suggesting that these players are not only effective scorers but also effective passers. Others in this category were considered to be strong offensive players and appear in the list of top players by goals and assists as well.
Table 2.2: Top 10 players in La Liga 2012-13 by APSV. We separate the positions into three categories: Offense, Midfield, and Defense.

Table 2.3: Top 10 players in La Liga 2012-13 by Goals and Assists

Even within the midfield and defense categories, we see players that are identified to play an “offensive” style. Midfielders in the top 10 by APSV like Ozil and Kaka play most often in the “attacking midfielder” position. Marcelo Viera and Eliseu play as backs, but are known for being capable at playing in the wing position as well. Others like Dani Alves are pure backs but are known for contributing on the offensive end.
2.6 Conclusion

We presented a novel method of utilizing soccer event data to understand the relationship between pass location and shot opportunities. We showed that the locations of the origins and destinations of passes in a possession relate strongly to whether that possession will end in a shot. Using supervised machine learning techniques, we built a model for predicting whether a possession will end in a shot. The model had an AUROC of 0.79.

We used the features of this model to create a map to understand the relative importance for generating shot opportunities of passing from one location to another. We then used this map to build a data-driven ranking of players by weighing a pass by its relative importance for generating a shot later in the possession. When we ranked all players in La Liga 2012-13 with more than 200 passes with this metric, we see some of the elite attacking players at the top. This ranking also correlates well with standard offensive box score metrics such as goals and assists, even though neither were directly used in its computation. We believe this warrants further investigation into its utility as a player comparison tool. Furthermore, we have outlined a framework for constructing data-driven player metrics. For example, using actions taken by players on defense for predicting defensive outcomes may be used to help rank players by defensive ability.

We believe that our results show that appropriate analyses of pass event data in soccer can provide sometimes non-obvious insights. However, soccer is a complicated sport with constantly changing game situations. Incorporating temporal information (e.g. duration of possession, time elapsed between passes) in any analyses would provide more situation-specific insights. Also, utilizing player-tracking data as a source dataset would better allow investigation into the strategic aspects of the game that are not directly involved with the ball. Expanding our features to include sequential information could give a more detailed understanding of how passing strategy relates to outcomes. Lastly, if a team had a large collection of event data from their own games, they could build team-specific models that could provide a better analysis into
which strategies are most promising in their system. Further investigation will better reveal how useful this type of analysis can be for gaining a deeper understanding of the world’s most popular game.
Chapter 3

Non-shooting Player Movement
Contributions on Offense in Basketball

3.1 Introduction

In this chapter, we present a method for quantifying how player movement during a basketball possession contributes to offensive success. While many current basketball metrics focus on the player who scores, we try to quantify how the players who do not take the shot contribute on offense. Similar to the previous chapter, we take a data-driven approach. We choose a target for possession success, and train a machine learning model for understanding the relationship between that target and actions taken by all players other than the shooter on offense.

There are some major differences between the methods presented in this chapter to that of Chapter 2. Instead of the sparsely collected event location data, we utilize rich player tracking data. In contrast to the focus on passing actions during possessions, we evaluate the motions of the players themselves. Finally, in place of the linear model used in Chapter 2, we use a neural network approach. We lose out on interpretability of model weights, but allow the model to learn patterns of player motion that lead to
We train deep neural network models to relate patterns in player movement to success at the end of the possession. We choose a shot quality metric as a measurement of possession success. Our model then allows us to understand how player movement relates to quality shots. We do not employ any feature engineering, and therefore do not include preconceived notions of which motions or movement patterns are best.

We use an architecture designed to separate individual player contributions on offense. For any given possession, we calculate a contribution value for each non-shooting player on offense based on the model's assessment of how their movement contributed to the possession outcome. Based on these contributions, we develop two player metrics, Points Contribution Value (PCV) and Quality Contribution Value (QCV), that can track how much a player contributes to scored points and quality shots by looking at their movement during the possession. This enables us to assign value to a player's movement when it contributes to offensive success even though that player was not the player who scored. We show these metrics represent a novel measurement of offensive ability of a player based on the patterns of movement they take.

### 3.2 Related Work

Our models and data representation are closely related to previous analysis training neural networks on the basketball tracking data. Shah and Romijnders trained RNN models on ball trajectories to predict whether a shot would be successful [59]. Others have represented player trajectories during a basketball possession as images, and then built neural networks to learn from those images. Wang and Zemel showed that convolutional and recurrent neural networks can be used to classify types of basketball plays [69]. We discuss this application in more detail in Chapter 5. Harmon and others used a similar representation for predicting made baskets [25]. Zheng and others also demonstrated the power of deep models learning on images of basketball possessions, but their focus was on prediction tasks of actions within a possession,
instead of the result of the possession on the whole [76]. We utilize a similar image-based representation used in these studies. However, we do not treat each player trajectory as part of the same input. Instead we utilize a shared-weight architecture that keeps the trajectories of each player as a separate input. This allows for focusing on individual contributions based solely on how a player moves.

While there have been a number of papers published on using convolutional neural networks for image classification [32, 60, 66, 26], the most relevant work is on utilizing “siamese” architectures. These architectures, first described by Bromely and others [7] consist of multiple identical sub-networks for processing multiple inputs. Most commonly, they are used to verify if two inputs are identical for applications such as facial recognition [14]. In the case of this work, we utilize the sub-networks to process the multiple player trajectories separately.

3.3 Description of the NBA Player Tracking Data

3.3.1 Data Overview

The data we use in this chapter and in Chapters 4 and 5 consists of player tracking data collected from over 4000 National Basketball Association (NBA) professional basketball games during the 2012-2015 seasons and playoffs. Since this is the same dataset used by this and all subsequent basketball-related chapters, we formally describe the data here.

The NBA player tracking data used throughout this thesis were collected using the Stats SportVu system [64]. This system, depicted in Figure 3-1, utilizes 6 cameras installed in every NBA arena. The cameras are used to locate and track the locations of the five players for each team as well as the ball. The locations are precise to within a foot and are recorded at a rate of 25 Hz. This system is used to generate two sets of data, EVENTS and POSITIONS.

The EVENTS data are a list of discrete player actions, referee calls, and stoppages that occur over the course of the game. It is similar to the event location data used in
Chapter 2. We show a short example of the EVENTS data for a single segment during a game in Table 3.1. These data are generated from a combination of hand annotated play-by-play data and the optical tracking system. Recorded events include passes, dribbles, shots, fouls, and turnovers.

The POSITIONS data consist of the raw coordinates of the players and the ball over the course of each game. An example from the POSITIONS data for a short segment of play is depicted in Table 3.2. The POSITIONS dataset is simply a list of the $x$ and $y$ coordinates of each player, and the $x$, $y$, and $z$ coordinates of the ball at a sampling rate of 25 Hz. (We do not utilize the $z$-coordinates in this work.) We used the POSITIONS data both for finding the locations of players during instances of EVENTS, as well as extracting the movement trajectories of the players.

Like the work done with the soccer data in Chapter 2, all of the basketball analysis focuses on possessions. We use the EVENTS data to segment each game’s data into a sequence of possessions. In this case, we define a possession to be an uninterrupted
Table 3.1: An a short sample of the EVENTS data. Events are recorded sporadically as they occur throughout the game. We primarily use them to mark the beginning and end of possessions, and for identifying possession results such as shots.

<table>
<thead>
<tr>
<th>Game-ID</th>
<th>Period</th>
<th>Time</th>
<th>Player</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014110110</td>
<td>2</td>
<td>222.04</td>
<td>J-Harden</td>
<td>Pass</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>221.52</td>
<td>T-Ariza</td>
<td>Ball Received</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>220.76</td>
<td>T-Ariza</td>
<td>Pass</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>220.36</td>
<td>I-Canaan</td>
<td>Ball Received</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>218.88</td>
<td>I-Canaan</td>
<td>Dribble</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>217.80</td>
<td>I-Canaan</td>
<td>Dribble</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>217.24</td>
<td>I-Canaan</td>
<td>Dribble</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>216.08</td>
<td>I-Canaan</td>
<td>Dribble</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>215.25</td>
<td>I-Canaan</td>
<td>Pass</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>214.04</td>
<td>D-Howard</td>
<td>Ball Received</td>
</tr>
<tr>
<td>2014110110</td>
<td>2</td>
<td>212.88</td>
<td>D-Howard</td>
<td>Made Shot</td>
</tr>
</tbody>
</table>

series of pass, ball received, or dribble events by one team. A shot, foul, turnover, or any stoppage of play denotes the end of the possession, and the immediate start of a new possession. Thus, even if a team misses a shot and successfully gathers the offensive rebound, we still consider the missed shot to be the end of the possession. The same team can have multiple possessions in a row. For the games in the dataset, we identified about 700,000 offensive possessions where all five players and the ball spent at least five seconds on the offensive end of the court.

3.4 Methods

3.4.1 Possession Quality

For each possession in the data, we needed a concrete way of identifying which were “good” possessions. We cannot use shots taken as an outcome like in Chapter 2, since nearly every basketball possession ends in a shot. Our first instinct was to use points scored, however, we found this target to be a poor indicator of possession success. Many of the best shooters miss 40% or more of their shots, so much of the time points will not be scored simply because of an errant shot. Thus, points scored is not
Game-ID | Period | Time | Team | Player | x    | y    | z   
--- | --- | --- | --- | --- | --- | --- | --- 
2014110110 | 2 | 212.88 | - | Ball | 82.99 | 24.01 | 9.21 
2014110110 | 2 | 212.88 | Houston | T-Ariza | 56.69 | 26.88 | - 
2014110110 | 2 | 212.88 | Houston | D-Howard | 82.87 | 25.32 | - 
2014110110 | 2 | 212.88 | Houston | J-Harden | 69.99 | 5.92 | - 
2014110110 | 2 | 212.88 | Houston | I-Canaan | 70.90 | 10.17 | - 
2014110110 | 2 | 212.88 | Houston | T-Jones | 89.47 | 34.31 | - 
2014110110 | 2 | 212.88 | Boston | R-Rondo | 85.88 | 16.49 | - 
2014110110 | 2 | 212.88 | Boston | J-Green | 88.69 | 27.18 | - 
2014110110 | 2 | 212.88 | Boston | E-Turner | 68.46 | 27.06 | - 
2014110110 | 2 | 212.88 | Boston | T-Zeller | 84.88 | 25.98 | - 
2014110110 | 2 | 212.88 | Houston | A-Bradley | 73.76 | 7.82 | - 
2014110110 | 2 | 212.84 | - | Ball | 82.99 | 23.84 | 9.65 
2014110110 | 2 | 212.84 | Houston | T-Ariza | 56.60 | 26.88 | - 
2014110110 | 2 | 212.84 | Houston | D-Howard | 82.68 | 25.13 | - 
2014110110 | 2 | 212.84 | Houston | J-Harden | 69.80 | 5.99 | - 
2014110110 | 2 | 212.84 | Houston | I-Canaan | 70.68 | 10.09 | - 
2014110110 | 2 | 212.84 | Houston | T-Jones | 89.46 | 34.23 | - 
2014110110 | 2 | 212.84 | Boston | R-Rondo | 85.93 | 16.70 | - 
2014110110 | 2 | 212.84 | Boston | J-Green | 88.71 | 27.40 | - 
2014110110 | 2 | 212.84 | Boston | E-Turner | 68.49 | 27.14 | - 
2014110110 | 2 | 212.84 | Boston | T-Zeller | 84.70 | 25.84 | - 
2014110110 | 2 | 212.84 | Boston | A-Bradley | 73.66 | 7.83 | - 

Table 3.2: A short sample of the POSITIONS data. Positions for all players and the ball are recorded at a sampling rate of 25Hz. The ball’s z coordinate is tracked in addition to its x and y coordinates.

necessarily a good outcome to look at for identifying well-executed possessions, since a team may execute their offense perfectly, only to have the shooter miss.

We also investigated using open shots as our outcome. We defined an open shot to be when the closest defender is a sufficient distance away from the shooter. However, a single defender distance cutoff does not work for all shots. For example, many three point shots are taken with defenders multiple feet away from the shooter, which is much more rare for shots closer to the basket.

Ultimately, we chose a target that is based on the “quality” of the shot. We choose a shot quality that is based on the shooter’s history at certain ranges and defender distances. A similar criterion is used in [12]. The shot quality label is based on the expected number of points from that particular shot, where expected number of
points is calculated as:

\[ \text{ExpPoints} = P(\text{shot}) \times (2 + TP) \]  

(3.1)

where \( P(\text{shot}) \) is the probability of the shot being made, and \( TP \) is 1 if the shot is a three-pointer, and 0 otherwise.

\( P(\text{shot}) \) is calculated based on the shooter's history. For every player in the league, we calculated their percentage of shots made from 8 different ranges of distances away from the basket, as well if they were closely defended or not at each one of those ranges. We chose a defender distance of 3 feet as an indication of whether a shot was closely defended. Therefore, for each player we have 16 different shot percentages based on the distance of the shot and whether the shot was contested. Because we chose a coarse division of shot ranges, situations where a shooter’s history at a particular range had low support were extremely rare. We show an example of the shooter history for a single player in Table 3.3. For any given shot, the appropriate \( P(\text{shot}) \) is chosen based on a lookup of that shooter’s history.

<table>
<thead>
<tr>
<th>Shot Distance</th>
<th>Open</th>
<th>Not Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 2ft</td>
<td>0.92</td>
<td>0.74</td>
</tr>
<tr>
<td>2-4ft</td>
<td>0.91</td>
<td>0.68</td>
</tr>
<tr>
<td>4-6ft</td>
<td>0.92</td>
<td>0.67</td>
</tr>
<tr>
<td>6-10ft</td>
<td>0.60</td>
<td>0.51</td>
</tr>
<tr>
<td>10-15ft</td>
<td>0.54</td>
<td>0.44</td>
</tr>
<tr>
<td>15-22ft</td>
<td>0.50</td>
<td>0.45</td>
</tr>
<tr>
<td>22-25ft</td>
<td>0.44</td>
<td>0.34</td>
</tr>
<tr>
<td>&gt; 25ft</td>
<td>0.41</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Table 3.3: The shooter history for Kevin Durant. This lookup table indicates Durant’s likelihood of making a shot at different shot ranges, with and without having a defender within 3 feet.

We chose a classification approach to our learning task so that our model would give a clear indication of a possession being “good” or “bad.” We converted the continuous expected points metric into a binary label. We assigned possessions a label of 1 if the expected number of points was greater than the average points per
Figure 3-2: An example player’s trajectory during a possession after converting to our 25 × 25 × 2 pixel image representation. We generate an image for each non-shooting player on offense during the possession. The trajectory is blue when the player was moving without the ball, and red when they had the ball. The “fading” employed by our image representation provides information about directionality of the player’s movement as well as their speed at different points along the trajectory. The court lines are displayed for visualization purposes, and are not included as part of the input to the model.

possession for a team, and 0 otherwise.

3.4.2 Data Representation

Building off of previous work on possession classification in basketball [69, 25], we chose an imaged-based representation for our model inputs. This representation allows us to take advantage of the power of CNNs for discovery of spatial patterns that relate to our task. An example of a single player’s trajectory during a possession after conversion to our image-based representation is presented in Figure 3-2. Each player on offense is represented by a separate 2-channel image: one channel for the player’s movement while they didn’t have the ball, and the other for their movement when they were the ball handler. We only consider the trajectories during the time that all offensive players and the ball are on the offensive end of the court. Each image is of size 25 × 25 pixels. The images are constructed by calculating a pixel intensity corresponding to every location the player reached during a possession. For
Figure 3-3: The general layout of the shared-weight model architecture. Player images are fed through subnetworks with shared weights. Each network produces a single output value, which is summed before sigmoid activation, producing the final output.

each location \((i, j)\), we calculate the corresponding pixel's intensity as:

\[
    p_{ij} = \frac{t_{loc}}{t_{poss}}
\]  

where \(t_{poss}\) is the length of the possession in seconds, and \(t_{loc}\) is the time elapsed from the start of the possession and when the player reaches that location. If the player reaches a single location more than once, than the max intensity value is used. Thus, the trajectories begin to darken as it gets later in the possession. This allows us to incorporate some temporal information into the image and makes the directionality of each trajectory clear.

All possession images are pre-processed with a per-channel gaussian blur. This helps remove effects from the arbitrary nature of the pixel border locations and recording noise. Additionally, we only represent the trajectories from the last five seconds of play before the shot occurs. We wanted all the possessions to have the same length of time window so that the time-based fading was consistent across multiple examples. Empirically, we found little gain in predicting shot quality from looking at longer trajectories. For later learning tasks discussed in Chapters 4 and 5, we use the entire possession as input. Any possessions less than five seconds long or not ending in a shot were discarded.
3.4.3 Model Architecture and Training

We chose a shared-weight architecture to account for the trajectories of all of the players. We illustrate this architecture in Figure 3-3. This architecture keeps the player trajectories as distinct inputs to the network, but shares model weights across all inputs. Each of the player images is fed through the same subnetwork. Each subnetwork produces a single output value. The outputs corresponding to each of the input images are summed before a sigmoid activation to produce the final prediction. We could use any order-agnostic merging in place of the sum (e.g., max or average) but we felt that a sum best represented our desire to understand how all trajectories contribute to the prediction of shot quality.

In our early experiments, we trained neural network architectures that only considered a single input. To get our four player images into a single input, we simply aggregated all of the player trajectories into a single image by taking a global pixel-wise max across the four images. The resulting image contains all four of the player trajectories in one image. However, many of the player’s paths cross one another, often resulting in an image where one player’s trajectory occludes another. Additionally, keeping the player trajectories as separate inputs allows us to directly determine a given player’s contribution on a possession from the model.

Previous work representing basketball possessions [69, 25] also utilized single-input architectures, but kept each player image as a separate channel in the image. However, this requires some explicit ordering of the channels. The channel ordering was typically determined by position or learned player roles [36]. In practice, we found it difficult to come up with method for ordering the player trajectories that generalized well across all possessions, games, and teams. In fact, classification performance from models trained on the multi-channel player images were worse than aggregating them all into one image. This is likely because of factors such as teams not playing the same set of positions, so it is difficult to find roles that are meaningful to assign for every player across all possessions. Our architecture allows the model to consider each of the player movements separately, but without needing to know specifically
Figure 3-4: The architecture of the player subnetworks. The input is passed through 3 successive layers of 2D convolutions and max-pooling, before a dense layer to produce a single output.

which player was performing which role. Thus, the contribution from each input is only a product of how the player moved, and not based on any preconceived notion of how their role should contribute.

For the player subnetworks, we chose a simple 3-layer convolutional architecture illustrated in Figure 3-4. We feed the $25 \times 25 \times 2$ pixel input through 3 successive layers of 2D convolutions and max-pooling. In all cases the kernel size for the convolutions is of size $3 \times 3$ pixels, and $2 \times 2$ for the pooling steps. The output depths for the convolutional layers are 16, 32, and 64, respectively. This results in the original input being converted to a $4 \times 4 \times 64$ image. We then flatten that image and feed it into a final dense fully-connected layer to get a single output value.

We experimented with more complex subnetworks with more convolutional layers, and additional dense hidden layers. However, we found these additions added no predictive value. Additionally, we found no increase in performance by having the subnetworks output more than a single value. The single output also has the advantage of providing a single interpretable value corresponding to the each player’s contribution to possession success.
3.4.4 Experimental Design and Testing

We split the possessions into a training, validation and test set. The models were all implemented in Tensorflow [2] using Keras [13], and trained until convergence of validation set classification performance. Predictive performance was then assessed on the test set by measuring the AUROC.

3.5 Classification Results

<table>
<thead>
<tr>
<th>Model</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>0.639 (± 0.003)</td>
</tr>
<tr>
<td>CNN - Only Last Second</td>
<td>0.563 (±0.003)</td>
</tr>
<tr>
<td>Linear</td>
<td>0.619 (±0.004)</td>
</tr>
</tbody>
</table>

Table 3.4: AUROC for our model compared to a two baselines. 95% confidence intervals were found through bootstrapping the test predictions. The first baseline is the same model, but only contains trajectories from the last second before the shot. The second contains the same inputs, but uses linear subnetworks.

Our model was able to predict shot quality with an AUROC of 0.639 (± 0.003). We compare this performance to a few different baselines in Table 3.4. These baselines were chosen with the goal of trying to determine if the CNN actually discovered useful patterns in the player trajectory images, or is only picking up on simple rules based on the player locations. The first baseline uses the same model, but only uses the player trajectories over the last second of the possession before the shot. Its AUROC of 0.563 (±0.003) indicates that the non-shooting player movements that come well before the shot are important. We also compare to a baseline with the same input data, but utilizing a single linear layer subnetwork, as opposed to the CNN subnetwork illustrated in Figure 3-4. This model also performs worse than our model. This suggests our model is learning more than just linear relationships between locations on the court and shot quality, and is taking advantage of the power of the convolutional layers for learning spatial patterns.

We further explored the relationship between the window of time provided to our model and performance. We plot the performance of our model over various time
Figure 3-5: AUROC for predicting shot quality vs. various input time windows. Performance increases if more than just the end of the possession is used, suggesting that it is important to account for non-shooting player movements and location that happen well before the shot.

windows at the end of the possession in Figure 3-5. This plot indicates the AUROC on the test set at various time windows starting from five seconds before the shot to only a second before the shot. We can see that our model's performance generally increases as more information is added about the player movements before the shot, up to about three seconds before the shot. This indicates that even movement patterns that happen well before the moment of the shot are important for prediction shot quality.

3.6 Player Contribution Metrics

After our model is trained, we can use our model to get per-possession measurements of each of the non-shooting players contributions to shot quality via their movement. These contributions are directly derived from the subnetwork outputs. Given a player's subnetwork output for a possession, we calculate their Contribution Value
(CV) as the sigmoid of the subnetwork output. This yields a continuous value with a range of [0, 1]. Since CV is calculated independently for each player, every non-shooting member on offense could have a high CV in a given possession. This would indicate a high likelihood of a quality shot being taken.

We then tested how CV might be utilized to assess a player's offensive ability over the course of an entire season. We developed a new offensive metric called Points Contribution Value (PCV). PCV for a possession $i$ and player $j$ is defined as:

$$PCV(i, j) = CV_{ij} \times \text{Points}_i,$$  \hspace{0.5cm} (3.3)

where $CV_{ij}$ is the contribution value of for player $j$ in possession $i$, and Points$_i$ is the number of points scored in the possession. This means we can attribute the points scored in any possession to the non-shooters based on how their movement contributed on offense.

We calculated PCV for all players for offensive possessions in the 2015-2016 NBA season. We list the top 25 players by their PCV per game Table 3.5a. This list includes many top offensive contributors across the league. Lowry, Wall, Green, James, Butler, Leonard, and Paul were all players selected to the all-star teams that season. However, PCV is different from traditional offensive metrics. We show the leaders in points and assists per game in Table 3.6b. Rondo, Wall, Paul, Green, James, and Lillard were all in the top ten for assist makers that year. On the other hand, James Harden was a top player in points and assists that year, but he is not a top player by PCV. This indicates that even though Harden contributes a lot on the offensive end, he does not contribute as much with his movement in possessions where he does not shoot.

We also list the top 25 players by PCV per 100 possessions (in which they were not the shooter) in Table 3.5b. This measurement won’t be biased towards the players who play the most. This list also has top performers from the 2015-2016 seasons, including Curry, who was the league’s “Most Valuable Player” that year. PCV per 100 possessions may be more suitable than PCV per game for assessing the non-
<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>PCVpg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R-Gobert</td>
<td>18.03</td>
</tr>
<tr>
<td>2</td>
<td>N-Batum</td>
<td>16.69</td>
</tr>
<tr>
<td>3</td>
<td>K-Lowry</td>
<td>16.42</td>
</tr>
<tr>
<td>4</td>
<td>M-Kidd-Gilchrist</td>
<td>16.25</td>
</tr>
<tr>
<td>5</td>
<td>J-Wall</td>
<td>16.25</td>
</tr>
<tr>
<td>6</td>
<td>R-Rondo</td>
<td>16.22</td>
</tr>
<tr>
<td>7</td>
<td>D-Green</td>
<td>16.20</td>
</tr>
<tr>
<td>8</td>
<td>J-Johnson</td>
<td>16.11</td>
</tr>
<tr>
<td>9</td>
<td>D-Jordan</td>
<td>16.07</td>
</tr>
<tr>
<td>10</td>
<td>G-Hayward</td>
<td>15.65</td>
</tr>
<tr>
<td>11</td>
<td>J-Jack</td>
<td>15.62</td>
</tr>
<tr>
<td>12</td>
<td>L-James</td>
<td>15.43</td>
</tr>
<tr>
<td>13</td>
<td>G-Hill</td>
<td>15.37</td>
</tr>
<tr>
<td>14</td>
<td>J-Butler</td>
<td>15.37</td>
</tr>
<tr>
<td>15</td>
<td>T-Harris</td>
<td>15.35</td>
</tr>
<tr>
<td>16</td>
<td>K-Leonard</td>
<td>15.19</td>
</tr>
<tr>
<td>17</td>
<td>K-Walker</td>
<td>15.14</td>
</tr>
<tr>
<td>18</td>
<td>D-Lillard</td>
<td>15.14</td>
</tr>
<tr>
<td>19</td>
<td>W-Matthews</td>
<td>15.03</td>
</tr>
<tr>
<td>20</td>
<td>K-Korver</td>
<td>15.03</td>
</tr>
<tr>
<td>21</td>
<td>G-Dragic</td>
<td>15.02</td>
</tr>
<tr>
<td>22</td>
<td>M-Morris</td>
<td>14.71</td>
</tr>
<tr>
<td>23</td>
<td>K-Caldwell-Pope</td>
<td>14.63</td>
</tr>
<tr>
<td>24</td>
<td>C-Paul</td>
<td>14.56</td>
</tr>
<tr>
<td>25</td>
<td>B-Knight</td>
<td>14.51</td>
</tr>
</tbody>
</table>

League Average 8.54

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>PCVper100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>K-Thompson</td>
<td>57.38</td>
</tr>
<tr>
<td>2</td>
<td>D-Green</td>
<td>57.14</td>
</tr>
<tr>
<td>3</td>
<td>R-Westbrook</td>
<td>55.88</td>
</tr>
<tr>
<td>4</td>
<td>M-Ginobili</td>
<td>55.58</td>
</tr>
<tr>
<td>5</td>
<td>C-Paul</td>
<td>54.98</td>
</tr>
<tr>
<td>6</td>
<td>D-Schroder</td>
<td>54.71</td>
</tr>
<tr>
<td>7</td>
<td>S-Curry</td>
<td>54.50</td>
</tr>
<tr>
<td>8</td>
<td>P-Mills</td>
<td>54.48</td>
</tr>
<tr>
<td>9</td>
<td>C-Watson</td>
<td>54.34</td>
</tr>
<tr>
<td>10</td>
<td>I-Smith</td>
<td>53.77</td>
</tr>
<tr>
<td>11</td>
<td>J-Simmons</td>
<td>53.77</td>
</tr>
<tr>
<td>12</td>
<td>A-Miller</td>
<td>53.70</td>
</tr>
<tr>
<td>13</td>
<td>K-Irving</td>
<td>52.96</td>
</tr>
<tr>
<td>14</td>
<td>D-Goodeen</td>
<td>52.77</td>
</tr>
<tr>
<td>15</td>
<td>J-Wall</td>
<td>52.64</td>
</tr>
<tr>
<td>16</td>
<td>B-Diaw</td>
<td>52.48</td>
</tr>
<tr>
<td>17</td>
<td>K-Durant</td>
<td>52.30</td>
</tr>
<tr>
<td>18</td>
<td>L-James</td>
<td>52.29</td>
</tr>
<tr>
<td>19</td>
<td>D-Cousins</td>
<td>52.13</td>
</tr>
<tr>
<td>20</td>
<td>J-Jenkins</td>
<td>52.05</td>
</tr>
<tr>
<td>21</td>
<td>E-Bledsoe</td>
<td>51.88</td>
</tr>
<tr>
<td>22</td>
<td>E-Mudiay</td>
<td>51.75</td>
</tr>
<tr>
<td>23</td>
<td>D-Motiejunas</td>
<td>51.72</td>
</tr>
<tr>
<td>24</td>
<td>L-Patterson</td>
<td>51.67</td>
</tr>
<tr>
<td>25</td>
<td>U-Haslem</td>
<td>51.67</td>
</tr>
</tbody>
</table>

League Average 42.21

(a) PCV per game

(b) PCV per 100 possessions

Table 3.5: Top 25 players by PCV in the 2015-2016 NBA season

starters for a team. For example, we see that Manu Ginobili is ranked 4th for the season by PCV per 100 possessions. Ginobili has not been a regular starter for the San Antonio Spurs since the 2010-2011 season, but he still plays quality minutes and is a valuable contributor to the Spurs’ offense.

While PCV provides a quantitative measure of the non-shooters contributions to a given possession, it is dependent on the outcome of the possession. All players will get a PCV of zero for a scoreless possession. As previously discussed, this reliance on whether a basket was scored may provide misleading insights into possession success.

53
Table 3.6: Leaders in points and assists per game during the 2015-2016 NBA season. There is some overlap with the leaders in PCV and QCV, but some of the most dominant scorers, such as James Harden, are not rated highly by PCV or QCV. A possession can be executed perfectly outside of a shot being missed.

To look at a measure of player contributions in successful possessions regardless of whether points were scored, we developed a second metric called Quality Contribution Value (QCV). Like PCV, QCV relies on the contribution value of a player, but QCV for a possession \( i \) and player \( j \) is defined as:

\[
QCV(i, j) = CV_{ij} \ast Quality_i, \tag{3.4}
\]

where \( CV_{ij} \) is the contribution value of for player \( j \) in possession \( i \), and \( Quality_i \) is the binary shot quality label for possession \( i \).

We show the top 25 players by QCV per game and QCV per 100 possessions in 3.7. QCV may not be as biased towards high performing and scoring teams as PCV. Players are still rewarded for contributing to a good shot, regardless of whether that shot goes in. The top two players by QCV per game and QCV per 100 possessions, Rajon Rondo and DeMarcus Cousins, both played for the Sacramento Kings in the 2015-2016 season. The Kings did not make the playoffs that season, but two of their starters had the highest QCV per game and QCV per 100 possessions.
<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>QCVpg</th>
<th>Rank</th>
<th>Name</th>
<th>QCVper100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R-Rondo</td>
<td>9.35</td>
<td>1</td>
<td>D-Cousins</td>
<td>31.88</td>
</tr>
<tr>
<td>2</td>
<td>K-Caldwell-Pope</td>
<td>8.97</td>
<td>2</td>
<td>T-Wroten</td>
<td>31.86</td>
</tr>
<tr>
<td>3</td>
<td>R-Gobert</td>
<td>8.66</td>
<td>3</td>
<td>K-Thompson</td>
<td>30.41</td>
</tr>
<tr>
<td>4</td>
<td>M-Morris</td>
<td>8.53</td>
<td>4</td>
<td>D-Schroder</td>
<td>29.95</td>
</tr>
<tr>
<td>5</td>
<td>R-Rubio</td>
<td>8.51</td>
<td>5</td>
<td>J-Teague</td>
<td>29.94</td>
</tr>
<tr>
<td>6</td>
<td>T-Harris</td>
<td>8.45</td>
<td>6</td>
<td>D-Green</td>
<td>29.75</td>
</tr>
<tr>
<td>7</td>
<td>D-Green</td>
<td>8.43</td>
<td>7</td>
<td>R-Rondo</td>
<td>29.68</td>
</tr>
<tr>
<td>8</td>
<td>G-Hayward</td>
<td>8.28</td>
<td>8</td>
<td>S-Curry</td>
<td>29.62</td>
</tr>
<tr>
<td>9</td>
<td>J-Wall</td>
<td>8.27</td>
<td>9</td>
<td>N-Vucevic</td>
<td>29.57</td>
</tr>
<tr>
<td>10</td>
<td>J-Jack</td>
<td>8.25</td>
<td>10</td>
<td>K-Marshall</td>
<td>29.05</td>
</tr>
<tr>
<td>11</td>
<td>E-Payton</td>
<td>8.18</td>
<td>11</td>
<td>A-Wiggins</td>
<td>29.05</td>
</tr>
<tr>
<td>12</td>
<td>A-Wiggins</td>
<td>8.12</td>
<td>12</td>
<td>K-Irving</td>
<td>28.95</td>
</tr>
<tr>
<td>13</td>
<td>N-Vucevic</td>
<td>8.09</td>
<td>13</td>
<td>R-Rubio</td>
<td>28.86</td>
</tr>
<tr>
<td>14</td>
<td>N-Batum</td>
<td>8.05</td>
<td>14</td>
<td>E-Payton</td>
<td>28.86</td>
</tr>
<tr>
<td>15</td>
<td>L-James</td>
<td>7.99</td>
<td>15</td>
<td>M-Dellavedova</td>
<td>28.61</td>
</tr>
<tr>
<td>16</td>
<td>M-Kidd-Gilchrist</td>
<td>7.78</td>
<td>16</td>
<td>M-Belinelli</td>
<td>28.48</td>
</tr>
<tr>
<td>17</td>
<td>K-Walker</td>
<td>7.73</td>
<td>17</td>
<td>M-Williams</td>
<td>28.42</td>
</tr>
<tr>
<td>18</td>
<td>T-Thompson</td>
<td>7.72</td>
<td>18</td>
<td>A-Miller</td>
<td>28.38</td>
</tr>
<tr>
<td>19</td>
<td>A-Drummond</td>
<td>7.71</td>
<td>19</td>
<td>R-Westbrook</td>
<td>28.20</td>
</tr>
<tr>
<td>20</td>
<td>J-Teague</td>
<td>7.70</td>
<td>20</td>
<td>R-Gay</td>
<td>27.87</td>
</tr>
<tr>
<td>21</td>
<td>D-Lillard</td>
<td>7.67</td>
<td>21</td>
<td>D-Rudez</td>
<td>27.83</td>
</tr>
<tr>
<td>22</td>
<td>K-Irving</td>
<td>7.65</td>
<td>22</td>
<td>A-Bogut</td>
<td>27.82</td>
</tr>
<tr>
<td>23</td>
<td>D-Cousins</td>
<td>7.63</td>
<td>23</td>
<td>D-Collison</td>
<td>27.74</td>
</tr>
<tr>
<td>24</td>
<td>E-Bledsoe</td>
<td>7.62</td>
<td>24</td>
<td>D-Wade</td>
<td>27.63</td>
</tr>
<tr>
<td>25</td>
<td>K-Korver</td>
<td>7.60</td>
<td>25</td>
<td>E-Bledsoe</td>
<td>27.33</td>
</tr>
</tbody>
</table>

| League Average | 4.24 | League Average | 22.77 |

(a) QCV per game  
(b) QCV per 100 possessions

Table 3.7: Top 25 players by QCV in the 2015-2016 NBA season

### 3.7 Conclusion

We trained a neural network to predict the quality of shots from the trajectories of the non-shooting players. This model uses a shared-weight architecture that is designed to account for the trajectories of each of the four non-shooting players. Even though the model does not see the shooting player's movements as input (or the ball while the shooter has control of it), we are still able to predict the shot quality of the possession with an AUROC of 0.639 (± 0.003). This model significantly outperforms
a linear model of similar design, and models that are only given the final locations of the non-shooting players. This suggests that the model is learning patterns about the movement trajectories of the players, and not just about specific locations on the court.

We designed our model to output a separate value for each player's input trajectory. This value represents each trajectory's contribution to the final prediction of shot quality. We used this value, called Contribution Value, as a measure of how much a given player's movement contributed to the possessions success. From the Contribution Value we defined two new player metrics, Points Contribution Value and Quality Contribution Value, that qualitatively assess the player's movement along with the possession outcome. When we ranked the top players in the 2015-2016 NBA season by these metrics, we see top offensive contributors across the league atop the rankings. These results suggest that PCV and QCV could be useful and interesting metrics for measuring non-shooting players' contributions on offense by their movements alone.

Since the completion of the work for this chapter, we've greatly improved upon many of our ideas for using neural networks for analyzing player movements in basketball. Many of these methods are described in Chapters 4 and 5. Specifically, we've found that for other tasks, using a video-based instead of an image-based representation allows for important temporal patterns to be learned. This work could benefit from such a change in representation as well. Secondly, it might be beneficial to learn from all of the players' trajectories instead of each one separately, since we've found player interactions can be important for other tasks such as play recognition. While this wouldn't allow us to keep player contributions explicitly separated, it could be the basis of interesting team-focused metrics and analysis. Finally, the analyses presented in Chapters 4 and 5 account for the movements of the players on defense. The actions of the defense provide the model a broader context for understanding how helpful certain player moments are on offense.
Chapter 4

Unsupervised Exploration of Basketball Possessions

4.1 Introduction

In Chapters 2 and 3, we used supervised machine learning to understand the relationship between the movement of non-shooting players and possession outcomes. We defined a notion of possession success, and trained a model to predict that target.

In this chapter, we take an unsupervised approach to analyzing offensive possessions in basketball. Instead of training a predictive model, we train a model that learns the general structure of player and ball movements. This allows us to discover interesting patterns in the trajectories without needing to relate them to a specific task or outcome. We then can use all available examples to let the data-space influence which patterns to learn, without specializing our model towards a specific application.

We use an auto-encoder neural network architecture to learn a low-dimensional encoding of the player and ball trajectories during an offensive possession. Similar to the work in Chapter 3, we use convolutional layers to learn spatial features in player movements. However, we use a “video-based” representation of the player and ball trajectories in place of the image-based representation. This representation allows learning of both spatial and temporal patterns of movement. Additionally, we
aggregate all players on the same team in each frame of the video. This loses out on player identity, but allows for learning patterns about player interactions.

We train a neutral network framework that contains a combination of convolutional and recurrent layers. This model learns to convert the high-dimensional input to a low-dimensional vector. The encoding contains features that are useful for querying similarly executed possessions. We then show how clustering the encoded possessions can lead to discovery of common offensive motifs. Finally, we present some results on how these motifs can provide insight into how team’s patterns of play are affected by different personnel, game situations, and coaching.

4.2 Related Work

The most relevant work from the sports domain has been on finding latent representations of raw positional basketball player data [47, 72]. Recently, Miller and Bornn built a vocabulary of “actions” to summarize player trajectories and applied topic modeling to learn distributions of actions that teams utilize over the course of a possession [48]. This resulted in a latent representation of possessions that is useful for grouping similar offensive strategies. While our models also project the tracking data onto a latent feature space, we take a neural network based approach.

As described in Section 3.2, we draw on much of the state-of-the-art in CNNs for designing our convolutional layers. The models used in this chapter also include recurrent layers. We chose to utilize Long Short-term Memory (LSTM) units as our particular recurrent layer implementation since they have been demonstrated to learn patterns that occur across many time steps in a sequence [27, 21]. More recently, deep networks using LSTMs have been found suitable for machine learning tasks such as speech recognition [23] and caption generation [68].

We utilize an encoder-decoder framework for the auto-encoder. This draws on previous work for training on both images and time series [41, 19]. Others have applied this framework to video data for tasks such as action recognition [62, 67]. Additionally, the recurrent layers in the decoder use principles from sequence-to-

58
sequence learning to reconstruct the original input [65].

We also draw from previous work on learning and exploring embeddings of high-dimensional inputs. This is similar to work in natural language processing involving learning word embeddings [46, 45] and their subsequent categorization [57].

4.3 Methods

4.3.1 Data Representation

We use the NBA player tracking dataset described in Section 3.3. We convert the set of the player and the ball trajectories, each a multivariate time series, to a video-based format. This representation is similar to that used in [69, 25]. It is an extension of the image representation we used in Chapter 3 to incorporate the time dimension.

First, all trajectories in all possessions are interpolated to the average duration between all players crossing half court and the possession end, approximately 10 seconds. While this interpolation significantly stretches or shrinks the time dimension of some possessions, the vast majority of possessions are within a few seconds of the average duration. We found that interpolation led to better encoding qualities than using padding or other methods of normalizing the length of the examples. We split each possession into $m$ segments, where $m$ is chosen such that each segment contains a half second of the possession. These segments correspond to the resulting “frames” in the video representation.

An example frame of the video-based representation for a single possession is presented in Figure 4-1. Each frame of the video consists of an image of size $32 \times 32$ pixels (each pixel covers roughly two square feet of the floor), and includes the locations of players or the ball that were reached in the time period used to construct the frame. We also spatially blur each frame slightly to account for the arbitrary locations of the pixel borders and measurement noise. Since we have an explicit time dimension, we did not find adding the “fading” over time (as used in Chapter 3) to be helpful. This results in converting a collection of time series trajectories to a
Figure 4-1: A single frame of a possession in our video-based representation. The frame consists of a three channel image, corresponding to the players on offense, the players on defense, and the ball. It indicates the trajectories during a half second segment of the possession. The court lines are shown for visualization purposes and are not included in the actual representation.

4-dimensional volume of size $21 \times 32 \times 32 \times 3$ where the dimensions are number of frames, height, width, and number of channels, respectively.

We aggregate the player trajectories from each team into a single image channel, leading to three channels total (offense, defense, and the ball). Previous work has kept the trajectories of the players separate. This requires ordering the players based on position, player role [36], or other statistics. However, it can be detrimental to assign a role to a player, when teams, players, and roles vary widely across possessions. We found that aggregating all the players for a team into the same channel worked better.

4.3.2 Auto-encoder Architecture

Figure 4-2 shows an overview of the encoder-decoder auto-encoder framework. The encoder converts the high-dimensional video input to a low-dimensional vector encoding. This encoding is then passed to the decoder layers. These layers convert the encoding back to the original input space.

The encoder architecture (shown in Figure 4-3) consists of a series of convolutional and pooling layers (Conv-E), followed by an LSTM (LSTM-E). We show a detailed breakdown of the convolutional layers in Figure 4-4. Each frame is input into the
convolutional layers separately. Each convolutional layer consists of 64 filters of size $3 \times 3$ pixels, and is followed by ReLu activation and 2D max-pooling. Weights for these filters are shared across time points. These layers find spatial filters that are invariant to where they happened on the court or when they occurred during the possession. For example, these layers could find patterns such as two players coming near each other to execute a screen. The convolutional and pooling layers also help to reduce the number of variables before the parameter-dense recurrent layers. After the sequence of inputs is passed through the convolutional layers, the resulting sequence is fed to the LSTM. This recurrent layer is used to find temporal patterns.

After feeding the entire video sequence through the convolutional and LSTM layers, the final LSTM state represents the low-dimensional encoding of the input. Empirically, we found an encoding size of 128 worked well in practice. The encoding contains information about both spatial and temporal patterns of movement.

The decoder network (shown in Figure 4-5) takes the low-dimensional encoding, and reconstructs the original input. First, the low-dimensional encoding is replicated for each output time step and is used as input into an LSTM (LSTM-D) to recover the sequence from the convolutional layers in the encoder. The original video frames are reconstructed using a series of convolutional layers (Conv-D) with upsampling. The upsampling performs the reverse operation of the max-pooling in the encoding layers by copying each pixel in the input to a $2 \times 2$ region in the output.

We compare the reconstructed input to the original input and use weighted L2 reconstruction loss as the network’s unsupervised learning objective. This weighted
Figure 4-3: Overview of the architecture of the encoder layers. The encoder consists of convolutional layers (Conv-E) followed by an LSTM (LSTM-E) to convert the input sequence into a low-dimensional encoding.

Figure 4-4: Breakdown of the convolutional layers. Each arrow represents a convolutional layer, followed by ReLu activation and 2D max-pooling. Each convolutional layer consists of 64 size $3 \times 3$ filters.
loss is defined as:

\[
R(x_n, \hat{x}_n) = \frac{1}{N} \sum_{n=1}^{N} \|A(x_n) \odot (x_n - \hat{x}_n)\|_2^2,
\]

(4.1)

where \( \odot \) indicates element-wise multiplication, \( x_n \) is the original input, \( \hat{x}_n \) is the reconstructed input, and \( A(\cdot) \) is an indicator matrix defined as:

\[
\forall a_{nij} \in A(x_n), a_{nij} = \begin{cases} 
1 & \text{if } x_{nij} > 0, \\
0.05 & \text{otherwise.}
\end{cases}
\]

(4.2)

This objective places a higher importance in reconstructing the non-zero pixels than the zero-valued pixels in the original input. Without the weighting, we found that network would often reach minima in which it would produce uniformly near-zero reconstructions. This likely happens because the original inputs are quite sparse.
4.3.3 Experimental Design and Testing

We set aside 20% of the data for validation purposes, although our models always converged before seeing all of the unlabeled training data. All models were trained using Keras with the TensorFlow backend [13, 2]. Models were trained until validation set performance converged. During training we used dropout in the recurrent layers to avoid overfitting.

4.4 Reconstruction Results

We explored how well our auto-encoder model's low-dimensional encoding represented their original video inputs. We show input frames from two representative example possessions with their corresponding reconstructions by the model in Figure 4-6.

In general, the reconstructions abstractly represent the locations of the players and the ball, despite converting the input from a high-dimensional, multivariate time series, to a fixed-size, low-dimensional vector. The reconstructed inputs are more blurry than the original, sometimes even blurring trajectories from two players into a single area. This is likely caused by the tendency for L2 loss to produce blurry reconstructions [42], as well as the variation seen across the examples. This blurriness suggests that our model is not simply memorizing the original locations of all the players and the ball at each frame. Instead the encoding has information about the general areas covered by the teams and the ball at any given time. This may help the model better understand the relationships between similar, but not identically executed possessions.

4.5 Possession Encoding Analysis

The 128-length vector encoding of each possession is a latent representation of the original player and ball trajectories. It contains information regarding the player locations, movement patterns, and interactions over the course of the entire possession. We explored how we can utilize possessions in their encoded format to provide insight...
Figure 4-6: Two examples of input frames and their corresponding reconstructions by the auto-encoder. The reconstructions are blurry in comparison to the original trajectories of the players and the ball at each frame. This blurriness indicates that the encoding does not exactly memorize the original locations, but preserves the general areas that are reached in a given time frame. The court lines are shown for visualization purposes.
into different offensive strategies.

4.5.1 Possession Querying

One application this encoding is useful for is for finding possessions that are “similar” to a query possession. Instead of visually searching through the possession database, or comparing possessions in their raw time series format, we can simply find the closest possessions to the query possession in the encoded space. These comparisons are fast and they compare features about the trajectories instead of the raw $x$ and $y$ coordinates.

We compare this method of querying to finding neighbors in the raw trajectory space. However, since the trajectories for each team are an unordered set of multivariate time series, we cannot represent the raw data as a feature vector. Therefore, we developed a distance metric for comparing possessions in the raw data space that finds an optimal alignment between players. Given two possessions $POSS_0 = (O_0, D_0, b_0)$ and $POSS_1 = (O_1, D_1, b_1)$, where $O$ represents the set of offensive trajectories, $D$ is the set of defensive trajectories, and $b$ is the trajectory for the ball, we compute the distance between $POSS_0$ and $POSS_1$ as:

$$\text{Dist}(POSS_0, POSS_1) = \text{MinAlign}(O_0, O_1) + \text{MinAlign}(D_0, D_1) + d(b_0, b_1) \quad (4.3)$$

The component $d(b_0, b_1)$ is the Euclidean distance between the two ball trajectories. $\text{MinAlign}$ is an “optimal” alignment between two sets of trajectories. It is computed by the following procedure for sets $S_0$ and $S_1$:

1. Compute every possible bijective mapping between $S_0$ and $S_1$.

2. For each mapping, compute the pairwise Euclidean distance between the mapped elements.

3. Return the smallest distance found over all the mappings.

Therefore, we find the minimum possible distance between the trajectories for both
the offense and defense in each possession. Since there are five players on each team, we need to search through a total of 240 (5! for offense, 5! for defense) total mappings.

In our experiments, we chose 20 random possessions from our database of offensive possessions to use as queries. We then found the five closest neighbors by both Euclidean distance in the encoded space and the raw data space as described above. We found that both methods were able to find extremely similar possessions to the query possession. We show two examples of query possessions and their closest neighbor by each method in Figure 4-7. We have no “ground truth” standard by which to evaluate the query results. However, in 20 query possessions we visually inspected, the neighbors from the encoded and raw data space were quite similar to the query possession. Through visual inspection, we could not tell a difference in quality of the matches between the two distance methods.

However, comparing possessions in the encoded space offers many advantages over comparing possessions in the raw data space. First, comparing the encodings is much faster than comparing the raw data trajectories. We computed the pairwise distances between 30,000 encoded possessions on a single CPU in roughly 44 seconds. Computing distances on the same examples using the raw trajectories would take about 47 days. Secondly, as previously described, the raw trajectories cannot be represented as a feature vector. This limits clustering the raw trajectories to connectivity-based methods such as hierarchical clustering.

4.5.2 Possession Clustering

The latent representation of possessions learned by our auto-encoder makes it simple to query similar possessions. By simply computing the encoding for a query possession of interest, one can find similarly executed possessions by finding the closest neighbors by Euclidean distance. This enables quick and easy querying of possessions, and further analysis of offensive motifs by cluster analysis.

We took 60,000 possessions from the 2014 and 2015 NBA seasons, and converted them to their corresponding encodings using our auto-encoder. We then clustered the encoded possessions using k-means clustering using the scikit-learn python package.
Figure 4-7: Two examples of query possessions and their corresponding closest neighbor by both encoding distance and distance in the raw data space. Both methods find possessions that are similar to the query possession. Only the players on offense (blue) and the ball (red) trajectories are shown.
We set the number of clusters $k = 30$, as it represented the “knee” in the response curve between $k$ and the average within-cluster distance.

The 30 resulting clusters each represent different offensive motifs. Each cluster contains around 2,000 of the 60,000 possessions, which is consistent with k-means’ preference for similar sized clusters. We found that the possessions distances to their corresponding cluster center were approximately normally distributed.

We extracted the 10 possessions that were closest to the cluster center. While these 10 possessions alone cannot account for all of the offensive strategies held within the cluster, we hypothesized that these possessions would characterize some of the common patterns of play. We visually examined the player and ball trajectories in these possessions to identify these patterns. By repeating this procedure for each of the 30 clusters, sampled some of the different regions in the distribution of possession encodings.

We found that the 10 closest possessions to the cluster center appeared to be very similarly executed possessions. We show three of the top 10 closest to the center of cluster 24 in Figure 4-8. When we visualize these three possessions, we see the same sequence of player and ball movements. One of the players sets up near the baseline and runs towards the post area. There he runs around a screen set by another offensive player. The ball handler then passes the ball to the open cutting player. All of the 10 possessions we examined had this same pattern of movement, regardless of whether the cut occurred on the left or right side of the floor.

We identified the common patterns that occurred for each of the other clusters. We describe the offensive movements taken in each cluster in Table 4.1. While we are not basketball experts, these descriptions provide a high level overview of the common offensive motifs that occur within the possessions we examined from each cluster. They can be used as a quick reference, or as a guide to finding offensive motifs that teams and other interested parties could use to explore these motifs in more detail.
Figure 4-8: Three examples of possessions that were close to the center of cluster 24. In all possessions the offense uses an off ball screen in the post to get a player open on the perimeter. Only the players on offense (blue) and the ball (red) trajectories are visualized for clarity.

4.5.3 Team Profiles

The ability to assign any given possession a cluster allows us to characterize teams by offensive motifs they tend to execute. We built a profile of each team based on how the possessions they run distribute among the various clusters. Given a collection of possessions executed by a team, we compute a profile vector of length 30 where each element represents the fraction of possessions that belong to that cluster.

By visualizing the profile vector for each team, we can gain insight into the types of offensive possessions run by each team. We show the cluster profiles for the Cleveland Cavaliers and the San Antonio Spurs over the course of the 2015-2016 NBA season.
<table>
<thead>
<tr>
<th>Cluster</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Screen to the right of the top of the key, pass to the right corner</td>
</tr>
<tr>
<td>2</td>
<td>Pick-and-roll at the top of the key</td>
</tr>
<tr>
<td>3</td>
<td>Weaving at the three point line, off-ball screens to set up cross-court passes</td>
</tr>
<tr>
<td>4</td>
<td>Passing back and forth across the perimeter</td>
</tr>
<tr>
<td>5</td>
<td>Pass to right post, players cutting into the paint</td>
</tr>
<tr>
<td>6</td>
<td>Inbounds play from back or side of offensive end</td>
</tr>
<tr>
<td>7</td>
<td>Short passes on the left side, off-ball screens on the right</td>
</tr>
<tr>
<td>8</td>
<td>Pass to right perimeter, on-ball screen</td>
</tr>
<tr>
<td>9</td>
<td>On-ball screen, pass to screener or far side corner</td>
</tr>
<tr>
<td>10</td>
<td>Pass to left elbow, drive to the basket or pass to perimeter</td>
</tr>
<tr>
<td>11</td>
<td>Pick-and-roll on the right side</td>
</tr>
<tr>
<td>12</td>
<td>Screen on the right side of the top of the key, passes across the perimeter</td>
</tr>
<tr>
<td>13</td>
<td>Ball handler drives to left side of the basket, then kicks it out or shoots</td>
</tr>
<tr>
<td>14</td>
<td>Screen on left side of the perimeter, then a shot or a across court pass</td>
</tr>
<tr>
<td>15</td>
<td>Isolation at the top of the key, then a on-ball screen</td>
</tr>
<tr>
<td>16</td>
<td>Pass to the left perimeter, drive or a pass back to the center</td>
</tr>
<tr>
<td>17</td>
<td>Offensive rebound, kick back out to the perimeter</td>
</tr>
<tr>
<td>18</td>
<td>Pick-and-roll on the right side of top of the key</td>
</tr>
<tr>
<td>19</td>
<td>Early pass to the right corner while everyone is setting up</td>
</tr>
<tr>
<td>20</td>
<td>Passes in and out of the right side of the paint, off-ball screens on the left side</td>
</tr>
<tr>
<td>21</td>
<td>Passing and cycling around the perimeter</td>
</tr>
<tr>
<td>22</td>
<td>Early drive or pass to the left side</td>
</tr>
<tr>
<td>23</td>
<td>Left side post-up</td>
</tr>
<tr>
<td>24</td>
<td>Off-ball screen down low, pass to the perimeter</td>
</tr>
<tr>
<td>25</td>
<td>On-ball screen on right elbow, drive or pass to the center</td>
</tr>
<tr>
<td>26</td>
<td>Early pick-and-roll while setting up</td>
</tr>
<tr>
<td>27</td>
<td>Screen on left elbow, then shot or pass to the corner</td>
</tr>
<tr>
<td>28</td>
<td>Early pass or drive to the right post</td>
</tr>
<tr>
<td>29</td>
<td>Passes along the perimeter, players weaving from the baseline to the perimeter</td>
</tr>
<tr>
<td>30</td>
<td>Screen on right side followed by a shot or pass to the middle</td>
</tr>
</tbody>
</table>

Table 4.1: Description of the offensive motifs seen within each cluster

in Figure 4-9. The radar charts allow for easy comparison of the possessions each team executed. During this season, Cleveland was well known for executing the pick-and-roll [34], which is reflected by their prominent peaks for clusters 2 and 11. The Spurs have long been known as a team that relies less on set plays, and more on ball movement, motion, and spacing [31]. This aligns well with their more frequent execution of possessions from clusters 10, 21, and 23. These clusters characterize an offense that relies on driving, passing, and off-ball player movement.

In addition to identifying differences between teams, the profiles can also be used to understand the effects of changes in coaching and personnel from season to season. In Figure 4-10, we show the cluster profiles for two teams for both the 2014-2015 and the 2015-2016 seasons. Both teams maintained the key players on their roster across both of these seasons. However, during the 2014 season, Oklahoma City suffered an injury to their star player, Kevin Durant, and hired a new coach in between seasons. The lack of Durant and the coaching change is reflected by the drastic change in
possessions run between the seasons. However, Golden State, who went to the NBA finals in both of these seasons, is consistent between the years.

The profiles can also be used to identify trends across different game moments. In Figure 4-11, we show again show a cluster profile for the Cleveland Cavaliers during the 2015-2016 season. To demonstrate how the Cavaliers change their strategy in different game situations, we compute their cluster profiles for each quarter separately. We see a change in offensive style from quarter to quarter. For example, in the forth quarter, we see an even more exaggerated reliance on pick-and-roll type possessions, indicated by the peak for cluster 2.

4.6 Conclusion

We explored the patterns of player and ball movement during possessions in basketball in an completely unsupervised manner. We trained an auto-encoder that uses convolutional and recurrent layers to learn a low-dimensional encoding of the the player and ball trajectories during a possession. Our model converts the high-dimensional inputs into a 128-dimension vector encoding.
Figure 4-10: Cluster profiles for the Oklahoma City Thunder and the Golden State Warriors during the 2014-2015 and the 2015-2016 seasons. The charts display how OKC's injuries and coaching change led to differences in their offensive style across seasons. Golden State was much more consistent between the two years.
Figure 4-11: Cluster profiles for the Cleveland Cavaliers for the 2015-2016 during the each quarter. The profiles indicate a noticeable change in strategy in different quarters.
The vector encoding learned by the auto-encoder contains features about spatial and temporal patterns that occur during the possession. By using the Euclidean distance in the encoded space, we can quickly and accurately find similarly executed possessions. We clustered a large set of encoded possessions. The resulting clusters characterized different offensive motifs used throughout the league. By creating a team profile based on the distribution of motifs utilized, we can create a “fingerprint” of a team’s offensive play over different seasons and game situations. This profile enables comparing teams by their offensive styles, and shows measurable effects of team strategy being changed by events such as injuries and coaching changes.

The offensive motifs laid out by our clusters are not an exhaustive list of offensive strategies used in the NBA. Instead, they demonstrate a framework for future unsupervised analysis of possessions in basketball using player tracking data. They can also be used as a guide for teams to a deeper exploration into different patterns of player and ball movement on offense.
Chapter 5

A Semi-supervised Framework for Basketball Play Recognition

5.1 Introduction

Set plays are preplanned and coordinated sequences of player and ball movement often used to try to gain an advantage over the opposing defense. Plays are an important part of many teams’ offensive strategy. Being able to automatically label a team’s own plays could be of great help for quantitatively understanding play success against different teams, defenses, and strategies. There are hundreds of thousands of offensive possessions in an NBA season. However, labeling a large number of them is infeasible because:

- They must be labeled by visually reviewing videos of possessions and identifying any set plays run during the possession.

- They require expert knowledge to label. Untrained observers cannot accurately recognize the plays easily, rendering crowdsourcing techniques unusable.

This difficulty in obtaining a large set of labeled examples impedes the performance of machine learning methods. Previous work has shown the feasibility of utilizing machine learning to classify set plays using player tracking data [29, 69]. However, these methods are limited by the amount of labeled training data. These authors
were able to obtain good performance by training on a small set of examples (n ≈1000), but making significant improvements on classification performance would likely require more examples.

In this chapter, we present a semi-supervised approach for classifying plays using player tracking data from NBA games. We build off the encoder-decoder architecture from Chapter 4. However, instead of training in an unsupervised fashion using only unlabeled data, this model optimizes a combination of supervised and unsupervised learning objectives. The unsupervised objective enables the model to utilize unlabeled data to understand movement patterns that occur in basketball possessions. The supervised objective helps the model learn features that are important for distinguishing between the labeled plays. We demonstrate that this combination leads to more generalizable models for play classification than supervised models trained on labeled data alone.

5.2 Related Work

Other work has focused specifically on using machine learning for annotating tracking data from the NBA. Previous models learned to recognize when screens happen, and how teams choose to defend them [44, 43]. Our work has a similar, but more challenging, annotation task of automatically recognizing when set plays occur.

Previous work has assessed the feasibility of recognizing set plays using player tracking data. Intille and Bobick used player tracking data to recognize different plays in American football [28]. Others have specifically worked on recognizing plays from basketball tracking data. Perše et al. used trajectories of players to identify motifs on offense and related those to known play types [52]. Kates used domain knowledge to hand-engineer features that characterize a small number of plays of interest [29]. Using these features they trained play-specific models that can accurately classify six different play types. We use the same set of labeled plays as Kates. However, we do not use hand-engineered features. Instead, we use a neural network architecture to learn which patterns are important. Wang et al. built such a model using a
different set of labeled plays than Kates [69]. We utilize similar techniques for our supervised models, and demonstrate how this idea can be extended to the semi-supervised setting.

Finally, we build on recent work on deep-learning based semi-supervised machine learning techniques. Rasmus et al. trained a network that combines a classification model with a denoising auto-encoder [54]. Others used stacked convolutional auto-encoders for image classification [75], and Zhang et al. demonstrated that this technique could be applied to a large-scale image classification task [74].

5.3 Data Overview

We use the NBA player tracking dataset described in Section 3.3. Additionally, we use the same data preprocessing and representation as described in 4.3.1.

The dataset includes 608 labeled offensive possessions. These labels identify one of six possible set plays that was executed during that possession. A set play is a predetermined, coordinated series of movements and passes by the offensive players. An example set play is diagramed in Figure 5-1. A label corresponding to a set play
indicates that those preplanned series of movements were executed at some point during the possession.

All labeled examples are for possessions executed by the same team over the course of a single NBA season. The team defending varies across examples. Only one label is assigned per example. The labels were assigned by a domain expert by reviewing video of the possession. We breakdown the number of examples per class in Table 5.1.

Table 5.1: The number of examples for each play class. Each of these examples was labeled by an expert to be an instance of one of six plays.

<table>
<thead>
<tr>
<th>Play</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play 1</td>
<td>99</td>
</tr>
<tr>
<td>Play 2</td>
<td>132</td>
</tr>
<tr>
<td>Play 3</td>
<td>40</td>
</tr>
<tr>
<td>Play 4</td>
<td>87</td>
</tr>
<tr>
<td>Play 5</td>
<td>156</td>
</tr>
<tr>
<td>Play 6</td>
<td>94</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>608</strong></td>
</tr>
</tbody>
</table>

5.4 Methods

5.4.1 Semi-supervised Model Architecture

We use a semi-supervised neural network architecture that combines supervised and unsupervised learning objectives. Figure 5-2 shows an overview of the encoder-decoder architecture. This architecture is nearly identical to the auto-encoder architecture used in Chapter 4. However, the semi-supervised model includes an additional pathway for predicting the play from the encoding layer.

The encoder (shown in Figure 5-3) converts the high-dimensional video input to a low-dimensional vector encoding. From the encoding, there are two branches continuing through the network. The first is a fully-connected layer with a softmax activation, predicting a play class. The supervised learning objective is defined as
Figure 5-2: Overview of encoder/decoder auto-encoder model architecture. This framework is nearly identical to the auto-encoder used in Chapter 4, but it also includes a pathway for predicting a play class from the encoding layer.

cross-entropy loss. The second branch is to the decoder layers.

Besides the fully-connected branch (denoted by the dashed lines in 5-3), the encoder architecture is identical to the one used by our auto-encoder. See Section 4.3.2 for a more detailed description of these layers.

The decoder network takes the low-dimensional encoding, and reconstructs the original input. We used the exact same decoder architecture as described in Section 4.3.2. Figure 4-4, shows an overview of this architecture. We use the same weighted L2 reconstruction loss as the auto-encoder model as well.

Learning Objective

The semi-supervised model minimizes the loss:

\[ L(x_n, \hat{x}_n, y_n) = \lambda_r R(x_n, \hat{x}_n) + C(x_n, y_n), \]
Figure 5-3: Overview of the architecture of the encoder layers. The encoder consists of convolutional layers (Conv-E) followed by an LSTM (LSTM-E) to convert the input sequence into a low-dimensional encoding. The pathway for the supervised learning objective (represented by the dashed lines) is only considered when training with labeled batches.

where \( R(x_n, \hat{x}_n) \) is the weighted L2 reconstruction loss (described in Section 4.3.2), \( C(x_n, y_n) \) is the classification loss, and \( \lambda_r \) is a hyperparameter that balances the importance of the reconstruction loss.

The model is trained with a combination of unlabeled and labeled batches. When training with unlabeled data, only the reconstruction errors \( R(x_n, \hat{x}_n) \) are taken into account. In our experiments, setting \( \lambda_r = 1 \), and using a ratio of 10 to 1 of unlabeled to labeled batches during training lead to the best results.

By training with a combination of supervised and unsupervised learning objectives, our model learns features that are useful for distinguishing the labeled plays, but are also heavily influenced by the structure of the data. Intuitively, the reconstruction term regularizes the supervised loss. We want to learn features that can best classify plays, but we also must be able to summarize characteristics that occur in the player movements found in offensive possessions in general. This enables us to
exploit the structure found in unlabeled examples, while still learning features that distinguish among the plays.

5.4.2 Supervised Model

We also tested a supervised model that was trained only using the 608 labeled examples. This model uses the same layers as the encoder in the semi-supervised network, but does not include any of the decoder layers.

5.4.3 Auto-encoder Model

We also used our auto-encoder model from Chapter 4 as another baseline model to test against. This model also learns a latent representation of basketball possessions, but in an entirely unsupervised fashion. This allows us to test how well just learning this representation from unlabeled data allows us to learn useful features for play prediction.

We convert each labeled example from our video representation to a 128-length vector using the auto-encoder. From the converted examples we train a multinomial logistic regression for classifying plays. This two-step approach to building a play classifier allows us to test how well we can recognize plays if we rely only on the unlabeled examples for training the convolutional and recurrent layers.

5.4.4 Experimental Design and Testing

For all supervised training during the frameworks, we split the labeled data into 80-10-10 percent splits of training-validation-test sets. For the auto-encoder and semi-supervised model, we set aside 20% of the unlabeled data for validation purposes, although those models always converged before seeing all of the unlabeled training data. All models were trained using Keras with the TensorFlow backend [13, 2]. Models were trained until validation set performance converged. Each classification model was then tested on the held out test set.
5.5 Classification Results

Table 5.2: Accuracy for the models tested. The semi-supervised model significantly outperforms the supervised and auto-encoder models on our 6-way play classification task.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>0.692 (± 0.020)</td>
</tr>
<tr>
<td>Auto-encoder</td>
<td>0.742 (± 0.016)*</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td><strong>0.777 (± 0.016)</strong>*~</td>
</tr>
</tbody>
</table>

* significantly > supervised (p < .05)
^ significantly > auto-encoder (p < .05)

We list the accuracy for our models in Table 5.2. The semi-supervised model significantly outperformed all models tested, with an accuracy of 0.777 on our six-way classification task.

To further explore why the semi-supervised model performs better than the supervised model, we compared the confusion matrices of each. We show the confusion matrix (aggregated across all test splits) for the supervised and semi-supervised model in Table 5.3. Except for a small decrease in accuracy for Play 3, the semi-supervised model is more accurate than the supervised model across all play types. This improvement in performance comes largely from a boost in performance for Play 6. The supervised model confused many instances of Play 6 with Play 1 and Play 4. All of these play types involve a pass to the wing or the perimeter. However, Play 1 and Play 4 include a screen to get the pass receiver open, while Play 6 does not. A screen is an extremely common strategy that is executed at least once in nearly every offensive possession in basketball. Since the semi-supervised model is trained on many more examples of offensive possessions, it may be better at recognizing screens and other common offensive strategies.

The model trained using expert-designed on this same set of labeled examples had an accuracy of 0.726. However, the task they trained on varied slightly from ours. They trained separate models for each play, and each model was given features from small segments of the possession that matched the expected play length. We chose
Table 5.3: Confusion matrices for the supervised and semi-supervised models. A large part of the semi-supervised models increase in performance over the supervised model comes from a greater accuracy classifying Play 6.

the more difficult task of using the entire possession as input and producing which play it contained. In spite of this, the semi-supervised model is still more accurate without using any domain knowledge for feature engineering.

5.6 Discussion

5.6.1 Classification

Our model strongly suggests that there is value in jointly training a model with labeled and unlabeled examples. The supervised objective pushes the model to find features that are best for discriminating between the different classes of possessions in the labeled set. While the unsupervised objective leverages a large number of unlabeled instances to find patterns that represent the structure of movement that occurs in offensive possessions in general. This combination leads to learning the patterns that are most conducive to classifying plays, but with a regularization of finding features
that are representative of offensive possessions outside of what is seen in the labeled set of examples. These results show that this type of regularization can be effective in cases where you have a small set of labeled examples, and a much larger set of unlabeled examples.

We also observed that the auto-encoder trained on unlabeled examples could be used to build a more accurate classifier than the supervised model that was trained on the labeled examples alone. The encoding was learned entirely on unlabeled examples. This suggests that much of the difficulty in training a model for recognizing plays is in first learning features about the general structure of basketball possessions. This is similar to how deep models for image classification first learn general features about the structure of images, and then learn features related to specific classes in layers further down the network.

5.6.2 Play Querying

We also wanted to explore how a team might utilize a play recognizer outside of automatically annotating their own plays. One potential application is querying data from other teams for similar instances of plays. For example, given a sufficient amount of known demonstrations of a particular play, can we discover if another team runs an identical or similar play?

To explore how well our models generalize to identifying plays outside of the domain that they were trained on, we also tried classifying sets of unlabeled examples. We used the semi-supervised model to assign a play-label to a large set of unlabeled examples. These unlabeled examples contain offensive possessions that were executed by different teams than the team that we have labels for. They were not used in the training of the model.

We focused on the top 5 plays by model confidence for each play type. We used the model as a guide to finding plays that are most likely to be matches to the ones we trained on. We visually examined these 30 plays (5 plays × 6 play types) and identified 16 instances of plays that seem to strongly match the patterns found in each play type, and 14 that did not. We present two of the matching instances in
Figure 5-4: A visualization of a ground truth example of plays 1 and 5, and similarly run plays in the unlabeled data found by our model. The black circles denote the location of each player on offense.
Some of the play types seemed to be easier to query in the unlabeled data than others. Each of the 5 unlabeled examples that were labeled play 1 seemed to be a true instance of play 1. For plays 2 and 3, only one or two of the 5 seemed to be a similarly executed play. In spite of this, we were able to find instances of all play types in the hundreds of thousands of unlabeled instances by manually reviewing only a few of the possessions that were labeled by our model. In contrast, when we reviewed 30 random possessions in the data, we found no possessions that looked like an instance of our labeled plays.

Extracting potential instances in the unlabeled data could also be the basis of an active learning system. The existing model could be used as a filter to find instances that are highly likely to be new labeled instances. This greatly prunes down the amount of data that has to be manually reviewed to find new positive instances to add to the labeled set. It also provides informative negative instances. By iteratively discovering, labeling, and retraining on new examples, the model’s accuracy could be improved even further. Future experiments could assess how well play classification could be improved by an active learning environment.

5.7 Conclusion

We demonstrated a real-world application of semi-supervised learning. The player tracking data is rich, high-dimensional, and noisy. Set plays are an abstract concept, and it is difficult to construct features for recognizing them in the data without expert domain knowledge, so its a task that’s well suited for neural network approaches to discovering patterns. However, it is difficult to collect more than a small set of labeled examples. Thus, a semi-supervised framework that naturally allows for training on both unlabeled and labeled examples is well suited for this particular dataset and task.

Our semi-supervised model is able to accurately recognize set plays in basketball
player tracking data. This model is jointly trained on labeled and unlabeled examples. It is significantly more accurate than a model trained on labeled examples alone, and one trained on encodings from a model that learned only from unlabeled examples.

This work shows the utility of deep-learning based semi-supervised frameworks, as well as deep-learning techniques in general on high-dimensional sports location data. Further work demonstrating this same technique on another set of labeled plays would help confirm the utility of this technique for play recognition and discovery. These same techniques could also be applied to sports applications other than play recognition, as well as datasets and tasks outside of the sports domain.
Chapter 6

Summary and Conclusion

In this thesis we presented multiple applications of machine learning for deriving insights from sports data. While much of past sports research has focused on discrete events, we applied our work to location data. These data provide new opportunities in sports analysis, allowing quantitative insight into patterns of position and movement. Location data are often high-dimensional, noisy, and contain information from multiple interacting agents. One may not know ahead of time what patterns and relationships are most interesting and actionable to learn. In spite of these challenges, we demonstrated approaches to using machine learning to better understand and utilize these data with multiple applications in two different sports.

Historically, sports analyses have relied heavily on expert knowledge, e.g., expert-designed metrics like WAR and PER. In our work, we tried not to put inherent domain expertise within our features or models, and let the data “speak for itself.” We used data-driven methods for a variety of different learning paradigms. We used supervised machine learning to relate patterns in location and movement to outcomes. We also explored the general structures and patterns of movement through unsupervised learning. Finally, we explored learning frameworks that utilize both labeled and unlabeled examples with semi-supervised models. Through each of these paradigms we were able to demonstrate methodologies for gaining new understanding in its respective domain.

In Chapter 2, we demonstrated how supervised machine learning can be used to
create new quantitative metrics in soccer. We trained a model that uses the location of pass origins and destinations during a possession, and accurately predicts whether the possession ends in a shot. We then developed a new passing metric, called Pass Shot Value, based on the model’s weighting of which locations are most conducive to leading to shots. This metric provides a way of quantitatively evaluating the offensive utility of any pass, and seems to correlate well with established measures of offensive ability. One drawback to this work is that we only had data from one season and a single league. Verifying this analysis on more datasets would allow us to better assess the robustness of this method. Secondly, in contrast to the other chapters in this dissertation, we used sparsely collected ball-event data. Having more fine-grained player tracking data could open the door to more detailed and accurate analysis of passing strategy in soccer.

In Chapter 3, we extended work similar to that presented in Chapter 2 to the basketball domain. However, instead of focusing on the locations of passes, we used player tracking data to learn patterns about the locations of the players themselves. We developed an image-based representation of the player trajectories during a possession. We trained a neural network model that takes in those images, and predicts whether the possession ends in a quality shot. We feed each player’s trajectory as separate inputs into the model to understand how much each player contributed to the predicted shot quality. In contrast to most shot-centric analysis, we only looked at the contributions of the non-shooting players. As a result our model has the ability to measure the contribution of all the non-shooting players to the predicted shot quality. We developed two new player metrics using this measure. This metric assesses a players offensive abilities based entirely on the way they move, and not discrete outcomes such as points or assists. However, this work is limited by the use of static images as the inputs to our model. As we discuss in Chapters 4 and 5, there is often much to be gained by including temporal information in the input. Secondly, the model evaluates each player trajectory separately. While this makes it simpler to calculate each individual players contribution to the possession, we miss out on patterns relating to player interactions. Including these pieces of information could make our
In Chapter 4, we took an entirely unsupervised approach to analyzing basketball possessions. Instead of training a predictive model, we trained an auto-encoder that learns a low-dimensional encoding of basketball possessions. By designing the model to convert the high-dimensional trajectory data into a 128-length vector, we encouraged the learning of generalizable patterns about player and ball movement. These patterns are reflected in each possession’s corresponding encoding. We showed that looking at nearest neighbors in the encoded space allows for fast and scalable querying of similar possessions. We also clustered a large corpus of possessions into 30 different clusters. By examining the possessions contained in each cluster, we identified a common offensive motif for each one. By profiling a team based on their cluster distribution across all of their possessions, we can gain insight into different team’s offensive styles, as well as how teams change their offense to account for the effects of injuries, coaching changes, and different game situations. These 30 motifs are certainly not an exhaustive list of all possible offensive strategies, but this framework could be useful for a deeper unsupervised dive into different offensive techniques in the NBA.

In Chapter 5, we explore a semi-supervised learning framework in the context of set play recognition in the NBA. While the NBA player tracking dataset contains hundreds of thousands of possessions, it is infeasible to hand label more than a small set of set plays. We developed a semi-supervised neural network architecture that can be trained on a combination of labeled and unlabeled examples. The unlabeled examples provide the data to learn generalizable features about offensive possessions, while the labeled data pushes the model to learn how to distinguish among the plays. By balancing supervised and unsupervised learning objectives, our model learned to recognize set plays with 78% accuracy on a six-way classification task. This is significantly better than a model trained on labeled examples alone. We then demonstrated that by using this model on unlabeled data, one can find examples of possessions that look strikingly similar to some of those in our labeled set. However, many of the unlabeled examples that our model confidently predicts to be of a certain class look...
dissimilar to the ground truth examples. Extending this framework to include an active learning component could help extend the usefulness of our model beyond the set of plays we trained on.

While each chapter addressed a specific application of using machine learning to understand some aspect of a particular sport, we adhere to a general pattern throughout the thesis. We identified an particular aspect or portion of the sport, chose a suitable representation of the raw data, and finally designed and trained a model for learning the underlying patterns of interest. This pattern could be used for many other applications not only in soccer and basketball, but to any other sport with high-resolution location data.

Though the methods used in this thesis are all applied in the sports domain, they could be useful in many other applications as well. For many multi-agent domains with motion data, it can be difficult to find a mapping between the individual agent trajectories. We demonstrated that aggregating agent trajectories into an image or video-based representation can assess this challenge, while still allowing for learning patterns in individual agent motion and agent interaction. Secondly, in many real-world applications, labeled data is scarce, but unlabeled data is plentiful. The semi-supervised framework we investigated is one way of utilizing unlabeled data to help learn more generalizable models for a supervised learning task.
Bibliography


[38] Patrick Lucey, Alina Bialkowski, Mathew Monfort, Peter Carr, and Iain Matthews. “Quality vs Quantity”: Improved Shot Prediction in Soccer using Strategic Features from Spatiotemporal Data. 2014.


[53] B. R. And the silver goes to... The Economist, September 2011.


