Statistical Methods for Tracking Data in Sports

by

Jacob Mortensen

M.Sc., Brigham Young University, 2016
B.Sc., Brigham Young University, 2014

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2020

Approval

Name:	**Jacob Mortensen**
Degree:	**Doctor of Philosophy (Statistics)**
Title:	**Statistical Methods for Tracking Data in Sports**
Examining Committee:	**Chair:** Gary Parker Associate Professor
	Luke Bornn Senior Supervisor Associate Professor
	Derek Bingham Supervisor Professor
	Lloyd Elliott Internal Examiner Assistant Professor
	Dave Higdon External Examiner Professor Department of Statistics Virginia Polytechnic Institute and State University
Date Defended:	**August 21, 2020**

Abstract

In this thesis, we examine player tracking data in basketball and soccer and explore statistical methods and applications related to this type of data.

First, we present a method for nonparametric estimation of continuous-state Markov transition densities, using as our foundation a Poisson process representation of the joint input-output space of the Markovian transitions. Representing transition densities with a non-stationary point process allows the form of the transition density to vary rapidly over the space, resulting in a very flexible estimator of the transition mechanism. A key feature of this point process representation is that it allows the presence of spatial structure to inform transition density estimation. We illustrate this by using our method to model ball movement in the National Basketball Association, enabling us to capture the effects of spatial features, such as the three point line, that impact transition density values.

Next, we consider a sports science application. Sports science has seen substantial benefit from player tracking data, as high resolution coordinate data permits sports scientists to have to-the-second estimates of external load metrics traditionally used to understand the physical toll a game takes on an athlete. Unfortunately, this data is not widely available. Algorithms have been developed that allow a traditional broadcast feed to be converted to x-y coordinate data, making tracking data easier to acquire, but coordinates are available for an athlete only when that player is within the camera frame. This leads to inaccuracies in player load estimates, limiting the usefulness of this data for sports scientists. In this research, we develop models that predict offscreen load metrics and demonstrate the viability of broadcast-derived tracking data for understanding external load in soccer.

Finally, we address a tactics question in soccer. A key piece of information when evaluating a matchup in soccer is understanding the formations utilized by the different teams. Multiple researchers have developed methodology for learning these formations from tracking data, but they do not work when faced with the heavy censoring inherent to broadcast tracking data. We present an algorithm for aligning broadcast tracking data with the origin, and then show how the aligned data can be used to learn formations, with performance comparable to formations learned from the full tracking data.

Keywords: tracking data; Poisson point process; Markov model; broadcast tracking data; mixture model

Dedication

To my girls: Hannah, Jane, Eden, and Rosemary.

Acknowledgements

First and foremost, I would like to thank my advisor, Dr. Luke Bornn. Luke has been an invaluable mentor and friend, and provided excellent guidance and direction throughout my PhD. His tremendous work ethic, professionalism, and selflessness are qualities I will forever try to emulate. He encouraged independence and expressed confidence in me when I did not always feel it myself. I know I am a better researcher for it.

I am also grateful for my co-advisor, Dr. Derek Bingham. I was consistently inspired by the intensity and focus he brought to bear on understanding scientific problems. A consummate teacher, I left all of our meetings feeling like he made an effort to make me not just a better statistician, but a better human being.

I appreciatively acknowledge the friendship of Nate Sandholtz. I could not have asked for a better person to experience the highs and lows of the past four years with. I look forward to coauthoring many papers together in the future.

Thank you to my parents, Todd and Rebecca Mortensen, and the rest of my family for their unwavering support, and most of all, thank you to my wife and daughters. My completion of this program would not have been possible without their sacrifice and this accomplishment is as much theirs as mine.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Fans, coaches, analysts, and athletes have long used statistics in order to better understand sports. Henry Chadwick is credited as the inventor of the box score in baseball, which was first published over 150 years ago in *Clipper* magazine in 1859 (Pesca, 2009). This is the first recorded example of using numerical summaries to describe the *actions* that occurred in a game rather than simply the *outcome*. Baseball is particularly well-suited to this type of analysis because much of the game consists of a series of discrete events between just two players, the pitcher and the batter. Long before games could be televised, daily readers of the newspaper were able to use this box score information to reconstruct much of the action that actually occurred in the game. Although box scores exist for other sports, such as basketball, hockey, and soccer, the continuous nature of these games and, in the case of soccer and hockey, low number of scoring events, mean that simple counts fail to capture crucial contextual factors impacting game outcomes and have inherently less value.

Moving beyond simple counts, event data (i.e., play-by-play data) adds more structure to sports data by providing temporal sequencing information. Each observation typically consists of an event, like a shot attempt, made goal, pass, or foul, the player (or players) involved in the event, a timestamp, and in some cases, limited location information, such as where a shot was taken from. This allows analysts to understand not just whether or not an assist occurred, for example, but the exact sequence of events that led to a goal, and use a model to understand how a given event impacts goal scoring. A variety of interesting research has been done on event level data in sports, including calculating changes in expected goals to evaluate the quality of passes in soccer (Bransen and Haaren, 2019; Power et al., 2017), evaluating player efficiency in regards to decision making in basketball (Goldman and Rao, 2011), and estimating per-down win probabilities in the National Football League (Lock and Nettleton, 2014). Additional examples of research done on event data include spatial models for shot chart data (Reich et al., 2006), building a basketball game simulator by applying a Markov model to play-by-play data (Vračar et al., 2016), and estimating player impacts on goal scoring in hockey with competing risk models (Thomas et al., 2013) and deep reinforcement learning (Liu and Schulte, 2018). Also in hockey, Yu et al. (2019) examined pace of play and its relationship with offensive outcomes such as shots on goal and shooting percentage. Event based analysis in soccer includes constructing play sequences in order to produce entropy maps and use them to characterize team behavior (Lucey et al., 2013a), identifying unique patterns of passes and considering their relationship with player success (Bekkers, 2017), and using deep reinforcement learning to value players and actions (Liu et al., 2020). Fichman and O'Brien (2019) borrowed ideas from economics to show how a Stackelberg equilibrium could be used to determine optimal shot selection strategies and suggested that the three point shot would become even more popular in the National Basketball Association, and Omidiran (2011) used play-by-play data and box score statistics to develop a more accurate adjusted plus/minus model.

While event data provides significantly more contextual information than a box score, it still lacks crucial information about what players were doing who were not directly involved with the recorded event, or what players did in between events. For example, in basketball, as a dangerous shooter like Klay Thompson of the Golden State Warriors moves around the court, he can draw defensive attention away from his teammates, leading to more valuable shot opportunities, without ever touching the ball. The need for data that better described player behavior in these continuous settings led to the introduction of player tracking data to the English Premier Football league by Prozone in the late 1990's (Medeiros, 2017). In this thesis, the term player tracking data refers to a stream of x-y coordinate data for players in a game at a high temporal density, typically ranging from 10 to 30 observations per second. This type of data can be generated using a wearable device via technologies like GPS or radio telemetry, but the data examined in the following chapters is more narrowly referred to as optical tracking data. Optical tracking data is typically generated by installing multiple stationary cameras throughout an arena (see `https://www.statsperform.com/team-performance/basketball/` for an example) and using computer vision techniques to convert the image in each video frame to a collection of player locations. Originally, Prozone accomplished this using pixel classification techniques (Vandenbroucke et al., 2003) though a vast literature exists on multi-object detection in multicamera settings (see (Wang, 2013; Luo et al., 2017; Ren et al., 2010; Hamid et al., 2014; Ristani et al., 2016; Beetz et al., 2006; Previtali et al., 2017) for just a handful of examples). In isolation these coordinate data are not especially useful, but when coupled with event data, such as when a shot, pass, or turnover occurs, these player locations provide valuable insight into how individual and coordinated player actions impact game outcomes.

Both Gudmundsson and Horton (2016) and Goes et al. (2020) provide useful, in-depth reviews of spatio-temporal analysis in team sports, but here we highlight some of the research done with player tracking data in order to provide the reader with a sampling of the kinds of questions being considered and methodologies being used to answer them. In some cases, the purpose of this research is simply to use the data to describe player behavior and connect it to more generalized sport concepts. For example, tracking data was used to develop shot charts and new defensive metrics for the National Basketball Association (NBA) (Goldsberry, 2012; Goldsberry and Weiss, 2013); to identify player roles, classify formations, and identify teams in soccer and field hockey Bialkowski et al. (2014); Lucey et al. (2013b); Bialkowski et al. (2015); to estimate optimal defensive positioning in soccer using deep imitation learning (Le et al., 2017); and to build a queryable database of basketball plays by dividing player tracks into segments and treating them as words in a topic model Miller and Bornn (2017). Other researchers have also sought to provide insight through a combination of a queryable database and visualization, in American football using Bezier curves and mixture models (Chu et al., 2019), in soccer by constructing descriptive features (Andrienko et al., 2019), and also in basketball by treating player tracks as images and applying a convolutional autoencoder to them (Nistala and Guttag, 2019).

Additional cases of descriptive analysis based on player tracking data abound. Low-dimensional, interpretable, spatial basis functions for shooting in the NBA were learned using non-negative matrix factorization (Miller et al., 2014). Neural network architectures were applied to tracking data to identify events in hockey and teams in the NBA (Mehrasa et al., 2017) and to classify play calls in basketball (Wang and Zemel, 2016). Also in basketball, conditional random fields with latent spatial factors and transformed coordinate systems were used to perform one-step ahead prediction (Yue et al., 2015), the convex hull created by players on the same team was examined to see how spacing changed based on game phase and players present on the court (Metulini et al., 2018), and rebounding was decomposed into three dimensions (positioning, hustle, and conversion) and used to compare players (Maheswaran et al., 2014). Finally, ball possession sequences were used to construct features and analyzed using principal components analysis to identify differences in performance against top and bottom teams (Gonçalves et al., 2019) and Gaussian mixture models were applied to tracking data provided by the National Football League (NFL) to identify defensive pass coverage types (Dutta et al., 2020).

In other cases, the primary purpose of this research is to determine how valuable certain actions or players are. This includes the work of Cervone et al. (2016), who introduced the concept of expected possession value (EPV) by using a stochastic process model to treat the evolution of a basketball posession as a stock ticker, evaluating its expected value over time and space. The EPV framework was adapted for soccer by introducing a convolutional neural network on top of a pitch control model (Fernandez and Bornn, 2018) to determine the likelihood that a particular action occurs (Fernández et al., 2019). EPV was also extended to American Football using a long short-term memory (LSTM) neural network to account for temporal relationships between events (Yurko et al., 2020). A nice feature of the EPV framework is that it permits the evaluation of both individual actions, such as taking a shot from a certain region of the court, or individual players. In a similar approach, a Markov reward process was used to represent an entire soccer game and visual analytics were used to illustrate how the value of various actions, such as a penalty kick or throw-in, changes across the pitch (Zhao et al., 2019).

Beyond EPV or EPV-adjacent approaches, researchers have used player tracking data to evaluate goalkeepers in soccer based on the probability an opponent scores (Yam, 2019); included spatial effects within a generalized linear model to quantify and visualize how teams in the English Premier League disrupt opposing teams' offense (Bojinov and Bornn, 2016); and developed advanced defensive metrics in basketball with non-negative matrix factorization and Bayesian hierarchical modelling (Franks et al., 2015). Additional examples of using player tracking data as a tool for evaluation include predicting the probability a pass is successful in soccer using player displacement variables (Kempe et al., 2018) and a physics-based model (Spearman et al., 2017), and performing discrimant analysis on a large number of spatiotemporal variables to see which best distinguish between all-star and non-all-star caliber players in the NBA (Sampaio et al., 2015).

1.2 Thesis Structure

Rather than focus on a single area of statistics and present associated methodological or theoretical developments, the work hereafter is united by its focus on player tracking data. The current chapter serves as a brief introduction to player tracking data and Chapter 6 summarizes the thesis and provides some concluding remarks. Chapter 3 introduces the idea of broadcast tracking data and presents several of the features that make it fertile ground for future research. The remaining chapters all consist of original work. I served as primary author for all chapters, wrote all the code, and generated all tables and figures. My senior supervisor, Dr. Luke Bornn, helped with the development of the entire thesis. My co-advisor, Dr. Derek Bingham, helped with the development of Chapters 2 and 4.

1.2.1 Overview of Chapter 2

This chapter is entitled "Constructing Flexible Markov Transition Models Through Conditional Poisson Point Processes" and has been submitted to the Canadian Journal of Statistics. In it we present a method for nonparametric estimation of continuous-state Markov transition densities, relying on a Poisson process representation of the joint input-output space of the Markovian transitions. Modeling a transition density as a point process creates a general framework that admits a variety of implementations and includes some historical methods for nonparametric transition density estimation as a special case. Representing transition densities with a nonstationary point process allows the form of the transition density to vary rapidly over the space, resulting in a very flexible estimator of the transition mechanism. A key feature of this point process representation is that it allows the presence of spatial structure to inform transition density estimation. We illustrate this by using our method to model ball movement in the National Basketball Association, enabling us to capture the effects of spatial features, such as the three point line, that impact transition density values.

1.2.2 Overview of Chapter 4

This chapter is entitled “Estimating Locomotor Demands During Team Play From Broadcast-Derived Tracking Data” and has been submitted to the International Journal of Sports Physiology and Performance. Sports scientists use metrics, such as high speed running distance or acceleration load, to monitor the physical stress of participating in athletic competition and training on a player’s body. The data for these metrics is often collected with a wearable device, though optical tracking data can be used as well. In this chapter we build a model to use broadcast-derived tracking data, which is heavily censored, to estimate these external load metrics for players who are offscreen. We show that predictions on a hold-out set are highly accurate, an important benchmark in establishing broadcast tracking data as a useful source of data for sports scientists.

1.2.3 Overview of Chapter 5

This chapter is entitled “Identifying Soccer Formations in the Presence of Heavy Censoring,” and it includes methodology for identifying soccer formations that is closely related to the work of Shaw and Glickman (2019) and Bialkowski et al. (2014), but adapted to apply to broadcast tracking data, which exhibits heavy censoring that their methods do not account for. Specifically, we introduce an algorithm that relies on atomic configuration distance to align partially observed camera frames. An additional advantage of this approach is that it is always invariant to player permutation. Once aligned, we show how a finite mixture model can be used to learn formations and compare formations learned from full multicamera tracking data to those learned from broadcast tracking data.

Chapter 2

Constructing Flexible Markov Transition Models Through Conditional Poisson Point Processes

2.1 Introduction

Discrete-time continuous-state Markov chains are used to model phenomena in a variety of disciplines, including finance (Bauwens et al., 2006; De Gooijer and Zerom, 2000; Bollerslev, 1986; Engle, 1982), physical sciences (Tagle et al., 2019; Lau and McSharry, 2010), and environmental science (Gloaguen et al., 2015; Patterson et al., 2008; Wiktorsson et al., 2004), to name just a few. Frequently, the principal concern with this type of model is identifying the underlying transition density, $p(X_t|X_{t-1})$. One way to estimate $p(X_t|X_{t-1})$ is with a dynamic linear model (West and Harrison, 1997). In this approach, a relationship between sequential elements in the Markov chain is modeled, typically of the form $X_t = AX_{t-1} + \epsilon$. Often ϵ is assumed to follow a $\mathcal{N}(0, \sigma^2)$ distribution, though other distributions can be used. As long as $|A| < 1$, such a model enforces weak stationarity, ensures a smooth transition density, and offers a simple way to estimate $p(X_t|X_{t-1})$ even in regions where data is sparse or completely unavailable. However, as data becomes increasingly nonlinear or non-Gaussian, model specification and inference can be difficult. Nonparametric approaches have a similar functional form, but rather than making any distributional assumptions about the error term, it is assumed to be nonparametric and may vary as a function of X_{t-1}.

A variety of nonparametric approaches have been developed to estimate transition densities, including delta sequences (Prakasa Rao, 1978), "look-ahead" density estimation (Henderson and Glynn, 2001), and projection onto finite dimensional linear spaces (Lacour, 2008). Some of the earliest work in nonparametric estimation of the conditional transition density $p(X_t|X_{t-1})$ was done by Roussas (1969), Rosenblatt (1970), and Yakowitz (1985, 1989), each demonstrating consistency of kernel density estimators for Markov transition densities under increasingly weak conditions. They construct estimators for the conditional transition density $p(X_t|X_{t-1})$ by taking the quotient of kernel density estimates for the joint and marginal distributions, $p(X_t, X_{t-1})$ and $p(X_t)$. A more modern approach to transition density estimation is the work of DeYoreo and Kottas (2017), who use a Bayesian nonparametric mixture of bivariate normal distributions to estimate nonstationary transition densities.

We propose a general framework for transition density estimation that is specifically targeted at estimating transition densities with distinct spatial effects. The key idea in our methodology is that the set of sequential observations in a Markov chain

$$\mathcal{S} = \{(X_{t-1}, X_t) : t = 1, \ldots, T\}$$

can be treated as a realization from a Poisson point process. There are two features that make this framework a valuable contribution to the work of nonparametric transition density estimation. First, using a nonstationary process results in a flexible representation of the transition mechanism. Because of this, the degree to which transition densities for neighboring X_{t-1} values impact one another can vary over the domain, resulting in certain regions of the domain where all conditional transition densities have similar forms and other regions where the transition densities appear entirely different. The second feature, related to the first, is that point processes provide a natural mechanism to explicitly account for the effects of spatial features. Because these features often induce nonstationarity in the point process, we can use that to adjust for their resulting impact on transition density estimates. Applications where spatial features impact transition densities are the primary use case we imagine for this methodology.

Related to our framework is the topic of nonparametric density estimation, in particular the use of Poisson processes for density estimation. Brown et al. (2004) use Le Cam's distance to explicitly demonstrate asymptotic equivalency of density estimation and Poisson processes for a moderately smooth target density f. Low and Zhou (2007) expand on this work by weakening the smoothness condition required for density estimation and Poisson processes to be asymptotically equivalent. However, this existing research relies on the assumption that data are independent and identically distributed, and thus does not apply to dependent data, a gap we fill with this work. In this research, we demonstrate equivalency between transition densities and Poisson processes, but rather than providing an argument based on any particular distance metric, we show that the Markov likelihood is equivalent to the likelihood for a conditional point process.

The remainder of this paper is structured as follows: Section 2.2 provides a pedagogical motivating example. Section 2.3 illustrates the connection between a Markov model's transition mechanism and a Poisson point process and outlines how we take advantage of it to estimate transition densities. Section 2.4 illustrates key features of this approach using simulated data and Section 2.5 presents an application where point processes are used to model ball movement in the National Basketball Association (NBA). Finally, Section 2.6 provides our concluding remarks.

2.2 Illustrative Example

Consider the transition density $f(x_t|x_{t-1})$ of a stationary, continuous Markov chain $\{X_t\}$, $t = 0, \ldots, T$ with values concentrated in some finite region $\mathcal{D} \subset \mathbb{R}^F$, $F > 0$, that we assume, without loss of generality, to be the interval $[0,1]$. In addition to the conditional transition density, denote the marginal density for the chain as $f(x)$ and the joint density of (x_{t-1}, x_t) as $f(x,y)$ so that $f(x_t|x_{t-1}) = f(x_{t-1}, x_t)/f(x_{t-1})$. A naive way to analyze these data is to partition $[0,1]$ into K intervals $A_1 = [0,h)\,, A_2 = [h, 2h)\,, \ldots, A_K = [(K-1)h, 1]$ where $h = 1/K$ and calculate the transition probabilities

$$p_{ij} = \mathbb{P}\left(X_t \in A_j | X_{t-1} \in A_i\right) = \int_{A_j}\int_{A_i} \frac{f(x,y)}{f(x)} dx dy \tag{2.1}$$

for all $j, i = 1, \ldots, K$, stored in the transition probability matrix $P = (p_{ij})$. The maximum likelihood estimate for p_{ij} is $\hat{p}_{ij} = N_{ij}/N_{i\cdot}$, where N_{ij} is the number of transitions from state A_i to A_j and $N_{i\cdot} = \sum_{j=1}^{K} N_{ij}$. With these probabilities, the transition density can be approximated using the histogram estimator $\hat{f}_n(x_t|x_{t-1}) = \sum_{i=1}^{K}\sum_{j=1}^{K} \frac{\hat{p}_{ij}}{h} \mathbb{1}(x_{t-1} \in A_i)\mathbb{1}(x_t \in A_j)$. Clearly, as $K \to \infty$ (and by implication $h \to 0$), $\hat{p}_{ij} = 0$ for the majority of i and j even though the associated p_{ij} are nonzero. Thus, in order to improve the performance of the histogram estimator some kind of smoothing is necessary. One way to deal with this is to follow the recommendation of Leonard (1973) and smooth the histogram by assuming that neighboring bins are correlated via an autoregressive process.

Continuing with the nomenclature from above, let

$$\eta_{ij} = \log\left(p_{ij}\right) + \log\left(\sum_{g=1}^{K} \exp\left(\eta_{ig}\right)\right)$$

so that $p_{ij} = \frac{\exp(\eta_{ij})}{\sum_{g=1}^{K}\exp(\eta_{ig})}$. Using a Bayesian model allows us to induce autocorrelation by assuming a multivariate normal prior on $\boldsymbol{\eta}_i = (\eta_{i1}, \eta_{i2}, \ldots, \eta_{iK})'$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. For a first order autoregressive (AR(1)) process, the jkth element of the covariance matrix is equal to $\frac{\sigma_\epsilon^2 \rho^{|j-k|}}{1-\rho^2}$, with $|\rho| < 1$. Equivalently, we could write this in the more familiar dynamic linear model formulation

$$\eta_{ij} = \rho\eta_{ij-1} + \epsilon,$$

with $\epsilon \sim N(0, \sigma_\epsilon^2)$. An interesting feature of a discrete AR(1) process is that as the difference in sampling locations goes to zero (or in our case, as $h \to 0$), it becomes an Ornstein-

Uhlenbeck (OU) process, defined

$$\xi(j) = \sigma \int_{-\infty}^{j} \exp\left(-\lambda(j-s)\right) dw(s), \tag{2.2}$$

where $\lambda > 0$, $\sigma > 0$, and w is a standard Wiener process. If (2.2) is sampled at points an equal distance τ apart, the resulting sequence is equivalent to (2.2), with
$\sigma_\epsilon^2 = \frac{\sigma^2}{2\lambda}\left(1 - \exp(-2\lambda\tau)\right)$ and $\rho = \exp(-\lambda\tau)$. Therefore, as $K \to \infty$, probabilities governed by an AR(1) process become densities governed by an OU process (Arratia et al., 2014).

This relationship is significant because it demonstrates a clear connection between the definition of a correlation structure on discrete transition probabilities and continuous transition densities with smoothness governed by a stochastic process. Critically, we are applying this autoregressive process to the transition probabilities themselves rather than the data. It is important to recognize that we are not trying to sample the data at a more fine-grained level; rather, our intent is to estimate the transition probabilities at a higher resolution. A relationship such as the one detailed between a discrete first order autoregressive model and an Ornstein-Uhlenbeck process allows us to estimate transition probabilities at any resolution we desire by adjusting the size of the states $A_1, \ldots, A_k$. However, the relationship described above is only inducing correlation across the columns of the transition probability matrix, not the rows. While it is possible to smooth over both rows and columns by using a Gaussian prior on $\boldsymbol{\eta} = (\eta_{11}, \ldots, \eta_{1K}, \eta_{21}, \ldots, \eta_{2K}, \eta_{31}, \ldots, \eta_{KK})$ with an appropriate covariance matrix, such as a conditionally autoregressive prior (Besag, 1974), to our knowledge such a model does not provide the clear theoretical relationship to a continuous stochastic process present in the previous case where correlation is induced across a single dimension.

In this work we borrow strength across the rows and columns of the transition matrix P by reparameterizing the Markov model as a Poisson point process in the combined input/output space. As in the AR(1)-OU case, we will take advantage of a relationship between assumptions about the structure of discrete states and a stochastic process that allows us to easily step between completely continuous or discrete states of any resolution.

In the next section we detail three relationships that provide intuition for why Poisson processes can be used to model transition densities: first, equivalence between the likelihood for a Markov model and the multinomial distribution; second, the multinomial-Poisson transformation; and third, the connection between the Poisson distribution and a Poisson point process.

2.3 Moving from a Markov chain to a Poisson point process

2.3.1 Multinomial representation of Markov chain

Consider the setup outlined at the beginning of Section 2.2. For sake of simplicity, we continue to assume that the chain $\{X_t\}$ is univariate on the interval $[0, 1]$ but note that the following arguments easily extend to higher dimensional chains. Let the transition probabilities p_{ij} be defined as in (2.1) and let $N_{ij} = \sum_{t=1}^{T} \mathbb{1}\left[x_{t-1} \in A_i, x_t \in A_j\right]$. Given the initial observation X_0, the conditional likelihood for the Markov chain is

$$L(P) = \prod_{t=1}^{T} P\left(X_t \in A_j | X_{t-1} \in A_i\right) = \prod_{i=1}^{K} \prod_{j=1}^{K} p_{ij}^{N_{ij}},$$

which is proportional to the product of K independent multinomial likelihoods (Rajarshi, 2013). Therefore, we can estimate transition probabilities by assuming that transitions from state i are realizations of a multinomial$(N_{i\cdot}, p_{i1}, \ldots, p_{iK})$ distribution for all i, where $N_{i\cdot} = \sum_{j=1}^{K} N_{ij}$.

2.3.2 Multinomial-Poisson transformation

Suppose $\mathbf{y} = (y_1, \ldots, y_M)$ are independent Poisson random variables with means $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_M)$. The joint distribution of $\mathbf{y}$ factorizes into the product of a multinomial distribution and a Poisson distribution over $n = \sum_{j=1}^{M} y_j$, i.e.,

$$f(\mathbf{y}) = \prod_{i=1}^{M} f(y_i) = \text{Poisson}\left(n|\Lambda\right) \text{Multinomial}\left(\mathbf{y}|\boldsymbol{\alpha}, n\right).$$

Here $\Lambda = \sum_{j=1}^{M} \lambda_j$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$ where $\alpha_i = \lambda_i / \sum_{j=1}^{M} \lambda_j$. Birch (1963) showed that under the constraint $n = \Lambda$, the maximum likelihood estimate for $\boldsymbol{\alpha}$ is equivalent whether we maximize over the multinomial density or the product of independent Poisson densities (see also Palmgren (1981); Baker (1994)). The connection to modeling Markov transitions follows directly; if we assume $N_{ij} \sim \text{Poisson}\left(\lambda_{ij}\right)$ then we can estimate the underlying transition probabilities by letting $p_{ij} = \lambda_{ij} / \sum_{j=1}^{K} \lambda_{ij}$.

2.3.3 Poisson point process

Our review of Poisson point processes is necessarily brief, but the interested reader is directed to Møller and Waagepetersen (2004) or Banerjee et al. (2015) for a more thorough exposition. The likelihood for the Poisson point process is

$$L(\Lambda(\mathbf{s}), s \in \mathcal{D}; \mathbf{s}_1, \ldots, \mathbf{s}_n) = \prod_i \Lambda(\mathbf{s}_i) \exp\left(-\Lambda(\mathcal{D})\right), \tag{2.3}$$

where $\Lambda(\cdot)$ is the point process intensity and $\Lambda(\mathcal{D}) = \int_{\mathcal{D}} \Lambda(\mathbf{s}) ds$. Suppose we partition $\mathcal{D}$ into a series of discrete regions $B_1, \ldots, B_M$. Conditional on the intensity function, if two regions $\mathcal{B}_1$ and $\mathcal{B}_2$ are disjoint, then $N(\mathcal{B}_1)$ and $N(\mathcal{B}_2)$ are independent Poisson random variables. Due to this property, it follows that the likelihood over this partition is:

$$\prod_m \exp\left(-\Lambda(B_m)\right)\left(\Lambda(B_m)\right)^{N(B_m)}/N(B_m)!.$$

As the partition grows increasingly fine, $N(B_m) = 1$ or 0, depending on whether or not there is an observation in B_m, and in the limit we reach (2.3) (Banerjee et al., 2015).

To illustrate the connection to Markov transition density estimation, let (x_{t-1}, x_t) be successive elements of our Markov chain for arbitrary t. To estimate the conditional transition density we would set

$$p(x_t|x_{t-1}) = \frac{\Lambda(x_{t-1}, x_t)}{\int \Lambda(x_{t-1}, \xi)d\xi} \tag{2.4}$$

which is a valid density. This becomes clear when we consider that intensity functions can be written as $\Lambda(\cdot) = \delta f(\cdot)$, where $\delta = \Lambda(\mathcal{D})$ is the expected number of points observed in $\mathcal{D}$ and $f(\cdot)$ is a density function, so that

$$\frac{\Lambda((x_{t-1}, x_t))}{\int \Lambda((x_{t-1}, \xi))d\xi} = \frac{\delta f(x_{t-1}, x_t)}{\int \delta f(x_{t-1}, \xi)d\xi} = \frac{f(x_{t-1}, x_t)}{f(x_{t-1})}.$$

Comparing (2.4) with the transition probability estimation outlined in Subsection 2.3.2, we can see that this is simply the continuous extension of $\lambda_{ij}/\sum_{j=1}^{K}\lambda_{ij}$.

Note that estimation of the point process intensity frequently necessitates discretization of the space for computational reasons. There is an interesting duality between discrete representations of the density created by binning the space and low-rank computational methods, both of which result in a coarsening of the transition surface. The benefit of the continuous representation we have described is that it allows the user to leverage all existing work for modeling point processes nonparametrically, making any potential coarsening a choice made according to the preference of the modeler rather than something intrinsically built-in to the method.

2.3.4 Model Choice and Inference

In summary, we have created a general framework wherein one can estimate continuous Markov transition densities in two steps: first, assume that sequential elements of the Markov chain (X_{t-1}, X_t) are realizations from a Poisson point process (note that X_{t-1} and X_t can be real or vector valued) and estimate the corresponding point process intensity function $\Lambda((X_{t-1}, X_t))$. Second, calculate the conditional transition density for fixed X_{t-1} as $p(X_t|X_{t-1}) = \Lambda((X_{t-1}, X_t))/\int \Lambda((X_{t-1}, \xi))d\xi$. Because our estimate of the transition density is a valid density, variates from it can be easily simulated using any method that allows for sampling from a nonstandard distribution, i.e., rejection sampling (von Neumann, 1951). At this point, it is also worth emphasizing that Poisson processes are only fit over a fixed region. In the basketball application below, selecting this region is trivial since the transition space is restricted to the basketball court. In cases where there is no obvious spatial boundary, we suggest simply using the range of the observed data, which is reasonable for chains that are stationary and concentrated within some finite region.

A key attribute of using Poisson processes to estimate transition densities is that the intensity function can be estimated using any method appropriate for Poisson point processes. As a result, one contribution of this paper is that our framework includes the transition density estimators of Roussas (1969) and Rosenblatt (1970) as a special case. To see this, consider the Poisson point process kernel density estimator introduced by Diggle (1985). He proposes to estimate the intensity function by

$$\hat{\Lambda}(\mathbf{s}) \equiv \frac{1}{p_b(\mathbf{s})} \sum_{i=1}^{m} K_b(\mathbf{s} - \mathbf{s}_i),$$

where $K_b(\cdot)$ is a kernel function with bandwidth b (for recommendations on selection of this bandwidth parameter see Diggle (2014), pg.86-87), and $p_b(\mathbf{s})$ is an edge correction that scales the intensity function to integrate to the appropriate count. If $p_b(\mathbf{s})$ is replaced by n, then $\hat{\Lambda}(\mathbf{s})$ is equivalent to the joint density estimator used by Roussas (1969) and Rosenblatt (1970). When the conditional transition density is estimated by taking the quotient of the joint and marginal densities (or intensities) the scaling constants, whether $p_b(\mathbf{s})$ or n, cancel out so that the point process estimator and traditional kernel density estimator for the transition density are equivalent.

The fact that a kernel density estimator of the transition density *is* a Poisson process estimator as well serves to illustrate a significant feature of this framework. Namely, one can increase flexibility in the form of $p(X_t|X_{t-1})$ by modeling the transition density with a nonstationary Poisson process. The kernel density approach pioneered by Roussas and Rosenblatt is an example of using an isotropic stationary point process to model the conditional transition density, because the correlation strength (as determined by the bandwidth) is the same throughout the domain and is independent of direction. As a result, although this nonparametric estimate does allow the transition density to vary as a function of X_{t-1}, it forces it to evolve smoothly at a constant rate. Alternatively, if a nonstationary process is used to model the transition density then the rate at which the transition density evolves as a function of X_{t-1} can vary over the input space, allowing the potential for neighboring X_{t-1} values to produce conditional transition densities with radically different shapes.

In the following sections we demonstrate the use of several different estimators for Poisson point processes, including this kernel estimator. An additional point process model we implement is the log Gaussian Cox process, which allows us to account for additional structural information and covariates when estimating the intensity function. Simply put, a log Gaussian Cox process is a Poisson point process with $\Lambda(s) = \exp(Z(s))$, where $Z(s)$ is a Gaussian process (for a fuller treatment see Møller and Waagepetersen (2004)). It is possible to account for structural information in the process by regressing the mean of $Z(s)$ on relevant covariates or through modeling choices related to the covariance function. Particularly, the log Gaussian Cox process allows us to increase the flexibility of $\Lambda(X_{t-1}, X_t)$ by using a nonstationary model for $Z(s)$ (Risser and Turek (2019) provide an overview of several prominent methods for nonstationary Gaussian processes). For the examples in Section 2.4, we fit both stationary and nonstationary log Gaussian Cox processes through integrated nested Laplace approximation (INLA) (Rue et al., 2009; Lindgren and Rue, 2015). For the application in Section 2.5 we model the transition density with a nonstationary log Gaussian Cox process by combining dimension expansion (Bornn et al., 2012) with process convolutions (Higdon et al., 1998).

2.4 Application to Simulated Data

In this section we briefly consider two simulated data sets. The first is $T = 1000$ steps of a first order autoregressive Markov chain with

$$X_t | X_{t-1} \sim \mathcal{N}(\rho X_{t-1}, \sigma^2),$$

where we set $\rho = 0.9$ and $\sigma = 0.15$. We compare three different methods for transition density estimation applied to this data: a mixture model, a log Gaussian Cox process, and a kernel density estimator. Note that the the latter two methods are both implementations of the point process approach. The mixture model was implemented by constructing a Bayesian model according to the specifications in DeYoreo and Kottas (2017) on which Markov Chain Monte Carlo was run until convergence, after which 250 posterior draws were used to construct estimates. Details for fitting the kernel density estimate are fairly straightforward (see Venables and Ripley (2010) for reference). Various bandwidth selectors for the kernel density estimates were considered, but the one that minimized the pointwise mean square error between the density estimates and true densities was the normal reference bandwidth (Silverman, 1986) so that is used throughout this section.

In order to fit the log Gaussian Cox process, we rely on INLA. Simpson et al. (2015) demonstrated convergence for INLA when used to model log Gaussian Cox processes, making it a suitable method for this application. The latent Gaussian surface $Z(s)$ is approximated using a stochastic partial differential equation which is equivalent to fitting a Gaussian process with a Matérn covariance function (Lindgren et al., 2011). We fix the smoothness, α, of the covariance function at $\alpha = 2$ and use a joint penalized complexity prior (Simpson et al., 2017) on the range, ϕ, and variance, δ^2, parameters of the covariance function. For this simulated dataset, we specify the hyperparameters of the penalized complexity prior indirectly by stipulating that the probability that $\phi < 0.01$ and the probability that $\delta^2 > 10$ both equal 0.05. For this application there is no clear spatial boundary, so we define the domain for the point process to be the square formed by $(-w, w) \times (-w, w)$,

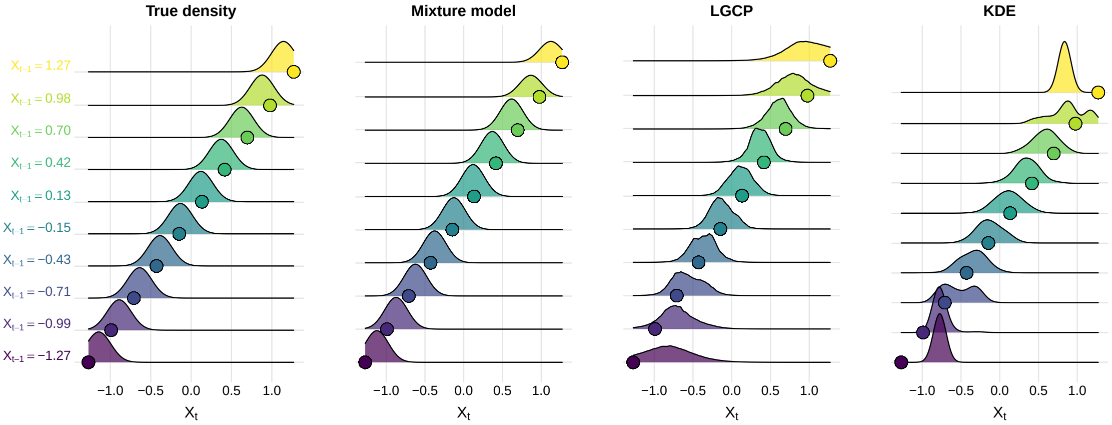


Figure 2.1: Estimates for the true conditional transition surface (left) for the first simulated data set from the various methods. The mixture model estimate is nearly identical to the true surface, while the log Gaussian Cox process provides a more accurate estimate than the kernel density method.

Method	RMSE	Coverage
Mixture model	0.17	99.5%
Log Gaussian Cox process	0.29	95.7%
Kernel density estimate	0.45	70.9%

Table 2.1: Pointwise RMSE and coverage values for each of the four methods when estimating the transition density for the first order autoregressive model.

where $w = \max(|x_t|) + 0.1$. We fit the model using the R-INLA package, available at `http://www.r-inla.org` (Martins et al., 2013).

These methods can be compared visually by examining Figure 2.1, which depicts estimates of the true transition density for each approach at 10 different values of X_{t-1}. In addition to visual inspection, we perform a numerical comparison by considering the collection of points included in the intersection of a 200 x 200 lattice with the convex hull created by the generated data, and calculating the pointwise root mean squared error (RMSE) and coverage for pointwise 95% uncertainty intervals at each of these locations. Because the mixture model and log Gaussian Cox process are fit using Bayesian methods, these intervals are estimated from posterior draws. We obtain uncertainty estimates for the kernel density estimates via a stationary bootstrap for dependent data (Politis and Romano, 1994; Politis and White, 2004) and follow the recommendation of Hall (1992) and undersmooth in order to reduce the impact of bias on the bootstrapped variance estimates. RMSE and coverage values for this first example are given in Table 2.1.

In the next example we demonstrate one way that this approach can explicitly account for spatial structure. Suppose that we have a Markov chain on the unit interval where the conditional transition density is dependent not only on the previous value, but also on where in the interval the origin value lies. Specifically, let the transition density be the piecewise

Method	RMSE	Coverage
Mixture model	0.29	86.9%
Log Gaussian Cox process	0.17	88.0%
Kernel density estimate	0.28	90.8%

Table 2.2: Pointwise RMSE and coverage values for each method when estimating the piecewise transition density.

function

$$f(x_t|x_{t-1}) = \begin{cases} 0.3\mathcal{N}_{[0,1]}(x_t|x_{t-1}, \sigma^2) + 0.7\mathcal{N}_{[0,1]}(x_t|x_{t-1} + 0.5, \sigma^2), & x_{t-1} \leq 0.5 \\ 0.5\mathcal{N}_{[0,1]}(x_t|x_{t-1}, \sigma^2) + 0.5\mathcal{N}_{[0,1]}(x_t|x_{t-1} - 0.5, \sigma^2), & x_{t-1} > 0.5 \end{cases}.$$

Here $\mathcal{N}_{[0,1]}(\cdot)$ represents the normal distribution, truncated to the interval $[0, 1]$. Figure 2.3 depicts the true conditional transition density surface for this chain, clearly illustrating the piecewise nature by the distinct boundary across the center of surface. The boundary in this example is somewhat contrived, but Section 2.5 provides a real-world illustration of how spatial features can impact transition densities. The log Gaussian Cox process allows us to account for the boundary by enforcing a border constraint at $X_{t-1} = 0.5$ (in this case, through the construction of the basis functions used in the integrated nested Laplace approximation). We use the same estimation setup as in the previous example, though for the kernel density estimate the normal reference bandwidth is replaced with the bandwidth estimator of Sheather and Jones (1991), as it improved estimates. Plotted estimates are depicted in Figures 2.2 and 2.3, with RMSE and coverage values provided in Table 2.2.

In the first example, the mixture model has the lowest RMSE and nearly perfect coverage, followed by the log Gaussian Cox process, which has an RMSE almost twice as large but with comparable coverage. Figure 2.1 reveals that at the boundaries of the observed data, the point process estimators fare poorly, with the kernel density failing to even capture the origin location. However, as seen in Table 2.2, the second example demonstrates a case where the point process estimators perform much better. The kernel density estimate has equivalent RMSE and slightly better coverage than the mixture model. By explicitly accounting for the boundary at $X_{t-1} = 0.5$ in the log Gaussian Cox process, we achieve an RMSE that is almost half that of the mixture model or kernel density estimate. Of particular interest is how the various estimators perform near this boundary. Figure 2.4 shows that the log Gaussian Cox process is able to successfully capture the change in direction of the conditional transition density on either side of the boundary, whereas the other two methods appear to average across the boundary and do not accurately estimate either density.

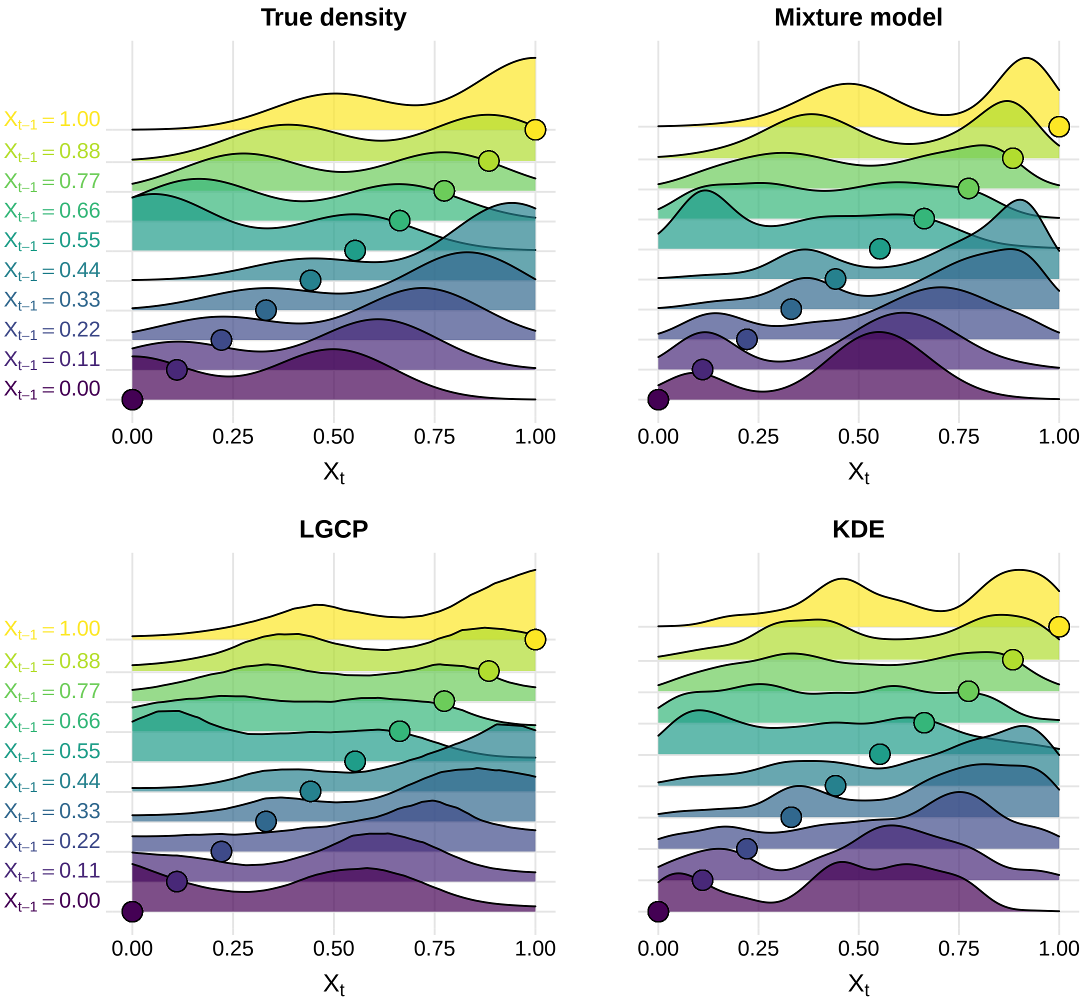


Figure 2.2: Plots of conditional transition densities for a Markov chain with a boundary at $X_{t-1} = 0.5$. The true conditional transition density from which the data was generated (upper left), mixture model (upper right), log Gaussian Cox process (lower left), and kernel density (lower right) estimates of the transition density surface.

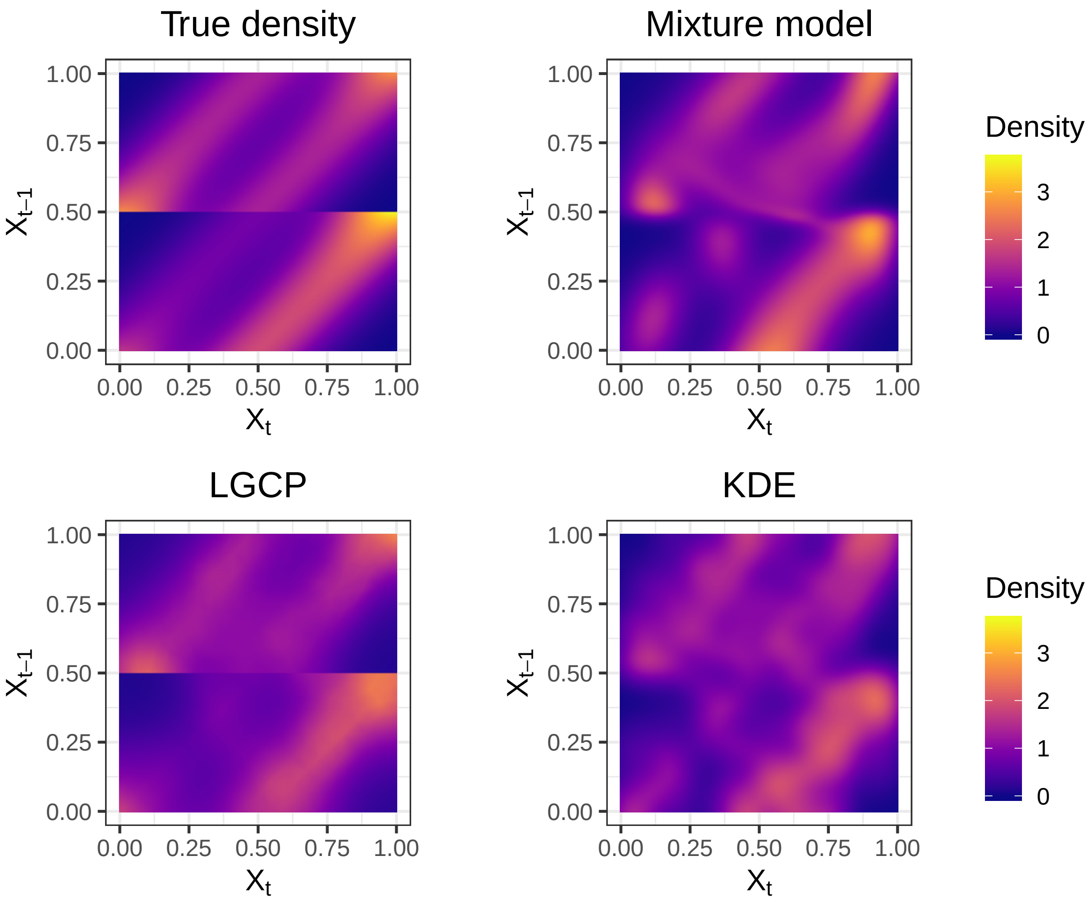


Figure 2.3: Full conditional transition surface estimates for a Markov chain with a boundary at $X_{t-1} = 0.5$. Clearly the log Gaussian Cox process estimate captures the effect of the boundary.

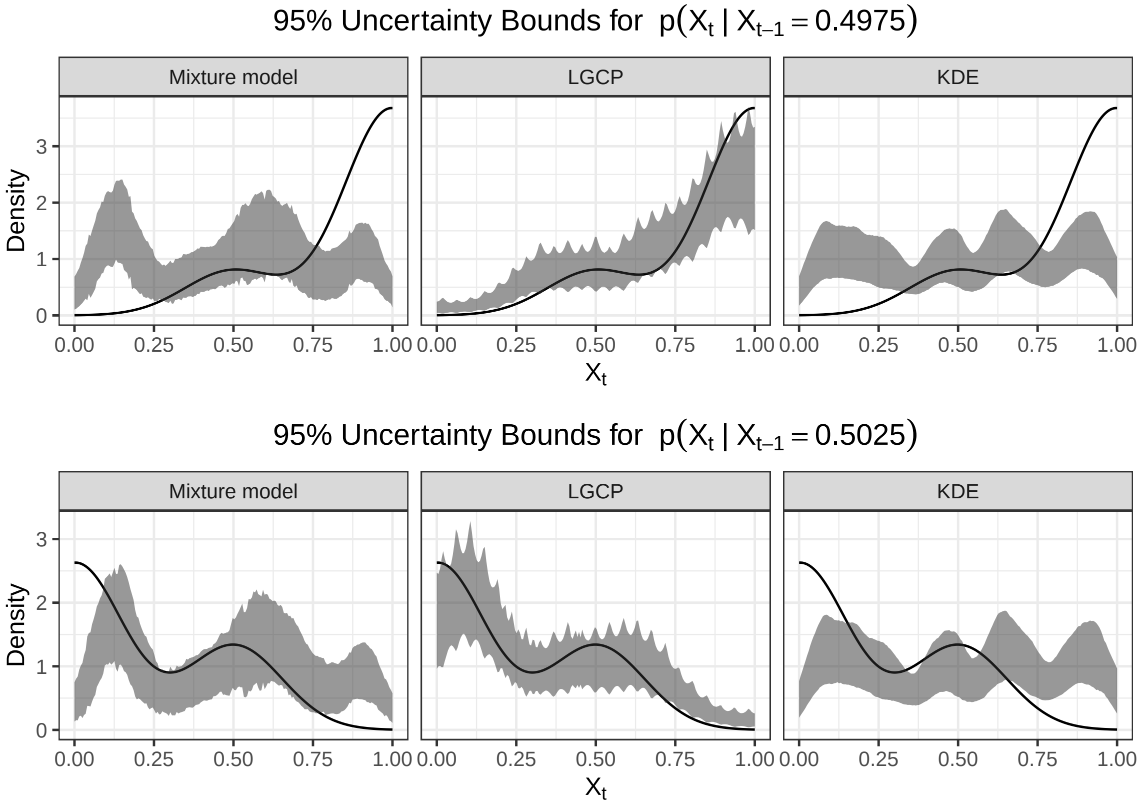


Figure 2.4: Uncertainty estimates for the conditional densities on either side of the boundary at $X_{t-1} = 0.5$. Because we can explicitly account for the boundary using a nonstationary Gaussian process, the log Gaussian Cox process estimate is able to capture the change in form of the transition density.

2.5 Modeling Ball Movement in the National Basketball Association

The National Basketball Association (NBA) collects two dimensional coordinate data for the ball and all 10 players on the court at a rate of 25 measurements per second. Event data is recorded in addition to location information, indicating when an action such as a dribble, pass, or shot occurs. Our aim is to use this data to better understand how each team moves the ball around the court and to identify differences between teams.

We analyze data from all 1230 regular season games for the 2014-15 NBA season, thinned so we only use ball locations where a pass is recorded, leaving us with 504,429 transitions. The movement of the ball during each possession is assumed to be a first-order Markov chain with shots, fouls, and turnovers serving as absorbing states. By considering each pass in terms of its origin and destination locations, we can model this as a four dimensional nonstationary log Gaussian Cox process (one dimension for each x and y location in the origin and destination) using a process convolution model (Higdon et al., 1998). The key idea for a process convolution model is that the latent Gaussian process $Z(s)$ can be replaced with a basis function expansion that relies on a smoothing kernel. In this analysis, we use a truncated normal kernel because a compact kernel creates a sparse matrix of kernel weights, increasing computational feasibility. In order to further improve computational performance, we assume that the covariance function is separable in the origin and destination dimensions, and divide the court into 1.5×1.5 foot squares, resulting in $n = 1088$ grid cells and n^2 origin-destination grid cell combinations. This grid cell size is somewhat arbitrary, but provides a good balance between computational feasibility and a high level of resolution for the transition surface. In order to calculate the process convolution weights, we place $k = 50$ kernel locations in a hexagonal grid over the court region, resulting in k^2 kernel locations in the origin-destination space.

In addition to the nonstationarity in X_{t-1} captured by the process convolution model, we also account for a systematic discontinuity induced by the three point line. In basketball, if a player makes a shot while standing behind this line, the made basket is worth three points instead of two. As such, we anticipate transitions to the area directly behind the three point line to occur with higher probability than transitions to the area directly in front of the three point line. We account for the effect of this line by adding a latent dimension to each pass location that is 0 if the ball is in front of the line and z if the ball is behind it (Bornn et al., 2012).[1] The ability to easily include this "basketball-intelligent" information highlights an advantage of point process transition density estimation over alternative methods. We determine the value of z and the standard deviation for the truncated normal kernel via

[1] Because we are working with ball locations rather than player locations, and a player can be standing behind the line even if the ball is not, we dilate the locations outwards from the hoop a small amount to account for this discrepancy.

grid search, using the out-of-sample log-likelihood as our objective. For the two teams we consider in this section, the Cleveland Cavaliers and Golden State Warriors, the optimal values for z were 5.5 and 4 feet, respectively. The estimated standard deviation for the kernel is 7.

Figure 2.5 shows transition density surfaces for the 2014-15 Cleveland Cavaliers and Golden State Warriors from three different locations on the court. Because we cannot visualize a four dimensional surface, we examine the surface conditioned on a specific transition origin location, indicated in the plots by the white dot (an interactive transition surface is available in the supplementary materials). Note the clear demarcation at the three point line as a result of the latent dimension expansion. There are some clear differences, but in order to compare the surfaces more easily, we subtract the surface for Golden State from the surface for Cleveland so that positive values indicate higher transition density for the Cavaliers and vice versa. The difference surface for one location is depicted in Figure 2.6. The majority of the locations in the plot have values of nearly zero, indicating negligible difference in transitions to those locations. However, examination of the difference surface shows that the Cavaliers have a higher probability of passing the ball beyond the three point line than the Warriors. At first glance, these differences may appear small enough to be inconsequential, but this is primarily because the normalization is occurring over such a large area. If we integrate this surface over the region beyond the discontinuity we get 0.164, which can be interpreted as, "for every 100 possessions that start at the white circle the Cavaliers pass it beyond the three point line approximately 16 more possessions than the Warriors." The nature of basketball is such that a few points per one hundred possessions often means the difference between winning and losing, making this a consequential difference.

2.6 Discussion

In this paper we have shown that Markov transitions can be accurately represented using conditional Poisson point processes. By presenting a general framework rather than a specific algorithm, there is significant flexibility in regards to implementation, allowing users to estimate transition densities using the point process model that best suits the application at hand. Additionally, nonstationary Poisson point processes result in flexible transition density estimators and provide a natural mechanism to leverage available spatial structure, as demonstrated by the three point line in our basketball application. Finally, a key advantage of this method is its conceptual simplicity. The linkage between Markov transition densities and Poisson point processes is clear and relatively easy to understand, making this approach straightforward to use in practice.

There are some promising opportunities for future research. Currently our method assumes that transition densities are temporally homogeneous, an assumption which will often

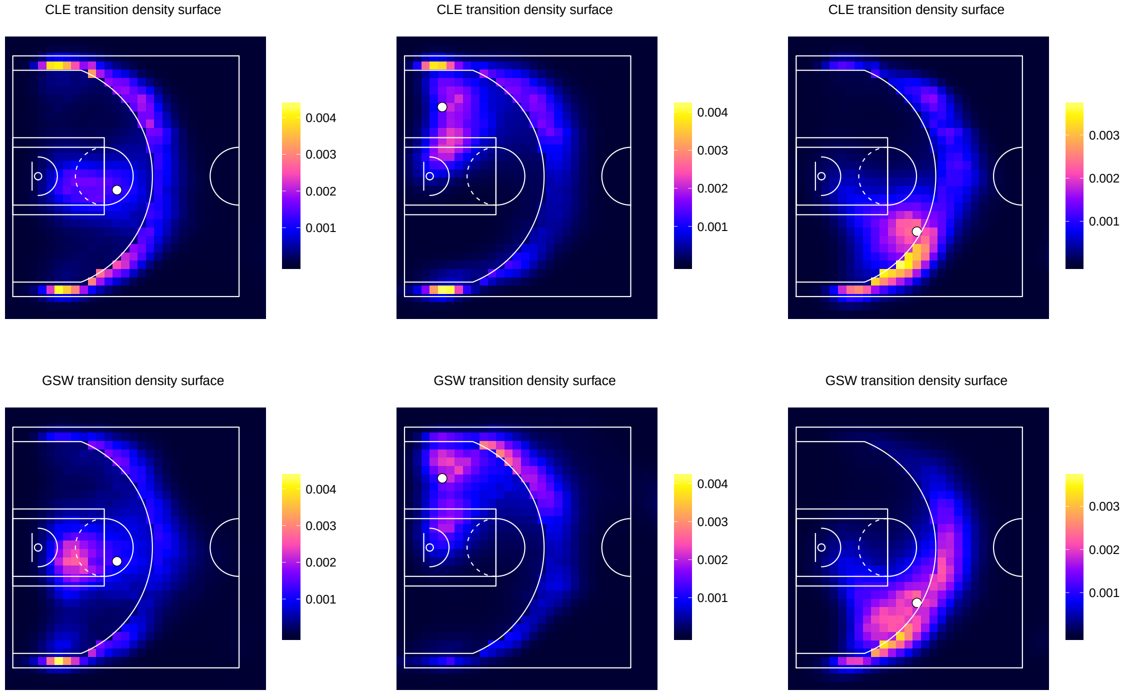


Figure 2.5: Transition density surfaces for the Cleveland Cavaliers (top row) and Golden State Warriors (bottom row), originating at three different locations indicated by the white dots. Note the clear demarcation just in front of the three point line induced by the dimension expansion.

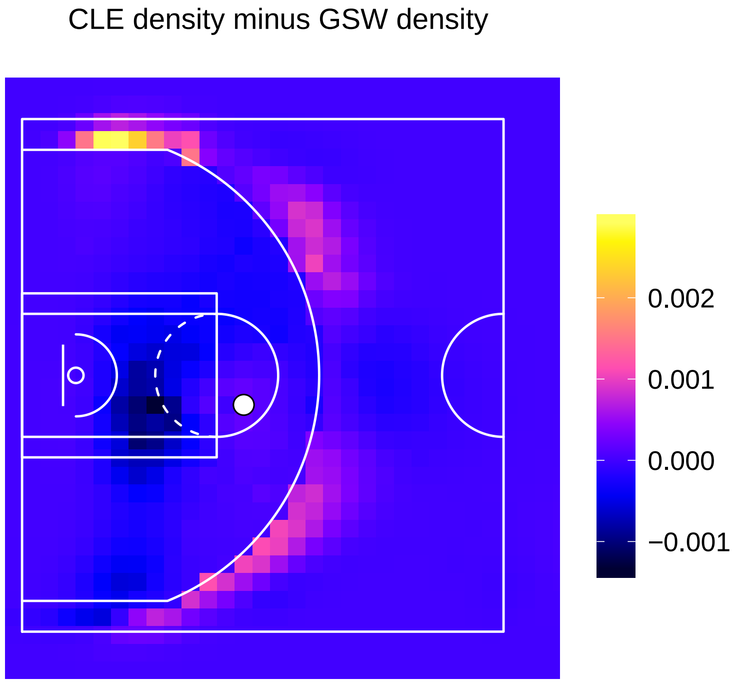


Figure 2.6: Difference surface between the Cleveland Cavaliers and Golden State Warriors. This is calculated by subtracting the surface for the Warriors from the surface for the Cavaliers, so negative values indicate areas for which the Warriors have higher transition densities, while positive values indicate areas where the Cavaliers have higher transition densities.

be violated. More work needs to be done to adapt this method to account for time-varying transitions. Another avenue for future research is scalability; representing transitions as a point process requires doubling the dimensionality of the transition space. We successfully fit a four dimensional point process in this paper, but applying this method in six dimensions and beyond poses an interesting challenge.

This methodology should be immediately impactful in any area of research where nonparametric estimation of transition densities is of interest, and will be of particular value in contexts with complex spatial domains, such as ecology or sports. Extending the connection between nonparametric density estimation and point processes to include dependent data was hitherto unexplored and the ease with which spatial features can be accounted for presents an exciting development in this area.

Chapter 3

Broadcast Tracking Data

The previous chapter introduced a technique for analyzing tracking data in which complete player tracks were observed. Because this data is generated by installing multiple cameras in an arena and converting the uninterrupted video to coordinate data, we refer to it as multicamera tracking data. The subsequent chapters present analyses of coordinate data which is generated by converting a traditional broadcast video feed, like the one you would see if you were to watch a basketball game on ESPN, for example, to player locations. To differentiate this type of tracking data from multicamera tracking data, we refer to it simply as broadcast (and occasionally broadcast-derived) tracking data.

Broadcast tracking data will never have the fidelity that multicamera tracking data can provide, but despite that shortcoming, it has two features that make it an exciting area of research. The first is availability. Because broadcast tracking data can be generated from a traditional video feed, it can be produced for leagues and organizations that cannot afford the multi-camera installation currently required to produce tracking data. The second is that the techniques used to generate broadcast tracking data can be used to process historical video. Sports fans and analysts love to compare the stars of today to hall of fame athletes from the past, and having access to historical tracking data would provide a fascinating angle to these ongoing debates. Fan interest aspect aside, historical tracking data could potentially provide new insight into why certain players, tactics, or strategies were effective in the past, and create new opportunity for analysts and executives.

3.1 Generating Broadcast Tracking Data

In general, the process to produce player tracking data from broadcast video consists of four steps: first, video segmentation; second, construction of a homography, or isomorphic mapping, between the locations in the video and the locations on the playing field; third, detection of players and construction of tracks; and finally, player identification and linking of players across the entire game. For a thorough review of generating tracking data from soccer video, we recommend Manafifard et al. (2017). Here, we provide a brief overview of how researchers have approached each of these problems in the following subsections.

3.1.1 Video Segmentation

In order to simplify processing of broadcast video, it is necessary to divide video for an entire game into single unbroken camera shots. The process by which this is done is referred to as segmentation. Breaks in the video can occur when a highlight is shown, or when commercials occur, necessitating segmentation. Methods for detecting these breaks generally consist of quantifying how many pixels have changed from one frame to the next. It is possible to look at pairwise changes of individual pixels and specify a boundary if the total pairwise changes exceed some threshold, but this approach fails to account for objects moving within the camera window and is prone to producing false positives. A more robust approach is to consider changes in summaries for each frame, such as histograms of red, blue, and green (RGB) color channel values for each image in the sequence (Zhang et al., 1993; Hanjalic, 2002). Because different camera shots in a soccer game can have similar histogram profiles due to the dominance of the green pitch, Ekin et al. (2003) introduced an adaptive threshold for determining segment boundaries based on the zoom level of the camera shot. Lu et al. (2013b) employed a hidden Markov model with Gaussian mixture emission densities, where each observation is the RGB histogram for the frame, in order to detect camera transitions in basketball.

3.1.2 Homography Construction

Mapping player locations to coordinates in Euclidean space requires a one-to-one mapping from points in the broadcast video to locations on the court or pitch that accounts for camera angle and the size of the camera window. We refer to this mapping as a homography. With static cameras, the homography can be calculated using a camera calibration algorithm (Zhang, 2000), but the ability of a broadcast camera to tilt, zoom, and pan means that such approaches do not apply. As a result, homography estimation with broadcast video typically involves matching templates on the court or field with their known Euclidean quantities. For example, in basketball Hu et al. (2011) used pixel detection, filtering, and quadrangle candidate generation to identify the four intersecting lines that make up the free throw lane and map that to the actual dimensions of the court. Also in basketball, Lu et al. (2013b) constructed a homography by filtering out players and spectators, computing edges (meaning lines in the image, not boundaries of the camera window) in the frame using a Canny detector (Canny, 1986), and matching those edges to a model of the court using an Iterated Closest Points algorithm (Zhang, 1994). Other examples of homography construction in sports include Hess and Fern (2007), who used locally distinct feature recognition to map a football field model to video, and Gupta et al. (2011), who constructed a homography for hockey data by minimizing residual error between ellipses and lines on the rink and a projected model.

Both of the following two chapters deal with broadcast tracking data in soccer. Soccer has a couple of distinct attributes that make homography estimation uniquely challenging. First, pitch dimensions are not uniform from field to field, and so a unique homography has to be estimated for every stadium. Second, there are fewer distinctive markings on a soccer pitch than a basketball court, football field, or hockey rink, and so, particularly if the camera is zoomed in, players can be isolated in a green field with no markings, making alignment with a pitch model difficult and heavily reliant on past or future orientations. Despite these challenges, much work has been done in this area, with homographies constructed based on marks for the penalty boxes (Yu et al., 2007), by using fast refinement of poor homographies Hadian and Kasaei (2015), and by taking advantage of the fixed nature of the broadcast camera in soccer and using a random forest to estimate the pan/tilt angle based on two different points. Another recent innovation is the work of Cuevas et al. (2020), who separate unwanted edge data from the relevant pitch information, use a probabilistic decision tree to classify pitch lines and use them to calculate camera tilt, and validate hypotheses connecting the image points with the key points in the model to determine accuracy.

3.1.3 Player Detection and Tracking

After establishing a homography, the next step is to detect players and estimate tracks within each video segment. Because players frequently overlap, any successful method must be robust to occlusion. A multitude of methods exist for player detection in a multicamera setting (Baysal and Duygulu, 2016; Liu et al., 2013) but we highlight some specific methods used for broadcast tracks. Some authors use a tracking by detection approach, which includes using the Deformable Part Model (Felzenszwalb et al., 2008) to detect players in conjunction with either particle filtering (Lu et al., 2013a) or a logistic team classifier and linear location model (Lu et al., 2013b) to create tracks. Another approach is a multiple object adaptation of the CamShift algorithm (Hu et al., 2011), which identifies players by calculating centroids based on a color probablity distribution. Liu et al. (2009) used Haar cascades to detect players, clustering of color histograms for the detected regions to determine team membership, and then associated tracks with detected player locations over time using a Bayesian model. Recent developments in multi-person pose estimation (Iqbal et al., 2017) have also made it viable for player tracking in sports (Bridgeman et al., 2019).

3.1.4 Player Identification

Having generated player tracks, the next task is to link tracks for each video segment across the entire game by associating them with players. Player identification is challenging because identifying features, such as jersey numbers, are frequently covered or at a bad angle. Additionally, the resolution of broadcast video is typically lower than is available with a multicamera approach, so facial recognition techniques are not feasible (Bertini et al., 2005; Ballan et al., 2007). A common approach is to use optical character recognition to identify numbers on player jerseys (Lu et al., 2013a; Gerke et al., 2015; Li et al., 2018), sometimes augmented with additional information such as location within a spatial constellation (Gerke et al., 2017). Moving away from simple number detection, a method that considers all player features is outlined by Lu et al. (2013b), who treat a combination of visual features as weak learners, and combine them with temporal smoothing and a per-frame uniqueness constraint within a conditional random field to estimate player identities. Relatedly, convolutional neural networks can be used to identify players based on their entire bodies (Senocak et al., 2018; Lu et al., 2018), though these generally classify frame by frame and fail to take advantage of valuable temporal information.

3.2 Research on Broadcast Tracking Data

Companies like Sportlogiq and SkillCorner are actively working to provide broadcast tracking data as a commercial product. However, to date the majority of research on broadcast tracking data has dealt with its production rather than its analysis and so there is a significant need to prove that this type of data is valuable and can produce meaningful and valid results. The main feature that prevents existing methods for multicamera tracking data from being applied directly to broadcast tracking data is the issue of censoring, illustrated in Figure 3.2. Put simply, players can leave and exit the camera window, resulting in discontinous player tracks and a variable number of players on screen. Additionally, consistent player identification across these discontinuous tracks is often unreliable. Consequently, as new methods for tracking data are developed they must be able to account for missingness and its effects.

In the following two chapters, we present some applications related to broadcast-derived tracking data. In Chapter 4, we consider a sports science application of load metric estimation. Load metrics are quantities that sports scientists monitor to manage strain and stress on athletes. We show that broadcast estimates of these values are highly accurate, despite the high degree of censoring. In Chapter 5, we address a more tactical question and show that broadcast tracking data can be used to effectively estimate soccer formations.

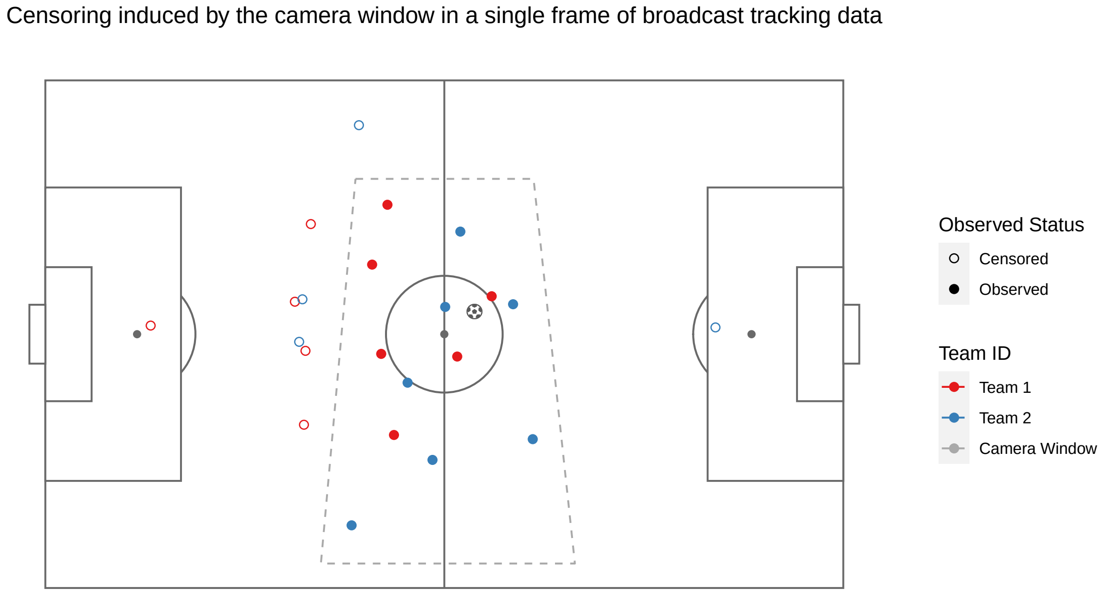


Figure 3.1: A single frame of data from a game. Multicamera tracking data would consist of all player locations (represented by both the filled and hollow circles) whereas broadcast tracking data would include only the locations within the camera window represented by the filled circles.

Chapter 4

Estimating Locomotor Demands During Team Play From Broadcast-Derived Tracking Data

4.1 Introduction

In order to reduce fatigue, prevent injury, and improve performance, sports scientists seek to monitor the physical impact that participation in training and competition has on an athlete. A number of different metrics, broadly referred to as load metrics, have been used to try and quantify the intensity of a given activity for an athlete. These metrics consist of two general categories: internal and external. Halson (2014) defines internal load as “the relative physiological and psychological stress imposed” on an athlete. Internal load measures are not treated in this work and we make no further note of them other than to mention that they exist and include, for example, individual reporting of perceived exertion, heart-rate-derived training impulse, and summated-heart-rate-zones (Borresen and Lambert, 2008; McLaren et al., 2018). External load is defined as “the work completed by the athlete, measured independently of his or her internal characteristics” (Halson, 2014). Metrics in this category include distance measures (both total distance over a training session or match and distance traveled stratified by intensity of the activity) (Coutts and Duffield, 2010; Rampinini et al., 2009; Dalen et al., 2016; McLaren et al., 2018) and acceleration-derived measures (Dalen et al., 2016; Delaney et al., 2016; McLaren et al., 2018; Nicolella et al., 2018; Boyd et al., 2011). Existing methods for capturing external load metrics consist of an athlete wearing a device, whether that be a global positioning system (GPS) (Sykes et al., 2013; Mullen et al., 2019) or local positioning system (LPS) tracker (Vázquez-Guerrero et al., 2019), or an accelerometer (Boyd et al., 2011), which can give accurate readings of instantaneous velocity and acceleration. However, it is not always possible for an athlete to wear such a device, and so multiple camera semiautomatic video tracking systems have been used as an alternative means to capture measures of external load (Gregson et al., 2010).

Initially introduced in soccer by ProZone in 1999 (Medeiros, 2017), multicamera optical tracking data has spread across a variety of sports, providing detailed location data for players multiple times per second. Much work has been done to demonstrate the validity and reliability of this type of data, beginning with Valter et al. (2006), who showed that video tracking data accurately represented velocities in a variety of controlled tests. Related work compared GPS, LPS, and video tracking systems and provided calibration equations between them, finding that optical tracking data tended to slightly overestimate distances relative to the other two and that data accuracy was impacted by both player velocities and varying pitch sizes (Buchheit et al., 2014). Other studies have corroborated the tendency of video tracking data to have slightly higher error than GPS or LPS systems when measuring velocities and accelerations (Linke et al., 2018; Pons et al., 2019), though Pons et al. (2019) also showed that errors for video tracking estimates from match play were well within an acceptable range for practical usage and Linke et al. (2018) showed that location and distance estimates from video tracking data were more accurate than those from GPS systems. More recent work has shown that later generation video tracking systems have a high degree of accuracy (Linke et al., 2020). Video tracking data has been used for motion analysis (Carling et al., 2008), and in particular has been used to examine high intensity activity (Di Salvo et al., 2009) and match to match variability of high-speed activities (Gregson et al., 2010) in the English Premier League. Broadly, tracking data is useful for sports scientists because it can be used to derive metrics related to distance, speed, and acceleration that serve as a proxy for the stress placed on a player's body as a result of their athletic performance. This source of data is most useful for match play, where data produced by wearable devices may not be available, whether due to league rules or player cooperativeness.

However, despite the distinct advantages provided by multicamera optical tracking data two issues remain: exclusivity and sparsity. This data is exclusive because obtaining it requires installation of expensive hardware and paying large licensing fees, restricting its availability to only the most elite leagues. The data is sparse in the sense that it has become widespread only in recent years, preventing historical comparison. As a result, any ability to draw conclusions about load metrics and their relationship to health outcomes is limited. Broadcast-derived tracking data has the capacity to overcome both of these problems because it allows locations to be extracted from regular broadcast video using computer vision techniques (Lu et al., 2013b). This eliminates the need to install special cameras and has the potential to provide x-y coordinate data for any game with a video feed. Currently companies such as Sportlogiq (2020) and SkillCorner (2020) are actively working to develop commercial products based on this technology and this type of data is something we expect to become broadly available in the near future.

Despite its clear value, broadcast-derived tracking data comes with one glaring issue: location data is only available for a player as long as they are observed within the camera frame. Our purpose in this paper is to assess the viability of broadcast-derived tracking data for estimation of a variety of external load metrics commonly used across sports. Specifically, our focus is on estimating load metrics during the time that a given player is offscreen. We do this by using games for which complete multicamera tracking data is available, and manually censoring observations to emulate the broadcast-derived tracking data. It is important to note that the purpose of this work is not to establish the validity of broadcast-derived tracking data in the sense of Pons et al. (2019); Valter et al. (2006); Linke et al. (2020) and others. Rather we assume that the player locations are accurate when they are observed and seek to assess the impact of the censoring inherent to broadcast tracking data. Approaching the problem in this way allows us to establish a ground truth, answering definitively, given that the broadcast tracks are accurate, whether or not broadcast-derived tracking data can be used to assess external load.

4.2 Data

Our data comes from 18 home games played by Chelsea FC in the 2014-15 English Premier League and includes information for 248 players (this number does not include goalkeepers, which are excluded from our analysis). The data contains complete location data for all players in each game at a frequency of 10 measurements per second, providing a continuous sequence of locations for each player, which we refer to as a track. Additionally, the data includes event information, which consists of the location and description of actions such as a touch, pass or tackle.

In order to emulate the broadcast-derived tracking data while retaining true offscreen values, a camera track is simulated by linearly interpolating between event locations and a 40×40 meter window is centered on the camera track. Player locations outside of the window are treated as unobserved, and metrics calculated from these censored tracks are the values predicted in this paper. Each time a player track transverses the edge of the camera window, the multicamera track is split into a new segment, referred to as a subtrack, and assigned a unique ID. This process results in 149,680 subtracks generated from an original 820 tracks, giving an average of 8315.6 (+/- 411.9) subtracks per game from a median of 45.5 (range=44-47) original player tracks per game. The median subtrack length is 17.0 m (range=0.0-830.1 m) with a median time of 8.1 s (range=0.1-556.7 s). Subtrack information is augmented by player position information scraped from transfermarkt.com, with each player classified as either a defender (n = 85), midfielder (n = 85), or forward (n = 78).

Figure 4.1 shows how the distributions of total time and total distance vary depending on whether or not players are onscreen. Defenders have the least variable distributions and also tend to spend the most time offscreen. Distributions for forwards and midfielders have larger variance and both exhibit some bimodality, though this is primarily driven by player substitutions. Midfielders travel the most distance and spend the most time onscreen,

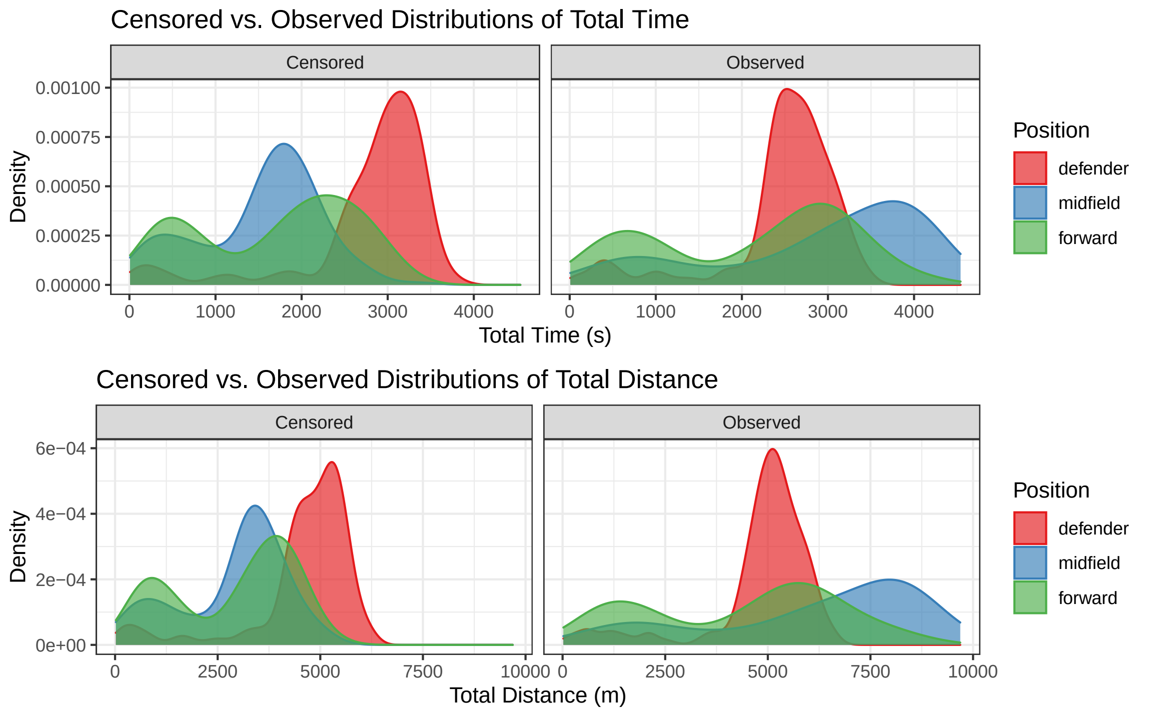


Figure 4.1: Distributions of total time and total distance for the observed and censored portions of each match.

followed by forwards. Generally, this figure shows that a player may spend anywhere from a quarter to as much as nearly two thirds of their play time offscreen.

4.2.1 Player load metrics

The suite of external load metrics considered in this work were selected because of their use throughout the literature (see, for example, Varley and Aughey (2013); Gabbett and Ullah (2012); Dwyer and Gabbett (2012); Johnston et al. (2014); Dalen et al. (2016); Borresen and Lambert (2008); McLaren et al. (2018)). Because definitions of load metrics may vary depending on the author, specific definitions for the metrics considered in this manuscript are provided in Table 4.1. Note that although this work is limited to predictions for the eight load metrics outlined in Table 4.1 that they fall into the three broad categories of distance, velocity, and acceleration derived measures. In general, any external load metric that falls into one of these categories can be calculated from broadcast-derived tracking data, though prediction accuracy for specific censored metrics should be assessed individually.

Exploratory analysis of this data and the calculated metrics reveals two patterns worth highlighting. The first is that there is a very strong correlation between the amount of censored playing time in a game and most of the censored load metrics, as shown in Figure 4.2 for total acceleration. When keepers are removed from the data, the values for Pearson's correlation coefficient between elapsed time for a censored subtrack and most of the other metrics range between 0.596 and 0.998, the exceptions being peak velocity and acceleration density. This suggests that in many cases fairly good estimates can be obtained by simply regressing the metric on censored subtrack time. The second pattern becomes clear if we assume that the censored data is missing completely at random (MCAR) (Rubin, 1974); that is, we assume that there is no relationship between the pattern of missingness and the values of the observed and censored load metrics. To illustrate, consider total distance (though the following relationship holds for the other player metrics). Under this assumption, the ratio of observed distance, D_o, to censored distance, D_c is equivalent to the ratio between observed time, T_o, and censored time T_c, or in mathematical notation,

$$\frac{D_c}{D_o} = \frac{T_c}{T_o}.$$

This in turn implies that we can estimate D_c by setting $D_c = D_o \frac{T_c}{T_o}$. Because we are simply scaling the observed metric value by the ratio of censored to observed time, we refer to this as a scaling estimator. Despite its appealing simplicity, examination of the residuals, shown for four of the metrics in Figure 4.3, reveals that the assumption that data is MCAR is incorrect. We see that residuals become increasingly negative (censored values are overestimated) as the amount of censoring increases for most of the metrics, with the exception being the amount of time spent in the slowest velocity band. The systematic differences between player movement on- and off-camera demonstrated in these plots can be summarized simply as "players move faster when on camera."

4.3 Methods

The purpose of this work is to determine whether or not broadcast-derived tracking data can be used to provide accurate estimates of external load metrics. This is accomplished by dividing the 18 observed games into two groups: 361 player-match observations from the first 13 games of the season are used as a training set to estimate model parameters and 138 player-match observations from the remaining 5 games serve as a test set on which the accuracy of model predictions can be assessed. If these predictions are highly accurate, that serves as evidence that broadcast tracking data is appropriate for load metric estimation.

Predictive accuracy is measured using root mean square predictive error (RMSPE), defined RMSPE $= \sqrt{\sum(y_i - \hat{y}_i)^2/n}$, where $\hat{y}_i$ is the predicted value for observation y_i and n is the total number of observations, and the coefficient of variation (CV), defined CV $= RMSPE/\bar{y}$, where $\bar{y} = \sum_{i=1}^{n} y_i/n$. The coefficient demonstrates how large the errors are relative to the values themselves, giving an indication of how much the overall variance in the data was reduced by the given model. In this application, the CV is useful because it allows comparison of predictive accuracy across metrics rather than simply across models, overcoming the wide range in magnitudes for the different load metrics.

Table 4.1: External load metric definitions

Category	Metric	Definition
Distance	Total distance	Sum of the distance travelled by an athlete.
	High speed distance	Sum of the distance travelled by an athlete with speed between 3.5 and 5.7 m/s.
	Very high speed distance	Sum of the distance travelled by an athlete with speed greater than 5.7 m/s.
Velocity	Time spent in velocity band $[x, y)$	Number of seconds spent with velocity (m/s), v, in the interval $x \leq v < y$. Intervals considered are $[0, 3.5)$, $[3.5, 5.7)$, and $[5.7, \infty)$, based on the work of Dwyer and Gabbett (2012).
	Peak x-second velocity	Max velocity of average velocities calculated over $x = 1, 3, 5$, and 10 second rolling windows.
Acceleration	Total Acceleration	Sum of the absolute values of acceleration at 0.1 second intervals.
	Acceleration density	Mean acceleration.
	Time spent in acceleration band $[x, y)$	Number of seconds spent with acceleration (m/s^2), a, in the interval $x \leq a < y$. Intervals considered are $[0.65, 1.46)$, $[1.46, 2.77)$, and $[2.77, \infty)$, based on the work of Johnston et al. (2014)

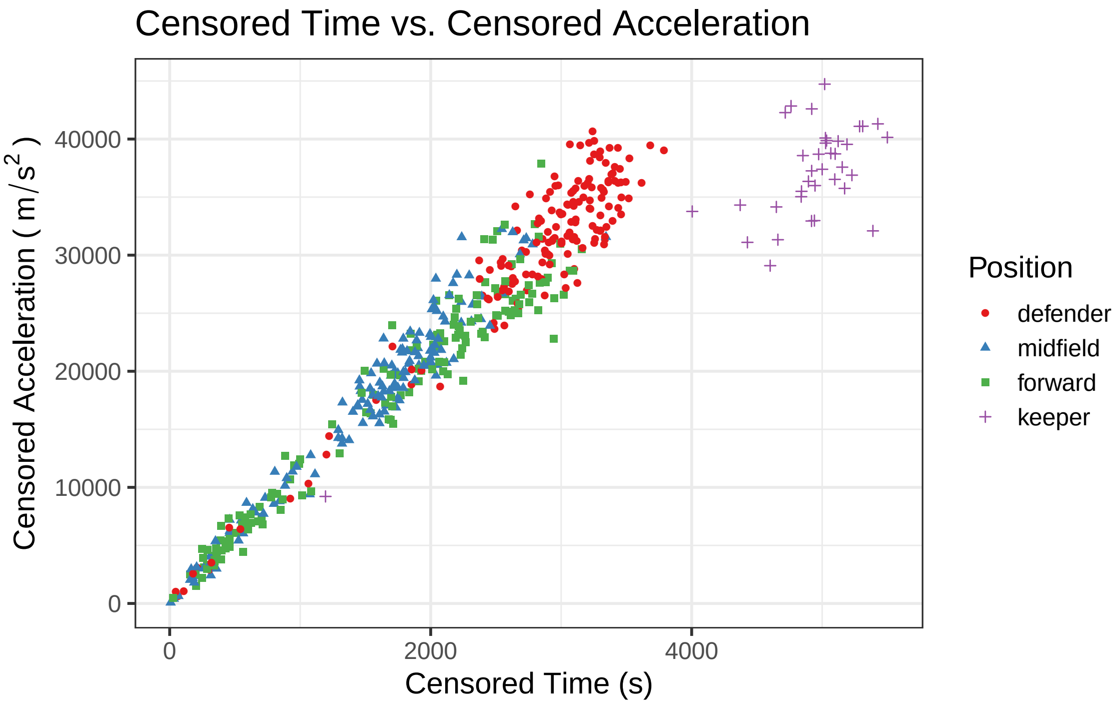


Figure 4.2: Censored time versus censored total acceleration. Note that once keepers are removed from the data, the relationship between censored time and censored total acceleration is almost exactly linear.

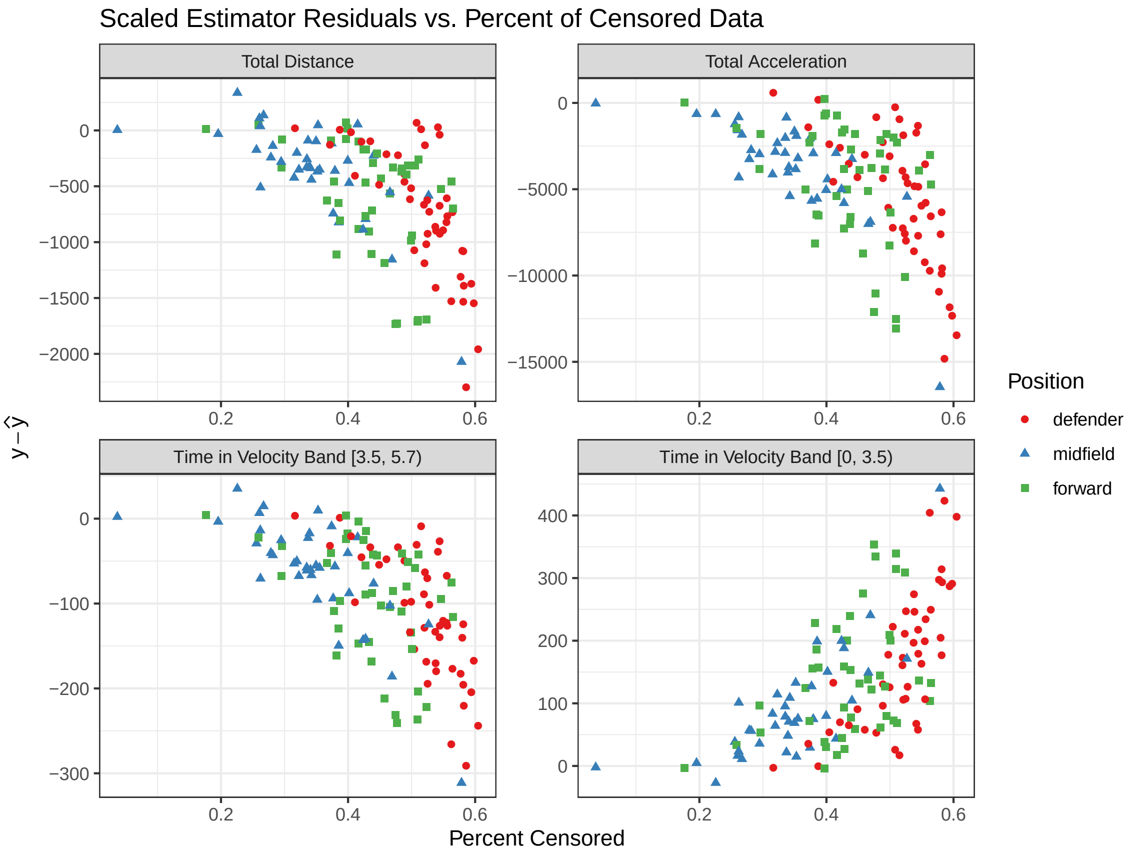


Figure 4.3: Residuals for the scaling estimator in Section 4.2.1 versus the percentage of data that is censored. The scaling estimator consistently underestimates the amount of time spent in the slowest velocity band, whereas it consistently overestimates values for the other load metrics, resulting in negative residuals. This is clear evidence that the way players move on camera differs systematically from how they move offscreen.

4.3.1 Statistical Models

A variety of statistical models are considered in the following sections. As a baseline level of comparison, a linear regression model for each load metric of the form

$$y_i = \beta_0 + \beta_1 T_i + \beta_1 x_i + \epsilon_i$$

is used, where i indexes the player-match combination, y_i is the value of the censored metric at the game level, T_i is the amount of censored time for the game, x_i is the value of the observed metric at the game level, and ϵ_i is a normally distributed error term.

The next two models considered are regularized linear regression models of the form

$$y_i = X_i \beta + \epsilon_i,$$

where X_i is a $1 \times P$ design matrix for player-match i, β is a $P \times 1$ vector of model parameters and ϵ_i is a Gaussian error term. These are regularized because β is estimated subject to the elastic net penalty, $\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1)$, where λ_1 and λ_2 are tuning parameters that affect the degree to which the L_1 and L_2 penalty terms, respectively, impact parameter estimates (Hastie, Trevor, Tibshirani, Robert, Friedman, 2009). The design matrix, X_i, for the first linear model consists of an intercept and main effects for the variables included in Table 4.2, features such as total observed distance and amount of time spent offscreen. The second linear model is similar to the first, but includes all two-way interactions in addition to the main effects. The final model we consider is a tree-based model, or random forest, which naturally accounts for nonlinear relationships between the predictors and response (Breiman, 2001).

Conveniently, the regularized regression models and random forest can all be fit using gradient boosting (Friedman, 2001) as implemented by the `xgboost` package (Chen and Guestrin, 2016) in the statistical programming language R (R Core Team, 2019). Gradient boosting is an ideal tool for this application for several reasons: it fits a model, then iterates, fitting a model on the residuals of each previous model until no further improvements can be made, then averages all of the models together, yielding very accurate predictions; it performs automatic variable selection, permitting a "kitchen-sink" approach in which all potential predictors can be considered for inclusion in the model; and finally, as implemented in `xgboost` it is extremely fast, allowing models to be fit to a large amount of data in just a short span of time.

These three different models (two linear, one random forest) are all fit at two different levels. First, censored metric values are aggregated for each player-match observation and then predicted. Second, values are predicted for each subtrack and subsequently aggregated for each player-match. Models fit at these two different levels are referred to as game and subtrack level models, respectively. A variety of predictors were constructed from the player tracks at both of these levels. For each subtrack, the x and y locations where the player left and re-entered the camera window was recorded and the Euclidean distance and time elapsed between them was calculated. Distance, velocity, and acceleration were calculated for each 0.1 second interval, and used to calculate the load metrics detailed in Table 4.1 as well as average velocity and average absolute acceleration in the 2 second intervals preceding and following each subtrack. The raw accelerations occasionally exhibit unrealistic values, with some instantaneous accelerations greater than 50 $\mathrm{m/s}^{-2}$, so in order to reduce this noise the accelerations were smoothed using a Nadaraya-Watson kernel smoother (Nadaraya, 1964; Watson, 1964). Variables considered for prediction at the game level mostly consist of the load metrics calculated for the observed portion of the game, but also include the total time in seconds that a player was censored, percent of playing time a given player was censored, and their average observed velocity. Player position is included as a predictor at both the subtrack and game levels.

A summary of the predictors included in the various models is shown in Table 4.2. All numeric variables were centered and scaled so that they have a mean of 0 and standard deviation of 1, which, for the linear models, allows the relative significance of each predictor in the model to be determined by comparing the size of its associated coefficient to the other coefficients in the model.

Examination of the 18 games in the data reveals that 100% of the peak x-second velocity values occur within the camera window, and as such, there is no need to try and estimate these values for the censored tracks. While peak velocities may not always be observed across all games, this suggests that any exceptions will be rare and so peak velocity is omitted from the subsequent analysis.

4.4 Results

The full results for each of the predicted metrics at the subtrack and game levels are shown in Tables 4.3 and 4.4, respectively. In all cases, estimating the player metrics at the subtrack level and then aggregating to obtain game level estimates outperforms making predictions purely at the game level, as seen in the lower RMSPE and CV values. For seven of the eleven player load metrics under consideration, the linear model with interactions performs the best, though the random forest has the lowest RMSPE and CV in three cases. Predicting acceleration density is the only case where the linear model with no interactions results in the best predictions.

CV values are less than or equal to 0.10 for six of the eleven models, indicating a significant reduction in standard error relative to the overall size of the response. The largest CV is 0.31, for both very high speed distance and time in velocity band $[5.7, \infty)$, both outcomes for which the nonlinear model performs the best. This result can be explained by noting that the correlations of these two metrics with censored total time are 0.599

Table 4.2: Model predictors. Inclusion for consideration in subtrack or game level models is indicated by the x. All variables that begin with "observed" are measured at the game level, so an x in the subtrack model column for "observed total acceleration" means that the sum of the absolute value of accelerations for the entire game is used as a predictor when estimating individual subtrack outcomes.

Predictor	Included at subtrack level	Included at game level
player position	x	x
offscreen time	x	
censored total time		x
offscreen distance	x	
observed total distance		x
average velocity in previous two seconds	x	
average velocity in following two seconds	x	
average absolute acceleration in previous two seconds	x	
average absolute acceleration in following two seconds	x	
observed average acceleration	x	x
observed total acceleration	x	x
observed average velocity	x	x
observed high speed distance	x	x
observed very high speed distance	x	x
observed time in velocity band $[0, 3.5)$	x	x
observed time in velocity band $[3.5, 5.7)$	x	x
observed time in velocity band $[5.7, \infty)$	x	x
observed time in acceleration band $[0.65, 1.46)$	x	x
observed time in acceleration band $[1.46, 2.77)$	x	x
observed time in acceleration band $[2.77, \infty)$	x	x

and 0.605, respectively, indicating that these predictions do not benefit from the strong relationship with censored total time that the other metrics do. Considering the RMSPE values themselves provides insight into just how well each metric is being predicted. For example, despite its CV of 0.31, the RMSPE for time in velocity band $[5.7, \infty)$ is still only 6.4 seconds. The RMSPE for total distance is 183 meters, minimal error considering the average censored distance traveled by players in each game is 3524 meters.

An examination of the residuals for a given response variable against the percent of data that is censored, shown for total distance in Figure 4.4, is illuminating. Unsurprisingly, the variability in the predictions increases with the amount of censored data, but in general the predictions appear unbiased, and even when as much as 50% of the data is missing, the range of the residuals being approximately (-500, 500) indicates a significant reduction in variability when compared to the empirical standard deviation for the response of 1506 meters. Figure 4.4 also shows a clear demarcation in the amount of censoring across the various positions, with defenders experiencing the most time off camera and midfielders spending the most time within the camera frame.

Due to the nature of gradient boosting, the ability to interpret coefficient values is limited, but a sense of the impact of certain predictors can be gained by taking the top five covariates with either the greatest importance (for the random forest) or largest coefficients[1] (for the linear models) for each model and tallying how frequently each is included. Offscreen time is included as the first or second most significant variable in the models for all eleven load metrics, a fact foreshadowed by the strong correlation noted in Section 4.2.1. Offscreen distance and position are both top variables in eight of the eleven models, while the velocity and acceleration entering and leaving the camera window are in the top five for four of the models, primarily the distance metrics. All remaining predictors occur in the top five for just three or fewer of the load metrics.

4.5 Conclusion

Broadcast-derived tracking data has the potential to greatly increase the amount of player tracking data available, but concerns about the large amount of censoring inherent to this data must be addressed in order to have confidence in its value for sports scientists. This work has demonstrated the viability of this type of data when evaluating player load in soccer. Because censored player load metrics are predicted using widely available machine learning methods rather than a custom built solution, this type of modeling is accessible and should be relatively easy to adopt in practice. Examination of RMSPE and CV values shows that in general, predictions for the various load metrics are very accurate. Some of the stratified metrics, i.e., time in velocity band $[5.7, \infty)$ and very high speed distance, have

[1] Recall that all variables were centered and scaled, making this a valid comparison.

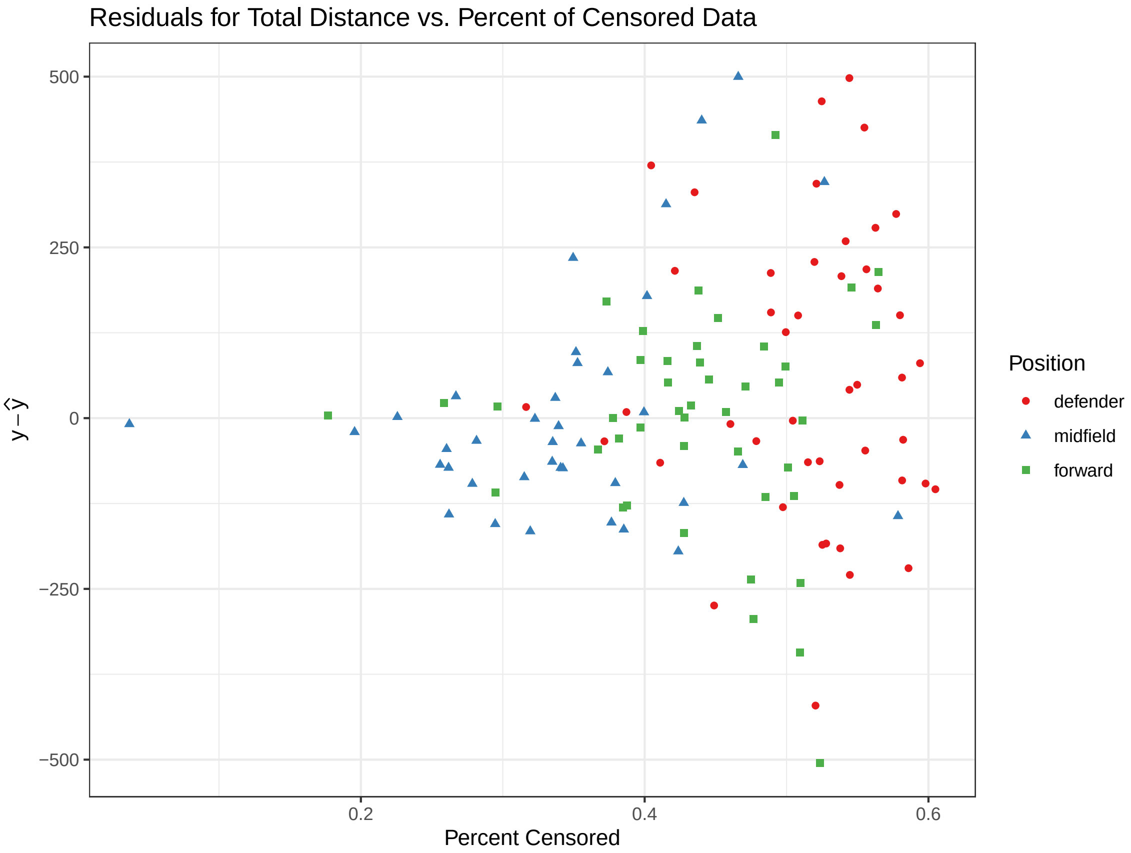


Figure 4.4: Residuals for total distance predictions versus percent of total data that is censored.

larger CV values due to the small amount of time players spend in these states, but the RMSPE in these cases is still low enough for use in a practical setting.

As this type of data is adopted, one important consideration is how estimate accuracy varies by position and across players. For example, when considering total distance, the RMSPE for defenders, midfielders, and forwards is 216, 164, and 160, respectively. This is driven primarily by differences in censoring rates, with forwards and midfielders being censored just 43.4 and 35.4 percent of the time, versus 52.5 percent of the time for defenders. Censoring rates also vary significantly from player to player, ranging from 19.7 to 60.04 percent of data censored. Accounting for differences in censoring can increase prediction accuracy and improve the efficacy of using broadcast-derived tracking data for external load metric estimation in soccer.

4.5.1 Future Work

There are two primary areas for future work. The first is to get a sufficiently large sample of broadcast-derived tracking data and assess its validity for sports science applications, as has been done for semiautomatic multiple camera video tracking systems. This paper has demonstrated that if the locations are accurate, censoring can be overcome to estimate load metrics. Further research needs to be done to determine whether or not the process to produce broadcast tracking data introduces additional error beyond established multicamera video tracking methods, and if so, how much. The second is to scale up the number of teams considered beyond the 19 here and replicate the results of this manuscript. Because Buchheit et al. (2014) indicated that pitch size impacted metric estimates, work needs to be done to ensure that the conclusions drawn here are consistent across all stadiums in the Premier League in addition to being consistent across multiple leagues.

Table 4.3: RMSPE and CV for the base model and subtrack level models on each of the responses

	Base model		**Linear model**		**Linear model w/ int**		**Random forest**	
	RMSPE	CV	RMSPE	CV	RMSPE	CV	RMSPE	CV
total distance (m)	288.2	0.08	202.0	0.06	188.3	0.05	**183.0**	**0.05**
high speed distance (m)	164.5	0.22	113.8	0.15	**106.0**	**0.14**	113.4	0.15
very high speed distance (m)	60.4	0.44	53.4	0.39	53.3	0.39	**42.8**	**0.31**
time in velocity band $[0, 3.5)$ (s)	49.9	0.03	30.4	0.02	**29.1**	**0.02**	29.8	0.02
time in velocity band $[3.5, 5.7)$ (s)	37.8	0.22	26.4	0.15	**24.5**	**0.14**	26.4	0.15
time in velocity band $[5.7, \infty)$ (s)	8.8	0.43	7.9	0.38	7.9	0.38	**6.4**	**0.31**
total acceleration (m/s^2)	2473	0.11	1448	0.07	**1365**	**0.06**	1366	0.06
acceleration density (m/s^2)	0.140	0.12	**0.113**	**0.10**	0.129	0.11	0.119	0.11
time in acceleration band $[0.65, 1.46)$ (s)	34.9	0.06	25.8	0.05	**25.7**	**0.05**	29.5	0.05
time in acceleration band $[1.46, 2.77)$ (s)	45.3	0.13	30.6	0.09	**27.8**	**0.08**	29.5	0.05
time in acceleration band $[2.77, \infty)$ (s)	36.5	0.22	21.5	0.13	**20.9**	**0.13**	22.3	0.13

Table 4.4: RMSPE and CV for the base model and game level models on each of the responses

	Base model		**Linear model**		**Linear model w/ int**		**Random forest**	
	RMSPE	CV	RMSPE	CV	RMSPE	CV	RMSPE	CV
total distance (m)	288.2	0.08	257.0	0.08	267.1	0.08	286.5	0.08
high speed distance (m)	164.5	0.22	150.6	0.20	153.1	0.21	159.1	0.21
very high speed distance (m)	60.4	0.44	60.4	0.44	61.7	0.45	68.1	0.50
time in velocity band $[0, 3.5)$ (s)	49.9	0.03	41.4	0.02	42.0	0.02	60.3	0.03
time in velocity band $[3.5, 5.7)$ (s)	37.8	0.22	34.8	0.20	35.2	0.20	37.5	0.22
time in velocity band $[5.7, \infty)$ (s)	8.8	0.43	9.0	0.43	9.1	0.44	9.91	0.48
total acceleration (m/s^2)	2473	0.11	1748	0.08	1658	0.08	2227	0.10
acceleration density (m/s^2)	0.140	0.12	0.126	0.11	0.140	0.12	0.156	0.14
time in acceleration band $[0.65, 1.46)$ (s)	34.9	0.06	25.7	0.05	27.4	0.05	38.1	0.07
time in acceleration band $[1.46, 2.77)$ (s)	45.3	0.13	32.3	0.09	31.7	0.09	40.4	0.12
time in acceleration band $[2.77, \infty)$ (s)	36.5	0.22	26.7	0.16	25.3	0.15	29.2	0.17

Chapter 5

Identifying Soccer Formations in the Presence of Heavy Censoring

5.1 Introduction

A significant tactical aspect of soccer is the arrangement of players on the pitch, commonly referred to as a formation. A team's formation impacts the way they move the ball towards the opposing team's goal and how they defend attacks. In soccer, each team fields 11 players: 1 keeper, and some mixture of defenders, midfielders, and forwards. Typically, the name of a formation describes the position of the 10 non-keeper players, starting with the defenders. For example, a 4-3-3 formation would consist of four defenders, three midfielders, and three forwards. There are numerous named formations, but these names serve as broad categories rather than rigid descriptions and there is a lot of individual variation in how a given formation is implemented from team to team. As a simple illustration, in a 4-3-3, the full backs can press forward, resulting in a curved back line, or hang back, producing a straight back line instead. Identifying these formations requires knowledgable observers to watch games and categorize teams, a time intensive task, and these categorizations frequently do not capture the nuance just described. Learning formations directly from data can overcome these limitations.

Optical tracking data has been used to identify soccer formations by a variety of researchers. Generally, these approaches involve aggregating the relative locations of the 10 non-keeper players on each team in a careful way to learn how they are arranged. Examples include Wang et al. (2015), who used latent dirichlet allocation (Blei et al., 2003) on match logs augmented with spatial analysis of pass locations to discern tactical patterns used by soccer teams in La Liga; Bialkowski et al. (2014), who applied minimum entropy data partitioning to each frame of tracking data to assign players to unique roles (i.e., positions within a formation) and then used agglomerative hierarchical clustering to classify these collections of roles into formations; and Shaw and Glickman (2019), who also applied hierarchical clustering to learn formations, but rather than assigning players to a role within each frame, averaged player locations over two minutes segments into observations and then performed clustering on the averaged player locations.

These methods have all been used to identify formations from multicamera tracking data, but they are not applicable when confronted with the heavy censoring intrinsic to broadcast tracking data. The primary contribution of this work is to present a method that extends automatic formation identification to tracking data even when a significant portion of that data is missing. To illustrate, in the five games of broadcast tracking data considered in this paper, the largest percentage of frames where all 10 outfield players were observed for a team was just 8.48 percent. Consequently, at best more than 90 percent of observed frames are missing at least one player location. This seemingly presents a substantial challenge, but in the following sections an approach is detailed for carefully aligning each frame of broadcast tracking data with the origin, allowing formations to be estimated and used to classify new observations with performance comparable to that of models fit with the full data.

In addition to addressing censoring, any method for classifying observed collections of player locations into formations must address some further nuisance factors. First, players can swap roles while maintaining the formation, so the method must account for player permutation. Second, formations can contract and expand while roles maintain the same relative position to each other, potentially leading to observations being categorized separately despite being the same formation, so a suitable method must account for the scale of the formation. These issues are also addressed in the following sections.

5.2 Data & Data Processing

The data in this paper comes from five games in a high level European soccer league for which both multicamera tracking data and broadcast camera window information are available. The multicamera tracking data consists of x- and y-coordinate locations for the eleven players on each team at a frequency of 24 frames per second, as well as team IDs, player IDs, and possession information indicating whether a team is on offense or defense. Player locations are transformed so that all teams are attacking from left to right. The camera window is used to censor the multicamera data according to the broadcast pattern, creating a set of "perfect" broadcast data. Simulating broadcast data in this way allows estimates from the full multicamera tracking data to be meaningfully compared with estimates from the broadcast tracking data to assess performance.

5.2.1 Frame Alignment

When using player locations to estimate and classify soccer formations, some notion of centrality is needed in order to align frames so that player locations can be grouped into formation roles. The approach taken in this paper is motivated by considering other proposed methods for doing so and detailing why they are inappropriate for broadcast tracking data.

The simplest technique for frame alignment is to subtract the centroid for each frame from the observed player locations. This works fairly well when all players are observed but with the censoring introduced by the broadcast tracking data centroid estimates become highly unstable. Calculating the centroid based on observed data and using it to center each frame results in diffuse clusters of player locations, as seen in Figure 5.1c. This significantly increases the variance for role estimates and makes identifying formations difficult or impossible.

An alternative approach, the one taken by Shaw and Glickman (2019), is to examine all pairwise differences between observed players and average across some specified window. Relying on pairwise differences allows us to center the data without worrying about the effects of censoring or outliers. To see this, let

$$S = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_P & y_P \end{bmatrix}$$

be the $P \times 2$ matrix of player locations for a given frame (P is equal to 10 for fully observed frames) and $\mathbf{s}_i = (x_i, y_i)$ be the xy coordinates for player i. The centroid for the frame is calculated by taking the mean of each coordinate, that is,

$$\boldsymbol{\delta} = \left(\frac{\sum_{i=1}^{P} x_i}{P}, \frac{\sum_{i=1}^{P} y_i}{P} \right).$$

Obviously, $\mathbf{s}_i - \mathbf{s}_j = (\mathbf{s}_i - \delta) - (\mathbf{s}_j - \delta)$ for all i, j; i.e., the pairwise differences are unaffected by translation.

In order to estimate roles using this alignment technique, it is necessary to be able to easily translate between the location and difference spaces. This can be done by defining a $(P(P-1)/2) \times P$ matrix, D, where each row consists of all zeroes except for a 1 in column i and a -1 in column j so that DS results in a $(P(P-1)/2) \times 2$ matrix of all pairwise differences. D itself is not invertible, since it is not square, but by letting D^- be the Moore-Penrose generalized inverse (Penrose, 1955) of D, then D and D^- can be used to move between the difference and location spaces. Specifically, $D^- DS$ will be a matrix in the location space with the centroid at $(0, 0)$ and with equivalent pairwise differences to S.

Note that if D were applied individually to each frame and then immediately multiplied by D^-, the lack of stability due to outliers and censoring would remain. This is overcome by averaging the differences and then translating back to the location space after the potential effect of any individual outlying observation or missing data has been decreased.

Relying on average differences works well as long as players maintain their roles within a formation, and equally important, are assigned the correct player ID in each segment. Unfortunately this is not always the case. Figure 5.2 shows an example of a possession sequence where players switch roles. In a possession where all players adhered perfectly to their role in the formation, once the frames were aligned, there would be little or no overlap between the tracks represented by the different colors in the left panel. In this possession, the players represented by the light purple and green tracks and the player represented by the dark purple and orange tracks exchanged roles. These kinds of switches are very common and can occur dozens of times over the course of a match. Taking the average differences to estimate role means results in the locations plotted in the top right panel of Figure 5.2. As a result of the switching, the role estimates for the light purple and green players are nearly identical and the estimates for the dark purple and orange players are contracted inwards. Using a permutation based method instead (minimum entropy data partitioning, which is introduced below) results in the more clearly defined roles shown in the lower right panel, better capturing the true formation.

The issue of player switching is compounded when it comes to broadcast tracking data, as accurate player identification from broadcast data is an open research problem. When a player re-enters the camera window, there is a nonzero probability that they will be assigned an ID incorrectly. Averaging player differences across frames can potentially collapse roles together and lead to inaccurate representations of formations in aggregate. Depending on the frequency of the permutation introduced by the broadcast video processing, these averages may be completely meaningless.

In this work, we introduce a new approach to frame alignment that overcomes all of the issues outlined above and works equally well for multicamera or broadcast tracking data. Broadly, it relies on translating successive frames within a possession to minimize the atomic configuration distance (ACD) (Ferré et al., 2015), using minimum entropy data partitioning (MEDP) (Roberts et al., 1999) to estimate a role distribution for these translated frames, and then centering all associated frames by moving the centroid of the role distribution to the origin. Because ACD and MEDP are central to the rest of the work in this chapter, background information on each of them is provided in the following sections.

5.2.2 Atomic Configuration Distance

Atomic configuration distance (ACD) is a permutation invariant distance measure with roots in chemical physics originally developed to compare configurations of atoms (Ferré et al., 2015), though it has also been used to compare configurations of players in basketball and soccer (Miller and Bornn, 2020). It measures the distance between two collections of points in $\mathbb{R}^D$, $X = \{x_1, \ldots, x_{N_x}\}$ and $Y = \{y_1, \ldots, y_{N_y}\}$ by associating a density with each collection of points and calculating the L^2 norm between the densities. More concretely,

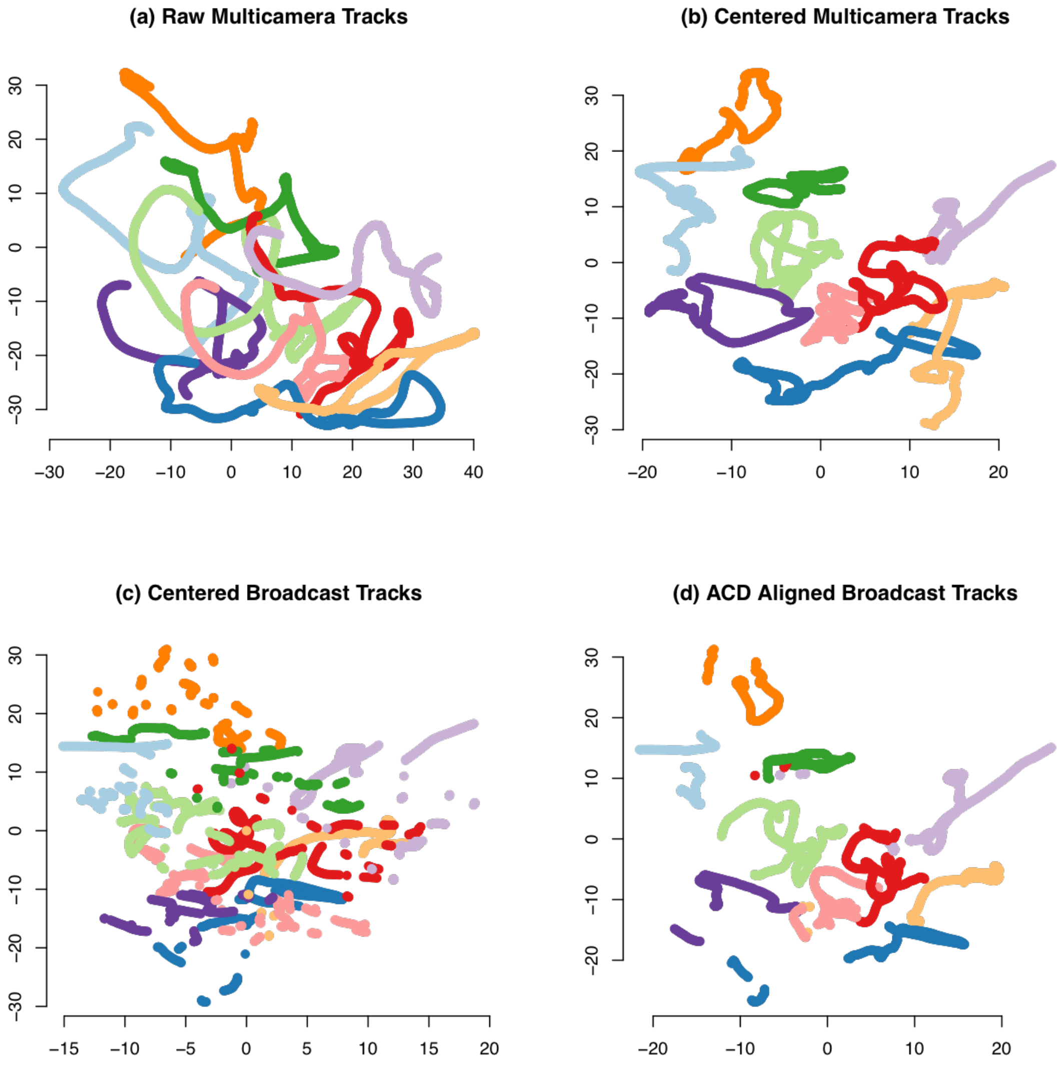
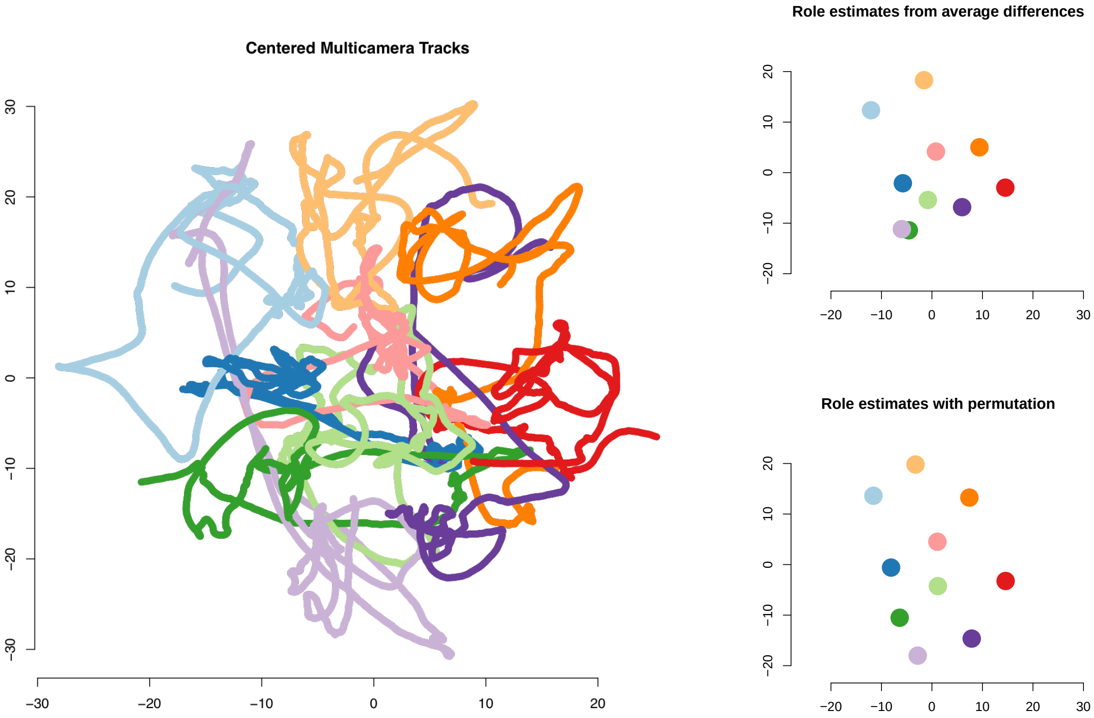


Figure 5.1: (a) Raw multicamera player tracks for one possession in a soccer match. (b) Multicamera tracks centered by placing the centroid of each frame at (0,0). (c) Broadcast tracks centered by placing the centroid of each frame at (0,0). (d) Broadcast tracks centered by minimizing the atomic configuration distance and then placing the centroid of the possession distribution means at (0,0). Simply centering the broadcast tracks (shown in panel (c)) results in poorly defined formation roles.


Centered Multicamera Tracks

Role estimates from average differences

Role estimates with permutation

Figure 5.2: The left panel shows centered multicamera tracks where multiple players switch positions within the formation. A possession where no players switched would have very little overlap between the different colors, but because the orange and dark purple and the light purple and green players overlap, this is evidence they have switched roles. Calculating formation roles by averaging data (top right panel) collapses them together, whereas using MEDP (bottom right panel) maintains distinct roles. Data is oriented so the team is attacking from left to right.

ACD is defined as

$$d^{(acd)}(X,Y) = ||\rho_X(s,\Sigma) - \rho_Y(s,\Sigma)||_2^2, \tag{5.1}$$

where $\rho_X(s,\Sigma) = \frac{1}{N_x}\sum_{i=1}^{N_x}\phi(s;x_i,\Sigma)$ is a kernel density estimator (KDE). Here $\phi(\cdot|m,\Sigma)$ is a multivariate Gaussian density function with mean m and covariance Σ, with Σ serving as a pre-specified multivariate bandwidth parameter.

Calculating (5.1) directly is computationally expensive because it involves an integral, but it can be reduced to an easily computed function of pairwise differences. The basis for this reduction is outlined here. Following Ferré et al. (2015), define the overlap integral

$$S(\rho_X,\rho_Y) = \int_{s\in\mathbb{R}^D} \rho_X(s,\Sigma)\rho_Y(s,\Sigma)ds,$$

which allows (5.1) to be rewritten as

$$\begin{aligned} d^{(acd}(X,Y) &= \int_{s\in\mathbb{R}^D} (\rho_X(s,\Sigma) - \rho_Y(s,\Sigma))^2 ds \qquad (5.2)\\ &= S(\rho_X,\rho_X) + S(\rho_Y,\rho_Y) - 2S(\rho_X,\rho_Y). \qquad (5.3)\end{aligned}$$

As shown in Miller and Bornn (2020), $S(\rho_X,\rho_Y)$ can be simplified to

$$\begin{aligned} S(\rho_X,\rho_Y) &= \int_{s\in\mathbb{R}^D} \rho_X(s,\Sigma)\rho_Y(s,\Sigma)ds \qquad (5.4)\\ &= \left(\frac{1}{4\pi}\right)^{D/2} \frac{|\Sigma|^{-1/2}}{N_x N_y} \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \exp\left(-\frac{1}{4}(x_i - y_j)'\Sigma^{-1}(x_i - y_j)\right). \qquad (5.5)\end{aligned}$$

The full derivation of this equality is provided in the appendix of Miller and Bornn (2020) but for completeness a similar derivation is included at the end of this chapter.

Substituting (5.5) into (5.3) means that the ACD can be calculated without computing any integrals, a key feature in making ACD useful. Reducing the overlap integral to a function of all pairwise differences between X and Y means that evaluation of $S(\cdot,\cdot)$ is both simple and rapid, and consequently calculating $d^{(acd)}$ is feasible even for a large number of observations.

Conceptually, ACD can be thought of as representing a collection of points as an image and assessing similarity on the image space rather than the location space. This results in two key properties that make it ideal for frame alignment. The first is that N_x need not equal N_y in order to calculate the distance, which is desirable since frames in the broadcast tracking data exhibit a variable number of players. The second is that it is permutation invariant, so collections of player locations can be compared without concern that the ordering of the players will impact distance calculations.

5.2.3 Minimum Entropy Data Partitioning

One way to consider formation estimation is as identifying the set of two dimensional densities that is most likely given the observed collection of player locations. From this perspective, the formation, or distribution for the whole team, is the density $P(X = \mathbf{x})$ defined by the linear combination

$$P(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} P_n(\mathbf{x}),$$

where $P_n(\mathbf{x})$ is a bivariate Gaussian representing the distribution for role n within the distribution. Within a formation, roles should exhibit minimal overlap as players spread out to fill the space, which is equivalent to minimizing overlap between each $P_n(\mathbf{x})$ and $P(\mathbf{x})$. Minimum entropy data partitioning (Roberts et al., 1999) is a method for estimating the distributions under this constraint which relies on Kullback-Leibler (KL) divergence, defined

$$KL(P||Q) = \int P(x)\log\left(\frac{P(x)}{Q(x)}\right)dx$$

for densities P and Q. Kullback-Leibler divergence is strictly positive and zero when densities perfectly overlap, so the distributions with minimal overlap can be found by maximizing

$$\sum_{i=1}^{N} KL(P_n||P),$$

or equivalently, minimizing

$$V = -\sum_{i=1}^{N} KL(P_n||P). \tag{5.6}$$

Roberts et al. (1999) shows that in this form, V is equivalent to

$$\int H(x)P(x)dx,$$

where $H(x)$ is the Shannon entropy, so that minimizing V is equivalent to minimizing the entropy over all data partitions, giving the method its name.

There is no closed form solution for (5.6), but Bialkowski et al. (2014) outlines an approximate solution. First, role distributions are initialized by assuming that a player role is constant through the match (or, as in this work, subset of frames) and estimating the associated distribution. In practice, this assumption will likely be violated, but it generally provides a sensible starting point. Then each frame is iterated through, and every player is assigned a unique role within each frame. This is done by constructing a cost matrix C where $C_{i,j} = -\log(P_j(\mathbf{x}_i))$, that is, role density P_j evaluated at player location $\mathbf{x}_i$. If any of the values of the cost matrix are negative, then $C := C - min(C)$ to ensure that all costs are non-negative. After the costs are computed, C is used in conjunction with the Hungarian algorithm (Kuhn, 1955; Munkres, 1957) to permute the player locations to minimize the total cost, and role densities are re-estimated with the permuted player locations. Estimation and reassignment are repeated until no further permutation takes place.

Because broadcast tracking data does not contain a constant number of observed players within each frame, entries in the cost matrix for missing players are set to the maximum observed cost plus one. This adaptation ensures that all observed player locations will be assigned to the "cheapest" role density while still permitting use of the Hungarian algorithm.

5.2.4 ACD Frame Alignment Algorithm

The alignment algorithm outlined in this section is the primary contribution of this paper. This algorithm overcomes both player permutation effects and the lack of a stable centroid within each frame caused by broadcast-induced censoring. At a high level, this is accomplished through aligning frames within a possession sequence by minimizing the ACD between them. After frames are aligned within a possession, MEDP is used to estimate a role distribution for the sequence and the centroid of the role distribution is used to center the aligned possession at the origin. Once all data has been centered at the origin, formation estimation can proceed.

MEDP is used to center frames at the origin (rather than, for instance, using ACD to align all frames in a match half) for two reasons. First, as long as all ten players are observed at least once in the possession sequence, a role distribution can be estimated with ten role means, even if no frame with all ten players is observed. As a result, centroid estimates are generally reliable even in the presence of significant missing data. Second, ACD tends to exhibit many local optima, and using it to try and align player locations that are not already close to each other is unlikely to produce the correct alignment. Relying on the centroid of the role distribution for centering instead bypasses this problem.

Before detailing the algorithm for aligning data for a complete game, it is instructive to discuss an algorithm for aligning just two arbitrary frames. In order to align two frames using ACD, define the function

$$f(\boldsymbol{\delta}|X,Y) = d^{(acd)}(X, \boldsymbol{\delta} + Y), \tag{5.7}$$

where $\boldsymbol{\delta}$ is a matrix with all rows equal to (δ_x, δ_y). This function is used to find $\hat{\boldsymbol{\delta}} = \arg\min_{\delta_x,\delta_y \in \mathbb{R}} f(\boldsymbol{\delta}|X,Y)$ and then set $Y^\star := Y + \boldsymbol{\delta}$ so that X and $Y^\star$ are aligned. Similarly, define

$$g(\mathbf{k}|X,Y) = d^{(acd)}(X, \mathbf{k}(Y - \bar{Y}) + \bar{Y}), \tag{5.8}$$

where each row of $\bar{Y}$ is equal to the column means of Y, and use it to find

$$\hat{\mathbf{k}} = \arg\min_{k \in \mathbb{R}^+} g(\mathbf{k}|X,Y).$$

In this case, the scaling factor $\mathbf{k}$ can either be a scalar or a 2-vector in order to scale the x and y dimensions separately, if desired. Equations (5.7) and (5.8) can be used together in an EM-like way to align X and Y by iterating translation and scaling until convergence.

Borrowing ideas from the algorithm to align two frames of broadcast tracking data, an algorithm for aligning an entire match is presented in the rest of this section. In order to prepare the broadcast tracking data for analysis, frames are divided into continuous segments, $\Xi_i = [S_{i1}, \ldots, S_{iT}]$, where at least one player was observed in each frame, S_{ij}, in the sequence. In other words, if at any point the player tracks of all ten outfield players are interrupted, whether that be because the broadcast cut away from on-pitch play to show a highlight or because all the players were out of the camera window, that results in a new Ξ_i. For successive frames S_{it} and S_{it+1} in Ξ_i estimate $\hat{\boldsymbol{\delta}}_{it}$ using (5.7), where $\hat{\boldsymbol{\delta}}_{it}$ denotes the constants used to translate frame S_{it} to align with frame S_{it+1}. Previously it was noted that (5.7) has many local optima. The negative effects of this are avoided by only estimating $\boldsymbol{\delta}_{it}$ for successive frames. Because these frames are separated by only a fraction of a second, the local optimum is generally the desired one.

After estimating the sequence $(\boldsymbol{\delta}_{i1}, \ldots, \boldsymbol{\delta}_{iT-1})$ frame S_{it} can be translated so that it aligns with S_{iT} by $S_{it}^{\Delta} = S_{it} + \sum_{j=t}^{T-1} \boldsymbol{\delta}_{ij}$. Conceptually this creates a telescoping effect where the first frame is translated by $\boldsymbol{\delta}_{i1}$ to align with the second frame, then the first and second frames are both translated by $\boldsymbol{\delta}_{i2}$ to align with the third frame, and so on, until frames 1 through $T-1$ are all translated by $\boldsymbol{\delta}_{iT-1}$ to align with the final frame of the sequence. The sum $\sum_{j=t}^{T-1} \delta_{ij}$ is not necessarily equivalent to $\arg\min_{\delta_x, \delta_y \in \mathbb{R}} f(\boldsymbol{\delta}|S_{iT}, S_{it})$ but it is approximately equal and has the advantage of being fast to compute and avoiding undesirable behavior driven by the local optima in (5.7) if one were to attempt to align S_{i1} with S_{iT} directly.

After aligning all frames with the final frame, centering the translated sequence Ξ_i^{Δ} at the origin simplifies estimation of the scaling parameters and provides a common point for aligning all match data. Because individual frames are incompletely observed, simply subtracting the means would be subject to the same centroid instability noted earlier. Instead MEDP is applied to each Ξ_i^{Δ} to estimate a canonical mean M_i for the segment and subtract its centroid from each S_{it}^{Δ}. MEDP is used to estimate the means rather than, for example, finding the most central S_{it}^{Δ} in Ξ_i^{Δ} because it may be the case that all ten players are not observed together in any frame, even though collectively they are all observed across the sequence.

Once all the frames in Ξ_i^{Δ} have been translated to the origin, a similar process can be followed to scale the frames to align with each other. As before, successive frames are aligned but this time (5.8) is applied to estimate scaling parameters $(\mathbf{k}_{i1}, \ldots, \mathbf{k}_{iT-1})$, where $\mathbf{k}_{it}$ represents the scaling factor used to align frame S_{it} with frame S_{it+1}. Because the frames are already centered at $(0,0)$, $\bar{Y}$ in (5.8) is assumed to be 0 for all frames. Once $(\mathbf{k}_{i1}, \ldots, \mathbf{k}_{iT})$ have been estimated, S_{it}^{Δ} can be scaled to align with S_{iT} by $S_{it}^{\star} = S_{it}^{\Delta} \prod_{j=t}^{T-1} \mathbf{k}_{ij}$. Care must

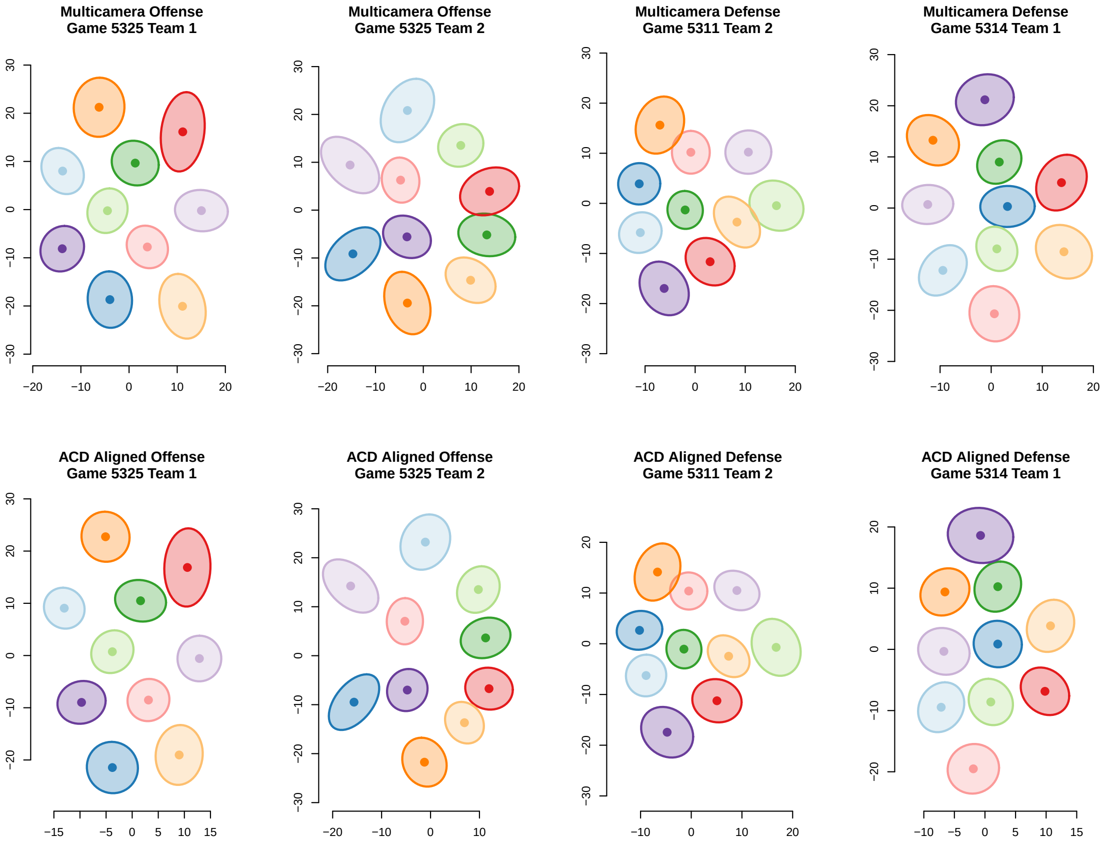


Figure 5.3: Formation estimates for four different teams over an entire period in each game. The top row shows estimates made from the full multicamera tracking data. The bottom row shows formation estimates generated from the ACD aligned tracking data.

be taken to ensure that $\mathbf{k}_{ij}$ does not shrink too much, as one way to minimize the distance between two sets of points is to multiply them by a very small amount. One way to account for this is to constrain $\mathbf{k}_{ij}$ to fall within $(0.9, 1.1)$, which is reasonable because relative player locations should not drastically contract or expand in just 1/24th of a second. If desired, translation and scaling of the sequence can be alternated as in the case for alignment of a single frame, though in the work shown in the rest of the chapter translation and scaling were performed only once. Pseudocode summarizing the algorithm described in this section is provided as Algorithm 1.

An example of a single possession aligned using this ACD method is shown in Figure 5.1d, which is encouragingly akin to the centered multicamera tracks shown in panel (b) above it. At the aggregate level, a simple visual comparison of formations estimated from the multicamera tracking data to ones estimated from the ACD aligned tracking data in Figure 5.3 shows that they are remarkably similar.

```
input  : $N$ arrays $\Xi_i = [S_{i1}, \ldots, S_{iT_i}]$ of dimension $10 \times 2 \times T_i$ containing continuous
         segments of broadcast tracking data, and a pre-specified ACD bandwidth
         parameter $\Sigma$
output: $N$ arrays $\Xi_i^\star$ of aligned broadcast tracking data centered at $(0, 0)$
begin
    for $i = 1, \ldots, N$ do
        /* First, each frame is aligned with the final frame in the
           sequence by minimizing ACD, which overcomes the issue of
           censoring.                                                  */
        $S_{iT_i}^\Delta \leftarrow S_{iT_i}$;
        for $t = T_{i-1} - 1, \ldots, 1$ do
            $\boldsymbol{\delta}_{it} \leftarrow \arg\min_{\boldsymbol{\delta} \in \mathbb{R}} f(\boldsymbol{\delta} | S_{it+1}, S_{it})$ /* ($f$ is defined in (5.7))   */;
            $S_{it}^\Delta \leftarrow S_{it} + \sum_{j=t}^{T_i - 1} \boldsymbol{\delta}_{ij}$;
        end
        /* Next, a role distribution is estimated from the aligned frames
           and the centroid of the role distribution means is used to
           translate all frames in the sequence to the origin. Array
           index notation is used to denote the taking of the mean of the
           x- and y-coordinates. Using MEDP overcomes role switching
           effects and produces a stable centroid even if no single frame
           has all 10 players present.                                 */
        $\Xi_i^\Delta \leftarrow \left[S_{i1}^\Delta, \ldots, S_{iT}^\Delta\right]$;
        $M_i \leftarrow$ Means of the role distribution estimated from $\Xi_i^\Delta$ using MEDP. $M_i$ is a
         $10 \times 2$ matrix.;
        $\bar{m}_x \leftarrow \sum_{k=1}^{10} M_i[k, 1]/10$ ;
        $\bar{m}_y \leftarrow \sum_{k=1}^{10} M_i[k, 2]/10$;
        $C_i \leftarrow \begin{bmatrix} \bar{m}_x & \bar{m}_y \\ \vdots & \vdots \\ \bar{m}_x & \bar{m}_y \end{bmatrix}$;
        for $t$=1, ..., $T$ do
            $S_{it}^\Delta \leftarrow S_{it}^\Delta - C_i$;
        end
        /* Finally, each frame is scaled to align with the final frame in
           the sequence. The scaling factor is estimated by minimizing
           the ACD between successive frames.                          */
        $S_{iT_i}^\star \leftarrow S_{iT_i}^\Delta$;
        for $t = T_i - 1, \ldots, 1$ do
            $k_{it} \leftarrow \arg\min_{k \in \mathbb{R}^+} g(k | S_{it+1}^\Delta, S_{it}^\Delta)$ ($g$ is defined in (5.8), though $\bar{Y}$ is
             assumed to be zero since each frame is approximately centered at the
             origin);
            $S_{it}^\star \leftarrow S_{it}^\Delta \prod_{t=1}^{T_i - 1} k_{it}$;
        end
        $\Xi_i^\star \leftarrow \left[S_{i1}^\star, \ldots, S_{iT_i}^\star\right]$;
    end
end
```

Algorithm 1: Pseudocode summarizing the algorithm in Section 5.2.4.

5.2.5 Aggregating Frames

Once frames have been aligned using Algorithm 1, it is helpful to condense the number of observations by averaging frames over some specified time window. Depending on the provider, tracking data is typically available anywhere from 10 to 30 frames per second, and this high temporal density results in strong correlation. Aggregating frames can reduce computational burden by decreasing the number of observations to process while simultaneously improving the ability to detect specific formations, as the averages are less variable than individual frames.

One way to approximate how large the aggregation window should be is to calculate the effective sample size (ESS), $N_{eff} = \frac{N}{1+2\sum_{i=1}^{\infty} \psi_i(x)}$ (Gelman et al., 2013, Section 11.5), where ψ_i is the autocorrelation of lag i for x, and use that to determine how many frames should be averaged together by setting the window size equal to N_{eff}/N (or equivalently $1 + 2\sum_{i=1}^{\infty} \psi_i(x)$). Converting the frames onto the difference space and then calculating the ESS for each difference sequence led to estimated window sizes ranging from 13 to 205 seconds, with a median of 36 seconds. In practice, clustering performance improved when the averaging window was above 120 seconds, so although that is on the upper end of the window sizes estimated via ESS, 120 second averaging windows are used throughout the remainder of this paper.

A variety of additional steps were taken in order to render the data more suitable for formation estimation. Possessions less than 5 seconds in length were removed from the data, because it is unlikely a team is in formation as they transition from defense to offense and back again. Further, averages do not span across periods or player substitutions, as generally if a change in tactics is to occur, it is likely to be accompanied by one of these two events. Finally, offensive and defensive possessions were averaged separately, as formations change when the team is in or out of possession and combining them could potentially wipe out meaningful distinctions.

There are several different ways to perform aggregation, two of which are compared in this section. The first involves aggregating frames by taking average differences. That is, each frame is multiplied by D, the average of the differences is calculated over 120 second windows, and the averages are converted back to the location space using D^{-}. This is done for both the full multicamera data and the broadcast tracking data, though due to the censoring, this approach has to be adapted for the broadcast data. Specifically, the average is taken just from the observed data so each average difference has a different denominator, which is denoted mathematically as $\bar{X} = (\sum_{i=1}^{N_1} \frac{x_1}{N_1}, \ldots, \sum_{i=1}^{N_L} \frac{x_L}{N_L})$, where N_i is the total number of non-missing differences i. If a difference is not observed at all (i.e., $N_i = 0$) within the averaging window, it is removed completely from the analysis, and then all remaining observations are translated back to the location space. The second method for aggregating is to estimate role means over the 120 second windows using MEDP and

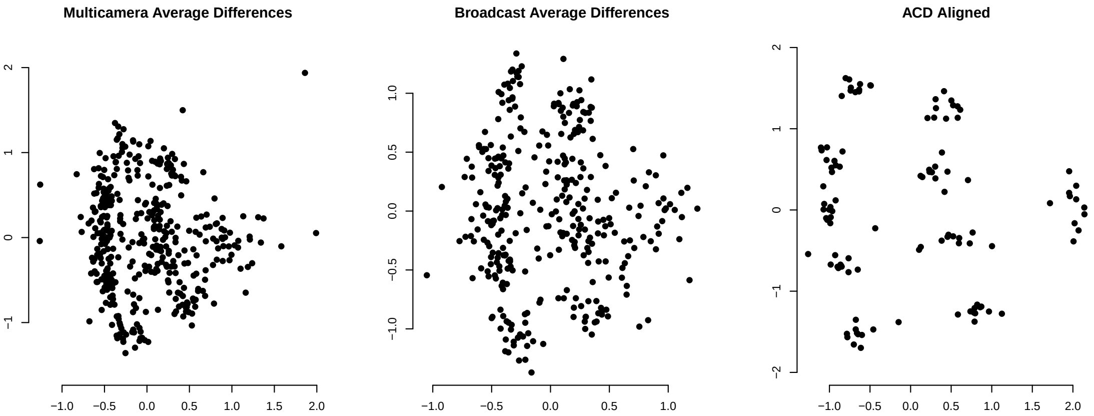


Figure 5.4: Examples of processed data with the three different approaches. Note that the reduced data for the ACD aligned approach is compensated for somewhat by the clarity of the observations.

treat the role means as the observations, which is the approach taken for the ACD aligned broadcast data.

For the complete data, the 120 second window results in 480 observations from the five games, whereas the censored data results in 422 observations. Because possessions in the ACD alignment process are only retained if all ten players are observed in at least one frame, the ACD aligned data is further reduced with just 195 observations.

5.2.6 Scaling

In Section 5.1, three nuisance factors were mentioned that need to be dealt with in order to accurately estimate soccer formations. The first two, censoring and player permutation, were addressed by the frame alignment algorithm in Section 5.2.4 and MEDP, respectively. The last factor is scaling. This is important because two observations should be classified as the same formation if the relative positions of the players within the formation are the same, even if one observation is more contracted or expanded than the other. It is possible to estimate a scaling parameter to address this, but in this work, contraction and expansion is handled simply by scaling the data. In other words, all the x- and y-coordinates for a match half are divided by the empirical standard deviation of the x- and y-coordinates, respectively. Examples of the scaled data for each of the three approaches is shown for one team on defense in Figure 5.4. Clearly, the 5-4-1 formation is discernible for all approaches.

5.3 Methods

5.3.1 Clustering Observations

In order to identify general categories of formations, clustering is performed on the three different sources of observations (full multicamera, average difference broadcast data, and ACD aligned broadcast data) using a finite mixture model (McLachlan and Basford, 1988),

$$\prod_{i=1}^{N}\sum_{k=1}^{K}\omega_k f(x_i|\Theta_k).$$

For this application, x_i is a $2P$ vector of player role means, averaged over a two minute window ($P = 10$, the number of non-keeper players on each time), ω_k represents the weight for each of K mixture components, and $f(x_i|\Theta_k)$ is some distribution governed by parameters Θ_k. As with all finite mixture models, there are two additional considerations that must be addressed: how to fit the model and how to choose the number of clusters.

The models in this paper are fit using expectation maximization (Dempster et al., 1977) within a Bayesian framework. The mixture weights, ω_k, are assumed to follow a symmetric Dirichlet distribution with concentration parameter $\alpha_0 = 50$ in order to encourage observations to be evenly distributed among the K available mixture components. The mixture distribution is assumed to be multivariate Gaussian with diagonal covariance. That is, $f(\cdot|\Theta_k) = \mathcal{N}(X_i|\mu_k, \Sigma_k)$, where the diagonal of Σ_k is $(\sigma^2_{k1}, \ldots, \sigma^2_{k2P})$ and all off-diagonal entries are zero. The parameters of the mixture distribution, $\boldsymbol{\mu}_k$ and σ^2_{ki} require priors, which are assumed to be $\boldsymbol{\mu}_k \sim \mathcal{N}(\mathbf{0}, s^2 I)$ and $\sigma^2_k \sim \text{Inv}-\chi^2(\nu_0)$, where $s^2 = 4$ and $\nu_0 = 52$. Setting $\nu_0 = 52$ means the prior expected value for σ^2_k is 0.02, which would be far too restrictive on the data's raw scale, but is reasonable after scaling.

There are numerous texts that detail how to fit a standard Gaussian mixture model using expectation maximization (see, for example, Chapter 22 of Gelman et al. (2013), Chapter 9.2.2 of Bishop (2006), or Chapter 8.5 of Hastie, Trevor, Tibshirani, Robert, Friedman (2009)) so those details are omitted here. However, as implemented for this research, there are a couple of significant changes to the standard algorithm. First, the Hungarian algorithm is used to account for the effects of player permutation, similar to how it is used in MEDP. The cost that is minimized is the Euclidean distance between the player locations in each observation and the mean in each mixture component. This reordering is performed within each iteration of the expectation maximization implementation, so observations are permuted to the closest available $\boldsymbol{\mu}_k$ each time the means are updated.

The second change is an idea borrowed from Ghahramani and Jordan (1994), who showed that incomplete observations could be included in a mixture model and just the observed elements used to estimate assignment probabilities. Expectation maximization works for mixture models by associating a latent indicator, $I(z_i = k)$, with each observation that determines which mixture component it belongs to. The expected value of this indicator is calculated

$$\hat{z}_{ik} = P(z_i = k) = \frac{\prod_{p=1}^{2P} \mathcal{N}(x_{ip}|\mu_{kp}, \sigma_{kp}^2)}{\sum_{\ell=1}^{K} \prod_{p=1}^{2P} \mathcal{N}(x_{ip}|\mu_{\ell p}, \sigma_{\ell p}^2)}. \tag{5.9}$$

In order to adapt this assignment probability for partially observed observations, Ghahramani and Jordan convert (5.9) to

$$\hat{z}_{ik} = P(z_i = k) = \frac{\prod_{p=1}^{2P} \left(\mathcal{N}(x_{ip}|\mu_{kp}, \sigma_{kp}^2)\right)^{I(o_{ip}=1)}}{\sum_{\ell=1}^{K} \prod_{p=1}^{2P} \left(\mathcal{N}(x_{ip}|\mu_{\ell p}, \sigma_{\ell p}^2)\right)^{I(o_{ip}=1)}}, \tag{5.10}$$

where o_{ip} is a dummy variable that is 1 if the pth element of observation i is observed and zero otherwise.

Because the observations being clustered in this paper are aggregated across partially observed frames, weights are calculated as $\mathbf{w}_i = (w_{i1}, \ldots, w_{i2P})$, where $w_{ip} = \sum_{n=1}^{N_i} o_{np}$ and assignment probabilities are calculated using (5.10) with w_{ip} in place of the indicator $I(o_{ip} = 1)$. Essentially, this means that if x_{i1} is observed just once out of the $24 * 120 = 2880$ total frames it is being averaged over, then it would only contribute 1/2880 of the amount that a completely observed element of $\mathbf{x}_i$ would to determining which formation it belongs to. These weights also affect parameter estimation in the mixture distributions, and derivations for these weighted estimates are included in the appendix.

The number of clusters to use is determined by examining the Akaike (AIC) and Bayesian information criteria (BIC) (Akaike, 1974; Schwarz, 1978), in addition to considering the silhouettes for each model (Rousseeuw, 1987), a measure which compares the average distance within and between clusters. A silhouette value (or width), $s(i)$, for each observation falls within $(-1, 1)$, with values close to 1 indicating that an observation is closer to the members of its assigned cluster than members of any other cluster. This corresponds to well-defined clusters with little overlap, and so choosing the number of clusters with the largest average silhouette width leads to the most well-defined formations.

5.3.2 Comparing Results

Once the models are fit, it is necessary to assess how well the formations learned from the average difference and ACD aligned broadcast data correspond to the formations learned from the multicamera tracking data. Visually, it is possible to compare the learned formations and get a general sense of how well they correspond. For a more statistically principled method of comparison, the learned formation means are used to classify new observations and see how accurately they predict. As a ground truth, the formation labels provided in the match summaries for the 5 games in the dataset on https://www.premierleague.com are used. Generally, it is not expected that learned formations directly correspond to labeled formations. However, because the dataset in this work consists of only five games and seven unique teams, each of which has a different labeled formation, this assumption is reasonable for this specific case. Put another way, this method of assessment expects that observations from the same team generally get classified into the same one or two learned formations.

The observations used for prediction are the 480 observations (240 for offense and 240 for defense) calculated using average differences on the full multicamera data. Values are classified according to the calculated assignment probability in (5.9), assigning an observation to the formation with the highest probability. This is equivalent to using a Bayesian classifier with a uniform prior over the mixture assignments. Because there are minor potential differences due to scaling from the different data processing methods, observation are aligned to each formation mean using the ACD alignment algorithm outlined in Section 5.2.4 prior to making a prediction. Classification accuracy is summarized by providing plots showing the proportion of observations within each labeled formation that were assigned to each of the learned formations.

5.4 Results

In order to select the number of formation categories, the Gaussian mixture model is fit for a range of two to fifteen components, and because mixture models tend to exhibit many local optima, it is fit ten times for each component amount. Figure 5.5 shows the AIC and BIC values for offense and defense for each of the three approaches. There are several patterns worth noting. First, AIC decreases monotonically in all plots making it useless as a model selection criteria in this instance. This is likely because the diagonal covariance specified is very flexible and the AIC penalty is insufficiently large to overcome the corresponding increase in the log likelihood. Second, there is a significant loss in fidelity when calculating average differences with the broadcast data, leading to BIC suggesting that just two formations is the correct model. The ACD aligned data does not suffer from this same issue. Based on the BIC values for the ACD aligned data and multicamera data, $K = 6$ is selected for the number of offensive formations and $K = 8$ for the number of defensive formations. Note that this may not be the optimal selection for all three data sources, but the same number of mixture components are used across all three in order to make comparison of the data processing methods easier.

Examination of the average silhouette widths in Figure 5.6 reveals two interesting patterns. First is that the silhouette widths for the multicamera and ACD aligned data suggest that six or seven is the correct number of defensive formations. The second is that the silhouette widths for the offense consistently decrease as the number of formations increases. This indicates that the clusters are not particularly well-defined for offense, and indeed, estimation of offensive formations is consistently more challenging than defensive formations.

Due to the large number of formation estimates, plots for most of them are relegated to Section 5.7, but for purposes of illustration the ACD aligned defense estimates are provided in Figure 5.7. This plot shows 8 obvious formations, with, for example, clusters 1, 6, 7,

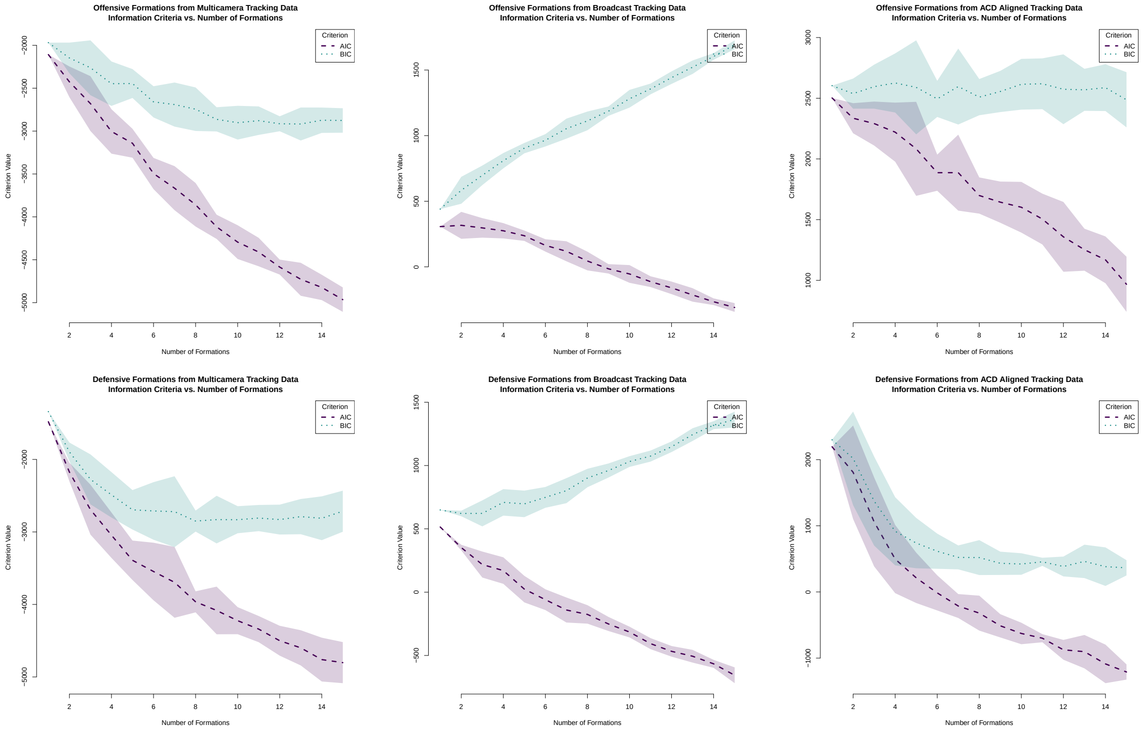


Figure 5.5: AIC and BIC values for each of the different data sources. Information criteria values for offensive formations are in the top row, and information criteria for defensive formations are in the bottom row.

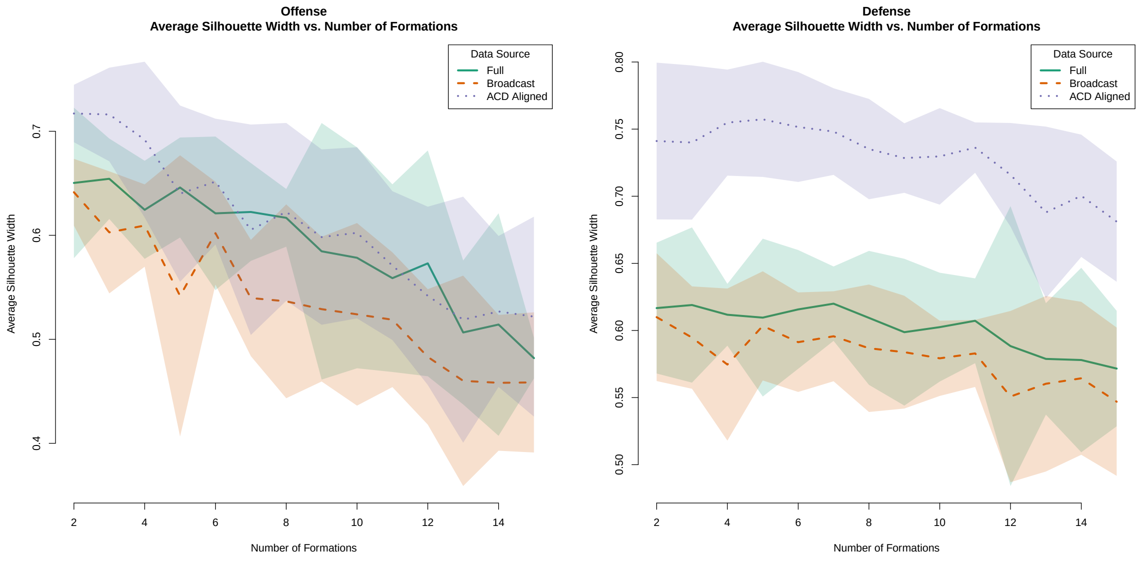


Figure 5.6: Average silhouette widths for offense (left) and defense(right) for each of the three methods for data processing. Larger average silhouette widths indicate better defined clusters.

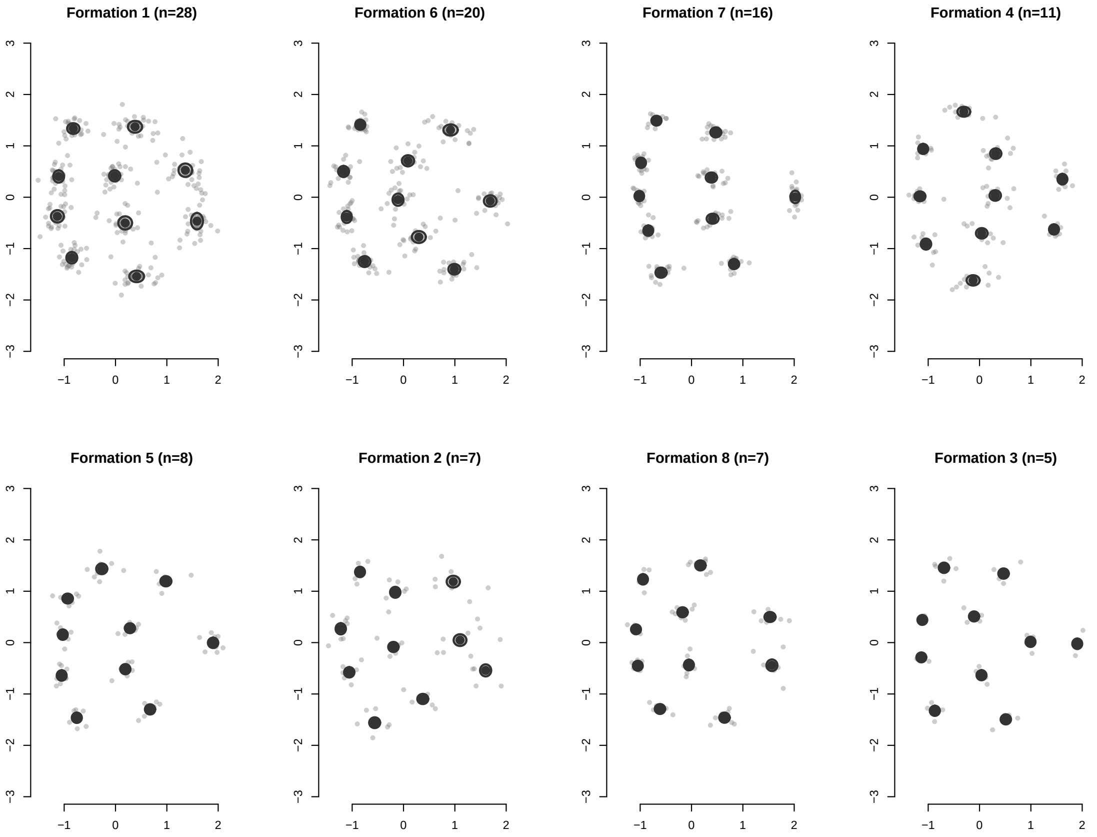


Figure 5.7: Defensive formation estimates for observations generated from ACD aligned tracking data using MEDP, with 1-σ ellipses and classified observations, sorted by size of each cluster.

and 4 clearly corresponding to the formations 4-4-2, 4-3-3, 5-4-1, and 3-5-2, respectively. Considering the enormous amount of data missing from the broadcast tracks, the clarity of these formations is quite remarkable.

Turning to comparison via prediction, consider the proportion of accurate predictions shown in Figure 5.8. On offense, the 3-5-1-1 is consistently classified correctly for all three methods. The ACD aligned data does a better job of separating out the 3-4-3, though it does combine it with some observations from the 5-4-1, and it does a better job of isolating the 4-2-3-1 than either of the other two approaches. The remaining labeled formations get grouped together, likely due to the fact that they all share a 4 defender back line. However, there is more discrimination between the 4-3-2-1, 4-3-3, and 4-4-1-1 with the two average difference methods, suggesting that the ACD aligned formation means are not capturing some features. The full multicamera data does the best job at detecting the 4-3-2-1 and 4-3-3, while the average differenced broadcast tracking data isolates the 4-4-1 most consistently.

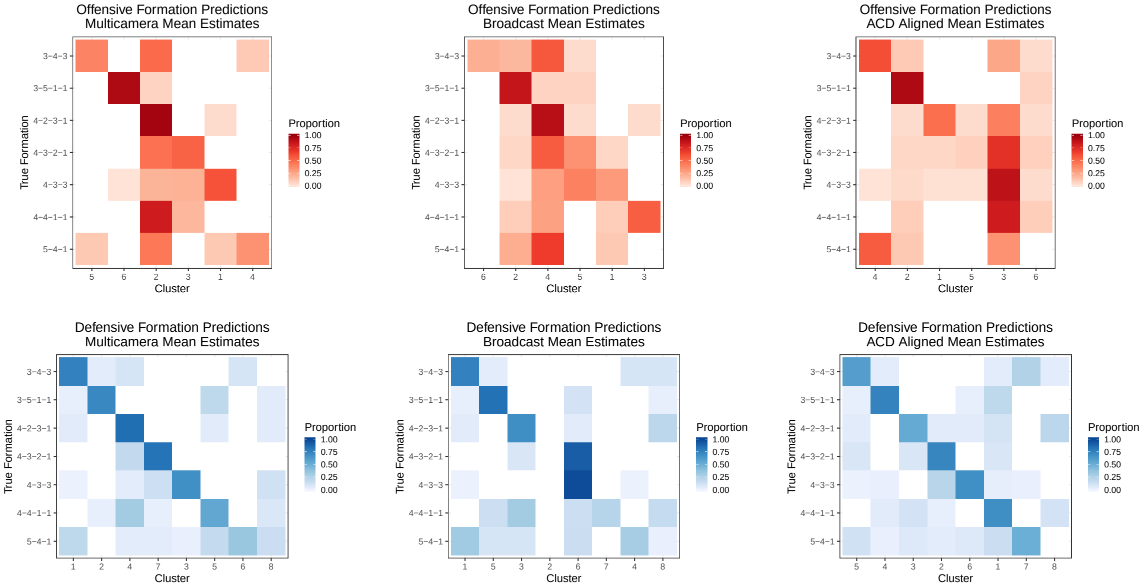


Figure 5.8: Matrices showing the proportion of observations based on the official formations listed on www.premierleague.com that correspond with each of the formation estimates.

Cluster consistency is much better when considering defense. Prediction performance for the ACD aligned estimates in the lower right is comparable to prediction performance for the full multicamera data, with the majority of observations in each named formation being assigned to a separate cluster. This is much better than the averaged differences with the broadcast tracking data, which combines the 4-3-2-1 and 4-3-3 formations into a single cluster and does not have any coherent clustering for the 4-4-1-1 or 5-4-1.

5.5 Conclusion

In this work, an algorithm for aligning broadcast tracking data using atomic configuration distance has been introduced. It has been demonstrated that data aligned by this algorithm produces formation estimates comparable to those estimated from uncensored multicamera tracking data, despite the significant loss of data inherent to broadcast tracking data. Aligning broadcast tracking data with ACD overcomes issues related to player permutation and the inconsistent number of on-camera players, and is even robust to inaccurate player identification. Although the ACD aligned estimates failed to capture two of the six offensive formations, it collapsed them in an understandable way by combining three formations with 4 defenders in the back line. The ACD aligned formation estimates successfully distinguished between the 7 labeled formations, consistently placing them in separate clusters. These findings suggest that broadcast tracking data is viable for estimation of formations and tactical analysis in soccer.

5.5.1 Future Work

The most obvious avenue for future work is the retention of partially observed segments. Currently, the alignment algorithm throws out any segment that does not contain all 10 players in at least one frame in the segment. This results in a significant reduction of data, leaving us with less than 50% of the observations generated from the multicamera tracking data. Research is actively being done on how to use ACD to align these partially observed segments with the rest of the data so that data loss is less severe.

5.6 Derivations

5.6.1 ACD Derivation

$$\begin{aligned}
S(\rho_x, \rho_y) &= \int_{s \in \mathbb{R}^D} \rho_X(s, \Sigma)\rho_Y(s, \Sigma) ds && (5.11)\\
&= \int_{s \in \mathbb{R}^D} \left(\frac{1}{N_x} \sum_{i=1}^{N_x} \phi(s; x_i, \Sigma) \right) \left(\frac{1}{N_y} \sum_{j=1}^{N_y} \phi(s; y_j, \Sigma) \right) ds && (5.12)\\
&= \frac{1}{N_x N_y} \int_{s \in \mathbb{R}^D} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \phi(s; x_i, \Sigma)\phi(s; y_j, \Sigma) ds && (5.13)\\
&= \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \int_{s \in \mathbb{R}^D} \phi(s; x_i, \Sigma)\phi(s; y_j, \Sigma) ds && (5.14)
\end{aligned}$$

where we use Fubini's theorem to swap the sums and integral based on the fact that ϕ is always nonnegative. We can expand out the integral to get

$$\begin{aligned}
&\int_{s \in \mathbb{R}^D} \phi(s; x_i, \Sigma)\phi(s; y_j, \Sigma) ds\\
&= \int_{s \in \mathbb{R}^D} (2\pi)^{-D} |\Sigma|^{-1} \exp\Big(-\frac{1}{2}(s - x_i)'\Sigma^{-1}(s - x_i)\Big) \exp\Big(-\frac{1}{2}(s - y_j)'\Sigma^{-1}(s - y_j)\Big) ds\\
&= (2\pi)^{-D} |\Sigma|^{-1} \int_{s \in \mathbb{R}^D} \exp\Big(-\frac{1}{2}[(s - x_i)'\Sigma^{-1}(s - x_i) + (s - y_j)'\Sigma^{-1}(s - y_j)]\Big) ds.
\end{aligned}$$

Considering just the contents of the exponent we get

$$\begin{aligned}
&-\frac{1}{2}[(s-x_i)'\Sigma^{-1}(s-x_i)+(s-y_j)'\Sigma^{-1}(s-y_j)]\\
&=-\frac{1}{2}\left[2s'\Sigma^{-1}s-2s'\Sigma^{-1}(x_i+y_j)+x_i'\Sigma^{-1}x_i+y_j'\Sigma^{-1}y_j\right]\\
&=-\left[s'\Sigma^{-1}s-s'\Sigma^{-1}(x_i+y_j)+\frac{x_i'\Sigma^{-1}x_i}{2}+\frac{y_j'\Sigma^{-1}y_j}{2}\right]\\
&=-\Bigg[s'\Sigma^{-1}s-s'\Sigma^{-1}(x_i+y_j)+\frac{(x_i+y_j)'\Sigma^{-1}(x_i+y_j)}{4}-\\
&\frac{(x_i+y_j)'\Sigma^{-1}(x_i+y_j)}{4}+\frac{x_i'\Sigma^{-1}x_i}{2}+\frac{y_j'\Sigma^{-1}y_j}{2}\Bigg]\\
&=-\left[\left(s-\frac{x_i+y_j}{2}\right)'\Sigma^{-1}\left(s-\frac{x_i+y_j}{2}\right)+\frac{1}{4}(x_i-y_j)'\Sigma^{-1}(x_i-y_j)\right]
\end{aligned}$$

since

$$(x_i-y_j)'\Sigma^{-1}(x_i-y_j)+(x_i+y_j)'\Sigma^{-1}(x_i+y_j)=2x_i'\Sigma^{-1}x_i+2y_j'\Sigma^{-1}y_j.$$

If we observe that

$$\int_{s\in\mathbb{R}^D}\exp\left(-\left(s-\frac{x_i+y_j}{2}\right)'\Sigma^{-1}\left(s-\frac{x_i+y_j}{2}\right)\right)ds=(2\pi)^{\frac{D}{2}}\left(\frac{1}{2}\right)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}$$

then we can write

$$\begin{aligned}
&(2\pi)^{-D}|\Sigma|^{-1}\int_{s\in\mathbb{R}^D}\exp\Big(-\frac{1}{2}[(s-x_i)'\Sigma^{-1}(s-x_i)+(s-y_j)'\Sigma^{-1}(s-y_j)]\Big)ds\\
&=(2\pi)^{-D}|\Sigma|^{-1}(2\pi)^{\frac{D}{2}}\left(\frac{1}{2}\right)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}\exp\left(-\frac{1}{4}(x_i-y_j)'\Sigma^{-1}(x_i-y_j)\right)\\
&=(4\pi)^{-\frac{D}{2}}|\Sigma|^{-\frac{1}{2}}\exp\left(-\frac{1}{4}(x_i-y_j)'\Sigma^{-1}(x_i-y_j)\right)
\end{aligned}$$

Plugging this into (5.14), we achieve

$$\frac{(4\pi)^{-\frac{D}{2}}|\Sigma|^{-\frac{1}{2}}}{N_xN_y}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\exp\left(-\frac{1}{4}(x_i-y_j)'\Sigma^{-1}(x_i-y_j)\right).$$

□

5.6.2 Weighted EM Derivation

First we show the derivation for the posterior maximum for μ_{kp}. Note that because we have assumed diagonal covariance we can use univariate estimates for each element of $\boldsymbol{\mu}_k$.

$$
\begin{aligned}
f(\mu_{kp}|\mathbf{x},\Theta) &\propto \prod_{i=1}^{N} f(\mathbf{x}_i|\boldsymbol{\mu}_k,\Sigma_k)^{z_{ik}} f(\boldsymbol{\mu}_k) \\
&\propto \prod_{i=1}^{N} \left[\left(\exp\left(-\frac{1}{2\sigma_{kp}^2}(x_{ip}-\mu_{kp})^2\right)\right)^{w_{ip}}\right]^{z_{ik}} \exp\left(-\frac{1}{2s^2}(\mu_{kp}-m_0)^2\right)
\end{aligned}
$$

Taking the log of both sides, we get

$$
\log f(\mu_{kp}|\mathbf{x},\Theta) \propto \sum_{i=1}^{N} \frac{-z_{ik}w_{ip}}{2\sigma_{kp}^2}\left(x_{ip}^2 - 2x_{ip}\mu_{kp} + \mu_{kp}^2\right) - \frac{1}{2s^2}\left(\mu_{kp}^2 - 2\mu_{kp}m_0 + m_0^2\right)
$$

after which we can take the derivative with respect to μ_{kp}:

$$
\frac{\partial}{\partial \mu_{kp}} \log f(\mu_{kp}|\mathbf{x},\Theta) = \sum_{i=1}^{N} \frac{z_{ik}w_{ip}x_{ip}}{\sigma_{kp}^2} + \frac{m_0}{s^2} - \left[\sum_{i=1}^{N} \frac{z_{ik}w_{ip}}{\sigma_{kp}^2} + \frac{1}{s^2}\right]\mu_{kp}
$$

Finally, setting equal to zero and solving for μ_{kp} gives

$$
\begin{aligned}
0 &= \sum_{i=1}^{N} \frac{z_{ik}w_{ip}x_{ip}}{\sigma_{kp}^2} + \frac{m_0}{s^2} - \left[\sum_{i=1}^{N} \frac{z_{ik}w_{ip}}{\sigma_{kp}^2} + \frac{1}{s^2}\right]\mu_{kp} \\
\mu_{kp}\left[\frac{\sum_{i=1}^{N} s^2 z_{ik}w_{ip} + \sigma_{kp}^2}{\sigma_{kp}^2 s^2}\right] &= \frac{\sum_{i=1}^{N} s^2 z_{ik}w_{ip}x_{ip} + \sigma_{kp}^2 m_0}{s^2\sigma_{kp}^2} \\
\mu_{kp} &= \frac{\sum_{i=1}^{N} s^2 z_{ik}w_{ip}x_{ip} + \sigma_{kp}^2 m_0}{\sum_{i=1}^{N} s^2 z_{ik}w_{ip} + \sigma_{kp}^2}
\end{aligned}
$$

5.7 Estimated Formations

This section provides plots for all formation estimates from the three different sources of data on both offense and defense.

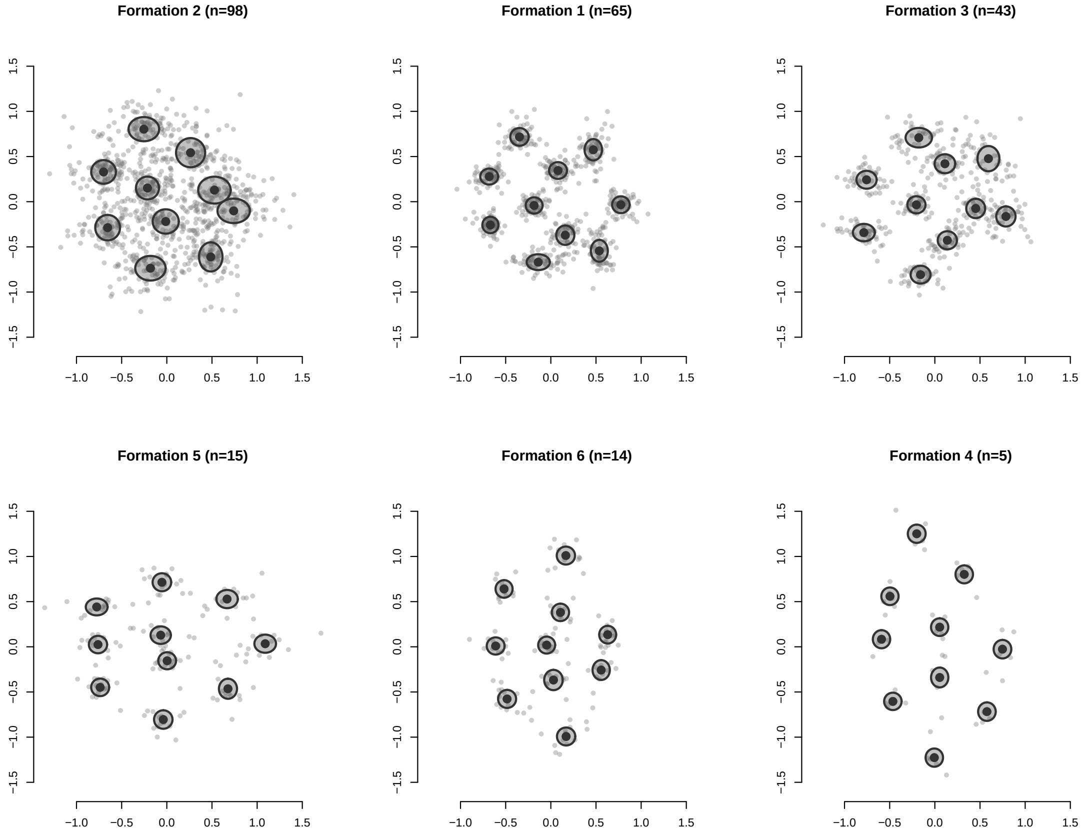


Figure 5.9: Offensive formation estimates for observations generated from multicamera tracking data using average differences, with 1-σ ellipses and classified observations, sorted by size of each cluster.

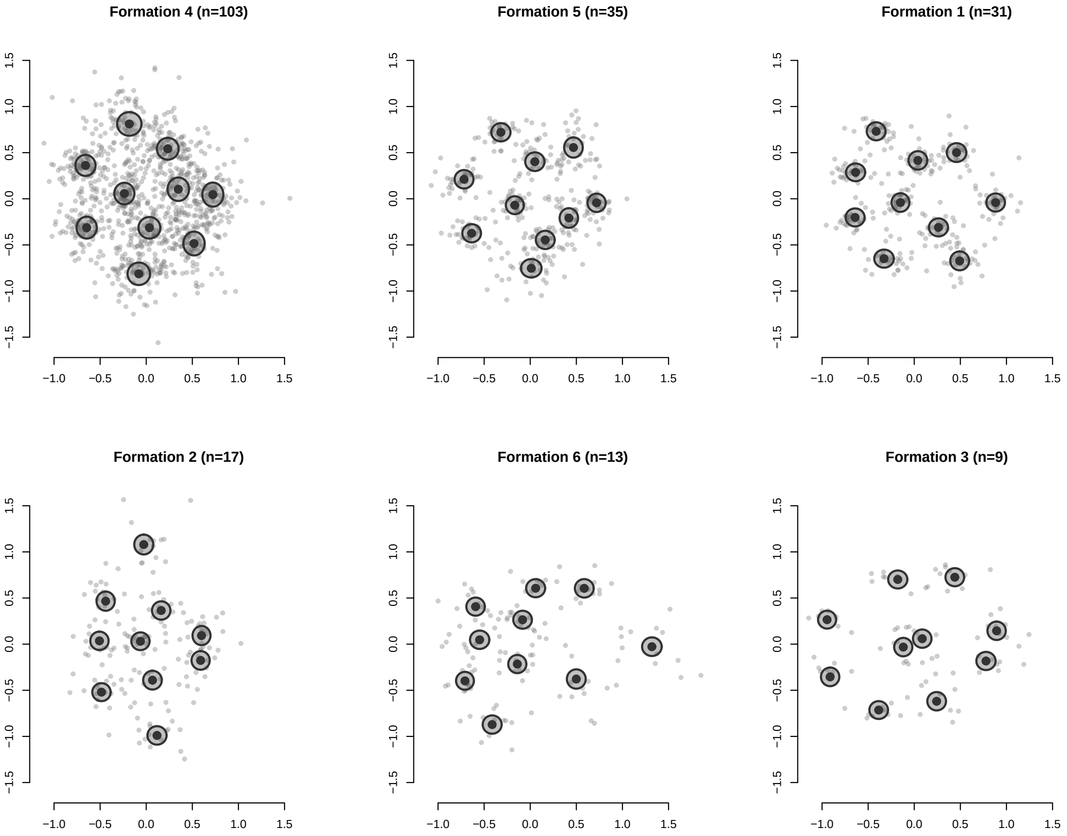


Figure 5.10: Offensive formation estimates for observations generated from broadcast tracking data using average differences, with 1-σ ellipses and classified observations, sorted by size of each cluster.

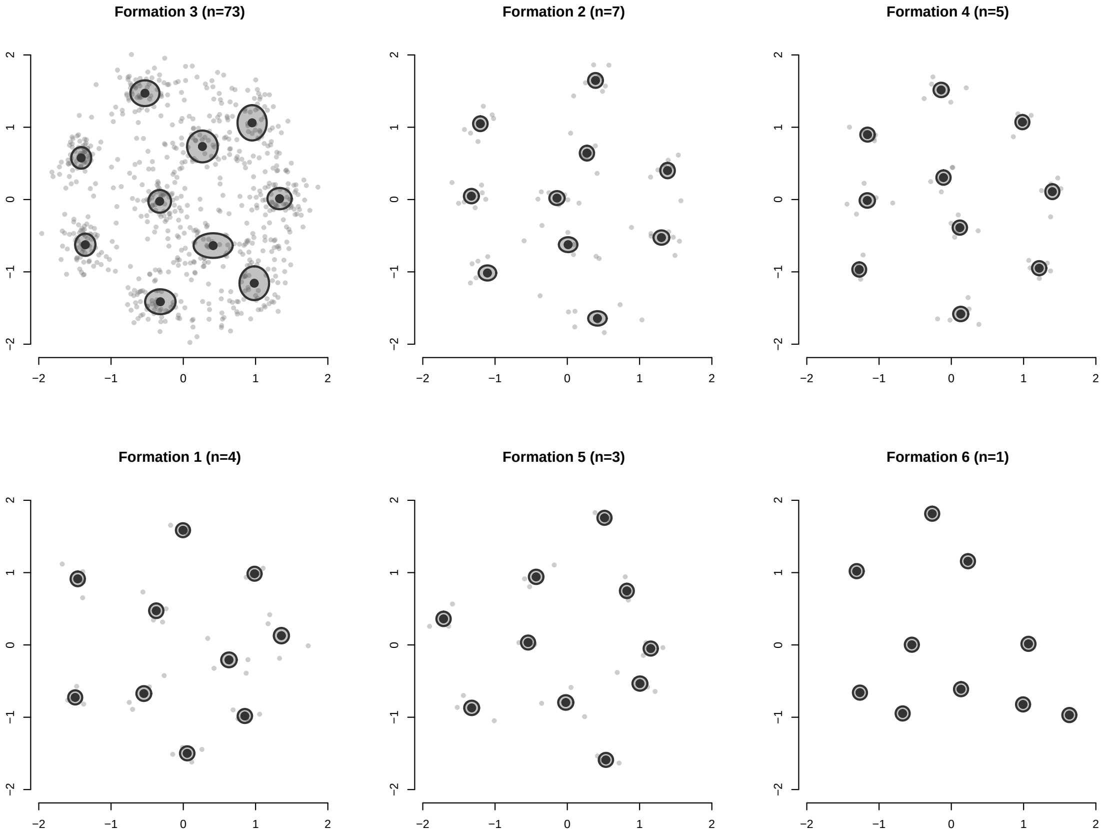


Figure 5.11: Offensive formation estimates for observations generated from ACD aligned tracking data using MEDP, with 1-σ ellipses and classified observations, sorted by size of each cluster.

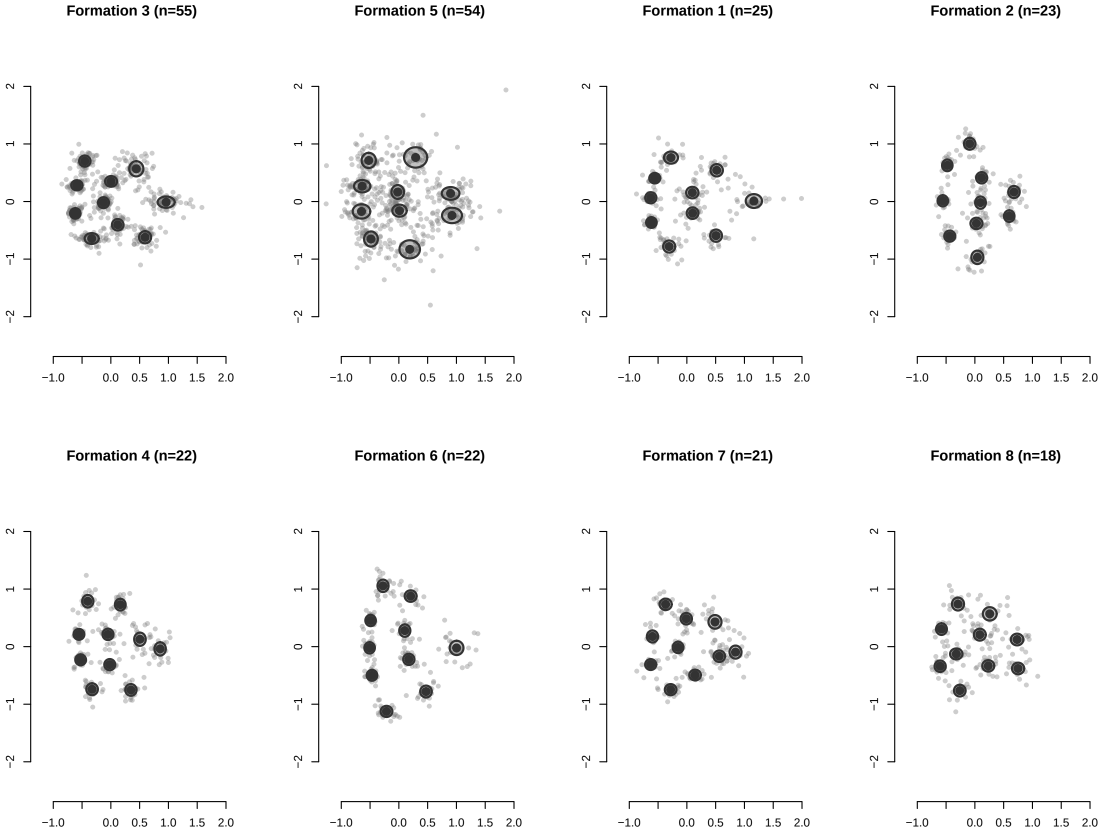


Figure 5.12: Defensive formation estimates for observations generated from multicamera tracking data using average differences, with 1-σ ellipses and classified observations, sorted by size of each cluster.

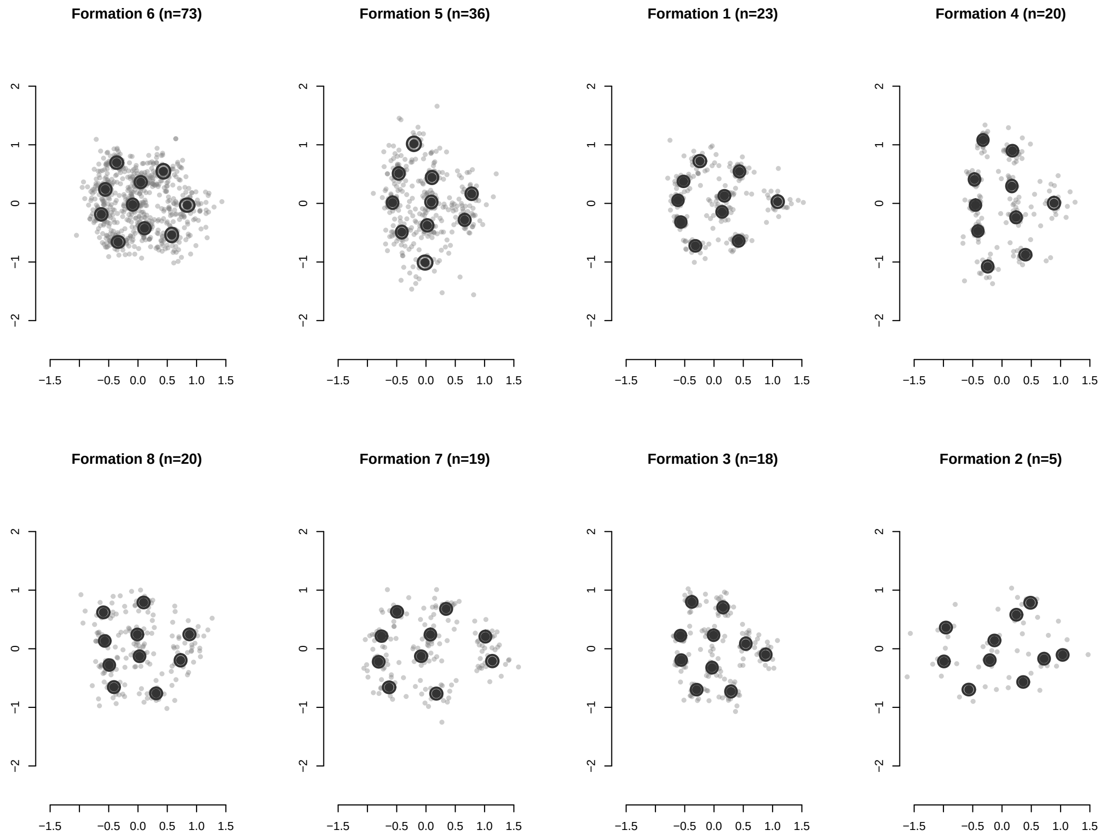


Figure 5.13: Defensive formation estimates for observations generated from broadcast tracking data using average differences, with 1-σ ellipses and classified observations, sorted by size of each cluster.

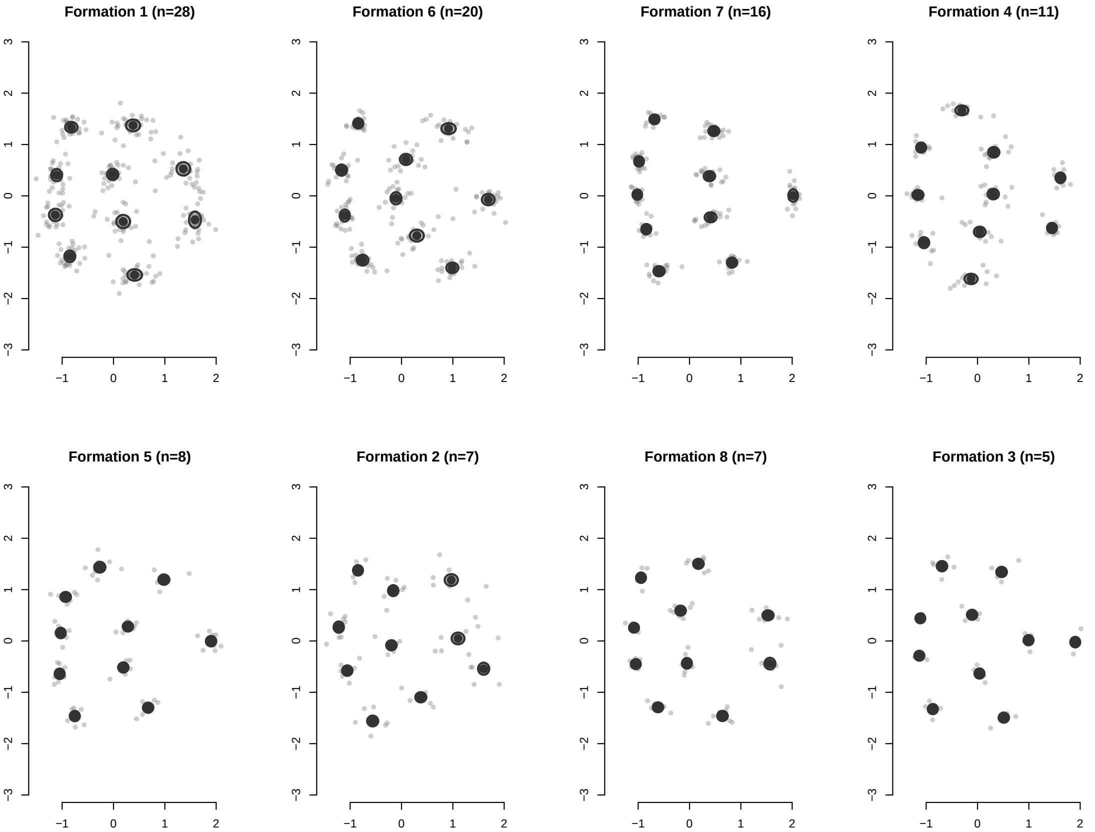


Figure 5.14: Defensive formation estimates for observations generated from ACD aligned tracking data using MEDP, with 1-σ ellipses and classified observations, sorted by size of each cluster.

Chapter 6

Conclusion

In this work, we have done a thorough review of player tracking data in sports, developed some new methods for its analysis, and shown examples of the kinds of questions that can be asked and answered with this type of data. Of particular note is the overview of broadcast tracking data in Chapter 3 and the demonstration of how it can be used despite its inherent shortcomings to answer questions of interest in Chapters 4 and 5.

Chapter 2 leveraged the relationship between the multinomial and Poisson distributions to show that Markov transition densities can be modelled as a Poisson intensity function. This results in a highly flexible nonparametric approach to density estimation that also allows for the incorporation of spatial features. Interestingly, some early methods for Markov transition density estimation using kernel densities can be shown to be special cases of the framework presented in this chapter. Crucially, representing transition densities in this way allows great flexibility in how intensities are estimated, and kernel density estimators, integrated nested Laplace approximation, and process convolutions were all used effectively in this chapter. A simulation study was used to demonstrate that in cases where boundaries impact transition densities, accounting for those boundaries by assuming a nonstationary spatial covariance function for the intensity improves density estimates. Finally, movements of the ball around the court in the NBA were modeled as a Markov process. Estimating the transition density as a non-stationary Poisson process captured a discontinuity induced by the three-point line and improved model fit.

Chapter 3 introduces broadcast tracking data and provides an overview of how it is produced, from segmentation of the broadcast video, to construction of an isomorphic mapping between the video and the court, to conversion of players in video to x- and y-coordinates, and identification of players across the game. Broadcast tracking data is an exciting development in research with tracking data because it has the potential to greatly democratize tracking data, possibly making it available for any sporting event that has been recorded once the technology is further refined. Several companies are currently working to develop commercial offerings for broadcast tracking data, but so far access to it has been limited, meaning there is ample opportunity for new research. Broadcast tracking data exhibits unique features that make it an interesting area for research, primarily that players are inconsistently observed and any analysis must account for a large amount of censoring. These features prevent existing methods for tracking data from being directly applied, requiring methodological innovation to be useful. Some examples of such innovation are shown in Chapters 4 and 5.

Chapter 4 addresses whether or not broadcast tracking data can be used to understand the impact that participation in an athletic event has on an athletes body. Without capturing internal metrics, such as heart rate, our ability to do so is limited, but sports scientists frequently use quantities such as total accelerations as a proxy measure for the physical stress undergone by an athlete. These quantities, referred to as external load metrics, are typically easily calculated from multicamera tracking data because players are continuously observed. However, because of the large amount of missing data, the viability of broadcast tracking data as a resource for sports scientists is called into question. In this chapter, a suite of external load metrics were considered and censored values for them were predicted using regularized linear regression models and random forests. The predicted quantities were highly accurate, suggesting that broadcast tracking data is a useful resource for sports scientists. Moreover, the methods that we use to provide these estimates do not require a significant amount of technical customization, which means that they could be widely adopted fairly easily.

Finally, Chapter 5 considers the problem of identifying soccer formations when the majority of data was only partially observed. Understanding what formation a team was using is important tactical information, but the ability to do so at scale is limited by the human factor of needing to watch games, so learning them directly from tracking data is a useful development. Methods have been developed to learn formations from multicamera tracking data but they break down due to the heavy censoring with broadcast tracking data. This issue was overcome using atomic configuration distance and minimum entropy data partitioning. Because atomic configuration distance is invariant to the number of players observed and permutation of their roles within the formation, it is an ideal candidate for comparing different frames of broadcast tracking data. Tracking data was aligned by minimizing this distance, after which minimum entropy data partitioning was used to summarize windows of observations. These in turn were clustered to learn general families of formations using a Gaussian mixture model. When the learned formations were used to make predictions, performance between the full multicamera tracking data and broadcast tracking data were comparable, suggesting that broadcast tracking data can successfully be used for this task.

Although the applications in this thesis are all related to applications in sports, tracking data is present in a broad range of applications, including ecology, security, transportation, and the social sciences, among others. As people continue to carry smartphones with GPS capability, this type of data will become increasingly common and understanding how to analyze it properly will become increasingly valuable. The techniques demonstrated in this thesis represent just a small section of the available methods for tracking data, but they have clear uses in other areas. Accounting for the effects of man-made barriers on peoples movement through a subway station could be accounted for using the framework in Chapter 3. Being able to estimate characteristics of animal behavior between observations, as another example, could be solved using an approach similar to that taken in Chapter 4. Similarly, using atomic configuration distance to compare collections of locations and classifying them based on their structure could be used in the physical sciences or to analyze the behavior of groups of animals.

6.1 Future Work

There are clear opportunities to do further work based on the research presented in this thesis. There is a clear link between Markov transition densities and Poisson intensities, but the theoretical foundation can be further shorn up by providing, as one example, convergence results. The Markov-Poisson approach can account for spatial non-stationarity but currently assumes that transition densities are temporally homogeneous. Figuring out how to model transition densities that vary temporally in addition to spatially would be an interesting development. Additionally, the ability of the Markov-Poisson method to scale up needs to be better understood. Because modeling a transition density with a Poisson process requires doubling the dimension of the transition space, extending this method to movement in three or more dimensions becomes computationally challenging. Similarly, currently the Markov-Poisson relationship has only been used for first-order Markov chains. Adapting it for higher order relationships is another interesting research question.

Broadly speaking, because academic work with broadcast tracking data has primarily been focused on its production rather than its analysis, there are myriad opportunities for future research. At the minimum, all methods that have already been developed for multicamera tracking data need to be modified to account for censoring in order to also apply to broadcast tracking data. More narrowly, focused on just the broadcast tracking research in this thesis, the most obvious limitation is that the research was conducted on multicamera tracking data that was censored artificially to mimic the missingness patterns of broadcast tracking data. True broadcast tracking data introduces additional uncertainty and the impact of that additional uncertainty needs to be assessed and understood, presenting a valuable opportunity for further work.

Another obvious opportunity is increasing the scale of the work done in Chapters 4 and 5. Work needs to be done to ensure the conclusions reached for estimating censored external load metrics hold when the number of teams or leagues is increased. In Chapter 5, only 5 games were available. Building a model to estimate the camera window based on those five games and using that to produce additional simulated broadcast tracking data would provide more insight into how well formations are estimated when the number of formations and amount of data is increased.

Finally, atomic configuration distance was used to align frames in the broadcast tracking data, but it could just as easily be used to measure the distance between observations and perform agglomerative hierarchical clustering. Doing so could greatly simplify the clustering process by avoiding the need to tune the mixture model and use the Hungarian algorithm to permute players within the model. In particular, removing the permutation step would significantly increase the computational speed at which clustering could be performed. Comparing the results from this approach to the ones achieved via the approach taken in Chapter 5 would be an interesting project.

Bibliography

Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Andrienko, G., Andrienko, N., Anzer, G., Bauer, P., Budziak, G., Fuchs, G., Hecker, D., Weber, H., and Wrobel, S. (2019). Constructing Spaces and Times for Tactical Analysis in Football. *IEEE Transactions on Visualization and Computer Graphics*, 14(8):1–1.

Arratia, A., Cabaña, A., and Cabaña, E. M. (2014). *Modeling Stationary Data by a Class of Generalized Ornstein-Uhlenbeck Processes: The Gaussian Case*, pages 13–24. Springer International Publishing, Cham.

Baker, S. G. (1994). The multinomial-Poisson transformation. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 43(4):495–504.

Ballan, L., Bertini, M., Del Bimbo, A., and Nunziati, W. (2007). Soccer players identification based on visual local features. *Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR 2007*, pages 258–265.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2015). *Hierarchical Modeling and Analysis for Spatial Data.* CRC Press, Boca Raton, FL, 2nd edition.

Bauwens, L., Laurent, S., and Rombouts, J. V. (2006). Multivariate GARCH models: A survey. *Journal of Applied Econometrics*, 21(1):79–109.

Baysal, S. and Duygulu, P. (2016). Sentioscope: A soccer player tracking system using model field particles. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(7):1350–1362.

Beetz, M., Hoyningen-Huene, N. V., Bandouch, J., Kirchlechner, B., Gedikli, S., and Maldonado, A. (2006). Camera-based observation of football games for analyzing multi-agent activities. *Proceedings of the International Conference on Autonomous Agents*, 2006:42–49.

Bekkers, J. (2017). Flow Motifs in Soccer: What can passing behavior tell us? pages 1–31.

Bertini, M., Del Bimbo, A., and Nunziati, W. (2005). Player identification in soccer videos. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR 05, page 2532, New York, NY, USA. Association for Computing Machinery.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B*, 36(2):192–236.

Bialkowski, A., Lucey, P., Carr, P., Yue, Y., Sridharan, S., and Matthews, I. (2014). Large-Scale Analysis of Soccer Matches Using Spatiotemporal Tracking Data. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2015-Janua(January):725–730.

Bialkowski, A., Lucey, P., Carr, P., Yue, Y., Sridharan, S., and Matthews, I. (2015). Identifying team style in soccer using formations learned from spatiotemporal tracking data. *IEEE International Conference on Data Mining Workshops, ICDMW*, 2015-Janua(January):9–14.

Birch, M. W. (1963). Maximum Likelihood in Three-Way Contingency Tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 25(1):220–233.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bojinov, I. and Bornn, L. (2016). The Pressing Game : Optimal Defensive Disruption in Soccer. pages 1–8.

Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31:307–327.

Bornn, L., Shaddick, G., and Zidek, J. V. (2012). Modeling nonstationary processes through dimension expansion. *Journal of the American Statistical Association*, 107(497):281–289.

Borresen, J. and Lambert, M. I. (2008). Quantifying training load: A comparison of subjective and objective methods. *International Journal of Sports Physiology and Performance*, 3(1):16–30.

Boyd, L. J., Ball, K., and Aughey, R. J. (2011). The reliability of minimaxx accelerometers for measuring physical activity in australian football. *International Journal of Sports Physiology and Performance*, 6(3):311–321.

Bransen, L. and Haaren, J. V. (2019). Measuring soccer players ' contributions to chance creation by valuing their passes. 15(2):97–116.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Bridgeman, L., Volino, M., Guillemaut, J. Y., and Hilton, A. (2019). Multi-person 3D pose estimation and tracking in sports. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019-June:2487–2496.

Brown, L. D., Carter, A. V., Low, M. G., and Zhang, C. H. (2004). Equivalence theory for density estimation, poisson processes and gaussian white noise with drift. *Annals of Statistics*, 32(5):2074–2097.

Buchheit, M., Allen, A., Poon, T. K., Modonutti, M., Gregson, W., and Di Salvo, V. (2014). Integrating different tracking systems in football: multiple camera semi-automatic system, local position measurement and GPS technologies. *Journal of Sports Sciences*, 32(20):1844–1857.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

Carling, C., Bloomfield, J., Nelsen, L., and Reilly, T. (2008). The Role of Motion Analysis in Elite Soccer Work Rate Data. *Sports Medicine*, 38(10):839–862.

Cervone, D., D'Amour, A., Bornn, L., and Goldsberry, K. (2016). A Multiresolution Stochastic Process Model for Predicting Basketball Possession Outcomes. *Journal of the American Statistical Association*, 111(514):585–599.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume KDD '16, pages 785–794, San Francisco, California, USA.

Chu, D., Reyers, M., Thomson, J., and Wu, L. Y. (2019). Route identification in the National Football League. *Journal of Quantitative Analysis in Sports*, 16(2):121–132.

Coutts, A. J. and Duffield, R. (2010). Validity and reliability of GPS devices for measuring movement demands of team sports. *Journal of Science and Medicine in Sport*, 13(1):133–135.

Cuevas, C., Quilón, D., and García, N. (2020). Automatic soccer field of play registration. *Pattern Recognition*, 103:107278.

Dalen, T., Jorgen, I., Gertjan, E., Havard, H. G., and Ulrik, W. (2016). Player load, acceleration, and deceleration during forty-five competitive matches of elite soccer. *Journal of Strength and Conditioning Research*, 30(2):351–359.

De Gooijer, J. G. and Zerom, D. (2000). Kernel-based Multistep-ahead Predictions of the US Short-term Interest Rate. *Journal of Forecasting*, 19:335–353.

Delaney, J. A., Duthie, G. M., Thornton, H. R., Scott, T. J., Gay, D., and Dascombe, B. J. (2016). Acceleration-based running intensities of professional rugby league match play. *International Journal of Sports Physiology and Performance*, 11(6):802–809.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38.

DeYoreo, M. and Kottas, A. (2017). A Bayesian nonparametric Markovian model for nonstationary time series. *Statistics and Computing*, 27(6):1525–1538.

Di Salvo, V., Gregson, W., Atkinson, G., Tordoff, P., and Drust, B. (2009). Analysis of high intensity activity in premier league soccer. *International Journal of Sports Medicine*, 30(3):205–212.

Diggle, P. (1985). A Kernel Method for Smoothing Point Process Data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 34(2):138–147.

Diggle, P. J. (2014). *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. CRC Press, Boca Raton, FL, 3rd edition.

Dutta, R., Yurko, R., and Ventura, S. L. (2020). Unsupervised methods for identifying pass coverage among defensive backs with NFL player tracking data. *Journal of Quantitative Analysis in Sports*, 16(2):143–161.

Dwyer, D. B. and Gabbett, T. J. (2012). Global positioning system data analysis: Velocity ranges and a new definition of sprinting for field sport athletes. *Journal of Strength and Conditioning Research*, 26(3):818–824.

Ekin, A., Tekalp, A. M., and Mehrotra, R. (2003). Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12(7):796–807.

Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4):987–1007.

Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

Fernandez, J. and Bornn, L. (2018). Wide Open Spaces : A statistical technique for measuring space creation in professional soccer. *MIT Sloan Sports Analytics Conference*, pages 1–19.

Fernández, J., Bornn, L., and Cervone, D. (2019). Decomposing the Immeasurable Sport : A deep learning expected possession value framework for soccer. In *Sloan Sports Analytics Conference*, pages 1–20.

Ferré, G., Maillet, J. B., and Stoltz, G. (2015). Permutation-invariant distance between atomic configurations. *Journal of Chemical Physics*, 143(10):1–12.

Fichman, M. and O'Brien, J. R. (2019). Optimal shot selection strategies for the NBA. *Journal of Quantitative Analysis in Sports*, 15(3):203–211.

Franks, A., Miller, A., Bornn, L., and Goldsberry, K. (2015). Characterizing the spatial structure of defensive skill in professional basketball. *Annals of Applied Statistics*, 9(1):94–121.

Friedman, J. (2001). Greedy Function Approximation : A Gradient Boosting Machine. *The Annals of Statistics*, 29(5):1189–1232.

Gabbett, T. J. and Ullah, S. (2012). Relationship between running loads and soft-tissue injury in elite team sport athletes. *Journal of Strength and Conditioning Research*, 26(4):953–960.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. CRC Press, Boca Raton, FL, 3rd edition.

Gerke, S., Linnemann, A., and Müller, K. (2017). Soccer player recognition using spatial constellation features and jersey number recognition. *Computer Vision and Image Understanding*, 159:105 – 115. Computer Vision in Sports.

Gerke, S., Müller, K., and Schäfer, R. (2015). Soccer jersey number recognition using convolutional neural networks. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 734–741.

Ghahramani, Z. and Jordan, M. (1994). Learning from incomplete data. Technical report, Massachusett Institute of Technology Artificial Intelligence Laboratory, Boston, MA.

Gloaguen, P., Mahévas, S., Rivot, E., Woillez, M., Guitton, J., Vermard, Y., and Etienne, M. P. (2015). An autoregressive model to describe fishing vessel movement and activity. *Environmetrics*, 26(1):17–28.

Goes, F. R., Meerhoff, L. A., Bueno, M. J., Rodrigues, D. M., Moura, F. A., Brink, M. S., Elferink-Gemser, M. T., Knobbe, A. J., Cunha, S. A., Torres, R. S., and Lemmink, K. A. (2020). Unlocking the potential of big data to support tactical performance analysis in professional soccer: A systematic review. *European Journal of Sport Science*, pages 1–16.

Goldman, M. and Rao, J. M. (2011). Allocative and Dynamic Efficiency in NBA Decision Making. *MIT Sloan Sports Analytics Conference 2011*, (2006):1–11.

Goldsberry, K. (2012). Courtvision: New visual and spatial analytics for the NBA. *Proc. 6th Annual MIT Sloan Sports Analytics Conference*, pages 1–7.

Goldsberry, K. and Weiss, E. (2013). The Dwight Effect : A New Ensemble of Interior Defense Analytics for the NBA. *MIT Sloan Sports Analytics Conference*, pages 1–11.

Gonçalves, B., Coutinho, D., Exel, J., Travassos, B., Lago, C., and Sampaio, J. (2019). Extracting spatial-temporal features that describe a team match demands when considering the effects of the quality of opposition in elite football. *PLoS ONE*, 14(8):1–20.

Gregson, W., Drust, B., Atkinson, G., and Salvo, V. D. (2010). Match-to-match variability of high-speed activities in premier league soccer. *International Journal of Sports Medicine*, 31(4):237–242.

Gudmundsson, J. and Horton, M. (2016). Spatio-Temporal Analysis of Team Sports – A Survey.

Gupta, A., Little, J. J., and Woodham, R. J. (2011). Using line and ellipse features for rectification of broadcast hockey video. In *2011 Canadian Conference on Computer and Robot Vision*, pages 32–39. IEEE.

Hadian, M. and Kasaei, S. (2015). Fast homography refinement in soccer videos. In *2015 9th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pages 185–188.

Hall, P. (1992). Effect of Bias Estimation on Coverage Accuracy of Bootstrap Confidence Intervals for a Probability Density. *The Annals of Statistics*, 20(2):675–694.

Halson, S. L. (2014). Monitoring Training Load to Understand Fatigue in Athletes. *Sports Medicine*, 44:139–147.

Hamid, R., Kumar, R., Hodgins, J., and Essa, I. (2014). A visualization framework for team sports captured using multiple static cameras. *Computer Vision and Image Understanding*, 118:171–183.

Hanjalic, A. (2002). Shot-boundary detection: unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):90–105.

Hastie, Trevor, Tibshirani, Robert, Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Second edition.

Henderson, S. G. and Glynn, P. W. (2001). Computing Densities for Markov Chains via Simulation. *Mathematics of Operations Research*, 26(2):375–400.

Hess, R. and Fern, A. (2007). Improved video registration using non-distinctive local image features. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Higdon, D., Swall, J., and Kern, J. (1998). Non-Stationary Spatial Modeling. (1992).

Hu, M. C., Chang, M. H., Wu, J. L., and Chi, L. (2011). Robust camera calibration and player tracking in broadcast basketball video. *IEEE Transactions on Multimedia*, 13(2):266–279.

Iqbal, U., Milan, A., and Gall, J. (2017). PoseTrack: Joint multi-person pose estimation and tracking. *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017:4654–4663.

Johnston, R., Watsford, M., Pine, M., and Spurrs, R. (2014). Standardisation of acceleration zones in professional field sport athletes. *International Journal of Sports Science and Coaching*, 9(5):1161–1168.

Kempe, M., Goes, F. R., and Lemmink, K. A. (2018). Smart data scouting in professional soccer: Evaluating passing performance based on position tracking data. *Proceedings - IEEE 14th International Conference on eScience*, 2018:409–410.

Kuhn, H. W. (1955). The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

Lacour, C. (2008). Nonparametric estimation of the stationary density and the transition density of a Markov chain. *Stochastic Processes and their Applications*, 118:232–260.

Lau, A. and McSharry, P. (2010). Approaches for multi-step density forecasts with application to aggregated wind power. *Annals of Applied Statistics*, 4(3):1311–1341.

Le, H. M., Carr, P., Yue, Y., and Lucey, P. (2017). Data-Driven Ghosting using Deep Imitation Learning. *Sloan Sports Analytics Conference*, pages 1–15.

Leonard, T. (1973). A Bayesian Method for Histograms. *Biometrika*, 60(2):297–308.

Li, G., Xu, S., Liu, X., Li, L., and Wang, C. (2018). Jersey number recognition with semi-supervised spatial transformer network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Lindgren, F. and Rue, H. (2015). Bayesian Spatial Modelling with R-INLA. *Journal of Statistical Software*, 63(19):1–25.

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society, Series B*, 73(4):423–498.

Linke, D., Link, D., and Lames, M. (2018). Validation of electronic performance and tracking systems EPTS under field conditions. *PLoS ONE*, 13(7):1–19.

Linke, D., Link, D., and Lames, M. (2020). Football-specific validity of TRACAB's optical video tracking systems. *PLoS ONE*, 15(3):1–17.

Liu, G., Luo, Y., Schulte, O., and Kharrat, T. (2020). Deep soccer analytics: learning an action-value function for evaluating soccer players. *Data Mining and Knowledge Discovery*, pages 1–29.

Liu, G. and Schulte, O. (2018). Deep reinforcement learning in ice hockey for context-aware player evaluation. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:3442–3448.

Liu, J., Carr, P., Collins, R. T., and Liu, Y. (2013). Tracking sports players with context-conditioned motion models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, J., Tong, X., Li, W., Wang, T., Zhang, Y., and Wang, H. (2009). Automatic player detection, labeling and tracking in broadcast soccer video. *Pattern Recognition Letters*, 30(2):103–113.

Lock, D. and Nettleton, D. (2014). Using random forests to estimate win probability before each play of an NFL game. *Journal of Quantitative Analysis in Sports*, 10(2):197–205.

Low, M. G. and Zhou, H. H. (2007). A complement to Le Cam's theorem. *Annals of Statistics*, 35(3):1146–1165.

Lu, C.-W., Lin, C.-Y., Hsu, C.-Y., Weng, M.-F., Kang, L.-W., and Liao, H.-Y. M. (2013a). Identification and tracking of players in sport videos. In *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*, ICIMCS 13, page 113116, New York, NY, USA. Association for Computing Machinery.

Lu, K., Chen, J., Little, J. J., and He, H. (2018). Lightweight convolutional neural networks for player detection and classification. *Computer Vision and Image Understanding*, 172:77–87.

Lu, W. L., Ting, J. A., Little, J. J., and Murphy, K. P. (2013b). Learning to track and identify players from broadcast sports videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1704–1716.

Lucey, P., Bialkowski, A., Carr, P., Morgan, S., Matthews, I., and Sheikh, Y. (2013a). Representing and discovering adversarial team behaviors using player roles. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2706–2713.

Lucey, P., Oliver, D., Carr, P., Roth, J., and Matthews, I. (2013b). Assessing team strategy using spatiotemporal data. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Part F1288:1366–1374.

Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X., and Kim, T.-K. (2017). Multiple Object Tracking: A Literature Review. *arXiv e-prints*, pages 1–18.

Maheswaran, R., Chang, Y.-h., Su, J., Kwok, S., Levy, T., Wexler, A., and Hollingsworth, N. (2014). The Three Dimensions of Rebounding. *MIT Sloan Sports Analytics Conference*, pages 1–7.

Manafifard, M., Ebadi, H., and Abrishami Moghaddam, H. (2017). A survey on player tracking in soccer videos. *Computer Vision and Image Understanding*, 159:19–46.

Martins, T. G., Simpson, D., Lindgren, F., and Rue, H. (2013). Bayesian computing with INLA: New features. *Computational Statistics and Data Analysis*, 67(2009):68–83.

McLachlan, G. J. and Basford, K. E. (1988). *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York.

McLaren, S. J., Macpherson, T. W., Coutts, A. J., Hurst, C., Spears, I. R., and Weston, M. (2018). The Relationships Between Internal and External Measures of Training Load and Intensity in Team Sports: A Meta-Analysis. *Sports Medicine*, 48(3):641–658.

Medeiros, J. (2017). How data analytics killed the Premier League's long ball game.

Mehrasa, N., Zhong, Y., Tung, F., Bornn, L., and Mori, G. (2017). Learning Person Trajectory Representations for Team Activity Analysis.

Metulini, R., Manisera, M., and Zuccolotto, P. (2018). Modelling the dynamic pattern of surface area in basketball and its effects on team performance. *Journal of Quantitative Analysis in Sports*, 14(3):117–130.

Miller, A., Bomn, L., Adams, R., and Goldsberry, K. (2014). Factorized point process intensities: A spatial analysis of professional basketball. *31st International Conference on Machine Learning, ICML 2014*, 1:398–414.

Miller, A. C. and Bornn, L. (2017). Possession Sketches : Mapping NBA Strategies. pages 1–12.

Møller, J. and Waagepetersen, R. P. (2004). *Statistical Inference and Simulation for Spatial Point Processes*. CRC Press, Boca Raton, FL.

Mullen, T., Twist, C., and Highton, J. (2019). Stochastic ordering of simulated rugby match activity produces reliable movements and associated measures of subjective task load, cognitive and neuromuscular function. *Journal of Sports Sciences*, 37(21):1–7.

Munkres, J. (1957). Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.

Nadaraya, E. (1964). On Estimating Regression. *Theory of Probability and Its Applications*, 9(1):141–142.

Nicolella, D. P., Torres-Ronda, L., Saylor, K. J., and Schelling, X. (2018). Validity and reliability of an accelerometer-based player tracking device. *PLOS ONE*, 13(2):1–13.

Nistala, A. and Guttag, J. (2019). Using Deep Learning to Understand Patterns of Player Movement in the NBA. pages 1–14.

Omidiran, D. (2011). A New Look at Adjusted Plus / Minus for Basketball Analysis.

Palmgren, J. (1981). The Fisher Information Matrix for Log Linear Models Arguing Conditionally on Observed Explanatory Variables. *Biometrika*, 68(2):563–566.

Patterson, T. A., Thomas, L., Wilcox, C., Ovaskainen, O., and Matthiopoulos, J. (2008). State-space models of individual animal movement. *Trends in Ecology and Evolution*, 23(2):87–94.

Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413.

Pesca, M. (2009). The Man Who Made Baseball's Box Score A Hit.

Politis, D. N. and Romano, J. P. (1994). The Stationary Bootstrap. *Journal of the American Statistical Association*, 89(428):1303–1313.

Politis, D. N. and White, H. (2004). Automatic Block-Length Selection for the Dependent Bootstrap. *Econometric Reviews*, 23(1):53–70.

Pons, E., García-Calvo, T., Resta, R., Blanco, H., del Campo, R. L., García, J. D., and Pulido, J. J. (2019). A comparison of a GPS device and a multi-camera video technology during official soccer matches: Agreement between systems. *PLoS ONE*, 14(8):1–12.

Power, P., Ruiz, H., Wei, X., and Lucey, P. (2017). "Not all passes are created equal:" Objectively measuring the risk and reward of passes in soccer from tracking data. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Part F1296:1605–1613.

Prakasa Rao, B. (1978). Density estimation for Markov processes using delta-sequences. *Annals of the Institute of Statistical Mathematics*, 30(A):321–328.

Previtali, F., Bloisi, D. D., and Iocchi, L. (2017). A distributed approach for real-time multi-camera multiple object tracking. *Machine Vision and Applications*, 28(3-4):421–430.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rajarshi, M. B. (2013). *Statistical Inference for Discrete Time Stochastic Processes*. SpringerBriefs in Statistics. Springer India, India.

Rampinini, E., Impellizzeri, F. M., Castagna, C., Coutts, A. J., and Wisløff, U. (2009). Technical performance during soccer matches of the Italian Serie A league: Effect of fatigue and competitive level. *Journal of Science and Medicine in Sport*, 12(1):227–233.

Reich, B. J., Hodges, J. S., Carlin, B. P., and Reich, A. M. (2006). A spatial analysis of basketball shot chart data. *American Statistician*, 60(1):3–12.

Ren, J., Xu, M., Orwell, J., and Jones, G. A. (2010). Multi-camera video surveillance for real-time analysis and reconstruction of soccer games. *Machine Vision and Applications*, 21(6):855–863.

Risser, M. D. and Turek, D. (2019). Bayesian nonstationary Gaussian process modeling: the BayesNSGP package for R. *arXiv*, http://arxiv.org/abs/1910.14101.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, volume 9914 LNCS, pages 17–35.

Roberts, S. J., Everson, R., and Rezek, I. (1999). Minimum entropy data partitioning. *IEE Conference Publication*, 2(470):844–849.

Rosenblatt, M. (1970). Density Estimates and Markov Sequences. *Nonparametric Techniques in Statistical Inference*, pages 199–210.

Roussas, G. (1969). Nonparametric estimation in Markov processes.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Rubin, D. B. (1974). Characterizing the estimation of parameters in incomplete-data problems. *Journal of the American Statistical Association*, 69(346):467–474.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society, Series B*, 71(2):319–392.

Sampaio, J., McGarry, T., Calleja-González, J., Jiménez Sáiz, S., Schelling I Del Alcázar, X., and Balciunas, M. (2015). Exploring game performance in the National Basketball Association using player tracking data. *PLoS ONE*, 10(7):1–14.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464.

Senocak, A., Oh, T.-H., Kim, J., and So Kweon, I. (2018). Part-based player identification using deep convolutional representation and multi-scale pooling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Shaw, L. and Glickman, M. (2019). Dynamic analysis of team strategy in professional football. *Barça sports analytics summit*, pages 1–13.

Sheather, S. and Jones, M. (1991). A Reliable Data-Based Bandwidth Selection Method for Kernel Density Estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):683–690.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, New York, first edition.

Simpson, D., Illian, J. B., Lindgren, F., Sørbye, S. H., and Rue, H. (2015). Going off grid: Computationally efficient inference for log-Gaussian Cox processes. *Biometrika*, 103(1):49–70.

Simpson, D., Rue, H., Martins, T. G., Riebler, A., and Sørbye, S. H. (2017). Penalising model component complexity: A principled, practical approach to constructing priors. *Statistical Science*, 32(1):1–28.

SkillCorner (2020). SkillCorner. *https://skillcorner.com.*

Spearman, W., Basye, A., Dick, G., Hotovy, R., and Pop, P. (2017). Physics-Based Modeling of Pass Probabilities in Soccer. pages 1–14.

Sportlogiq (2020). Sportlogiq. *https://sportlogiq.com/en/*.

Sykes, D., Nicholas, C., Lamb, K., and Twist, C. (2013). An evaluation of the external validity and reliability of a rugby league match simulation protocol. *Journal of Sports Sciences*, 31(1):48–57.

Tagle, F., Castruccio, S., Crippa, P., and Genton, M. G. (2019). A Non-Gaussian Spatio-Temporal Model for Daily Wind Speeds Based on a Multi-Variate Skew-t Distribution. *Journal of Time Series Analysis*, 40(3):312–326.

Thomas, A. C., Ventura, S. L., Jensen, S. T., and Ma, S. (2013). Competing process hazard function models for player ratings in ice hockey. *Annals of Applied Statistics*, 7(3):1497–1524.

Valter, D. S., Adam, C., Barry, M., and Marco, C. (2006). Validation of Prozone ®: A new video-based performance analysis system. *International Journal of Performance Analysis in Sport*, 6(1):108–119.

Vandenbroucke, N., Macaire, L., and Postaire, J. G. (2003). Color image segmentation by pixel classification in an adapted hybrid color space. Application to soccer image analysis. *Computer Vision and Image Understanding*, 90(2):190–216.

Varley, M. C. and Aughey, R. J. (2013). Acceleration profiles in elite Australian soccer. *International Journal of Sports Medicine*, 34:34–39.

Vázquez-Guerrero, J., Jones, B., Fernández-Valdés, B., Moras, G., Reche, X., and Sampaio, J. (2019). Physical demands of elite basketball during an official U18 international tournament. *Journal of Sports Sciences*, 37(22):1–8.

Venables, W. N. and Ripley, B. D. (2010). *Modern Applied Statistics with S*. Springer Publishing Company, Incorporated.

von Neumann, J. (1951). Various techniques used in connection with random digits. In Householder, A., Forsythe, G., and Germond, H., editors, *Monte Carlo Method*, pages 36–38. National Bureau of Standards Applied Mathematics Series, Washington D.C.

Vračar, P., Štrumbelj, E., and Kononenko, I. (2016). Modeling basketball play-by-play data. *Expert Systems with Applications*, 44:58–66.

Wang, K.-c. and Zemel, R. (2016). Classifying NBA Offensive Plays Using Neural Networks. *MIT Sloan Sports Analytics Conference*, pages 1–9.

Wang, Q., Zhu, H., Hu, W., Shen, Z., and Yao, Y. (2015). Discerning tactical patterns for professional soccer teams: An enhanced topic model with applications. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015-Augus:2197–2206.

Wang, X. (2013). Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3–19.

Watson, G. (1964). Smooth Regression Analysis. *Sankhy: The Indian Journal of Statistics; Series A*, 26(4):359–372.

West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models.* Springer-Verlag New York, Inc., Berlin, Heidelberg, 2nd edition.

Wiktorsson, M., Rydén, T., Nilsson, E., and Bengtsson, G. (2004). Modelling the movement of a soil insect. *Journal of Theoretical Biology*, 231(4):497–513.

Yakowitz, S. (1989). Nonparametric Density and Regression Estimation Markov Sequences without Mixing Assumptions. *Journal of Multivariate Analysis*, 136:124–136.

Yakowitz, S. J. (1985). Nonparametric Density Estimation, Prediction, and Regression for Markov Sequences. *Journal of the American Statistical Association*, 80(389):215–221.

Yam, D. (2019). A Data Driven Goalkeeper Evaluation Framework. pages 1–18.

Yu, D., Boucher, C., Bornn, L., Sportlogiq, J., and Montreal, C. (2019). Playing Fast Not Loose: Evaluating team-level pace of play in ice hockey using spatio-temporal possession data. *MIT Sloan Sports Analytics Conference*, pages 1–28.

Yu, X., Tu, X., and Ang, E. L. (2007). Trajectory-based ball detection and tracking in broadcast soccer video with the aid of camera motion recovery. In *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo*, pages 1543–1546.

Yue, Y., Lucey, P., Carr, P., Bialkowski, A., and Matthews, I. (2015). Learning Fine-Grained Spatial Models for Dynamic Sports Play Prediction. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2015-Janua(January):670–679.

Yurko, R., Matano, F., Richardson, L. F., Granered, N., Pospisil, T., Pelechrinis, K., and Ventura, S. L. (2020). Going deep: Models for continuous-time within-play valuation of game outcomes in American football with tracking data. *Journal of Quantitative Analysis in Sports*, 16(2):163–182.

Zhang, H., Kankanhalli, A., and Smoliar, S. (1993). Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.

Zhao, K., Osogami, T., and Morimura, T. (2019). Visual analytics for team-based invasion sports with significant events and Markov reward process. *SportRxiv.*