JQAS special issue on NFL player tracking data submission

Route Identification in the National Football League

An Application of Model Based Curve Clustering Using the EM Algorithm

Abstract: The National Football League (NFL) released a sample of Next Gen Stats player tracking data to the public in 2019 as part of their inaugural Big Data Bowl. The released data describes every play from the first six weeks of the 2017 season and measures active player positions on the field every tenth of a second. Each play is a sequence of spatial-temporal measurements that vary in length depending on the duration of the play. In this paper, we demonstrate how model-based curve clustering of observed player trajectories can be used to identify the routes run by eligible receivers on offensive passing plays. We use a Bernstein polynomial basis function to represent cluster centers, and the Expectation Maximization algorithm to learn the route labels for each of the 34,698 routes run on the 6,963 passing plays in the data set. We go on to suggest ideas for new potential receiver metrics that account for receiver deployment. The resulting route labels can also be paired with film to enable streamlined queries of game film.

Keywords: Model-based curve clustering, Route identification, Functional data, Expectation Maximization algorithm

1 Introduction

Curve clustering is the process of finding a latent class structure for observations of functional data and has wide applications across industries such as biology, finance and environmental science [Aghabozorgi et al., 2015]. Many methodologies have been developed to deal with such data [Dong et al., 2018]. These methodologies fall into four main methods; shape-based, compression based dissimilarity, feature based, and model-based clustering [Aghabozorgi et al., 2015].

This paper will focus on a model-based route clustering methodology using Gaussian mixture methods. These have been used in previous literature to cluster gene expression [McNicholas and Murphy, 2010], recognize Arabic characters [AlShaher, 2018], and distinguish regions based on temperature data [Bouveyron and Jacques, 2011]. In sport literature, model-based clustering has been used to cluster swimming progression in competition [Leroy et al., 2018], and in basketball to cluster player trajectories in the National Basketball Association (NBA) [Miller and Bornn, 2017].

The player trajectory data used by Miller and Bornn has been available to NBA teams since 2013 [Nba, 2013]. Similar data has recently been made available in the NFL by Next-Gen Stats and affiliate organizations [Nfl, 2019]. The player tracking data in the NFL is collected differently than in the NBA, utilizing a chip in the shoulder pads of players for the entirety of collection rather than computer vision tools. Although the data is collected in different ways, the player tracking data is fundamentally the same and allows for analysts in each sport to identify player movement over time. Capturing these insights has proven valuable in the NBA - leading the NFL to pursue the same approach.

For this reason, the NFL hosted the inaugural NFL Big Data Bowl, a competition that released player tracking data to the public for the first time, in January of 2019 to help teams gain insights from this data. Previous work in football had relied on play-by-play event data which was popularized and made readily available by the nflscrapr package [Horowitz et al., 2018] for the R Project for Statistical Computing. This led to work such as Expected Points Added, Win Probability Added, and Wins Above Replacement models in football [Yurko et al., 2019] which had already been readily available in sports such as golf [Broadie, 2011], basketball [Stern, 1994] and baseball [Baumer et al., 2013].

The data made available to the public for the NFL Big Data Bowl was NFL Next-Gen Stats tracking data which is gathered by Zebra Technologies and Wilson Sporting Goods through the use of radio-frequency identification (RFID) chips. These chips measure the field position (x, y) of each player and the ball at tenth of a second increments. Key events are listed at the moment they occur, such as the snap, a tackle, or a fumble. Using this information, the shape of each route run by every receiver can be considered a finite function, with a known start and ending point.

1.1 Routes in the NFL

A route in football is the path or pattern that an eligible receiver runs on a passing play. Coaches and quarterbacks plan the route for every eligible receiver prior to the start of the play. In some cases receivers may have an option to run one of many predetermined routes where they decide on the specific route to run based on how the defense is set up. These are called option routes. Additionally, the predetermined routes are subject to change. A quarterback can communicate to his receivers new routes before the play begins or a receiver can make an adjustment while running the route.

Football teams of all levels dedicate staff to tagging videos with the routes run by receivers on passing plays. Doing so is long and tedious work but it allows teams to query plays that meet certain search criteria. For example, to prepare for a playoff game, a coach may want video of all plays where a specific team has receivers run a three route combination including the flat, in, and post routes. The same coach may want to evaluate his own team's success and may want to know the results of all plays where there were receivers on the same side of the field that run an out route and a go route. Further, a defensive back in preparation for a marquee match-up with a top wide receiver may want to watch film of all the plays where that wide receiver ran a post route.

With the new player tracking data and the long amount of hours spent tagging film, the automated detection and labeling of routes is of interest to football teams. Before the availability of player tracking data, attempts were made to use computer vision and machine learning to identify routes and formations from game film [Ajmeri and Shah, 2012]. Now that the NFL offers tracking data, machine learning techniques have been instead focused on route identification through feature based supervised learning techniques [Hochstedler and Gagnon, 2017].

We propose using a model-based unsupervised learning approach to clustering routes using the new player tracking data. We will implement this by using Bézier curves to define cluster means and then learn the cluster parameters and membership probabilities using the Expectation-Maximization algorithm. We will then label the clusters and provide direction for the use of these labels in further analysis.

Labelling the routes run on a given play provide the information needed for a more nuanced analysis of receiver play in the NFL. Statistics like targets over expectation and air yards over expectation require information about receiver deployment on passing plays. Automating this labelling will help save hundreds of hours manually tagging plays and make querying for plays of interest easier while making the evaluation of receivers more nuanced.

This sharing of ideas and methodologies across sports and industries leads us to present our route identification methods for routes in the NFL.

2 Bernstein Basis & Bézier Curves

In pursuing a model-based approach to clustering routes, we will make specific use of Bézier curves. First established by Pierre Étienne Bézier, these curves are capable of representing complicated "free-form" shapes of infinite points. The fundamentals of this approach originate in Bernstein basis polynomials [N. Bernstein, 1911]. The connection between Bernstein Basis Polynomials and Bézier curves is straightforward. First, we can define the basis polynomials for a degree P on $t \in [0,1]$ by

$$b_p^P(t) = \binom{P}{p} t^p (1-t)^{P-p}, \; p = 0, \dots, P \tag{1}$$

Extending these polynomials to the Bézier setting is then done through applying a weight to each term of the polynomial, called control points, by

$$\mathbf{B}(t; \boldsymbol{\theta}) = \sum_{p=0}^{P} \theta_p b_p^P(t), \; t \in [0,1], \tag{2}$$

with control points $\boldsymbol{\theta}_0, \dots, \boldsymbol{\theta}_P$. The output for a given input t, given the control points $\boldsymbol{\theta}$, are the coordinates of the corresponding point on the Bézier curve. The collection of outputs over the inputs t then forms the Bézier curve.

Bézier curves are parametric curves and are easy to implement. They generalize well to higher dimensions and make for an excellent general tool. Though their application has been extensively explored in computer graphics, groundwork has been started with player tracking data in the NBA [Miller and Bornn, 2017]. Our work will look to expand this introductory work into the NFL for receiver routes.

3 Fitting Bézier Curves

Adapting the work of [Gaffney, 2004], let $\mathbf{Y}$ be a set n observed player trajectories, $\{\mathbf{y}_1, \dots, \mathbf{y}_n\}$. Each observed curve y_i has m_i observations along its trajectory. These points describe the location at each observed time in two dimensional space. Therefore, y_i is a $m_i \times 2$ matrix.

Each observed trajectory is measured at times $\mathbf{t}_i$. We assume that each curve can be described by a Bézier curve with degree P, defined by $\boldsymbol{\theta}$, which is a $(P+1) \times 2$ matrix, and an additive Gaussian error term $\boldsymbol{\epsilon}_i$, a $m_i \times 2$ matrix. The jth term of $\boldsymbol{\epsilon}_i$ is $\epsilon_{ij} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, and ϵ_{ij}, σ^2 which are both 1×2 matrices ($i = 1, 2, \dots, n$, $j = 1, 2, \dots, m_i$).

Since each trajectory can be described by a Bézier curve we are using a Bernstein polynomial basis expansion where the control points are the coefficients or weights to the basis function. To fit the control points of a Bézier curve, consider the form of Equation (2). The structure of this equation is similar to a regression model. Using this as motivation, we can then naturally summarize the relationship between time points and control points with the following regression equation:

$$\mathbf{y}_i = \mathbf{T}_i\boldsymbol{\theta} + \boldsymbol{\epsilon}_i \tag{3}$$

The regression matrix, $\mathbf{T}_i$, has dimension $m_i \times (P+1)$ and is evaluated at t_{ij}, where $t_{ij} \in [0,1]$.

The fitting of Bézier curves can then be seen as equivalently fitting a multivariate linear regression.

$$\mathbf{T}_i = \begin{bmatrix} b_0^P(t_{i1}) & b_1^P(t_{i1}) & \dots & b_P^P(t_{i1}) \\ b_0^P(t_{i2}) & b_1^P(t_{i2}) & \dots & b_P^P(t_{i2}) \\ \vdots & \vdots & \ddots & \vdots \\ b_0^P(t_{im_i}) & b_1^P(t_{im_i}) & \dots & b_P^P(t_{im_i}) \end{bmatrix} \tag{4}$$

Now that we have the tools to fit a Bézier curve we will discuss how we use these curves to define cluster means and how to calculate how well an observed trajectory fits to a Bézier curve.

4 Model Based Curve Clustering

Assuming any route run by a player approximates one of the finite predefined routes, the goal of our work then becomes to try and classify each observed route as a realization of one of the existing predefined routes. This is equivalent to clustering the n routes into K clusters, which we will refer to as labeling the route. Labeling will be done iteratively with labels updated at each step. As such, we let $\mathbf{z} = (z_1, \dots, z_n)$ be the current label of a given route, such that $z_i \in \{1, \dots, K\}$. Our assumption claims that there exists some correct set of labels $\mathbf{z}*$ for our collection of routes.

The regression matrix defined in (4) can be used to define the conditional Probability Density Function (PDF) of $\mathbf{y}_i$ given $\mathbf{t_i}$ as $f(\mathbf{y_i}|\mathbf{t_i}) = \mathcal{N}(\mathbf{y}_i|\mathbf{T}_i\boldsymbol{\theta}, \sigma^2\mathbf{I})$. Now we can consider a mixture of K of these conditional distributions. Then the probability of observing the i^{th} curve can be defined as:

$$P(\mathbf{y}_i|\mathbf{T}_i, \boldsymbol{\theta}_k, \sigma_k^2) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mathbf{y}_i|\mathbf{T}_i\boldsymbol{\theta}_k, \sigma_k^2\mathbf{I}) \tag{5}$$

where α_k is the mixing weights of the k^{th} cluster, and $\sum_{k=1}^{K} \alpha_k = 1$. The log-likelihood of observing all n curves is then defined as the log of the product of the probability of observing each curve. Which in turn is the sum over all the curves of the log of the probability of observing the curve:

$$\log(P(\mathbf{Y}|\mathbf{T}, \Theta)) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mathbf{y}_i|\mathbf{T}_i\boldsymbol{\theta}_k, \sigma_k^2\mathbf{I}) \tag{6}$$

We have that z_i is the cluster membership for curve i. Then the joint density of $\mathbf{y}_i$ and z_i is

$$P(\mathbf{y}_i, z_i|\mathbf{t}_i) = \alpha_{z_i} \mathcal{N}(\mathbf{y}_i|\mathbf{T}_i\boldsymbol{\theta}_{z_i}, \sigma_{z_i}^2\mathbf{I}) \tag{7}$$

This leads us to use the Expectation Maximization (EM) Algorithm introduced by Dempster [Dempster et al., 1977] to learn the Maximum Likelihood Estimates (MLE) for a model with a latent variable. The algorithm consists of three parts: Initialization, the Expectation Step, and the Maximization step. We will first discuss here the general process for the Expectation and Maximization steps. We will discuss the specifics of the data pre-processing and initialization procedure in the next section.

4.1 Expectation Step

In this step, the current estimates are used to evaluate the conditional expectation. Based on Bayes' rule, the membership probability π_{ik} that the i^{th} curve was generated from cluster z_i is defined as

$$\pi_{ik} = P(z_i = k|\mathbf{y}_i, \mathbf{T}_i) \tag{8}$$

It can be calculated by computing the probability that the i^{th} curve is generated from cluster k

$$\pi_{ik} = \alpha_k \frac{P(\mathbf{y}_i|\mathbf{T}_i, z_i = k)}{P(\mathbf{y}_i|\mathbf{T}_i)} \tag{9}$$

which is the product of the the probability of generating each m_i observed points on the curve from cluster k.

$$\pi_{ik} = \alpha_k \frac{\prod_{j=1}^{m_i} \mathcal{N}(\mathbf{y}_{ij}|\mathbf{T}_{ij}\boldsymbol{\theta}_k, \sigma_k^2\mathbf{I})}{\sum_{k=1}^{K} \prod_{j=1}^{m_i} \mathcal{N}(\mathbf{y}_{ij}|\mathbf{T}_{ij}\boldsymbol{\theta}_k, \sigma_k^2\mathbf{I})} \tag{10}$$

In practice we can scale each $\mathcal{N}(\mathbf{y}_{ij}|\mathbf{T}_{ij}\boldsymbol{\theta}_k, \sigma_k^2\mathbf{I})$ by a constant without changing π_{ik} to prevent underflow errors.

With regards to implementation, the heaviest computational part of calculating π_{ik} is computing $\mathcal{N}(\mathbf{y}_{ij}|\mathbf{T}_{ij}\boldsymbol{\theta}_k, \sigma_k^2\mathbf{I})$ for every observed point. In practice, this calculation is performed in parallel in order to efficiently implement the algorithm.

4.2 Maximization Step

The updates rules for the parameters $\boldsymbol{\theta}_k, \sigma_k^2, \alpha_k$ are found by maximizing the log-likelihood. $\hat{\alpha}_k$ is the mean posterior probability that the i^{th} curve was generated from cluster z_i, and π_{ik} is obtained from the E-step:

$$\hat{\alpha}_k = \frac{1}{n}\sum_{i=1}^{n} \pi_{ik} \tag{11}$$

$\hat{\boldsymbol{\theta}}_k, \hat{\sigma}_k^2$ are found through weighted least squares where the weights matrix $\mathbf{W}_k$ is a diagonal matrix with the diagonal elements the elements of the vector $\mathbf{w}_k$, defined as

$$\mathbf{w}_k = [\underbrace{\underbrace{\pi_{1k}, \ldots, \pi_{1k}}_{m_1 \text{ elements}}, \underbrace{\pi_{2k}, \ldots, \pi_{2k}}_{m_2 \text{ elements}}, \cdots \underbrace{\pi_{nk}, \ldots, \pi_{nk}}_{m_n \text{ elements}}}_{\sum_{i=1}^{n} m_i \text{ elements}}] \tag{12}$$

This leads to the following weighted least squares solutions (see for example [Gaffney, 2004], [Chamroukhi, 2013], [Faria and Soromenho, 2010])

$$\hat{\boldsymbol{\theta}}_k = (\mathbf{T}^\intercal \mathbf{W}_k \mathbf{T})^{-1}\mathbf{T}^\intercal \mathbf{W}_k \mathbf{Y} \tag{13}$$

$$\hat{\sigma}_k^2 = \frac{1}{\sum_{i=1}^{n} \pi_{ik}}(\mathbf{Y} - \mathbf{T}\hat{\boldsymbol{\theta}}_k)^\intercal \mathbf{W}_k (\mathbf{Y} - \mathbf{T}\hat{\boldsymbol{\theta}}_k) \tag{14}$$

Since we have assumed the variance covariance matrix of the Normal distribution is diagonal we take only the diagonal elements of $\hat{\sigma}_k^2$.

In practice, the parameters $\hat{\boldsymbol{\theta}}_k$ and $\hat{\sigma}_k^2$ for different clusters can be computed in parallel and the $\mathbf{W}_k$ matrices can be stored as sparse matrices. This gives performance improvements when the number of clusters gets large.

We repeat the E and M steps until the change in log likelihood reaches a pre-defined tolerance (10^{-6} in our case).

5 Route Identification

5.1 NFL Tracking Data

The data has a play ID variable that contains all players positional and directional data within each play. There is an event variable indicating the start, end, and key moment within each play. These events can be easily pulled from the data to identify the play ID for each passing play, which contains the corresponding position of each player on the field for the play's duration. We only looked at those events indicating a pass was attempted (6963 plays). Additionally only trajectories of offensive players who play Wide Receiver, Tight End, Running Back, or Full Back are analyzed (33967 trajectories). While other players can be eligible receivers we have no clear way of identifying them in the data.

5.2 Pre-processing

We now perform 3 transformations to the trajectory data.

1. **Cut Trajectories**: Each player trajectory in its raw form starts before the event "*ball snap*", and continues until the play is over, indicated by a number of possible events defined in the appendix. We only consider trajectory data beginning from when the ball is snapped, and ending when the play is over; the rest is cut from our analysis. The shift before the snap is outside the scope of this paper.
2. **Standardize to Line of Scrimmage and Play Direction**: Since offensive plays start at varying distances from the target end zone, and the target end zone shifts for a given team every quarter; trajectories that represent similar routes will appear to be different x y space. Therefore we move all routes to a common line of scrimmage and orient player trajectories so that all routes are moving towards a common end zone.
3. **Flip Trajectories**: We would expect any route run from one side of the quarterback to be approximately the mirror of the same route run from the other side (see Figure 3). All receiver routes are flipped to start from the same side of the quarterback, and translated to a single point of origin (see Figure 1)

These pre-processing steps make it easier to identify common patterns and route labels. A shallow "in" route should look approximately the same now regardless of who ran it, and from where on the field they started from at the snap. After the clustering process we use the features of the non-processed data for further investigation.

These simplifications make it easier to identify common patterns and route labels. An example of what the pre-processed data looks like is available in Figure 1. After the clustering process we use the features of the non-transformed data for further investigation.

5.3 Initialization

Finally, in order to run the EM algorithm we need initial values for the curve centers. We determine these curve centers by assigning each observed curve to an initial cluster and then calculating initial control points, variance parameters, and cluster weights based on the curves in each cluster.

We use k-means clustering on the last observed point on each observed trajectory to use for the initialization of the cluster centers. This is a sensible idea since each trajectory has already been transformed to start at the same point, so much of the information about what route was run on the play is available in the last observed point.

This is implemented on $K = 30$ clusters. After getting our initial weights and parameters, the EM algorithm is run for four steps. Unfortunately the data does not include the true route run by each player, so labelling the curves was done manually.

5.4 Implementation

We implemented the curve clustering algorithm on the data for $K = 30$ clusters. This data consists of 33,967 routes from 6,963 passing plays. In total there are 1,438,133 measurements for an average of 42 measurements per route.

We used a computer with 8 CPUs and 52 GB of RAM. On average each Expectation step takes 1,910 seconds and each Maximization step takes 19 seconds. We run the algorithm for 4 steps. The log likelihood at each step is -6090879, -6077207, -6069274, -6064259. The total run time was 7,745 seconds.

5.5 Route Labelling

We have identified routes with similar structure but it is now our goal to add football context to our work by labelling the clusters. The cluster means obtained from our curve clustering process resemble those of route trees in football see Figure 3. We use labels provided by Ben Minaker formerly of the Simon Fraser University Football Team to manually label the mean curve of each cluster. Some clusters end up representing similar routes so we end up condensing the 30 clusters into 12 route groups. These route groups can be plotted back in "football space" and are displayed in Figure 5. This then leads us to perform brief sanity checks that our labelling process worked. In Figure 6 we can compare the route distribution across positions. Despite, the algorithm having limited indicators of a player's position the routes assigned to each position align with our intuition of player positions. Finally, we can look at very basic trends about route usage. For example we could look at Figure 7 to see which players run which routes most often or Figure 8 to see which 3 WR design play concepts are run most often. These plots provide basic information provided by our labelled routes. However, we believe that there is much more that can be discovered by using route labels in analysis. We have implemented basic versions of them but leave it to future work to develop them fully.

6 Future Work

There are 2 main directions for future work in this space. The first is to improve the clustering process. One way in which this can be achieved is by improving the computational efficiency of the clustering process. This would allow for more clusters to be fit. We've considered that perhaps clustering based on derivatives and second derivatives of the position vectors may yield improved results. A major drawback is the ability to identify a comeback vs. a go route as the functions look nearly identical to our clustering algorithm. Derivatives of the function will show velocity over time, and second derivatives acceleration. Finally, there are more complex models in [Gaffney, 2004] that may yield better clustering results.

The second is to augment the analysis of football players. As mentioned in the previous section, we see major opportunities to use these labelled routes to understand and account for player usage across receiver statistics.

Additionally, the results obtained thus far cannot be proven to work without knowing the true route name for each of the passing plays in our data set. Our only method for checking the results is to compare to our intuition of player tendencies. In the future we hope to work with teams to calibrate our algorithm for accuracy route labelling.

6.1 Potential Uses

The automated labelling of routes provides potential avenues for more in depth analysis, e.g. with labelled routes on each play we can build models to understand player deployment, receiver statistics that account for usage [Rossler, 2019], and build better defensive statistics for coverage players.

We can calculate statistics like type of route run over expectation per 100 plays (see Figure 9) to understand player usage compared to an average player while accounting for position and game situation. These can then be further used to cluster players based on their usages above expectation of an average player. This can then be extended to replicate the work of [Rossler, 2019] to compute targets over expectation for various wide receivers. This can be broken down in a number of different ways. In Figure 10 we break down Targets Over Expectation by 4 different routes to see which receivers are being targeted more often then expected on specific route types. This methodology can be extended to the concept of Air Yards. As provided in [Horowitz et al., 2018], Air Yards can be used to contextualize and quantify receiving opportunities. We plot in Figure 11 the top players for Air Yards Over Expectation per 100 Routes from a preliminary model. As we have demonstrated briefly here there is a large array of applications that can use route labels. One that we did not provide an example for is improving the evaluation of defensive players. The labelling of routes from tracking data is therefore valuable for further analysis as it is for automated film tagging.

7 Conclusion

In this work we demonstrate a method for labelling routes for player trajectory data of varying lengths in the National Football League. We do so by using a model based clustering approach and the Expectation Maximization algorithm. The probabilistic model works as a mixture of Gaussian distributions centered at Bézier curves. This provides the potential to understand player usages, efficiencies, and improve defensive evaluation.

References

[Aghabozorgi et al., 2015] Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. (2015). Time-series clustering – a decade review. *Information Systems*, 53:16–38.

[Ajmeri and Shah, 2012] Ajmeri, O. and Shah, A. (2012). Using computer vision and machine learning to automatically classify nfl game film and develop a player tracking system. In *Proceedings of the 2012 MIT Sloan Sports Analytics Conference.*

[AlShaher, 2018] AlShaher, A. A. (2018). Arabic character recognition using regression curves with the expectation maximization algorithm. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 12(12):1087–1091.

[Baumer et al., 2013] Baumer, B., Jensen, S., and Matthews, G. (2013). Openwar: An open source system for evaluating overall player performance in major league baseball. *Journal of Quantitative Analysis in Sports*, 11.

[Bouveyron and Jacques, 2011] Bouveyron, C. and Jacques, J. (2011). Model-based clustering of time series in group-specific functional subspaces. *Advances in Data Analysis and Classification*, 5(4):281–300.

[Broadie, 2011] Broadie, M. (2011). Assessing golfer performance on the pga tour. *Interfaces*, 42.

[Buccaneers.com, 2015] Buccaneers.com (2015). Red chalk talk: Route tree (3 of 4). [Online; posted 30-August-2015].

[Chamroukhi, 2013] Chamroukhi, F. (2013). Robust em algorithm for model-based curve clustering. *arXiv e-prints*, page arXiv:1312.7022.

[Dempster et al., 1977] 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 (Methodological)*, 39(1):1–38.

[Dong et al., 2018] Dong, J. J., Wang, L., Gill, J., and Cao, J. (2018). Functional principal component analysis of glomerular filtration rate curves after kidney transplant. *Stat Methods Med Res*, 27(12):3785–3796.

[Faria and Soromenho, 2010] Faria, S. and Soromenho, G. (2010). Fitting mixtures of linear regressions. *Journal of Statistical Computation and Simulation*, 80(2):201–225.

[Gaffney, 2004] Gaffney, S. (2004). *Probabilistic Curve-Aligned Clustering and Prediction with Mixture Models.* PhD thesis, University of California, Irvine.

[Hochstedler and Gagnon, 2017] Hochstedler, J. and Gagnon, P. T. (2017). American football route identification using supervised machine learning. In *Proceedings of the 2017 MIT Sloan Sports Analytics Conference.*

[Horowitz et al., 2018] Horowitz, M., Yurko, R., and Ventura, S. (2018). *nflscrapR: Compiling the NFL Play-by-Play API for easy use in R.* R package version 1.8.1.

[Leroy et al., 2018] Leroy, A., MARC, A., DUPAS, O., REY, J. L., and Gey, S. (2018). Functional data analysis in sport science: Example of swimmers' progression curves clustering. *Applied Sciences*, 8(10):1766.

[McNicholas and Murphy, 2010] McNicholas, P. D. and Murphy, T. B. (2010).

Model-based clustering of microarray expression data via latent gaussian mixture models. *Bioinformatics*, 26(21):2705–2712.

[Miller and Bornn, 2017] Miller, A. C. and Bornn, L. (2017). Possession sketches : Mapping nba strategies. In *Proceedings of the 2017 MIT Sloan Sports Analytics Conference*.

[N. Bernstein, 1911] N. Bernstein, S. (1911). Démonstration du théorème de weierstrass fondée sur le calcul des probabilités. *Communications de la Société Mathématique de Kharkov 2*, 13.

[Nba, 2013] Nba (2013). Nba partners with stats llc for tracking technology. [Online; posted Sep 5, 2013].

[Nfl, 2019] Nfl (2019). Nfl next gen stats. https://operations.nfl.com/the-game/technology/nfl-next-gen-stats/. Accessed: 2019-04-23.

[Rossler, 2019] Rossler, B. (2019). Introducing targets above expectation.

[Stern, 1994] Stern, H. S. (1994). A brownian motion model for the progress of sports scores. *Journal of the American Statistical Association*, 89(427):1128–1134.

[Yurko et al., 2019] Yurko, R., Ventura, S., and Horowitz, M. (2019). nflwar: a reproducible method for offensive player evaluation in football. *Journal of Quantitative Analysis in Sports*.

*List of Figures

Fig. 1 A sample of 500 transformed curves according to our pre-processing steps — **15**
Fig. 2 Cluster Means for 30 Clusters — **16**
Fig. 3 Example of a Route Tree [Buccaneers.com, 2015]. — **17**
Fig. 4 Route tree (a) when compared to the results of the clustered means (b) — **18**
Fig. 5 Labelled cluster means plotted with respect to the pre-transformed space — **19**
Fig. 6 Routes per Position — **20**
Fig. 7 Players who run the most of each route — **21**
Fig. 8 3 WR Designs — **22**
Fig. 9 RB Routes Run Over Expectation per 100 Plays — **23**
Fig. 10 Targets Over Expectation per 100 of each Route by Route — **24**
Fig. 11 Air Yards Over Expectation per 100 Routes — **25**

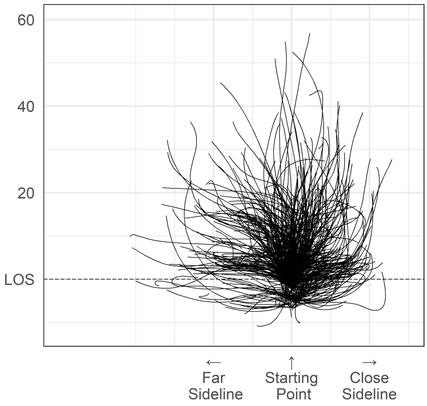


Fig. 1: A sample of 500 transformed curves according to our pre-processing steps

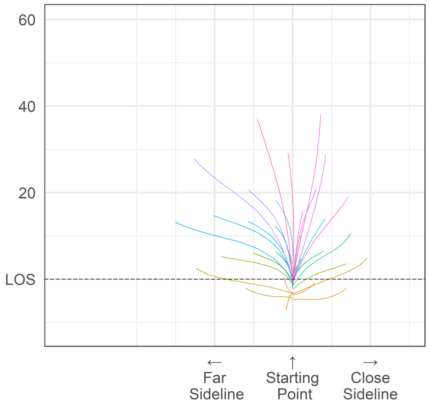


Fig. 2: Cluster Means for 30 Clusters

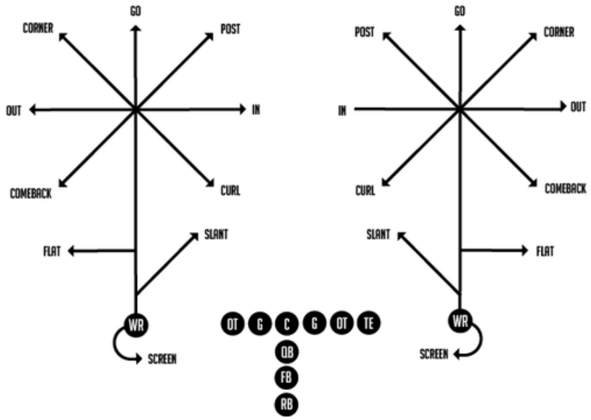
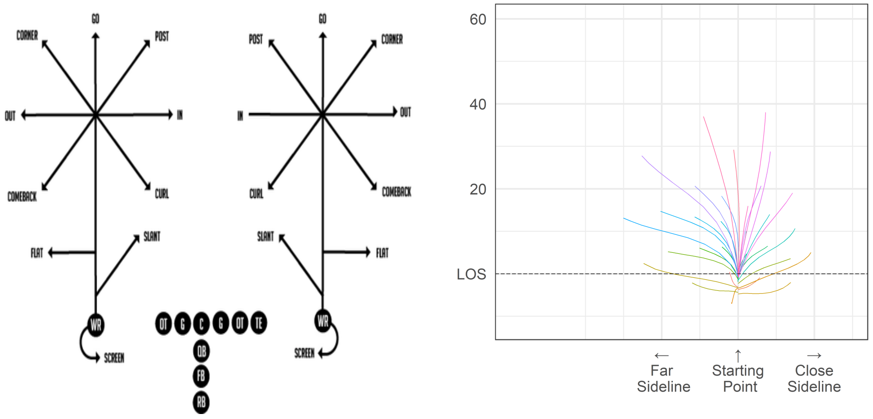


Fig. 3: Example of a Route Tree [Buccaneers.com, 2015].



(a) Route tree for offensive receivers

(b) Clustered means

Fig. 4: Route tree (a) when compared to the results of the clustered means (b)

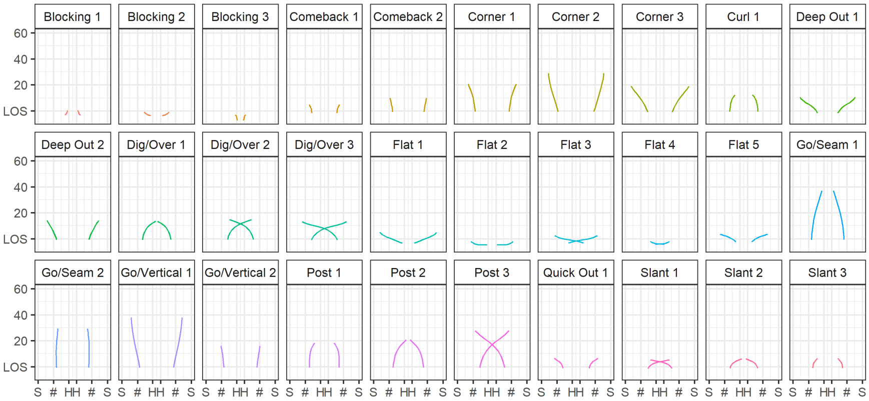


Fig. 5: Labelled cluster means plotted with respect to the pre-transformed space

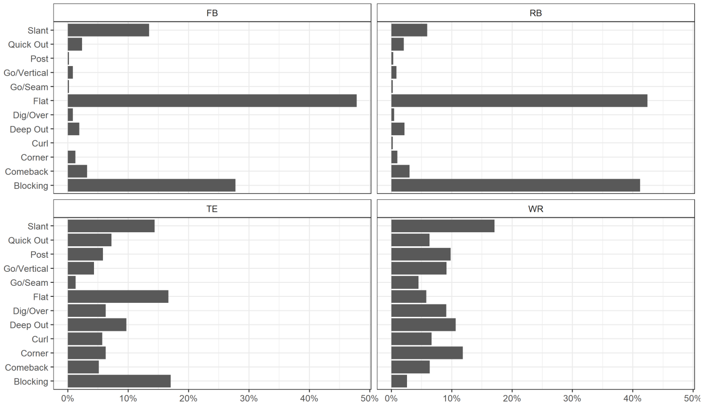


Fig. 6: Routes per Position

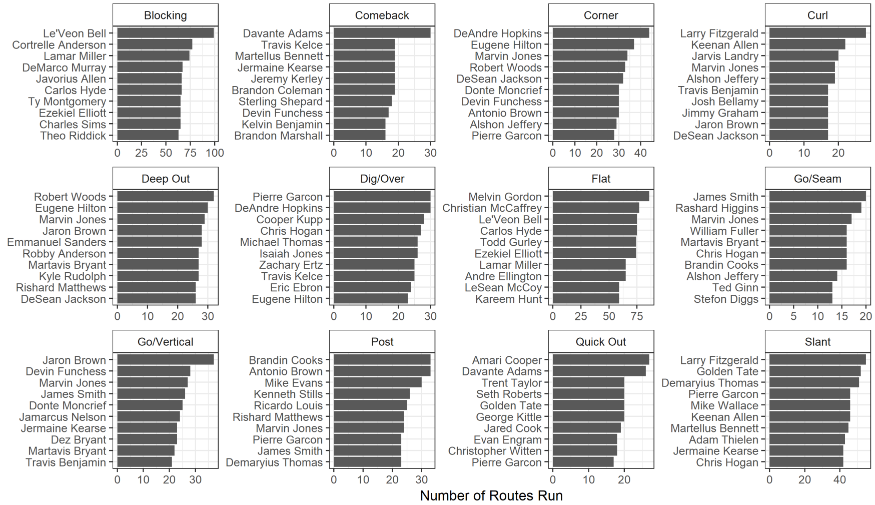


Fig. 7: Players who run the most of each route

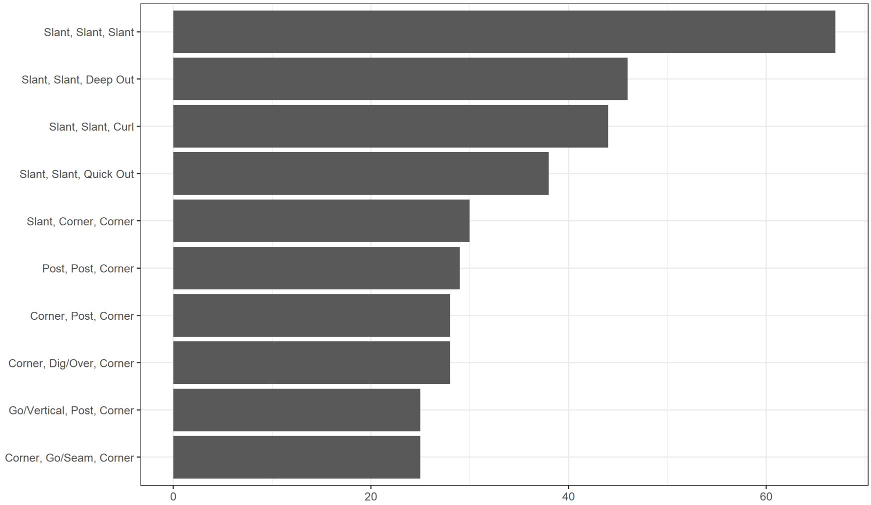


Fig. 8: 3 WR Designs

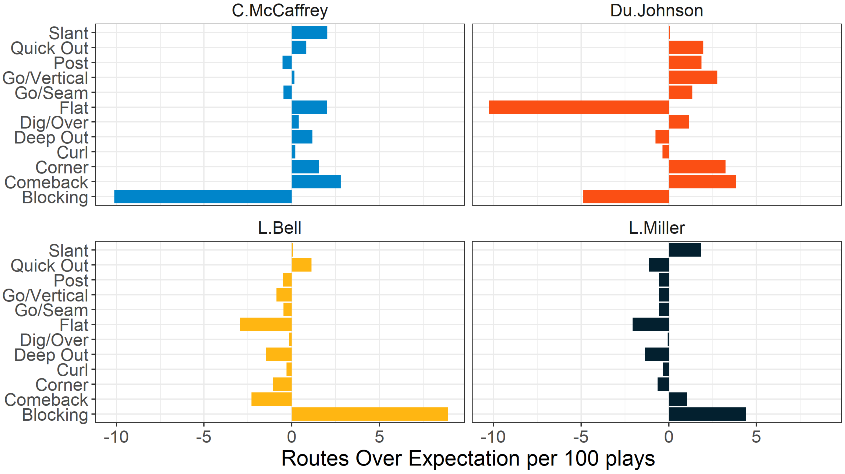


Fig. 9: RB Routes Run Over Expectation per 100 Plays

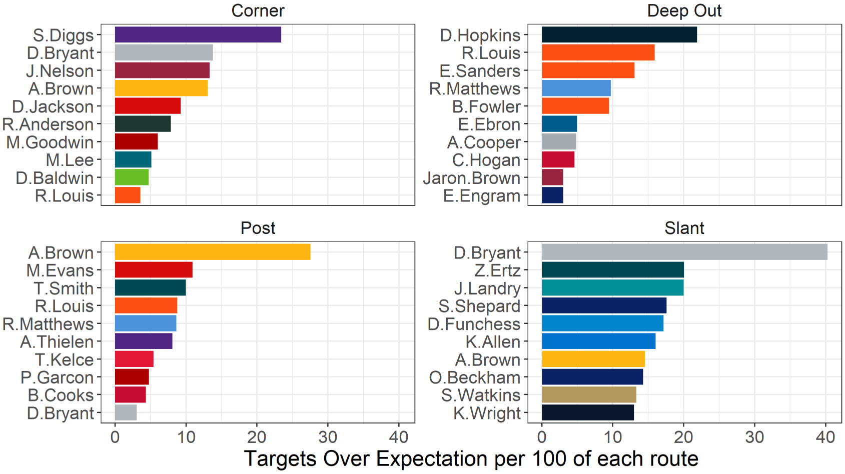


Fig. 10: Targets Over Expectation per 100 of each Route by Route

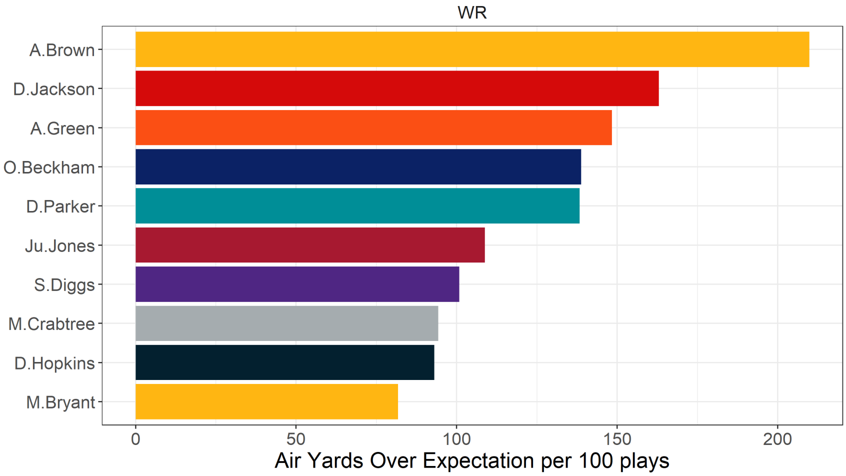


Fig. 11: Air Yards Over Expectation per 100 Routes