Analyzing Basketball Motion Data

Novel classification techniques

Edie Espejo, Chirag Kashyap, Patrick Vacek

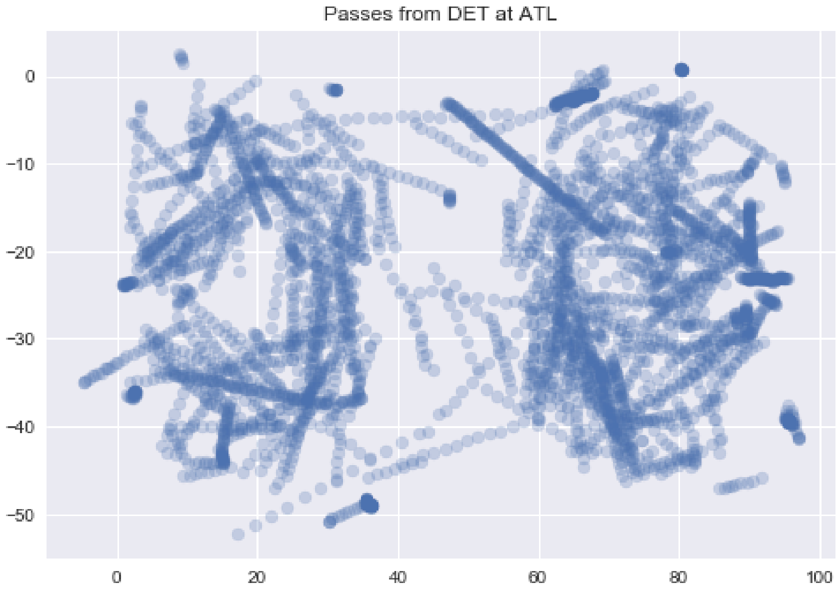
STA 160, June 12th 2017

Outline of Presentation

1. Passing

2. Shooting

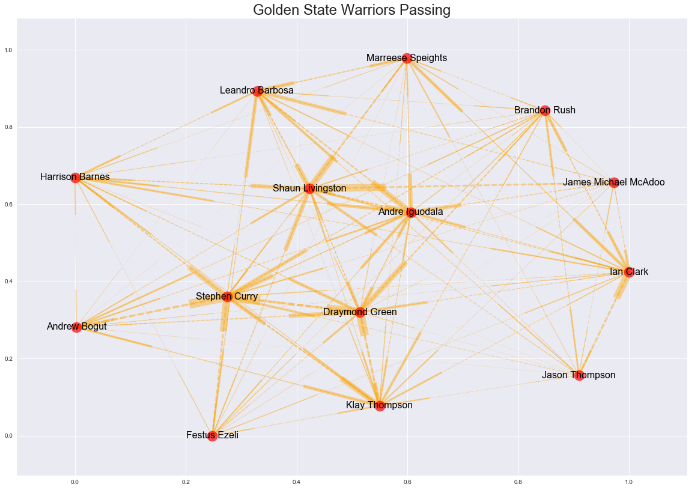
Passing: Motivating visual



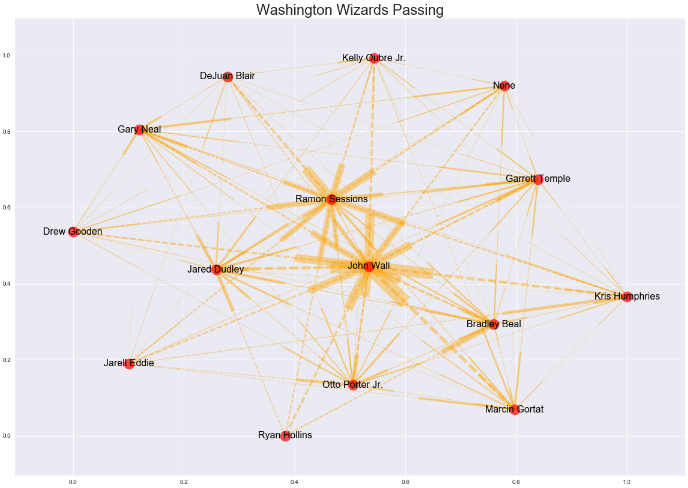
Passing: What we answered

- Part of our analysis included analyzing the nature of passes in basketball.
- Two questions we answered:
  - Whom passes to whom?
  - Can we determine which person the ball is being passed to?
- To answer the first question, we visualized the passes as **networks**.
- The second, we answered with **the Gradient Algorithm**.

Passing: Networks



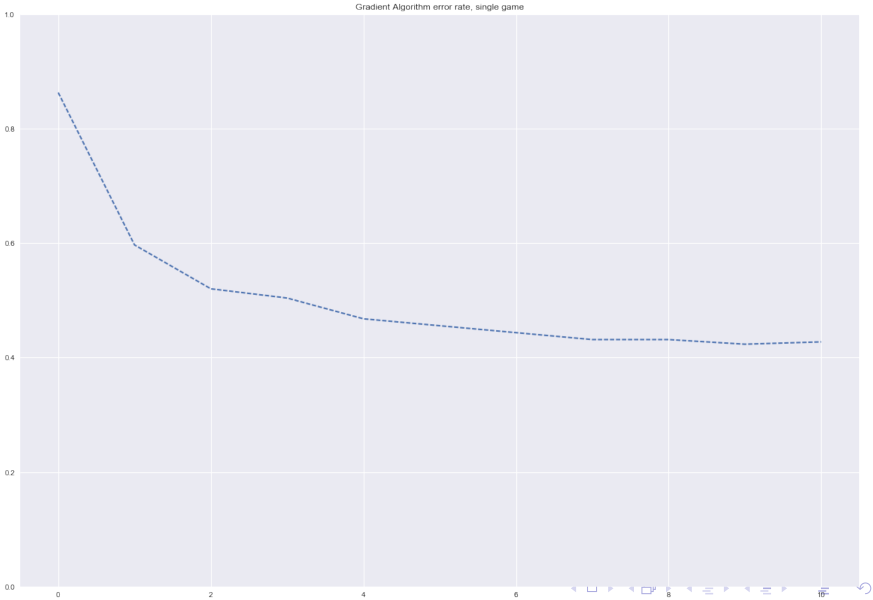
Passing: Networks



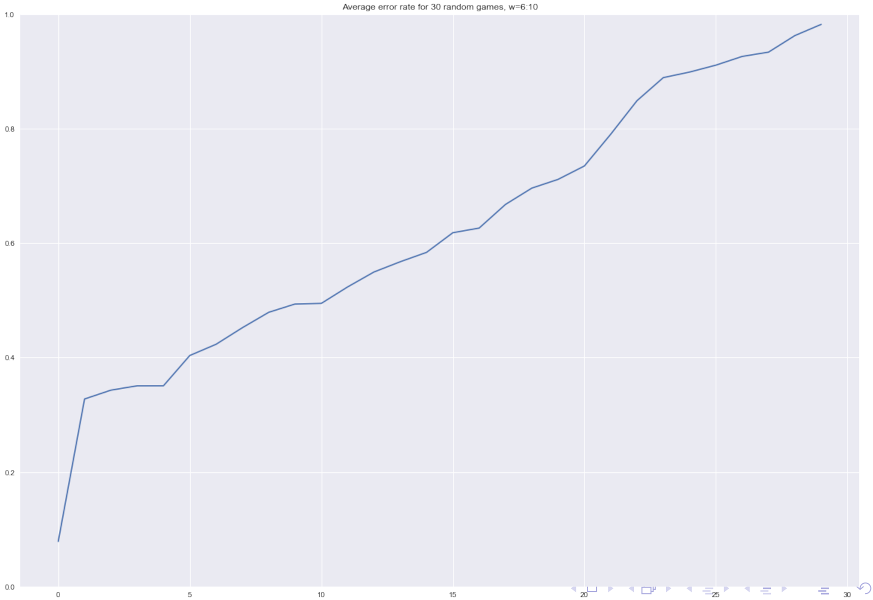
Passing: Gradient Algorithm

- We developed a novel method for classifying passes, the **Gradient Algorithm**
- As a pointer, a **gradient** can be described is a multi-variable version of the derivative, so we use it for vectors.
- Since we are short on time, we'll give a short heuristic description:
  - **Step 1:** At time t, examine the locations of the players and ball in x, y coordinates.
  - **Step 2:** We transform the coordinates by weighting the gradient by a scalar w and adding it to the x and y coordinates.
  - **Step 3:** The closest person to the ballw with these transformed coordinates is considered the *target* of the pass.
- We examined this heuristic upon intervals in which the ball is not close to any players, and we knew that it was being passed between players on the same team.
- Our goal was to see if the algorithm **could see the target that we see when we're watching basketball**.

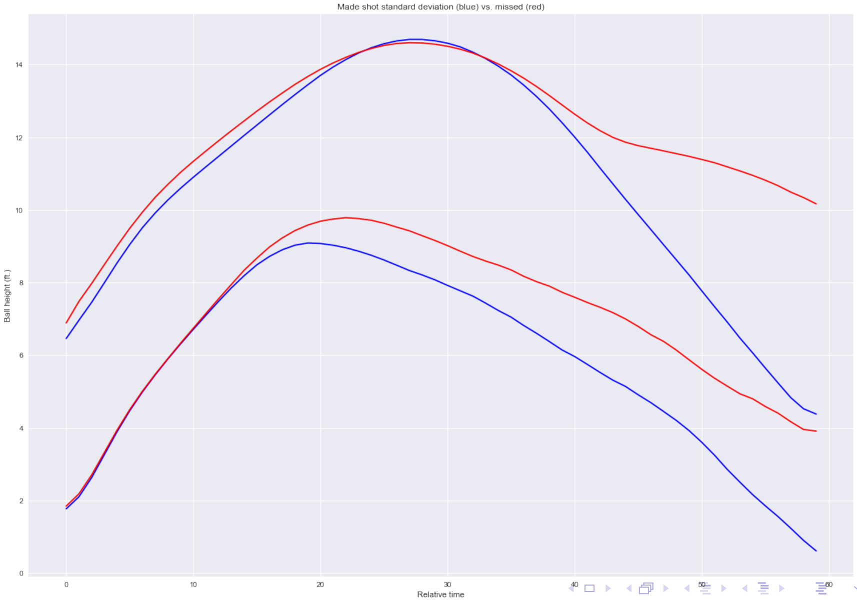
Passing: Gradient Algorithm, choosing a weight



Passing: Gradient Algorithm, error rates by game



Shooting: Motivating visual



Shooting: What we answered

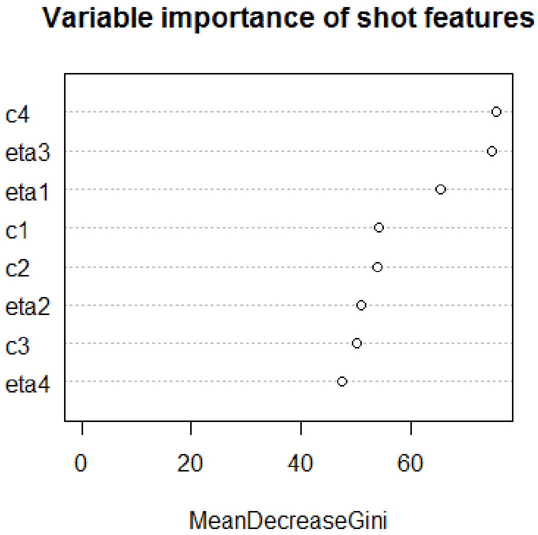
- Our main focus was classifying made and missed shots.
- We used different features for machine learning, this includes:
  - **Polynomials** We estimated the shot functions as cubic polynomials and attempted to see if the coefficients for made or missed shots differed significantly.
  - **Eigenfunctions** Using Functional Data Analysis, we used the eigenfunctions that explained shot variation to also classify.
  - **Time points** We made each shot have 50 time points, with each time point having a z coordinate for ball height.
- Questions we answered:
  - Which machine learning methods work best?
  - Do certain features work better at classification?
  - Which coefficients within features explain made and missed shots?
  - What time points are the most important at determining shot outcome?

Shooting: Classification results

model	data	subjects	train	test	accuracy	sensitivity	specificity
QDA	Polynomial Coefficients	All shots	1000	393	0.6972	0.7197	0.6822
QDA	Polynomial Coefficients	Jump shots	400	251	0.7161	0.7777	0.6927
QDA	Eigenfunctions	All shots	1000	393	0.6946	0.6815	0.7033
QDA	Eigenfunctions	Jump shots	400	251	0.7625	0.6315	0.8086
SVM	Polynomial Coefficients	All shots	1000	393	0.6488	0.5325	0.7366
SVM	Polynomial Coefficients	Jump shots	400	251	0.8675	0.6779	0.9314
SVM	Eigenfunctions	All shots	1000	393	0.6870	0.6012	0.7478
SVM	Eigenfunctions	Jump shots	400	251	0.8990	0.7692	0.9397
Random Forest	Time Points	All Shots	1000	393	0.7531	0.7405	0.7617
Random Forest	Poly+Eigen	All Shots	1000	393	0.6972	0.6708	0.7148

- For all shots, our best performing algorithm was using **RandomForest** with time points. It also appears that eigenfunctions outperformed using simple polynomial coefficients.

Shooting: Coefficient importance



Shooting: Time importance