One Pager (Nothing but Elastic Net)

Derek Owens-Oas, Megan Robertson, Fan Bu, Federico Ferrari

September 23, 2017

1 One Pager

1.1 Intro:

NBA commentators often point out when a team goes on a run. Having a well-defined, automated way of detecting when a run is occuring relieves commentators of the burden of keeping track manually and allows them to focus on other aspects of the game. One way to define a run is when a team scores on multiple consecutive possessions without the opponent scoring any points in between. This definition captures many runs, but it does not capture the scenario in which a team makes three-pointers on three consecutive possessions, and the other team scores a quiet lay-up in between. We propose a novel and flexible definition of runs which captures scenarios like this.

With runs well defined, we set out with a goal to learn which covariates, or features of the current gameplay, correlate strongly with runs. A few specific hypotheses motivate our analysis. We think there are scenarios in which a streak is more likely. A motivating halftime speech, a large skill differential between the offense's and defense's lineups, or an opponent with a propensity to turnover the ball may increase the likelihood of a streak. On the other hand, some features may decrease the likelihood of a run. Playing the second game of a back to back is one example.

1.2 Data:

To address this question, we begin by labeling the possession data from last season with our run detection algorithm. Now every possession is either part of a run or it is not. Next, we use the other data sets to create additional variables which are predictors of a run. The variables in our model are described below:

- Response: `RUN`
- Predictors: `TEAM`, `NUMBER OF SHOTS`, `NUMBER OF REBOUNDS`, `DRIBBLES`, `PASSES`, `TOUCHES`, `GAME CLOCK START`, `POSSESSION LENGTH`.

1.3 Model:

We will begin by training a model on a random sample of 90% the games and holding out 10% of the games as testing data. We then frame run prediction as a binary classification problem. Each possession is either a run or it is not. A common approach for binary classification is logistic regression. This statistical model is capable of learning coefficients, which are interpreted as the change in log odds of a run.

1.4 Results:

With a fitted logistic regression model, it is then possible to input a scenario and obtain the correponding prediction that a run is occurring. One option is to input a typical scenario and estimate the probability of a run occuring. Then, relative to this baseline, we can input our hypothesis: a team enters the game after halftime, there is a large skill differential between lineups, and the opponent is playing the second game of a back to back. We can then predict the probability of a streak occurring. We plan to use a shiny app to visually present probabilities of a run for various in game scenarios, allowing a user to compare and contrast them.