2020

How Machine Learning and Probability Concepts Can Improve NBA Player Evaluation

Harrison Miller

Claremont McKenna College

How Machine Learning and Probability Concepts Can Improve NBA Player Evaluation

Submitted to

Professor Mark Huber

by

Harrison D. Miller

for

Senior Thesis

Spring Semester 2020

June 15, 2020

HOW MACHINE LEARNING AND PROBABILITY CONCEPTS CAN IMPROVE NBA PLAYER EVALUATION

HARRISON D. MILLER

Abstract. In this paper I will be breaking down a scholarly article, written by Sameer K. Deshpande and Shane T. Jensen, that proposed a new method to evaluate NBA players. The NBA is the highest level professional basketball league in America and stands for the National Basketball Association. They proposed to build a model that would result in how NBA players impact their teams chances of winning a game, using machine learning and probability concepts. I preface that by diving into these concepts and their mathematical backgrounds. These concepts include building a linear model using ordinary least squares method, the bias variance trade off, regularization and three methods of regularization, Gibbs samplers, and kernel density estimation. Furthermore, I explain how each of these concepts affect the process of building their model. Lastly, I explain the effectiveness of their methodology, as well as its flaws and how I would improve it.

1 Introduction

The ultimate goal of an NBA franchise is to win the NBA Championship, being crowned the best team in the NBA. In order to achieve this goal, an NBA franchise must build the best possible roster. This an extremely difficult task, because of the complicated issue of player evaluation. It is a difficult task to decide whether raw statistics and advanced statistics paints a better than how a player appears to play while watching when evaluating a player's effectiveness. Furthermore, when analyzing a player from a statistical standpoint it is a challenge to build a model or a statistical metric that accurately expresses a player's effectiveness. This results in the necessity of an accurate method of player evaluation, when

analyzing a player from a statistical standpoint. Player evaluation in the NBA has been an ever evolving competition with NBA general managers always trying to find a better way to view performances of NBA players. While statistical player evaluation has become much improved, Deshpande and Jensen proposed a fatal flaw with player evaluation, from a statistical standpoint, context. With the goal of winning in mind, a player statistical output has significantly less meaning without the context of when it occurs. If a player were to score 10 points, in the 5 minutes of the game, while his team is losing or winning by 30, his production would hardly change the chances of a win or loss. On the other hand if the same time and scoring production occurred during a tie game, that would significantly alter the chances of a team winning. Because of this, Deshpande and Jensen proposed using the change in winning probability as its main predictor in evaluating a player.

In this paper, I will first be discussing the mathematical theories behind the methodology of the player analysis done by Deshpande and Jensen. I will then be discussing the actual methodology behind the attempt to improve how one views player performances, from a evaluation standpoint. In section 2, I briefly explain what a general linear regression is, and its properties. The ordinary least squares method of building a linear model will be briefly discussed as well. In section 3, the concepts of bias and variance, resulting from training a model, will be explained. On top of that, the bias variance trade off will be explained, one of main reasons behind the creation of regularization. In section 4, I introduce the concept and explain the motivation behind regularization. I also introduce how to apply the idea of regularization to a linear model. More specifically, in section 4.1, I explain the method of a ridge regression, its positives and shortcomings. In section 4.2, I explained a regularization method, lasso regression, that was made with the intent to improve on the ridge regression. I also discuss its origins, flaws and when it excels. In section 4.3, I further explore further

improvements on the regularization techniques discussed in section 4.1 and 4.2. This improved method is called the elastic net, and it fixes the inherent flaws of lasso and ridge regressions. In section 5, I introduce the concept and theory of a Gibbs sampler. A Gibbs sampler is used to find the posterior distribution from the prior distribution. In section 6, I explain the concept and use of kernel density estimation. The kernel density estimation is a method that is used to create an estimated probability function for a random variable from outputs from that random variable. In section 7, I discuss previous attempts to evaluate NBA players, from a statistical standpoint, and explain how they fell short. In section 8, I introduce the methodology, created by Deshpande and Jensen, used to evaluate NBA players' effect on winning probability. In section 8.1, I dive into the creation and components of the model built by Deshpande and Jensen. In section 8.2, I explain the process of creating a model for probability of an NBA team winning, conditioned on current point differential and amount of the game that has been played. In section 8.3, I explain the process of how Deshpande and Jensen built their model. Specifically, how they applied regularization methods in order to create an accurate model. In section 9, I discuss how, after creating the Bayesian linear model, Deshpande and Jensen used the model and a Gibbs sampler, built for a lasso regression, to generate independent samples for each player. Furthermore, I discuss their method of estimating a posterior distribution for each player, describing the effect each NBA player has on winning probability. Lastly, I explain Deshpande's and Jensen's method to compare players.

2 Building a Linear Model

Creating a linear model is one the simplest and the one of the most widely used statistical technique for predictive and descriptive modeling. The goal of it

is to acquire an equation in the form of:

$$[15]\hat{y} = w_0x_0 + w_1x_1 + \cdots + w_nx_n + b \tag{2.1}$$

This is the general form of a linear model based on n number of variables, or also known as predictors or independent variables. In equation 2.1 the $\hat{y}$ represents our dependent variable, the data we are trying to predict using out independent variables. Next, b represents our y-intercept. Lastly, the w_n represents the coefficients of our model, the variables we are trying to manipulate to create a predictive model. The overall goal of using a linear model is to minimize the equation by finding w_j's:

$$[15]\sum_{i=1}^{m}(y_i - \hat{y})^2 = \sum_{i=1}^{m}(y_i - (\sum_{j=1}^{n} w_jx_{ij}))^2 \tag{2.2}$$

In this equation, m represents the amount of data points we have, and n represents the number of features the linear model has. When using the equation 2.2 in order to fit the linear model in equation 2.1, it is called using the ordinary least squares method for fitting a model. This equation (2.2) has many names: sum of squares of the residuals, the ordinary least squares, Root Mean Square Error, and lastly the cost function. I will be referring it to the cost function for ordinary least squares for the remainder of this paper.

3 Bias Variance Trade Off

While creating a linear model from data is a great way to represent the data and as a prediction tool, it is not perfect. This method, at its core, has flaws that can lead to overfitting and underfitting. Overfitting occurs when a model, generated from the given data, models the data too well without taking into account the possibility of more data. This is a bad situation because it results in a model that doesn't correctly model the data. On the other hand, underfitting

occurs when a model doesn't model the data its built on well. Again, this is a poor situation, because the model is too general and is it not a great predictor. In order to find out if overfitting or underfitting is occurring, it is common practice in machine learning to implement a concept called supervised learning. The first step of this process is to partition the data set into two parts, a training set and test set. The size of the partition is up to creator, but it is common practice to use cross validation to find the correct partition ratio. One then creates a model using the training set to train the model. Next, the model is used to predict the data in the test set. Those values are compared and are used to find the bias and variance for a second time. This gives the creator a better scope when trying to analyze the accuracy of the model, since the bias and variance found from the data used to train the model can be misleading because it doesn't fully describe the model's ability to correctly predict the data.

Overfitting and underfitting are caused by the values of bias and variance of a model. The bias of the model is found from:

$$[3]Bias = |\mathbb{E}[D]y(x;D) - f(x)| \tag{3.1}$$

The variance of the model is found from:

$$[3]Variance = \mathbb{E}[D]y(x;D)^2 - (\mathbb{E}[D]y(x;D))^2. \tag{3.2}$$

Bias error is the error of the model from assumptions made while it was trying to fit the data. Having a high bias can cause a model to miss relevant relationships between the dependent and independent variables. Having a high bias is considered underfitting. Next, variance is an error caused by high sensitivity to small fluctuations in the data set. Having a high variance causes the model to put more weight into the noise of the data causing an inaccurate model. When this occurs, it is considered overfitting. For a better visualization, imagine someone playing

darts, with the intended target of hitting the bulls-eye every throw. When we have low bias and low variance, his darts will be a tight cluster centered at the bulls-eye. When bias is low and variance is high, we get a dart board with a loose cluster of darts, but they are still visibly surrounding the bulls-eye. Next, when bias is high and variance is low, we get a tight cluster but it is not centered at the bulls-eye. Lastly, if bias and variance are both high, we get a dart board with a loose cluster of dart not centered at bulls-eye.

The end goal of the model is to minimize both bias and variance, but this is an extremely difficult task. This difficulty is the direct result of the bias variance tradeoff. The bias variance trade off is a property of predictive models where there is a lower bias in parameter estimation there is a higher variance of the parameter estimates across samples. This works the other way as well. While this concept is not universal, it is still an extremely common issue that is run into during the creation of a predictive model. The ultimate goal of this is to create a model with a low bias, a model that correctly models the relevant relationships between the predictions and the actual data points, without increasing the variance too much, a model that doesn't overcompensate for noise in the data set. On other words, one wants to ideally choose a model that both accurately captures the regularities in the training set and generalizes well to unseen or unknown data. One method to accomplish this goal in linear models is to apply a regularization term to the model creation. This is explained in the next section.

4 Regularization

In this section I will be discussing the concept of regularization, also known as penalization techniques, its uses and introducing three different types of it. Regularization is the process in which one "shrink" or constrains the coefficients predictors of a model by penalizing higher valued predictors. The goal of this process is to attempt to lower the variance of our model without raising our bias

a significant amount, essentially attempting to fix overfitting. The bias and the variance are two types of errors we have when building a model. First we have our predictive model.

$$[3]t \sim f(x) + \epsilon \tag{4.1}$$

We denote t as our predicted data, $f(x)$ as our unknown function, that is gathered from a sample denoted as $D = \{(t_1, t_1), \ldots, (y_n, x_n)\}$, and the error is denoted as ϵ. Furthermore, our ultimate goal is to estimate our function f from D, using a parametric family of normally linear functions represented as such:

$$[3]y(x; w) = x^T w \tag{4.2}$$

where w is a vector of parameters, our coefficients in our future model. Our next goal is to measure the quality of our estimator, y. We accomplish this goal by calculating our expected squared loss which can be written as:

$$[3]\,\mathbb{E}[L] = \int \{y(x) - f(x)\}^2 p(x)dx + \int \{f(x) - t\}^2 p(x,t)dxdt \tag{4.3}$$

In equation 4.3, the second term is the noise of the model, the variation in the data that can't be explained by the model. Hence, our noise is the minimum of the expected loss equation. Next, our goal is to choose a $y(x)$, such that the first term is minimized. Since, the first term is non-negative, the smallest achievable minimum is zero. Furthermore, the expected loss equation(4.3) can be rewritten as the sum of the bias(3.1) squared, the variance(3.2) and the noise of the model. For the proof see [3]. As mentioned above, the overall goal of model building is to minimize this expected loss equation. This becomes a difficult task since bias and variance are inversely related. This leads us to the use of regularization

As mentioned earlier, regularization is the technique in which one "shrinks" the predictors. In turn, the effect of the predictors are reduced with the goal of

reducing error in the model. On top of that, with certain techniques to be discussed later, the penalty term implements variable selection as well as shrinking the predictors. In this section, I will be discussing three different types of regularization techniques that were discussed and used in *Estimating an NBA player's impact on his team's chances of winning*[5]. These three methods are commonly used and each has its own advantages and disadvantages that can be exploited to create the best possible model. The general concept of the techniques that will be covered are an addition to our cost function. The original cost function can be seen in equation 2.2. We then add a penalty term to our cost function which gives us the general form:

$$[15]\frac{1}{2}\sum_{n=1}^{N}\{t_n - w^\top\phi(x_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q \tag{4.4}$$

We also constrain the regularization term such that:

$$[15]\sum_{j=1}^{M}|w_j|^q \leq \eta \quad for some \;\; \eta > 0 \tag{4.5}$$

In equation 4.4, the new cost function with a regularization term, λ is known as the tuning parameter. In this new cost function, we introduce a regularization term. The regularization term shrinks the values of $w^\top$, the coefficients of the predictors, by increasing the value of the cost function, which in turn decreases the value of the predictors. The scale of the effect from the regularization term, on the predictor coefficients, has a direct relationship with the regularization. Changing the tuning parameter is how we change the amount of regularization that is occurring. The tuning parameter directly affects how much shrinkage occurs from the regularization term. This means that finding the correct value for the tuning parameter is extremely important in order to optimize the positive effects resulting from using regularization. As for the behavior of the effect of the

cost function from the tuning parameter, as $\lambda \to 0^+$ we get:

$$\lim_{\lambda \to 0} \mathop{\mathbb{E}}_{D} + \frac{\lambda}{2} (\sum_{j=1}^{M} |w_j|) = \mathop{\mathbb{E}}_{D}$$

which is the exact same as the cost function for ordinary least squares. On the other hand, as $\lambda \to \infty$ we get:

$$\lim_{\lambda \to \infty} \mathop{\mathbb{E}}_{D} + \frac{\lambda}{2} (\sum_{j=1}^{M} |w_j|) \implies w_j = 0 \quad \forall j$$

This means that as λ approaches infinity, the effect of each of the predictors on the prediction shrinks, eventually having no effect. In the succeeding subsections, three different versions of the general regularization cost function, the ridge, lasso, and elastic net regressions, will be discussed and explained.

4.1 Penalty with a Normal Prior

The first technique I will be discussing is the Ridge method, also known as using an L_2 regularization or placing a normal prior on our distribution or model. This regularization changes our model by changing our cost function to:

$$[12]\hat{E}(w) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2} \sum_{j=1}^{M} |w_j|^2 \tag{4.6}$$

In equation 4.6, $\lambda \sum_{j=1}^{n} w_j^2$ is considered our penalty term, previous discussed in the abstract, where λ, lambda, is our penalty term.

The Ridge regression has one major shortcoming that the lasso and elastic net regressions have, variable selection. This can be problematic in certain cases where there are variables that are essentially irrelevant. When those irrelevant variables are included in the model, it can lead to increased sensitivity to noise. If the sensitivity to noise is severe enough, it will lead to the model being overfit, giving you an inaccurate model. On the other hand, the Ridge regression still has a lot of advantages over the ordinary least squares method. First, it shrinks

irrelevant, or less relevant predictors, towards zero, decreasing their effect on the prediction and increasing the accuracy of the model. The Ridge regression also excels, comparatively to the ordinary least squares method in situations where there are more predictors than data points. Lastly, it performs much better when there are multicollinearity appears in the model. Multicollinearity can cause extreme sensitivity, in the model, to minor changes in the data and leads to overfitting of the model. While the Ridge regression has its limitations and has been improved, mentioned in section 4.2 and 4.3, it is still an improved method from the ordinary least squares method for model creation.

4.2 Penalty with a Laplacian Priors

One other popular technique is called the lasso, which is an acronym for "least absolute shrinkage and selection operator"[19]. This concept was created in order to combine the advantages from using a ridge regression and using a subset selection method. This method originates from a proposal from Brieman[4]. His non-negative garotte minimizes:

$$[4]\sum_{k}\left(y_n - \sum_{k} c_k\hat{\beta}_k x_{kn}\right)^2 \tag{4.7}$$

under the constraints of

$$[4]c_k \leq 0 \qquad \sum_{k} c_k \leq s \tag{4.8}$$

Now let $\tilde{\beta}_k(s) = c_k\hat{\beta}_k$ be the new predictor coefficients. Furthermore, as s decreases, the garrote is drawn tighter. This results in more c_k's becoming 0 and shrinkage of the nonzero $\tilde{\beta}_k(s)$'s. This procedure is known as the non-negative garrote. The consequences of this method are feature selection and coefficient shrinkage.

The negative of this method is that its solution is dependent on both the sign and the magnitude of the Ordinary Least Squares(OLS) estimates. For example, if our model results in an over-fit or high co-linearity, resulting in poor estimates from the OLS, the garotte method may suffer at fixing this problem. On the other hand, avoids this problem by not explicitly using the OLS estimates. Instead it places a prior on the initial data, also called the Laplacian prior, in order to find the new distribution. The general form of this prior is shown by Frank and Friedman[7]. They proposed using a bound on the parameters in the general form of an L^q-norm, such that $q0$. For the lasso, we explore the case in which $q = 1$.

We now come into the definition of the Lasso and some of its properties.

$$[19](\hat{\alpha}, \hat{\beta}) = \arg\min\{\sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2\} \quad subject\ to\ \sum_j |\beta_j| \leq t \tag{4.9}$$

In this, t $\geq$ 0 is considered our tuning parameter. Furthermore, a term is added to the cost function, denoted as $\mathbb{E}_D$, from the ordinary least squares' cost function to give us:

$$[19]\ \underset{D}{\mathbb{E}} + \frac{\lambda}{2}(\sum_{j=1}^{M}|w_j|), \quad such\ that \quad \sum_{j=1}^{M}|w_j| \leq \eta \tag{4.10}$$

One of the most important parts of the lasso technique is the tuning parameter value. Choosing the right value for λ is how the lasso regression is able to perform feature selection, one of the most important results of applying the Laplacian prior. This makes our parameter selection an important process. There are two popular options, cross-validation and generalized cross-validation. Suppose that

$$t = \eta(X) + \epsilon$$

where $\mathbb{E}(\epsilon) = 0$ and $\text{Var}(\epsilon) = \sigma^2$. The mean-squared error of an estimate $\hat{\eta}(X)$ is defined by:

$$[21]MSE = \mathbb{E}\{\hat{\eta}(X) - \eta(X)\}^2 \tag{4.11}$$

Furthermore, we find our prediction error of $\hat{\eta}(X)$ by

$$PE = \mathbb{E}\{Y - \hat{\eta}(X)\}^2 = MSE + \sigma^2$$

We then estimate the prediction error for the lasso procedure by five-fold cross-validation as described in chapter 17 of Efron and Tibshirani[6].

While the lasso regression method has a lot of positives and advantages to its counterpart, the ridge regression, it also has its limitations. There are three main limitations for the lasso regression. For these limitations, let n denote the number of data points there are, and p denote the number of predictors in the model.

(1) [21] In the $p > n$ case, the lasso selects at most n variables before it saturates, because of the nature of the convex optimization problem. This seems to be a limiting feature for a variable selection method. Moreover, the lasso is not well defined unless the bound on the L1-norm of the coefficients is smaller than a certain value.
(2) If there is a group of variables among which the pairwise correlations are very high, then the lasso tends to select only one variable from the group and does not care which one is selected. See Section 2.3.
(3) For usual $n > p$ situations, if there are high correlations between predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression[19].

As a whole, the lasso regression was created to solve the shortcomings of the Ridge regression. In other words, it solves the biggest issue with the Ridge regression by changing the penalty term to allow feature selection to occur. Since it linearly shrinks each predictor, instead of asymptotically like the Ridge regression, certain predictors are set to 0, in most situations. Another way to think of this difference is that the Ridge regression scales the coefficients by a constant factor while the lasso regression translates the coefficients by a constant factor, with a floor of 0. While the lasso regression does fix the issues that arise from using a ridge regression, it has its own drawbacks(4.2). In the next section, I will be discussing an improved version of the ridge and lasso regression, the elastic net.

4.3 Penalty with a Laplace-Gaussian Priors

A Laplace-Gaussian prior, also known as an elastic net regression is another form of regularization and variable selection technique that is commonly used when trying to fit a model to data. As mentioned in section 4.1 the ridge regression, placing a Normal prior, has its advantages, but at the same time it has glaring limitations. On the other hand, the lasso regression, section 4.2, is a more proficient method of regression where the ridge regression does not excel, but again there is still no significant overall improvement with every type of linear model. As mentioned in the previous section (4.2), we have concrete limitations of the lasso regression. Fixing these limitations, specifically the first(1) and second(2) reference in 4.2, was the intention of Zou and Hastie[21] when they came up with the elastic net method. They wanted to create a new method that is proficient as the lasso, in situations where the lasso is at its best, while acting the same as the best method when situations in 1, and 2 occur. Lastly, they wanted to improve the results of the lasso when the 3 case occurs. These goals led them to come up with the elastic net method of regularization.

First comes the naive elastic net representation[21]. Let n denote the number of data points, or observations in the data set, and let p denote the number of predictors, or covariates, in the model. Let $y = (y_1,\ y_2, \ldots,\ y_n)^\top$ is a vector of our observed dependent variables. Let $X = (x_1|\ x_2| \ldots |\ x_p)$ be the model matrix, where $x_j = (x_{1j},\ x_{2j}, \ldots,\ x_{nj})^\top$, such that $j = 1,\ 2, \ldots,\ p$ are the predictors. Once a location and scale transformation are completed they assume the dependent variables are centered and predictors are standardized such that:

$$[21]\sum_{i=1}^{n} y_i = 0, \quad \sum_{i=1}^{n} x_{ij} = 0, \quad \text{and} \quad \sum_{i=1}^{n} x_{ij}^2 = 1 \quad \text{for } j = 1,\ 2, \ldots,\ p \tag{4.12}$$

Let $\lambda_1,\ \lambda_2 > 0$ and both are fixed, the naive elastic net criterion is defined as

$$[21]L(\lambda_1,\ \lambda_2,\ \beta) = |y - X\beta|^2 + \lambda_2|\beta|^2 + \lambda_1|\beta|_1 \tag{4.13}$$

such that

$$[21]|\beta|^2 = \sum_{j=1}^{p} \beta_j^2 \quad \text{and} \quad |\beta|_1 = \sum_{j=1}^{p} |\beta_j| \tag{4.14}$$

Furthermore, the naive elastic net estimator, denoted as $\hat{\beta}$, is the minimizer of equation 4.13:

$$[21]\hat{\beta} = \arg\min_{\beta}\{L(\lambda_1,\ \lambda_2,\ \beta\} \tag{4.15}$$

As you can see from 4.13 and 4.14, this definition combines the penalty terms from both the ridge regression(4.6) and the lasso regression(4.10) and adds them to the Ordinary Least Squares equation(2.2). Furthermore, they each penalty term has their own tuning parameter. Next, let $\alpha = \frac{\lambda_2}{\lambda_1+\lambda_2} \implies \alpha \in [0, 1]$. Then when solving for $\hat{\beta}$ in equations 4.13 is essentially the process of optimizing:

$$[21]\hat{\beta} = \arg\min_{\beta} |y - X\beta|^2, \quad \text{subject to } (1-\alpha)|\beta|_1 + \alpha|\beta|^2 \leq t \quad \textit{for some } t \tag{4.16}$$

The function $(1-\alpha)|\beta|_1+\alpha|\beta|^2$ is called the elastic net penalty. Furthermore, the special case where $\alpha = 1$, the elastic net is the exact same as the ridge regression. On the other hand, when $\alpha = 0$ the elastic net is the exact same as the lasso regression. Otherwise, the elastic net is a combination of both styles of regression.

Maybe add more about actual proof it solves those issues

Lemma 4.1. [21] *Given data set* (y, X) *and* (λ_1, λ_2)*, define an artificial data set* $(y^{\cdot X)}$ *by:*

$$X^*_{(n+p)\times p} = (1+\lambda_2)^{-1/2}\begin{pmatrix} X \\ \sqrt{\lambda_2}I \end{pmatrix}, \quad y^*_{n+p} = \begin{pmatrix} y \\ 0 \end{pmatrix}$$

Let $\gamma = \frac{\lambda_1}{\sqrt{1+\lambda_2}}$ *and* $\beta^* = \sqrt{1+\lambda_2}\beta$ *This allows us to change the elastic net criterion to be:*

$$L(\gamma, \beta) = L(\gamma, \beta^*) = |y^* - X^*\beta^*|^2 + \gamma|\beta^*|_1$$

Let

$$\hat{\beta}^* = \arg\min_{\beta^*} L\{(\gamma, \beta^*)\}$$

; Then

$$\hat{\beta} = \frac{\hat{\beta}^*}{\sqrt{1+\lambda_2}}$$

Because the sample size in the augmented problem is $n+p$ and X^* has rank p, the naive elastic net can potentially select all p predictors, removing the limitation (1) of the lasso regression.

In order to overcome the challenge of accurately modeling when there is a larger number of predictors than data points, specifically when $p >> n$, the process of grouping variables becomes almost essential. Grouping variables, also known as grouped variable selection, is the process that when you discover multiple co-linear predictors the value of their coefficients become regularized such that they become equal. This concept is one of the limitations of the lasso regress, because the lasso regressions tends to select just one of the co-linear variables and doesn't care which one is the one left. While in some situations this might

be very beneficial, in the case where $p > n$ this is not the best way to approach the problem. Before introducing the lemma for the variable grouping we must first show the generic penalization method:

$$[21]\hat{\beta} = \arg\min_{\beta} |y - X\beta|^2 + \lambda J(\beta) \tag{4.17}$$

such that $J(\cdot)$ is positive valued for $\beta \neq 0$

Lemma 4.2. [21] *Assume that $x_i = x_j$ and $i, j \in \{1, 2, \ldots, p\}$*

(1) *If $J(\cdot)$ is strictly convex, then $\hat{\beta}_i = \hat{\beta}_j$, $\forall \lambda > 0$.*

(2) *If $J(\beta) = |\beta|_1$, then $\hat{\beta}_i \hat{\beta}_j 0$ and β^* is another minimizer of equation 4.17, such that:*

$$\hat{\beta}_k^* = \begin{cases} \hat{\beta}_k & \text{if } k \neq i \text{and } k \neq j \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (s) & \text{if } k = i \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (1 - s) & \text{if } k = j \end{cases}$$

for any $s \in [0, 1]$

From lemma 4.2, there is a clear difference between the strictly convex penalty functions and the lasso penalty. The beauty of the elastic net method is that in the case where $\lambda_2 > 0$, the elastic penalty is strictly convex. Furthermore, when the penalty is strictly convex, the variable grouping occurs when identical predictors appear. As shown above, it can be easily proven and shown that the elastic net method of regularization overcomes the two biggest limitations of the lasso regression, while performing as good or better when the lasso regression is at its best.

5 Gibbs Sampler

In Bayesian models, such as the one that will be discussed 8, as the model becomes increasingly complicated and a better predictor a more efficient inference

method is required. That leads us to Bayesian Inference. The goal of using Bayesian Inference is to maintain a full posterior probability distribution over a set of random variables. As the model becomes more complicated, using and maintaining the posterior distribution becomes increasingly more difficult until it becomes intractable. This is the result of the necessity of constantly calculating rigorous integrals. A way to work around this is the use of Monte Carlo Markov Chain, also known as MCMC, sampling techniques for simulation, with the goal of finding the posterior distribution. The logic of MCMC sampling is that if we have N simulated samples from the distribution, it is possible to compute any statistic of a posterior distribution:

$$[20]\,\mathbb{E}[f(s)]_\rho \approx \frac{1}{N}\sum_{i=1}^{N} f(s^{(i)}) \tag{5.1}$$

Such that ρ is the posterior distribution, $f(s)$ is the desired expectation, and $f(s^{(i)})$ is the i^{th} simulated sample from ρ. An example of this is that we can estimate the mean by taking $\mathbb{E}[x]_\rho = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$

Now were faced with actually obtaining the samples from the posterior distribution. This is where the Gibbs sampling technique comes into play. The intuition behind the Gibbs sampling is to obtain samples from the posterior distribution by sweeping through each variable with the goal to sample from the distribution conditioned on the chosen variable, with the remaining variables fixed to their current values. For a simple example let X_1, X_2, X_3, X_4 be four random variables. The first step is setting these four random variables to their initial values, which are normal sampled from the prior distribution: $x_1^0, x_2^0, x_3^0, x_4^0$. Then for the i^{th} iteration of this loop, we sample $x_1^{(i)} \sim p(X_1 = x_1 | X_2 = x_2^{(i-1)}, X_3 = x_3^{(i-1)}, X_4 = x_4^{(i-1)}$. We then sample $x_2^{(i)} \sim p(X_2 = x_2 | X_1 = x_1^{(i-1)}, X_3 = x_3^{(i-1)}, X_4 = x_4^{(i-1)}$, then sample $x_3^{(i)} \sim p(X_3 = x_3 | X_1 = x_1^{(i-1)}, X_2 = x_2^{(i-1)}, X_4 = x_4^{(i-1)}$, and lastly sample $x_4^{(i)} \sim p(X_4 = x_4 | X_1 = x_1^{(i-1)}, X_2 = x_2^{(i-1)}, X_3 = x_3^{(i-1)}$. This process is

repeated until the sample values have the same distribution as if they were from the true posterior joint distribution.

6 Kernel Density Estimation

Density Estimation is the process in which given an unknown probability density and outputs from a random variable, is the construction of an estimate probability density function for the random variable. One such type of density estimation is kernel density estimation, which is what I will be discussing in this section.

Let $X_1, X_2, \ldots, X_n$ denote a sample of size n from a random variable with density f[17]. The kernel density estimate of f at point x is given by:

$$[17]\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{6.1}$$

The kernel is denoted as K, and satisfies $\int K(x)\ dx = 1$. The smoothing parameter is denoted as h, also known as the bandwidth. Furthermore, generally a kernel is chosen to be a unimodal probability density that is symmetric about zero. This implies the satisfaction of three conditions [17]:

(1) $\int K(y)\ dy = 1$

(2) $\int yK(y)\ dy = 0$

(3) $\int y^2K(y)\ dy = \mu_2(K) > 0$

While there are many different types of kernel density estimators(uniform, triangular, biweight, triweight, Epanechnikov, Gaussian, and more), only the Gaussian kernel will be explained in this paper. For the Gaussian kernel we get:

$$[17]K(y) = \frac{1}{\sqrt{2\pi}} \exp -\frac{y^2}{2} \tag{6.2}$$

Based on the Gaussian kernel equation 6.2 and the kernel density estimation equation 6.1, it can be seen that what is occurring is finding a new density from

applying a normal distribution around each of the data point with the bandwidth acting as a tuning parameter. After actually completing the Gaussian kernel density estimation, we get:

$$[17]Bias[\hat{f}_h(x)] = \frac{h^2}{2}\mu_2(K)f''(x) + o(h^2) \tag{6.3}$$

and

$$[17]Var[\hat{f}_h(x)] = \frac{1}{nh}R(K)f(x) + o\left(\frac{1}{nh}\right) \tag{6.4}$$

such that $R(K) = \int K^2(y)\ dy$.

The next crucial step in the process of a kernel density estimator is bandwidth selection. Bandwidth is the tuning parameter for this process. When the bandwidth is too small it represents the scattered data too much, creating a distribution that is not very smooth and can represent the noise of the random variables too much. On the other hand, if the bandwidth is too large, the distribution becomes too flattened and smooth, such that it does not represent the random variables enough. This makes bandwidth selection an important piece of the process. One of the most commonly used method for this are cross validation algorithms, returning a favorable bandwidth.

7 Similar Previous Works

Before getting into how the model was built and how the results were interpreted, I will be discussing previous works that attempt to create an improved method of comparing NBA player performances. One of the first ways NBA player evaluators compared players was solely their scoring output combined with their true shooting percentage, which is a way to comprehensively measure how efficient a player scores. It is calculated by:

$$[2]TS\% = \frac{PTS}{(.44 \times FTA + FGA) \times 2} \tag{7.1}$$

Points scored is denoted by PTS. FTA denotes free throw attempts, and FGA denotes field-goal attempts, which are attempted shots during the live game. This equation essential results in points per possession divided by two, so that it looks as if it is a field goal percentage. Later, players were judged on their plus-minus. Plus-minus is the scoring differential that occurs while the player is on the court. The problem with that method is it doesn't adjust for the player's teammates effect on the game, resulting in a lot of misleading data. In order to fix this problem, in 2004, Rosenbaum [16] came up with idea of adjusting a player's affect based on who he was sharing the court with. Furthermore, in 2008, Ilardi and Barzilai [10] furthered this concept of adjusting for who was on the court with the player being analyzed. These methods were flawed, since collinearity created volatile results, which led to incorrect representations of player. In 2014, Ilardi [9] furthered his study to create real plus-minus. While all of these methods are based on team performance and not the box score, a player's stats during a game, one popular method of comparing player is calculated from the box score and pace of play, Player Efficiency Rating, PER[8]. This metric is calculated from the sum of differently weighted raw stats and then adjusting that value in order to get a per-minute rating of player. This method is also flawed, since it doesn't adjust for teammates, opponents, context of the stats, and playing style.

8 Building the Model

Deshpande's and Jensen's goal of their article was to build a new statistic that improves on previous attempts to represent an NBA player's effect on their team's chances of winning a game. The first step in this process is isolating the time a player is actually on the court. This is achieved by breaking up games into different shifts. A shift is the time period between two moments of substitutions, when at least one of the ten players on the court changes, in turn changing the effect of the other players to the player we are analyzing. In previous methods

of representing an NBA player's effect on a team's chance of winning, the goal was to measure solely the effect of point differential for when a player is playing, controlling for the for the player's teammates and opponents effectiveness, and adjusting the value to be an average it over 100 possessions. Using 100 possessions to normalize teams' offense and defense effectiveness is a common practice used in basketball analysis, because it is around the average amount of possessions per game. Because we "adjust" a players plus minus to 100 possessions is why that method is called Adjusted Plus Minus. This method had an enormous flaw, it was not isolating a player's effect enough, since their Adjusted Plus Minus would be skewed if a player shared the majority of his time on the court with a top tier player, driving their Adjusted Plus Minus up, or if they shared the court with a very bad player, driving their Adjusted Plus Minus down. This is where Real Plus Minus comes in. Real Plus Minus uses the technique mentioned above, a Ridge Regression to better adjust for a small sample size, in turn decreasing its bias error. While both of these types regress the point differential of a player's time on the court, Deshpande and Jensen proposed the concept of regressing the change in the home team's win probability between shifts. The goal of this is to provide context to a player's production instead of just taking out of context data. For example, a player scoring 10 points in a shift has a significantly larger effect on a team's chances of winning if it occurs when the game is tied and there's five minutes left in a game. On the other hand, if the same team was down by 30 with the same amount of time left, it would have almost no effect on a team's winning probability. Because of that problem, Deshpande and Jensen believed a player's effect on a team's winning probability better provides context to a player's production.

8.1 Model Creation

[5] The next step in the modeling process is creating the actual structure of the model. Deshpande and Jensen propose a model in the form of:

$$[5]y_i|h_i, a_i = \mu + \theta_{h_{i1}} + \cdots + \theta_{h_{i5}} - \theta_{h_{i1}} - \cdots - \theta_{h_{i1}} + \tau_{H_i} - \tau_{A_i} + \sigma\epsilon_i \tag{8.1}$$

In this equation, y_i is the change in winning probability of the home team during the i^{th} shift. We then denote the partial effects of each individual player as θ. As a whole, $\theta = (\theta_1, \ldots, \theta_{488})^T$, representing the partial effects of all 488 players that are used for the model. Furthermore, we denote the partial effects of each of the 5 individual players of the home team as θ_h, going from h_1 to h_5. On the other hand, the individual players' partial effects for the away team is denoted as θ_a. They then denote the partial effect of the teams themselves as τ_H and τ_A, home and away team effects respectively. As a whole, $\tau = (\tau_1, \ldots, \tau_30)^T$, representing the partial effects of all 30 individual teams. μ is considered the league-average "home-court advantag". Home-court advantage is a common concept in team sports, where the home team has a natural advantage stemming from their familiarity of their surroundings and facilities, lack of needed travel to play, and the fans. Next, we denote the independent standard normal variable as ϵ_i for each shift, and σ is the measure of variability in y_i that comes from the uncertainty and the inherent variability in win probability that cannot be explained by the data.

8.2 Process of Estimating Win Probabilities

Since the main goal of the paper by Deshpande and Jensen is to give NBA player production more context, so that they can better view how NBA players affect winning probabilities. They achieve this goal by measuring win probability changes between shifts. Measuring win probability has been a common theme in

many sports, originating from Mills and Mills[14], who used the concept to evaluate how players in the MLB performed. On top of that, Lindsey [11] estimated win probabilities for MLB games in tangent. Furthermore, this concept has been expanded to basketball by Stern [18], as well by Maymin, Maymin, and Shen [13].

Deshpande and Jensen [5] used this concept of measuring win probabilities in order create their regression model. In their analysis, they modeled winning probabilities of the home team under two conditions. The first condition is the number of seconds that have been played in the game, denoted as T seconds. The second condition was how many points the home team is winning, a positive number, or losing by, a negative number, denoted as L points. With these two conditions they denote the probability of a home team winning a game as $p_{T,L}$. Deshpande and Jensen believed that $p_{T,L}$ is a smooth curve, and their hypothesis is backed by applying the probit model created by Stern [18]. They then needed to decide if it was the correct way to model $p_{T,L}$, which was accomplished by creating a predictive model from the empirical data from their data set, the games from the regular seasons of 2006-2007 season to 2012-2013 season. They then see flaws in both models: the probit model from Stern underestimates $p_{T,L}$ when $L > 0$, and at the same time the probit model overestimates $p_{T,L}$ for when $L < 0$. Deshpande and Jensen attribute this discrepancy to the fact the Stern's model only considers the value of L at the end of the first three quarters, instead of reevaluating $p_{T,L}$ at smaller intervals. For these reasons, they see that the empirical model is much more favorable compared to the probit model. While the empirical model is preferable to the probit model, it still has its flaws, specifically its ability to predict rare occurring values of $[T, L]$. This is the result of limited sample sizes. For example, Deshpande and Jensen point out that there had been one occurrence in which the home team trailed by 18 points after 5 minutes of play. This game resulted in the home team winning the game. This leads the empirical model to tell us that if a home team is trailing by 18 with 5 minutes of game time played,

the home team will win. With basic knowledge of the NBA, that prediction is wrong. This now tells Deshpande and Jensen that their empirical model is highly sensitive to extreme and rare conditions.

Since there is an inherent issue with how the empirical model predicts winning probabilities, stemming from a small sample size, Deshpande and Jensen proposed a new model that would be somewhat of a combination of the two models, the empirical model and probit model. Since basketball possessions are extremely unlikely to last shorter than six seconds, meaning six seconds is most likely the smallest amount of time where the score can change, they decided to create time intervals instead of taking individual seconds. Furthermore, since the maximum points that can be scored in a single possession is 4, a 4 point interval was used. This can be calculated from multiplying points per 100 possessions by effective field goal percentage, combining weighted percentages of 2-point shots, 3-point shots and free throws, and dividing points per 100 possessions by that value. They then decided to create a new model around these concepts. The goal was to create smoother estimates than the empirical model, without sacrificing the complexity that is given by the empirical.

Let $N_{T,L}$ be the number of games in which the home team has led by l points after t seconds of game time has been played [5]. Furthermore, let $L - h_l \leq l \leq L + h_l$ and let $T - h_t \leq t \leq T + h_t$, such that $h_t, h_l \in \mathbb{Z}^+$. Another way to think of h_t, h_l is the range of our intervals mentioned above. Next, let $n_{T,L}$ be the number of games which the home team ended victorious from each window. They then modeled $n_{T,L}$ as a Binomial distribution, conditioned on $N_{T,L}$ and $p_{T,L}$, in other words:

$$n_{T,L} \sim Bin(N_{T,L}, p_{T,L})$$

Given the intervals mentioned above, the assumption is that the win probability should be consistent when $h_t = 3, h_l = 2$. In other words, between the intervals $[T-3, T+3] \times [L-2, L+2]$ [5].

The next task was fixing the problem of a small sample size in extreme leads. They accomplished by placing a conjugate $Beta(\alpha_{T,L}, \beta_{T,L})$ prior on $p_{T,L}$ and the estimate of $p_{T,L}$. This results in a posterior distribution with a mean of $\hat{p}_{T,L}$ such that

$$\hat{p}_{T,L} = \frac{n_{T,L} + \alpha_{T,L}}{N_{T,L} + \alpha_{T,L} + \beta_{T,L}} [5] \tag{8.2}$$

Furthermore, $\alpha_{T,L}$ is defined as pseudo-wins and $\beta_{T,L}$ is defined as pseudo-losses. The goal of these pseudo-games is to shrink the empirical model estimates for $p_{T,L}$ towards $\frac{\alpha_{T,L}}{\alpha_{T,L}+\beta_{T,L}}$, specifically in the extreme and rare events. Since we split up T, L into windows $[T-3,\ T+3] \times [L-2,\ L+2]$, we then get 35-unit cells of the form $[t, t+1] \times [l, l+1$ for each value of T and L. They then added 350 pseudo-games into each window $[T-3,\ T+3] \times [L-2,\ L+2]$, 10 per each unit cell $[t,\ t+1] \times [l,\ l+1]$. Since not all scores result in a 50% split of wins and losses, Deshpande and Jensen made the pseudo-wins and pseudo-losses conditioned on the value of l. If $l < -20$, then 10 pseudo-losses were added to that cell, and if $l > 20$, then 10 pseudo-wins were added to that cell. Lastly, 5 pseudo-wins and 5 pseudo-losses were added to each of the other unit cells.

The goal of this regularization method is to fix the model so that it better predicts the extreme situation, without compromising the model's ability to predict situations that occur very often. In other words, they wanted $\hat{p}_{T,L}$ to be driven by mainly the observed values in common combinations of T and L and when the combination of T and L is rare, they wanted $\hat{p}_{T,L}$ to be more driven by $\alpha_{T,L}$ and $\beta_{T,L}$. This is evident in the example given. When $T = 423$, the observed values of $N_{423,-10} = 4018$, $N_{423,-5} = 11,375$, $N_{423,0} = 17,724$, $N_{423,5} = 14,588$, and lastly $N_{423,10} = 5460$. Furthermore, in cases where $T = 423$ and $-10 \leq L \leq 10$,

the posterior standard deviation of $.003 \leq p_{T,L} \leq .007$, both very small values. This shows very little uncertainty of the estimate $\hat{p}_{T,L}$. On the other hand, when $N_{T,L}$ is close or much less than 350, the influence of $\alpha_{T,L}$ and $\beta_{T,L}$ is increased, smoothing the results of the rarer occurrences of T and L. There is still a much larger uncertainty observed in these situations. This is represented in the much higher standard deviation of the posterior of $p_{T,L}$, which is .035.

The overall goal of this process was to create a smoother model for winning probabilities, conditioned on seconds played and point differential. This goal was accomplished by first by taking a page out of the probit model, a model that only analyzed point differentials at the end of quarters. Using this idea, they chose to increase the amount of seconds between each moment they analyzed point differential and how large of a change in point differential that would result in a change of the probability of winning. These changes were based around how basketball is played, most possessions take at least six seconds, and rules, it is impossible to score more than 4 points in a possession. These changes helped reduce the complexity of the model, and made it more realistic, since basketball has restrictions in game play. The next way they were able to create a smoother and less complex model was placing the conjugate $Beta(\alpha_{T,L}, \beta_{T,L})$ prior on $p_{T,L}$ and the estimate of $p_{T,L}$. The intention of this was to make up for small sample sizes of certain predictable events. The rarity of an early 20-point deficit or lead can cause unwanted effects on a model. If a team is able to come back from this large deficit, it could lead to a model that incorrectly predicts a victory for a team in that situation. By placing the conjugate $Beta(\alpha_{T,L}, \beta_{T,L})$ prior on $p_{T,L}$ and the estimate of $p_{T,L}$, and adding the pseudo-wins and pseudo-losses, those rare occurrences have less of an effect on our model, thus resulting in a more accurate and smoother model.

8.3 Building the Bayesian Linear Regression Model

Based on the equation 8.1, we can see there are a lot of covariates, predictors, in the model created by Deshpande and Jensen. These situations normally go hand in hand with high degree of collinearity, which means it is hard to differentiate which covariate is causing the change in the prediction. This leads us to the application of the regularization techniques mentioned earlier. Deshpande and Jensen chose to apply a Laplace prior on the covariates. The reason for a Laplace prior was chosen over a normal prior is its ability to shrink the covariates at a faster rate than the shrinkage rate from Normal prior. On top of that, the Laplace prior does a much better job at variable selection than the Normal prior. This is because, as mentioned before, when using a Laplace prior on the covariates, they shrink at a much faster, a linear, rate than the Normal prior. This causes some of the covariates to have zero effect on the prediction. In other words, the Laplace prior kills two birds with one stone. This method decreases the complexity of the model, decreasing the bias error, and does variable selection at the same time.

While placing a Laplace prior on the covariates is a better method than placing a Normal Prior in most situations and is used by Deshpande and Jensen [5], it is not the best method of regularization that can be used. The method that is best fit in fitting the model is actually the Laplace-Gaussian prior. The Laplace-Gaussian prior is gives the best results when it comes to regressions with highly correlated independent variables. This method is explained in section 4.3. While it is the best method for regularization, it was unable to be used because there were limitations at the time of the paper was written when it comes to its implementation. At the time, there a widely-available Gibbs sampler for the Laplace-Gaussian prior did not exist.

Given this information, they then build the model conditioned on P^i and T^i [5]. Let P^i be a vector that tells us which player is on the court during the i^{th}

shift. This then makes the j^{th} entry, $P_j^i = 1$ if the j^{th} is on the home team, -1 if they're on the away team and 0 otherwise. Furthermore, let T^i be a vector that denotes which teams are playing during the i^{th} shift. In the same way, for the k^{th} entry, T_k^i is a 1 if the k^{th} team is the home team and -1 if it is the away team and 0 otherwise. y_i is then modeled, conditioned on P^i and T^i:

$$y_i|P^i, T^i \sim N(\mu + P^{i^\top}\theta + T^{i^\top}\tau, \sigma^2)[5] \tag{8.3}$$

The Laplacian priors are then placed on each part of θ and τ, conditional on the corresponding noise parameters σ^2. The conditional prior densities for (θ, τ) given σ^2 is:

$$p(\theta, \tau|\sigma^2) \propto \left[\frac{\lambda}{\sigma}^{488} \times exp\left(-\frac{\lambda}{2\sigma}\sum_{j=1}^{488}|\theta_i|\right)\right] \times \left[\frac{\lambda}{\sigma}^{30} \times exp\left(-\frac{\lambda}{2\sigma}\sum_{j=1}^{30}|\theta_i|\right)\right] [5] \tag{8.4}$$

λ is the the penalty term, such that $\lambda > 0$. They then place a flat prior on μ, a Gamma(r, δ) hyper-prior on λ^2 and non-informative hyper-priors σ^2, r, δ.

Since the current joint posterior distribution of $(\mu, \theta, \tau, \sigma^2)$ is not solvable analytically, is not analytically tractable, another method is needed. Deshpande and Jensen propose using Monte Carlo Markov Chain simulation in order to estimate it [5]. Furthermore, in order to complete the model, other assumptions are needed. The first is the assumption that each component of the vector y_i is independent from the others [5]. While this assumption is not perfect, since most times a substitution does not involve replacing all 10 players on the court, there is likely to be auto-correlation. Though through further examination, it is found that the auto-correlation between y_i and y_{i+1} is -.1. While there is auto-correlation, it is a small amount and the assumption can still be a reasonable assumption. This assumption also takes out the concept of hot and cold streaks. Hot and cold streaks are stretches where a player is playing extremely efficiently for a period of time, while a cold streak is when a player is extremely poor for

a period of time. This would then create situations where the y_i's wouldn't be independent. The second assumption is that the distribution for y_i's are Gaussian with a constant variance, conditional on (P_i, T_i) [5]. The last assumption is that every shift is treated as the same length of time. The idea behind this is that time is irrelevant since the goal of this model is to model overall impact not impact per minute basis. For example, there would be no difference between a shift of 20 seconds, and the winning probability increased by 15%, than a shift of a minute with the same increased win probability. This model creates a situation in which the duration of the shift does not affect the posterior analysis.

9 Results of Model

In order to finally gain values for the model, finding the posterior distributions of each component of y, using a Monte Carlo Markov Chain simulation was needed. In this case, a Gibbs sampler was used. For more information on what a Gibbs sampler entails, view section 5. Using the Gibbs sampler function integrated int the monovm package in R, they were able to obtain 1000 independent samples from the full posterior distribution of $(\mu, \theta, \tau, \sigma^2)$. These data samples aren't enough to consider as the results of this process. Those values need interpretation and need to be used further in order to show not only what the players have done but more a range of how each player effected winning probabilities. This is necessary since the goal of the study is not to just find the average output of each player, but to create a way to compare players. Since average isn't an accurate measure for comparing the players' effects on winning probabilities, outliers cause averages to be an inaccurate tool for comparing, another distribution must be made for each of the players. In order to create a distribution from the 1000 independent samples, for each player, a Gaussian kernel density estimator is used, view section 6. Applying a Gaussian kernel density estimator, centered at each of the average winning probability impact, for each of the player's 1000

independent samples, a posterior distribution density is created for each of the players.

After creating the new posterior densities for each, there is more information that is needed to be found in order to compare each player's effect. The problem with treating each player the same is because different players play in different contexts. In this situation, context references the difference in the amount of shifts each player is a part of and the probability of the player's team winning when they enter the game. This problem caused Deshpande and Jensen to come up with another metric for each player, a leverage profile. This is calculated from a combination of the total shifts for a player, the team's winning probability when entering the game, the average duration of the shifts, and the average length of each shift. After the leverage profiles of each player is calculated, players with similar leverage profiles are then compared.

10 Conclusion

Being able to accurately evaluate NBA players' effect on a team's success, has been an integral part of building a championship team in the NBA. Throughout the years, there have been many attempts to create a metric to rank players on their effectiveness, but they all lack a key component in their analysis, context. Context of a player's production is an extremely important factor when analyzing a player's effect on winning. Not using context in previous player evaluation methods led Deshpande and Jensen to build a model that includes context when analyzing an NBA player's effectiveness. Using machine learning and probability concepts, Deshpande and Jensen were able to create a model that more accurately model how NBA player's effect winning probabilities than previous attempts.

This does not mean it is the best method for NBA player evaluation and still has its shortcomings because it still does tell the whole story. Since basketball is a complex game that is partially dictated by strategy, how and when a player is

used by a coach can greatly affect their effect on winning probability in this model. This means that if a player's role were to be changed, changing teams, injuries, players joining their team, a coach changing their role, etc., their effect on winning probability can completely change. This shortcoming of the model causes it to be much more of descriptive model than a predictive model. On the other hand, it can be used as a moderately accurate predictive model in situations in which a player's surroundings to not alter significantly as well. Another flaw of this model is how the home-court advantage is calculated. Home-court advantage is not uniform for each NBA team, it comes in different shapes and sizes. A perfect example of this is how the Philadelphia 76ers and Dallas Mavericks perform home and away. The Philadelphia 76ers have been the second best home team in the league in the 2019-2020 season. At home, the Philadelphia 76ers have 10.3 net rating [1], meaning they outscore their opponents 10.3 points every 100 possessions, on average. When playing as the away team, they have a net rating of -5.4 points [1]. This is a significant decrease in net rating for playing on the road, a 15.7 net rating differential to be exact. On the other hand, the Dallas Mavericks have a net rating of 5.8 [1] at home and a net rating of 5.7 [1] on the road, a 0.1 differential in net rating. This tells us that some teams' home-court advantage is a huge factor while others are not a factor. This leads me believe that home-court advantage should not be treated the for every team, but instead but individually calculated for each team. One last flaw of Deshpande's and Jensen's model was the use of a lasso regression instead of an elastic net in order to train the original model. This flaw was addressed by the authors themselves[5], but chose not to fix it because of the difficulties that came with the fix. While using an elastic net would have improved the results, Deshpande and Jensen elected to use a lasso because of increased computation difficulty and lack of a widely-available Gibbs sampler function for the elastic net in R[5]. Even though there are flaws in some of their methodology, Deshpande and Jensen were

able to create extremely accurate model using machine learning and probability techniques.

References

1. *Teams advanced*, https://stats.nba.com/teams/advanced/?sort=NET_RATING&dir=-1&Season=2019-20&SeasonType=RegularSeason.
2. *True shooting percentage*, https://www.basketball-reference.com/about/glossary.html#:~:text=TS%-TrueShootingPercentage;,isFGA0.44*FTA.
3. Christopher M Bishop, *Pattern recognition and machine learning*, springer, 2006.
4. Leo Breiman, *Better subset regression using the nonnegative garrote*, Technometrics **37** (1995), no. 4, 373–384.
5. Sameer K Deshpande and Shane T Jensen, *Estimating an nba player's impact on his team's chances of winning*, Journal of Quantitative Analysis in Sports **12** (2016), no. 2, 51–72.
6. Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al., *Least angle regression*, The Annals of statistics **32** (2004), no. 2, 407–499.
7. LLdiko E Frank and Jerome H Friedman, *A statistical view of some chemometrics regression tools*, Technometrics **35** (1993), no. 2, 109–135.
8. John Hollinger, *The player efficiency rating*, 2009.
9. Steve Ilardi, *The next big thing: real plus-minus*, Apr 2014.
10. Steve Ilardi and Aaron Barzilai, *Adjusted plus-minus ratings: New and improved for 2007-2008*.
11. George R Lindsey, *An investigation of strategies in baseball*, Operations Research **11** (1963), no. 4, 477–501.
12. Donald W Marquardt and Ronald D Snee, *Ridge regression in practice*, The American Statistician **29** (1975), no. 1, 3–20.
13. Allan Maymin, Philip Maymin, and Eugene Shen, *How much trouble is early foul trouble? strategically idling resources in the nba*, Strategically Idling Resources in the NBA (February 21, 2012). NYU Poly Research Paper (2012).
14. Eldon G Mills and Harlan D Mills, *Player win averages: A complete guide to winning baseball players*, (1970).
15. Megha Mishra, *Regularization: An important concept in machine learning*, Jun 2018.
16. Dan T Rosenbaum, *Measuring how nba players help their teams win.*

17. Simon J Sheather, *Density estimation*, Statistical science (2004), 588–597.
18. Hal S Stern, *A brownian motion model for the progress of sports scores*, Journal of the American Statistical Association **89** (1994), no. 427, 1128–1134.
19. Robert Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society: Series B (Methodological) **58** (1996), no. 1, 267–288.
20. Ilker Yildirim, *Bayesian inference: Gibbs sampling*, Technical Note, University of Rochester (2012).
21. Hui Zou and Trevor Hastie, *Regularization and variable selection via the elastic net*, Journal of the royal statistical society: series B (statistical methodology) **67** (2005), no. 2, 301–320.