Modeling Basketball's Points per Possession With Application to Predicting the Outcome of College Basketball Games

Ryan J. Parker
rjparker1@edisto.cofc.edu
College of Charleston

Bachelor's Essay

May 5, 2010

Advisors
Amy N. Langville
Martin L. Jones

Abstract

In this paper we consider how to model basketball's points per possession data, and we show that the flexibility provided by a multinomial logistic regression is required for modeling this type of data. We show how to apply this model to ranking college basketball teams, and a method for estimating team win probabilities with this model is provided. We show how to use these win probabilities to fill out an NCAA tournament bracket, and we compare the results of filling out tournament brackets with the multinomial model to the results of a simpler model. We find neither model to be better than the other at predicting NCAA tournament games (in terms of statistical significance).

1 Introduction

In the game of basketball each team works to outscore their opponent in order to win the game. Each opportunity to score points in the game comes in the form of a *possession*, which is the duration of play that takes place from the time the team obtains the basketball until the time their opponent gains possession of the basketball (Kubatko et al., 2007). The focus of this paper will be on analyzing these basic units of play.

Although Oliver (2004) popularized the use of offensive and defensive efficiency ratings (the number of points each team scores and allows per hundred possessions) to evaluate team performance, it can be beneficial to model the probability of scoring a specific number of points on each possession. One benefit of modeling possessions in this way is that one can then estimate the probability of a team scoring more points than their opponent after playing some number of possessions.

In this paper we consider how to model the number of points a team scores on an individual possession. In Section 2 we present how simple models fit points per possession data, and we illustrate which of these models is the best for working with this data. Section 3 shows how this model can be used to rank college basketball teams and predict probabilities of winning future games. Finally, Section 4 concludes the paper with a summary of our findings and results.

2 Modeling Points per Possession

To best work with points per possession data we must first determine which simple model best fits this type of data. This is done so that we can build confidence in the results of a regression analysis on points per possession data, and it is important because we want to use this model to realistically simulate the number of points that are scored on a possession.

We will use Boston's offensive points per possession data from the first game of the 2008-09 regular season to perform this analysis. This is done so that the analysis is on a smaller, more manageable scale. Section 2.1 provides a descriptive statistical analysis of this data set, and Section 2.2 details the method we use to summarize the fit of the model to the data. Sections 2.3-2.7 present the results of fitting various simple models to this data set, and Section 2.8 summarizes these results and makes a case for which model best fits the data.

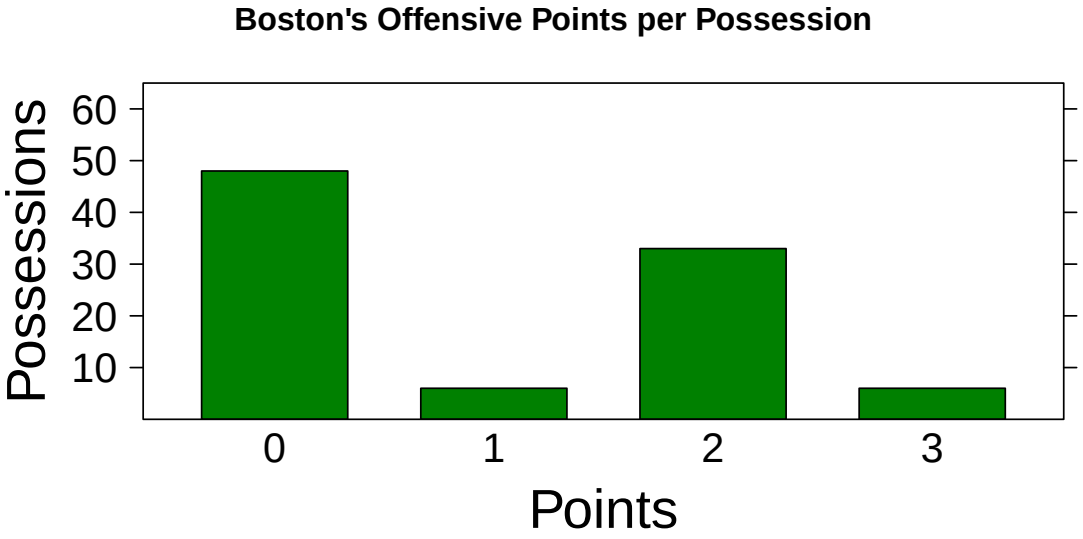


Figure 1: Boston's offensive points per possession versus Cleveland on October 28, 2008.

2.1 Distribution of Points per Possession

The distribution of the Boston Celtics' offensive points per possession from their game against the Cleveland Cavaliers on October 28, 2008 is shown in Figure 1. With peaks at zero and two, this is the typical type of bimodal distribution that is exhibited by points per possession data.

In this game Boston scored a mean points per possession of 0.97 with a standard deviation of 1.07 points. Although these basic summary statistics are helpful, they do not fully capture the structure of the distribution. Thus in Sections 2.3-2.7 below we examine how effective simple models are at capturing the structure of this distribution. First, however, Section 2.2 outlines how we compare the effectiveness of these simple models.

2.2 Checking Model Fit

We will use the simulation procedures described in Chapter 6 of Gelman et al. (2004) and Chapter 8 of Gelman and Hill (2007) to check the fit of each model for choosing the model that best captures the structure of the distribution of points per possession data. These simulations are graphed so that we can visually check the fit of each model for determining which is best at modeling this type of data.

These visual checks are created by replicating the data set using the fitted model with the sample size of the replicated data set equal to the sample size of the original data set. This process is repeated 100 times for each model, and the results are graphed against the original data set to identify inadequacies with the model fit.

2.3 Linear Regression

In this case we fit a normal model of this data of the form

$$Y_i \sim \mathrm{N}(\mu, \sigma^2), \tag{1}$$

where Y_i is the number of points scored on possession i, μ is the mean number of points scored, and σ^2 is the variance of the distribution. To find the distribution of this model we must estimate the mean μ and the standard deviation σ. Using this data set, these estimates are $\hat{\mu} = 0.97$ and $\hat{\sigma} = 1.07$.

With these estimates we can determine how well this model fits the data. This, however, is a continuous distribution, thus we must discretize the outputs. To do this we assume that all values ≤ 0.5 are 0, > 0.5 and ≤ 1.5 are 1, > 1.5 and ≤ 2.5 are 2, and > 2.5 are 3.

Figure 2(a) shows replicated data sets that were simulated from this model. This graphical summary illustrates that this model fails to accurately model the proportion of possessions in which one points are scored. It also does a poor job modeling the number of possessions in which zero and two points are scored.

2.4 Poisson Regression

In this case we fit a Poisson model of this data of the form

$$Y_i \sim \mathrm{Poisson}(\lambda), \tag{2}$$

where Y_i is the number of points scored on possession i, and λ represents the mean number of points scored and variance of the distribution. To find the distribution of this model we must estimate λ. Using this data set this estimate is $\hat{\lambda} = 0.97$.

Figure 2(b) shows replicated data sets that were simulated from this model. This graphical summary illustrates that this model fails to accurately model the proportion of possessions in which one points are scored. It also does a poor job of modeling the number of possessions in which zero and two points are scored.

2.5 Negative Binomial Regression

In this case we fit a Negative Binomial model of this data of the form

$$Y_i \sim \text{NegBin}(\mu, \theta), \tag{3}$$

where Y_i is the number of points scored on possession i, μ represents the mean of the distribution, and the variance of the distribution is $\text{Var}(Y_i) = \mu + \frac{\mu^2}{\theta}$ (Venables and Ripley, 2002). To find the distribution of this model we must estimate μ and θ. Using this data set these estimates are $\hat{\mu} = 0.97$ and $\hat{\theta} = 2.68$.

Figure 2(c) shows replicated data sets that were simulated from this model. This graphical summary illustrates that this model fails to accurately model the proportion of possessions in which one and two points are scored.

2.6 Zero-Altered Poisson Regression

The zero-altered Poisson (ZAP) model allows us to account for extra zeros in the distribution of the data by assuming the zeros come from a binary component and the non-zero values come from a Poisson component. Thus this model is constructed so that:

$$\Pr(Y_i = 0) = \pi \tag{4}$$

$$\Pr(Y_i = y) = (1 - \pi) \times \frac{\lambda^y \times e^{-\lambda}}{y! \times (1 - e^{-\lambda})}, y > 0 \tag{5}$$

In this model Y_i is the number of points scored on possession i, π represents the probability of a zero, and λ represents the mean of the Poisson component (Zuur et al., 2009). To find the distribution of this model we must estimate π and λ. Using this data set these estimates are $\hat{\pi} = 0.48$ and $\hat{\lambda} = 1.6$.

Figure 2(d) shows replicated data sets that were simulated from this model. This graphical summary illustrates that this model fails to accurately model the proportion of possessions in which one and two points are scored.

2.7 Multinomial Logistic Regression

In this case we fit a Multinomial model of this data of the form

$$Y_i \sim \text{Multinomial}(p_0, p_1, p_2, p_3), \tag{6}$$

where Y_i is the number of points scored on possession i, and p_i represents the probability of scoring i points on the possession. To find the distribution

of this model we must estimate p_0, p_1, p_2, and p_3. Using this data set these estimates are $\hat{p}_0 = 0.52$, $\hat{p}_1 = 0.06$, $\hat{p}_2 = 0.35$, and $\hat{p}_3 = 0.06$.

Figure 2(e) shows replicated data sets that were simulated from this model. This graphical summary illustrates that this model is a good fit for modeling the number of possessions in which zero, one, two, and three points are scored.

2.8 The Best Model

We can choose the best model of points per possession data by analyzing the graphical checks of the above fitted models. Below is a summary of how each of the models fit the data:

- **Linear**: This model has two parameters, and it appears to do a good job of modeling the proportion of possessions with three points scored. It appears to do a poor job of modeling the proportion of possessions with zero and one points scored.

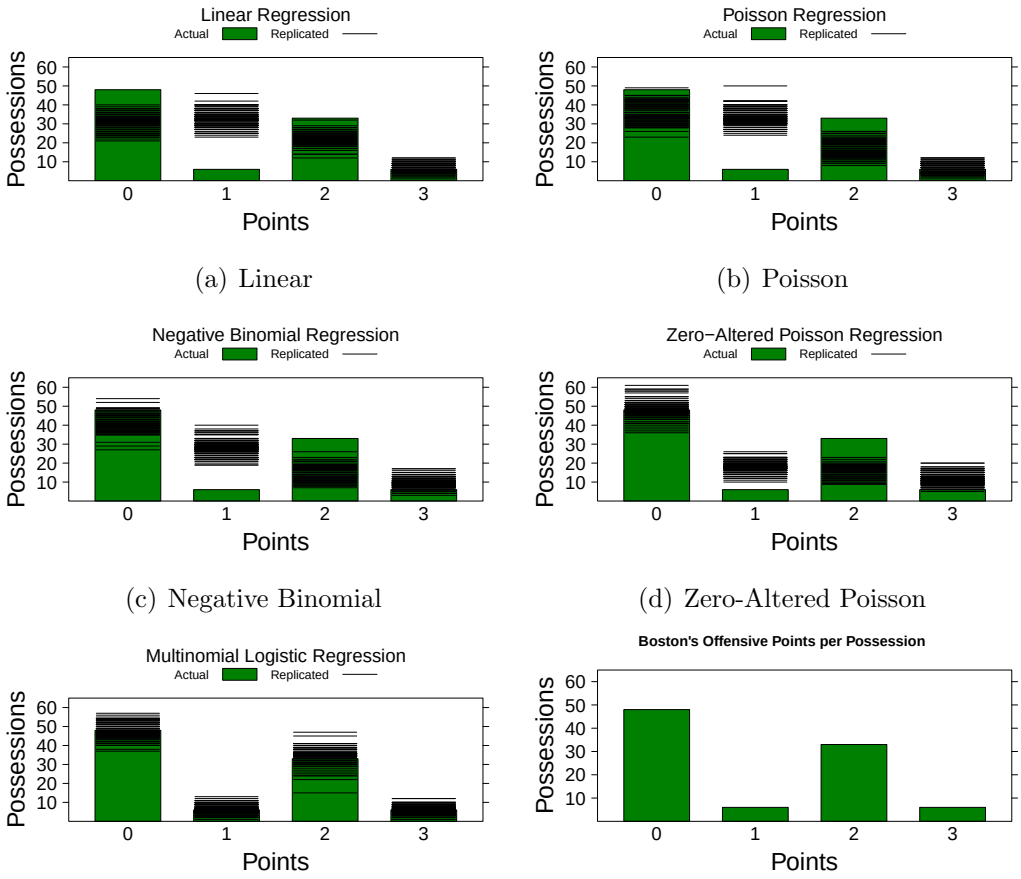
- **Poisson**: This model has one parameter, and it appears to do a good job of modeling the proportion of possessions with three points scored. It appears to do a reasonable job modeling the proportion of possessions with zero points scored, but it does a poor job of modeling the proportion of possessions with one and two points scored.

- **Negative Binomial**: This model has two parameters, but it is similar to the Poisson model. It appears to do a good job of modeling the proportion of possessions with three points scored, and it appears to do a reasonable job modeling the proportion of possessions with zero points scored. Again, like the Poisson model, this model does a poor job of modeling the proportion of possessions with one and two points scored.

- **Zero-Altered Poisson**: This model has two parameters, and it appears to do a good job of modeling the proportion of possessions with zero points scored. It appears to do a poor job of modeling the proportion of possessions with one, two, and three points scored.

- **Multinomial**: This model has four parameters, and it appears to do a good job of modeling the proportion of possessions with zero, one, two, and three points scored. The graphical check of this model shows no obvious signs of an inadequate fit.

Based on these results, the multinomial model is the only model that appears to adequately model the proportion of possessions with zero, one, two,



(a) Linear

(b) Poisson

(c) Negative Binomial

(d) Zero-Altered Poisson

(e) Multinomial

(f) Original Data

Figure 2: The actual number of possessions where zero, one, two, and three points were scored versus replicated data sets from the (a) linear, (b) Poisson, (c) negative binomial, (d) zero-altered Poisson, and (e) multinomial logistic regression models of Boston's points per possession data. The original data (f) are included without replicated data sets for further comparison.

and three points scored. Although some models do a reasonable job with one or two point categories, none of these other models with fewer parameters perform as well as the multinomial logistic regression with four parameters. Therefore, we will use the multinomial logistic regression to accurately capture the distribution of the data.

3 Ranking College Basketball Teams

One way to apply a model of points per possession data is to rank college basketball teams. This area of application is of interest to the many fans that fill out brackets during the NCAA tournament every year, as these rankings can be used to select teams to win tournament games.

In this section we detail how a multinomial logistic regression of college basketball points per possession data can be used to make selections in an NCAA tournament bracket. Section 3.1 outlines the model used to rate each team, and Section 3.1.1 shows how the data used to fit this model is estimated from box score data. Section 3.1.2 shows how to use this model by using an example from the 2009-10 season.

Section 3.2 shows how to use this model for making predictions by estimating win probabilities in Section 3.2.1 and filling out a bracket in Section 3.2.2. In Section 3.2.3 we introduce a simple model to use for comparison, and in Section 3.2.4 we compare how this model performs versus the multinomial model.

3.1 The Model

Based on the analysis in Section 2, we will use a multinomial logistic regression to rank college basketball team's points per possession data. This means that we will assume

$$Y_{ij} \sim \text{Multinomial}(p_{ij0}, p_{ij1}, p_{ij2}, p_{ij3}), \tag{7}$$

where Y_{ij} is the number of points team i scores on an offensive possession against team j, and p_{ijk} is the probability of team i scoring k points against team j on the possession. To estimate these probabilities we fit a generalized linear model of the form

$$\log\left(\frac{p_{ijk}}{p_{ij0}}\right) = \alpha_k + \beta_k(\text{home}) + \gamma_{ik} + \delta_{jk}, \tag{8}$$

where p_{ijk} are as defined above. In this model α_k is the intercept, β_k is the home court advantage, γ_{ik} is the offensive rating for team i, and δ_{jk} is the defensive rating for team j for point totals $k = 1, 2, 3$. In this model point total 0 is the *baseline category*, and thus all $k = 0$ coefficients are fixed at 0.

This model requires that we know the frequency with which teams score points against other teams on each possession. Section 3.1.1 below provides a way to estimate this data from box score statistics, and Section 3.1.2 shows the results of fitting this data for the 2009-10 college basketball season.

3.1.1 Estimating Points per Possession

To fit the model above we must know the number of possessions that each team scored 0, 1, 2, and 3 points[1]. The best way to collect this data is with the play-by-play, but college basketball play-by-play data is scarce. Therefore, from a sample of 2,673 college basketball games from the 2005 to 2008 seasons, we built models for predicting these play-by-play values with box score data.

A linear regression was used to fit these models, and simplified formulas for estimating these values are:

$$\begin{aligned}
\text{zeros} &= FGA - FGM + 0.27(FTA - FTM) - OREB + TOV && (9)\\
\text{ones} &= 0.35(FTA) - 0.25(FTM) && (10)\\
\text{twos} &= 0.95(FGM - FG3M) + 0.36(FTM) && (11)\\
\text{threes} &= 0.02(FGM - FG3M) + FG3M + 0.03(FTM) && (12)
\end{aligned}$$

In these models FGA are field goal attempts, FGM are field goals made, FTA are free throw attempts, FTM are free throws made, OREB are offensive rebounds, TOV are turnovers, and FG3M are three point field goals made. All parameters are statistically significant, and Table 1 presents the actual estimates and standard errors for each of the models.

3.1.2 Model Fit for 2009-10 Season

The values for α_k, β_k, γ_{ik} and δ_{jk} for the model from Section 3.1 were estimated with `R` using `multinom` from the `nnet` package using data from the 2009-10 season. A small set of these coefficients are listed in Table 2.

[1]For simplicity we assume 3 points are scored on the rare possessions in which 4 or more points are scored. From 2005 to 2008 only 0.10% of college basketball possessions ended with 4 or more points being scored. Similarly, approximately 0.17% of NBA possessions end with 4 or more points being scored.

(a) Zeros

$FGA - FGM$	0.974	0.004
$FTA - FTM$	0.269	0.009
$OREB$	−0.959	0.009
TOV	1.020	0.005

(b) Ones

FTA	−0.252	0.012
FTM	0.347	0.009

(c) Twos

$FGM - FG3M$	0.945	0.003
FTM	0.359	0.004

(d) Threes

$FGM - FG3M$	0.020	0.002
$FG3M$	0.995	0.005
FTM	0.033	0.003

Table 1: This table shows the parameter estimates and standard errors for the predictors in the zeros(a), ones(b), twos(c), and threes(d) models for predicting the number of possessions that end with zero, one, two, and three points scored in a game for a college basketball team based on their box score statistics.

		$k = 1$	$k = 2$	$k = 3$
Intercept:	$\hat{\alpha}_k$	−2.39	−0.67	−1.36
Home Court Advantage:	$\hat{\beta}_k$	0.08	0.05	0.04
Offensive Ratings:	$\hat{\gamma}_{(\text{CofC})k}$	−0.46	0.05	0.36
	$\hat{\gamma}_{(\text{UNC})k}$	0.38	0.40	−0.10
Defensive Ratings:	$\hat{\delta}_{(\text{CofC})k}$	0.06	0.29	−0.43
	$\hat{\delta}_{(\text{UNC})k}$	−0.50	−0.17	−0.32

Table 2: A selected list of coefficients from the multinomial college basketball ranking model (Section 3.1) fit to data from the 2009-10 season. Listed are the intercepts, home court advantage, offensive ratings, and defensive ratings for the CofC and UNC.

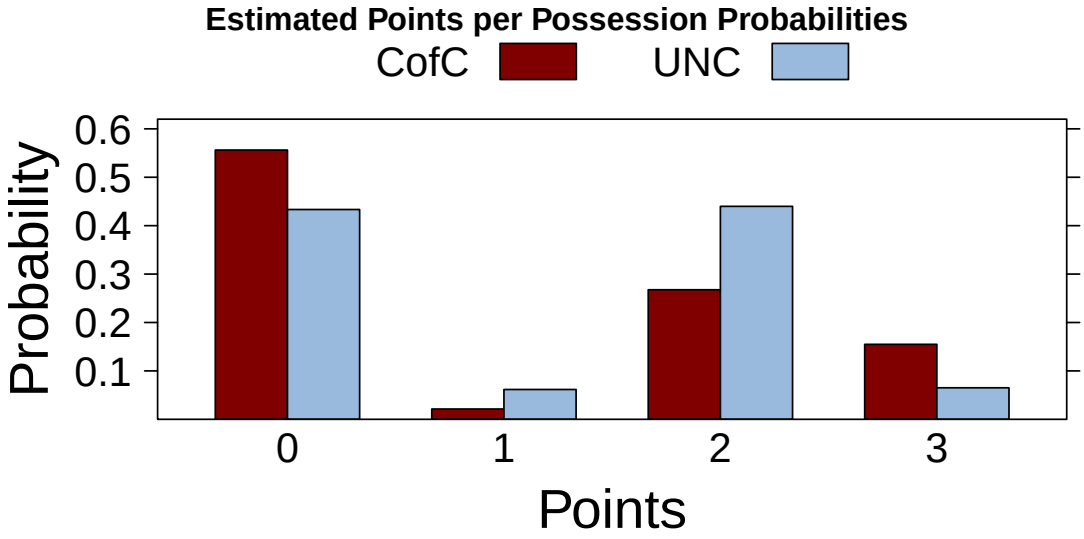


Figure 3: Estimated probabilities for the College of Charleston and UNC scoring 0, 1, 2, or 3 points on a possession for a game played at the College of Charleston's court.

These coefficients can be used to estimate the probability of the College of Charleston (CofC) scoring a specific number of points on an offensive possession versus the University of North Carolina at Chapel Hill (UNC). For example, the estimated probability that the College of Charleston scores 0 points on an offensive possession at home against UNC, $\hat{p}_{(\text{CofC})(\text{UNC})0}$, can be calculated with the following formula:

$$\begin{aligned} \hat{p}_{(\text{CofC})(\text{UNC})0} &= \frac{1}{1+\sum_{k=1}^{3} e^{\alpha_k+\beta_k+\gamma_{(\text{CofC})k}+\delta_{(\text{UNC})k}}} \\ &= 0.558 \end{aligned} \tag{13}$$

These probabilities for $k = 1, 2, 3$ can be calculated for the College of Charleston and UNC using similar equations (Agresti, 2007), and Figure 3 compares these probabilities for each team. These probabilities mean that we would expect the College of Charleston and UNC to score 102 and 114 points per hundred possessions, respectively.

3.2 Predicting the NCAA Tournament

Filling out an NCAA tournament bracket is one of the most popular reasons for wanting to predict the results of future college basketball games, and the model from Section 3.1 can be used to make these predictions. In Section 3.2.1 we provide an algorithm that uses the multinomial model (Equation 7) to estimate probabilities of winning future games, and in Section 3.2.2 we show how you can use these win probabilities to fill out a bracket.

In Section 3.2.3 we introduce a simple model of net efficiency data that can also be used to fill out a bracket, and the results of this model are compared to the multinomial model. The results of this comparison is provided in Section 3.2.4.

3.2.1 Estimating Win Probabilities

The estimated probabilities of scoring 0, 1, 2, and 3 points, calculated with Equation 13, can be used to estimate the probability of each team winning a game. Algorithm 1 specifies how we estimate a team's probability of winning. This algorithm requires that the number of possessions in the game, $nposs$[2], be positive.

This algorithm assumes that possessions are independent, and it assumes that each team will have $nposs$ possessions. Also, ties are ignored. Once the algorithm is complete, team i's probability of beating team j is stored in ρ_{ij}.

3.2.2 Filling Out a Bracket

Tournament brackets are scored by rewarding the player with 2^{r-1} points for correctly selecting a team to win a game in each round $r = 1, 2, \ldots, 6$. Because of this, a player can maximize the number of points they expect to score by making their selections based on each team's probability of progressing to round $r + 1$ (or the team's probability of winning the tournament in the case of $r = 6$).

In order to make these selections the player must calculate the probability of each team progressing to each round. For example, team i's probability of winning the tournament, π_{i7}, can be calculated with

$$\pi_{i7} = \pi_{i6} \times \sum_{j \in O_6} \rho_{ij} \pi_{j6}, \tag{14}$$

[2]We estimate $nposs$ by using a linear regression with the number of possessions in each game as the response and indicators for each team as predictors.

Algorithm 1 Calculate probability that team i beats team j

Require: $p_{ijk} \geq 0$, $nposs > 0$

$nsims \leftarrow 10000$
$wins \leftarrow 0$
$losses \leftarrow 0$
$c \leftarrow 0$
while $c < nsims$ **do**
  $n_i \leftarrow$ (sim from multinomial dist with $p = p_{ijk}, n = nposs$)
  $n_j \leftarrow$ (sim from multinomial dist with $p = p_{jik}, n = nposs$)
  $pts_i \leftarrow n_i \cdot (0, 1, 2, 3)$
  $pts_j \leftarrow n_j \cdot (0, 1, 2, 3)$
  if $pts_i > pts_j$ **then**
    $wins \leftarrow wins + 1$
  else if $pts_j > pts_i$ **then**
    $losses \leftarrow losses + 1$
  end if
  $c \leftarrow c + 1$
end while
$\rho_{ij} \leftarrow wins/(wins + losses)$

where O_r is the set of all possible opponents for team i in round r and ρ_{ij} is the probability that team i beats team j. In general, π_{ir} can be calculated as

$$\begin{aligned} \pi_{ir} &= \pi_{i(r-1)} \times \sum_{j \in O_{r-1}} \rho_{ij} \pi_{j(r-1)} \\ &= \prod_{k=1}^{r} \sum_{j \in O_k} \rho_{ij} \pi_{j(k-1)} \end{aligned} \tag{15}$$

where $\pi_{i0} = 1 \ \forall i$. By using Algorithm 1 to calculate ρ_{ij}, it is then possible to use Equation 15 to calculate $\pi_{ir} \ \forall i, r$. These probabilities can then be used to make selections by choosing the team with the highest probability of winning at each round.

In Section 3.2.3 we introduce an alternative model for calculating ρ_{ij} (and thus π_{ir}), and in Section 3.2.4 we compare how this alternative model predicts the outcome of NCAA tournament games compared to the combination of the multinomial model and Algorithm 1.

Season	Linear	Multinomial	Difference
2003	790	590	200
2004	740	810	-70
2005	1310	1450	-140
2006	730	670	60
2007	1010	730	280
2008	1480	1570	-90
2009	750	780	-30
2010	760	1110	350
Mean	946	964	-18

Table 3: ESPN scores for the linear (Equation 16) and multinomial (Equation 7) models from the 2003 to 2010 NCAA tournaments.

3.2.3 Comparison Model

In this section we present a simple model of efficiency data that can be used to estimate the probability of a team winning a college basketball game. This simple model has the form

$$D_{ij} \sim \mathrm{N}(\alpha(\text{home}) + \beta_i - \beta_j, \sigma_d^2), \tag{16}$$

where D_{ij} is the difference in team i's and team j's efficiency[3] in the game, α estimates home court advantage, and β_i is the rating for team i. In this model the probability of team i beating team j is simply the probability that $D_{ij} > 0$.

3.2.4 Comparison Results

The first way to compare the models is is to examine how they have performed in historical tournaments, and Table 3 compares the ESPN score of each model from the 2003 to 2010 tournaments. (ESPN awards $2^{r-1} \times 10$ points for correct selections in round $r = 1, 2, \ldots, 6$.) This table shows that the multinomial model had a higher average score over this time period, and it also shows that the multinomial model scored more points in 5 of the 8 tournaments.

This type of analysis, however, does not give us much confidence in choosing a model, as a 95% confidence interval for the proportion of tournaments the multinomial model will win over the linear model is (26%, 90%). Therefore we

[3]The number of possessions in the game are calculated from box score statistics with the formula $FGA - OREB + TOV + 0.475 \times FTA$ averaged over the results for both teams. Each team's efficiency is then calculated with points / possessions. (Pomeroy, 2005)

will also consider what happens when the models disagree, an analysis used by Kvam and Sokol (2006).

From 2003 to 2010 the models disagreed on the outcome of 37 tournament games. Of these 37 games the multinomial model predicted 15 (41%) correctly. The 95% confidence interval for the proportion of games the multinomial model predicts correctly when the two models disagree is (25%, 58%). Thus we still do not have conclusive evidence that either model is better than the other.

This analysis does not give us much confidence in choosing one model over another. Thus future work to be done in this area would be to compare these models to other existing models, and also to compare how these models predict regular season games instead of simply NCAA tournament games.

4 Conclusion

In this paper we showed how to best model basketball's points per possession data. We found that the flexibility of a multinomial logistic regression is required to model this data, as the models fewer parameters (linear, Poisson, negative binomial, and zero-altered Poisson) did not have enough flexibility to fully model the bimodal distribution of this type of data. Finding a model to accurately model the distribution of the data was a necessity to realistically simulating the number of points that are scored on a possession.

The multinomial logistic regression gives us confidence that we can model the distribution of this data, and we applied this model to rating college basketball teams. We showed how you can combine this model with Algorithm 1 to estimate win probabilities, and we showed how these probabilities can be used to fill out an NCAA tournament bracket.

We were unable to find a statistically significant difference in the performance of predicting NCAA tournament games between the multinomial model and a simpler linear model. Future work, however, can be done to compare these models with even simpler ranking models that rank teams based on the number of points they score and allow. Also, there is room for further model comparison by estimating the prediction error of each model with cross-validation.

References

Agresti, A. *An Introduction to Categorical Data Analysis*. Wiley, second edition, 2007.

Gelman, A. and Hill, J. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge, 2007.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. *Bayesian Data Analysis*. Chapman & Hall/CRC, second edition, 2004.

Kubatko, J., Oliver, D., Pelton, K., and Rosenbaum, D. T. A Starting Point for Analyzing Basketball Statistics. *Journal of Quantitative Analysis in Sports*, Vol. 3, Iss. 3, Article 1, 2007.

Kvam, P. and Sokol, J. S. *A Logistic Regression/Markov Chain Model for NCAA Basketball*, volume 53 issue 8. Naval Research Logistics, 2006.

Oliver, D. *Basketball on Paper*. Brassey's, Inc., 2004.

Pomeroy, K. *Stats Explained*. http://kenpom.com/blog/index.php/C24/P5/, August 2005.

Venables, W. and Ripley, B. *Modern Applied Statistics with S*. Springer, fourth edition edition, 2002.

Zuur, A. F., Leno, E. N., Walker, N. J., Savelieve, A. A., and Smith, G. M. *Mixed Effects Models and Extensions in Ecology with R*. Springer, 2009.

Appendices

The appendices below provide R code for replicating some of the analyses presented in this paper. These are provided in an effort to help others reproduce this research. Also, some of the data used in this paper and in the code below can be found at

http://www.basketballgeek.com/downloads/be_data.zip

A Visualizing Points per Possession Data

This R code is used for visualizing Boston's points per possession data.

```
library(lattice)

data <- read.csv("BOS.ppp.csv")

ppp <- data.frame(team=c("BOS","BOS","BOS","BOS"), pts=c("0","1","2","3"),
          num=data$n, pct=data$n[1:4]/sum(data$n))

fig <- barchart(num ~ pts, data=ppp, groups=team, xlab="Points",
          ylab="Possessions", main="Boston's Offensive Points per Possession",
          col=c("#008000"), ylim=c(0, 65))
print(fig)
```

B Fitting the Simple Models

This R code is used for fitting Boston's points per possession data with linear, Poisson, negative binomial, zero-altered Poisson, and multinomial logistic regressions.

```
library(MASS) # for glm.nb
library(pscl) # for hurdle
library(nnet) # for multinom

data <- read.csv("BOS.ppp.csv")
pdata <- rep(data$points,data$n)

fit.norm <- lm(pdata~1)
```

```
fit.poi <- glm(pdata~1, family=quasipoisson(link="log"))
fit.negbin <- glm.nb(pdata~1, link="log")
fit.zap <- hurdle(pdata~1, dist="poisson", link="logit")
fit.multinom <- multinom(pdata~1)
```

C Estimating CBB Points per Possession

This R code is used for fitting models that estimate the number of points a college basketball team scored per possession in a game based on their box score statistics.

```
data <- read.csv("cbb_pbp_ppp.csv")

fit.0 <- lm(zero~0+I(fga-fgm) + I(fta-ftm) + oreb + to, data=data)
fit.1 <- lm(one~0+ftm+fta, data=data)
fit.2 <- lm(two~0+I(fgm-fgm3)+ftm, data=data)
fit.3 <- lm(three~0+I(fgm-fgm3)+fgm3+ftm, data=data)
```

D Rating College Basketball Teams

This R code is used for fitting the multinomial logistic regression for rating college basketball teams from the 2009-10 season.

```
library(nnet) # for multinom

ppp <- read.csv("2010.cbb_ppp.csv")

fit <- multinom(points~oloc + factor(oteam) + factor(dteam),
          weights=n, data=ppp, maxit=5000, MaxNWts=5000)
```

This R code is used for fitting the linear regression (the comparison model from Section 3.2.3) for rating college basketball teams from the 2009-10 season.

```
data <- read.csv("2010.mean_ppp.csv")

formula <- as.formula(readLines("2010.mean_ppp.formula"))

fit <- lm(formula, data=data)
```

E Estimating CBB Possessions

This R code is used for fitting a model that estimates the number of possessions in a college basketball game for use with Algorithm 1.

```
data <- read.csv("2010.pace.csv")

formula <- as.formula(readLines("2010.pace.formula"))

fit <- lm(formula, data=data)
```

F Estimating Win Probabilities

This R code is used for estimating the probability of a team beating an opponent using the multinomial model of the probabilities of a team scoring 0, 1, 2, or 3 points on a possession. In this code *nposs* is the expected number of possessions in the game and p_o is a vector of probabilities for the probability the team scores 0, 1, 2, or 3 points on an offensive possession. Similarly, p_d is for defensive possessions.

```
nsims <- 10000
w <- 0
l <- 0
pts.t1 <- rmultinom(nsims,nposs,p_o)
pts.t2 <- rmultinom(nsims,nposs,p_d)

for (k in 1:nsims) {
  t1 <- sum(pts.t1[,k] * c(0,1,2,3))
  t2 <- sum(pts.t2[,k] * c(0,1,2,3))
  if (t1 > t2) {
    w <- w+1
  } else if (t1 < t2) {
    l <- l+1
  } # ignore ties
}

pr_win <- w/(w+l)
```