Predicting NBA Game Outcomes with Hidden Markov Models

Vashisht Madhavan - CS281A: Final Project

vashisht.madhavan@berkeley.edu

Abstract

With the large amounts of recorded data and the recent emergence of advanced statistics, decision making in the NBA has become more data-driven than ever. Despite the plethora of available data, NBA analysts rely on rudimentary ranking systems to predict team performance, failing to leverage powerful statistical estimation methods. In this project, I rely on Hidden Markov Models (HMMs) to model the progression of wins and losses over time, using advanced statistics from NBA game data as features. This approach is able to reasonably model game outcomes in an unsupervised, achieving a prediction accuracy of 73%. The HMM also shows an improvement of 13% over its non-temporal counterpart, the Gaussian Mixture Model (GMM), validating the notion that winning streaks and losing streaks affect the outcome of future games. As advanced statistics also show promising results with supervised methods, I used posterior probabilities from a logistic regression model to project each team's win total for the current NBA season.

1 Introduction

For many years, fans and experts alike relied on "eye-test" hunches and simple statistics to make predictions for NBA games. Although statistics were meticulously recorded, a lack of numerical analysis led many to take crude views about the value of players and the performance of teams. In recent years, however, the NBA has drawn from the success of sabermetrics and has adopted numerous advanced statistics that measure team and player performance. Many general managers rely on statistical analysis to make trades, construct rosters, and even decide on salaries. Despite the surge of data-driven decision making, simple tasks like predicting winners for each game, rely on rudimentary formulas and fail to leverage more advanced statistical estimation methods.

Recent work has shown the effectiveness of mixture models in assigning more refined positions to playersLutz [2012],Lebefure [2014] and better understanding player progressionChase et al. [2015]. Although informative, these works do not take on the task of predicting who wins and loses in each NBA game. Current models such as CARM-Elo from FiveThirtyEightBoice et al. [2015], concentrate on the win prediction problme, but only take into account simple player personnel metrics and ELO ratingsElo [1978]. Although effective, these ratings only look at wins and losses, failing to account for the numerous other statistics that can impact NBA games. For this project, I aim to use more refined statistics from each game as observations for a HMM. This not only takes advantage of the success of graphical models for structured prediction, but also the benefits of higher-dimensional feature spaces.

Additionally, many sports teams show 'hot' and 'cold' streaks over the course of a season, resulting in a series of successive wins or losses. These streaks may increase or decrease a team's chance of winning a priori. Although the idea of streaks is conventional wisdom, there are no direct statistics to back it up. To explore the influence of streaks, I analyze the transition probability estimates from the HMM and also compare accuracy on a test set with that of a GMM, which does not model dependence between games. Section 4 takes a closer look at these results and reveals that game outcomes are indeed affected by the

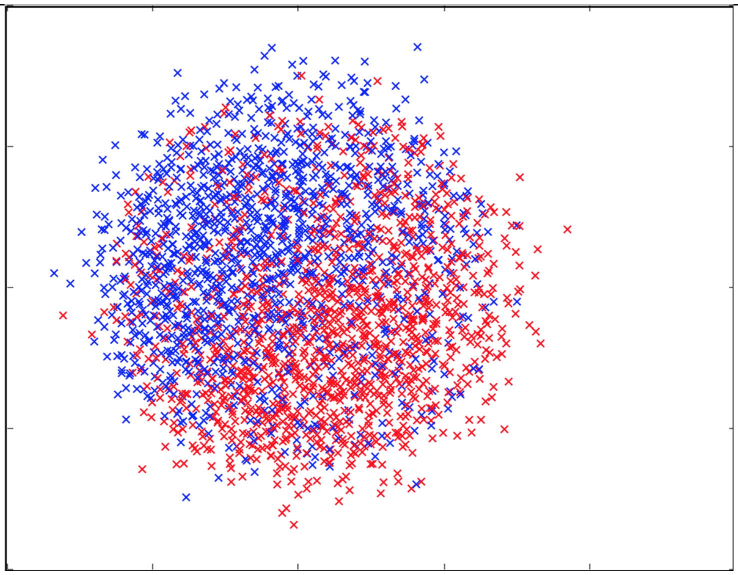


Figure 1: A visualization of the clusters generated by the HMM, with red representing wins and blue being losses. The features are originally in higher dimensions, but I use metric multi-dimensional scalingBorg and Groenen [2005] to project these features onto $\mathbb{R}^2$

results of previous games.

As a final extension to my analysis,I compare the graphical model accuracy with that of supervised alternatives, such as Logistic Regression and Support Vector Machines (SVMs). This comparison not only reveals the impact of supervision on win prediction, but also provides a simple way to determine how well a feature set is able to discriminate wins and losses. Although the supervised models do not consider interactions between games, they outperform HMMs and obtain very good results($\sim$ 98% accuracy). As a result, I used the posterior probabilities of wins and losses from the logistic regression model to run Monte Carlo simulations of the current NBA season. After 5000 simulations, I was able to get reasonable estimates of the expected wins for each team. I was able validate these estimates using current NBA standings and other expert predictions. Section.4 details these experiments and Section. 5 further investigates the results from these models.

2 Related Work

As mentioned in the previous section, there has been much work on learning parametric models from NBA data. Wang and Zemel [2016] relied on recurrent neural networks (RNNs) to classify the types of offensive plays a team runs. Although RNNs show state-of-the-art results for supervised sequence prediction, the focus of this work is on simpler parametric models that learn from data in an unsupervised manner. In addition the RNN proposed in the paper has thousands of parameters, making it a bit over-parameterized for the task at hand. In terms of using graphic, Piette et al. [2011] model player value as a bayesian network and use play by play data to estimate model parameters. Lutz [2012] shows one of the most popular applications of graphical models to NBA data, and uses mixture models to cluster current players based on their per-game statistics. The author also uses prior knowledge about points from each cluster to determine what type of player that cluster typifies. Although these two works provide interesting directions for using graphical models on NBA data, they don not address the task of predicting game outcomes.

Recently, there has also been work on using supervised models to predict outcomes of NBA games.Cheng et al. [2013] and Yang [2015] both tackle the problem of using team information to predict win percentage. Cheng et al. [2013] only uses team efficiency rating, a metric derived from player efficiency rating (PER), to regress to team win percentage. Although the regression model shows small residuals, it only predicts the number of wins for the entire season, and not on a game by game basis. The work by Yang [2015] uses support vector machines and causal weighted regression to predict the betting line for NBA games. Although the method considers uses a complex multivariate feature descriptor to model games, it focuses on a slightly more difficult problem. The betting line approximates how much a team will win or lose by, requiring a more extensive label set and thus making the problem much more nuanced. Ultimately, both of these related works rely on supervised approaches to understanding games. In my project, the HMM clusters wins and losses in an unsupervised manner, while also modeling the temporal evolution of game outcomes.

3 Background

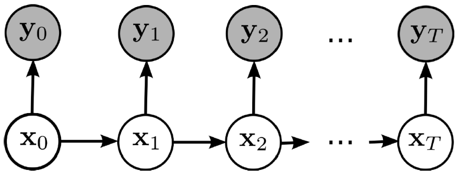


Figure 2: The shaded variables, $\mathbf{y}$, are the observed statistics from game logs over a season. I set $\mathbf{T} = 82$ for each game in the regular season and set $\mathbf{x} \in \{0, 1\}$, with 0 representing a Win and 1 representing a Loss

In the classical formulation of HMMs, $\mathbf{y} \in \mathbb{R}^{d\times T}$ represents observations about a state x over a time horizon T. Each observation, y_i can be a vector of arbitrary dimensions, as long as the emission distribution $P(y_t|x_t)$ can assign the observation a probability. The variables $x_1, x_2, ..., x_T$ are $\{0, 1\}$ random variables that represent latent states, which in this case are wins and losses . The simplest assumption was to model the emission probabilities as multivariate Gaussian random variables, $P(y_t|x_t = i) \sim N(\mu_i, \Sigma_i)$. This not only makes estimation using the EM algorithm simpler, but also enables direct comparison with GMMs, as they too use multivariate Gaussian emissions.

The Viterbi algorithm and Baum-Welch algorithm are both popular methods for estimating HMM parameters, analogous to the max-product and sum-product algorithms for tree-structure graphical models respectively. For this project I chose the Baum-Welch algorithm, as it relies on a simple extension of EM updates and achieves the same log likelihood on the data as the Viterbi algorithm. The parameters to be estimated from the data are the transition matrix, $A_t = P(x_{t+1}|x_t)$, the emission probabilities, $B_t = P(y_t|x_t) \sim N(\mu_{x_t}, \Sigma_{x_t})$, and the prior on the initial state, $\pi_0 = P(x_0)$. These parameters are all initialized randomly. Since EM is an iterative optimization algorithm for a non-convex objective function, the final parameters are subject to bad estimates if the algorithm gets stuck in bad local minima. To mitigate these optimization issues, the best model was taken over multiple restarts. Although there is no guarantee for convergence to a 'good' local minima, multiple restarts is a common way to ensure somewhat reasonable estimatesXu and Jordan [1995]. The updates for the parameters are described in the steps below: Let $\alpha_t(x_t) = P(x_t, y_1, ..., y_t)$

Let $\beta_t(x_t) = P(y_{t+1}, ..., y_T|x_t)$

E-Step:

$$\alpha_t(x_t) = \sum_{x_{t-1}} A_t(x_{t-1}, x_t)B_t(y_t, x_t)\alpha_{t-1}(x_{t-1}) \quad \alpha_0(x_0) = \pi_0 B(y_0, x_0) \tag{1}$$

$$\beta_t(x_t) = \sum_{x_{t+1}} A_t(x_t, x_{t+1}) B_t(y_{t+1}, x_{t+1}) \beta_{t+1}(x_{t+1}) \quad \beta_T(x_T) = 1 \tag{2}$$

M-Step:

Let $\theta_t(x_t) = P(x_t) \propto \alpha_t(x_t)\beta_t(x_t)$ Node Marginals

Let $\theta_{t,t+1}(x_t, x_{t+1}) = P(x_t, x_{t+1}) \propto \alpha_t(x_t)A_t(x_t, x_{t+1})\beta_{t+1}(x_{t+1})B(y_{t+1}, x_{t+1})$ Edge Marginals

$$\pi_0^* = \theta_0 \tag{3}$$

$$A_t(x_t, x_{t+1}) = \sum_{t=0}^{T} \frac{\theta_{t,t+1}(x_t, x_{t+1})}{\theta_t(x_t)} \tag{4}$$

$$\mu_0 = \sum_{t=0}^{T} \frac{y_t}{T_0} \quad \forall t \quad s.t. \quad \theta_t(0) > \theta_t(1) \tag{5}$$

$$\Sigma_0 = \frac{1}{T_0} \sum_{t=0}^{T} (y_t - \mu_0)(y_t - \mu_0)^T \quad \forall t \quad s.t. \quad \theta_t(0) > \theta_t(1) \tag{6}$$

The same updates apply for μ_1 and Σ_1. One nice property of using multivariate gaussian emissions is that μ and Σ are easy to calculate and serve as sufficient statistics for the emission probability distributions.

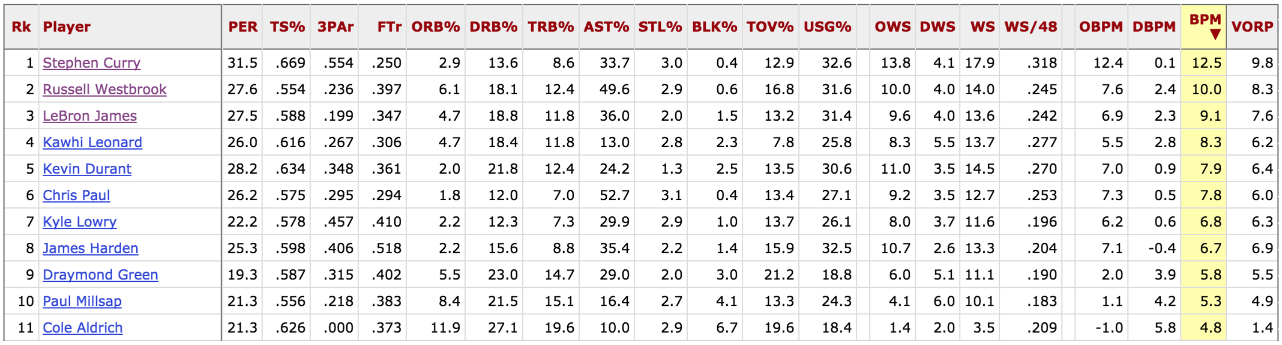
4 Experiments

4.1 Data and Features

The first step to getting a HMM, or any statistical learning method, to work is to choose the appropriate feature space. All of the data used for this project was taken from `http://www.basketball-reference.com/`, which provides both basic and advanced statistics for each team, each season. In addition, the site provides well organized player statistics that contain both player production measures(i.e. points, rebounds, assists, etc.) and advanced measures like Box Plus-Minus (I will get to this later in the section). The raw data is provided in well-formatted CSV files for seamless parsing in python environments. For the experiments in this project, I used data from each game of the 2015-2016 NBA season. Since there are 30 teams and 82 games per team, there are 2460 game observations in total.

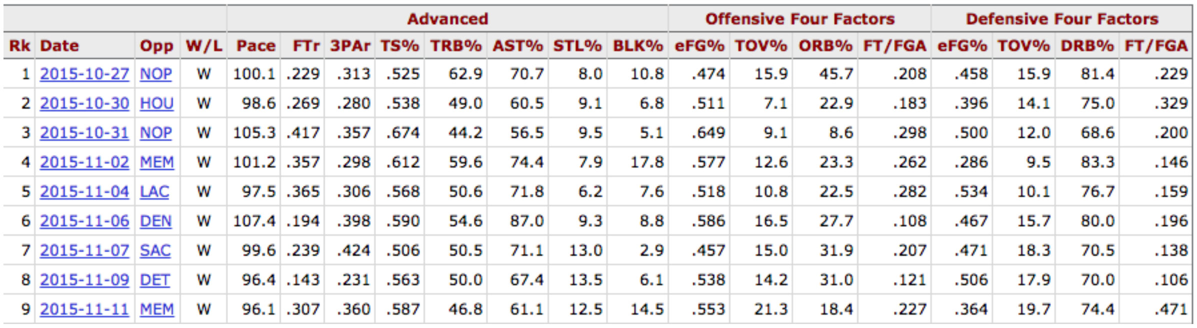
I ran many ablation studies to choose the appropriate feature set, validated by the experiments in Subsection. 4.2. As a first go, I chose to use basic team stats, like points,rebounds,assists,etc. I also used advanced team stats, similar to those shown in Figure.4. More detailed information about these statistics can be found in the Appendix, but for now lets assume that they are good baselines for predicting wins. Although they provide reasonable estimates as to who will win, these features fail to account for opponent information, which can significantly impact game dynamics. If the Golden State Warriors (GSW) play a strong team like Cleveland Cavaliers (CLE), their chance of winning is much lower than if they played a weak team like the Minnesota Timberwolves (MIN). A sufficient and simple way to account for opponent strength is to append opponent statistics to the baseline features. Aside from modeling games on a team level, I also accounted for the impact of individual players on team win percentage. For example, Kevin Durant's offseason move to GSW makes the team much more talented on average, leading to a higher probability of winning. Conversely, in losing Kevin Durant and Serge Ibaka, the Oklahoma City Thunder (OKC) have much less talent on their roster, which can lead to significantly fewer wins. The way I chose to account for these factors is described in the next paragraph

There are a number of statistics, both basic and advanced, that measure a player's impact on team success. The simplest and most effective measure, however, is Box Plus-Minus(BPM). Its a metric that relies on a game's final box score and a player's contribution to that box-score over an average league player. For the 2015-2016 NBA season, players like Stephen Curry and LeBron James lead the league in BPM, providing some external validation for using this metric. However, BPM does not account for the number of minutes played by each player, which can increase or decrease the relative 'value' of his BPM contribution. To account for playing time, I assigned weights to player BPM values based on percentage of total minutes played. These weighted BPM values are the summed up for all players on a given roster, as shown in Equation 7.



Rk	Player	PER	TS%	3PAr	FTr	ORB%	DRB%	TRB%	AST%	STL%	BLK%	TOV%	USG%	OWS	DWS	WS	WS/48	OBPM	DBPM	BPM ▼	VORP
1	Stephen Curry	31.5	.669	.554	.250	2.9	13.6	8.6	33.7	3.0	0.4	12.9	32.6	13.8	4.1	17.9	.318	12.4	0.1	12.5	9.8
2	Russell Westbrook	27.6	.554	.236	.397	6.1	18.1	12.4	49.6	2.9	0.6	16.8	31.6	10.0	4.0	14.0	.245	7.6	2.4	10.0	8.3
3	LeBron James	27.5	.588	.199	.347	4.7	18.8	11.8	36.0	2.0	1.5	13.2	31.4	9.6	4.0	13.6	.242	6.9	2.3	9.1	7.6
4	Kawhi Leonard	26.0	.616	.267	.306	4.7	18.4	11.8	13.0	2.8	2.3	7.8	25.8	8.3	5.5	13.7	.277	5.5	2.8	8.3	6.2
5	Kevin Durant	28.2	.634	.348	.361	2.0	21.8	12.4	24.2	1.3	2.5	13.5	30.6	11.0	3.5	14.5	.270	7.0	0.9	7.9	6.4
6	Chris Paul	26.2	.575	.295	.294	1.8	12.0	7.0	52.7	3.1	0.4	13.4	27.1	9.2	3.5	12.7	.253	7.3	0.5	7.8	6.0
7	Kyle Lowry	22.2	.578	.457	.410	2.2	12.3	7.3	29.9	2.9	1.0	13.7	26.1	8.0	3.7	11.6	.196	6.2	0.6	6.8	6.3
8	James Harden	25.3	.598	.406	.518	2.2	15.6	8.8	35.4	2.2	1.4	15.9	32.5	10.7	2.6	13.3	.204	7.1	-0.4	6.7	6.9
9	Draymond Green	19.3	.587	.315	.402	5.5	23.0	14.7	29.0	2.0	3.0	21.2	18.8	6.0	5.1	11.1	.190	2.0	3.9	5.8	5.5
10	Paul Millsap	21.3	.556	.218	.383	8.4	21.5	15.1	16.4	2.7	4.1	13.3	24.3	4.1	6.0	10.1	.183	1.1	4.2	5.3	4.9
11	Cole Aldrich	21.3	.626	.000	.373	11.9	27.1	19.6	10.0	2.9	6.7	19.6	18.4	1.4	2.0	3.5	.209	-1.0	5.8	4.8	1.4

Figure 3: A sample of player stats used in calculating a team's baseline talent level (i.e. win probability)



				Advanced								Offensive Four Factors				Defensive Four Factors			
Rk	Date	Opp	W/L	Pace	FTr	3PAr	TS%	TRB%	AST%	STL%	BLK%	eFG%	TOV%	ORB%	FT/FGA	eFG%	TOV%	DRB%	FT/FGA
1	2015-10-27	NOP	W	100.1	.229	.313	.525	62.9	70.7	8.0	10.8	.474	15.9	45.7	.208	.458	15.9	81.4	.229
2	2015-10-30	HOU	W	98.6	.269	.280	.538	49.0	60.5	9.1	6.8	.511	7.1	22.9	.183	.396	14.1	75.0	.329
3	2015-10-31	NOP	W	105.3	.417	.357	.674	44.2	56.5	9.5	5.1	.649	9.1	8.6	.298	.500	12.0	68.6	.200
4	2015-11-02	MEM	W	101.2	.357	.298	.612	59.6	74.4	7.9	17.8	.577	12.6	23.3	.262	.286	9.5	83.3	.146
5	2015-11-04	LAC	W	97.5	.365	.306	.568	50.6	71.8	6.2	7.6	.518	10.8	22.5	.282	.534	10.1	76.7	.159
6	2015-11-06	DEN	W	107.4	.194	.398	.590	54.6	87.0	9.3	8.8	.586	16.5	27.7	.108	.467	15.7	80.0	.196
7	2015-11-07	SAC	W	99.6	.239	.424	.506	50.5	71.1	13.0	2.9	.457	15.0	31.9	.207	.471	18.3	70.5	.138
8	2015-11-09	DET	W	96.4	.143	.231	.563	50.0	67.4	13.5	6.1	.538	14.2	31.0	.121	.506	17.9	70.0	.106
9	2015-11-11	MEM	W	96.1	.307	.360	.587	46.8	61.1	12.5	14.5	.553	21.3	18.4	.227	.364	19.7	74.4	.471

Figure 4: Advanced statistics from ten games into the 2015-2016 Season for the Golden State Warriors. Shown are baseline features for estimating wins. Opposing Team statistics were added as well.

$$V_{team} = \sum_{i=1}^{N_{team}} \frac{M_i}{TotalMinutes} * BPM_i \tag{7}$$

P_{team} is a team's value in terms of its players.M_i denotes the minutes played by each player on the team and BPM_i is that player's box plus-minus.

4.2 Supervised Prediction Results and Strength of Features

With all the feature configurations described in Subsection 4.1, understanding which features best discriminate wins and losses was the natural next step. Although intuition posits that the most extensive feature will perform the best, a numerical estimate was needed for validation. Feature selection can be built into the

Table 1: Supervised Prediction Accuracy on Game Log Data

Experiment	Linear SVM	Logistic Regression
Basic Team Stats	83.6%	83.6%
Advanced Team Stats	96.8%	97.0%
Basic Team + Opponent Stats	96.5%	96.8%
Advanced Team + Opponent Stats	98.2%	98.0%
Basic Team + Opponent Stats + Player Info	96.1%	96.7%
Advanced Team + Opponent Stats + Player Info	98.1%	98.3%

Table 2: Unsupervised Prediction Accuracy on Aggregate Game Data

Experiment	GMM	HMM
Basic Team Stats	62.9%	65.2%
Advanced Team Stats	67.1%	72.6%
Basic Team + Opponent Stats	56.3%	66.3%
Advanced Team + Opponent Stats	57.4%	69.0%
Basic Team + Opponent Stats + Player Info	66.2%	67.5%
Advanced Team + Opponent Stats + Player Info	59.2%	73.3%

graphical model with regularization Schmidt [2010], but a model agnostic approach helps separate learning issues from feature construction issues. One good proxy for evaluating the discriminability of features is supervised classifier accuracy.

With learning algorithms and hyper-parameters being equal, it is fair to assume that features that are more discriminatory will result in higher classification accuracy. As mentioned in the introduction, I trained logistic regression and SVM classifiers for all the different feature configurations. For all experiments I used the same data, taking 80% of available data for training, and leaving 20% for model evaluation. The results shown in Table. 1 not only confirm my intuitions about the most extensive feature set being the most powerful, but also reveal that basic statistics can provide a very good baseline for predicting team performance.

4.3 Unsupervised Prediction Results

To see how the different features would perform in an unsupervised setting, I trained both GMMs and HMMs for all the configurations tested in the supervised case. Comparing HMM results with those of sequence-agnostic GMMs helps uncover the influence of previous wins and losses. As Table. 2 shows, conditioning wins and losses on even one previous game provides a significant benefit over no conditioning at all. To get the actual prediction accuracy, cluster labels are assigned to the most common true label for each of their data points.

Although game logs provide good features for model training, adapting the trained model for inference on future match ups is unclear. Using a team's average statistics from the previous year seems like a promising direction, yet issues arise when rosters change or teams show progression. The next section covers these issues in depth.

4.4 Simulation Results

As mentioned in the previous section, it is insufficient to just consider previous year averages for win projection. Mainly because of the variation per game and change in personnel year to year. To account for the player changes, the team value quantity described in Equation. 6 takes previous year weighted BPM values and applies them to 2016-2017 rosters. This assumes that a player's contribution to his team's wins should

Table 3: Projected Records for the 2016-2017 NBA Season

Team	Record	Team	Record
GSW	64-18	TOR	60-22
LAC	60-22	CLE	58-24
SAS	57-25	BOS	50-32
UTA	50-32	CHI	47-35
HOU	50-32	CHO	47-35
OKC	44-38	DET	44-38
POR	42-40	MIL	41-41
MIN	41-41	ATL	40-42
LAL	38-44	WAS	40-42
MEM	36-46	IND	36-46
DAL	33-49	NYK	33-49
SAC	31-51	MIA	28-54
DEN	30-52	BRK	27-55
NOP	30-52	ORL	24-58
PHO	28-54	PHI	18-64

more or less be the same wherever he goes. Although this assumption may not be totally fair, as a player's role and impact can change based on his environment, for this experiment it is reasonable.

After addressing roster changes, I needed to account for game-by-game variation of advanced team statistics.I modeled each team's advanced statistics as a multivariate Gaussian random variable, setting μ_{team} and Σ_{team} to the sample mean and sample covariance of last year's team statistics. A sample from a team's Gaussian and the opposing team's Gaussian were then concatenated for each game in this season and served as feature set for input to the model. This method produced more reasonable projections than simple averaging, but tended to overestimate wins for teams that had done well the previous year, such as OKC, for which the model projected 64 wins despite the loss of star players. In the end I set the Gaussian parameters for each team using statistics from the small sample of games played this season. Although questionable, These averages provided more realistic measures of the impact of roster changes on team success, beyond just the team value metric.

After deciding on the method for generating new data, I used posterior probabilities from each model to simulate the current NBA season. Since HMMs seem to grossly overestimate or underestimate wins, I used probabilities from the logistic regression model to run Monte Carlo simulations. Essentially, the classifier assigns a probability of winning, $p_x$to each feature vector. From this probability, I draw samples uniformly over $[0, 1]$ and assign the vector to a win if the samples drawn fall below p_x and to a loss if the samples are greater than p_x. By doing this many times, for this experiment 5000, the model achieves a more reliable estimate of the expected number of wins. The results are shown in Table 3

5 Analysis

From a glance at the results in Table 1 and Table 2, it is clear that the HMM and GMM perform significantly worse than their supervised counterparts. A closer look at the true positive (TP) and true negative (TN) prediction accuracy for each team sheds some light as to why this happens. In addition, analysis of cluster means and the fitted transition matrix help explain why the HMM does really well for some teams and fails for the others.

The results in Table 4 show that the HMM does well in predicting wins for teams that are very good(e.g.

the San Antonio Spurs (SAS) and Cleveland (CLE)) or very bad.(e.g. the Los Angeles Lakers (LAL) and the Philedelphia 76ers (PHI)). These teams can be thought of as easy cases, as their advanced statistics are almost always going to be either better or worse than those of their opponents. However, the increased prediction accuracy can also be attributed to the transition matrix. The resulting matrix, $A = \begin{bmatrix}0.536 & .464\\ 0.423 & .577\end{bmatrix}$ shows an increased likelihood of staying in the same state as the previous game. The top row represents transition probabilities from "Win" states and the bottom row represents those from "Loss" States. Though marginally higher than random chance, these probabilities provides evidence that previous wins and losses, and furthermore winning and losing streaks, have a definitive impact on NBA games. Since the Markov property forces the window of previous influence to extend only one game back, the transition probability estimate does not account for long-term dependencies. Graphical models that consider these long-term influences will certainly represent transition dynamics better, but are left for future work.

As teams on both ends of the Win/Loss spectrum are 'easy' cases, one might naturally expect difficulty to increase for teams that are closer to the 'middle of the pack'. The results show quite the contrary, as teams that are quite good (e.g. the Los Angeles Clippers (LAC)) achieve worse performance than most middling teams(e.g. the Utah Jazz (UTA) and Houston Rockets(HOU)). The TP and TN rates in Table. 4 suggest that the model effectively predicts losses, but really struggles to predict wins for these good teams. Visualizing the absolute difference between μ_0 and μ_1, as shown in Figure. 5, explains why this happens.

Surprisingly, cluster averages suggest that rebounding percentage (TRB%, ORB%, DRB%), is the statistic with the highest variance between the two clusters, and thus the measure that contributes most to discriminating wins from losses. Good teams for which the model struggles to predict wins, like ATL or LAC, often have much lower rebounding percentages than other, similarly good, teams. In the same vein, bad teams that show poor prediction results have much higher rebounding percentages than other bad teams, causing the model to make a large number of incorrect predictions. This analysis clearly displays the shortcomings of using HMMs to model wins and losses, as total rebounding percentage is often not the most important factor in winning. Analysis of true wins and true losses suggests that true shooting percentage (TS%) is much more critical to wins, making the case that supervision may be necessary to overcome the faulty clustering from the HMM.

Table 4: HMM Prediction Accuracy per Team. They are listed based on Western Conference Standings(Left) and Eastern Conference Standings(Right) for the 2015-2016 Season

Team	Record	Accuracy	TP	TN	Team	Record	Accuracy	TP	TN
GSW	73-9	69.5%	71.2%	55.5%	CLE	57-25	82.9%	89.4%	68.0%
SAS	67-15	75.6%	79.1%	60.0%	TOR	56-26	71.9%	75.0%	65.3%
OKC	55-27	73.2%	87.2%	44.4%	MIA	48-34	63.4%	70.8%	52.9%
LAC	53-29	52.4%	32.1%	89.6%	ATL	48-34	54.8%	39.5%	76.4%
POR	44-38	68.2%	72.7%	62.2%	BOS	48-34	67.1%	62.5%	73.5%
DAL	42-40	56.1%	42.8%	70.0%	CHO	48-34	64.6%	54.1%	79.4%
MEM	42-40	56.1%	42.8%	70.0%	IND	45-37	65.8%	64.4%	67.6%
HOU	41-41	70.7%	63.4%	78.0%	DET	44-38	65.8%	84.1%	52.6%
UTA	40-42	70.7%	87.5%	54.7%	CHI	42-40	63.4%	71.4%	55.0%
SAC	33-49	64.6%	60.6%	67.3%	WAS	41-41	68.2%	63.4%	73.1%
DEN	33-49	65.8%	87.8%	51.0%	ORL	35-47	67.0%	57.1%	74.4%
NOP	30-32	71.9%	66.6%	75.0%	MIL	33-49	53.6%	42.4%	61.2%
MIN	29-53	62.1%	65.5%	60.3%	NYK	32-50	59.7%	59.3%	60.0%
PHO	23-59	65.8%	86.9%	57.6%	BRK	21-61	58.5%	52.3%	60.6%
LAL	17-65	75.6%	58.8%	80.0%	PHI	10-72	81.7%	70.0%	83.3%

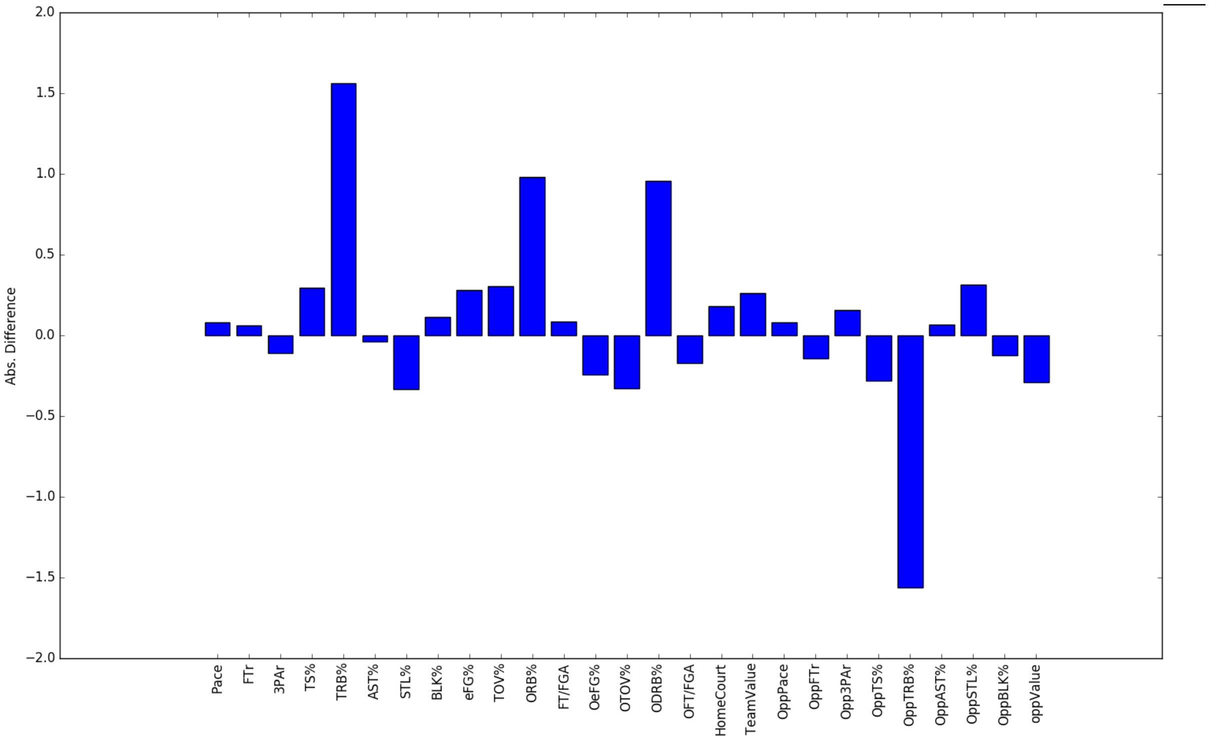


Figure 5: Absolute difference between Win and Loss mean vectors from the HMM model. The positive values show factors associated with winning and negative values show factors associated with losing

Table 5: Team Value by Weighted BPM for the 2015-2016 Season

Team	PV	Team	PV
GSW	13.0	CLE	7.39
SAS	13.5	TOR	5.50
OKC	10.1	MIA	1.79
LAC	4.8	ATL	5.6
POR	2.5	BOS	3.8
DAL	0.4	CHO	3.4
MEM	-0.5	IND	2.2
HOU	1.2	DET	0.2
UTA	3.5	CHI	-0.9
SAC	-2.6	WAS	1.5
DEN	-2.0	ORL	-0.7
NOP	-4.3	MIL	-3.7
MIN	-3.2	NYK	-3.1
PHO	-7.3	BRK	-8.3
LAL	-9.6	PHI	-10.1

6 Conclusion and Future Work

Although the HMM did not achieve stellar performance with the current feature set, the method showed promise in modeling temporal trends between wins and losses. A positive is that the HMM clustered wins and losses at a clip well above random and significantly outperformed the GMM. The performance increase provides some evidence for using sequence models for future work in NBA game prediction. The Markov property forces each latent state in the HMM to only be conditioned on its previous state, often masking long-term dependencies. To better model the evolution of wins and losses over a season, future work may consider more extensive conditioning and look into auto-regressive models.

The model already takes advanced player and team statistics into account, yet it fails to consider factors like injuries and player progression between seasons which can definitively affect game outcomes. Wins in basketball also rely highly on teammate interaction and players filling specific roles on their teams. A lineup of all-star centers may not be better suited to win than an average starting lineup, which is also something this model does not take into account. Although relevant to win probability, these factors are quite difficult to measure quantitatively and no definitive metrics exist. For future work, smart ways to quantitatively measure these factors can go a long way in helping the HMM perform well. However, increasing the number of features also increases the complexity of the model, which can cause it to suffer from the 'curse of dimensionality'.

Another interesting path for further analysis is supervised learning for outcome prediction, the current features and a simple logistic regression classifier already achieve great accuracy($\sim$ 98%). The accuracy could be further improved with sequence modeling and some of the more refined measures discussed in the previous paragraph.

All code and data can be found at `https://github.com/VashishtMadhavan/HmmNba`

References

J. Boice, R. Fisher-Baum, and N. Silver. How Our 2015-16 NBA Predictions Work. 2015. URL `http://fivethirtyeight.com/features/how-our-2015-16-nba-predictions-work/`.

I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications.* Springer, 2005.

H. Chase, N. V. Steeg, and D. Smith. Player Progression in the NBA. *Harvard Sports Analytics Collective*, 2015.

B. Cheng, K. Dade, M. Lipman, and C. Mills. Predicting the Betting Line in NBA Games. 2013. Class Project for Stanford CS229: Machine Learning,Winter semester 2013.

A. E. Elo. *The rating of chessplayers, past and present.* Arco Pub., New York, 1978. ISBN 0668047216 9780668047210. URL `http://www.amazon.com/Rating-Chess-Players-Past-Present/dp/0668047216`.

L. Lebefure. Understanding Player Positions in the NBA. 2014. Class Project for Stanford CS229: Machine Learning,Winter semester 2014.

D. Lutz. Cluster Analysis of NBA Players. *MIT Sloan Sports Conference*, 2012.

J. Piette, L. Pham, and S. Anand. Evaluating Basketball Player Performance via Statistical Network Modeling. *MIT Sloan Sports Analytics Conference*, 2011.

M. Schmidt. *Graphical Model Structure Learning with L1-Regularization.* PhD thesis, University of Alberta, 2010.

K.-C. Wang and R. Zemel. Classifying NBA Offensive Plays Using Neural Networks. *MIT Sloan Sports Analytics Conference*, 2016.

L. Xu and M. I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8:129–151, 1995.

Y. Yang. *Predicting Regular Season Results of NBA Teams Based on Regression Analysis of Common Basketball Statistics*. PhD thesis, UC Berkeley, 2015.

A Basic Stats

- **FG%** - team percentage of field goals(both 2 and 3 Point shots) made
- **3P%** - team percentage of 3-Point attempts made
- **FT%** - team percentage of Free Throw attempts made
- **ORB** - team offensive rebounds
- **TRB** - team total rebounds
- **AST** - team total assists
- **STL** - team total steals
- **BLK** - team total blocks
- **TOV** - team total turnovers
- **PF** - team total personal fouls

B Advanced Stats

- **Pace** - estimate of possessions per 48 minutes
- **Ftr** - number of free throw attempts per field goal attempt
- **3PAr** - percentage of field goal attempts from 3 point range.
- **TS%** - true shooting percentage. A measure of shooting percentage that accounts for the increased value of 3-Pointers.
- **TRB%** - percentage of available rebounds grabbed by a team
- **AST%** - percentage of team field goals that are the result of an assist
- **STL%** - percentage of opponent possessions that end with a steal
- **BLK%** - percentage of opponent possessions that end with a block
- **TOV%** - estimate of team turnovers per 100 possessions
- **ORB%** - percentage of available offensive rebounds grabbed by team
- **DRB%** - percentage of available defensive rebounds grabbed by team
- **FT/FTA** - team free throw attempts per field goal attempt