Who's hot and who's not? Uncovering consistency in NBA players using Hidden Markov Models

Felix Adam[a], Julia Barnett[a], Oscar Martínez[a], Guillem Sitges[a]

[a]*Barcelona Graduate School of Economics, Barcelona, Spain*

Keywords: Hidden Markov Models, Sports Analytics, Dynamic Programming

1. Introduction

The hot hand fallacy is a well known issue in sports analytics and human perception. First discussed by Gilovich et al. (1985), it describes the misconception that players tend to have higher chances of scoring points if they have scored in the previous shot. However, a look at sports media reveals that the concept of being "hot" is still in use today. We try to approach the concept of "hotness" from a different angle; we look at overall game performance rather than single outcomes such as shots made. We assume that a player's performance in each NBA game in fact depends on an underlying hidden state such as "hot" or "cold". Using game-level data for the 2017-18 and 2018-19 NBA seasons, we fit a Hidden Markov Model to uncover these states. Our results closely align with common conceptions about current players in the NBA. Superstars such as James Harden and Lebron James tend to stay "hot", and even when "cold" they have above-average games. The average player, however, tends to only have an average game when he is hot. While our results should not be interpreted as a proof for "hotness", they can be used as a tool for scouting reports or management decisions because they jointly describe a player's performance and consistency.

Email addresses: felix.adam@barcelonagse.eu (Felix Adam), julia.barnett@barcelonagse.eu (Julia Barnett), oscar.martinez@barcelonagse.eu (Oscar Martínez), guillem.sitges@barcelonagse.eu (Guillem Sitges)

2. Data and Features

Our analysis is based on player-level data for each game in the 2017-18 and 2018-19 NBA seasons. We have obtained the data from www.basketball-reference.com using web-scraping methods. The data of interest to us was specifically taken from the game logs of players during the 2017-18 season and the 2018-19 season. There are 82 games in a season, so with data on 497 players we have a total of 26,003 observations in the first data set (2017-18 season) and 387 players and 20,228 observations in the second data set (2018-2019 season, which has yet to finish).The data included information for every player and each game played; some statistics reported include seconds played, shot percentages, rebounds, assists, steals, blocks, turnovers, points, the overall game score, and their plus minus score. Overall distributions for these statistics can be found in Appendix A.

Instead of focusing on a single outcome measure, we created a "Game Value Index" (GV) using a combination of these statistics. This simplified the analysis given that the different statistics have different distributions. Further, it can be used to obtain an overall view of a player's impact in a game. The metric is derived by combining the most important positive individual statistics (field goal percentage, three-point percentage, free throw percentage, total rebounds, steals, assists, blocks, and total points), subtracting the negative one (turnovers), and finally dividing by the seconds played per game to standardize the metric per game (see Appendix B for overall distributions).

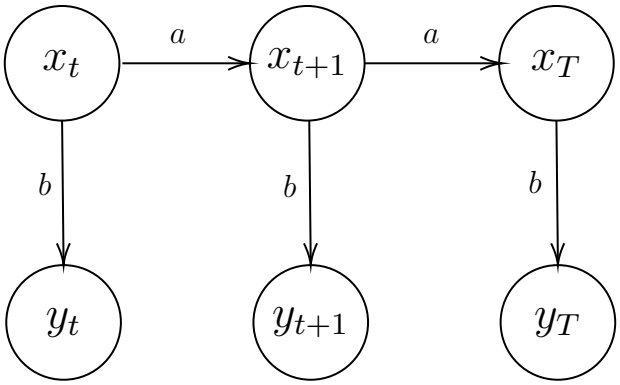
$$GV = \frac{\%\text{FG} + \%\text{3PT} + \%\text{FT} + \text{Rebound} + \text{Steal} + \text{Assist} + \text{Block} + \text{Points} - \text{Turnovers}}{\text{Seconds Played}}$$

To further simplify the analysis, we created a discrete variable called "Game Quality" from the Game Value Index. A player had a good game if his Game Value Index was above the 70^{th} percentile of all games, a bad game if his outcome was below the 30^{th} percentile and an average game between these two. We then used this Game Quality measure as our observation in the context of a Hidden Markov Model.

3. Analysis

3.1. Model

As described, we assumed that a player's performance can be described by a Hidden Markov Model with the following structure:



In which we only observe the outcomes y_t. The primitives are defined as follows:

- $x_t \in$ [Cold, Hot], the hidden state
- $y_t \in$ [Bad, Average, Good], the observed outcomes as described in the previous section
- $b_{ij} = P(Y = j|X = i)$, the emission probabilities of observing a game outcome conditioned on being hot or cold
- $a_{ij} = P(X_{t+1} = j|X_t = i)$, the transition probabilities of changing from a given state to another (or staying at the same state) between periods
- π_i: the initial probabilities of transitioning to a hidden state (priors)

The most important model assumption is that the state x_t only depends on the previous state x_{t-1} and is conditionally independent of all other previous states. In the basketball context, this means that the probability of being hot or cold in the current game only depends on the previous game. We acknowledge that this is quite a strong assumption, particularly given the research on the hot hand fallacy. However, even though we cannot assume any predictive power of our model, we argue that it helps to uncover overall player

quality. This is further supported by the consistency of our results over the two observed seasons as shown in section 4.

3.1.1. Fitting the Model

Our main interest lies in the transition and emission probabilities for each player as a measure of consistency and overall performance. We use the Baum-Welch algorithm to uncover these as the parameters of the Hidden Markov Model. The algorithm makes use of dynamic programming and an Expectation-Maximization procedure to fit the model to a given data set. We follow Rabiner (1989) for the derivation of the algorithm.

The parameters to be estimated are those of the HMM:

- **Transition matrix**: denoted by $A = \{a_{ij}\} = P\left(X_t = j | X_{t-1} = i\right)$
- **Emission matrix**: denoted by $b_j\left(y_i\right) = P\left(Y_t = y_i | X_t = j\right)$ and B the collection of b_j. Generally, a distribution is assumed and this matrix contains the different parameters for each state of the assumed distribution
- **Initial parameters/Priors**: denoted by $\pi_i = P\left(X_1 = i\right)$

Given a sequence of observations $Y = y_1, ..., y_T$, the algorithm will find the parameters A, B and π that maximize the probability of the observed sequence. We will now describe the algorithm in detail, following the steps described by Rabiner (1989).

3.1.2. Forward step

In the forward step we compute the probability of observing the sequence Y and being in a state x at time t, given the parameters θ as $\alpha_i(t) = P\left(Y_1 = y_1, \ldots, Y_t = y_t, X_t = i | \theta\right)$. This step is computed from the first period to the last. It is initialized in period one using the priors and the emission probabilities b:

$$\alpha_i(1) = \pi_i b_i\left(y_1\right)$$

The subsequent steps are computed as:

$$\alpha_i(t+1) = b_i\left(y_{t+1}\right) \sum_{j=1}^{S} \alpha_j(t) a_{ji}$$

3.1.3. Backward Step

In the backward step, the probabilities of observing the remaining sequence given an assumed state x in time t are computed as $\beta_i(t) = P\left(Y_{t+1} = y_{t+1}, \ldots, Y_T = y_T | X_t = i, params\right)$. This step is computed from the last period to the first. The last period is initialized as $\beta_i(T) = 1$.

Then, using induction, each previous step will be computed as:

$$\beta_i(t) = \sum_{j=1}^{S} \beta_j(t+1) a_{ij} b_j\left(y_{t+1}\right)$$

Both the backward and forward step make use of dynamic programming by storing previous results for subsequent calculations. Using the forward and backward step, we do not need to compute all possible probabilities for all sequences at each step which dramatically reduces computational complexity.

3.1.4. Updates

Once the Forward and Backward step are computed, we can proceed to compute the temporary variables γ and ξ that will be used to update the parameters. This is the estimation step.

$\gamma_i(t)$ describes the probability of being in state i at time t given the sequence Y and the current parameters θ.

$$\gamma_i(t) = P\left(X_t = i | Y, \theta\right) = \frac{P\left(X_t = i, Y | \theta\right)}{P(Y|\theta)} = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$$

ξ is used to compute the probability of transitioning from a state i to another state j at a given point in time. It computes the joint probability of observing the sequence and a de-

fined transition, and also normalizes it by the probability of observing the sequence.

$$\xi_{ij}(t) = P\left(X_t = i, X_{t+1} = j | Y, \theta\right) = \frac{P\left(X_t = i, X_{t+1} = j, Y|\theta\right)}{P(Y|\theta)}$$
$$= \frac{\alpha_i(t) a_{ij} \beta_j(t+1) b_j\left(y_{t+1}\right)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i(t) a_{ij} \beta_j(t+1) b_j\left(y_{t+1}\right)}$$

Having calculated γ and ξ, we can update $\theta = (A, B, \pi)$ in the maximization step.

- Transition matrix:

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

- Emission matrix:

$$b_i^*\left(v_k\right) = \frac{\sum_{t=1}^{T} 1_{y_t = v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

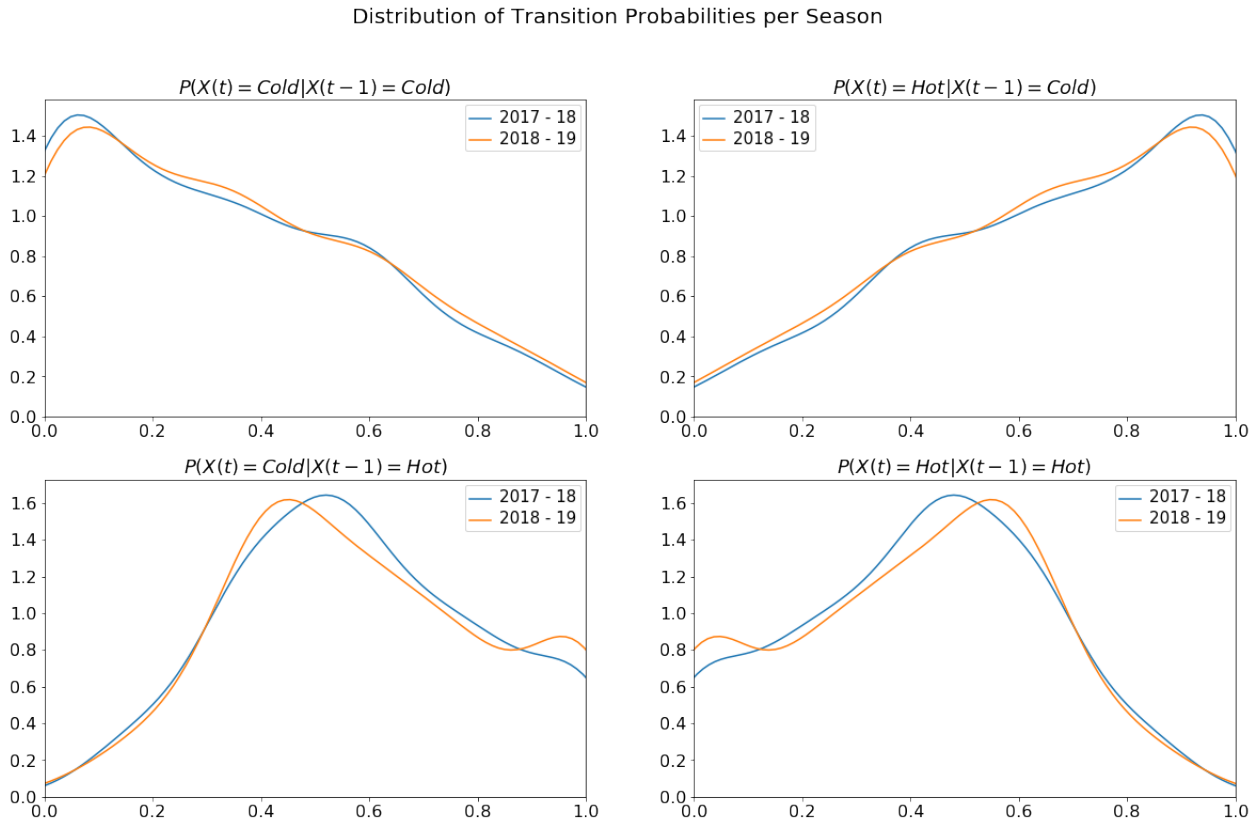
- Priors: given by the first parameter of the γ sequence $\pi_i = \gamma_i(1)$

These steps are repeated iteratively. In our case we have used a fixed number of iterations. Generally, the algorithm is guaranteed to find a better set of parameters θ at each step k. It is guaranteed that at each step $P(Y|\theta_{k+1}) > P(Y|\theta_k)$. However, the algorithm might converge to a local maximum, so it is best practice to initialize the algorithm with new starting parameters and repeat the estimation (Rabiner (1989)). Then, the results can be averaged to obtain the final parameters. The implementation of the algorithm can be found in Appendix C

4. Results

We run the algorithm for each player and season to obtain the emission and transition parameters. For each player, we use the same priors to initialize the algorithm. The maximum number of iterations is set to 50. First, we present the overall parameter distributions.

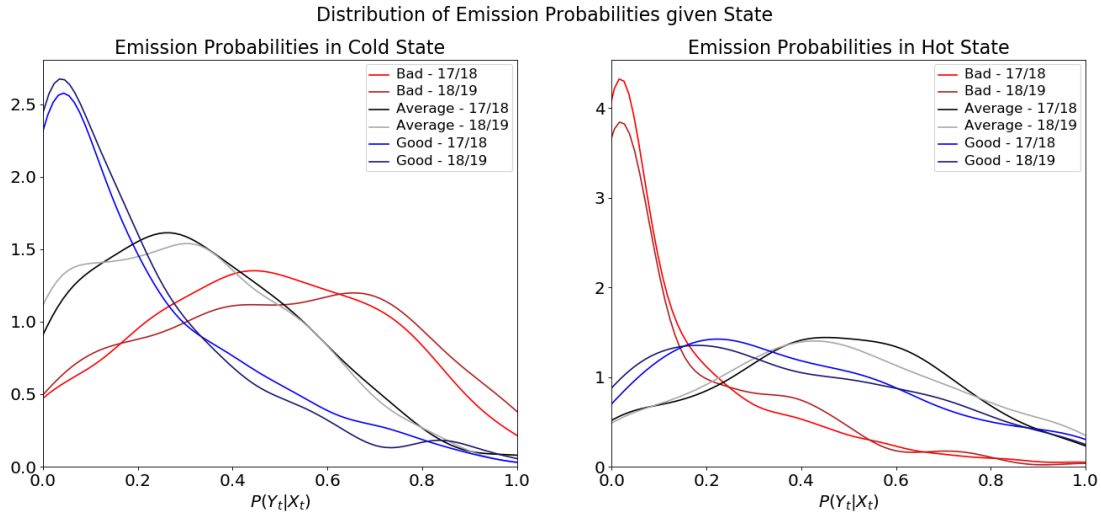
4.1. Transmission and Emission Probabilities



First we note that the distributions are complements, which is given due to the fact that we only included the two hidden states. Looking at the top row, we see that it is actually quite unlikely for players to stay cold. As shown in the top right plot, most players have a high probability of transitioning from cold to hot. However, we also see that not many players have a high probability of staying hot, as shown in the bottom right plot. This makes sense given the many players in the NBA yet the select few that are identified as superstars. These consistent players are on the right tail of the Hot→Hot plot (bottom right); this zone of outstanding players seems to be even rarer than the awful players who seem to fall in the Cold→Cold zone for a large proportion of the time. A key takeaway

is that what makes a good player is the high probability of staying in this hot zone, not just the ability to recover from a cold game.

Obviously, the transition probabilities alone are not conclusive; it is important that when the player is hot, he also has a good game. In order to better understand this we need to also analyze the emission data.



We can see that for most players in the cold zone, it is almost impossible to have a good game (blue lines), and having a bad game is the most likely event although some players have average games while cold. Contrarily, when a player is hot, it is unlikely they will have a bad game. Interestingly, the emission probabilities of having an average game when hot are on average higher than the probability of having a good game. We can therefore conclude that average players tend to have average games when hot. This leads us to the observation that superstars are the players who almost certainly perform good while hot, and still perform average or even good while cold.

For both emission and transition probabilities, we see almost no changes in distribution. This supports the underlying model assumptions. If the model would just fit noise, we'd expect the distributions to differ over seasons.

4.2. Top Players

We now use our results to identify the top performers of the 2017-18 season. These are players that have a low probability of staying cold, a high probability of staying hot, a

high probability of having a great game even while cold, and even further are almost certain to have a good game while hot. The results are summarized in Appendix D.

The majority of these players seem very intuitively correctly identified; our algorithm consistently identifies the superstars of the NBA. These are the players who rarely have a bad game and stay hot with much higher probabilities than average. Interestingly, even these superstars stay in the hot zone with a probability less than 0.65, so this might not be a defining characteristic of a good player. Instead, what seems more important is that even if the player is not in a hot zone, they are still outperforming the other players (who could very well be hot). Additionally, once these players are hot they are certain to dominate the court; even when they are cold, Lebron, Davis, Embiid, and Harden will have a good game with probability greater than 0.8.

Further we can look into changes between the two seasons: players who dropped out or are rising stars in 2018-19 (see Appendix D). Unsurprisingly, Giannis is a rising star during this season. Cousins and Whiteside fell out of the leaderboard, but the rest of our stars maintained their positions with dignity.

The consistency of our results on a player basis further supports the use of a HMM.

4.3. Bad Players

Finally, we can look at the apologetically dreadful players (Appendix E). These guys are the ones who almost always stay cold, when they are cold will certainly have a bad game, and even when hot will only have an average game. Interestingly, they only ever seem to have average games. However, these are still NBA players and are certainly earning salaries that help them get through seeing their name on this pitiful list (for reference, Wesley Johnson is projected to earn $6,134,000 during this season).

5. Discussion

5.1. Extensions

Ultimately, even though our model results seem to be consistent over seasons, the next step would be to perform model diagnostics such as testing for Markov properties. Further, our

analysis relies on the use of the discrete Game Quality observation. Subsequent analyses could use different outcome measures to describe the joint distributions of statistics such as shots made and assists.

5.2. Potential Use Cases

The GameValueIndex itself could be modified to fit the goals of the end user. For example, if a coach wanted to sign a good defensive player then they could assign greater weights to rebounds, blocks, steals, etc. Likewise, the analysis could be used to select the Most Valuable Player of Defensive Player of the Year. From a management perspective you could also filter for players with low contracts in order to find highly consistent players that are extremely valuable for their cost. Most improved players are easy to identify; a player that becomes more consistently good is much more desirable than a streaky player. Coaches could also use this information for load-management for players. They could build optimal player rotations which ensure that at least one or two consistent players are playing in each game.

6. Conclusion

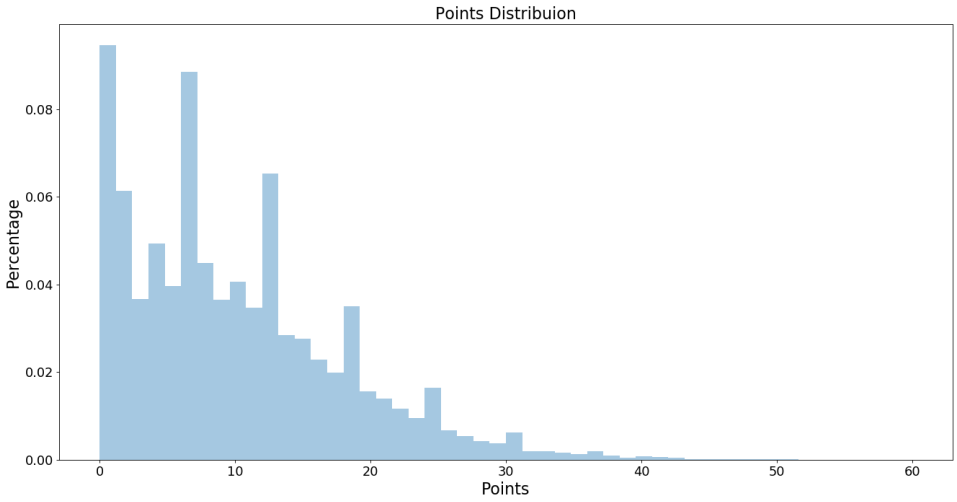
We have fit a Hidden Markov Model on player level data for the NBA seasons 2017-18 and 2018-19 to uncover player quality and consistency. We have assumed that players can be either hot or cold and perform accordingly. Our results are closely aligned with the common perception of players, such as superstars like Harden and James. Very good players tend to have average or even good games when hot and are unlikely to stay cold. Average players tend to have only average games, even when hot. Our anlaysis can be used for scouting reports or team-management decisions.

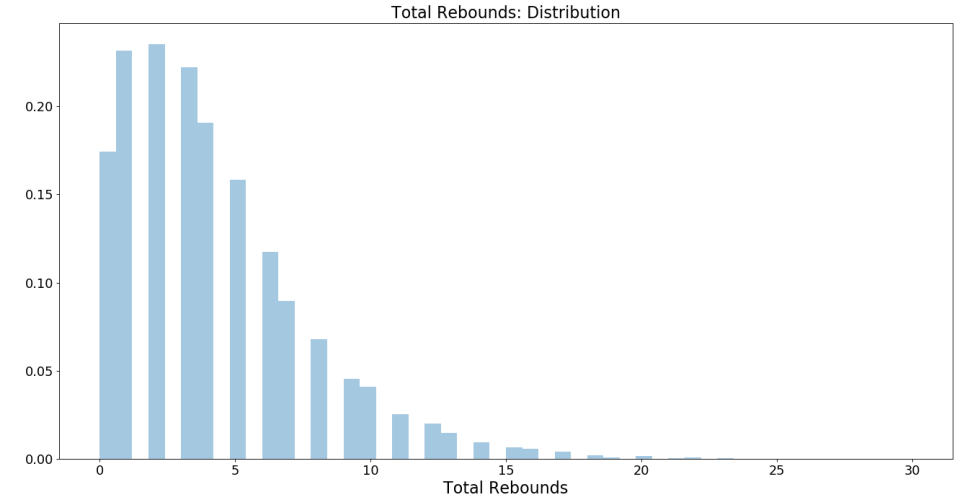
7. References

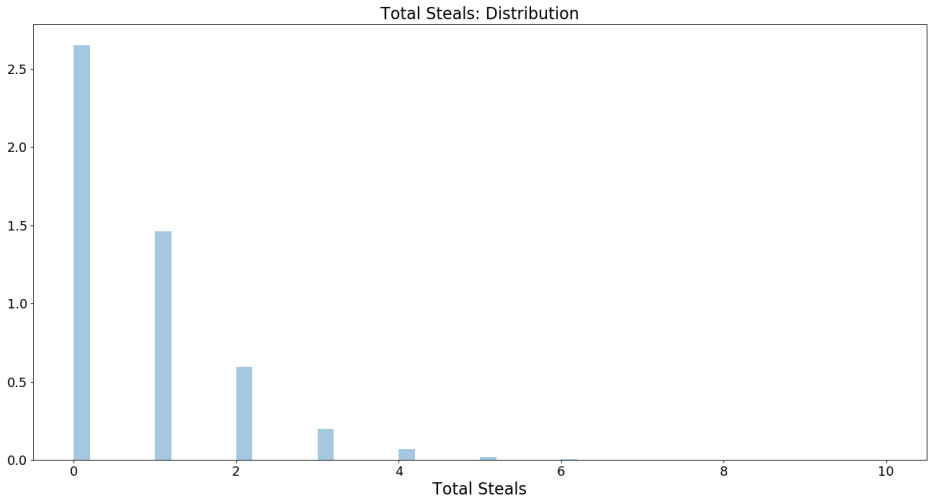
Gilovich, T., Vallone, R. and Tversky, A. (1985), 'The hot hand in basketball: On the misperception of random sequences', *Cognitive psychology* **17**(3), 295–314.

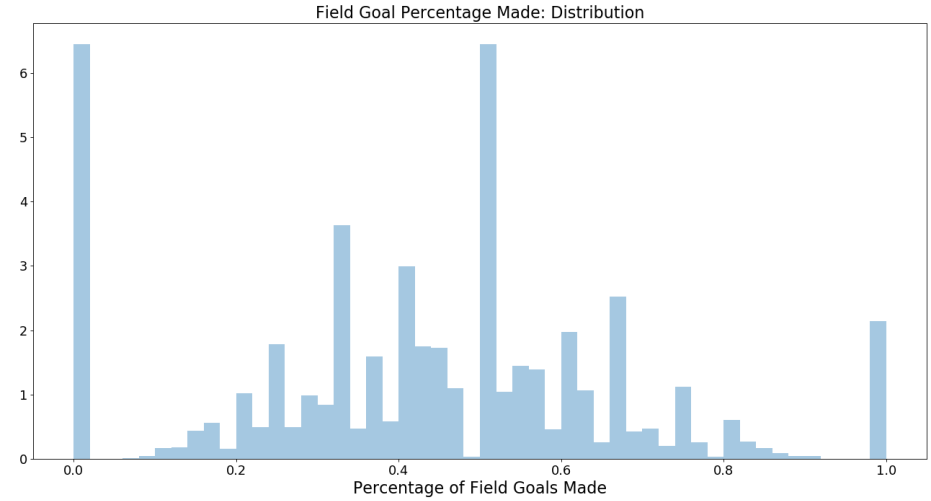
Rabiner, L. R. (1989), 'A tutorial on hidden markov models and selected applications in speech recognition', *Proceedings of the IEEE* **77**(2), 257–286.

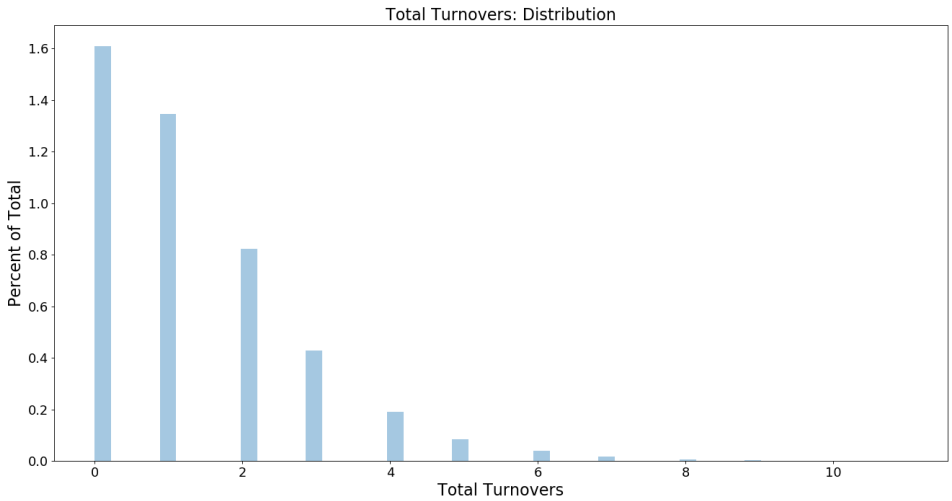
Appendix A. Overall Distributions

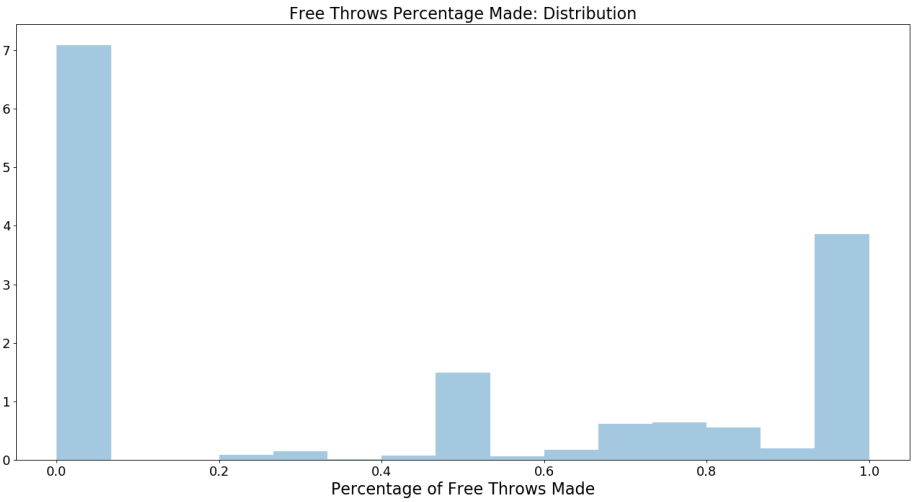




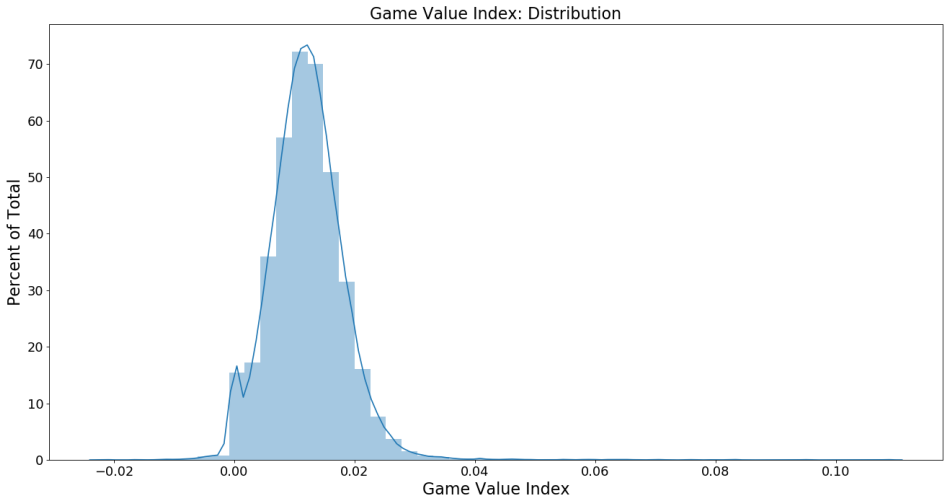

Total Steals: Distribution
2.5
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8
10
Total Steals


Field Goal Percentage Made: Distribution
6
5
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0
Percentage of Field Goals Made


Total Turnovers: Distribution
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Percent of Total
0
2
4
6
8
10
Total Turnovers



Appendix B. Game Value Index



Appendix C. Algorithm in Python

In [ ]:

```
def forwardStep(outcomes, a, b, priors):
    # Setting up Alpha Vector with Zeros
    alpha = np.zeros((outcomes.shape[0], a.shape[0]))
    alpha_norm = np.zeros((outcomes.shape[0], a.shape[0]))

    # Initializing Alpha vectors at starting point
    alpha[0] = priors * b[:,outcomes[0]]

    # Induction Step, updating alphas
    for t in range(outcomes.shape[0]-1):
        for i in range(a.shape[0]):
            alpha[t+1,i] = b[i,outcomes[t+1]] * alpha[t].dot(a[:,i])

    return alpha

def backwardStep(outcomes,a,b):
    # Initialize Betas
    beta = np.ones((outcomes.shape[0], a.shape[0]))

    # Backward Induction
    for t in range(outcomes.shape[0] -2,-1,-1):
        for i in range(a.shape[0]):
            beta[t,i] = (beta[t+1] * b[:,outcomes[t+1]]) @ a[i,:]

    return beta

def baumWelch(outcomes,a,b,priors, max_iters =100):
    T = len(outcomes)
    # Repated steps
    for r in range(max_iters):

        # Get Alphas and Betas at current step
        alpha = forwardStep(outcomes,a,b,priors)
        beta = backwardStep(outcomes,a,b)

        # Initializing xi matrix
        xi = np.zeros((a.shape[0],a.shape[0],T-1))

        # Filling XI up
        for t in range(T-1):
            # Denominator is always the same
            denominator = (alpha)[t,:] @ a * b[:, outcomes[t + 1]] @ beta[t + 1, :]
            for i in range(a.shape[0]):

                numerator = alpha[t, i] * a[i, :] * b[:, outcomes[t + 1]].T * beta[t + 1, :].T
                xi[i, :, t] = numerator / denominator

        # We define Gamma as the sum of the corresponding Xis, see tutorial by Rabiner
        gamma = np.sum(xi, axis = 1)

        # Updating Transition probabilities
        a = np.sum(xi, 2) / np.sum(gamma, axis=1).reshape((-1, 1))

        # Add additional T'th element in gamma
        gamma = np.hstack((gamma, np.sum(xi[:, :, T - 2], axis=0).reshape((-1, 1))))

        # Updating emission probabilities
        K = b.shape[1]
        denominator = np.sum(gamma, axis=1)
        for l in range(K):
            # Calculate each b for given state l
             b[:, l] = np.sum(gamma[:, outcomes == l], axis=1)

        # Divide by the common denominator
        b = np.divide(b, denominator.reshape((-1, 1)))

    # Return transistion and emission probabilities
    return a , b
```

Appendix D. Top Players

Transition matrix of top players in the NBA 2017 - 2018 season				
Player Name	Cold Cold	Cold Hot	Hot Cold	Hot Hot
Karl-Anthony Towns	0.112	0.888	0.389	0.611
Anthony Davis	0.144	0.856	0.346	0.654
LeBron James	0.141	0.859	0.345	0.655
Hassan Whiteside	0.070	0.930	0.399	0.601
Joel Embiid	0.124	0.876	0.352	0.648
DeMarcus Cousins	0.144	0.856	0.345	0.655
Russell Westbrook	0.045	0.955	0.433	0.567
Stephen Curry	0.008	0.992	0.354	0.646
James Harden	0.011	0.989	0.493	0.507

Emision matrix of top players in the NBA 2017 - 2018 season						
Player Name	Bad Cold	Average Cold	Good Cold	Bad Hot	Average Hot	Good Hot
Karl-Anthony Towns	0.040	0.455	0.505	0	0.065	0.935
Anthony Davis	0	0.131	0.869	0	0.115	0.885
LeBron James	0	0.117	0.883	0	0.107	0.893
Hassan Whiteside	0.062	0.354	0.584	0	0.061	0.939
Joel Embiid	0	0.185	0.815	0	0.015	0.985
DeMarcus Cousins	0	0.117	0.883	0	0.099	0.901
Russell Westbrook	0	0.282	0.718	0	0	1
Stephen Curry	0	0.451	0.549	0	0	1
James Harden	0	0.126	0.874	0	0	1

Transition matrix of top players in the NBA 2018 - 2019 season				
Player Name	Cold Cold	Cold Hot	Hot Cold	Hot Hot
Giannis Antetokounmpo	0.118	0.882	0.353	0.647
Anthony Davis	0.129	0.871	0.359	0.641
LeBron James	0.137	0.863	0.341	0.659
Joel Embiid	0.143	0.857	0.356	0.644
Russell Westbrook	0.146	0.854	0.346	0.654
Kevin Durant	0.060	0.940	0.440	0.560
Stephen Curry	0.000	1.000	0.499	0.501
James Harden	0.011	0.989	0.459	0.541

Emision matrix of top players in the NBA 2018 - 2019 season						
Player Name	Bad Cold	Average Cold	Good Cold	Bad Hot	Average Hot	Good Hot
Giannis Antetokounmpo	0	0.124	0.876	0	0.014	0.986
Anthony Davis	0.062	0.078	0.861	0	0.071	0.929
LeBron James	0	0.082	0.918	0	0.075	0.925
Joel Embiid	0	0.171	0.829	0	0.094	0.906
Russell Westbrook	0	0.136	0.864	0	0.098	0.902
Kevin Durant	0.046	0.420	0.534	0	0.039	0.961
Stephen Curry	0.050	0.527	0.423	0	0.061	0.939
James Harden	0	0.184	0.816	0	0	1

Appendix E. Bad Players

Transition matrix of bad players in the NBA 2017 - 2018 season				
Player Name	Cold Cold	Cold Hot	Hot Cold	Hot Hot
Wesley Johnson	0.799	0.201	1	0
Alex Abrines	0.574	0.426	1	0
Solomon Hill	0.846	0.154	1	0
Wilson Chandler	0.641	0.359	1	0
Skal Labissiere	0.648	0.352	0.852	0.149

Emision matrix of bad players in the NBA 2017 - 2018 season						
Player Name	Bad Cold	Average Cold	Good Cold	Bad Hot	Average Hot	Good Hot
Wesley Johnson	0.901	0	0.099	0.002	0.995	0.004
Alex Abrines	1	0	0	0.363	0.318	0.318
Solomon Hill	0.899	0.101	0	0.006	0.821	0.173
Wilson Chandler	0.851	0.149	0	0.008	0.733	0.260
Skal Labissiere	0.817	0.019	0.164	0.006	0.994	0