Comparing Hard and Soft Clustering using NBA Shot Charts

Eric Gerber and Kara Keller

Department of Statistics, Purdue University.
250 N. University Street, West Lafayette, IN 47907-2066, USA

November 30, 2015

Abstract

Shot selection in NBA games is a vital component of both player and team success. Different locations on the NBA court provide different probabilities of success for making shots, and profiling a team, or player's, shot selection trend can help dictate strategy for both the offensive and defensive teams. With access to spatial data of individual player shooting patterns, we can use clustering algorithms to begin answering questions concerning player's shooting tendencies, grouping similar players, and whether shooting clusters of teams correspond to team level success.

Keywords: K-means, EM Algorithm, Clustering, NBA, shot charts.

1. Introduction

All points tallied in the sport of basketball are dependent on the location on the playing surface from which a shot was made. In the highest level of basketball, the National Basketball Association (NBA), balls shot from beyond an arc 7.24 meters away from the target basket are worth three points, while balls shot from within the arc are worth two. This framework for points scoring has often thus been fertile ground for examining which shooting strategies are superior in the sport. Shots taken closer to the basket are considered higher percentage shots, yet if a player can hit from beyond the arc with only greater than a two-thirds higher probability than inside, the optimal shot (according to expected value) would be the three-point shot. This question, of strategy, has led to many in-depth analysis of NBA

shooting charts, describing teams and players ability to make shots at certain locations on the basketball court. Using the K-means and the Expectation-Maximization (EM) algorithms, this project seeks to determine how these perform in clustering teams and players by shot chart, as well as determine if any light can be shed on optimal strategy for shot selection. The differences in the algorithms, with K-means as a Maximization-Maximization in contrast to EM, could also provide insight into which method may be more useful for analyzing data of this spatial nature.

2. Data

The data consist of play-by-play accounts of all regular season NBA games from the 2009-2010 season, collected from the website basketballgeek.com. The data utilized consist of the team, player, and x-y coordinates of all made shots over the course of the year, where the x-y coordinates correspond to a 51 by 51 grid representing the half court of the basketball court. For the purposes of the project, two subsets will be used; one containing all shots for the thirty NBA teams, and one containing all shots from the 440 players who attempted a shot in the 2009-2010 season. Overall team performance and overall player performance data was also collected from the website basketball-reference.com, in order to investigate underlying traits of the produced team clusters. Analyzing team clusters might allow us to determine which shooting strategies correspond to better team performance (in terms of offensive output, or team wins) or could potentially be able to separate teams based on strengths (teams with good three-point shooters vs. post-players). Clusters of players, conversely, might illustrate positional differences for shot selection, or cluster on the import of players to their teams offence.

Final analysis is conducted on vectors of length 2601 (51 x 51) for all teams and all players, representing their corresponding shot chart. An example shot chart is presented in Figure 1. While the theory is explained in more detail in the next section, we note now that K-means analysis is performed on three versions of the data, while the EM algorithm is performed on one version. The original data consist of counts of the number of shots made at a spot (x-y coordinate) over the course of the year for the team/player. We perform K-means clustering on this, as well as two transformed versions; one where each element of the vector is a probability [0,1] of the team/player

having made a shot at that court location that year (since the NBA season has 81 games, produced by forcing the maximum number of shots made to be 81 then dividing by 81), and another version where we treat each spot as a Bernoulli $\{0, 1\}$ where one corresponds to having a more than 2.5 probability of having made a shot in a game that year at that spot. The EM algorithm, due to some limitations discussed later, is only used for this last format.

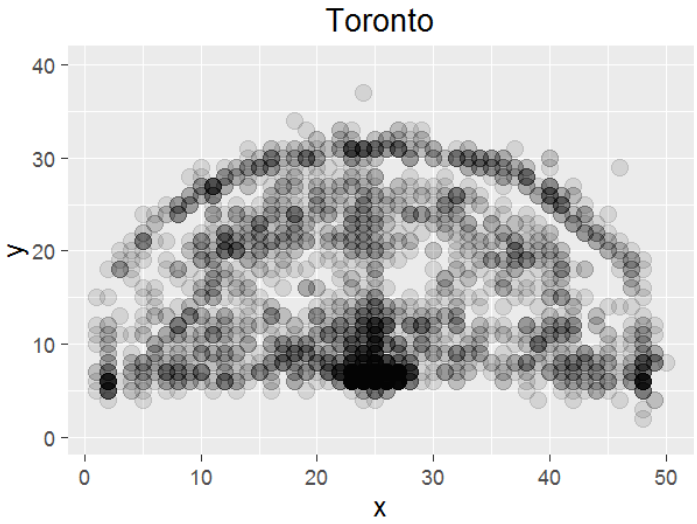

Figure 1: Toronto Raptors shot chart for 2009-2010 NBA Season. All teams had similarly large values directly beneath the basket; yet some significantly differed in pattern farther away

3. Methods

3.1. K-Means Algorithm

The K-Means algorithm can also be seen as a Maximization-Maximization algorithm. After initializing some arbitrary cluster parameters, based on a predetermined number of clusters, the K-Means algorithm iteratively assigns observations in the data set to the cluster whose parameter the observation is closest to via some distance formula (hard thresholding), and then recalculates the cluster parameters based on those assignments. As such, it will almost universally be the case where the K-Means algorithm converges to the point where cluster assignments do not change, and no other stopping criteria are required. Also, since no distributional assumptions are made on the

cluster parameters, K-Means may be run on many different kinds of data. In the context of this project, K-Means assumes, based on the number of clusters, that all teams/players belong to some cluster with true parameter θ_k, which would be the true shot distribution. After initializing θ arbitrarily, one step of the K-Means algorithm would work thusly, using the Euclidean distance formula as our distance measure:

- Calculate distances from each observation to each cluster parameter: $(\theta_1, ..., \theta_k)$,
$$D_{ij} = \sqrt{(X_i - \theta_j)^2}$$
where X_i is the observation vector for observation i and θ_j is the parameter vector for cluster j.

- Assign observations to closest cluster, based on $C_i = min_j D_{ij}$, where C_i is the cluster assignment of observation i.

- Recalculate cluster parameters θ by averaging all observations within each cluster: $\theta_j = \Sigma_i \frac{X_i}{N_j}$, such that $X_i \in$ cluster j, where N_j is the number of observations in cluster j

3.2. EM Algorithm

In contrast to K-Means, the EM algorithm is a soft thresholding algorithm which assumes the final clustering depends on some latent variables with an assumed distribution. Initially, for this project, two different EM algorithms were considered. The first is an EM algorithm for a mixture of Bernoulli random variables, based on the data formatted $\{0, 1\}$ where 1 is having a more than 2.5 probability of having made a shot in a game that year at a spot. The second model considered was an EM algorithm for a mixture of Poisson random variables for use on the observed count vectors. We believe the disproportionate number of zero counts (court locations where a shot was never made) caused this algorithm to fail, and perhaps a different distributional assumption would have been more appropriate. The count data did not resemble either Gaussian or Poisson random variables, though in theory we assumed the Poisson to be most appropriate. After graphically analyzing the distribution of the counts, it seemed that the many zero counts more closely resembled the Pareto distribution. It could be that the Pareto, or a zero-inflated Poisson distribution could be more robust to these zero counts.

For the mixtures of Bernoulli EM algorithm, we can once again specify the number of clusters we are interested in, which will allow us to compare the results with those from the analogous K-Means operation. Within the algorithm, we first initialize parameters μ and π, which represent the cluster means and probability of each cluster respectively. The algorithm calculates the Maximum Likelihood Estimates (MLEs) of these parameters iteratively by first calculating the expected cluster assignments of the data given the parameters, then using those cluster assignments to calculate the MLEs based on the mixtures of Bernoulli. While the EM algorithm will never converge, the log-likelihood of each subsequent iteration will never decrease, allowing us to set a threshold value to stop the algorithm once the variational lower bound (variational free energy, which is a lower bound to the log-likelihood) stops increasing above that threshold. One iteration of the EM algorithm, for both the Bernoulli and Poisson mixtures, proceeds as follows.

- Initialize parameters (μ, π): cluster means and assignment probabilities.
- E-step: Calculate expected cluster assignments of data given initialized parameters for:

  Mixtures of Bernoulli:

$$p(c_j|x_j, \mu, \pi) = \frac{\pi_k \left[\prod_{i=1}^{2601} \left(\mu_i^k\right)^{x_{ji}} \left(1-\mu_i^k\right)^{1-x_{ji}}\right]}{\Sigma_{k=1}^{K} \left[\prod_{i=1}^{2601} \left(\mu_i^k\right)^{x_{ji}} \left(1-\mu_i^k\right)^{1-x_{ji}}\right]}$$

  Mixtures of Poisson:

$$p(c_j|x_j, \lambda, \pi) = \frac{\pi_k \left[\prod_{i=1}^{2601} \frac{exp\{-\lambda_i^k\}\left(\lambda_i^k\right)^{x_{ji}}}{x_{ji}!}\right]}{\Sigma_{k=1}^{K} \pi_k \left[\prod_{i=1}^{2601} \frac{exp\{-\lambda_i^k\}\left(\lambda_i^k\right)^{x_{ji}}}{x_{ji}!}\right]}$$

  where c_j is the cluster assignment for observation j, x_j is the observation, μ_i^k is the Bernoullie mean of location i in cluster k, λ_k^k is the analagous Poisson mean, K is the number of clusters, and π_k is the probability of being assigned to cluster k.

- M-step: Calculate the MLEs for parameters given data and expected cluster assignments

$$\hat{\pi}_k = \frac{\Sigma_{j=1}^{N} E\left[c_{jk}\right]}{N}, \hat{\lambda}_k or \hat{\mu}_k = \frac{\Sigma_{j=1}^{N} E\left[c_{jk}\right]\left(x_{ji}\right)}{\Sigma_{j=1}^{N} E\left[c_{jk}\right]}$$

  where $E\left[c_{jk}\right]$ are the expected cluster assignments: $\prod_{j=1}^{N} p(c_j|x_j, \mu, \pi)$, and N is the number of observations.

- Calculate variational free energy and compare to previous to determine convergence.

4. Results

For the analysis in R, we made sure to set the seed that was used for all implementations of the algorithms as the same (100). It should be noted that different seeds were tested, and results tended to vary quite a bit. Our overall conclusions seem to hold for all seeds, yet the specific results presented in the following section are based on the above particular seed in R.

4.1. K-Means

While we performed K-Means on all three forms of the data described in the second section, since only the Bernoulli data was used for the EM, and comparison was the goal, only the results for K-Means using the data in the Bernoulli format will be reported. We did this for K = 2 and 3 clusters for teams, and K = 2, 3, and 4 for players. For each implementation of K and teams or players, we attempted to understand what the underlying clusters could mean via simple comparison of means tests, using team and player data for the 2009-2010 season. Based on the logic of the data structure, we theorized that clustering in teams could possibly be due to volume of 3 point shots made (offensive style), points per game (offensive success), or team wins (team success). For players, we examined if clustering could be due to 3 point shots made, total field goals made, or position. To determine this last point, since both clusters and position can be treated as a factors, a simple χ^2 contingency table test was used.

For Teams, both with 2 and 3 clusters, 3 point shots made was found to be significantly different between the clusters. Using the Tukey significant

difference method, it was found that for K = 2, cluster 2 made on average 96.5 more 3 point shots than cluster 1, which was a significant difference based on the p-value of 0.026. The two cluster K-Means algorithm also produced a very convenient split, where half of the teams ended up in cluster 1 with the other half in cluster 2. With an additional cluster, cluster 1 becomes the group with the teams that score more 3 point shots than either of the other two clusters, and is narrowed down from 15 teams to 9 teams. The algorithm converged almost immediately for both implementations, due to the small number of observations (30 teams). A comparison of the two clusters created by K-Means, K = 2 for the Team data is presented in Figure 2.

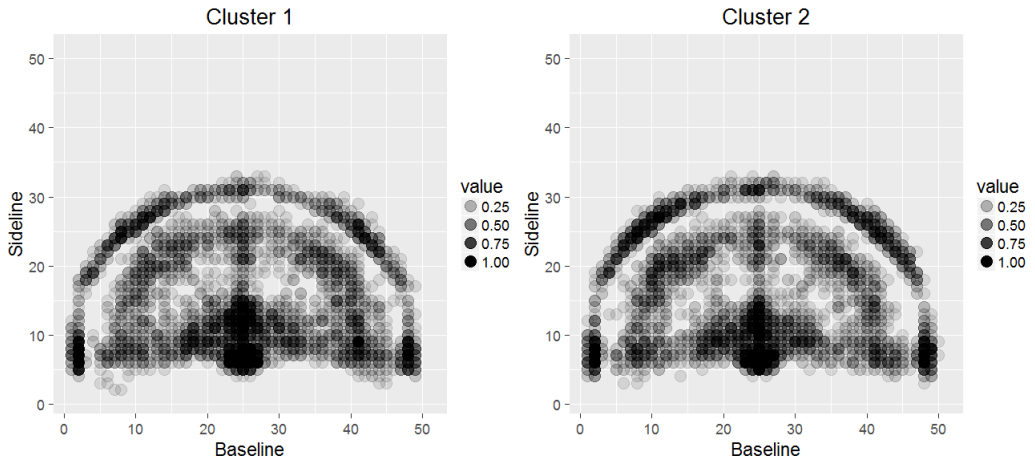

Figure 2: K-Means K=2 Cluster means for Cluster 1 (left) and Cluster 2 (right). Cluster 2 was found to score significantly more 3 point shots over the course of the year when compared to Cluster 1

Conversely, the K-Means for Player data only produced clusters that differed significantly in terms of 3 point shots made when K = 4. Otherwise, clustering appeared to be occurring based on number of field goals made, indicating perhaps a confounding factor of exposure. What this means is that number of shots made for players varies drastically depending on number of shots attempted, which is in turn affected by playing time. The difference between players who only made one shot and players who made many because they were playing every day, dominated the clustering assignments. In the future, it would be prudent when analyzing the player data to take this into account, so that perhaps some true clustering of player ability or role

could be recovered. As it is, this problem with exposure also likely impacted the non-significant results from testing the association between position and cluster. Position should have a major effect on where on a court a player is making shots, since big men rarely shoot 3 pointers and tend to stay only about the rim.

The larger number of observations for the players (440) led to a slightly longer iteration time. It took 7, 10, and 12 iterations respectively for convergence of the algorithm when K = 2, 3, and 4. However, the loss function nearly plateaued five iterations prior to the final iteration in all cases (Figure 3).

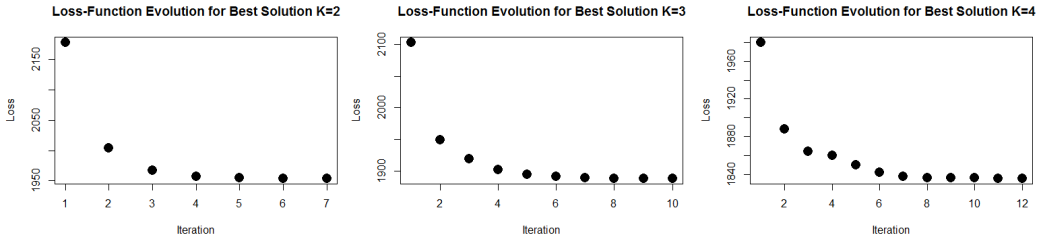

Figure 3: Evolution of Loss Function for K-Means Players K = 2, 3, 4 (left to right)

4.2. EM

The EM algorithm did not appear to find the same relationships between clusters as K-Means. For the Team data, only at K = 3 was anything significant, and that the points per game of the teams. However, in this situation, only one team made up each of cluster 2 and 3, with all other 28 teams in cluster 1. While this is less than ideal, the single team in cluster 2 was the Phoenix Suns, who led the league in PPG with 110.2, indicating that at the very least there was some impact of PPG on the clustering.

Overall, it was much the same for the Player data, in that the EM struggled to find any consistency in clustering the players based on our chosen explanatory variables. For K = 2, there were no differences found in clusters based on our criteria, and at K = 4, only field goals were significant. However, at K = 3, both 3 point shots made and field goals made were significant. While there are issues with 3 point shots being a subset of field goals, the EM algorithm did manage to find a cluster of 177 players who made significantly more of both 3 pointers and field goals than the other two clusters of players. It is

likely that this cluster is made up of the regular, every day players while the other two consist of reserve players who have less opportunities to make shots. The cluster means for players generated from the EM algorithm with K = 3 are displayed in Figure 4.

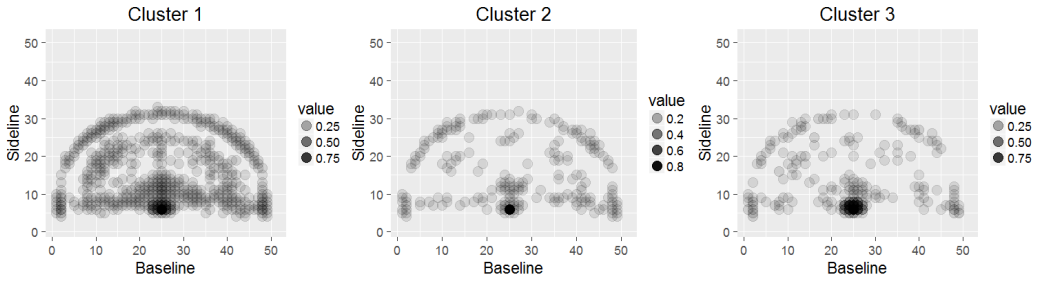

Figure 4: Cluster 1 (left), Cluster 2 (center), Cluster 3 (right) produced from the EM Algorithm on Player data. Distribution of players was, respectively, (40.2%, 52.3%, 7.5%) and saw Cluster 1 have significantly different means of 3 pointers and field goals.

As with K-Means, the EM Algorithm took nearly no time to converge when dealing with the Team data, yet took longer with the Players. Interestingly, while both K = 2 and 4 required approximately a dozen iterations each to converge, K = 3 required nearly 90. This was not necessarily the case for all random seeds attempted in R, and proved to show the wide variability in results that were possible using the EM algorithm, providing more evidence that K-Means may be a more reliable algorithm for this situation. The evolution of the variational free energy for all three implementations is presented in Figure 5.

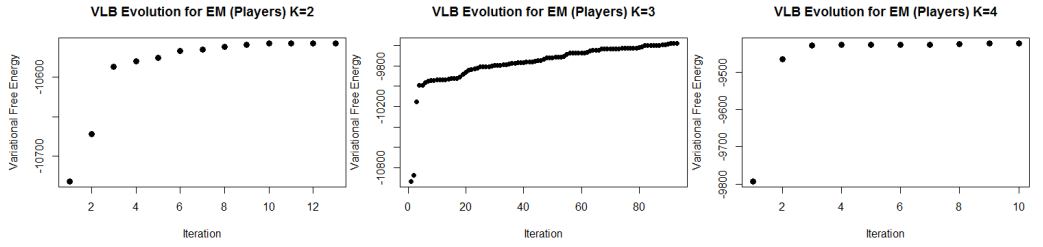

Figure 5: Variational Lower Bound Evolution for the EM Algorithm on Player data for K = 2, 3, and 4 (left to right)

5. Discussion

It seems that for this particular kind of data that the K-Means algorithm is more appropriate, likely due to the simplicity. A contributing factor could also be that the EM algorithm might perform better with a different distributional assumption. Below is a comparison of the P-values produced for the three candidate explanatory variables for both K-Means and the EM algorithm, as well as for both Team data and Player data (Tables 1, 2).

K-M Teams			
K (clusters)	3P Test	PPG Test	W Test
2	0.0255	0.0967	0.5210
3	0.0015	0.0041	0.2920
K-M Players			
K (clusters)	3P Test	PPG Test	POS Test
2	0.8320	0.0015	0.5071
3	0.9840	0.0006	0.4876
4	0.0042	0.0001	0.3355

Table 1: P-values for tests of explanatory variables effects on cluster assignments for K-means algorithm

EM Teams			
K (clusters)	3P Test	PPG Test	W Test
2	0.9600	0.4860	0.5560
3	0.1330	0.0367	0.5780
EM Players			
K (clusters)	3P Test	PPG Test	POS Test
2	0.3710	0.4980	0.2845
3	< 0.0001	< 0.0001	0.9468
4	0.7490	0.0060	0.2576

Table 2: P-values for tests of explanatory variables effects on cluster assignments for EM algorithm

Based on the criteria we expected to see clustering for, K-Means tended to provide more evidence of clustering on these. This is not to say there may be some other clustering effect that we did not take into account, and also we must keep in mind the previously mentioned possible complications of our work. Future investigations should take into account the exposure or opportunity of players made shots, instead of only the count data, and for use of the EM algorithm, another distribution other than the Bernoulli mixture model should be considered.

6. Code

All code is available via a GitHub repository:
https://github.com/gerber19/clusternbashots/.

7. Acknowledgments

We would like to acknowledge Barret Schloerke for providing the inspiration and data for this project. We would also like to thank and acknowledge Professor Rao for all his help and instruction during this course. For this project, Kara and Eric shared the coding with the K-Means and EM algorithms, respectively, though each assisted the other with their part when difficulties arose. While many more output was generated for the many implementations of both algorithms, only included are the output deemed salient to the final report.