Adaptive Bayesian Learning of Playstyles

Mirabai Smoot and Nana Anikuabe
Advised by Aline Normoyle

Abstract

Clustering is a useful tool in game analytics for understanding how players interact with the game. It allows developers to classify players based on their preferences and abilities.

Two commonly used clustering methods are:

K-means: Groups data into k clusters depending on their euclidean distance. Each sample belongs to a single cluster.

Gaussian Mixture Model (GMM): Also groups data into k clusters based on their distance. Each sample can have partial membership in multiple clusters.

Our research focuses on the implementation of a Bayesian clustering algorithm that iteratively chooses the number of clusters that best fits the data. Also, it clusters based on in-game user behavior rather than directly on the outcome.

Methods

We primarily tested the algorithm against simulated datasets, which allowed us to test whether the algorithm was implemented correctly. We also acquired and ran preliminary tests on datasets from Battlefield 3 and Dota 2.

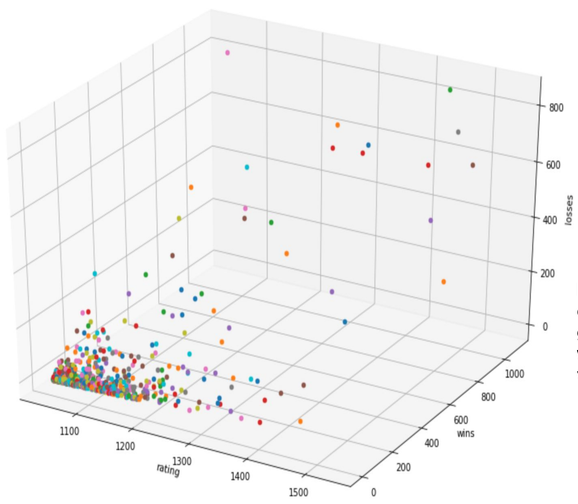


Fig 1 - A Dota 2 exploratory analysis graph showing the rank, wins, and losses for various players.

Algorithm

The algorithm fits a linear regression model to each player's score as a function of the player's rank, selected character, and various in-game decisions. The algorithm them clusters based on the player-specific weights.

1. Initialize clusters with k-means

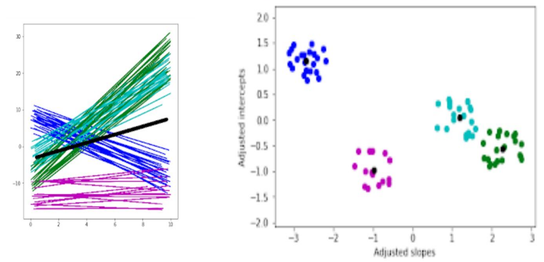


Fig 2 - Player data from a toy dataset expressed at left as the original data and at right as points determined by the line's slope and intercept. The colors indicate the initial k-means clusters.

2. Compute the overall multivariate normal distribution.

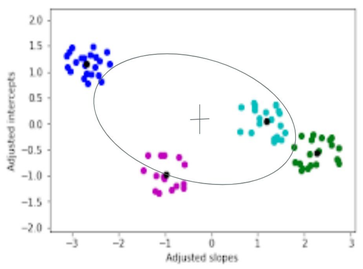


Fig 3 - The sample data from Figure 2, with an ellipse to indicate the overall center and variance.

3. Compute Root Mean Squared Error (RMSE) based on cluster centers, a matrix of scores (Y_i), and a matrix of projected scores ($\hat{Y}_{jk}$).

$$\text{RMSE} = \sqrt{\frac{\sum(Y_{res} - \hat{Y})^2}{M}} \qquad \text{Set } \sigma^2 = \text{RMSE}^2;$$

4. For each iteration

   a. For each player j

      i. For each cluster k

         1. Compute Q score (a measure of how well player j fits into cluster k

$$Q_k = \log(n_k) - \frac{1}{\sigma^2}\sum((Y_j - \hat{Y}_{jk})^2)$$

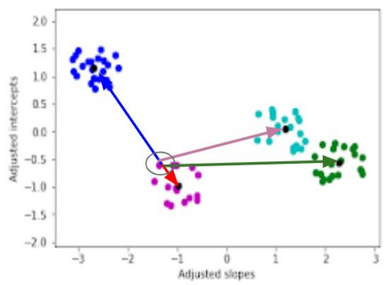


Fig 4 - The sample dataset from above, with arrows to illustrate the distance between a given point and all cluster centers.

Results

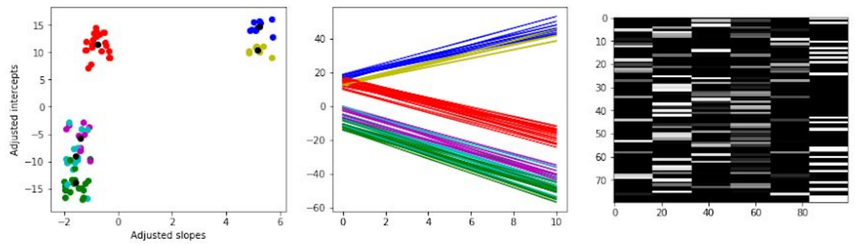
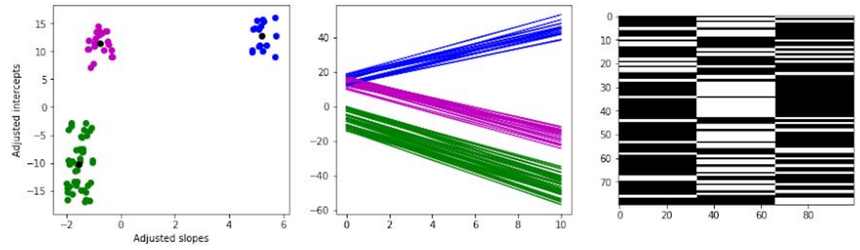


Fig 5 - Clusters after one iteration for a toy dataset. The left plot represents slope-intercept pairs of regression lines (on the middle plot) created from players' match data. Points and lines with the same color belong to the same cluster. The plot on the right includes a column for each cluster and a row for each player and is shaded based on how well the player fits into the given cluster.



The algorithm must be tested more extensively before any meaningful conclusions can be drawn about its behavior. However, the algorithm successfully clusters players whose data can be fit with a linear model.

Next Steps

Next steps include testing the Bayesian algorithm on more datasets from real games and datasets with less clearly defined clusters. By expanding and testing its functionality, we can better determine how the algorithm may be applicable on a larger scale.