Analyzing Sports Training Data with Machine Learning Techniques

Rehana Mahfuz, Zeinab Mourad, and Aly El Gamal
Purdue University, rmahfuz, zmourad, elgamala@purdue.edu

***Abstract*- In the sports industry, there has not been enough effort in analyzing the personalized monitoring data of athletes collected during training sessions. This research is an attempt to find meaningful patterns in the Purdue Women's Soccer training data that could help the coach design more efficient training sessions. We are specifically interested in studying this problem as an unsupervised learning problem. Our initial attempt is to cluster the players as well as drills into groups using k-means and spectral clustering algorithms, combined with feature transformation and reduction steps. These basic algorithms serve as a benchmark to measure performance improvements when suggesting more advanced methods. In spectral clustering, the Gaussian kernel similarity function was used, in which kernel bandwidth and the number of clusters were matched using the eigengap method. The Pearson correlation was used to eliminate highly correlated features, and Principal Components Analysis was used to find mutually orthogonal axes with maximum variance. Three features were eliminated with negligible loss in accuracy. Satisfactorily consistent clusters were identified, where by "consistent", we mean the clustering results that we get through multiple algorithms. The next step will be to evaluate the quality of the clustering, and perform semi-supervised learning after labelling the clusters.**

Index Terms- Sports data analysis, Purdue Women's Soccer, Unsupervised learning, Data Science, Spectral Clustering.

INTRODUCTION

Data mining has become a useful tool in many areas such as healthcare, financial security, marketing, manufacturing, etc. [7,8,9,10]. There is potential in analyzing sports data to better coach athletes, as is already being done in baseball in the name of Sabermetrics [6]. In this study, we delve into analyzing training data collected from the Purdue Women's Soccer Team during Spring 2016.. The specific goal of this study is to find structure in sports training data from the Purdue Women's Soccer team. It is anticipated that the findings will help the coach design better training sessions.
Nine features of athletes (shown in Table 1) such as High Metabolic Load (HML) Distance Per Minute and Average Heart Rate were recorded during the training over a span of multiple sessions, each of which was composed of one or more of the forty-eight drills. The performance of each of the twenty-two players in a particular drill during a particular session was represented in a row. Each row represents a feature vector consisting of the nine features.

The data was recorded in order of drills in sessions. Not every session had the same drills, and not every player was present for each session. A unique challenge was to handle the three components of this data set wisely: players, drills and features.

METHODS

The statistical analysis language R [1] was chosen to analyze the data after importing it from an Excel spreadsheet. First, the data had to be restructured to the expected format for data mining algorithms. Then,

this was chosen to be viewed as an unsupervised learning problem, and various techniques of clustering and dimensionality reduction were applied.

(I) Data Preprocessing

First, the data was anonymized, since working with names is tedious. Conversion keys from names to unique identifying numbers were formulated for both drills and players

Next, we had to choose how we wanted to deal with the data: according to players, or according to drills. We chose to arrange it in both ways. To arrange the data according to players, the performance of each player across all drills was averaged out and stored. The result was a nine-dimensional data set with twenty-two observations, each corresponding to a player. Likewise, to arrange the data according to drills, the performance in each drill was averaged out across all players, resulting in a nine-dimensional data set with each of the forty eight observations corresponding to a drill.

Another obstacle in fair clustering was that each feature had different ranges corresponding to the way in which it was measured. This would lead to unwanted prioritization of features which had large magnitude. So there was a need to scale the features and bring them to a common range. Initially, the method followed was that the maximum value observed in each feature was set to 100, and all the other observations were scaled as shown in (1).

$$Observation = \frac{Observation * 100}{max(feature)} \tag{1}$$

The problem with such scaling was that the maximum could be an outlier. So a second approach was to scale according to the measure of central tendency: ie, median as shown in (2).

$$Observation = \frac{Observation * 100}{median(feature)} \tag{2}$$

(II) Feature Reduction

Zeroing down to the most significant features is often helpful to mitigate the ‘Curse of Dimensionality’. While reducing the number of features considered may result in a loss of accuracy. There is a compromise between accuracy and computation time.

Two main methods were used to extract the most important features: feature selection by eliminating correlated features, and feature transformation by Principal Components Analysis.

To find correlated features, Pearson’s formula was used:

$$\frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{(n\Sigma x^2 - (\Sigma x)^2)(n\Sigma y^2 - (\Sigma y)^2)}} \tag{3}$$

Table 1: Correlation matrix between features found using Pearson's formula (darker values are smaller).

	Distance Total	Distance Per Min	High Speed Running	HML Distance	Sprints	Accelerations	Decelerations	HML Distance Per Minute	Average Heart Rate
Distance Total	1	0.816409	0.509291384	0.8801776	0.467558	0.58785068	0.758543067	0.69699156	0.3546495
Distance Per. Min	0.816408761	1	0.506864762	0.776868	0.451238	0.557046314	0.577801514	0.856229013	0.4445654
High Speed Running	0.509291384	0.506865	1	0.6807722	0.936345	0.543808345	0.34653259	0.672995175	0.2780591
HML Distance	0.880177603	0.776868	0.680772154	1	0.641398	0.729258893	0.759452618	0.888649897	0.3717938
Sprints	0.467558	0.451238	0.936345048	0.641397	1	0.498600946	0.310257041	0.630290947	0.2674412
Accelerations	0.587850688	0.557046	0.543808345	0.729258	0.498601	1	0.590234653	0.680096261	0.3074548
Decelerations	0.758543067	0.577802	0.346532459	0.759452	0.310257	0.590234653	1	0.601159871	0.2674412
HML Distance Per Minute	0.696991556	0.85622	0.672995175	0.888649	0.630291	0.680096261	0.601159871	1	0.4272912
Average Heart Rate	0.354649454	0.44456	0.278059138	0.371793	0.267441	0.307454778	0.267441202	0.427291186	1

Table 1 shows the correlation matrix. The three features eliminated were Distance Per Minute and HML Distance (both because of high correlation with Distance Total and HML Distance Per Minute) and Sprints (because of high correlation with High Speed Running). It can be roughly said that the threshold was 0.8, above which one of any correlated features is eliminated. This is a very subjective choice of features to be eliminated and is not necessarily the best choice. Other measures such as Kendall's tau [11] and Spearman rank [12] were also used, and they gave similar correlations.

In Principal Components Analysis (PCA) [3], mutually orthogonal axes with maximum variance are found. Maximizing the variance can be intuitively viewed as the opposite of the case when a feature is almost constant over observations, and does not tell us much about the data set. On the contrary, a feature with maximum variance gives a lot of information about the data set. Another subtlety to note is that once we find an axis along which variance is maximum, finding another axis having large variance can be just a matter of tilting the first axis a little. But that would give another feature which is almost the same as the first one. To exclude this possibility, PCA only finds axes that are mutually orthogonal.

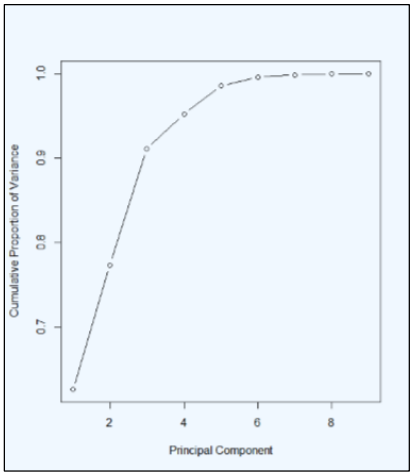

Figure 1: Cumulative proportional variance of the principal components.

Figure 1 shows the cumulative proportional variance of the nine principal components. The first six account for about 99.6% of the variance, which is why we will exclude the last three principal components in our clustering.

(III) Clustering

Two main clustering methods were used: k-means clustering and spectral clustering.

The initial attempt was to cluster the data using called k-means clustering. In this method, there is an initialization followed by expectation and maximization steps which stop with a terminating condition. First, *k* random data points are initialized as centers, where *k* is the desired number of clusters. In the expectation step, each data point is assigned to the closest center, where closeness is determined by the Euclidean distance. All points assigned to the same center form a cluster. In the maximization step, new centers are calculated as mean of all the points in that cluster. The expectation and maximization steps continue with reassignment of points to centers after reassignment of centers, till the centers converge. In other words, the terminating condition is that each center corresponds to the mean of all the points in that cluster.

The drawback encountered with this method was that random initialization of centers gave different results for each execution. One way to find out the best result was to minimize a parameter called total sums of squared distances. The sum of squared distances for a point is its squared distance from every point in the cluster apart from itself summed. The total sums of squared distances would be the sum of all individual sum of squared distances. The lesser the magnitude of this parameter is, the tighter the clusters are.

In spectral clustering [1,2], the data set is represented as a graph, with the weights of edges being calculated by the Gaussian kernel similarity function.

$$W_{i,j} = e^{\frac{-|x_i - x_j|^2}{2\sigma^2}} \tag{4}$$

The weights of the edges are stored in an affinity matrix. The laplacian of the affinity matrix is taken using the formula

$$L = D^{-\frac{1}{2}} A D^{\frac{-1}{2}} \tag{5}$$

Where *A* is the *nxn* affinity matrix, and *D* is an *nxn* diagonal matrix where *D(i,i)* is equal to the sum of the i^{th} row of *A*. Non-diagonal elements of *D* are zero.

The first *k* eigenvectors of the laplacian are stacked as rows in an *nxk* matrix called *X*, from which another *nxk* matrix *Y* is derived by renormalizing the rows of *X*.

$$Y(i,j) = \frac{X(i,j)}{\left(\sum_k^n X(i,k)^2\right)^{\frac{1}{2}}} \tag{6}$$

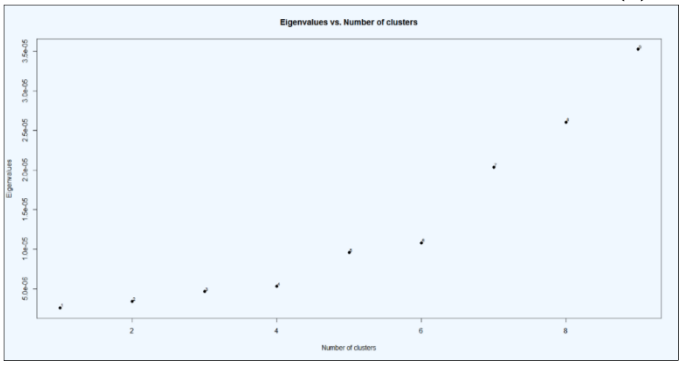

Figure 2: Eigenvalues of the laplacian. The first big jump in eigenvalues is between the fourth and the fifth, suggesting that the σ *used to find this affinity matrix would work well for four clusters.*

The challenge faced with this method is that $\boldsymbol{\sigma}$, the kernel bandwidth, being a free parameter, was hard to choose. The eigengap method, which is normally used to find the optimum number of clusters, was employed to find an appropriate $\boldsymbol{\sigma}$ for four clusters. For the eigengap method, the lower index of the first big jump in eigenvalues of a matrix gives the optimum number of clusters. In our case, since we wanted four clusters, we picked the value of $\boldsymbol{\sigma}$ for which a large gap between the fourth and fifth eigenvalues is observed, as shown in Figure 2.

(IV) Matching Clusters across Results

After using two methods of dimensionality reduction and two methods of clustering, we were left with four different clustering results for each data set. Corresponding clusters across results were then found.

First, all possibilities of matching of clusters between two results were identified. Suppose the clusters from the first result are named *{a1, a2, …, aN}* and clusters from the second results are named *{b1, b2, …, bN}*, where *N* is the number of clusters (which is four in this case). Then there are *N* different possible clusters from the second result to pair with *a1*, *N-1* different possible clusters to pair with *a2*, and so on. There will only be one possible cluster from result 2 to pair with *aN*. Thus we see that there are *N!* different possible pairings of clusters from the two results.

Once we are able to identify and generate all possible matchings, we compute a "score" for each matching. A "difference" is calculated for each of the *N* pairs in the matching. The score of a matching is the sum of those *N* differences. To calculate the difference for a pair, it is initialized to zero. Then the first cluster in the pair is skimmed. Every time there is an element in the first cluster that is not found in the second cluster, the difference is increased by one. Similarly, the second cluster is skimmed, and every time there is an element in the second cluster which is not found in the first cluster, the difference is increased by one. The maximum difference for a pair is the sum of the number of elements in each cluster. So the maximum score for a matching would be two times the number of points in that data set. The lesser the score is, the better the matching between the two clustering results.

RESULTS

Figure 3 illustrates one of the best matchings of clustering for both players and drills.

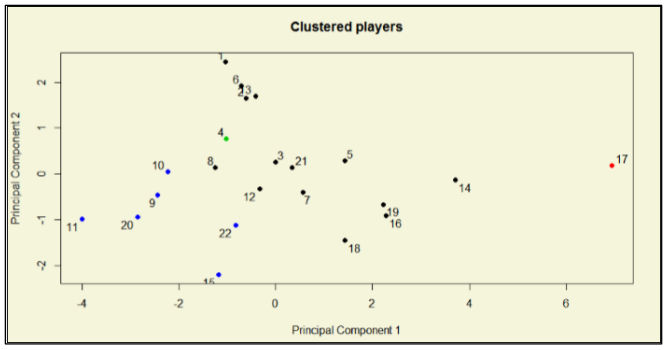

Figure 3(a): Clustered players

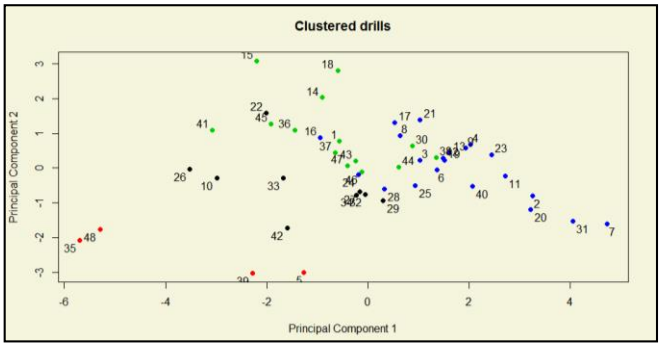

Figure 3(b): Clustered drills

The first two principal components were chosen for the plot because those axes have maximum variance which will help to see the clusters clearly. It would otherwise be hard to select two of the recorded features for plotting.

CONCLUSION AND DISCUSSION

In this paper, we have described an unsupervised learning approach to analyze data collected from the Purdue Women's Soccer team's training sessions during Spring 2015. Dimensionality reduction methods such as eliminating correlated features and using the first few principal components were employed. Then, k-means clustering and spectral clustering were performed. Finally, matching clusters were found.

The clustering result is currently very subjective, depending on a lot of factors such as methods of dimensionality reduction, clustering methods, and choice of $\boldsymbol{\sigma}$ in spectral clustering. It would be wise to

evaluate the quality of clustering. One possible way to do this would be to use the Average Silhouette Width [5], where the degree of belonging of each point to its cluster, or to another cluster is determined.

In the long run, we hope to be able to label the players as well as drills, and then perhaps perform semi-supervised clustering with future data. A standardized method of data analysis in soccer could be eventually developed, just like Sabermetrics is a standardized method for dealing with baseball data.

ACKNOWLEDGEMENTS

We are pleased to thank the Purdue SURF Program for funding this project. We would also like to recognize Jampani Dwaraknath Reddy's help and support.

REFERENCES

[1] Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems,* Advances in Neural Information Processing Systems, 2002.

[2] Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing,17*(4), 395-416.

[3] Shlens, J. (2014). A Tutorial on Principal Component Analysis.

[4] R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[5] Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journa l of Computational and Applied Mathematics, 20*, 53-65.

[6] Costa, G. (1991). Baseball in the University Classroom: Sabermetrics.*Education, 112*(2), 273.

[7] Milley, Anne. (2000). Healthcare and Data Mining.(Technology Information).*Health Management Technology , 21*(8), 44.

[8] P. Seemakurthi, S. Zhang, and Y. Qi. Detection of fraudulent financial reports with machine learning techniques. In 2015 Systems and Information Engineering Design Symposium, pages 358–361. IEEE, 4 2015.

[9] G. Cui, M. L. Wong, and H.-K. Lui. Machine learning for direct marketing response models: Bayesian networks with evolutionary programming. Management Science, 52(4):597–612, 4 2006.

[10] S. Fern´andez, R. Aler, and D. Borrajo. Machine Learning in Hybrid Hierarchical and Partial-Order Planners fo r Manufacturing Domains. Applied Artificial Intelligence: An International Journal, 19(8):783–809, 2005.

[11] Romdhani, Lakhal-Chaieb, & Rivest. (2014). Kendall's tau for hierarchical data.*Journal of Multivariate Analys is, 128*, 210-225.

[12] Sedgwick, P. (2014). Spearman's rank correlation coefficient. *BMJ : British Medical Journal, 349*, BMJ : Britis h Medical Journal, 28 November 2014, Vol.349.