A Comparative Study of Clustering Techniques and their Applications

Arpan Banerjee
UFID: 9359-9083
Pattern Recognition, CISE
University of Florida

***Abstract*—Clustering is an important method in several areas like computational biology, image processing and computer science for the detection of patterns and visualization of similarities in unstructured or unlabeled data. In this paper, three separate clustering algorithms are studied; k-means, hierarchical clustering and Gaussian mixture models, along with descriptions of their implementation. These algorithms are compared on several synthetic toy datasets which highlight the pros and cons of each algorithm and the different applications that are suitable for each algorithm. Finally I perform k-means clustering on a FIFA 19 soccer database and analyze the results.**

***Index Terms*—Data clustering, K-Means, Hierarchical clustering, Expectation Maximization (EM), Gaussian Mixture Models (GMM)**

I. Introduction

Data analysis finds use in almost every major industry during the design phase or for monitoring and optimizing the performance. More often than not, the data is unlabelled which stops us from using classic regression techniques. Cluster analysis can be used in unsupervised learning to separate the data into clusters based on similarity. There are several definition of clustering, but one of the most widely accepted definitions is as follows [1]:

> "Patterns within the same cluster are more similar to each other than they are to patterns belonging to a different cluster."

It is important to distinguish the task of clustering (unsupervised classification) from supervised classification. In the latter, we are tasked with learning labelled patterns and the problem is to label a new pattern. The model tries to learn the descriptions of different classes which are then used to classify the test data. However, in the case of clustering, the problem is to group unlabelled patterns into meaningful clusters. Here, the clusters and cluster labels are entirely data-driven and it is up to us to analyze the results and draw conclusions.

Clustering is commonly utilized for exploratory pattern analysis in the fields of genetics and computational biology, document analysis, customer segmentation etc. Often, there is very little prior information available about the data in terms of a distribution and the user must make as few assumptions as possible so the analysis remains valid for new data.

II. Related Work

There are several types of clustering algorithms based on the approach taken:

- *Partition clustering*: It simply divides the data into k disjoint subsets or clusters such that each data point belongs to exactly one cluster.
- *Hierarchical clustering*: It generates a tree-based representation of the data called a dendrogram which tells us the relation between clusters. This allows the user to choose clusters by cutting off the dendrogram at the desired level of similarity instead of providing the number of clusters directly.
- *Model based clustering*: This assumes that the observations are generated by sum of probability distributions. Model-based clustering can be used to estimate the parameters for each component (cluster).
- *Density based clustering*: These work by identifying densely populated clusters of data points, separated by large sparse areas. In some implementations, points outside dense regions are labelled as noise.

There is a wide range of clustering algorithms that can be used based on the application, however, I have chosen three of the most popular algorithms.

A. K-means Clustering

The k-means algorithm was first proposed by Lloyd (1957) [2] for pulse code modulation. The standard algorithm, sometimes called Lloyd's algorithm or "naive k-means", is an iterative algorithm.

Algorithm 1 k-means algorithm

```
k ← number of clusters
c_1, c_2, ... c_k ← centroid initialization
while not-converged do
    for all points x_i do
        c_j ← closest centroid(c_1, ... c_k)
        Cluster_j ← x_i
    end for
    for each cluster i ← 1 to k do
        c_i ← mean(x_j)   ∀x_j ∈ Cluster_i
    end for
end while
```

The error metric commonly used with k-means is the Within

Cluster Sum of Squares (WCSS).

$$\text{WCSS} = \sum_{j=1}^{k} \sum_{i=1}^{n_j} ||\mathbf{x}_i^{(j)} - c_j||^2$$

where k is the number of clusters, $\mathbf{x}_i^{(j)}$ is the i^{th} data point belonging to the j^{th} cluster and c_j is the centroid of the j^{th} cluster.

Typically the algorithm is said to converge when there is not much change in the assignment of clusters or the WCSS. One of the major problems with k-means is that it is sensitive to the initialization of the centroids and may converge to a local optimum instead of a global optimum. Usually, the cluster centers are initialized randomly, however, a smart initialization can be made by utilizing domain knowledge and analyzing the data.

There are several variations of k-means in the literature. K-medoids or PAM allows minimizing distances for arbitrary metrics [3]. K-means++ chooses the initialization in a way which proves an upper bound for the WCSS [4]. I'm using the k-means++ initialization in my code.

B. Hierarchical clustering

The idea behind this approach is to construct a hierarchy among data in order to perform clustering. The output of this algorithm is a *dendrogram* which represents the similarity levels of different clusters.

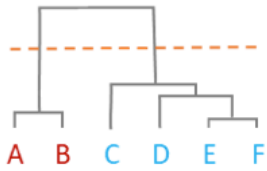

Fig. 1. Example of a dendrogram. The x-axis represents data points and the y-axis represents dissimilarity levels. The figure shows A and B are much more similar than A and C. The horizontal line is used to form clusters (two in this case) at the desired (dis)similarity level.

The algorithm can work in one of two ways:

- *Agglomerative clustering*: Each pattern begins as its own cluster. The most similar clusters are then repeatedly merged to form larger clusters.
- *Divisive clustering*: The entire data starts as one giant cluster. It is repeatedly split into two depending on the similarity metric.

In practice, agglomerative hierarchical clustering finds many more uses. Some popular options for computing the similarity metric between two clusters are:

- *Single linkage*: The distance between two clusters is the *minimum* of distances between all pairs of patterns, one from each cluster. It can cause early merging of clusters resulting in elongated cluster shapes.
- *Complete linkage*: The *maximum* distance among all pairs of patterns in the two clusters is considered. It produces compact and tightly packed clusters.
- *Ward's minimum variance*: The objective in this technique is to minimize the WCSS. In each step, two clusters are merged which result in the smallest increase of WCSS.

Algorithm 2 Hierarchical agglomerative clustering

```
distance matrix ← pairwise pattern distance
while number of clusters > 1 do
    Find two most similar clusters from distance matrix
    Merge these two clusters.
    Update the distance matrix to reflect the merge.
end while
```

The distance matrix is generated depending on the linkage method chosen. Hierarchical clustering works great for small datasets, however is inefficient with large datasets due to a space complexity of $\mathcal{O}(n^2)$ as the entire proximity matrix needs to be stored.

C. Expectation Maximization Algorithm

The idea behind this approach is assuming that the data was generated using a mixture of several distributions, where each pattern is generated by one of the components. The goal then is to estimate the parameters for each distribution. For most applications, the components are chosen to be Gaussian to model real world data and the model is known as a Gaussian Mixture Model (GMM).

The main challenge is that one usually doesn't know which data sample came from which component of the distribution. Expectation maximization solves this using an iterative approach starting with random components. The algorithm broadly consists of two steps:

1) *Expectation*: In this step, we calculate the expectation of points belonging to each component. For each pattern, the probability of being generated by each of the k distributions is computed. Let ϕ_j be the weight of the j^{th} Gaussian and μ_j, σ_j be its parameters. Then the probability that the i^{th} pattern was generated by the j^{th} component is:

$$W_i^{(j)} = \frac{\phi_j \mathcal{N}(\mathbf{x}_i; \mu_j, \sigma_j)}{\sum_{r=1}^{k} \phi_r \mathcal{N}(\mathbf{x}_i; \mu_r, \sigma_r)}$$

2) *Maximization*: Then we use the probabilities to update the parameters for the components, namely, weights, means and covariances for Gaussian components. These are set using aggregations weighted by the probabilities calculated in the expectation step as so:

$$\phi_j = \frac{1}{N} \sum_{i=1}^{N} W_i^{(j)}$$

$$\mu_j = \frac{\sum_{i=1}^{N} W_i^{(j)} \mathbf{x}_i}{\sum_{i=1}^{N} W_i^{(j)}}$$

$$\sigma_j = \frac{\sum_{i=1}^{N} W_i^{(j)} (\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T}{\sum_{i=1}^{N} W_i^{(j)}}$$

Expectation maximization is a general algorithm for estimation taking the form of different implementations such as in natural language processing, medical images and quantitative genetics. Looking at k-means closely reveals it to be a special case of the EM algorithm where it performs hard partitioning instead of calculating probabilities. The expectation step translates to assigning points to clusters and the maximization step corresponds to updating the centroids based on the assigned points.

D. Elbow method

The k-means algorithm requires the number of clusters k as an input. One of the most practical and commonly used heuristics for determining k is the elbow method. A graph of WCSS vs k is computed for a range of k-values. The WCSS values decrease rapidly at first and gradually at the end. The k corresponding to the *elbow* of the curve is chosen (or a few values around it).

E. Silhouette analysis

Another method of selecting k is using silhouette analysis. Silhouette value is a measure which quantifies how close a pattern is to patterns in the same cluster and different clusters. Its value ranges from $[-1, 1]$. A value close to $+1$ indicates that the pattern is well matched with its own cluster, a value near 0 means the pattern is near the boundary of its own cluster and a neighboring cluster and a negative value indicates that the pattern might be in the wrong cluster. However, silhouette analysis is very expensive when the dataset is large. Calculation of each silhouette value requires linear time, which results in a quadratic time complexity overall.

III. Description

A. Configuration of the three algorithms

The k-means algorithm is used with the k-means++ initialization. It is run 10 times with different initial seeds and the best output in terms of WCSS is chosen.

Hierarchical clustering uses the Euclidean distance metric for computing simmilarities, and the linkage is selected from one of single, complete or ward.

The Gaussian mixture model is initialized using the following parameters; ”full” covariance meaning each component has its own generalized covariance matrix, the weights are initialized using k-means.

I have used the three clustering algorithms mentioned above on some toy datasets [5] to compare their results. These datasets also have labels assigned so the accuracy of clustering can be measured.

B. Toy Datasets

- *S-sets*: 2 dimensional data with $N = 5000$ patterns and $k = 15$ Gaussian clusters (Fig. 2). There are 4 datasets with different degrees of cluster overlap.
- *Spiral*: 2 dimensional data with $N = 312$ patterns and $k = 3$ spiral shaped clusters (Fig. 3).

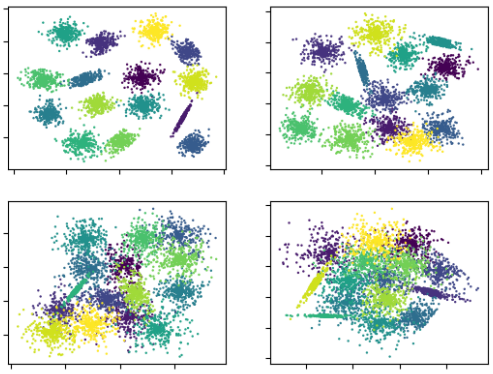

Fig. 2. S-sets: Gaussian clusters with increasing degree of overlap.

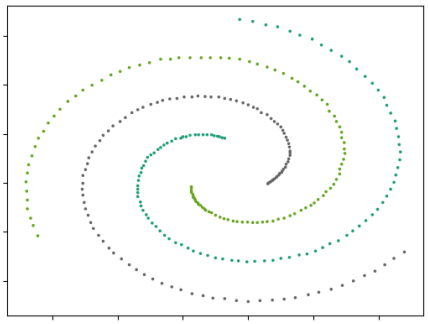

Fig. 3. Spiral dataset

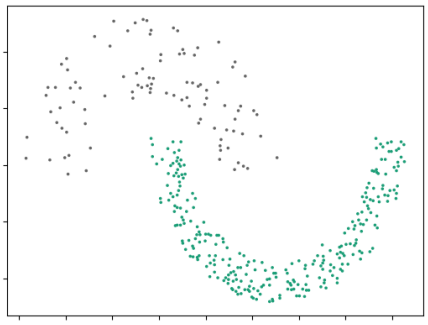

Fig. 4. Jain dataset

- *Jain*: 2 dimensional data with $N = 373$ patterns and $k = 2$ clusters in the shape of a yin-yang (Fig. 4).
- *Flame*: 2 dimensional data with $N = 240$ patterns and $k = 2$ clusters shaped like a flame (Fig. 5).

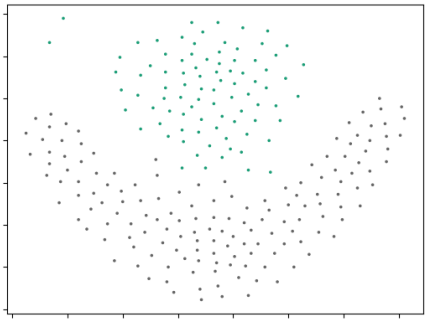

Fig. 5. Flame dataset

C. FIFA 19 Dataset

I've also chosen a FIFA 19 complete player dataset to perform clustering and analyze the results. The dataset contains detailed attributes such as age, nationality, overall, position and skill attributes for every player registered in the FIFA 19 dataset.

The clustering is expected to group similar players together. This can be utilized in different ways by sports analysts as well as managers for comparing players or shortlisting potential players for transfers who might be a good fit for their team.

IV. Evaluation

The toy datasets are clustered using each of the three algorithms and the accuracy is calculated by comparing the predicted labels against the ground truth labels. This causes an issue because even though the clustering might be as expected, the cluster labels could be permuted in any order. I solve this using a *contingency matrix*, where each predicted label is mapped to the ground truth label which has the most matches.

A. S-sets

TABLE I
Accuracy on S-sets

Dataset	k-means	Hierarchical	GMM
S-set 1	99.38%	99.22%	99.52%
S-set 2	96.96%	95.38%	97.2%
S-set 3	85.48%	82.04%	85.76%
S-set 4	79.72%	68.14%	80.44%

We can see (in Table I) that the well separated clusters in S-sets 1 and 2 are identified really well by all the three algorithms. However, k-means and GMM perform better in the problematic S-sets 3 and 4 which have a lot of overlap. For k-means, this can be attributed to the k-means++ initialization. GMMs naturally perform better with Gaussian clusters as the parameter estimation can be accurately done.

B. Spiral dataset

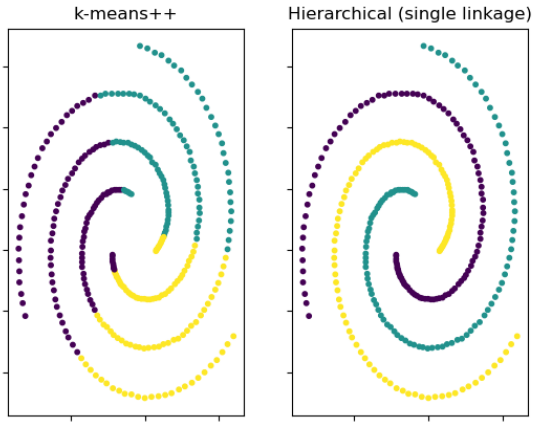

Fig. 6. Clustering on spiral dataset

K-means struggles on the spiral dataset scoring only about 27% accuracy which makes sense looking at the clustering (Fig. 6). GMM algorithm also performs similarly which is to be expected as the clusters are as far from Gaussian as could be. The hierarchical clustering algorithm shows similar results with Ward's linkage, however, when used with single linkage interestingly shows perfect clustering with 100% accuracy. This is due to the well spaced clusters which leads to the single linkage growing along the spiral lines.

C. Jain dataset

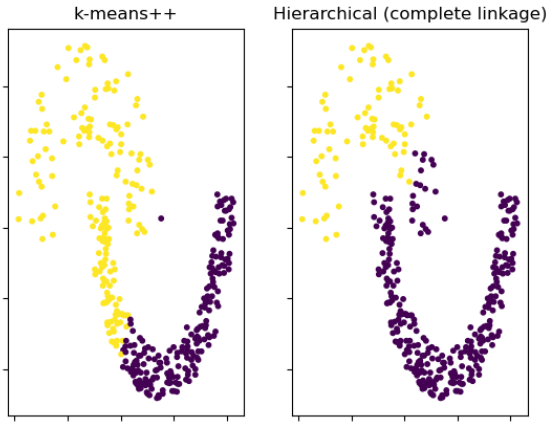

Fig. 7. Clustering on Jain dataset

Similar poor results (Fig. 7) are observed with k-means and a GMM with the Jain dataset. However, for this case hierarchical clustering used with complete linkage shows near perfect clustering with an accuracy of 94.63%. This is due to the tight grouping of the patterns. However, single linkage

doesn't perform well here as there is a lot of noise between clusters.

D. Flame dataset

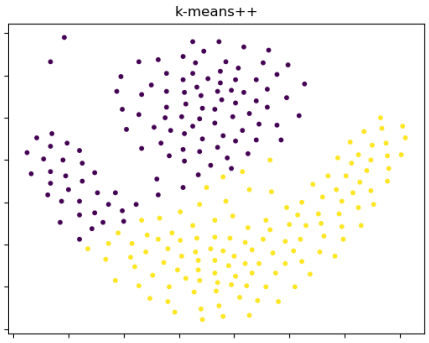

Fig. 8. Clustering on Flame dataset

This particular dataset is unable to be clustered well by either of the three algorithms (Fig. 8). In fact, to solve this and similar problems with DNA microarray data, L. Fu et al. [6] developed a novel algorithm *FLAME* which uses a fuzzy clustering method.

E. FIFA19 dataset

The first step is cleaning and preprocessing the data. I've considered only the skill attributes which are numerical (num features $= 34$). Then all of the rows with NA values are removed. Now the data is ready to be clustered.

The number of clusters must be determined, for which the elbow method is used. Looking at the graph (Fig. 9), the elbow occurs at $k = 4$ which is chosen as the number of clusters.

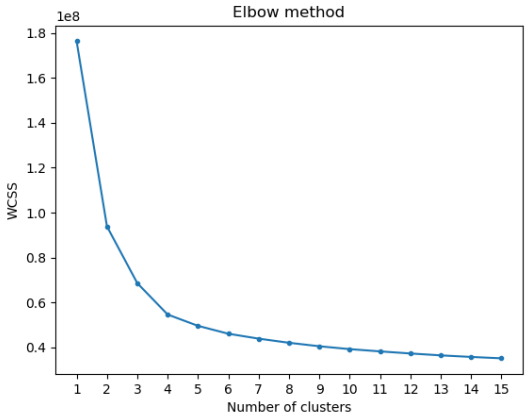

Fig. 9. WCSS vs k graph for elbow method

I also use silhouette analysis (Fig. 10) to check the quality of clustering. The vertical line represents the average silhouette value. Since all the clusters are in the positive and past the average, the quality of the clustering can be ensured.

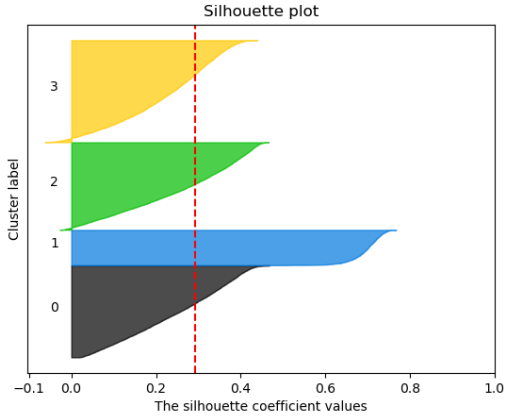

Fig. 10. Silhouette graph for each pattern

The data is then clustered using the k-means++ algorithm. Since there are no ground truth labels, an accuracy cannot be calculated. However, we can analyze these clusters to see if they are meaningful and can be utilized.

Let's look at the distinguishing features in each cluster. Distinguishing features are those skill attributes which are much higher for a particular cluster when compared to the overall mean for that feature.

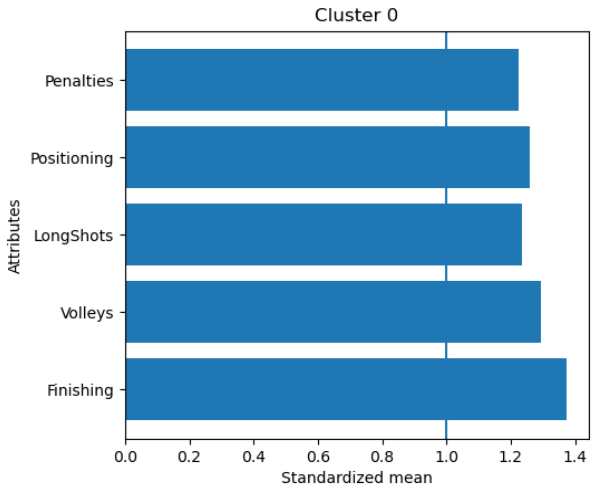

Fig. 11. Distinguishing features in cluster 0 - Attackers

The distinguishing features in the first cluster (Fig. 11) are all goal scoring attributes, hence this cluster mainly contains attacking players. The second cluster (Fig. 12) clearly corresponds to goalkeepers. The third cluster (Fig. 13) has a mix of defensive and attacking skills. This cluster corresponds to midfielders and more balanced players. The final cluster (Fig. 14) having defensive skills as the distinguishing features corresponds to defensive players.

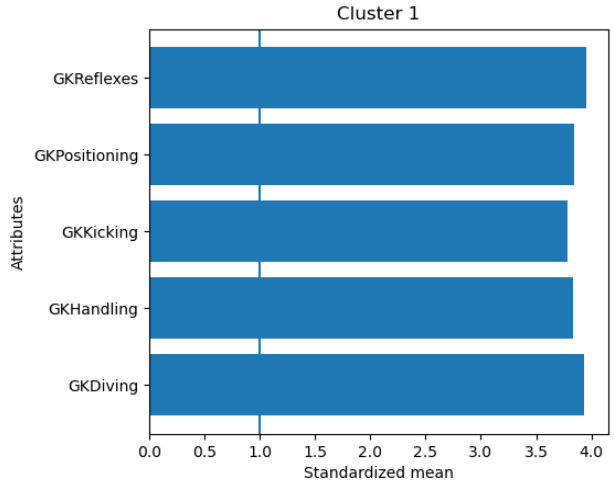

Fig. 12. Distinguishing features in cluster 1 - Goalkeepers

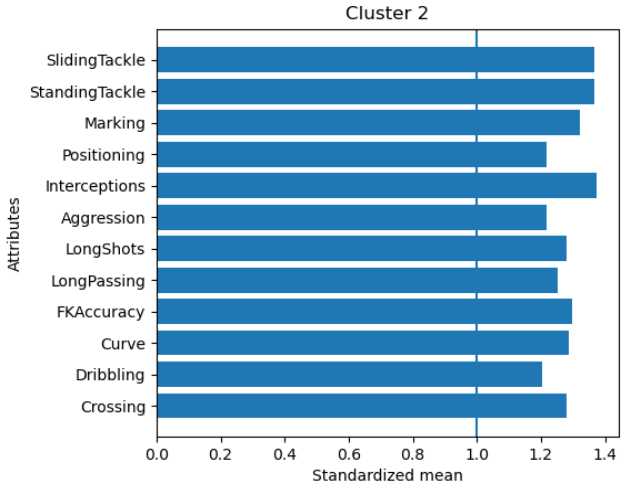

Fig. 13. Distinguishing features in cluster 2 - Midfielders

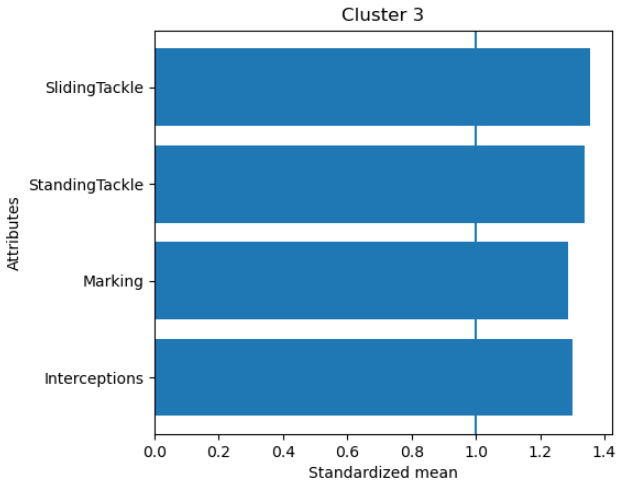

Fig. 14. Distinguishing features in cluster 3 - Defenders

V. Summary and Conclusions

The toy dataset results help in understanding the pros and cons of each algorithm. All the three algorithms show great performance on well separated data, however differences start to emerge when the clusters overlap and have non linear boundaries. Hierarchical clustering struggles with S-sets of overlapping Gaussian clusters, however it excels with single linkage on the spiral dataset and complete linkage on the Jain dataset. K-means does not perform well when there are non linear boundaries between clusters. The Kernel k-means clustering (KKC) algorithm is a nonlinear extension of k-means. In conclusion, any clustering algorithm cannot be blindly applied on the data and be expected to get good results. One must know the data inside out to be able to determine which algorithm would be suitable and also the parameters for the algorithm.

All the three algorithms give similar clustering on the FIFA19 dataset. The clustering results on the soccer database is interesting as it successfully clusters players with similar attributes and playing similar positions together. This can be extended to various applications; getting the top players from each cluster, searching for the most similar replacements for a player or even suggesting transfers based on a team's player attributes.

References

[1] A. Jain, R. Dubes, "Algorithms for clustering data", Prentice-Hall, Inc., January 1988.
[2] S. Lloyd, "Least square quantization in PCM", Bell Telephone Laboratories Paper, 1957.
[3] L. Kaufman, P.J. Rousseeuw, "Clustering by means of Medoids, in Statistical Data Analysis Based on the L_1–Norm and Related Methods", edited by Y. Dodge, North-Holland, 405–416, 1987.
[4] D. Arthur, S. Vassilvitskii, "k-means++: the advantages of careful seeding", Society for Industrial and Applied Mathematics Philadelphia, PA, USA. pp. 1027–1035, 2007.
[5] P. Fränti, S. Sieranoja, "K-means properties on six clustering benchmark datasets", 2018, http://cs.uef.fi/sipu/datasets/
[6] L. Fu, E. Medico, "FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data", BMC bioinformatics, 2007.
[7] K. Gadiya, "FIFA 19 complete player dataset", Kaggle, 2018, https://www.kaggle.com/karangadiya/fifa19