UNIVERSITAT DE BARCELONA
Facultat de Matemàtiques i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

Extracting insights from the shape of EuroLeague data using statistics and topology

Autor: Albert Ratera Gispets

Directors: Dr. Carles Casacuberta i Dr. Josep Vives
Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 20 de juny de 2021

Contents

Abstract

Mapper is a tool for topological data analysis (TDA) which was designed in 2007 in order to study shape features of high-dimensional data sets, allowing to visualize them in a more comprehensive way.

Our work is mainly practical but it also provides a theoretical background that sustains the methods used and the results obtained. We combine statistical techniques with topological data analysis to discern the structure of our data. We performed a principal component analysis, computed persistent homology, and applied Mapper to the EuroLeague data from the 2019–2020 season without taking into account the playoff matches. Our goal was to determine the underlying distribution of types of players and compare our results with those of a previous study based on data from the NBA.

Resum

Mapper és un algoritme de l'anàlisi topològica de dades (TDA) que va ser dissenyat el 2007 per extreure informació sobre les característiques latents en conjunts de dades en dimensió arbitrària, per tal de visualitzar-les de manera eficient.

El nostre treball és bàsicament pràctic, però també conté una descripció de les eines teòriques que el sustenten i tant dels mètodes utilitzats com dels resultats obtinguts. Hem combinat mètodes d'estadística amb l'anàlisi topològica de dades.

Per fer-ho, s'ha dut a terme en primer lloc una anàlisi de components principals, i a continuació s'ha calculat l'homologia persistent i s'ha aplicat l'algoritme Mapper a les dades de l'Eurolliga (la màxima competició europea de bàsquet) de la temporada 2019–2020 sense tenir en compte els partits del playoff. L'objectiu ha estat discernir la distribució per tipologies dels jugadors a partir de les dades i comparar els resultats obtinguts amb els d'un estudi previ dut a terme a la NBA.

Vull donar les gràcies als meus tutors, en Carles Casacuberta i en Josep Vives, per acompanyar-me tot aquest temps. Gràcies per la dedicació i la disponibilitat que sempre han tingut per poder reunir-nos en qualsevol moment. I sobretot, gràcies per les converses que m'han permès seguir aprenent fins al final.

Introduction

Back in 2013, the well-known and highly reputable science magazine, Nature, published, in its category of scientific reports, the article titled "*Extracting insights from the shape of complex data using topology*" [13]. This article presented the study, with recent novel techniques, of different cases where the huge dimension of the data could provoke many difficulties when trying to understand the shapes and the insights in each case. One of the cases in the article studied the National Basketball Association (NBA) shapes. The result was a very accurate classification of the league players breaking the idea of the existing five positions in the game. The results ignored the traditional five positions: *point guard*, *shooting guard*, *small forward*, *power forward*, and *center*, giving us a more precise classification defining thirteen positions.

Data processing is one of the most important areas in mathematics and engineering. Data of various kinds is being produced at an unprecedented rate and some times the large number of variables to study create point clouds with huge dimension that makes it really difficult attempting to extract sustainable information. A classical problem that scientists face when studying massive data is the aim to find common patterns that apparently the data does not show. There are many ways to discern this problem and the most common one is to make a partition of the data into clusters containing data with similar characteristics. There are many ways to cluster the data (see [1]), but it was in 2007 when Gunnar Carlsson, Facundo Mémoli and Gurjeet Singh presented an innovative method that slightly differed the traditional approach of the problem.

The method presented was called the Mapper algorithm and the captivation that the article had was the combination of conventional statistical methods with topology. This started a new way to approach the problem concerning massive data creating what now is known as *topological data analysis* (TDA). Giving the notion of a distance to the data, these novel technique allowed, unlike other techniques, to maintain the topological structure of the data. As a brief summary, the Mapper algorithm defines a map $f\colon X \to S$, where X is a point cloud and S a parameter space, this map is called a filter, then it constructs a covering $(U_\alpha)_{\alpha\in A}$, where A is called an index set, and the subsets of X_α calculating $f^{-1}(U_\alpha)$. Then the Mapper

algorithm applies a clustering algorithm to the set X_α. This technique provided a new way to visualize the data giving, at the end of the algorithm, the graph of its construction. In a certain way the desire is to visualize n-dimensional real spaces in two or three dimensions but then it emerges the question of how we can reduce the dimensionality of the data without losing information. The Mapper algorithm allows, through different filters that can be chosen, to reduce its construction and project it to a two-dimensional space. There are many possibilities and methods that face this problem; see [4].

Aligned with the notion of combining statistics and topology, Gunnar Carlsson published in 2009 an article with the title *Topology and Data* in the Bulletin of the American Mathematical Society where the bases of topological data analysis were established. If Mapper gives a qualitative analysis of the data, TDA can give quantitative results by exploring the persistence of certain homological features, informally k-dimensional holes that appear when constructing simplicial complexes born from the data set itself.

A commonly used procedure to compute the persistent homology associated to a data set is the construction of a filtration of simplicial complexes. There are various ways to proceed but in practical works the scheme that is normally used is the Vietoris-Rips scheme [2]. Then, observing the homological features calculated, it is appropriate to perform a qualitative analysis of the data studying the distribution of this feature in its persistence diagram. This procedure has been applied in sports before (see [10]), and it will be included in the methodology of the present work.

In these notes we combine statistical traditional methods with topological methods and algorithms to provide both a quantitative as well as a qualitative analysis of the shape of the EuroLeague, the most prestigious competition in European basketball. The main objective is to replicate the article mentioned at the beginning of the introduction giving an accurate classification of the EuroLeague players and, not only try to obtain the same new positions but also compare the shape of the American basketball with the European basketball. Historically, there has been a confrontation between these two playing styles, the American and the European, the first criticised for being less tactic and less defensive and the second one for being less talented. The debate generated can be almost infinite but here we will try to compare the results obtained with the results of [13].

We will first perform a principal component analysis, a traditional method commonly used in data analysis [8]. Then, we will analyse the homological, especially the 1-dimensional holes, and, in the end, we will apply the Mapper algorithm to compute the graph with the shape of the EuroLeague.

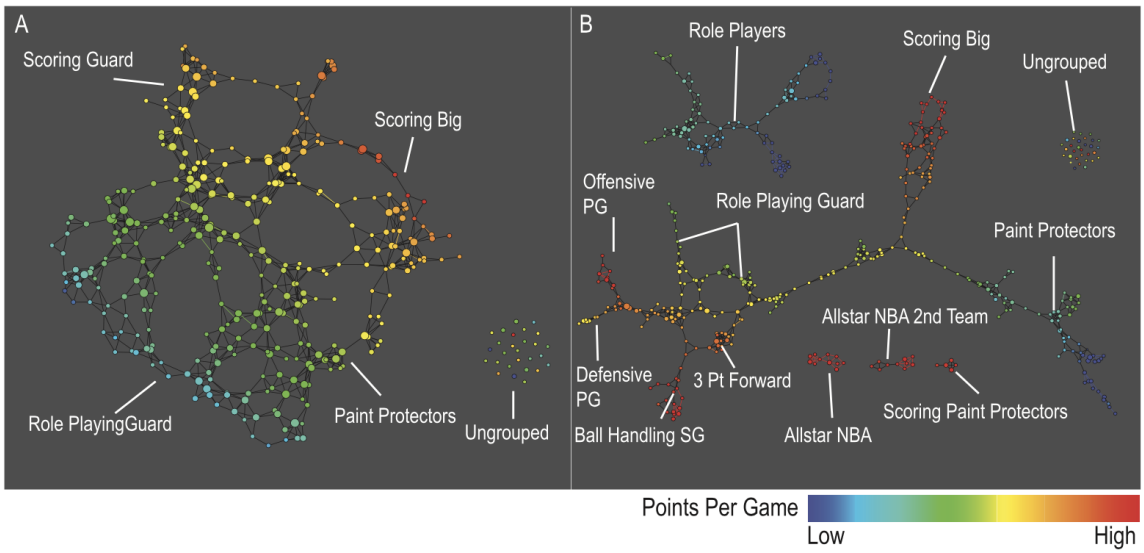


Figure 1: This figure shows the shape of the National Basketball Association (NBA) and it is the one we will try to replicate.

Data Sources

Before starting to describe the methods used, we will explain how we have obtained the data that we are going to work with. The Euroleague website offers extensive and detailed data sources from the last twenty years. It is really easy to access to this data in the *Stats* section on its website. There, it is possible to find stats from the lasts twenty seasons and one can decide to select data from the regular season or the whole season, that is the regular season plus the playoff.

For every player and every season, the website offers, stats like the number of games played, the total of points made and the number of attempted shots and its percentage of success in every field, particularly, free throws and two-point or three-point shots. It also shows the total amount of offensive and defensive rebounds, the number of committed and supported fouls, and many other stats like assists, turnovers, steals or blocks.

The data we have chosen in these notes is from the last season, the 2019–2020 one, and we only center in the regular season. It is true that the number of games played is a little less than any other season because of the stop caused by the pandemic situation, but it is enough to find profitable results.

Then we proceeded as [13] and we focus on seven stats: *points*, *rebounds*, *assists*, *steals*, *blocks*, *turnovers* and *fouls*. For every stat we got the total value and the we normalized them by dividing per the total amount of minutes played in the regular season. This way we obtained a list of two hundred and ninety eight points with seven variables per each point that we treat as a point cloud embedded in $\mathbb{R}^7$.

Once we had the data set ready to perform the statistical and topological methods with the aim to cluster the players of the Euroleague, we removed some players from the data set because the value of their components could slightly manipulate the final results. Consider, for example, a young player playing in the second team who is given the chance to play for the first team in the last match of the season and his performance consists in two fouls in three minutes. This would be reflected in the data set with a point with six components equal to zero and another one with an overstated value compared with the other players. For this reason we chose to remove every player with zeros in at least four components. This way we removed

thirty one players. The final data set consists in two hundred and sixty seven players embedded in a seven dimension real space. This is the point cloud with which we will work in every method. From now, the following chapters will be the basis that sustain the final results.

Chapter 1

Statistical Methods

In this chapter we give some statistical notions and a theoretical background for the method we apply in our work.

1.1 Principal components

Let us introduce first, for a better understanding of our work, the principal components. The principal components are the features Y, being a linear combination of the observable features X_i, with the property that they have the maximum variability. We will need the following theorem to explain how we obtain the principal components.

Theorem 1.1. *Let A and B be two covariance matrix, A positive definite or semidefinite positive, and B positive definite. The eigenvectors of A relative to B, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$,*

$$AV_i = \lambda_i B V_i, \quad V_i = (V_{1i}, \ldots, V_{ni})^{\mathrm{T}}, \quad \forall i = 1, \ldots, n$$

define n features $F_1, \ldots, F_n$

$$F_i = V_{1i}X_1 + \cdots + V_{ni}X_n, \quad \forall i = 1, \ldots, n$$

that verify:

i) *They are simultaneously orthogonal (uncorrelated) for A and B*

$$\mathrm{cov}_A(F_i, F_j) = \mathrm{cov}_B(F_i, F_j) = 0.$$

ii) *They are unitary features for B and respectively maximum variance features for A. The maximum values are precisely the eigenvalues:*

$$\mathrm{var}_B(F_i) = 1, \quad \forall i = 1, \ldots, n$$

$$\mathrm{var}_A(F_1) = \lambda_1 \geq \mathrm{var}_A(F_2) = \lambda_2 \geq \cdots \geq \mathrm{var}_A(F_n) = \lambda_n.$$

From now and then, let us suppose two covariances defined with the features. The first one is the real covariance between X_i, and the associated matrix is the covariance matrix C, that we assume it has range n. The second one is the one corresponding to the experimental metric, i.e, a distance previously defined between individuals, and the associated matrix is the identity I.

The principal components are the n features

$$Y_i = t_{1i}X_1 + \cdots + t_{ni}X_n \quad \forall i = 1, \ldots, n$$

unitary regarding to I, and with maximum variances. As a consequence of the Theorem 1.1, they verify:

1. The vectors $(t_{1i}, \ldots, t_{ni})^{\mathrm{T}}$ are orthonormal, that is,

$$\sum_{h=1}^{n} t_{hi}^2 = 1, \quad \sum_{h=1}^{n} t_{hi}t_{hj} = 0, \quad i \neq j$$

2. They are the eigenvectors of the covariance matrix C, that is, if $V_i = (t_{1i}, \ldots, t_{ni})^{\mathrm{T}}$ then $CV_i = \lambda_i V_i$ being $\lambda_1, \ldots, \lambda_n$ the eigenvalues of C.

3. The principal components $Y_1, Y_2, \ldots, Y_n$ are uncorrelated random variables which their variance is the maximum respectively

$$\mathrm{var}(Y_1) = \lambda_1 \geq \mathrm{var}(Y_2) = \lambda_2 \geq \cdots \geq \mathrm{var}(Y_n) = \lambda_n$$

The principal components are obtained diagonalizing the covariance matrix

$$C = TD_\lambda T^{\mathrm{T}} \quad (TT^{\mathrm{T}} = T^{\mathrm{T}}T = I)$$

being D_λ the diagonal matrix $D_\lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ that contains the eigenvalues of C. Then, the principal components are the linear combinations which coefficients are the columns of the orthogonal matrix

$$\begin{array}{c} \\ T = \end{array} \begin{array}{c} \begin{array}{cccc} Y_1 & Y_2 & \cdots & Y_n \end{array} \\ \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{nn} \end{pmatrix} \end{array} \begin{array}{c} \\ \begin{array}{c} X_1 \\ X_2 \\ \vdots \\ X_n \end{array} \end{array}$$

The principal components objective is to simplify the structure of our data, to explain, with a few components, most of the information contained in our variables.

1.2 Principal Component Analysis

The principal component analysis (PCA) is a commonly used method to reduce the dimensionality of the data while retaining most of the variation of the data set. Dimensionality reduction techniques allow us to make the data more understandable giving us an easier comprehension, and sometimes they let us to remove the noise. Given a point cloud $\mathbb{X} \subset \mathbb{R}^n$, PCA transforms the data set coordinate system to a new coordinate system chosen by the data itself. This change consists in rotate the axes of the data. Specifically, this rotation is determined by the data itself.

Given a number n and a set S of points in $\mathbb{R}$. This is called a *point cloud*. The cardinality of S is the number of points in S. Such point cloud will be called n-dimensional.

The first axis is rotated to cover the largest variation of the data. Then, the next axis, which has the second most variability, is chosen to be orthogonal to the first axis. We repeat this procedure for as many features as we had in the original data.We will observe that if the first few axes of our data, contain the majority of the variance, then we can ignore the rest of the axes and reduce the dimensionality.

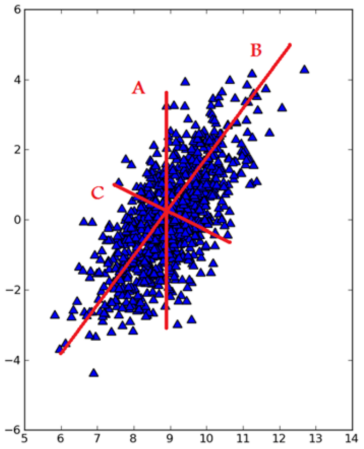


Figure 1.1: An example of how PCA changes the coordinate system of the data and align the new axes with the directions with most of the variability. In this figure B would be the axis with the first principal component direction.

Once we have the axes rotation down, the principal components will be on the same directions than the axes. The first principal component will be on the same directions as the first axis and the same for the other respectively.

1.3 Singular values

Proposition 1.2. *Let A be an $n \times n$ matrix with* $\operatorname{rank} A = r$*. If λ is an eigenvalue of the matrix $A^{\mathrm{T}}A$, then $\lambda \geq 0$.*

Proof. Let x be an eigenvector of $A^{\mathrm{T}}A$ with eigenvalue λ. We compute:

$$\|Ax\|_2 = (Ax)(Ax) = (Ax)^{\mathrm{T}}(Ax) = x^{\mathrm{T}}A^{\mathrm{T}}Ax = x^{\mathrm{T}}\lambda x = \lambda x^{\mathrm{T}}x = \lambda\|x\|_2$$

Since $\|Ax\|_2 \geq 0$ it follows that $\lambda\|x\| \geq 0$. Hence $\lambda \geq 0$. □

Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of $A^{\mathrm{T}}A$, with repetitions. We order them such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. Let $\sigma_i = \sqrt{\lambda}_i, \forall i = 1, \ldots, n$.

Definition 1.3. The values $\sigma_1, \ldots, \sigma_n$ are the singular values of A.

As $\lambda_i \geq 0, \forall i = 1, \ldots, n$, then $\sigma_i \geq 0, \forall i$.

Proposition 1.4. *The number of singular values of A is equal to the rank of the matrix.*

Proof. As the rank of a matrix equals to the number of nonzero eigenvalues, with repetitions, then the number of singular values of A equals to the rank of $A^{\mathrm{T}}A$. As $A^{\mathrm{T}}A$ and A have the same kernel, then it follows, from the rank-nullity theorem, that $A^{\mathrm{T}}A$ and A have the same rank. □

1.4 Singular Value Decomposition

The singular value decomposition (SVD) is a method that can be used to reduce the dimension of a point cloud, but it is also useful on obtaining the principal components of our data.

Theorem 1.5 (Singular Value Decomposition)**.** *Let A be an $n \times n$ real matrix. Then $A = U\Sigma V^{\top}$ where U is an $n \times n$ orthogonal matrix, V is a $p \times p$ orthogonal matrix and Σ is an $n \times p$ diagonal matrix whose elements are the singular values of the matrix A.*

Calculating the singular value decomposition consists in computing the eigenvalues and the eigenvectors of $A^{\mathrm{T}}A$ and AA^{T}. The eigenvectors of $A^{\mathrm{T}}A$ are the columns of V and the eigenvectors of AA^{T} are the columns of U. Taking this into account, the following method explains how we can reduce the the rank of the matrix, reducing then the number of singular values.

We start calculating the sum of the square of the elements in Σ, $s = \sum_{i=1}^{n} \sigma_i^2$, and the we assume an optimization threshold, for example a 90%. Then we sum the

Σ values, remember they are ordered, until we get the 90% of s. Let r the number of necessary singular values to get the percentage, then we take the r firsts columns of U, the r firsts rows of V and the $r \times r$ submatrix of Σ. Finally we calculate $\tilde{A} = U_r \Sigma_r V_r^\top$

Computing $B = U_r \Sigma_r$ we obtain the principal components of our data. Specifically, they are the columns of B.

Chapter 2

Topological methods

2.1 Mapper algorithm

Mapper is a computational method created in 2007. Its purpose relies on extracting simple descriptions and to provide a qualitative analysis of high dimensional data sets. Even when our data set has a low dimension, it is not easy to visualize or to discern its structure. Given an input point cloud $\mathbb{X} \subset \mathbb{R}^n$, Mapper reduces the point cloud into a simplicial complex $\mathcal{C}$ with far fewer points which can capture topological and geometric information at a specified resolution. The method is a combination of dimensionality reduction, clustering and graph networks techniques used to get an insightful understanding of the structure of data.

Given a set S, a *simplicial complex* with vertices S is a pair (S, Σ), where S is a finite set denoted by $S = s_0, \ldots, s_n$, and Σ is a family of non-empty subsets of S such that if $\sigma \in \Sigma$ and $\tau \subseteq \sigma$ then $\tau \in \Sigma$. The elements of S are called *vertices* of the simplicial complex and the elements of Σ are called *simplices*.

For a better comprehension of the data, Mapper allows us to select features that best discriminate the data, detect clusters that traditional methods fail to find and simplify and visualize the shape of our data set computing a graph that captures the topological structure of the data. This topological structure is coordinate-free. The Mapper construction provides a coordinatization by the creation of the simplicial complex, to which the data set maps, and not by using real valued coordinate functions. Also, this structure is invariant under small transformations. In certain way, it can represent complex figures using only a few points and edges.

As we see in [5], the method starts with a real valued function $f\colon \mathbb{X} \to \mathbb{R}$, where $\mathbb{X}$ is a point cloud, to produces a graph. This function is called a *filter function*. The method can use more than a filter function, for example one to reflect geometric properties of the data and another one to reflect properties of the data set itself, as the first principal component of the data. The space to which we produce a map is

determined by these filter functions, so we can simply modify the method to obtain a map to, for example, $\mathbb{R}$, $\mathbb{R}^2$ or the unit circle S^1.

When the created map goes to the parameter space $\mathbb{R}$, the Mapper construction produce a two dimensional simplicial complex and a map from the data set to it. This constructions conforms a stochastic version of the *Reeb graph* associated with the filter function.

Definition 2.1. Given a topological space X and a continuous function $f\colon X \to \mathbb{R}$, and assume $\sim$ to be an equivalence relation where $p \sim q$ if p and q belong to the same connected component of a single level set $f^{-1}(c)$ for some $c \in \mathbb{R}$, then the Reeb Graph is the quotien space $X/\sim$ endowed with the quotien topology.

For a further comprehension of the Reeb graphs we refer the reader to [3].

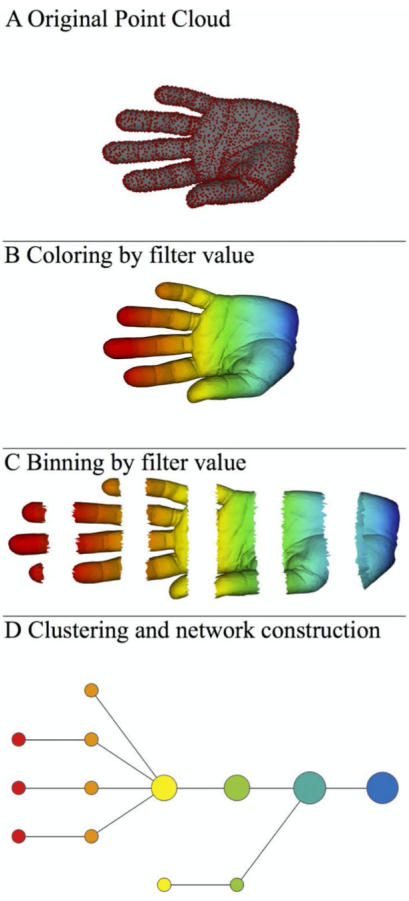


Figure 2.1: Graph generated by Mapper

Given a point cloud $\mathbb{X}$ Mapper maps it to a lower dimensional space using the filter function. Once the data has been filtrated, Mapper constructs a cover $(U_i)_{i\in I}$ of the projected space normally in the form of a set of overlapping intervals which have constant length. For each of these intervals, the algorithm applies a clustering scheme, usually the *single linkage clustering*, to cluster the points in $f^{-1}(U_i)$ into sets $C_{i,1}, \ldots, C_{i,k_i}$.

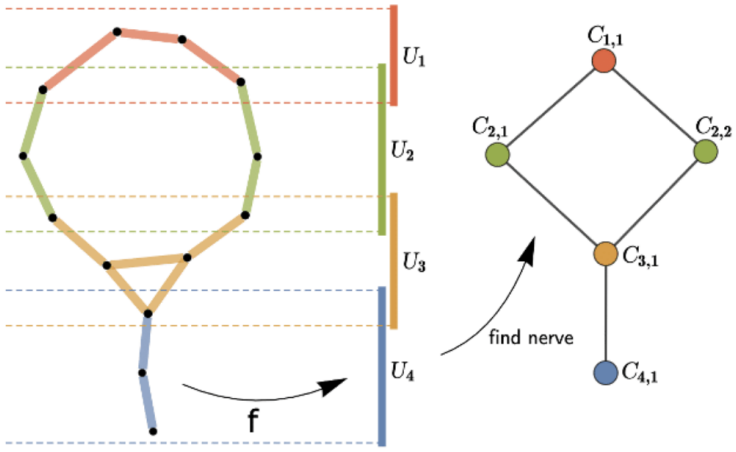


Figure 2.2: In this figure, the data is represented in black points. The filter function is a projection to the y axes and the covering is composed for four intervals. Then, each preimage of the filter function is clustered.

What it is interesting in this method, referring to its construction,it is when applying it to a point cloud $\mathbb{X}$, but first we will provide a topological background explaining some topological constructions with the purpose to let the reader have a theoretical idea of this method.

2.1.1 Topological notions

This theoretical background is based on the theoretical background provided in [5]. Let $\mathcal{U} = (U_\alpha)_{\alpha\in A}$ be a finite covering of a space X. Consider the following definitions to obtain a map from X to $N(\mathcal{U})$.

Definition 2.2. The *nerve* of the covering $\mathcal{U}$ is the simplicial complex $N(\mathcal{U})$ whose vertex set is in the indexing set A, and where a family $\alpha_0, \alpha_1, \ldots, \alpha_k$ spans a k-simplex in $N(\mathcal{U})$ if and only if $U_{\alpha_0} \cap U_{\alpha_1} \cap \cdots \cap U_{\alpha_k} \neq \emptyset$

Definition 2.3. A partition of unity subordinate to the finite covering $\mathcal{U}$ is a family of real valued functions $\{\varphi_{\alpha\in A}\}$ with the following properties:

1. $0 \leq \varphi_\alpha(x) \leq 1, \quad \forall \alpha \in A, \ \forall x \in X.$
2. $\sum_{\alpha \in A} \varphi_\alpha(x) = 1, \quad \forall x \in X.$
3. The closure of the set $\{x \in X | \ \varphi_\alpha > 0\}$ is contained in the open set U_α

Consider now the vertices $v_0, \ldots, v_k$ of a simplex. The points v in the simplex correspond in one-to-one and onto way to the set of ordered k-tuples of real numbers $x_0, \ldots, x_k$ satisfying $0 \leq x_i \leq 1$. This correspondence is the barycentric coordinatization, and the numbers x_i are the *barycentric coordinates* of the point v. Now, for any point $x \in X$, consider $T(X) \subseteq A$ the set of all α such that $x \in U_\alpha$. Consider now $p(x) \in N(\mathcal{U})$ the point in the simplex spanned by the vertices $\alpha \in T(X)$ whose barycentric coordinates are $(\varphi_{\alpha_0}(x), \varphi_{\alpha_1}(x), \ldots, \varphi_{\alpha_l}(x))$ where $\alpha_0, \alpha_1, \ldots, \alpha_l$ is an enumeration of the set $T(x)$. We can see that the map p is continuous and gives, let us say, a partial coordinatization of X, with values in the simplicial complex $N(\mathcal{U})$.

Now, given a space equipped with a continuous map $f\colon X \to Z$, where Z is a parameter space equipped with a covering $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$, whith A an indexing set. As f is continuous, the sets $f^{-1}(U_\alpha)$ form an open covering of X. We consider now, for every α the decomposition of $f^{-1}(U_\alpha)$ into its path connected components. Then we can write $f^{-1}(U_\alpha) = \bigcup_{i=1}^{j_\alpha} V(\alpha, i)$, where j_α is the number of connected components in $f^{-1}(U_\alpha)$. Finally, we call $\tilde{U}$ the covering of X obtained this way from $\mathcal{U}$.

Observe that if we have a map of coverings from $\mathcal{U} = (U_\alpha)_{\alpha \in A}$ to $\mathcal{V} = (V_\beta)_{\beta \in B}$, that is a map of sets $f\colon A \to B$ satisfying the conditions above, then we have an induced map of simplicial complexes $N(f)\colon N(\mathcal{U}) \to N(\mathcal{V})$, given on vertices by the map f. As a consequence, when we have a family of coverings $\mathcal{U}_i$, $i = 1, \ldots, n$ and maps of coverings $f_i : \mathcal{U}_i \to \mathcal{U}_{i+1}$, we obtain a diagram of simplicial complexes and simplicial maps

$$N(\mathcal{U}_0) \overset{N(f_0)}{\longrightarrow} N(\mathcal{U}_1) \overset{N(f_1)}{\longrightarrow} \cdots \overset{N(f_{n-1})}{\longrightarrow} N(\mathcal{U}_n).$$

When we have a space X equipped with a $f\colon X \to Z$, where Z is a parameter space, and we have a map of coverings $\mathcal{U} \to \mathcal{V}$, then there is a corresponding map of coverings $\tilde{\mathcal{U}} \to \tilde{\mathcal{V}}$ of the space X.

Taking into account this background, the main idea in passing from topology to statistics is that the clustering should be considered as the statistical interpretation of the geometrical idea of partitioning a space into its connected components.

2.1.2 Applying the algorithm

Given a point cloud $\mathbb{X} \subset \mathbb{R}^n$, assume that the point cloud contains N points $x \in \mathbb{X}$ and that given a filter function we know its value for the N points. Moreover, assume that we can compute the distance between every $x, y \in \mathbb{X}$ when $x \neq y$.

The algorithm starts by finding the range of the function restricted to the points $x \in \mathbb{X}$ to find a covering of the data set. To get this covering, the range of the function is divided into a set of intervals $\mathcal{S}$ which overlap. This way we obtain two parameters, a number n of intervals and a p percent of overlap. This two parameters helps us to control the resolution of the final graph.

The Mapper algorithm receives five different inputs. The first one, a distance matrix $D \in \mathcal{M}(\mathbb{R}^{n\times n})$ to compute the simplicial complex and a filter function used to produce the graph. Then, the two parameters given above and finally, the algorithm receives a clustering method that normally is the *single linkage clustering*.

Once the Mapper algorithm has been explained, we continue in the attempt to replicate the graph published in [13]. The objective is to achieve, as possible, the accuracy in order to determine the different groups of players existing in the European basketball.

2.1.3 Our method

In this section we will explain the steps we follow to get the results. We apply the same method than in [13] when they study the shape of the NBA in a topological way.

Before explaining the method, consider the following definition of a metric distance matrix. It will help to understand the first step of the algorithm applied. Note that a distance matrix of a point cloud $\mathbb{X}$ is a square matrix that contains the distances, taken pairwise, between the elements of $\mathbb{X}$.

Definition 2.4. A metric distance matrix is a square matrix that respects the metric axioms. That is, given a matrix $M = (x_{i,j})$ with $1 \leq i, j \leq N$, then:

1. The entries on the main diagonal are all zero, that is, $x_{i,i} = 0$ for all $1 \leq i \leq N$.
2. M is a non-negative matrix, $x_{i,j} > 0$ when $i \neq j$.
3. M is symmetric
4. For any i, j, $x_{i,j} < x_{i,k} + x_{k,j}$ for all $k \in N$.

First, we associate a metric to our data to obtain a metric distance matrix. In this case, the distance used is the *variance normalized Euclidean distance*. To obtain

the matrix associated to this metric, we normalized the data before computing the Euclidean distance for every pair of points in the data.

Once we have the distance matrix, we use as filter functions the first and the second principal components of our data. To get them, we perform the singular value decomposition as it is explained in the previous chapter.

The resolution parameters chosen are a total of thirty intervals and the percentage of overlapping is such that each interval overlaps with half of the adjacent intervals.

Then the clustering scheme applied is the single linkage clustering. The single linkage clustering is a commonly used hierarchical clustering method. This clustering method is predetermined in Mapper.

As it is said in [1], a hierarchical clustering is a sequence of partitions in which each partition is nested into the next partition in the sequence. The hierarchical clustering method consists in transforming a proximity matrix, a square matrix in which the entry in cell (i, j) is the distance between the items to which row i and column j correspond, into a sequence of nested partitions.

For more information about how the single linkage clustering works, we referred the reader to [1] and [12].

2.1.4 Applying Mapper in R

In these notes we have worked with R to compute the Euroleague Mapper graph to have an accurate visualization of the shape of the league in a topological sense.

We use the library *TDAmapper* from *paultpearson/TDAmapper: Analyze High-Dimensional Data Using Discrete Morse Theory* github that was created following the theoretical concepts given in the previous sections. There are two main functions in the package: $mapper1D$ and $mapper2D$. The difference between the first one and the second one is that the first receives a distance matrix $D \in \mathcal{M}(\mathbb{R}^{n\times n})$ and an other filter function and the second receives also a distance matrix but two more filter functions. In [13] the graph is generated with two filter functions but we tried to get truthful results by using and comparing both functions *mapper1D* and *mapper2D*.

To compute the final graph in R, each function receives the five inputs described before and we will see in the following chapter the consequences when varying this parameters.

Finally, we use a force network algorithm using the function *forceNetwork* from the library *networkD3* to compute an interactive plot with all the players included in different nodes that will be connected with an edge if they share at least one point. We refer the reader to [7] and [14] if there is an interest to compute a Mapper graph.

2.2 Persistent homology

In the Introduction we briefly stated how topological data analysis proceeds to study the homological features of data, or, in an informal way, the k-dimensional holes appearing when we construct a simplicial complex. In this section, we define Vietoris-Rips complexes and describe how we can compute persistent homology with such a construction. We assume that the reader is acquainted with the basic topological concepts.

Classical homology theory describes the presence and number of holes in a given topological space or in a geometric shape, in algebraic ways. One of the most useful algebraic invariants in a given dimension is the corresponding *Betti number*.

Definition 2.5. Let k be a non negative integer. The *k-th Betti number* $b_k(\mathcal{C})$ of a simplicial complex $\mathcal{C}$ is the rank of the k-th homology group $H_k(C)$.

2.2.1 Filtrations

Filtrations are the main focus on this section and later will be the basis of our analysis.

Definition 2.6. A subsimplicial complex of a simplicial complex $\mathcal{C} = (S, \Sigma)$ is a simplicial complex $\mathcal{S} = (S', V')$ where $S' \subset S$ and $\Sigma' \subset \Sigma$.

A *filtration* of a simplicial complex is an ascending sequence

$$\emptyset = S_0 \subset S_1 \subset \cdots \subset S_k = S$$

where k is a positive integer.

Let us define now the complexes used in these notes. We have chosen the *Vietoris-Rips complexes* for their optimal computational cost compared to the *Čech complexes*.

Definition 2.7. Given a finite point cloud $\mathbb{X}$, a number $\epsilon \geq 0$ and a distance d, the Vietoris-Rips complex $R_\epsilon(\mathbb{X})$ is the simplicial complex with vertex set $\mathbb{X}$ and a k-simplex spanned by a set $S = \{x_0, x_1, \ldots, x_k\} \subseteq \mathbb{X}$ if and only if $d(x_i, x_j) \leq \epsilon$ for all $i, j \geq 0$.

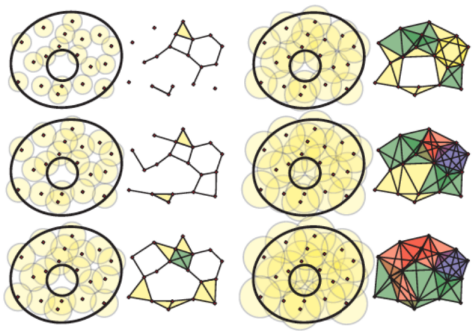


Figure 2.3: Example of a sequence of the Vietoris-Rips complexes. Figure extracted from [2].

Figure 2.3 shows what that as ϵ increases, holes appear and disappear. This is what persistent homologies mainly study. In general, regarding Definition 2.2, the Betti numbers of the Vietoris-Rips complexes $b_0(R_k)$, $b_1(R_k)$, $b_2(R_k)$ denote the number of connected components, 1-dimensional holes and 2-dimensional holes respectively.

Persistent homology focuses on studying the life time of these holes assigning them a birth and a death time ϵ.

Each Vietoris-Rips filtration determines a *barcode* depicting life times of homology generators, and a *persistence diagram* containing a point (b, d) for each homology generator in any dimension which is born at time b and vanishes at time d. Here is an example:

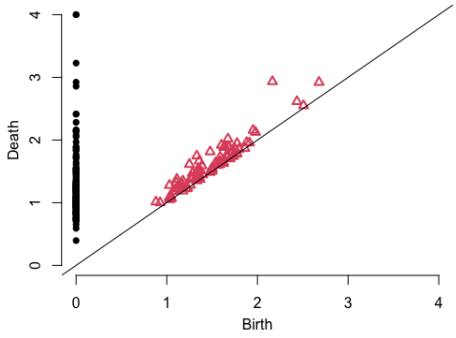


Figure 2.4: Persistence diagram of the EuroLeague data set dropping the forwards.

Chapter 3

Results and discussion

In this chapter we will present the final results obtained when applying the methods explained in the previous chapters. We will follow the same process on the first two methods. First we analyse the data set containing the whole EuroLeague players and then we will perform a partial analysis for every group following the distinction given in the EuroLeague website that classifies the players into *guards*, *forwards*, and *centers*.

We will see that the principal component analysis provide a first but lower comprehension of the data set giving us few information about existing groups of players inside the EuroLeague. Then we will see that focusing on persistent homology computation, through the Vietoris-Rips complexes, we can add significant information to show the existence of different groups in the league, especially, when doing the partial analysis.

Finally, we will see that the results obtained applying the Mapper algorithm are the ones that present a better and insightful understanding of the different groups of players that there exist in European basketball.

3.1 Principal Component Analysis

We started evaluating our data with a traditional statistical method. The chose of this method was made attending the importance it has in the treatment of massive data and because it is highly used in practice.

When this method is applied in R, one has the option to scale the data to unit variance before the analysis but we did not do it because every variable is measured with the same unit. When we made the analysis we obtained the upcoming results.

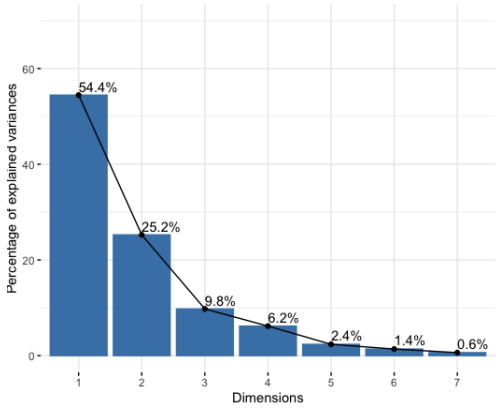


Figure 3.1: This figure shows the level of variance contained in every principal component.

We can see that with the firsts two principal components we get the 79.6% of the variance of the data. This results are fairly good since they let us remove five of the seven components with a few loss of information, concretely a loss of the 20.4% of the variance. This also let us to visualize how the players are distributed in $\mathbb{R}^2$ when we take as axes the directions of the first and the second principal components.

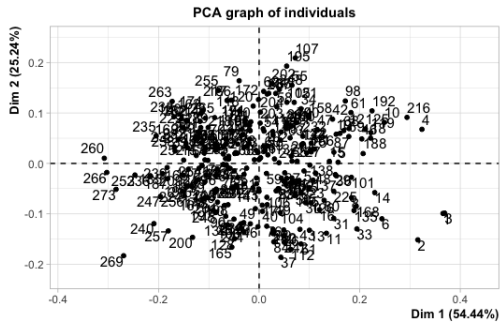
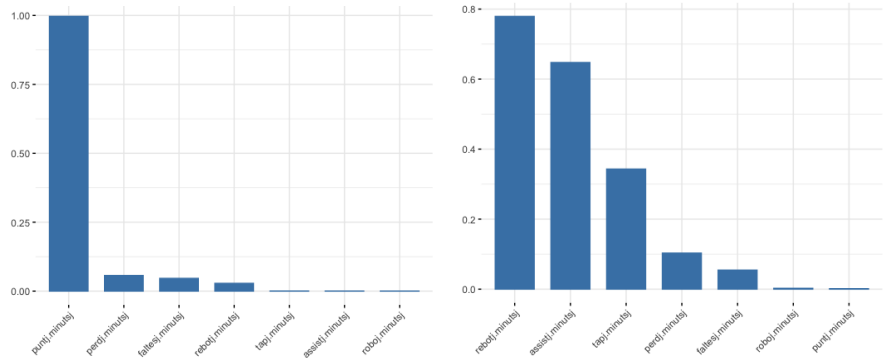


Figure 3.2: This figure shows the distribution of every single player.

We can observe in Figure 3.2 that visualizing the image it is possible to intuit the existence of different groups of players that could form different clusters. For example, we can see on the right of the image that number 1 and number 3 are almost in the same place. These numbers correspond to *Mike James* and *Shane Larkin*, two guards that have a similar playing style. To have a further comprehension of why

these two players are almost together and why other players stay in the same zone of the graphic let us show the following figure. These two graphics describe the contribution of every variable to the first two principal components.



(a) First principal component. (b) Second principal component.

Figure 3.3: Comparison of the two foremost dimensions.

Note that the first principal component direction is a linear combination of the seven variables of the data set. This direction is very similar to the *points* direction because the other six variables do not affect so much to its rotation. On the other side, we can see that there are three variables that notably affect the direction of the second principal component. This two directions are the ones that form a basis in $\mathbb{R}^2$ and are the ones who determine the distribution of the players.

Let us now show the results of implementing the method to the three groups given by the EuroLeague. Remember they are *guards*, *forwards*, and *centers*. Starting with the guards, we calculated the principal components for this new data set. The next figure presents the results achieved.

In this case, the first and the second principal components keep the 84.9% of the variance. This means that the graph that projects the data in two dimensions will be very representative. Notice that the loss of information is very low. The next figure illustrate the guards distribution regarding their principal components.

Again, the numbers 1 an 3, still corresponding to *Mike James* and *Shane Larkin* appear really close. We will see that this is because the first dimension direction is very similar than the *points* direction, like in the general case.

One of the interesting facts that come into view when we perform the partial analysis is the difference that we find comparing the guards with the whole data. In this case, the second principal direction is mainly influenced by the *assists*, and not by the *rebounds*. This fact could seem trivial because the guards do not have the objective of getting rebounds. They are the players who are running the ball the

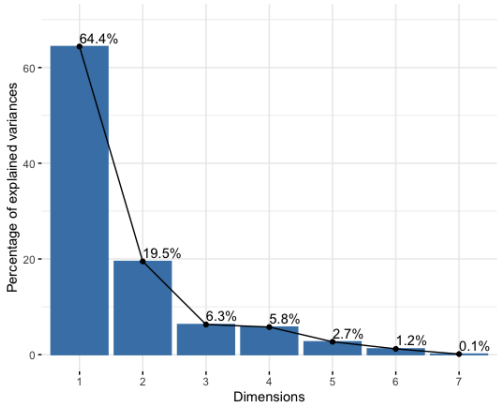


Figure 3.4: Percentage of variance contained in every guards principal components.

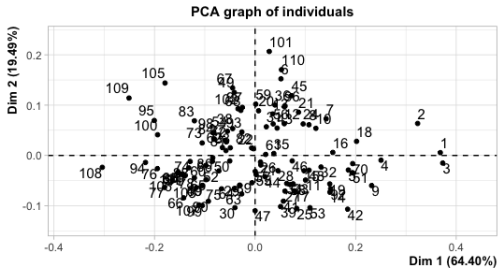
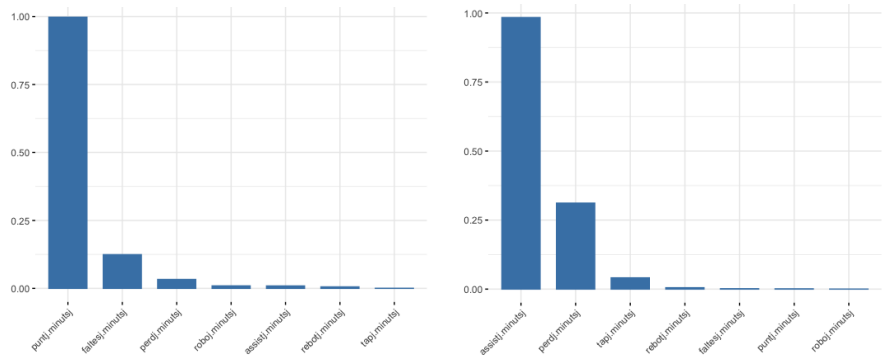


Figure 3.5: Guards distribution in the new axes.

major part of the match and often they are the best passers of the team. Then, with this analysis we can intuit the existence of two different groups of guards, ones with a high scoring average and the others with the aim of moving the team and help the rest of the players to make it easier to score. But looking the Figure 3.5 the groups are not clearly defined.

Observe now in Figure 3.6 how the initial variables affect the new axes for the guards case. Notice that after *assists*, the second variable with more influence is *turnovers*. This fact support our intuition. The guards with more assists use to have more turnovers because they run the ball more time than other guards and they take more risks than a guard focused on shooting and scoring.

Let us continue the partial analysis studying the forwards case. In this situation



(a) First principal component. (b) Second principal component.

Figure 3.6: Influence of the variables to the principal components.

it is possible to intuit more than one group of players inside the data with the results obtained. See first the figure containing the variance percentage included in each principal component.

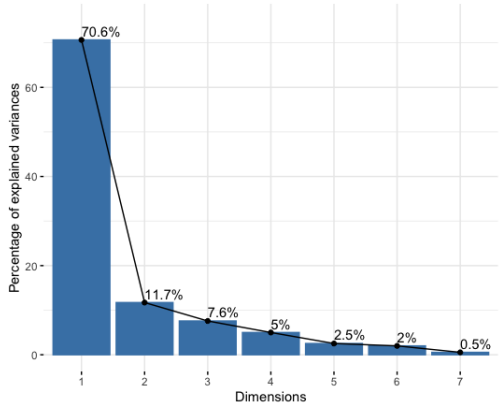
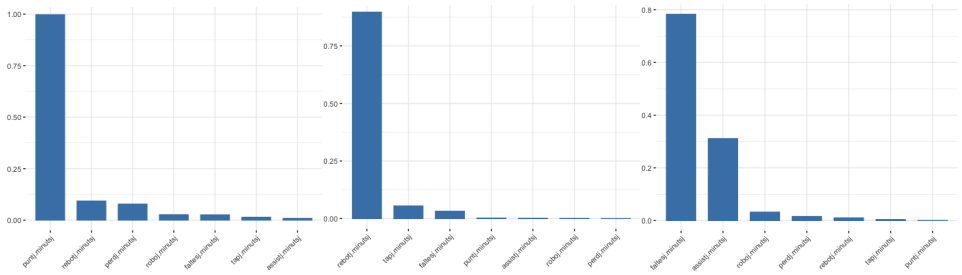


Figure 3.7: Percentage of variance contained in every forwards principal components.

Here, we have achieved a 82.3% of the variance with only the first two principal components. Observe that in forwards case there is less difference between the second and the third principal components than in the guards case. Go to Figure 3.8 to check the influence of the variables to each component. In the guards case, the two variables that characterized more a player, after a principal component analysis, where the *points* and the *assists*. Now the variables will be the points in the first principal component, and then with less significance, the rebounds for the second and the fouls for the third.

This could mean that we can find more than two groups in the data, but observe



(a) First principal component. (b) Second principal component. (c) Third principal component.

Figure 3.8: Influence of the variables to the principal components.

that in the case they exist, the following figure do not help us to visualize them clearly. Again, as in the previous cases, we show the distribution of every forward regarding the firsts two dimensions and we can see that in the center of the image, where the dimension two is negative there are three bunch of point that could create different clusters of players

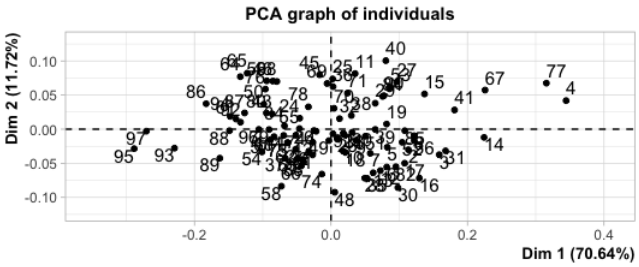


Figure 3.9: Forwards distribution

Finally, to end with the partial analysis, we study the centers case. We start showing the distribution of every center to see that the are many collections of few points that appear really close from each other in the same position of the map. Figure 3.10 reveal different centers with the same playing style. This tool could be nice if we want to replace a player finding another one with the same characteristics but it is not good enough if our desire is to group the whole centers into few new positions that can accurate the traditional classification.

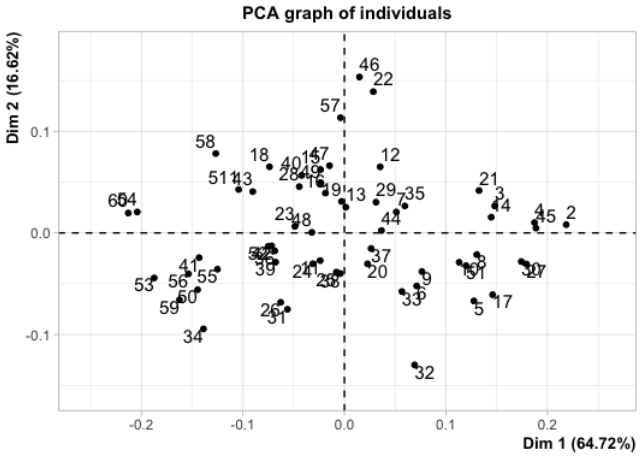


Figure 3.10: Centers distribution

To continue with the analysis, again, like we have made in the previous case, we will show the percentage of variance remaining in every principal component and how the initial seven variables have determined the new axes conforming a basis of $\mathbb{R}^7$.

Note that there is an interesting fact in this last case as it is the one where we have obtained the more variance in the third principal component. This could affect the loss of information fact but notice that in total, looking at the first two components to see if the projected graphic is significant, we get a 81.3% of the variance.

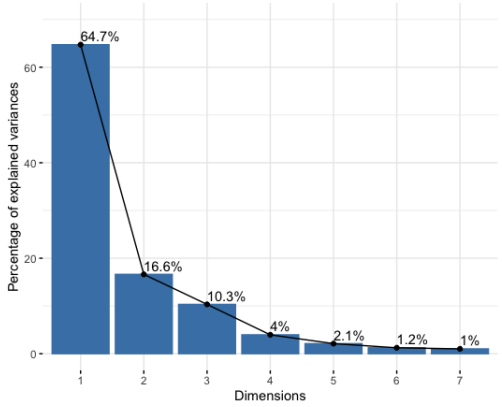
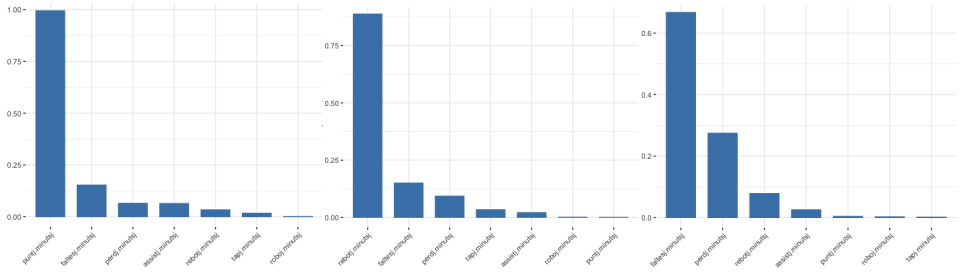


Figure 3.11: Percentage of variance contained in every principal component.



(a) First principal component. (b) Second principal component. (c) Third principal component.

Figure 3.12: Influence of the variables to the principal components.

3.2 Persistent homology

As we have introduced in this chapter, topological data analysis will provide additional information, regarding the observation of different clusters of players, aligned with the intuitions made in the previous section.

It was one of the biggest issue of these notes to evaluated the accuracy of the topological data analysis when trying to detect different clusters in a massive data set and the results are not as good as Mapper but they give significant information about the shape of our data.

We computed the persistent diagram, through the Vietoris-Rips complexes, for the whole players data set and another one for every single group of guards, forwards and centers. This way we can discern about the distribution of the 0-dimensional and 1-dimensional homologies around the persistent diagrams diagonal plus how far they are from it for the purpose of giving an interpretation of the hidden groups in the data.

In the previous section, we worked without normalizing the data but in this case, as we need to choose a distance to compute the persistent diagrams, we used the variance normalized Euclidean distance like in the next section since it is the one used in [13]. We did not calculate the 2-dimensional homologies because they had a high computational cost and they did not bring meaningful information. Then, the results obtained computing the persistent diagram for the whole data set are presented in Figure 3.13. Note that most of the 1-dimensional homologies remain near the diagonal but it is interesting to see that few ones appear far from the diagonal and they are born relatively late. This shows the existence of grouped points in the data set and notice that these points will stay separated from the main bunch of points in the point cloud.

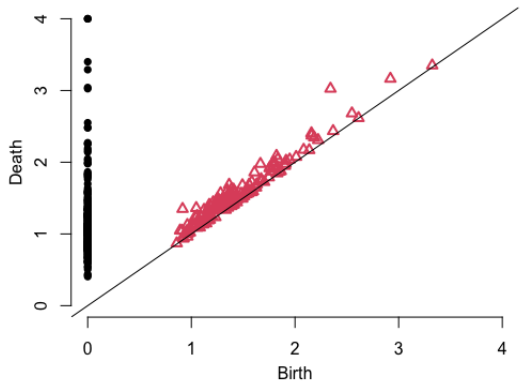


Figure 3.13: Persistence diagram from the EuroLeague data set.

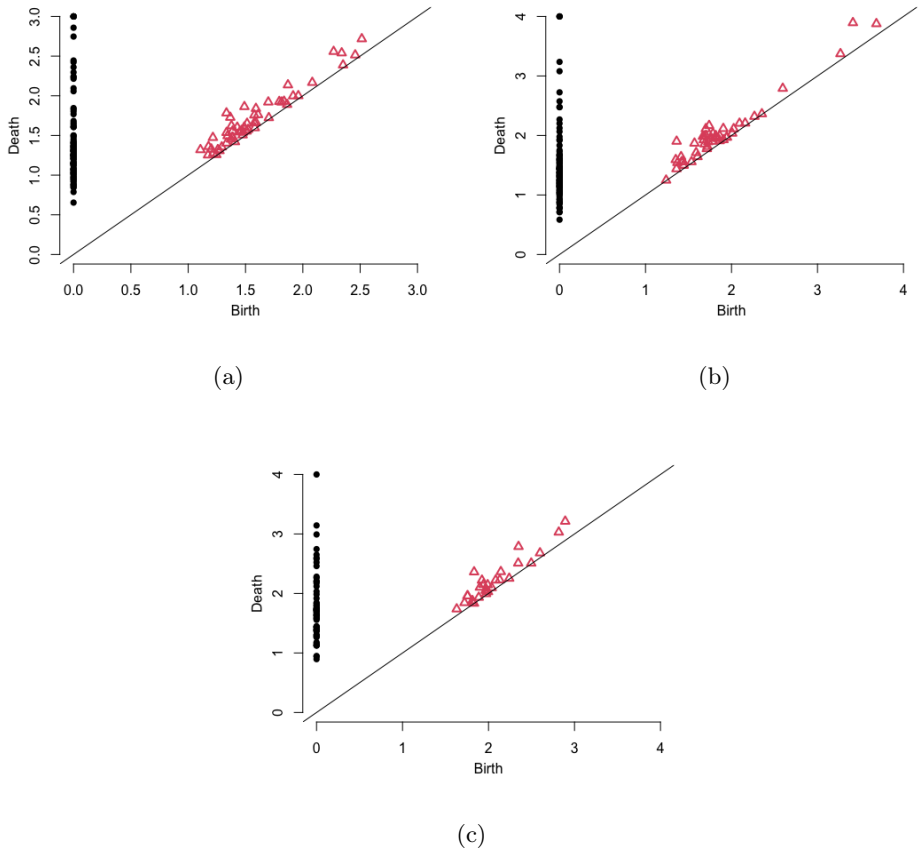
Now we continue analysing the persistent homologies that appear when we select the different groups given by the EuroLeague giving an interpretation of the persistence diagrams in a qualitative sense.

As we expected, we can reasonably talk about the existence of hidden subgroups in every group only watching the distribution of the 1-dimensional homologies in every persistent diagram.

Starting with the guards case, observe that in Figure 3.14(a) the main part of 1-dimensional homologies stays near the diagonal. This means that they live few time, i.e. their birth is almost equal to their death. The ones that provides major information are the ones separated of the central bunch. We can see that few of them remain far from the diagonal. These loops live a considerable time and, like in the general case, they represent, if we consider a guards point cloud, points that are far away from the main group. Then, clearly we perceive that there exists more than one group of guards.

In certain way, the same happens when we analyse the forwards case. Here, we can intelligibly distinguish the presence of different groups inside the forwards point cloud. Notice that apparently the number of groups will be less than in the guards case there because there are less 1-dimensional homologies apart from the diagonal and borned later than the major part.

Lastly, the center case shows apparently the less significant information but it if we do a deeper analysis we will see that we can conclude that there exist more than one group of them. Fist of all, it is the group with less players and see that this fact is corresponded in their persistence diagram, in 3.13(c), as it is the one with less



(a) (b)

(c)

Figure 3.14: Persistence diagrams for data sets containing (a) guards, (b) forwards and (c) centers, respectively.

1-dimensional homologies. However, notice that there is one loop created a bit later than the major part that stay alive for a sustainable time, and also, there are two homologies borned realtively later than all the other ones, and again, they are not in the diagonal.

Finally, to end this section, we will show what happen if we drop a group of playes from the whole EuroLeague data set. For example, consider a point cloud with all the guards and all the forwards and see how its persistent diagram changes comparing it with the general one. We have performed this analysis for every case and the most meaningful one was the case where we dropped the guards. See that there is a group of 1-dimensional homologies borning when the time is equal to one, or nearly equal, in Figure 3.13 that disappear in Figure 3.15. These homologies remain in the persistence diagram when we dropped the forwards or the center.

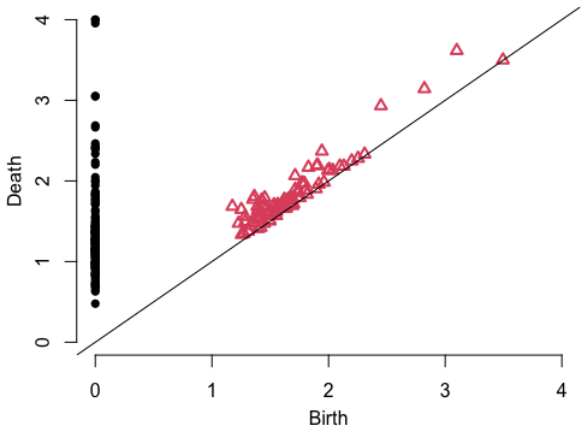


Figure 3.15: Persistence diagram of the EuroLeague data set dropping the guards.

3.3 Mapper results

As we have been saying in this work, the most accurate results were obtained when we applied the Mapper algorithm. We will show in this section the results for two different cases determined by the number of filters applied in the algorithm. The filters were two for the first case and three for the second, counting the metric used as a filter. It was in the second case where we got the most satisfactory results obtaining the whole EuroLeague shape explained in a similar graph that the one given in [13],see Figure 1. The 13 positions found in the article were: *offensive ball handlers, defensive ball handlers, combo ball handlers, shooting ball handlers, role playing ball handlers, 3-point rebounders, scoring rebounders, paint protectors, scoring paint protectors, role player, NBA first team, NBA second team and one of a kind.* We will make the analysis considering this positions.

3.3.1 Mapper with two filters

In the first case, the filters used were the variance normalized Euclidean distance metric and the first principal component of the data obtained after applying the singular value decomposition. We started computing the graph in R, with the function *mapper1D*. The resolution parameters chosen were the same as in [13]: a total of thirty intervals and a 50% of overlap. The results computing this function were the following.

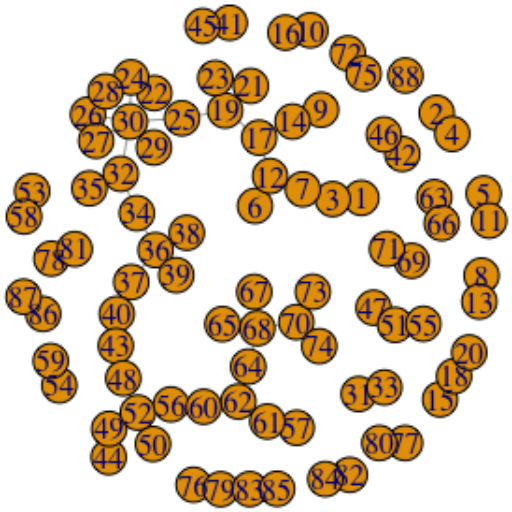


Figure 3.16: Mapper graph computed with *mapper1D*.

In this construction, every node contain one or more players and two nodes are connected if they share at least one player. It is possible to see which players are in every node and this way we saw one the one hand that this primary analysis gave optimistic results. If we follow the numbers of the nodes, the line they form starts with the guards, continues with the forwards and ends with the centers. On the other hand, we see that there are many players that stay ungrouped.

We tried to solve this problem by increasing the overlapping percentage. Increasing it to the 65% provoked that some intervals, that before had a null intersection, overlapped and this made the nodes to be connected with an edge. See the difference between Figure 3.17 and Figure 3.16. Furthermore, the numbers of nodes increased and we could distinguished more than one group of guards in the graph.

See that in Figure 3.17 node number 12 clearly separated two groups of connected group. If we see the player in the group that has node 7 in the center we find names like *Facundo Campazzo* or *Nick Calathes*. Bothe players are well known player in EuroLegue for being leaders in their teams. The irst one presents good numbers in many of the seven components studied, he scores easily, he has also good assisting average, but he stands out for being more efficient in defense than other guards.

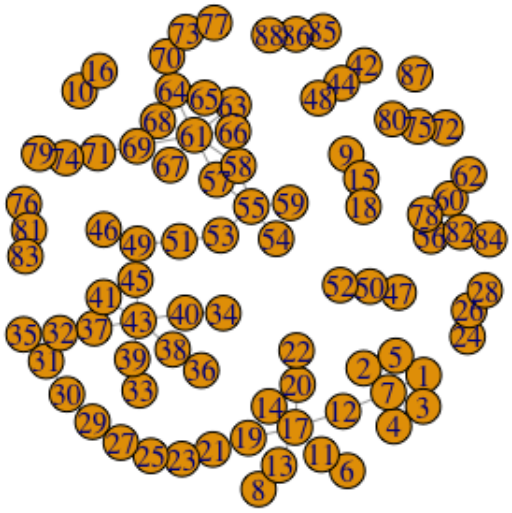


Figure 3.17: Mapper graph computed with *mapper1D* with a 65% of overlapping percentage.

Then we could say that the nodes in this area belong to the defensive ball handlers. In the other group next to node number 12, we find players like *Sergio Rodríguez*, *Vasilije Micic* or, again, Nick Calathes. In this area players have the characteristics as before but we also start finding players like *Thomas Heurtel*, a player that have elevated ratios on point an assist but is not as good in defense as the players in the first group. This means that near the node 17 we finde the combo guards. If we continue thought the line created, we find players like *Mike James* or *Alexey Shved*. Both players are guards that have an elevated ratio of points so we can say that we are moving from the combo ball handlers to the offensive and shooting ball handlers.

For the forwards the discerning is not as accurate as before. The nodes contain many players but it is true that the line follow a transition from the guards to the centers respecting the fact that the forwards that have a playing style more similar with the guards than with the centers appear before than those ones who play almost like a center.

When the firsts centers start appearing the analysis become more intelligible. For example in node 61 appear players like *Nikola Milutinov*, a center with an elevated

average in points and rebounds and then we see that the line bifurcates into two different branches with players with a less ratio on rebounds. In one line we find players with elevated percentages of blocks and points and in the other line, only blocks. The interpretation here is that we pass from the scoring rebounders to the paint protectors and the scoring paint protectors, every group in a different branch.

To see this analysis with more detail, we computed an interactive plot that shows which players remain in every node and also allows us to colour the nodes depending on the weight of the variables. The size of the nodes has been modified in order to easy the visualization of the data. The node with more players will be much bigger than the nodes with one or very few players.

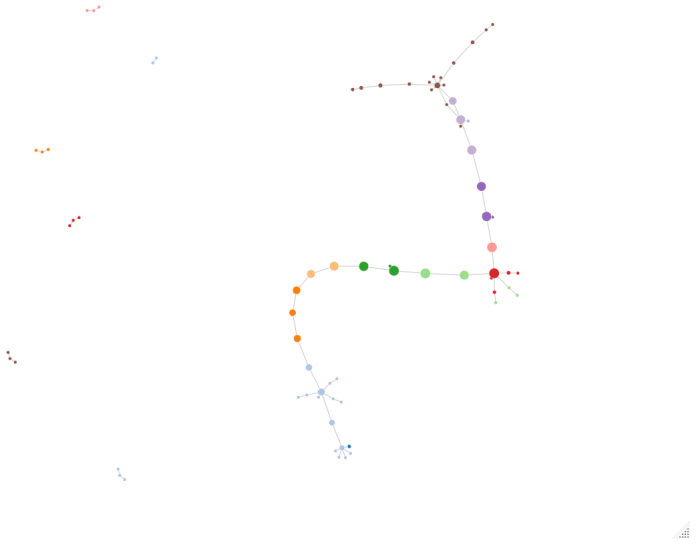


Figure 3.18: Interactive Mapper graph coloured by the first principal component.

As we stated in the previous chapters, we know the filter value for every point in our data. Consider a vector such that every component is the filter value of the point. This case, the filter was the first principal component and we have created ten quantiles dividing the vector into ten intervals. Then, every node was coloured depending of the interval it belonged. We could not compute a proper colour scale but it is interesting to see that the group of nodes belonging to each quantile appear ordered in the graph. Respect the points far away from the line, they correspond to ungrouped players. The interactive plot has a zoom an if we made the image smaller we would see that there are more ungrouped players.

3.3.2 Mapper with three filters

In the last case of the work, Mapper has been performed with three filters obtaining the graph with the shape of EuroLeague. We have used the variance normalized Euclidean distance and, this time, we have used the first and the second principal components obtained through the singular value decomposition.

We have computed two graph, like in [1] and this time, the resolution parameters chosen have been 20 and 30 intervals respectively and a 50% of overlap for both.

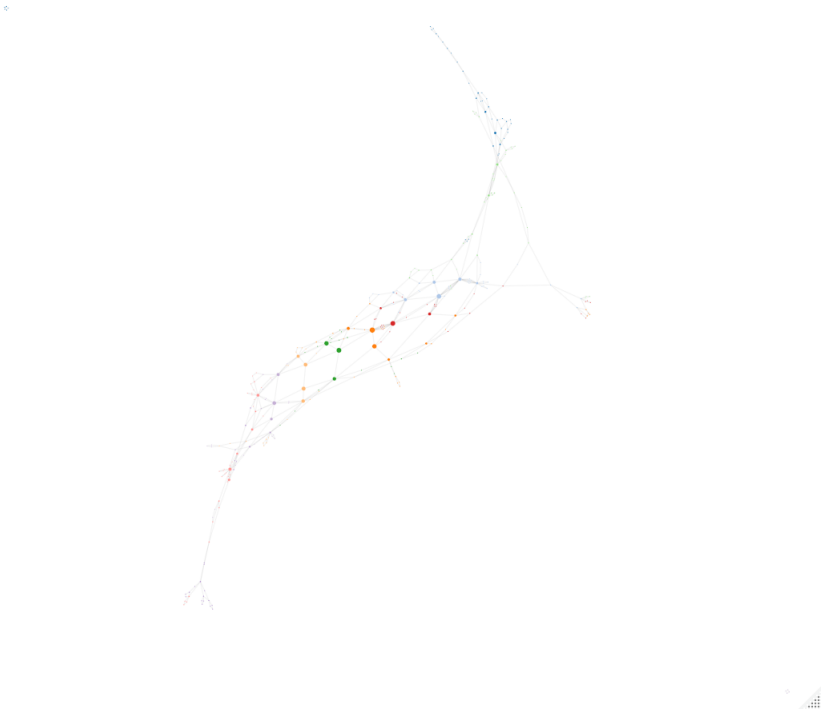


Figure 3.19: Interactive Mapper plot considering three filters and 20 intervals.

Note that the shape of Figure 3.19 is a kind of extended version of 3.17. This time the order goes the other way around, on top, the we find the defensive ball handlers and in the end there are the scoring rebounders. But there is a really interesting fact in this case. If we move on top of the figure, there is a group that is not shown in Figure 3.19.

Figure 3.20: Top part of the interactive Mapper plot considering three filters and 20 intervals.

On the lowest part of the plot we see the defensive ball handlers. But there is a group of players disconnected with the main graph on the top right of the figure. There we find players like *Nikola Mirotic* or *Will Clyburn*. They are two of the best players in the league, they have fantastic ratios in almost every variable studied. They can score easily and create many situations to let their teammates to play better. Here we see what in [13] it is called NBA first team. Note that NBA has much more players than EuroLeague, and furthermore, this study is performed to 267 players. Then, probably, the positions NBA first team and NBA second team

will appear mixed in this plot. It is interesting, for example, to see the presence of *Bostjan Dubljevic*, a center playing in Valencia that has similar ratios than *Nikola Mirotic* and *Will Clyburn* but he is playing in a team that normally finishes in the middle of the final classification.

Finally we compute the graph with the same filters and the same resolution parameters than the graph published in [13], and the results were really encouraging.

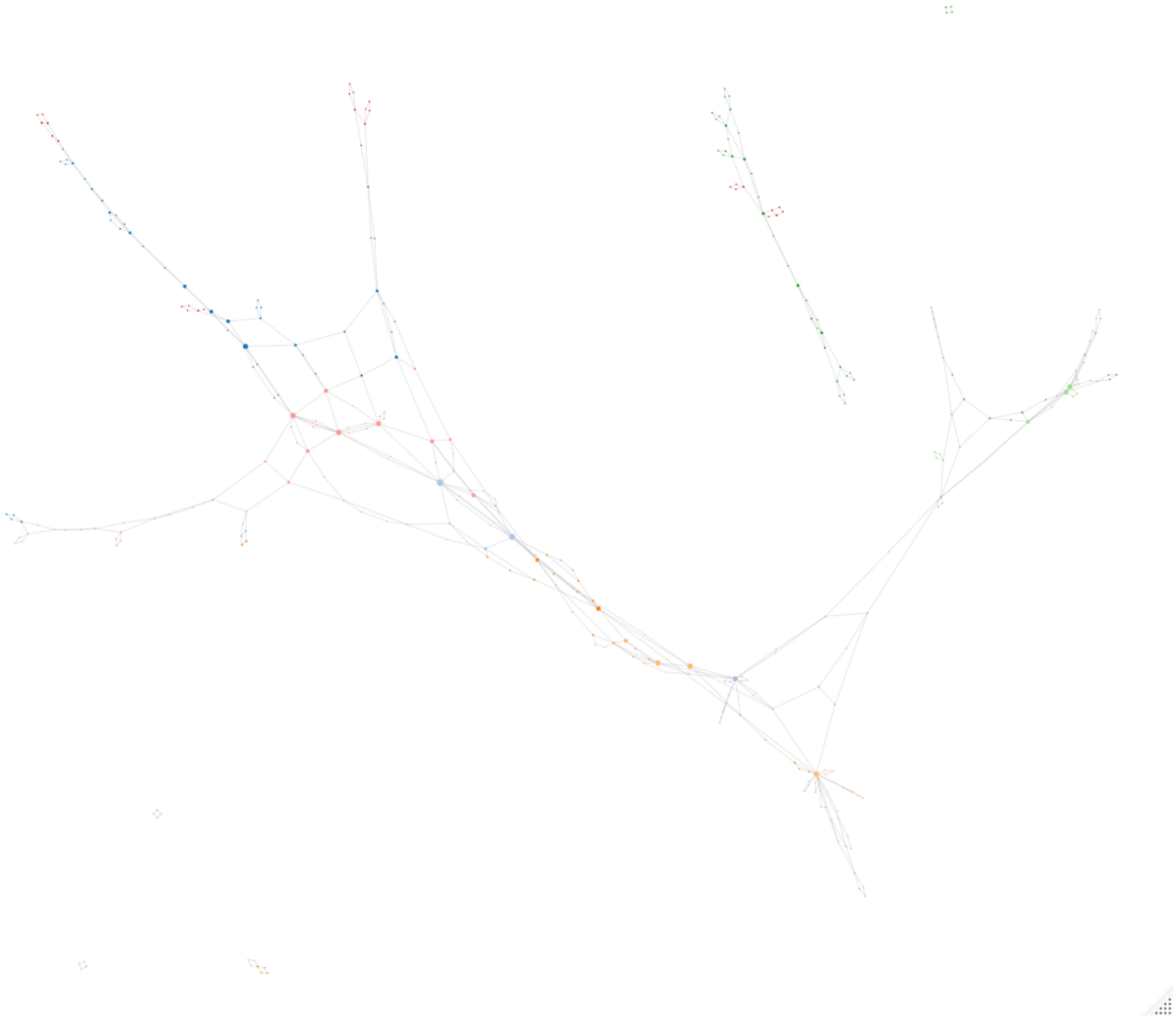


Figure 3.21: Shape of EuroLeague

See that there are great similarities between the shape of Figure 3.21 and the shape of Figure 1, but, observe in the middle of the graph that there is a group of notes disconnected of the main figure. This nodes consist in players like *Sergio Rodríguez* and many other guards. In [13] all the guards where connected with the main group of nodes but it is different for the European basketball. This could show existing difference between American and European basketball. In Europe, the role of some guards in the court slightly depart from the rest of the players. This shows a grade of uniqueness in their playing style that could characterize the European playing style.

Chapter 4

Conclusion

Historically, like in American basketball, the players in Europe have been classified following the traditional scheme of five positions. We have attempted to prove the existence of more than only five and have tried different methods to achieve it. The first step in our procedure was to perform a principal component analysis and this gave us a primary approach of the existence of hidden groups inside the EuroLeague data. Then we studied homological features, considering our data set as a point cloud $\mathbb{X} \in \mathbb{R}^7$ trying to find groups of points staying far away from the main bunch and we did it by studying persistence diagrams. Finally, we obtained a graph with the distribution of every EuroLeague player grouped into several positions.

We have also found interesting details in this work that we have not studied further, but which could be interesting new lines of investigation. We have seen that Mapper sometimes shows a huge similarity of the playing style between two players. This is the case, for example, of *Facundo Campazzo* and *Peyton Siva*. Campazzo was one of the best players in the league last season; however, this year, in the middle of the season, he changed the EuroLeague for the NBA. His team lost an important player in the middle of the competition and this affected their results.

One of the examples of the significant information that one can obtain by applying topological data analysis is given in [10]. In this paper, the authors compute a barcode for every team in the National Hockey League (NHL) and study its shape by comparing the teams on the top and the worst teams. The results are very illuminating and describe how a team's barcode changes if a player is replaced. This could perfectly be applied in basketball, for example in the case we have mentioned in the previous paragraph.

Bibliography

[1] Anil K. Jain and Richard C. Dubes, *Algorithms for Clustering Data*, Prentice Hall Advanced Reference Series. Prentice Hall Inc., Englewood Cliffs, NJ, 1988. (58-89)

[2] L. Aromi, *Analysis of Financial Time Series using TDA: Theoretical and Empirical Results*, (2020).

[3] S. Biasotti, B. Falcidieno, M. Spagnuolo *Extended Reeb Graphs for Surface Understanding and Description*, Istituto per la Matematica Applicata, Consiglio Nazionale delle Ricerche.

[4] K. Borgwardt, M. Horn, M. Moor, B. Rieck, *Topological autoencoders*, arXiv:1906.00722v5 [cs.LG], (2021).

[5] G. Carlsson, F. Mémoli, G. Singh, *Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition*, Eurographics Symposium on Point-Based Graphics, (2007).

[6] G. Carlsonn, *Topology and data*, Bull. Amer. Math. Soc. 46 (2009), 255-308.

[7] F. Chazal, B. Michel, *Mapper Algorithm with the R-package TDAmapper*, (2016).

[8] C. M. Cuadras, *Métodos de análisis multivariante*, Colección, Estadística y Análisis de Datos, (1991).

[9] B. T. Fasy, J. Kim, F. Lecci, C. Maria, *Introduction to the R package TDA*, arXiv:1411.1830v2 [cs.Ms], (2015).

[10] D. Goldfarb, *An application of topological data analysis to hockey analytics*, arXiv:1409.7635v1, (2014).

[11] P. Harrington, *Machine learning in action*, Manning Publications Co, (2012).

[12] S. C. Johnson, *Hierarchical clustering schemes*, Psychometrika 2 (1967), 241-254.

[13] P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, G. Carlsson, *Extracting insights from the shape of complex data using topology*, (2013).

[14] P. Pearson, D. Muellner, G. Singh, *Analyze High-Dimensional Data Using Discrete Morse Theory*, (2016).