Topological Data Analysis with Metric Learning and an Application to High-Dimensional Football Data

David Alejandro Perdomo Meza

Abstract

We explore the exciting interaction between topology, algebraic topology and data mining applied to a real life data set of football statistics. In particular, we investigate how to apply the methods of TDA Pipeline and Persistent Homology on the football data, and show how to interpret these results and gain information from them.
We explain in detail our implementations of each step of the Pipeline, and in the process and through the applied example exhibit some of its important flaws, and wrap up proposing guidelines for future investigation into improving the method.

1 Introduction

The rate at which information is being accumulated in the world is daunting. Globalization has not only brought along seemingly endless quantities of information to be learned, recorded and collected; but also the increasing need to interpret it.
Governments have copious amount of data about millions of citizens, banks have thousands of financial indicators to keep track of, and medical investigators have record amounts of biological information in their quest to rid the world of disease; and all of them have something to gain through interpreting the information at hand.

Vectorization of the data has summoned mathematics to the forefront, demanding for elegant approaches to exploiting the possibilities.

In this paper, we will discuss the prominent role that methods inspired by topological and geometrical theories have come to play in data mining.
The philosophy is that when data is represented as a point cloud in euclidean space, it will naturally lie along a geometric object like a manifold, perhaps with noise.
Topological study of this object will provide *qualitative* insight into the intrinsic nature of data, free from the *quantitative* aspects that may be artifacts of the specific vectorization.
The TDA approach is essentially *coordinate-free*, as it will focus on inherent *closeness* and *shape* of the data, rather than it's specific numerical representation.

Topological Data Analysis (TDA) is not a quantitative method that will offer intervals of confidence, hypothesis testing or density estimation; but rather it will provide the investigator with *knowledge* about the nature of the data, from which more quantitative hypothesis may be formulated.
The subtle difference between TDA and other statistical methods can be likened to the difference between Clustering and Pattern Recognition: while a pattern recognition application can provide a method to differentiate two groups of essentially different data in a point cloud, clustering will first of all tell the investigator the inherent property of there *being* two groups to begin with.

Throughout this document, we will introduce some TDA methods, particularly the TDA Pipeline, and exhibit their applications to a football data set with in-game player statistics for the 2011-2012 season of the Premier League.
The data can be utilized either to study the performances of individual players throughout the season, or to study team performances as a whole.
We will show how the usage of these techniques can lead us to answer *qualitative* questions about the football data at hand. Some of these are:

1. Individual Player Performance
   (a) Can playing positions be topologically distinguished from the in-game statistics?
   (b) Can "top" players be separated from the rest of the pack?
2. Team Performance
   (a) Are victories, draws and defeats topologically identifiable?
   (b) Can different game strategies and playing formations be distinguished?
   (c) Which teams employ similar styles?
   (d) Can "top" teams be separated from the others?

The layout of the document is as follows:

In Section 2 we introduce some of the topological theoretical background needed to formulate the applied TDA Pipeline technique. The protagonists of this section are the Čech Complex and the *Nerve Theorem*, which we will transport from the context of topological spaces into the discrete setting.
We will also briefly introduce the theory of Persistent Homology.

In Section 3 we will carefully introduce the TDA Pipeline method, which is a visualization technique that has been successfully applied to problems in medicine, sports and politics to gain important knowledge from raw data. The outcome of the technique is a TDA Graph, which we will show how to interpret later on.

In Section 4 we show in detail some of our implementations for the Pipeline. These include Metric Learning algorithms to obtain a meaningful metric in the problem, the filtering functions used and our own methodology to choose clusters in the data.

In Section 5 we give further details on our football data, and more importantly show the results of the TDA Pipeline application to this data set, addressing the questions we just set forth.

Finally, in Section 6 we conclude and reflect on our adventuring investigation into the possibilities of TDA, and propose some guidelines on how the methods of TDA can be improved, and what problems should shepherd future investigations on the subject.

2 Topology and Data: Mathematical Background

In this section we will give an overview of the topological concepts and results that inspire the computational techniques we wish to present.

Definition 2.1. An **abstract simplicial complex** is defined as a pair $(\Lambda, \mathfrak{B})$ where Λ is a finite set and $\mathfrak{B} \subseteq \mathbb{P}(\Lambda)$ such that whenever $X \in \mathfrak{B}$ and $Y \subseteq X$ with $Y \neq \emptyset$, we have that $Y \in \mathfrak{B}$.
The set Λ is referred to as the **vertex set** of the simplicial complex, and a set $\{\alpha_0, ..., \alpha_k\} \in \mathfrak{B}$ is referred to as a k**-simplex**.

Figure 1 illustrates the geometric interpretation of simplicial complexes.

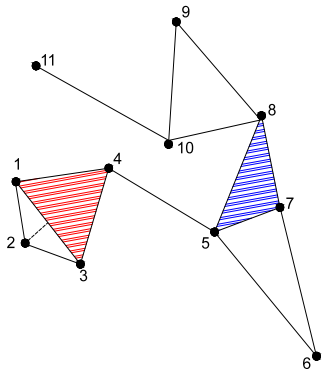

Figure 1: *The vertex set is* $\{1, 2, ..., 9, 10\}$*, while 1-simplices are seen as edges, 2-simplices as blue triangles and 3-simplices as red pyramids.*

Figure 2 shows the result of adding the 2-simplex $\{8, 9, 10\}$:

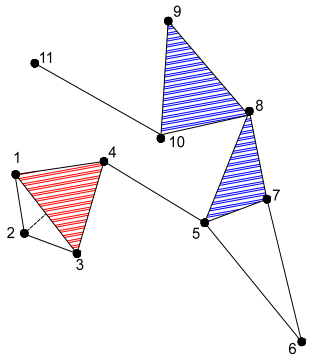

Figure 2:

Definition 2.2. Let X be a topological space and $\{\mathfrak{U}_\alpha\}_\Lambda$ a covering of X. The **nerve** of $\mathfrak{U}$, denoted by $N(\mathfrak{U})$, is defined as the abstract simplicial complex with vertex set Λ, where $\{\alpha_0, ..., \alpha_k\} \in N(\mathfrak{U})$ (i.e. $\{\alpha_0, ..., \alpha_k\}$ span a k-simplex) if and only if $\mathfrak{U}_{\alpha_0} \cap ... \cap \mathfrak{U}_{\alpha_k} \neq \emptyset$.

Nerves are a crucial concept in algebraic topology, because they guarantee the link between the homotopy types of general topological spaces and simplicial complexes. Since the homology of simplicial complexes is algorithmically computable, this connection is theoretically very powerful. The link comes in the form of the Nerve Theorem, which is originally attributed to Karol Borsuk:

Theorem 2.3. Nerve Theorem: Suppose X is a topological space and $\mathfrak{U}$ is an open and numerable cover. Furthermore, suppose that $\forall S \subseteq \Lambda$ with $S \neq \emptyset$, we have that $\bigcap_{s\in S} \mathfrak{U}_s$ is either empty or contractible. Then $N(\mathfrak{U})$ is homotopy equivalent to X.

A covering that holds the hypothesis of the Nerve Theorem is, with good reason, referred to as a **good covering**.

Remark 2.4. It's important to disallow coverings which index empty sets.
While technically a covering with an empty set can be a good covering, since the additional empty set doesn't interfere with this condition, the node of the empty set will be isolated within the graph, technically adding an additional connected component.

Now, when $\mathfrak{U}$ is an *open* cover of X, the nerve of $\mathfrak{U}$ is also referred to as the **Čech complex of X attached to $\mathfrak{U}$**, and denoted by $\check{C}(\mathfrak{U})$.

In the special cases where X is a metric space and $\mathfrak{U} = \{B_\epsilon(v)\}_{v\in V}$ for some $V \subseteq X$, we write $\check{C}(V, \epsilon)$.

The Nerve Theorem extends to Riemannian manifolds in the following version:

Theorem 2.5. Let M be a compact Riemannian manifold. Then $\exists\epsilon_0 > 0$ such that $\check{C}(M, \epsilon)$ is homotopy equivalent to X whenever $\epsilon \leq \epsilon_0$. Moreover, $\forall\epsilon \leq \epsilon_0$, there is a finite subset $V_\epsilon \subseteq M$ such that $\check{C}(V_\epsilon, \epsilon)$ is also homotopy equivalent to M.

Notice that the quality of the approximation of the homotopy type of a space X by that of $\check{C}(\mathfrak{U})$ depends on the *goodness* of $\mathfrak{U}$. We are going to take a small step further, and define a modification of the Čech complex which will provide a slight improvement in the approximation of the homotopy type of X:

For a covering $\mathfrak{U} = \{\mathfrak{U}_\alpha\}_{\alpha\in\Lambda}$, define the **Čech complex by connected components**, denoted by $\check{C}^{\pi_0}(\mathfrak{U})$, as the nerve of the covering $\{\mathfrak{U}_{(\alpha,\xi)}\}_{(\Lambda,\Xi)}$, where ξ indexes the path connected components of $\mathfrak{U}_\alpha$.
Figure 3 shows the improvement:

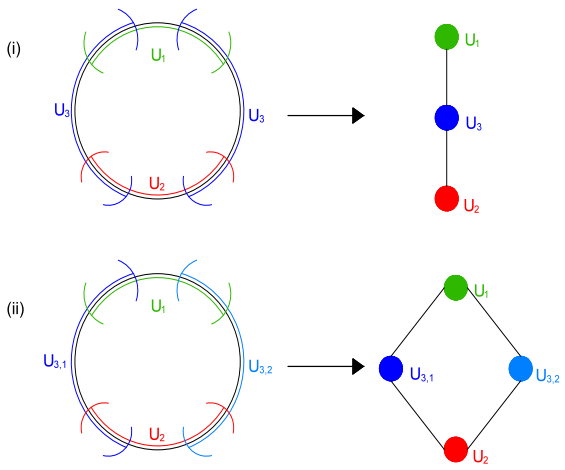

Figure 3: *The first panel shows the cover of a circle and its corresponding Čech complex, while the second shows the Čech complex by connected components of the cover. Clearly, only the second is homotopy equivalent to the circle.*

This improvement can be generalized in the following lemma:

Lemma 2.6. $\{\mathfrak{U}_\alpha\}_\Lambda$ is a good covering $\Rightarrow$ $\{\mathfrak{U}_{(\alpha,\xi)}\}_{(\Lambda,\Xi)}$ is a good covering.
The converse does not generally hold.

Proof. The idea of the proof is to show that any set of the form $\mathfrak{U}_{(\alpha_1,\xi_1)} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_k)}$ is either empty, or equal to $\mathfrak{U}_{\alpha_1} \cap ... \cap \mathfrak{U}_{\alpha_k}$.

This is fairly straightforward:
Lets notice first of all that $\mathfrak{U}_{(\alpha_1,\xi_1)} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_k)} \subseteq \mathfrak{U}_{\alpha_1} \cap ... \cap \mathfrak{U}_{\alpha_k}$ for any choice of $(\xi_1, ..., \xi_k)$.
On the other hand, $\mathfrak{U}_{\alpha_1} \cap ... \cap \mathfrak{U}_{\alpha_k}$ is contractible, and therefore connected.
Since $\mathfrak{U}_{\alpha_1} \cap ... \cap \mathfrak{U}_{\alpha_k} \subseteq \mathfrak{U}_{\alpha_1}$ and it is connected, it must happen that $\exists \xi_{1_0}$ such that $\mathfrak{U}_{\alpha_1} \cap ... \cap \mathfrak{U}_{\alpha_k} \subseteq \mathfrak{U}_{(\alpha_1,\xi_{1_0})}$, and therefore:

$$\mathfrak{U}_{\alpha_1} \cap \mathfrak{U}_{\alpha_2} \cap ... \cap \mathfrak{U}_{\alpha_k} = \mathfrak{U}_{(\alpha_1,\xi_{1_0})} \cap \mathfrak{U}_{\alpha_2} \cap ... \cap \mathfrak{U}_{\alpha_k}$$

Iterating this reasoning we obtain that there exists a choice $(\xi_{1_0}, \xi_{2_0}, ..., \xi_{k_0})$ such that

$$\mathfrak{U}_{\alpha_1} \cap \mathfrak{U}_{\alpha_2} \cap ... \cap \mathfrak{U}_{\alpha_k} = \mathfrak{U}_{(\alpha_1,\xi_{1_0})} \cap \mathfrak{U}_{(\alpha_2,\xi_{2_0})} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_{k_0})}$$

Since for any different choice $(\xi_1, ..., \xi_k) \neq (\xi_{1_0}, ..., \xi_{2_0})$ we have that:

1. $\left(\mathfrak{U}_{(\alpha_1,\xi_1)} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_k)}\right) \bigcap \left(\mathfrak{U}_{(\alpha_1,\xi_{1_0})} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_{k_0})}\right) = \emptyset$
2. $\mathfrak{U}_{(\alpha_1,\xi_1)} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_k)} \subseteq \mathfrak{U}_{(\alpha_1,\xi_{1_0})} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_{k_0})}$,

then it must be that $\mathfrak{U}_{(\alpha_1,\xi_1)} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_k)}$ is empty.

We have established that sets of the form $\mathfrak{U}_{(\alpha_1,\xi_1)} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_k)}$ are either empty or contractible.
Figure 3 serves as a counterexample to establish that the converse does not hold in general.
□

2.1 From Topological Spaces to Point Clouds

The challenge that we now face, is transporting the theoretical machinery of algebraic topology we just presented from the context of topological spaces to the context of data point clouds.

So far, we have constructed abstract simplicial complexes related to an underlying space X which under some circumstances contain qualitative topological information of X.
Suppose now we have a space X along with a good covering $\mathfrak{U}$ and a finite subset $\mathbb{X} \subset X$, which is sufficiently densely sampled so that for any $S \subset \Lambda$ with $\bigcap_{s\in S} \mathfrak{U}_s \neq \emptyset$, $\exists x \in \mathbb{X}$ with $x \in \bigcap_{s\in S} \mathfrak{U}_s$. Lets call such a sample a *good sample subject to* $\mathfrak{U}$.
In this scenario, we can begin to envisage a construction parallel to that of the Čech complex which attempts to recuperate the same qualitative information from the space X using solely information from the good sample $\mathbb{X}$.

The fundamental approach of topological data analysis is to assume an underlying Riemannian manifold for data point clouds (which when large enough we suppose are *good samples* for some good covering of this space), and to use these constructive ideas to make an *educated guess* as to what the topological qualities of this underlying space should be.
We have arrived at the central question of our investigation:

What qualitative topological information can we retrieve from an unknown space X when all we have at our disposal is a finite sample $\mathbb{X}$? How can we obtain this information?

Lets observe first of all that in the context of finite point clouds $\mathbb{X}$, a covering is a set of finite subsets of $\mathbb{X}$. The notion of *connected components* of these sets is no longer meaningful, since they will always be discrete. This notion will now be replaced with *clusters*.
Suppose we have a covering $\mathfrak{U} = \{\mathfrak{U}_\alpha\}_{\alpha\in\Lambda}$ of $\mathbb{X}$. We envisage a slight improvement parallel to the one achieved by $\check{C}^{\pi_0}(\mathfrak{U})$ by indexing a new covering by *clusters* of each open set $\mathfrak{U}_\alpha$, so that our new covering is $\mathfrak{U} = \{\mathfrak{U}_{(\alpha,\xi)}\}_{(\Lambda,\Xi)}$, where ξ indexes the clusters of points in $\mathfrak{U}_\alpha$.

Now, lets call a pair $(\mathbb{X}, \mathfrak{U})$ a *good discrete covering of* X when $\mathfrak{U}$ is such that $\mathfrak{U} = \{\mathfrak{U}_{(\alpha,\xi)}\}_{(\Lambda,\Xi)}$ is the restriction $\mathfrak{U} = \mathfrak{V}|_{\mathbb{X}}$ for some *good* covering $\mathfrak{V}$ of X, and $\mathbb{X}$ is a good sample subject to $\mathfrak{V}$.

Definition 2.7. For a discrete point cloud $\mathbb{X}$ and a covering $\mathfrak{U}_\Lambda$, we define its **point cloud Čech complex by clusters**, which we will denote by $\check{C}^{\pi_0}(\mathbb{X}, \mathfrak{U})$, as the abstract simplicial complex whose vertex set is (Λ, Ξ), where $\{(\alpha_0, \xi_0), ..., (\alpha_k, \xi_k)\}$ span a k-simplex if and only if $\mathfrak{U}_{(\alpha_0,\xi_0)} \cap ... \cap \mathfrak{U}_{(\alpha_k,\xi_k)} \neq \emptyset$.

The strength of this definition comes in the form of the following theorem, which we will call the *Discrete Nerve Theorem:*

Theorem 2.8. Discrete Nerve Theorem:
Let X be a topological space and $(\mathbb{X}, \mathfrak{U})$ a *good discrete covering*. Then $\check{C}(\mathbb{X}, \mathfrak{U})$ is homotopy equivalent to X.

Proof. If $(\mathbb{X}, \mathfrak{U})$ is a good discrete covering, then by definition $\exists\mathfrak{B}$ such that $\mathfrak{B}$ is a good covering of X, and $\mathfrak{B}|_{\mathbb{X}} = \mathfrak{U}$ and $\mathbb{X}$ is a good sample of $\mathfrak{B}$.
We will show that $\check{C}(\mathbb{X}, \mathfrak{U})$ is equivalent to $\check{C}(\mathfrak{B})$, and since by the Nerve Theorem the latter is homotopy equivalent to X, then we will be finished.
This is however fairly straightforward: We *obviously* (see Remark 2.9) have a one-to-one correspondence between vertices of both complexes, since both $\mathfrak{B}$ and $\mathfrak{U}$ are indexed by the same set Λ (see Remark 2.9).
Also, if $\mathfrak{B}_{\alpha_1} \cap ... \cap \mathfrak{B}_{\alpha_k} \neq \emptyset$, since $\mathbb{X}$ is a good sample of $\mathfrak{B}$, there exists $x_0 \in \mathbb{X}$ such that $x_0 \in \mathfrak{B}_{\alpha_1} \cap ... \cap \mathfrak{B}_{\alpha_k}$, and therefore $\mathfrak{U}_{\alpha_1} \cap ... \cap \mathfrak{U}_{\alpha_k} \neq \emptyset$.
With the other direction being trivial, this establishes that:

$$\mathfrak{B}_{\alpha_1} \cap ... \cap \mathfrak{B}_{\alpha_k} \neq \emptyset \Leftrightarrow \mathfrak{U}_{\alpha_1} \cap ... \cap \mathfrak{U}_{\alpha_k} \neq \emptyset,$$

establishing the equivalence between the complexes.

□

Remark 2.9. It's not exactly true that $\mathfrak{B}$ and $\mathfrak{U}$ are indexed by the same set.
It is perfectly plausible that $\exists \alpha_1, \alpha_2$ with $\alpha_1 \neq \alpha_2$ such that $\mathfrak{B}_{\alpha_1}|_{\mathbb{X}} = \mathfrak{B}_{\alpha_2}|_{\mathbb{X}}$, and is therefore unclear whether we index $\mathfrak{U}_{\alpha_1}$ and $\mathfrak{U}_{\alpha_2}$ separately or together.
Whatever choice we make, a minute's thought should convince the reader that this in no way damages the result of the Discrete Nerve Theorem.

Of course in a real scenario where X is not really known, the statement "$(\mathbb{X}, \mathfrak{U})$ is a good discrete covering" is not really verifiable. However, it is plausible to assume that any large enough sample will be *goodable*, meaning that it will be *good* for some discrete cover.

The matter of real importance then, is how to measure the *goodability* of a cover $\mathfrak{U}$ in the discrete setting.
We have evidently taken a step forward in ensuring the goodness of our cover by taking *connected components* in the form of *clusters* of our open set.
Figure 4 shows that this clearly isn't enough.

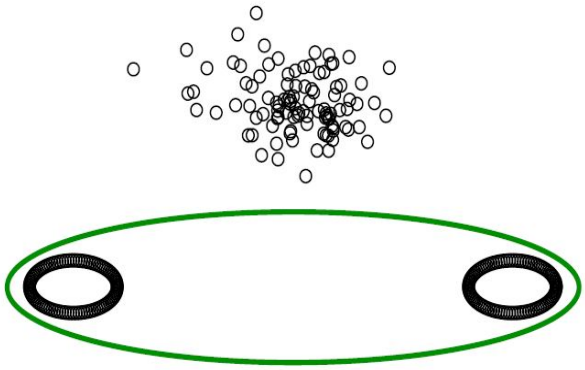

Figure 4: *The green oval represents an open set in the cover. Any sensible clustering scheme will separate the two circles into two clusters.*

Even though the $\check{C}^{\pi_0}(\mathbb{X}, \mathfrak{U})$ complex resulting from this cover recognizes the two circles as two distinct clusters of the set $\mathfrak{U}_\alpha$, it collapses each of the circles into a single vertex, losing this underlying qualitative information in the process.
The main problem this example has is that is assumes that *cluster components* necessarily

exhibit *contractible behavior*, which is of course not the case; as the example establishes. However, this problem should by sidestepped not from the *clustering* algorithm, which in this case seems to make the natural clustering choice of separating the two circles into two clusters, but rather from the *goodness* of the cover in the data set.
Figure 5 shows a different cover for this data set.

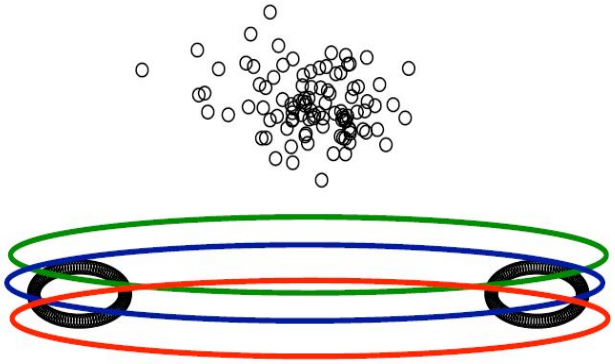

Figure 5: *Each oval represents an open set in this new cover. Each open set will now be recognized as having two clusters, one for each segment of the circle it contains, but the resulting Čech complex* ***will*** *pick up the circular structures.*

This example allows us to further our understanding on the virtues of our construction, and the virtues of having a good covering. The resulting complex $\check{C}^{\pi_0}(\mathbb{X}, \mathfrak{U})$ in this case should be homotopically equivalent to the underlying space, recognizing the *circle structure* in the data where the previous covering failed to do so.

The question we must now ask ourselves is how to *measure* the *goodness* of a discrete cover by identifying when its *cluster components* exhibit a certain manner of discrete contractibility. Now, the notion of *contractible* has not yet been convincingly integrated into the discrete point cloud setting. However a necessary condition (although not truly equivalent) for contractibility of a space is for it to be *acyclic*, i.e. to have trivial homology groups for all dimensions greater than 0.
Therefore, a valid step towards validating contractibility is verifying this condition.

The following section attempts just that.

2.2 Persistent Homology

The theory of *persistent homology* attempts to calculate homological information from a discrete data set. For more involved accounts of the theory the reader can reference [ZC05] or [Car09].

The essence of the theory is based on the result from Theorem 2.5.
Suppose we have a sample $\mathbb{X} \subset X$. How much information does $\check{C}(\mathbb{X}, \epsilon)$ provide?
Figure 6 (taken from [Ghr08]) shows the construction of $\check{C}(\mathbb{X}, \epsilon)$ when $\mathbb{X}$ is sampled from an annulus for different values of ϵ.

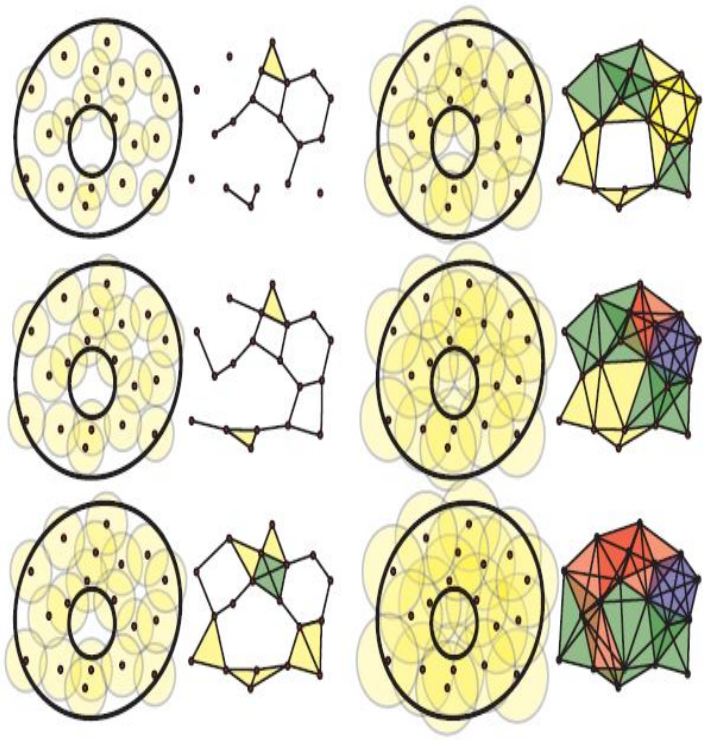

Figure 6: $\check{C}(\mathbb{X}, \epsilon)$ *for progressively larger values of* ϵ.

For small values of ϵ, $\check{C}(\mathbb{X}, \epsilon)$ is simply a set of $\#(\mathbb{X})$ isolated vertices. For large values of ϵ, $\check{C}(\mathbb{X}, \epsilon)$ is a single $(\#(\mathbb{X}) - 1)$-dimensional simplex.
However, for intermediate values, homological structure appears and disappears. The philosophy behind the approach of persistent homology is that in all likelihood, no particular ϵ will yield a complex $\check{C}(\mathbb{X}, \epsilon)$ with the exact homology type of X, but the structural elements that *persist* for several values of ϵ are true structural characteristics of X, while those that are short-lived are simply noise from the sample $\mathbb{X}$.
Our job is to attempt to quantify this notion of *persistence*.

Consider a sequence $0 < \epsilon_1 < \epsilon_2 < ... < \epsilon_N$. For each value ϵ_i, we have a chain complex $(\Delta^i_\bullet, d_\bullet)$, given by:

$$... \xrightarrow{d} \Delta^i_{k+1} \xrightarrow{d} \Delta^i_k \xrightarrow{d} ... \xrightarrow{d} \Delta^i_1 \xrightarrow{d} \Delta^i_0,$$

where Δ^i_k is the free abelian group generated by the k-simplices of $\check{C}(\mathbb{X}, \epsilon_i)$, and the boundary maps are the standard boundary maps of simplicial complexes.

Additionally, it should be clear that as simplicial complexes, whenever $\epsilon_i < \epsilon_j$, we have that $\check{C}(\mathbb{X}, \epsilon_i) \subseteq \check{C}(\mathbb{X}, \epsilon_j)$: if $B_{\epsilon_i}(x_0) \cap B_{\epsilon_i}(x_1) \cap ... \cap B_{\epsilon_i}(x_n) \neq \emptyset$, it is of course true that $B_{\epsilon_j}(x_0) \cap B_{\epsilon_j}(x_1) \cap ... \cap B_{\epsilon_j}(x_n) \neq \emptyset$ as well.
Therefore we have a map naturally induced by the inclusion, $\phi^i_k : \Delta^i_k \to \Delta^i_{k+1}$.
It is easily verified that $\phi^\bullet$ is a chain map on the sequence of chain complexes $\{(\Delta^i_\bullet, d_\bullet)\}_i$, i.e. ϕ and d commute:
If $\{x_0, ..., x_k\} \in \Delta^i_k$, then $\{x_0, ..., x_k\} \in \Delta^{i+1}_k$ and $\phi(\{x_0, ..., x_k\}) = \{x_0, ..., x_k\}$.
On the other hand, $d(\{x_0, ..., x_k\})$ is a linear combination of the $(k-1)$-simplices that constitute the large k-simplex. Since both $\Delta^i_\bullet$ and $\Delta^{i+1}_\bullet$ are simplicial complexes, then all of these $(k-1)$-faces belong to both Δ^i_{k-1} and Δ^{i+1}_{k-1}, so obviously $\phi \circ d(\{x_0, ..., x_k\}) = d \circ \phi(\{x_0, ..., x_k\})$.

This situation is reflected in the following commuting diagram:

$$\begin{array}{ccccccccc}
... & \xrightarrow{d} & \Delta^{i-1}_{k+2} & \xrightarrow{d} & \Delta^{i-1}_{k+1} & \xrightarrow{d} & \Delta^{i-1}_k & \xrightarrow{d} & \Delta^{i-1}_{k-1} & \xrightarrow{d} & ... \\
 & & \downarrow\phi & & \downarrow\phi & & \downarrow\phi & & \downarrow\phi & & \\
... & \xrightarrow{d} & \Delta^{i}_{k+2} & \xrightarrow{d} & \Delta^{i}_{k+1} & \xrightarrow{d} & \Delta^{i}_k & \xrightarrow{d} & \Delta^{i}_{k-1} & \xrightarrow{d} & ... \\
 & & \downarrow\phi & & \downarrow\phi & & \downarrow\phi & & \downarrow\phi & & \\
... & \xrightarrow{d} & \Delta^{i+1}_{k+2} & \xrightarrow{d} & \Delta^{i+1}_{k+1} & \xrightarrow{d} & \Delta^{i+1}_k & \xrightarrow{d} & \Delta^{i+1}_{k-1} & \xrightarrow{d} & ...
\end{array}$$

A chain map is easily shown to induce a map of homology groups.
In this case specifically, for any i, $\phi^* : H^i_k(\mathbb{X}) \to H^{i+1}_k(\mathbb{X})$ is defined by $\phi^*([x]) := [\phi(x)]$.
In general, for $i < j$, we define $\phi^*_{i \to j} : H^i_k(\mathbb{X}) \to H^j_k(\mathbb{X})$ by:

$$\phi^*_{i \to j} := \underbrace{\phi^* \circ \phi^* \circ ... \circ \phi^*}_{(j-i)-times}$$

Through this construction we can *keep track of homology classes* when jumping from $\Delta^i_\bullet$ to $\Delta^{i+1}_\bullet$, and see where each class is born and where it dies off.

If we only studied the homology of each $\check{C}(\mathbb{X}, \epsilon_i)$ individually, without considering the *chain map structure* we just discussed, then we would have no way of knowing whether the appearing homological classes are sampling noise or truly structural.
For example, if for i_0 and k_0 we found that both $\check{C}(\mathbb{X}, \epsilon_{i_0})$ and $\check{C}(\mathbb{X}, \epsilon_{i_0+1})$ had three k_0-dimensional homology classes, without applying the *tracking* tool of persistent homology, we would have no way of knowing if the three classes for i_0 had died and three new ones had been born; or if the three classes in $i_0 + 1$ were in fact survivors from the classes of i_0.
Without this *persistence* criteria, the *noise structure* is indistinguishable form the *true underlying structure* of the data.

In the literature on the subject, the usual visualization technique for Persistent Homology analysis is presented in the form of *barcodes*.
A *barcode diagram in dimension k* is represented on a plane where the horizontal axis represents the value of the parameter ϵ, and where along the vertical axis we have line segments (placed arbitrarily in the vertical direction) where each segment represents a k-dimensional homology class, starting at the value of ϵ where it was "born", and likewise ending where it "died off".
In this context, a vertical cut of a k-barcode diagram at ϵ_0 should cut exactly the amount of *barcodes* as k-homology classes $\check{C}(\mathbb{X}, \epsilon_0)$ has.

Obviously, we have a different barcode diagram for each value of k.

Figure 7 illustrates the barcode diagrams for $k = 0, 1, 2$ in the example from Figure 6.

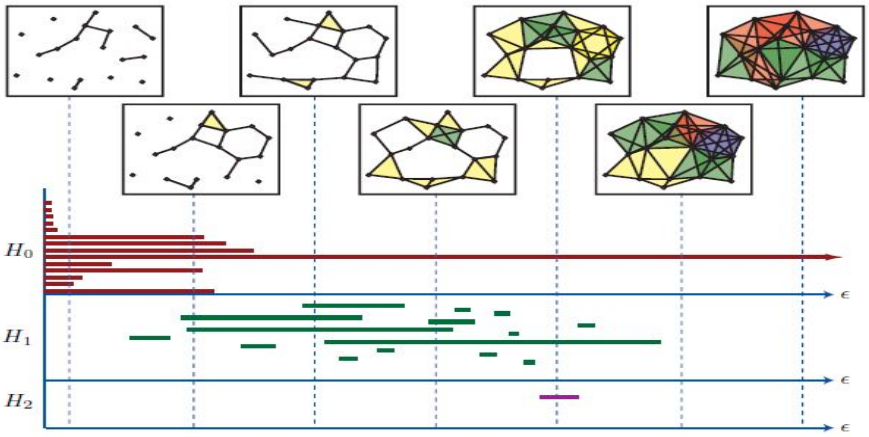

Figure 7: *The longer barcodes are indicative of underlying homological structure. From the evidence seen in the barcodes for different dimension k, we can assume that there is a single connected component, but can't truly decide on one 1-dimensional homology class as is expected.*

Returning to the example from Figure 4, the barcode diagram for grade 1 of the whole data set is shown in Figure 8.

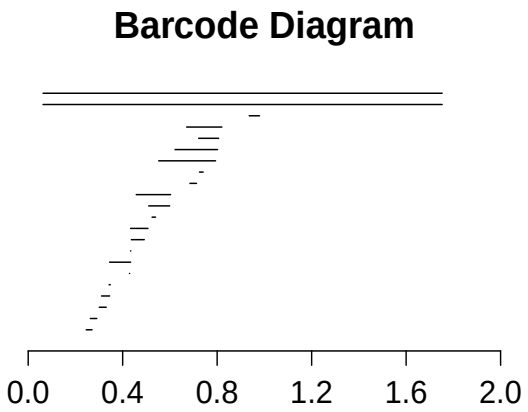

Figure 8: *The two stable lines obviously represent the two circles in the data, being born at* $\epsilon = 0.063$ *(the first value at which the circle appeared) and dying off at* $\epsilon = 1.75$ *(where the parameter is large enough to fill in the circles' interior and kill off the homology class).*

This evidence towards 2 homological classes of dimension 1 is also seen when plotting the barcode diagram of just the points in each circle, which would be sets in our discrete covering.

The application of the idea of persistent homology to the validation of *goodness* of discrete coverings is clear: a covering where each *cluster component* shows no homological structure for $k > 0$ is more likely to be *good*.
To the point, the barcode diagram for the covering shown in Figure 5 for the 2-circle data set shows no such homological structure, and therefore no *red flag* is raised warning us of the non-contractible nature of it's *cluster components.*

This validation technique can also be utilized for the finite intersections of sets in the covering, in the spirit of validating the hypothesis of the Nerve Theorem. However, this quickly becomes computationally very expensive, since one has to consider all possible combinations of finite sets.

Remark 2.10. This is a powerful theory. Discovering homological information from discrete data point clouds can be very revealing of the nature of the data observations, as we will point out with an example in Section 6.
The reader might ask why the persistent homology technique isn't applied to the whole point cloud at hand, instead of just the *cluster components* of the discrete cover. Certainly, a

technique that allows homological information to be discovered should have a more central role in a data mining investigation.
The reason is that the computation of the barcode diagrams is computationally very strenuous, its complexity increasing both with the size of the point cloud as with the dimension of the points when metric learning is desired. This usage for the technique is a way of introducing it into the problem of studying the topology of a point cloud in a computationally viable way.

3 TDA Pipeline

The **Topological Analysis Pipeline** is a visualization technique stemming from the ideas presented in the previous section. It is essentially a visualization of the 1-skeleton of $\check{C}^{\pi_0}(\mathbb{X}, \mathfrak{U})$ for a given covering $\mathfrak{U}$ and a given vectorization of the data $\mathbb{X}$. Since a 1-skeleton of a simplicial complex is basically a graph, this can be easily represented in $\mathbb{R}^2$.
While a lot of information from $\check{C}^{\pi_0}(\mathbb{X}, \mathfrak{U})$ is lost when using only the 1-skeleton, the visual representation allows the investigator to nevertheless understand the basic layout of a high dimensional data set, and identify potentially important subgroups and relationships inside the data that are probably invisible to other statistical methods like direct clustering, principal component analysis, etc.

To showcase the type of results that can be expected from a TDA analysis, let's begin with a couple of toy examples.

Toy Example 3.1. 120 points were sampled uniformly around the unitary circle in $\mathbb{R}^2$. Figure 9 shows the resulting point cloud:

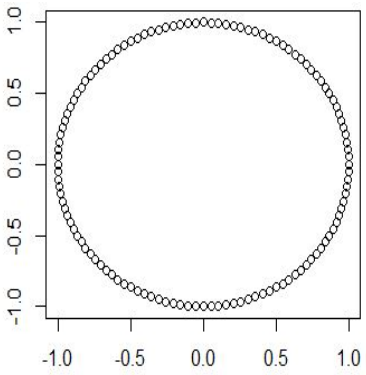

Figure 9:

When a TDA Pipeline method was applied to this toy data set and the corresponding TDA Graph produced, the result was the following:

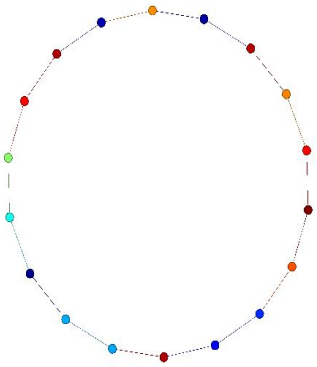

Figure 10:

The important *qualitative* feature of the underlying manifold to the point cloud, the circle, is recovered through the TDA Graph.

Toy Example 3.2. To the data set from the previous example we added 50 points uniformly sampled across the x-axis diameter of the circle. Figure 11 shows the resulting data point cloud:

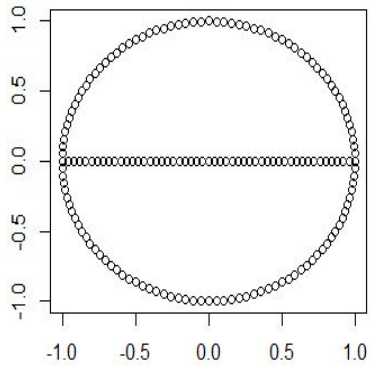

Figure 11:

The TDA Graph we produced was the following:

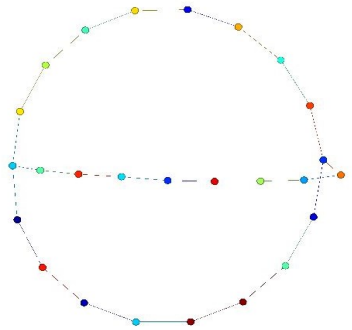

Figure 12:

Once again, the important *qualitative* aspects of *shape* and *homotopy type* are recovered through our methodology.

Now that we have seen the type of *topological information* that the TDA Pipeline attempts to recover in a couple of toy examples, lets show some examples of how this method is applied in real life data sets to discover *knowledge* from the raw information.
The following examples were taken from [LSL$^+$13]:

Example 3.3. The NKI cancer data set, consisting of gene expression levels for 1500 genes in 272 different breast tumors was applied the TDA Pipeline technique.
Figure 13 taken from [LSL$^+$13] shows the result they obtained:

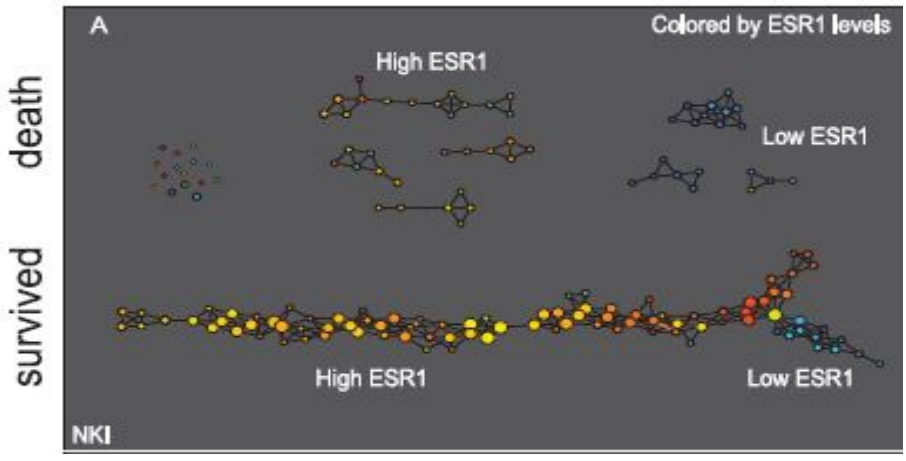

Figure 13: *The graph is colored by ESR1 levels.*

In medical folklore, it is well known that low ESR1 levels are usually related to a poor prognosis, but this *topologically identifiable* subgroup of survivors managed this with low ESR1

levels.
Since the feature vector consists of 1500 gene expressions which determine the topology of the data set, these must somehow determine why these patients survived.
Stop to consider the difference of these patients appearing in a topological structure of the data set versus them having appeared randomly in the graph. Random appearance would suggest that their survival was just a coincidence (not an uncommon explication in medicine), while this topologically structural appearance suggests that there is something in the nature of their gene expression data that determined their survival.

The answer to this question of what determined their survival can be looked for with all the machinery of statistics and applied mathematics; but it was TDA that suggested the question in the first place.

Example 3.4. Figure 14 (taken from [LSL$^+$13]) shows the **TDA Graphs** generated by two different covers (a "big" and a "small" cover respectively) on a point cloud data set vectorized from in-game statistics of basketball players:

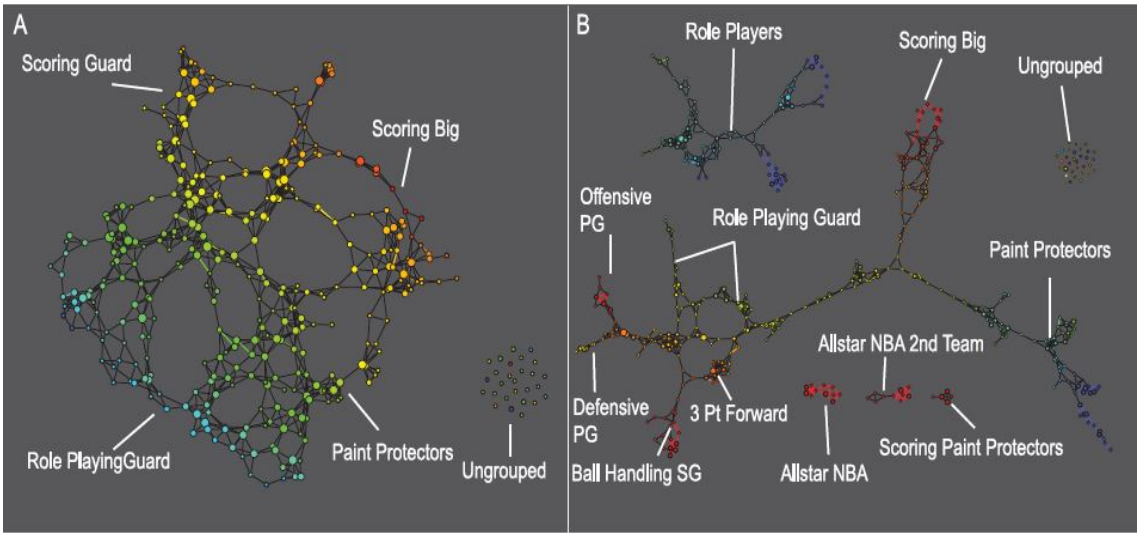

Figure 14: *The graph is colored by points scored.*

In [LSL$^+$13] it's suggested that through the result of the graph for the "smaller" cover, we can identify a host of different playing positions in the sport.

3.1 The Pipeline

In this subsection, we give a detailed account of the process through which the resulting graphs we just exemplified are constructed. We will explain each step and illustrate by highlighting the corresponding step in our first toy example.

First, the raw data must be *vectorized* and manipulated. This first step consists of the necessary *tinkering* of the data so that our method is applicable. For example, a usual manipulation the user must make is to normalize each feature of the data to ensure that the metric within the data is not dominated by the values of a feature that by its very nature has large values. For example, if our data has two features, one of fundamental importance whose values are between 0 and 1, versus another whose importance is less but whose values are between 1,000 and 2,000; then the metric of the problem will be dominated by this feature. Normalization will help us avoid these issues. Other previous steps can include feature selection, or the inclusion of binary or categorical features in an appropriate numerical way. The result should be a numerical data matrix, where each row represents an observation point as a vector in $\mathbb{R}^d$, and constitutes our discrete sample $\mathbb{X}$.

Now we need a method for constructing a covering $\mathfrak{U}$. In the TDA literature, this is achieved by applying what is called a *filter function*, $f : \mathbb{X} \to \mathbb{R}^d$, and using $\mathfrak{U} = \{f^{-1}(\mathfrak{V}_\alpha)\}_\Lambda$ where $\mathfrak{V}$ is a cover of $\mathbb{R}^d$.
The choice of the filter function is clearly crucial. It determines the *goodness* of the discrete cover $(\mathbb{X}, \mathfrak{U})$, and therefore determines whether the resulting graph will indeed represent the 1-skeleton of a simplicial complex homotopically equivalent to the supposed underlying space. With this in mind, most applications choose statistically sensible functions. The basketball players example uses the projection onto the first two principal components, whose arrival space is $\mathbb{R}^2$. The arrival space in the breast cancer example is also $\mathbb{R}^2$, but with a slightly different design. The first component of this filter function is the L-infinity centrality (a statistically relevant choice), while the second component is a numerical representation of the binary feature *death*, which allows the resulting graph to be separated between patients who died and patients who survived, and study each occurrence independently. This function generates a cover $\mathfrak{U}$ such that $\neg\exists\alpha, x, y$ with $x, y \in \mathfrak{U}_\alpha$ such that x lived and y died.
Other filter functions can be applied, like a ranking for the observations, or the projection onto a specific feature if the investigator wishes to investigate the geometry along this feature.

In our toy example, the filter function is projection onto x.

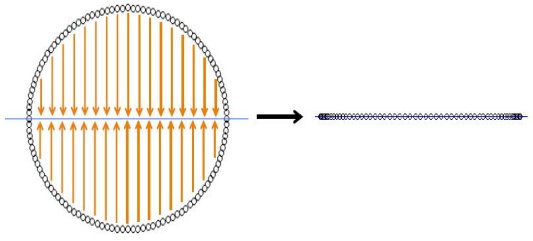

Figure 15:

Notice that f is not the only relevant choice when determining $\mathfrak{U}$, but also the choice of $\mathfrak{V}$ (we will refer to $\mathfrak{V}$ as the *target cover*). Figure 16 shows the role of the cover $\mathfrak{V}$.

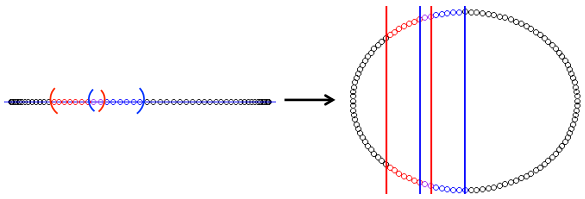

Figure 16: *The red and blue points constitute two overlapping open sets, whose intersection are the purple points.*

Later on we will give details as to how these two choices were made in our implementation.

Once we have $\mathfrak{U}$, we need to construct $\{\mathfrak{U}_{(\alpha,\xi)}\}_{(\Lambda,\Xi)}$. This is done by clustering each set $\mathfrak{U}_\alpha$ and indexing by Ξ, as we have already explained. Figure 17 shows the step in question as seen in our toy example.

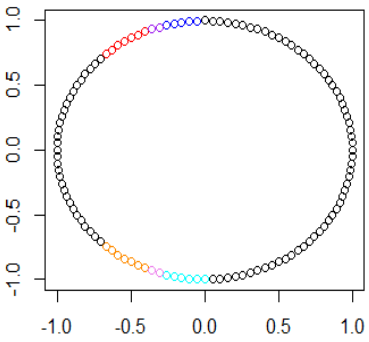

Figure 17: *The red bin was divided into two clusters, the red and orange points, while the blue bin was divided into the blue and cyan points.*

Finally, we construct the **TDA graph** by making the abstract graph whose vertices are the set (Λ, Ξ), and where (α_1, ξ_1) and (α_2, ξ_2) share an edge if $\mathfrak{U}_{(\alpha_1,\xi_1)} \cap \mathfrak{U}_{(\alpha_2,\xi_2)} \neq \emptyset$.

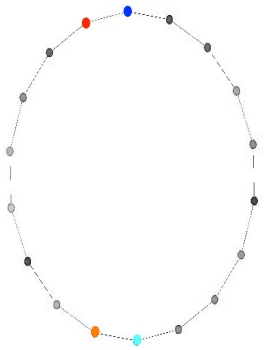

Figure 18: *The overlapping clusters are joined by an edge.*

A flowchart summarizing the steps of the pipeline is shown below.

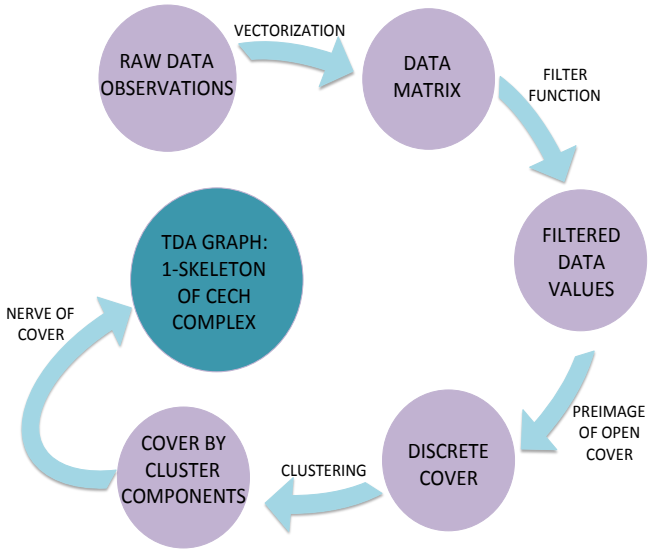

Figure 19:

4 Pipeline Methods

In this section we go over some of the methods involved in the build up to our TDA Graphs.

In Section 4.1 we speak of metric learning by side information (labels) algorithms and how we use them to pre-process our data in the hope that we will gain a more meaningful metric for our vectorization. The outcome of these algorithms will be a linear transformation on our vectorized data such that the image has the learned metric as its euclidean metric.
In truth, a full rank linear transformation shouldn't truly affect the *topological* outcome of TDA, which precisely is conceived to be "metric and coordinate-free", but the inclusion of clustering methods into the technique inevitably brings the specific metric into the fray.

In Section 4.2 we showcase the filter functions typically used in TDA literature, and specify how they are used in tandem with a *target cover* in the euclidean arrival space of the filter to generate the discrete covering of our point cloud data set, that is usually referred to as "bins" in the literature.

Finally, in Section 4.3 we discuss our own implementation for deciding the number of clusters in a data set, which we have called the *Modified Gap Statistic*.

Sections 4.1 and 4.3 can be read independently from the rest of the document, but given their central role in our implementation of the TDA Pipeline, we have decided to include them.

4.1 Distance Metric Learning

At this point, we might find ourselves with a *matrix representation* of our data, where each row of the matrix represents the numerical representation of the features at our disposal for each observation. In this sense, each observation can be seen as a point in a euclidean space, whose dimension is the number of columns of the matrix.
However, nothing ensures us that the euclidean metric of this space represents a meaningful concept of distance between our points/observations. Since our investigation will inevitably depend on the distances between points (for clustering, *filtering*, etc), we wish to have an acceptable degree of certainty that the distances between our points are meaningful in the context of the problem at hand.

With this need in mind, our objective is to *learn* a new metric to be implemented in the problem. Specifically, we wish to learn a **Mahalanobis** metric, which is defined by:

$$D_M(x, y) := \sqrt{(x - y)^t M (x - y)},$$

with M any positive semidefinite matrix. It is easy to show that if M is of full rank, D_M is a metric, otherwise, it is a pseudometric (i.e., $d(x, y) = 0 \leftrightarrow x = y$ doesn't hold).

Alternatively, we could also propose a metric defined by:

$$\hat{D}_L(x, y) = \| L(x - y) \|,$$

where L is any matrix (here we view it as a linear transformation), and $\| \cdot \|$ is the euclidean distance in the target space. Again, if L is of full rank, $\hat{D}_L$ defines a metric and a pseudometric otherwise.
The following simple results from linear algebra relate these two families of metrics:

Proposition 4.1. For M and L matrices with $M = L^t L$:

1. M is positive semidefinite.
2. $D_M = \hat{D}_L$.
3. L uniquely determines M. On the other hand, for any L' such that $M = (L')^t L'$, we have that $\hat{D}_L = \hat{D}_{L'}$.

Proposition 4.1 shows that when looking for our optimum metric/matrix, we can either use the unconstrained optimization on L as suggested by $\hat{D}_L$ or use semidefinite optimization on M as suggested by D_M.

Now, in order to be capable of evaluating when a metric is or isn't meaningful in the context of a problem, we must count with additional information indicating which points we wish to be close to each other, or which points we want to be far from each other. This information could come in the form of *labels*.

Remark 4.2. In our particular data set, each player comes *labeled* with a playing position, with four different labels in total: Goalkeeper, Defender, Midfielder and Striker.
We can also consider our observations labeled by team and result of the match.

In [XJRN02], the following optimization problem is suggested:
Let $\mathbb{X}$ be the set of samples, and $S \subseteq \mathbb{X} \times \mathbb{X}$ be such that $(x_i, x_j) \in S$ if they are *similarly labeled.* Lets also write $D := S^c$.

$$\begin{boxed}
\min_M \sum_{(x_i,x_j)\in S} (D_M(x_i, x_j))^2 \quad \textbf{with restrictions:}

\textbf{(1)} \ \sum_{(x_i,x_j)\in D} D_M(x_i, x_j) \geq 1

\textbf{(2)} \ 0 \preceq M
\end{boxed} \tag{4.3}$$

By what we said before, it should be clear that this problem is equivalent to:

$$\boxed{\begin{array}{l}\min_L \sum_{(x_i,x_j)\in S}(\hat{D}_L(x_i,x_j))^2 \quad \textbf{with restriction:} \\ \\ \textbf{(1)}\ \sum_{(x_i,x_j)\in D} \hat{D}_L(x_i,x_j) \geq 1\end{array}} \tag{4.4}$$

Depending on the optimization technique used, one formulation should be more convenient than the other.
The constant 1 in the restriction of the problem is irrelevant, since for any other constant $c > 0$ the solutions are equivalent up to multiplication by c. This restriction is important however since it ensures that the optimum isn't achieved by *collapsing* the whole data set into a single point (i.e. $L = 0$ or $M = 0$), which would trivially be a solution.

However, this approach has two significant flaws:
On one hand, the computational cost is usually prohibitive. S and D are usually extremely large in most problems.
Also, the approach is ill-equipped to handle with problems where categories are multi-modal (i.e., with several means), since it will then attempt to bring together points that don't belong close by.

Remark 4.5. In the context of our data set, this approach seemed ineligible on both counts. The computational cost was too large, since S had upwards of 30,000 entries, while D had upwards of 65,000 (this for the individual player approach with the team performance approach being actually much larger). Also, it seems fair to assume that our categories are multi-modal, since for example both center-backs and full-backs are categorized as *defenders*, but one would expect their feature values to be distinguishable. Also, when bringing together matches by the same team, one could expect two victories from this team to be close by, but not necessarily close to two defeats.

LARGE MARGIN NEAREST NEIGHBOR DISTANCE LEARNING
The approach we selected to apply in our investigation was proposed by Weinberger and Saul ([WBS05]). It is originally intended to learn a metric which improves the training error of *k-nearest neighbor classification*, but is perfectly applicable to our problem.

The main idea is to learn a metric in a similar way to 4.4, except that instead of looking to bring together all similarly labeled points, we only expect to bring together a point with some predefined *target neighbors*. This approach can be implemented in a way that substantially reduces computational cost, and at the same time is capable of dealing with broad categories with multiple means.
The choice of *target neighbors* for a point x_i is done by fixing k, and choosing the k-nearest (in the euclidean space in which the points lie) similarly labeled points.

The optimization problem 4.4 is then reformulated as follows:
Define $i \rightsquigarrow j$ to indicate that x_j is a target neighbor of x_i, that is to say that x_j and x_i have the same label, and x_j is one of the k-nearest neighbors to x_i with the same label.

Remark 4.6. Notice that the relationship "$\rightsquigarrow$" is not reflexive, since $i \rightsquigarrow j$ doesn't imply $j \rightsquigarrow i$.

Define $T := \{(i,j) : i \rightsquigarrow j\}$, and $I = \{(i,j,l) : (i,j) \in T \text{ and } (i,l) \notin T\}$.

The formulation that [WBS05] suggests is:

$$\begin{boxed}
\min_M \mu \sum_T (D_M(x_i, x_j))^2 + (1-\mu) \sum_I \zeta_{i,j,l} \quad \textbf{with restrictions:}

\textbf{(1)} \; D_M(x_i, x_l)^2 - D_M(x_i, x_j)^2 \geq 1 - \zeta_{i,j,l}

\textbf{(2)} \; \zeta_{i,j,l} \geq 0

\textbf{(3)} \; 0 \preceq M.
\end{boxed} \tag{4.7}$$

The variables $\zeta_{i,j,l}$ are *slack variables*, and once again the constant 1 in the first restriction is irrelevant; it establishes a radius around a point x_i inside of which *impostors* (x_l with $(i,l) \notin T$) are *punished* by a slack variable in the objective function, but the actual size of the radius is unimportant.
The value of μ is a *weight* attributed to the terms in the objective function, i.e. it weighs the importance of bringing together target neighbors versus pushing away impostors. Usually $\mu = 0.5$ works fine.

Lets take a look at an example that illustrates the *multi-modal* quality of this approach:

Example 4.8. The data set in Figure 20 was generated.

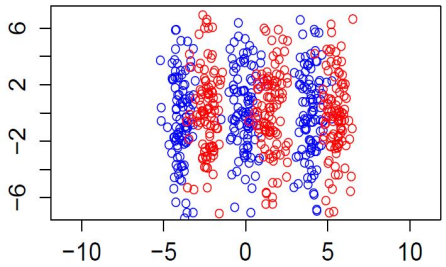

Figure 20: *The labels are represented by red and blue categories*

We expect that the metric in the solution should *weigh* the distance on the x-axis as more relevant when distinguishing between categories.

Figure 21 shows the same data set plotted after the linear transformation Lx where L is such that $L^tL = M$ (recall Proposition 4.1):

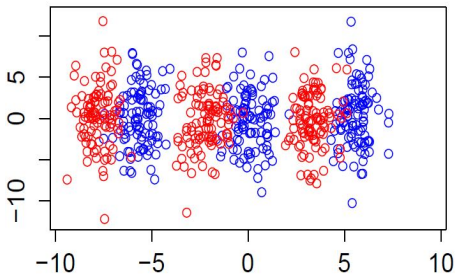

Figure 21:

We can appreciate the applicability of the method in this case where the categories weren't a single distinct point cloud, but rather several clouds.

Finally, it's worthy of mention the computational soundness that the approach allows.

The reason this computational advantage exists is that for almost all triplets $(i, j, l) \in I$, $D_M(x_i, x_j) < D_M(x_i, x_l)$, so no *violation* by x_l as an *impostor* within the neighbors of x_i must be monitored.
This results in the optimum lying along $\zeta_{i,j,l} = 0$. Since this is the case for almost all triplets $(i, j, l) \in I$, there are truly very little active restraints in 4.7, and the vector ζ is highly sparse.

Weinberger and Saul implemented a special purpose solver (please reference [REF]) that exploits this fact.
The main idea is to iteratively solve optimization problems like 4.7, except that only a fraction of the variables ζ are used as constraints. In each iteration, the fraction of considered restraints is augmented and the solution of the previous step is utilized as the starting point. This strategy, coupled with the massive reduction of terms in the *non-slack* sum term of the objective function in 4.7 in comparison with the one in 4.3 and 4.4 greatly reduce the computational cost in this approach.

More details on the special purpose solver implemented can be found in [WBS05].

Remark 4.9. We have mentioned before that it is usual to normalize the data so as to avoid the metric of the problem to be undesirably dominated by a feature whose values are naturally large. We might ask ourselves whether the metric learning algorithm makes the

normalization process redundant. At first glance, it seems that the metric learning will handle the relevance of the features with large values to make sure the metric of the problem isn't affected by this characteristic, which was precisely the objective of normalizing. However, it IS important to normalize before running the metric learning algorithm, since the algorithm chooses the targets neighbors using the original euclidean metric of the problem, so previous normalization might help us make this choice better.

4.2 Filter Functions and Coverings

As we already commented, the choice of filter function is crucial. The nature of the filter function determines the nature of the cover of our point cloud, and therefore determines whether $\check{C}^{\pi_0}(\mathbb{X}, f^{-1}\mathfrak{U})$ provides meaningful topological information about the data.

In this section, we will go over the filter functions we have used to generate the covers for our data point clouds, and see how the nature of their covers integrates into the methodology of the problem.

1. **Principal Components**

   As a filter function we can use the projection onto $\mathbb{R}$ or $\mathbb{R}^2$ by either the first or the first two principal components.

2. **Data Depth**

   Data depth is a notion that attempts to quantify the *centrality* of a point inside a point cloud, as well as a relative density. The precise formula is:

   $$DD(x) = \frac{1}{\#(\mathbb{X})} \sum_{x' \in \mathbb{X}} \| x - x' \|_2$$

3. **L-Centrality**

   L-Centrality is an alternative approach to measuring the *centrality* of points within their point clouds. Its formulation is:

   $$LC(x) = \max_{x' \in \mathbb{X}} \| x - x' \|_2$$

4. **Data Density**

   Our formulation for this filter is as follows:

   Let k be a fixed positive integer. For a point $x \in \mathbb{X}$, $\delta(x) := \frac{1}{\|x - x_k\|_2}$ where $x_k \in \mathbb{X}$ is the k-th nearest point to x.

   It should be clear that if $\delta(x)$ is large it means that $\| x - x_k \|_2$ is small, i.e. the k-nearest neighbor is close by and therefore there is a greater concentration of points around x than around points z where $\| z - z_k \|_2$ is larger, and therefore $\delta(z)$ is smaller.

Once we have applied a filter function, we must generate a covering of the target space that we will "pull back" towards our space via preimages.
The standard way is to use a *grid* in the resulting euclidean space, where the *cells* are overlapping.
The important choice is how to determine both the size of the *cells* and the *overlap.*

We have implemented two ways of doing this:

1. **Length Parameters**
   We give an explicit length for each cell, and an explicit length for the invasion into the adjacent cells.
   Of course, both the length of the cells and the length of the overlap must be given as vectors in the dimension of the filter function's target space, where each entry determines the length of the cell in each direction.

2. **Percentage Parameters**
   We can also give the length of the cell as a *percentage* of the total range of the function, and the overlap as a *percentage* of each cell.
   For example is f arrives in $\mathbb{R}$ and $\max_{\mathbb{X}} f(x) = b$ and $\min_{\mathbb{X}} f(x) = a$, then a length parameter of 5% would mean that each cell is of length $\frac{b-a}{20}$, while an overlap parameter of 20% would mean each cell is *fattened* to invade its adjacent cells by $\frac{b-a}{100}$.
   Similarly, these parameters must be given in a vector of the dimension of the target space.

Figure 22 illustrates the process:

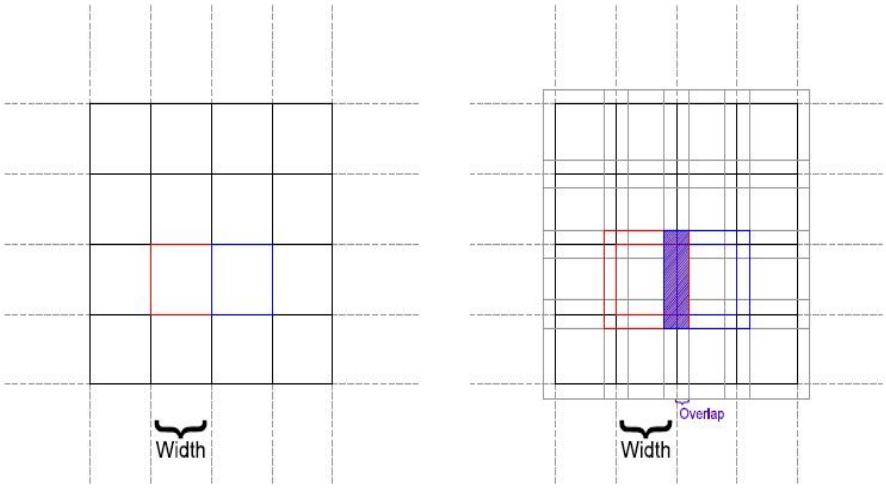

Figure 22:

Lets take a look at how the coverings of our functions work with a toy data set:

Toy Example 4.10. The following data set was generated:

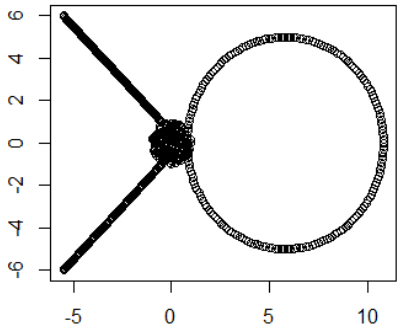

Figure 23:

The qualitative topological features of the point cloud we would expect to recuperate would be the two *flares*, the uniform *cloud* of the center and the *loop*.

Figure 24 shows the performance of the projection onto the first principal component as a filter:

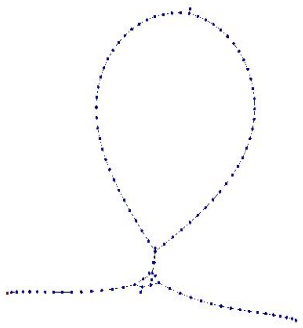

Figure 24:

The filter function clearly generates a good cover of the space.

Before we get carried away with our success, lets take a look at the *Data Depth* filter.
The results for this filter, albeit "unsuccessful", are very telling of the influence of parameters in the result, and show exactly what can go wrong.
Figure 25 shows the result for this filter function where the *target space cover* was determined by percentage parameters of 2% and 25%.

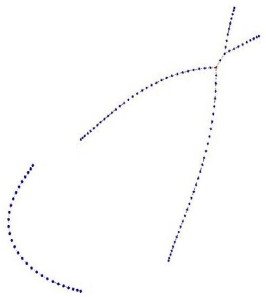

Figure 25:

The method appears to have failed in closing the *loop*, showing instead 4 distinct flares and a broken off piece.
However, a (painfully) careful analysis establishes that what truly happened was that the *width percentage* parameter was so small that it failed to *overlap* two naturally connected "line segments" of discrete points, i.e. failing to connect 2 nodes which should be connected.
Figure 26 illustrates this scenario:

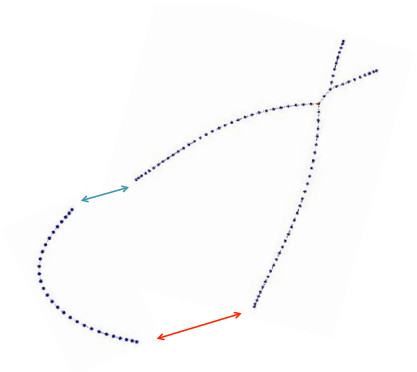

Figure 26: *The overlap in two segments of the loop was not large enough, and the signaled pieces resulted separated even though we expect for them to be connected.*

A larger parameter should therefore solve this "connectivity problem".
Figure 27 shows the result for parameters 2.22% and 25%:

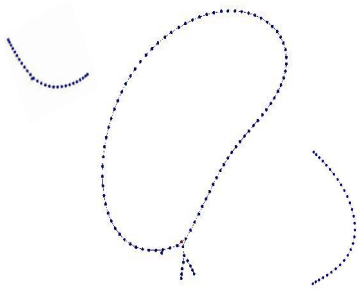

Figure 27:

This time, the *loop* has in fact appeared, but the graph is again disconnected and the two *flares* aren't connected to the loop.
What went wrong this time?
Figure 28 shows the first bin in the pipeline for these parameters, and the 5 clusters into which it was broke into:

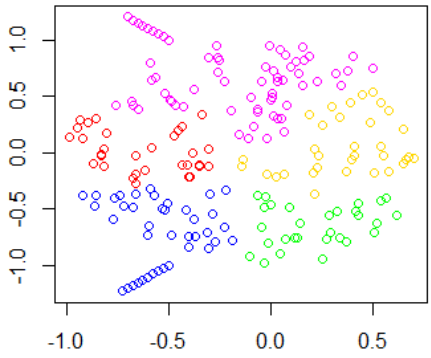

Figure 28:

This *breaking up* of the central cloud makes each node of the cloud disconnected to the others.

Since each *flare* and the *loop* join the *cloud* from a different cluster (the loop joins in the gold cluster, while the flares join in the violet and blue clusters), they end up disconnected from each other (see Remark 4.11). The smaller parameter of 2% width took the first bin as a single cluster, avoiding this issue.

The covers generated by *LC Centrality* behave in a very similar way to those of *Data Depth* so the same problems are encountered; particularly the tough balancing act between a parameter small enough to not partition the central cloud, and large enough to not break up the graph.

Finally, Figure 29 shows the extremely poor performance of *Data Density*:

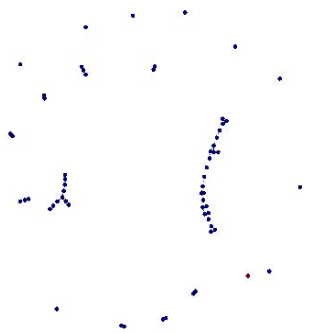

Figure 29:

This is simply due to the fact that the loop and flares were sampled as a uniform sequence, so the Density function is constant in large portions and the bins are pretty much useless.

Remark 4.11. The partitioning of the central *cloud* into 5 clusters that we saw in Example 4.10 when using the Data Depth function seems unnatural.
A choice of a single cluster certainly seems more natural, and would effectively solve the problem we encountered.

This problem is typical in clustering applications. Clustering algorithms are usually competent when recognizing well separated clouds of points, but there are many *shapes* of clouds where they are a bit at a loss. A *circular* point cloud, or a *long continuous band* point cloud are difficult shapes for clustering algorithms to handle.
If we have two distinctly separated circular point clouds, like in Figure 4, the clustering algorithm will usually make the sensible choice. However, when we have a single circular point cloud we are vulnerable to strange choices like the choice of 5 clusters we just encountered.

Since the Data Depth and L-Centrality functions' nature applied to the example's data set generates circular open sets, a poor performance can occur for some parameters.

This in no way establishes a ranking of our filter function candidates in which PCA comes out victorious. PCA also has trouble with some shapes within the data. For instance if our data set has three directions of high variance that are somewhat orthogonal, PCA's open sets will be tough to handle for clustering algorithms.

We have highlighted some important difficulties that we encounter when performing the Pipeline with certain filter functions and target cover parameters.
It is important for the investigator to understand them and keep them in mind so he can successfully avoid them when applying the technique to real life data sets.

4.3 Clustering the Bins

The clustering algorithm we used in our implementations was complete-linkage hierarchical clustering.

A hierarchical clustering structure is a sequence of partitions of a data set, where the first (or last) partition is the completely trivial n-cluster partition with each point being its own cluster, while the last (first) is the 1-cluster partition, with all points conforming a single large cluster.
The scheme is called *agglomerative* when we begin with n clusters and in each step *merge* the two in a certain definition *closest* clusters into a single cluster.
In complete-linkage, the notion of distance between two clusters X and Y is defined as $D(X, Y) := \max_{x \in X, y \in Y} d(x, y)$.

Most statistical computing environments like R or MatLab have ready made implementations for hierarchical clustering.
These methods usually return a *dendogram*, which is a representation of what points belong to each of the k-clusters in the $(n - k)$-th step of the hierarchical sequence.

The question of true importance is how to decide the correct value of k.

4.3.1 Choosing the Correct Number of Clusters

Suppose you have a data set of n points, and two different partitions of the points into k clusters.

How do you measure which one is better?

Intuitively, one expects that points that naturally belong in the same clusters should be closer together than those who don't belong together. A criterion which measures in some way the distances of points inside the same clusters should then be lower in the better partition.

These criteria are called *scatter criterion functions*.

The following function is the one we will use in our calculations to determine the best partition *and* the best k:

Definition 4.12. For a data set $\mathbb{X}$ and a k-partitioning $\mathfrak{C}^k = \{\mathfrak{C}_1^k, \mathfrak{C}_2^k, ..., \mathfrak{C}_k^k\}$ with $\mathfrak{C}_i^k \subset \mathbb{X}$ and $\mathfrak{C}_i^k \cap \mathfrak{C}_j^k = \emptyset$ whenever $i \neq j$, we define the **within cluster scatter** of $\mathfrak{C}^k$ by:

$$W(\mathfrak{C}^k) = \sum_{i=1}^{k} \sum_{x \in \mathfrak{C}_i^k} \parallel x - \hat{m}_i \parallel,$$

where $\hat{m}_i$ denotes the mean of $\mathfrak{C}_i^k$.

The following methodologies attempt to use this function to determine the best value for k where we obtained partitions $\mathfrak{C}^k$ for each value $k \in \{1, 2, ..., n\}$ by complete hierarchical clustering.

From here on, we assume we have that hierarchical scheme fixed, and unequivocally denote $W(k) := W(\mathfrak{C}^k)$.

The Gap Statistic

A standard method for choosing the correct number of clusters intrinsic to a set of data was proposed by Stanford professors Tibshirani, Walther and Hastie in their 2001 paper [REF GAP], which they called the *gap statistic*.

It is clear that $W(k)$ is monotonically decreasing with k. It is natural to suppose, however, that when a natural *clustering* of the data occurs at k groups, $W(k)$ should decrease noticeably in comparison to $W(k+1)$, and should flatten thereon. Let's exemplify this heuristic reasoning with a couple of examples:

Example 4.13. We randomly generated four distinct multivariate normal point clouds of 100 points each, with means $\langle 5, 0\rangle$, $\langle 0, 5\rangle$, $\langle -5, 0\rangle$ and $\langle 0, -5\rangle$ respectively. Figure 30 shows the result.

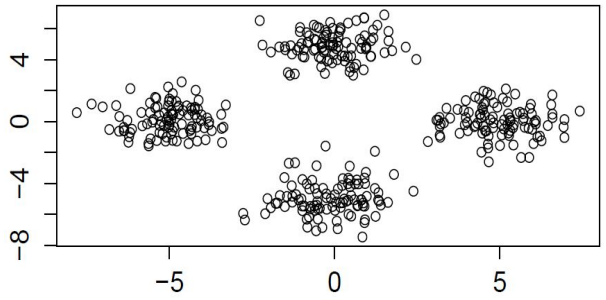

Figure 30:

Next, we calculated $W(k)$ for $k \in [1, 10]$. Figure 31 shows the plot of $W(k)$ against k for each value.

Figure 31:

Notice that, as suggested, the graph *flattens* noticeably after $k = 4$, where a natural clustering of the data is occurring.

Example 4.14. Next, we randomly generated eight distinct multivariate normal point clouds, of 50 points each. The means were $\langle -5, 50\rangle$, $\langle 5, 50\rangle$, $\langle 45, 0\rangle$, $\langle 55, 0\rangle$, $\langle 5, -50\rangle$, $\langle -5, -50\rangle$, $\langle -45, 0\rangle$ and $\langle -55, 0\rangle$ respectively. Figure 32 illustrates the result.

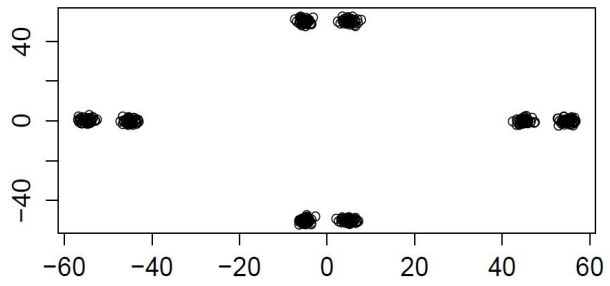

Figure 32:

Notice that now we have natural clustering structure occurring at $k = 4$ and $k = 8$, and again we would expect for this to somehow be expressed in the graph of $W(k)$. Figure 33 shows this graph for $k \in [1, 15]$.

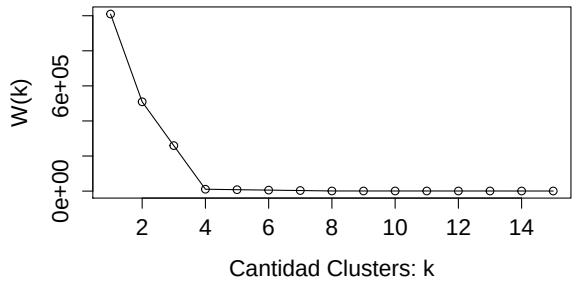

Figure 33:

It is not immediately evident that something is occurring at $k = 8$, but graphing only from $k = 6$ onward and re-scaling the range of the $W(k)$-axis, we obtain:

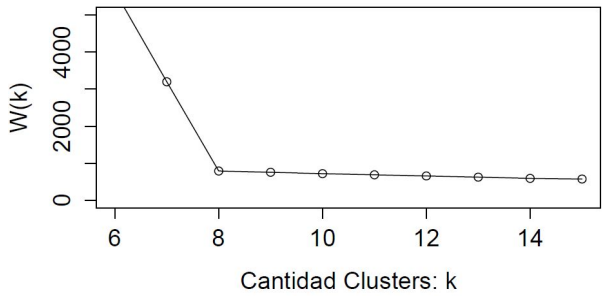

Figure 34:

These two examples have supported the heuristic claim that the graph of $W(k)$ should flatten when a natural clustering structure exists. The objective now is to mathematically formalize a procedure to identify these *elbows*.

We begin by observing the following property: consider a uniformly distributed data set, whose support is the range of our observed data set, and whose sizes coincide. In the context of our Example 4.13, we generate 400 uniformly distributed points whose support is $[-7.5, 7.5] \times [-7.5, 7.5]$. Figures 35 and 36 show the comparison between the plot of $W(k)$ and $log(W(k))$ respectively against k for this data set and the one from Example 4.13.

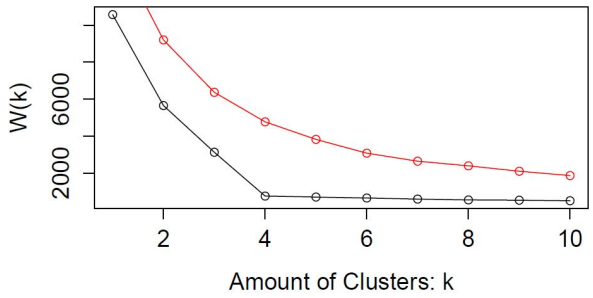

Figure 35:

This graph validates our claim that the *elbow* is indicative of underlying clustering structure, since the uniform data doesn't present these *elbows*.

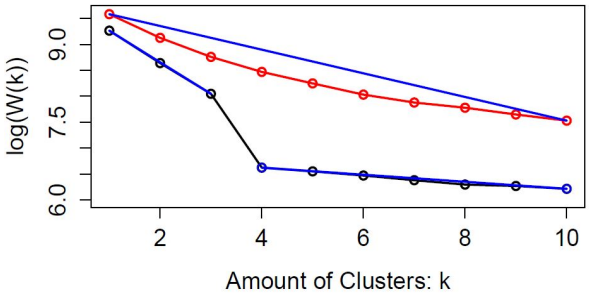

Figure 36:

Now, the basic presumption is that $log(W(k))$ should be more or less linear between *meaningful* values of k, at least when only small values of k are considered. In the data of Example 4.13, the *rate* (the slope of the linear approximation) at which $W(k)$ decreases *before* $k = 4$ is considerably higher than the smoothed out rate at which the uniform data's curve decreases. On the other hand, the rate at which $W(k)$ for the 4-cluster data decreases *after* $k = 4$ should be slower than that of the uniform data. This is easy to see since they both are 0 when $k = 400$, and $W(k)$ is always less for the 4-cluster data.
If these assumptions hold true, then it is easily observed that by defining:

$$Gap(k) := log(W(k)_{uniform}) - log(W(k)_{4-cluster}),$$

the highest value should be obtained at $k = 4$.
Indeed, Figure 37 proves us correct.

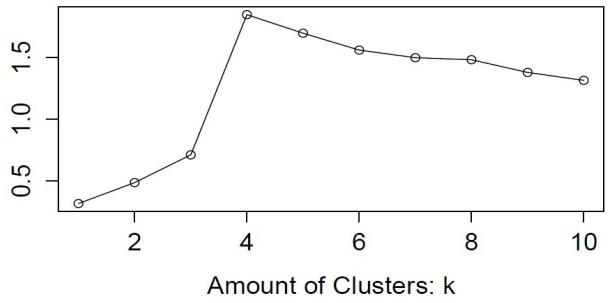

Figure 37:

What is proposed by the Stanford professors in order to dispose of the reliance of the generation of a *null reference* distribution, as the uniform one we have used to illustrate the heuristics of the choice of clusters, is to define the gap statistic by:

$$Gap(k) = E_n^* \left(log(W(k))\right) - log(W(k)),$$

where E_n^* denotes the expected value under n samples generated by a reference uniform distribution.

The *Modified* Gap Statistic

The Gap statistic as proposed by Tibshirani, Walther and Hastie works well for some verifiable examples, like our toy examples from section 2, where it chose 2/3 clusters for all the bins except the two bins at either end of the data, where it chose 1 as is expected.
However, the Gap statistic exhibits a couple of systematic flaws. Since our investigation is not verifiable, we believe these flaws could generate undesired biased results.

Lets expose the first of its flaws by revisiting Example 4.14.
We already discussed how there is a natural clustering structure in $k = 4$ and $k = 8$. However, one is inclined to suggest that $k = 4$ is the *most natural* choice of clustering, since the 4 larger clusters are much clearly separated in comparison to the 8 smaller ones.
Figure 38 demonstrate the quantities involved when calculating the gap statistic for this data set.

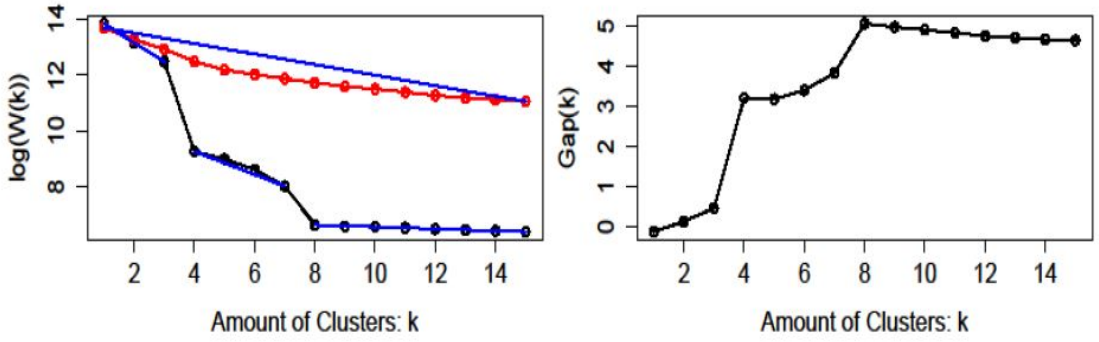

Figure 38:

As we can see, the gap statistic would favor $k = 8$ over $k = 4$.
Further evidence to point towards the biased-ness of this behavior can be shown with an extreme example. Figure 39 show the gap analysis for a similarly structured data, except that now the 4 large clusters are located at $\langle 150, 0\rangle$, $\langle 0, 150\rangle$, $\langle -150, 0\rangle$ and $\langle 0, -150\rangle$, while the 8 smaller clusters within each of these is separated by 5 units.

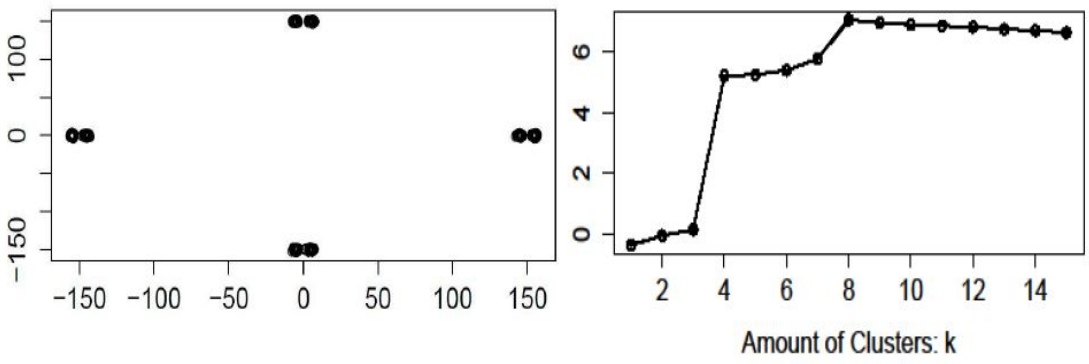

Figure 39:

Again, the gap statistic chose $k = 8$ over a more natural $k = 4$. In fact, $k = 4$ doesn't even generate a local maximum of the Gap curve. If we were to truncate the curve (as is usually done) at $k = 6$, the gap analysis would prefer 6 clusters over 4.
We have exposed one of Gap's weak spots: it is ineffective when the data has a *natural clustering structure at more than one value of* k, and it is biased towards the largest of said values of k.

In the context of our investigation, this biased behavior towards the largest k with natural structure rather than the *most natural* structure can be unfavorable, especially since our data is high dimensional the method could tend to *over-partition* the natural clusters in the data.

Lets take a look at a second important flaw:

Example 4.15. We generated a data set of 4 well separated clusters, and one single outlier lying near each of the clusters. The large clusters are separated by 50 units while the outlier lies just 5 units separated from each cluster. Figure 40 plots the data.

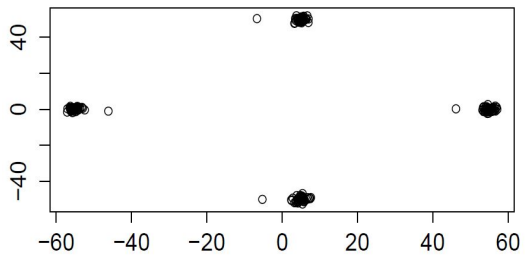

Figure 40:

Figure 41 shows the *gap* curve for this data set.

Figure 41:

The difference between $Gap(4)$ and $Gap(8)$ is difficult to perceive visually, but gap would in fact prefer $k = 8$, showing it's second flaw: extreme sensibility to outliers in the data.

As an alternative, we propose the following modification:

$$Gap_{mod}(k+1) = Gap(k+1) - Gap(k) \tag{4.16}$$

The heuristic reasoning behind why this would work is that while *Gap* tends to choose the value of k after which $log(W(k))$ is the *flattest* (i.e. the largest k with structure), it seems reasonable to choose k as the value where $log(W(k))$ *flattens* the most. Accepting a *linearity* of $log(W(k))$ between *meaningful* values of k when we consider only relatively small values of k, and observing that with each *meaningful* k the *negative slope* is decreasing (i.e $log(W(k))$ is becoming flatter), then this should be obtained when $Gap_{mod}(k)$ is the largest.
Additionally, this reasoning suggests a *cheaper* alternative that should work just as well when considering only a small amount of possible number of clusters. The approximated linearity of the $log(W(k))$ curves for both distributions allows us to assume that $E_n^*(log(W(k+1))) - E_n^*(log(W(k)))$ is constant for all considered values of k, and hence:

$$Gap_{mod}(k+1) = log\left(\frac{W(k+1)}{W(k)}\right) \tag{4.17}$$

should be highest at the same point that 4.16 is highest, obtaining a computationally lighter alternative to choosing the correct number of clusters.

Remark 4.18. *Notice that our modified gap method would be incapable of recognizing when the data only has 1 cluster (which is of course an important case in TDA), so in practice one should always test for 1 cluster with another methodology, and then if evidence points towards more than 1 cluster, use modified gap.*

Lets put out method to the test with some examples:

Example 4.19. Let's retake Example 4.14, where we saw that *gap* preferred $k = 8$ over our preferred $k = 4$. When we graph $Gap_{mod}(k)$ against k we obtain:

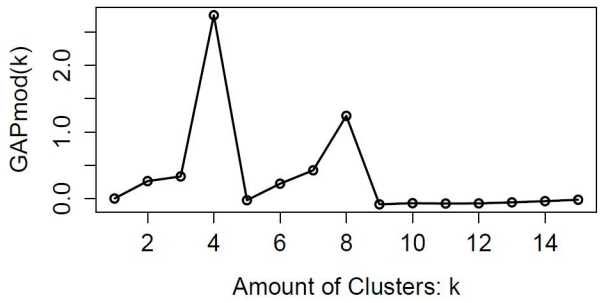

Figure 42:

Example 4.20. Recall the extreme example (Figures 39) of 4 large clusters separated by 150, en 8 smaller sub-clusters separated by 5. The *modified gap* analysis for this data set is shown in Figure 43:

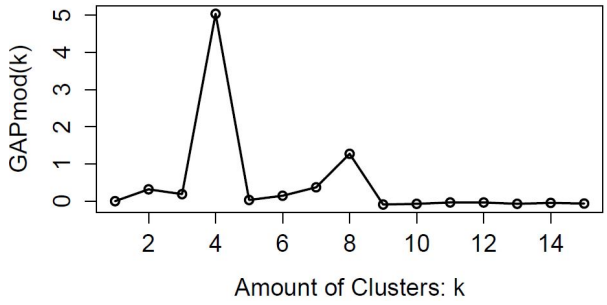

Figure 43:

These two examples show some important virtues of our modification. First, we are now choosing our preferred partition of $k = 4$ in both cases, while *gap* chose $k = 8$.
Also, $k = 4$ and $k = 8$ are the two largest values in both cases, and are both local maximums, while in *gap* $k = 4$ wasn't. This gives a pleasant degree of control to the TDA programmer, since he can choose to truncate wherever he wants, or choose to go with either the first, second or whichever local maximum he sees fit, and still obtain a *meaningful* value of clusters. This was not the case for *gap*.
Finally, the modified gap curve has a quantitative comparison characteristic between *meaningful* values of k. As we can see, in Example 4.19 the difference between $Gap_{mod}(4)$ and $Gap_{mod}(8)$ is not as large as in Example 4.20, where $k = 4$ is much clearly preferable to $k = 8$.

We can wrap up this discussion by further validating our method with a couple of differently structured examples:

Example 4.21. We generated a similar set of data except that the 4 large clusters are now separated by 20, and the 8 smaller clusters remain separated by 5.

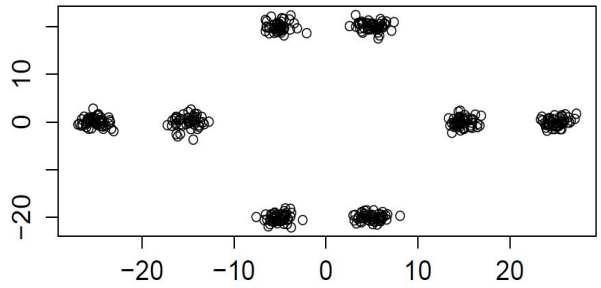

Figure 44:

The modified gap analysis for this data set is shown in Figure 45

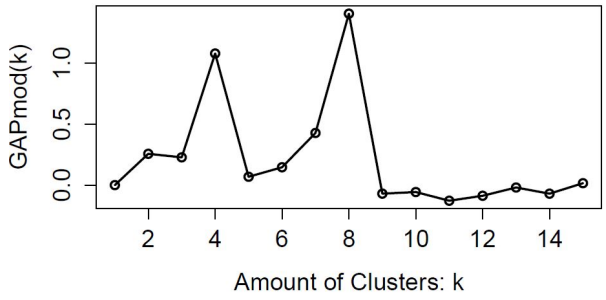

Figure 45:

This example proves a valuable point: Our modified method is not biased towards the smaller *meaningful* k, since this time it chooses $k = 8$.

Example 4.22. Let's provide further validation of the non-biased nature of our modified method with an example a bit more extreme. Now, we generate a data set with the larger clusters separated only by 15.

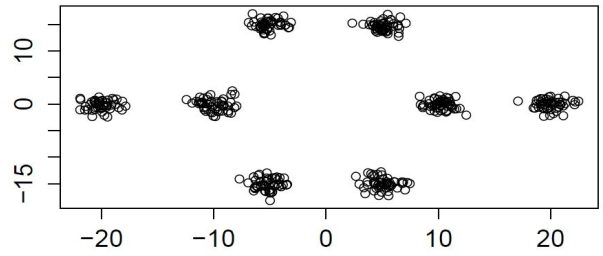

Figure 46:

Figure 47 shows the modified gap curve.

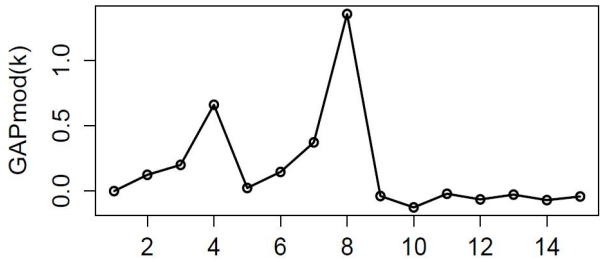

Figure 47:

In this example, the domination of the choice $k = 8$ over $k = 4$ is more pronounced, as would be expected.

Finally, our method can also be tested against the sensitivity of outliers we demonstrated in Gap analysis.
Figure 48 shows the modified gap curve for the data set from Example 4.15:

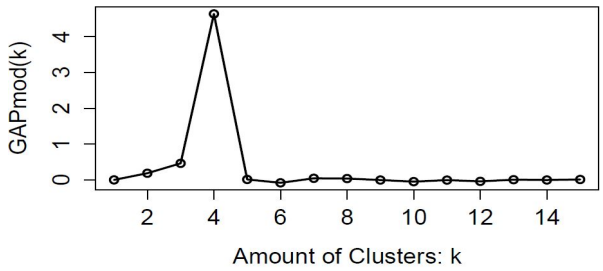

Figure 48:

Again, modified gap chooses $k = 4$ wisely, giving $k = 8$ little importance.

Remark 4.23. Both expressions, 4.16 and 4.17, were consulted in all examples, and always gave the same result.
We've said several times that a key characteristic that allows the heuristic reasoning behind the modified gap method to work in practice is the approximate linearity of $log(W(K))$ in small intervals of k.
Notice that for data sets with n observations, $W(k)$ can be calculated for $k \leq n-1$. In our toy examples of this section, where $n = 400$, *small* meant that we only considered $k \in \{1, 2, ..., 15\}$.
Had we considered $k \in \{1, 2, ..., 399\}$, the supposed linearity of the *log* curves is very far-fetched. However, this has nothing to do with the *proportionally small* size of 15 compared to 399.
If a 16 point data set is generated, and the $log(W(k))$ curve studied for $k \in \{1, 2, ..., 15\}$, it is still acceptably linear. This validation is important since in a TDA Pipeline investigation, the algorithm is constantly faced with deciding the number of clusters in data sets of *small sizes*, so our method must be competent in these scenarios, which it effectively is.

5 Application to Football Data

5.1 The Football Data

The data base on which we applied the presented techniques was OPTA's database for the 2011-12 season of the English Premier League. Each row of the data matrix consists of in-game statistics for a single player during a single game. A full list of the in-game statistics available in the database can be found in the appendix.

Now, lets recall that our objective was to test the capacity of the TDA technique applied to two football-related questions; one related to individual player's playing style and position, and another related to team performance, formation and team style.
Let's take a look on how we readied the data for each problem:

INDIVIDUAL PLAYER PROBLEM:

First of all, we summed the statistics for each player throughout all the matches he played, since considering statistics for a single game is heavily susceptible to game specific situations. This gave us a matrix where each row represents a player as a vector, and each column represents the sum of a single feature throughout the whole season, i.e. total goals in the season, passes, interceptions, etc. Each player comes labeled as either a goalkeeper, defender, midfielder or striker.
Also, goalkeepers were excluded from the data matrix since their in-game statistics are virtually incomparable to those of the outfield players.
Another group of players excluded were those who didn't add 90 minutes of time played

throughout the season, since their feature vector could very likely represent an outlier due to a *sampling bias*.

Once this *player selection* was done (leaving us with 450 players), we then put the data through a process of *feature selection*.
Features were either selected or discarded on two criteria:

1. There are some linear dependencies within the features. For example, the feature *total shots* can be seen as the sum of the features *total shots from inside the area, total shots from outside the area* and *total headed shots*. Many other linear relationships like this one exist.
   In these cases, either the big feature was kept and it's components discarded, or the components kept and the big feature discarded, depending on whether having the feature disaggregated was deemed relevant or not.

2. Other "*uninteresting*" features were omitted, like *successful throw-ins* or *shots cleared off line.*

Consult the appendix for a detailed account of this process of feature selection.
Additionally, it makes sense to weight a player's statistics by minutes played, since after all we want to recognize position and playing styles which shouldn't be affected by this feature.
Finally, we normalize each feature to avoid having our metric dominated by features with large values, like successful passes or touches, in detriment of other small value-features like goals or assists.
Figure 49 shows the resulting data plotted onto it's first two principal components, and constitutes the representation of the data with which we worked.

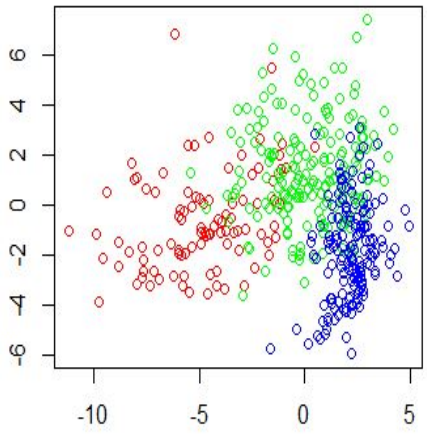

Figure 49: *Strikers are plotted in red, Midfielders in green and Defenders in blue.*

TEAM PERFORMANCE PROBLEM:

For each match of the season we have 199 in-game statistics from every player who participated in the game, ranging from 11 to 14 (including substitutes) for each of the two teams.
This left us with the possibility of viewing each match as two distinct points (one for each team), in $\mathbb{R}^{2786}$. To significantly reduce dimensionality, we performed two feature selection processes:

1. Just as with the individual player data base, we sought to eliminate linear dependencies within the data, as well as uninteresting variables.

2. Once these features were eliminated, we undertook a second stage of feature selection, in which each feature fell under one of two categories.
   The first category consists of features whose distinction between different positions is irrelevant. For example, set piece goals, corners taken or shots from outside the box are features that may characterize a team's performance, regardless of which specific player carried them out. On the other hand, features like crosses completed, key passes or interceptions characterize a team performance depending on which players (defenders, midfielders or strikers) are performing them. These form the second category.
   Once a classification into these two categories is performed on the remaining features, we sum over all players for the features in the first category, and sum over all defenders, midfielders and strikers for features in the second category.
   That is to say, our final vector for each performance has a feature indicating *set piece goals* for the team as a whole, and 3 different features for *key passes*; the *key passes* made by defenders, midfielders and strikers.

Once again, refer to the appendix to see the details of this feature selection.
As expected, we also normalize each feature column in the data. Figure 50 plots the result onto its first two principal components.

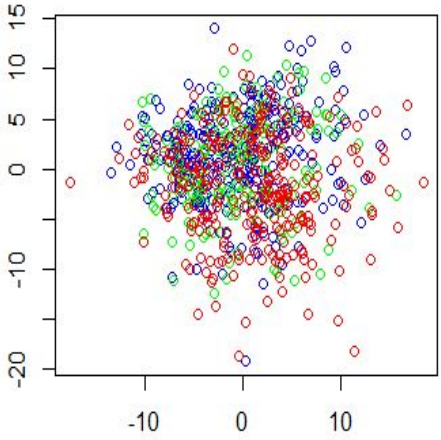

Figure 50: *Defeats are plotted in blue, draws are plotted in green and victories in red.*

The result is much more chaotic than in the individual players problem, but not to worry, rest assured all will be sorted out.

5.2 Learned Metrics

Lets move on to the metric learning algorithms we used to learn a truly meaningful metric inherent to the problem at hand.
Recall that in order to learn a metric through the Large Margin Nearest Neighbor algorithm, we need a vector of labels that will determine the target neighbors.
Lets take a look at the process for each scenario:

INDIVIDUAL PLAYER PROBLEM:

We learned a metric for each of the following labels:

1. **Position:** The labels corresponding to defender, midfielder and forward.
2. **Quadrant:** Each team was assigned a number according to their final position in the league table. Teams finishing in positions 1 through 4 receive number 1, teams finishing from 5 to 8 receive number 2, and so on. At the end, each player is assigned the number assigned to the team he plays for, and these are the labels used. In the end, we expect players from top teams to be near each other, as well as players from bottom teams.
   In our implementation, only 8 target neighbors are chosen, so we expect the target neighbors of strikers in top teams to be strikers in top teams, as in midfielders or defenders.

3. **Position+Quadrant:** A small modification from the *quadrant* labels, except that we now explicitly tell the algorithm to pair by quadrant AND position.

Figure 51 shows the result:

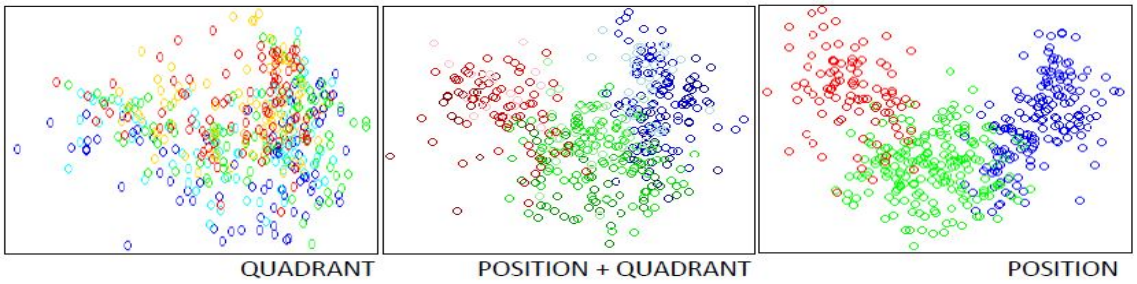

Figure 51: *For quadrants, blue represents the highest placed teams, red the lowest and the intermediate colors represent sequentially the increase. The positions are represented as in Figure 49, and the position+quadrant case the darker the color (either blue, red or green) the higher the quadrant*

TEAM PERFORMANCE PROBLEM:

The following labels were utilized to learn metrics for the problem:

1. **Result:** Matches were labeled by defeat, draw and victory. Recall that each match has two distinct vectors, one for each of the contesting teams.

2. **Goal Difference:** Matches were labeled by the scoreline, where a "2" means "won by 2 goals", and "-3" means "lost by three goals".

3. **Team:** Matches were labeled by an ID representing their teams (numbers from 1 to 20), so that matches from the same team are targeted to be together. In our implementation 10 target neighbors were chosen, so that defeats, draws and victories from the same team should be target neighbors for each other.

4. **Quadrant:** A slightly less stringent label was used, similarly labeling matches from teams from the same *quadrant*.

Remark 5.1. Since a match observation has features which indicate both the number of goals scored and the numbers of goals received, learning a metric by result could be quite artificial.
To the point, Figure 52 shows the plot onto the first two principal components of the projected data set once the L had been learned.

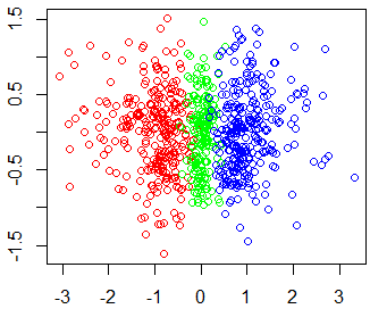

Figure 52: *Defeats are plotted in blue, draws are plotted in green and victories in red.*

Comparing the result with what we saw in Figure 50, it seems as though the distinction is now perfect.
However, to avoid this artificiality, when learning a metric by *Result* labels, we omitted any feature which reveals the number of goals scored, including assists.
More details can be found in the appendix.

Figure 53 shows the results:

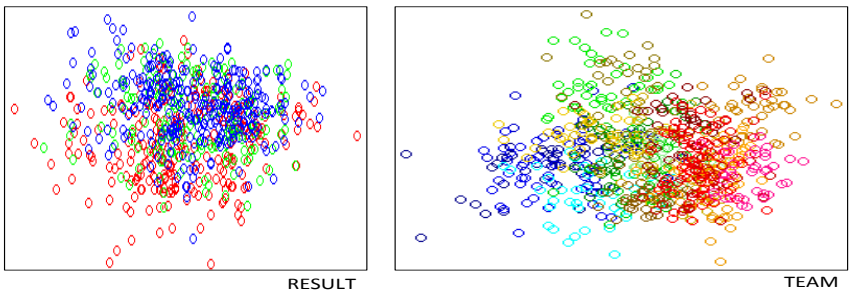

Figure 53: *Teams are colored in the color range from blue to red, with blue teams being those at the top of the table. Results are colored as before.*

Even though an improvement is clearly not distinguishable to the PCA eyes, the algorithm

reduced the number of active impostors from 168,411 to 5,159 when using the Team labels for 10 target neighbors, from 9,697 to 5,491 when using the Quadrant labels for 15 target neighbors; and when the goals-scored related features are omitted, from 85,099 to 8,472 when using the Result labels for 15 target neighbors and from 206,387 to 14,006 when using the Goal Difference labels.

Figure 54 shows the plot of the eigenvalues for two examples of learned metrics we have used:

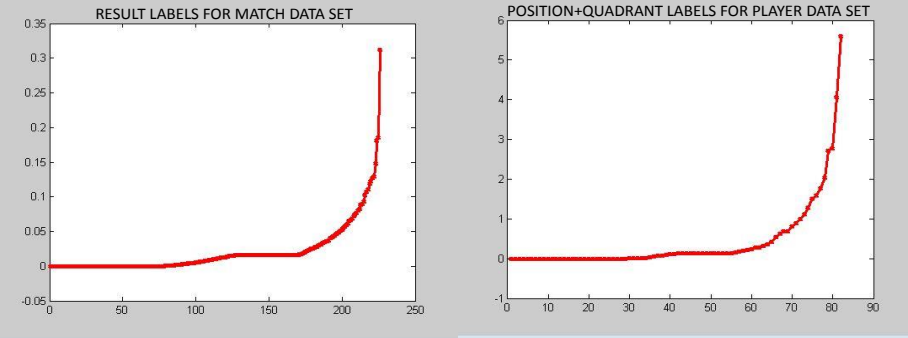

Figure 54:

We can appreciate that in both cases we have eigenvalues near 0, and a small percentage which are considerably larger than the rest. This establishes that the learned metrics are mainly relying on the projection onto a small amount of linear combinations of certain feature vectors to determine the new metric, which is reasonable. The amount of clearances and goals should clearly differentiate between strikers and defenders, and a projection onto a linear combination of these two features should carry plenty of weight in the new metric.

5.3 TDA Graphs and Interpretations

We are finally in a position to show the resulting **TDA Graphs**.

To recap, each graph will be determined by 4 aspects:

1. What data set, *Individual Players* or *Team Performance*?
2. What labels were used in the metric learning?

3. What filter function was used: *PCA, Data Depth, L-Centrality* or *Data Density*?

4. How was the *target cover* determined: by length or percentage of the width and epsilon parameters? What were these values?

In this section, we will show some selected examples of **TDA Graphs** constructed from our football data, and attempt to draw relevant conclusions as was done in Examples 13 and 14.

Example 5.2. We begin with a result for the **Individual Player** data set.

- Position+Quadrant Labels
- Principal Component Analysis with $k = 2$
- Length Parameters

Figure 55 shows the resulting graph for $width = 0.5$ and $overlap = 0.2$.

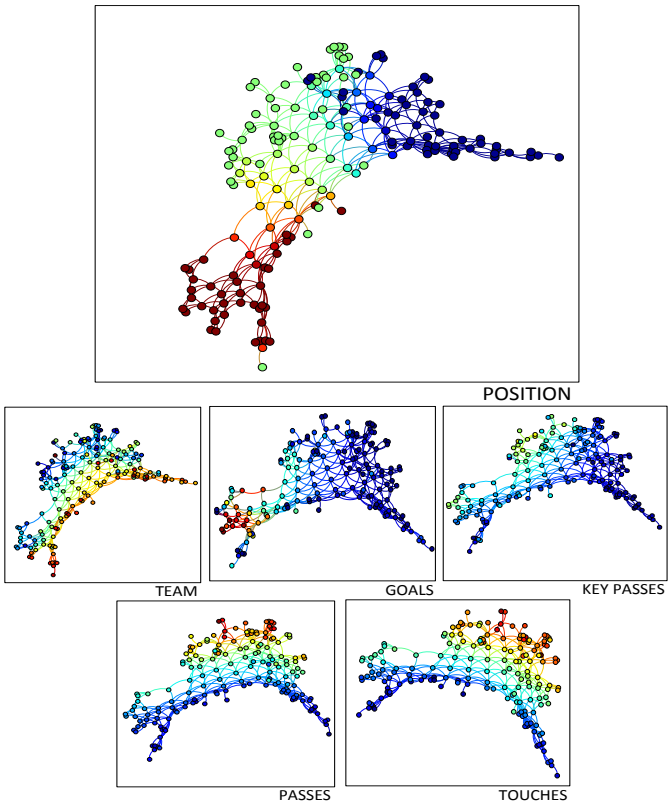

Figure 55: *In general, red represents high values in the coloring scale. In team, for example, the dark red nodes are composed of players from teams with a high (as in 19 or 20) finishing position. This is general for all the graphs from now on.*

Simple observation of the results answers some initial questions. For example, we have strong evidence that the available *in-game statistics* vector of players can distinguish (in a topological layout sense) the range of playing positions from defender to striker.
Also, it can seemingly distinguish players playing for differently positioned teams; and even topologically distinguishes some classical categories in football folklore like Key Passers of the Ball (the "number 10" players).

Now, how can we further stretch our method and take advantage of all its possibilities?

Examples 13 and 14 showed applications from the literature on the subject, and their basic premise was the following: By decreasing the *width* and *overlap* parameters, the graph will gain a less uniform structure and qualitative features will appear, like flares or loops, which should represent a significant grouping inside the data
Eventually, the graph will become partitioned, and the small broken up pieces should also represent a classification in the data.

The situation is not as simple.
It is true that in general the graph becomes increasingly structured and partitioned as the parameters decrease, but there is not a strict ordering in this process.
What we mean by this is that just because a piece of the graph brakes off for one set of parameters it must remain broken off for all smaller parameter values. Similarly, a loop can easily brake into two flares for one set of parameters and then for smaller values merge together again.
The fact that this happens is not unnatural. If we reflect on how the grids in the *target space* look for different sets of parameters, these in general have nothing to do with each other. Even if we partition an existing grid in a careful design, for example taking each square of the grid and partitioning it into 4 smaller squares, the inclusion or exclusion of a single point in a bin can completely alter the outcome of the clustering dendogram and the *Modified Gap* analysis; so not only are the edges of the graph altered, but the sets of nodes are essentially incomparable.

More on this issue will be discussed in Section 6, but it is clear that studying the resulting graph for a single set of parameters is not necessary insightful into the general structure of the data.

This discussion can be highlighted in the example at hand:
Figure 56 shows the graph for $width = 0.225$ and $overlap = 0.06$:

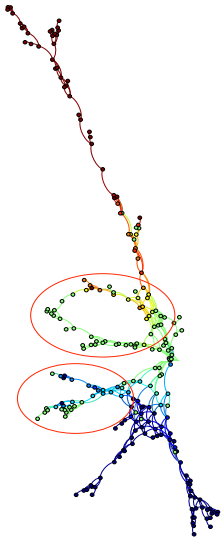

Figure 56: *Colored by position*

The structural elements of note are the circled flare and loop of midfielders.
Reviewing what players are present in these structures, we obtained the following summary:

1. **Flare:** Some examples of the players present are Nigel de Jong, Michael Carrick, John Obi Mikel, Michael Essien, Cheik Tiote, Gareth Barry, Paul Scholes, etc. A full list of the players can be found in the appendix. However, out *football savvy* readers might have noticed that these players are usually empirically classified as *holding* or *defensive midfielders*.

2. **Loop:** Two distinct empirical categories are present.
   Going up the side of the loop closer to the flare there are players such as Yaya Toure, Alexandre Song, Jordan Henderson, Aaron Ramsey, Steven Gerrard, Tom Cleverley, Ross Barkley, Frank Lampard, Ryan Giggs, Samir Nasri, Tomas Rosicky, etc. Again, for a full list see the appendix. These players are usually referred to as *box to box midfielders*.
   Once we reach the summit of the loop and start descending back down to the body of the graph, a different type of player begins to appear. Some examples are: Antonio Valencia, Gylfi Sigurdsson, Adam Johnson, Gareth Bale, Dirk Kuyt, Alex Oxlade-Chamberlain, Andrey Arshavin, Ashley Young, Nani, etc. These players are usually classified as *wide midfielders.*

Now, Figure 57 shows the graph for the slightly different set of parameters $width = 0.2225$ and $overlap = 0.06$:

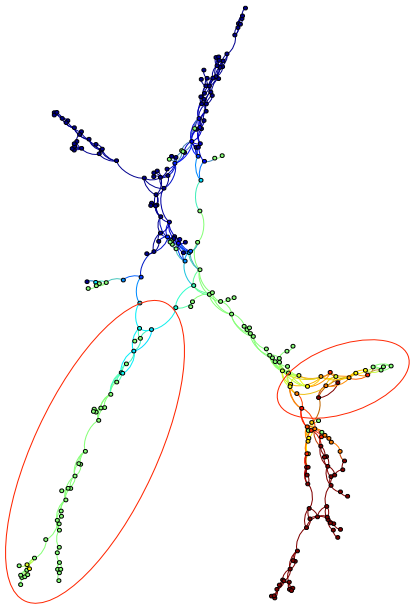

Figure 57: *Colored by position*

This time, two distinct flares occur. This is a summary of their composition:

1. **Flare 1:** The previously mentioned group of *holding midfielders* constitutes the first half of the flare, and attached at the tip (the player who makes the attachment is interestingly Yaya Toure) prolonging the flare is the *box to box* group.

2. **Flare 2:** The *wide midfielders* group.

From these two examples we can infer that all these midfielders (holding, box-to-box and wide) are layed out in a structure which should resemble the number 8, but the sensitivity of plotting the graph for a single set of parameters has so far failed to recognize the whole structure without inadvertently breaking off or damaging part of it.

After an excruciatingly long trial and error search, the following graph was found for $width = 0.2525$ and $overlap = 0.6$:

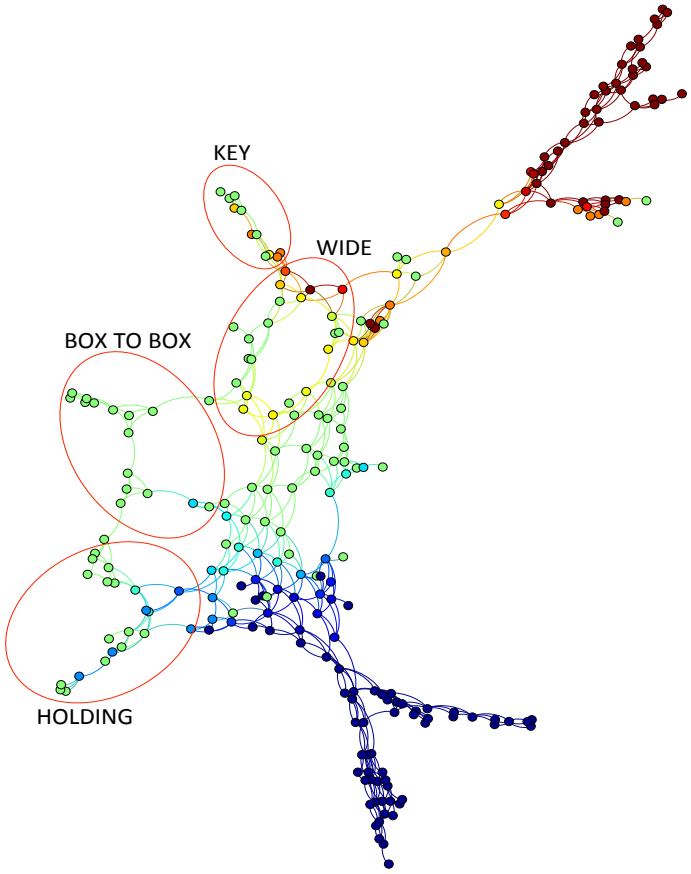

Figure 58: *Colored by position*

An additional subgroup which had remained broken off for other parameters has emerged and joined the graph at the end of the *wide midfielders* loop, which we have empirically labeled as the *key* or *number 10 midfielders*, consisting of Adel Taarabt, Florent Malouda, Rafael van der Vaart, Juan Mata, Steven Pienaar and David Silva.

Remember that these groups were identified empirically using a priori football knowledge, but the only labels available in the data set are defender, midfielder and striker. This means that through TDA we have *validated* statistically the existence of empirically determined groups in football folklore.

Figure 59 shows the same graph colored by different criteria:

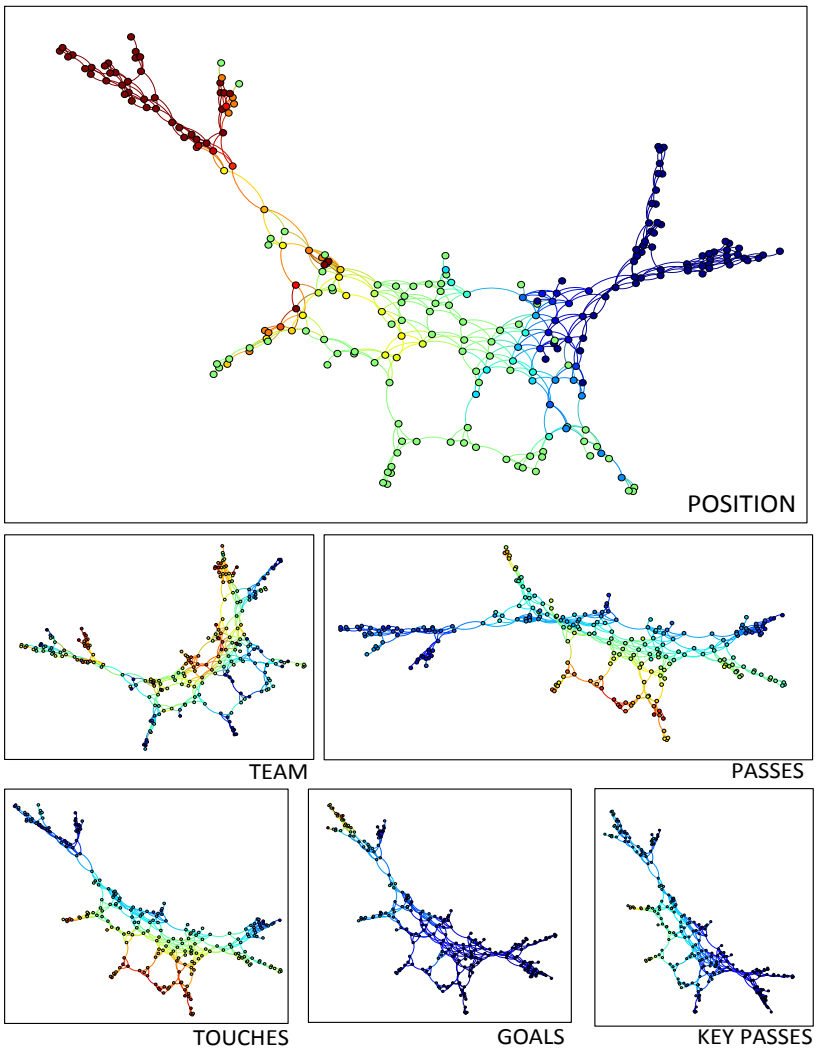

Figure 59:

Remark 5.3. Notice that the *number 8* figure we are commenting on is predominantly blue in the graph colored by team position, meaning that these midfielder structures we have identified are specifically for midfielders in top teams (as the reader might have noticed from the specific names).

When tracking these four groups of midfielders for smaller parameter values we confirm that they are significant groupings of the data. Each group eventually separates from the body of the graph and from each other, becoming an isolated flare.
However, had we focused only on breaking up the graph into small parts rather than tracking how these parts are interrelated between themselves, valuable information would have gone unnoticed. Its *qualitatively* different for these groups to appear in 4 distinct flares than in a *number 8* shape, and the capacity to retrieve this *knowledge* is precisely one of the strengths

of TDA.

This is not to be interpreted as saying that further groupings can't be found by using sequentially smaller parameters, but rather to highlight the importance of keeping track of the results for several parameters.
By decreasing the set of parameters we can recognize some other persistent groupings. Figure 60 shows the graph for $width = 0.2$ and $length = 0.06$:

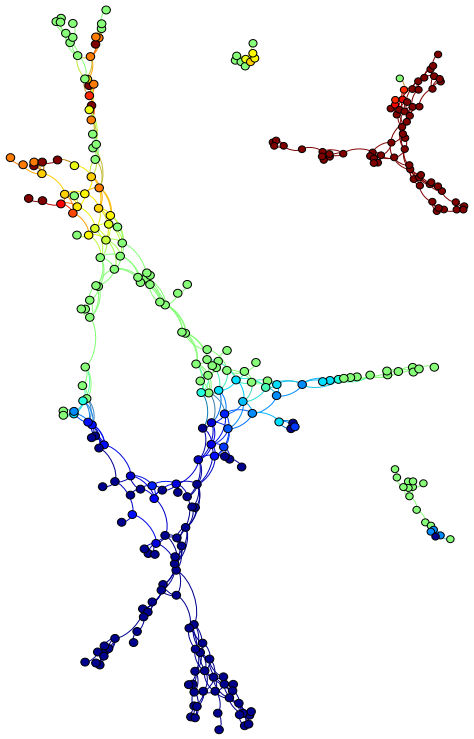

Figure 60: *The red piece broken off is a structuring of strikers. There are also some small parts broken off from the number 8 midfielder structure like the key passers and some of the box-to-box players.*

The striker structure has broken off, and consistently begins to show a structure of 3 flares joined at the base. The player who has the distinction of making the connection is Dimitar Berbatov, a widely recognized all-round player.
The major strikers from top teams in the Premier League like Robin van Persie, Emmanuel Adebayor, Sergio Agüero, Mario Balotelli, Didier Drogba and Edin Dzeko appear in one of these flares (Luis Suarez and Wayne Rooney are isolated nodes).
Another flare has renowned *wingers* like Gervinho, Theo Walcott and Danny Welbeck.
An empirical classification of the flares would be: *small teams' central strikers, top teams' central strikers,* and *wingers.*

For a full list of the players by flare, refer to the appendix.

A further recognizable grouping is the division of central defenders into those from top teams and those from small teams (this is evident in the graphs colored by team position).
Surprisingly, full-backs have no structure that clearly set them apart, and are rather spread out in the limbo between central defenders and midfielders.

Another interesting possibility of our approach is the identification of potential. As we have shown, the method competently distinguishes the players from teams on the upper end on the table versus the rest, and the layout of the data respects some aspects of *footballing value* of players, like Key Passes.
In this spirit, we can assume that younger unknown players which appear *topologically close* (by this we mean either strictly metrically or also structurally, like belonging to the same loop or flare) to widely recognized important players should potentially be rising stars.
Since the data frame worked on is from the 2011-2012 season, and this investigation is being carried out in 2015, some of these cases can actually be verified by the player's current status. For example, Luka Modric, today a fundamental piece in Real Madrid's team, appears in the same *box-to-box* structure as then seasoned and acclaimed midfielders such as Frank Lampard, Steven Gerrard and Yaya Toure. Even younger players like Aaron Ramsey and Ross Barkley, who only had their breakout season last year, appear in this same group. Another example is Alex Oxlade-Chamberlain, who was 18 years old in 2011-2012 and is today a world-class winger for Arsenal, and in our implementation appears close to Gareth Bale.

Example 5.4. Lets turn our attention to the **Team Performance** data set.

- Team Labels
- Principal Component Analysis for $k = 2$
- Length Parameters

Figure 61 shows the results for $width = 0.15$ and $overlap = 0.05$:

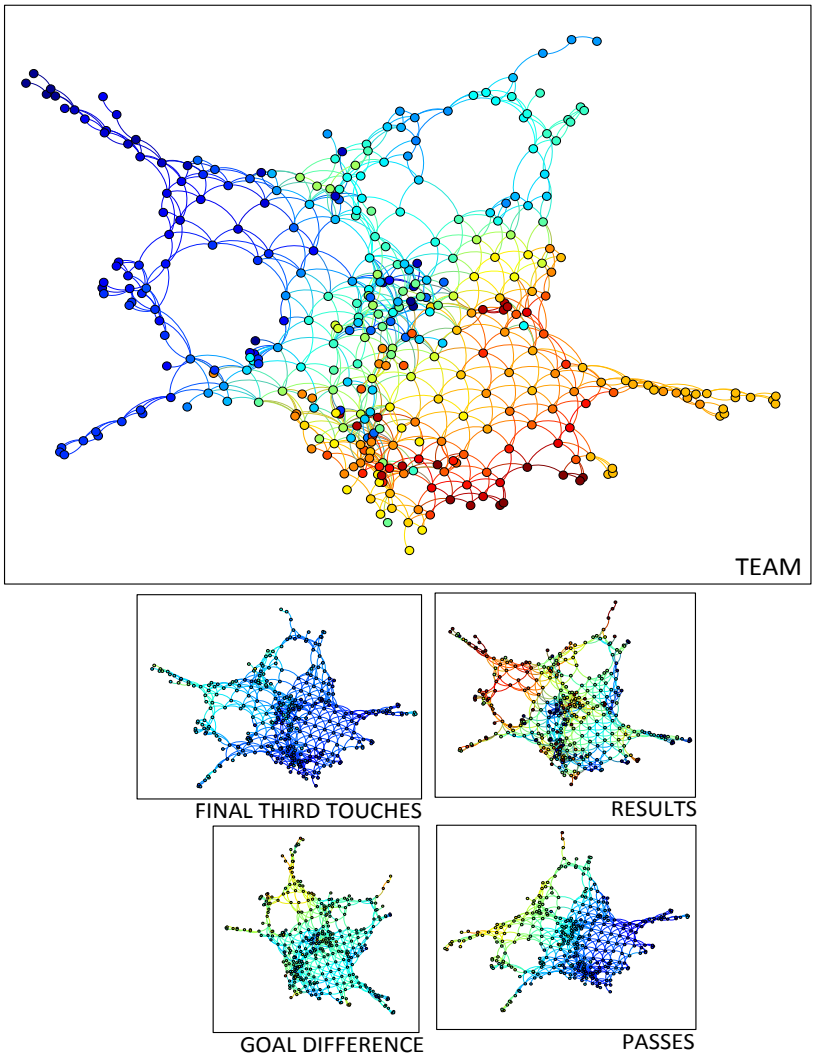

Figure 61:

Again, the results leave us with satisfactory conclusions. It is clear that we can topologically distinguish the performances of teams according to their final position in the league, and do so progressively; i.e. the teams in the lower positions are closer to the mid-table teams than to the top teams.
Also, individual teams' playing style seem to be distinguishable, since the matches of a single team appears predominantly in one area of the graph.

As an important commentary, even though the coloring by result doesn't seem too bad for this large set of parameters, as we decrease the values of the parameters this coloring gets consistently worse. Distinguishing by victory, draw or defeat seems to be the main loser in "topological distinguishability".

Lets now look for further categorization of our data. As we did previously, lets sequentially decrease the parameter set and observe the outcomes.
Figure 62 shows the result for $width = 0.085$ and $overlap = 0.02$ colored by team position:

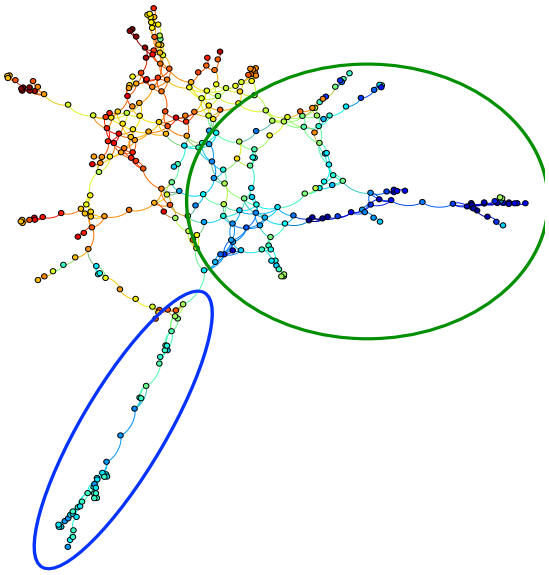

Figure 62: *Colored by team*

The flare circled in blue, with the exception of a single Manchester United game, consists entirely of games played by Chelsea and Fulham, which leads us to conclude that they have a similar playing style, distinct from the rest of the teams in the division.

This distinct structure persists for several parameters until it eventually breaks off from the rest of the graph. Lets decrease the parameters further and look for further structuring:
Figure 63 shows the result for $width = 0.075$ and $overlap = 0.02$:

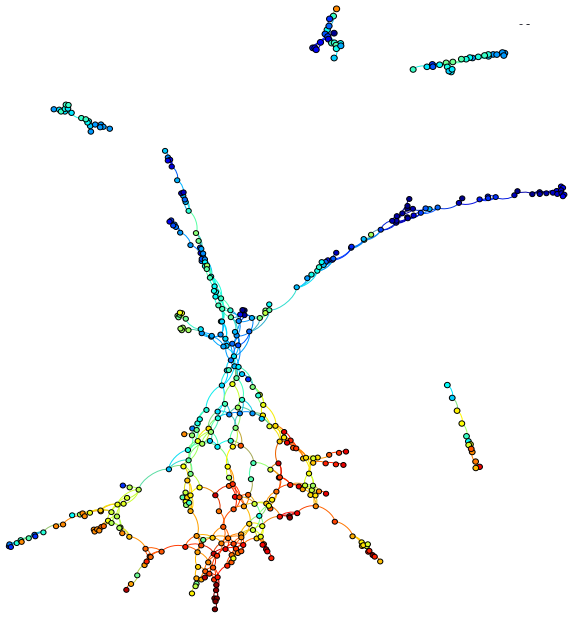

Figure 63: *Colored by team*

The nodes consisting of Chelsea and Fulham nodes have now broken off from the large portion of the graph, and two new flares of matches from blue (i.e. top of the table) teams have appeared.

These two flares are somewhat less directly identified as containing specific teams. Manchester City and Arsenal are dominant in one of the flares, while Swansea is dominant in the other; but Manchester United and Liverpool are somewhat equally divided between the two. The nodes near the base of the flares that connect them to the body of the graph consist predominantly of matches from Newcastle and Everton, ranked 5th and 7th respectively in that season's league table.

There is a final important lesson to be learned from this example: in Figure 63, the Chelsea and Fulham flare is no longer even a disconnected piece, and has actually been broken up into oblivion, now being broken up into several small pieces and isolated nodes.

We're sure the reader agrees that this doesn't invalidate these matches claim to be recognized as an important grouping inside the data; and yet, the parameter needed to separate the two flares from Figure 63 has discarded the existence of the Chelsea-Fulham grouping. This reflection further ratifies our insistence that a single set of parameters is insufficient to provide the whole wealth of information available; but it also introduces an improvement

idea:
The parameter that correctly recognizes the flares from Figure 63 is too small locally for the Chelsea-Fulham flare, so perhaps the problem is that there is no reason for uniformly determining the grid for the target cover. A better target cover could be obtained by using a grid that is small around the images of the matches in the flares from Figure 63, and larger around the images of Chelsea and Fulham matches.
More on this reflection will be explored in Section 6.

Example 5.5. Finally, lets take a look at another **Team Performance** example, this time the cropped data set without any *goals scored* features.

- Result Labels
- Principal Components with $k = 2$
- Length Parameters

Figure 64 shows the result for $width = 0.15$ and $overlap = 0.05$.

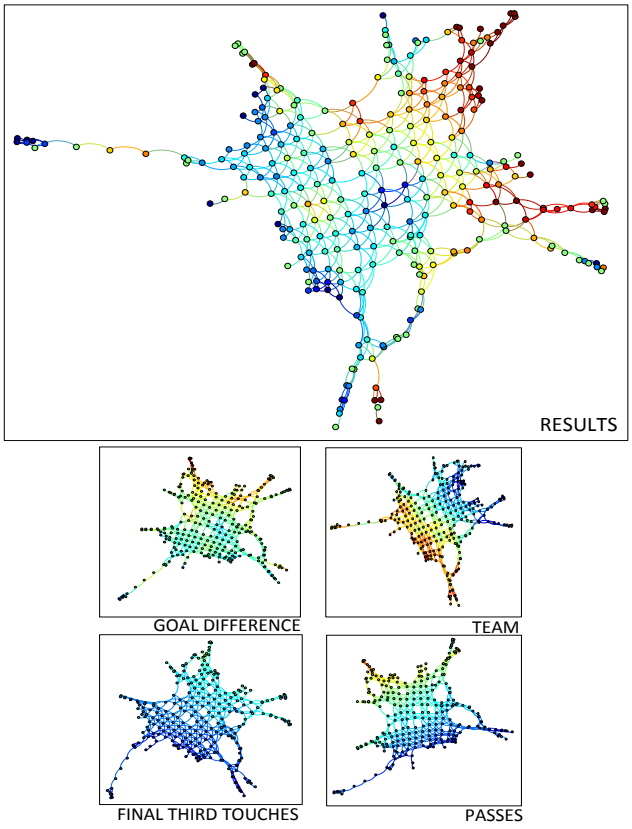

Figure 64:

This result shows that viewed with this large lens, victories are in general distinguishable from defeats in a nice progressive manner, albeit of course not perfect.

By reducing the parameter values we can get a closer look at what the data has to tells us:
Figure 65 shows the results for $width = 0.1$ and $overlap = 0.03$:

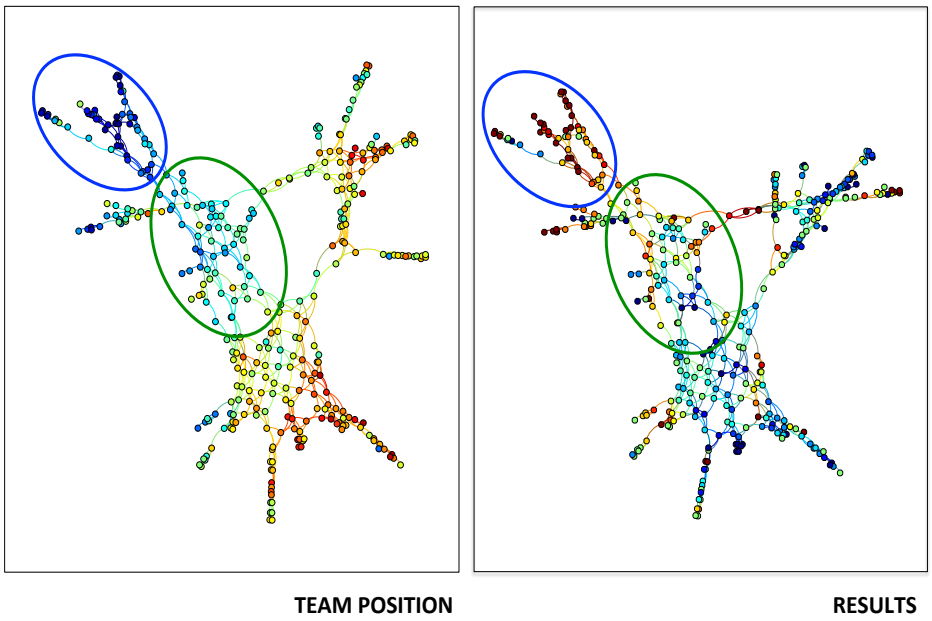

Figure 65:

The coloring by results is no longer as clear as for the previous set of parameters, leading us to believe that the outcomes of the matches are not really being distinguished with a great deal of success.
However, by simultaneously observing the coloring by *team position*, an idea of what may be going on becomes clear.
The graph seems to distinguish in a broader sense by team position, but does it then distinguish the outcomes of matches inside zones determined by team position?

The answer seems to be yes.

Out of 59 matches played by teams finishing in the top 8 in the league inside the blue circle, 41 were victories, 12 were draws and only 6 area defeats.
This means that 135 out of 177 possible points were taken, i.e. 76%.

On the other hand, out of 80 matches played by the top 8 teams in the green circle, 26 were victories, 26 were draws and 28 were defeats.
This means that 104 out of 240 possible points were obtained, a meager 43%.

The difference in the success percentage suggests that the performances are in fact distinguishable, since the unpredictability of football matches' outcome cannot be accountable for such a significant difference.

Figure 66 shows the graph for $width = 0.095$ and $overlap = 0.025$ colored by Team Position, Results and Passes Completed, which further justifies our claim and shows the key role that pass completion plays in winning a game.

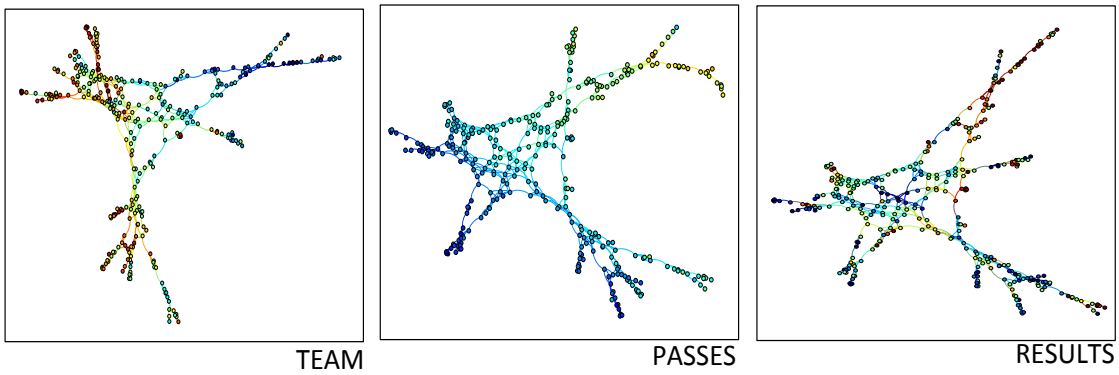

Figure 66:

There is an interesting result when the parameters are further decreased, which we can visualize for $width = 0.08$ and $overlap = 0.02$:

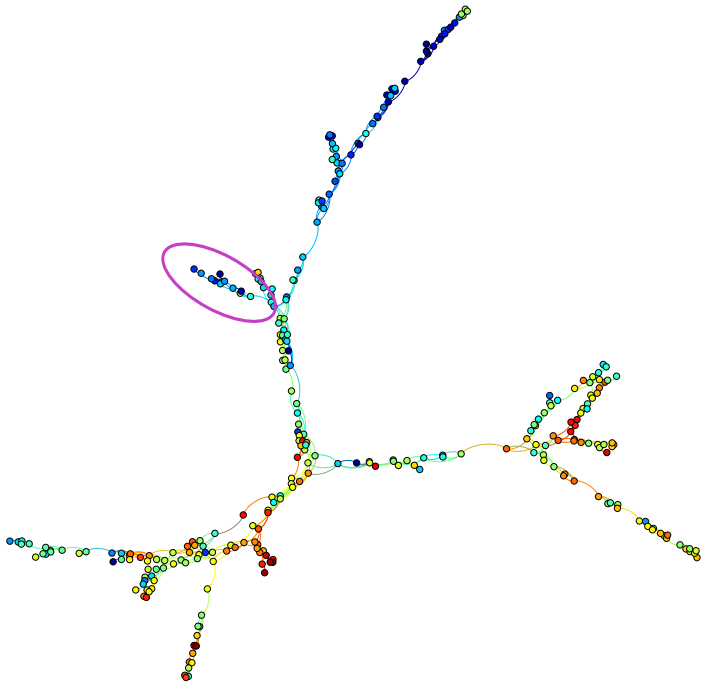

Figure 67: *Colored by team*

The long flare is the equivalent to the zone circled in blue in Figure 65, and as we gown down towards its base and the body of the graph we find the nodes of the zone circled in green.
In this graph, the small circled flare in an interesting grouping: out of 17 matches in present in the nodes of the flare, 12 are between teams finishing in the top 8. Some examples are Arsenal 0-0 Chelsea, Manchester City 1-0 Arsenal, Manchester United 3-3 Chelsea, etc. For a full list see the appendix.
This result means that the feature vector for a team in the top eight's performance is topologically distinguishable depending on the "quality" of the opponent, a conclusion that makes perfect footballing sense.

To showcase the merit of using the TDA Pipeline, Figure 68 shows how this grouping whose significance is empirically validated is not picked up directly by projecting into the first two principal components:

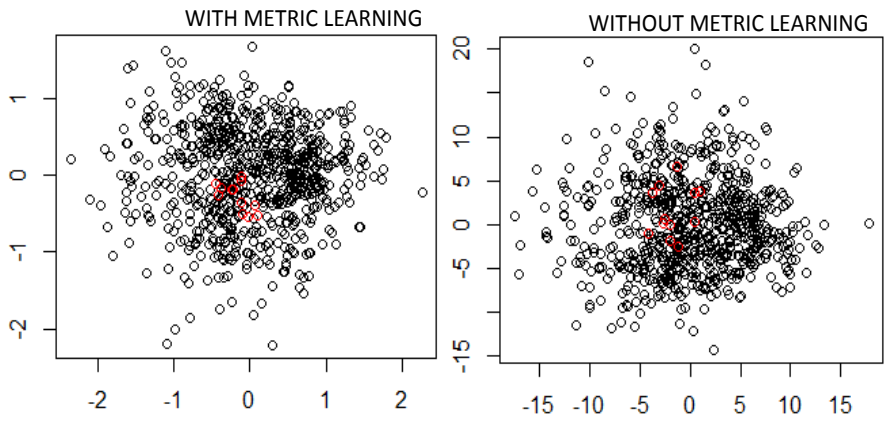

Figure 68: *The red points correspond to the grouping we found in Figure 67, either having learned the metric by Reult labels or not. Evidently after having learned the metric the points are closer, but even then they are in the midst of many other points and aren't recognized as a grouping through PCA analysis.*

Lets close out the Results Section by remarking on a couple of aspects that may have generated some questions.

Remark 5.6. The reader might ask himself whether the metric learning algorithms is heavily affecting the results in an artificial way.
This is certainly a concern. Figure 69 shows the TDA Graph of the Team Performance data set with a metric learned through match results **without** removing the features which explicitly have goals scored and goals received information.

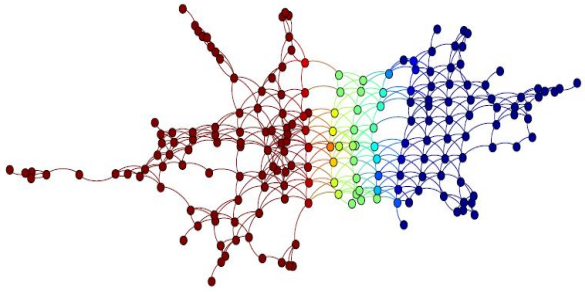

Figure 69: *Colored by result*

The almost perfect result clearly conflicts with our previous claim that specific results weren't topologically identified as easily.
However, we have also studied the graphs created without any learned metric, and most of our broad hypothesis like the fact that playing positions are distinguishable or the fact that distinguishing results in the team data sets is secondary to distinguishing specific teams' matches. Even the fact that Chelsea and Fulham's style of play clearly differs to that of other teams is also observable without metric learning.

Of course some specific groupings at a smaller scale are affected by our choice of using position, team, result, quadrant or indeed no labels; but deciding whether this is a positive change is as subjective as the labeling itself.
The important matter is that unless a particularly poor choice of labels is made (like "result labels" in the team performance problem where a match's result is explicitly determined by 2 features in the feature vector), the metric learning algorithm seems to respect the overall essential information of the data set.

Remark 5.7. The reader might have noticed that we only showed results for TDA Graphs using principal components as their filter function.
The reason for this is quite simply that the other filter functions didn't work as well.
Figure 70 shows some results using other filter functions:

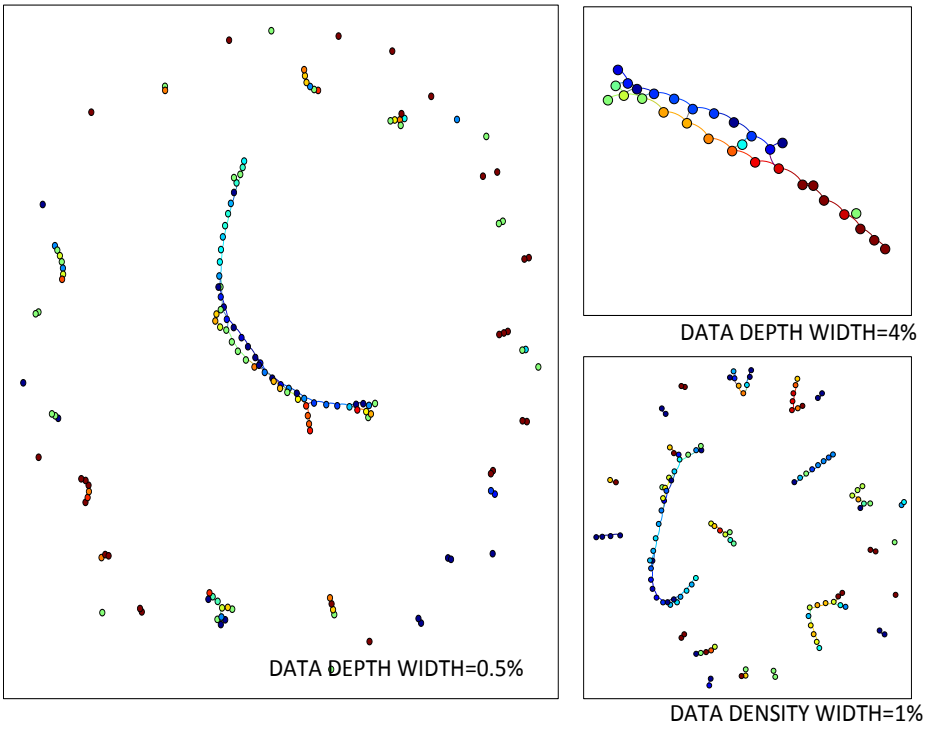

Figure 70:

Nevertheless, the reason why they didn't work well can be informative.

For example, as we mentioned towards the end of Example 5.4, the uniformity of the target grid is unjustifiable. Locally, there are always grid parameters which pick up the pertinent structure successfully; but these parameter values that work well at some parts of the target space may be too small or too large in others.
Figure 71 shows the values of the Data Depth function applied to the Individual Player data set (with LC-Centrality working somewhat similarly):

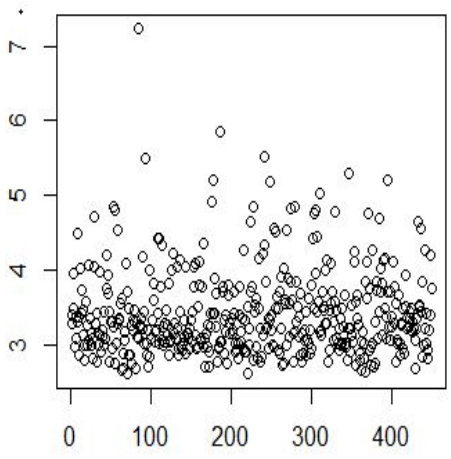

Figure 71: *On the x-axis are the 450 points of the data set, and on the y-axis their data depth is plotted.*

In effect, the parameter values that are large enough to avoid isolating the nodes with higher data depth values are too large to pick out any meaningful structure in the graph for the lower data depth nodes, while the smaller parameters who do justice for the smaller data depth nodes isolate the nodes with higher values.

Naturally, we would expect better performance by having a target grid in $\mathbb{R}$ that is small near to 0, and increasingly larger for higher values.
We will return to this topic in Section 6.

On the other hand, the failure of the Data Density filter can't be attributed to this scale problem since its values are acceptably uniform across its range.
Its performance has been determined rather by the problem we commented on in Remark 4.11 the *shape* of its bins.

Of course we have no way to get a visual grasp on the problematic shapes in dimensions greater than 4, but we can trace the problematic performance in a way to sustain our claim: When studying the bins that were broken into several clusters in a Pipeline using a Density filter and recording the comparison of distances between points of different clusters and points of the same cluster, we discovered that there was in many cases not a clear difference. In fact, we found plenty of points that were closer to points in other clusters than to points in their own cluster.
In contrast, when this same meticulous analysis was done for a Pipeline using a PCA filter, bins that were broken into several clusters exhibited well separated clusters, where the distances of a point to points in his same cluster was clearly inferior to the distances to points in different clusters.

This is reminiscent of the problems of Example 4.10 when we asked a clustering algorithm

to perform the tough task of clustering a circular point cloud, so it is sensible to assume that in our examples the bins generated by a Density filter are handing over tough shapes for the clustering algorithm, and the wrong choices are being made.

Remark 5.8. Finally, it is important to mention that the method we mentioned in Section 2.2 to verify the acyclicity of the discrete cover through Persistent Homology techniques was attempted but produced no results of note.
This is unsurprising since our resulting "open sets" consist of no more than 20 points for the parameters of interest.
20 points in such high dimension as ours ($\mathbb{R}^{80}$ and higher) are laughable.
Even when introducing our whole data set into the Persistent Homology algorithm the resulting barcode diagram shows no interpretable behavior.

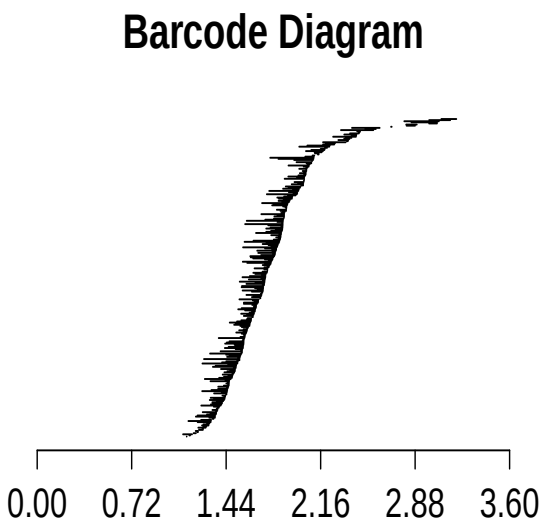

Figure 72:

We still have faith however that this tool can be valuable in data sets that are more numerous relative to their dimension.

6 Conclusions: Where to Go From Here

We have shown, both theoretically and in practice, how to build a point cloud Čech complex by clusters for a data set. As we stated, the visualization of the 1-skeleton of the complex allows us to perceive the topological layout of high-dimensional data, and identify subgroups,

relationships and the overall positioning of the observations.

However, we have only scratched the surface of applying topology to data mining. A whole wealth of obtainable topological information remains out there.
Lets take a moment to look at the following motivating example:

Example 6.1. In [LPM03], A. Lee, K. Perdersen and D. Mumford apply the techniques of persistent homology to a data set of black and white pictures of urban and rural landscapes. They consider all the 3×3 patches of pixels, i.e. each patch is a point in $\mathbb{R}^9$, and each feature is the grey-scale value of a pixel.
In $\mathbb{R}^9$ they define what they call the *D-norm*, which is a measure of contrast in the *patch*. They choose the patches in the highest 20% of this norm value, and mean center and normalize the resulting data set so that the *lightest* pixel in each patch is taken as a 0 (i.e. is completely white), and each patch has *D-norm* 1.
With this data set at hand, they performed the Persistent Homology analysis to attempt to discover the homological structure present.
The barcode diagram analysis clearly pointed to a first Betti number of one.
To interpret this outcome, lets take a look at Figure 73 which is taken from [Car09]:

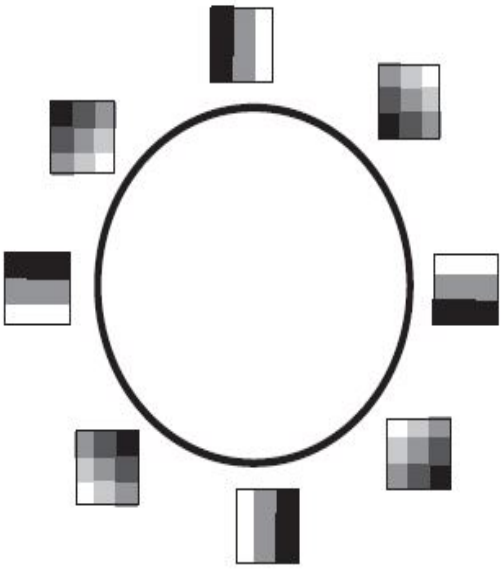

Figure 73:

The construction of the data set in [LPM03] is done in a fashion so that the points lie naturally in the 7-dimensional sphere in $\mathbb{R}^8$. The density estimation of the data points along the sphere through Voronoi cells shows that the data points are concentrated along a smooth 2-dimensional submanifold with 1-homological class of dimension 1, as was suggested by the barcode analysis.

The gist is that in real life photos, a 3×3 patch of pixels with high contrast will not appear in a lawless, *every which way*-manner.
It is natural that contrast in real photos appears gradually, with one color gradually giving way to the next in any direction (hence the circle).

Notice the *qualitative knowledge* on the nature of the data this has allowed us to point out. This example should convince the reader that there is important information to be learned from data sets by studying qualitative topological information from their vectorizations.

We said before that the theoretical role of simplicial complexes in algebraic topology is the fact that their homology can be algorithmically found. However, there are yet to be significant advances in providing a computationally viable calculation of the homology of simplicial complexes of even medium sizes, like the ones we construct with Topological Data Analysis.

Clearly, a wealth of information is being left untapped.

Persistence homology attempts to get a summary of topological information of $\check{C}(X, \epsilon)$ for several values of ϵ. When X is a compact riemannian manifold, we know that the number of barcodes from 0 to a certain $\epsilon_0 > 0$ are the correct Betti numbers for each dimension.
However, when $\mathbb{X}$ is a discrete point cloud, there is a value $\epsilon_0 > 0$ such that for $\epsilon \in [0, \epsilon_0]$ we have that $\check{C}(\mathbb{X}, \epsilon)$ is n isolated vertices (i.e. there are n barcodes in dimension 0 and no barcodes in other dimensions); and a value ϵ_1 where for $\epsilon > \epsilon_1$ we have that $\check{C}(\mathbb{X}, \epsilon)$ is a single $(n-1)$-dimensional simplex.
The idea of this discrete setting is to determine whether some barcodes exhibit a *context-long* behavior.
The fact that $\check{C}(\mathbb{X}, \epsilon_i) \subseteq \check{C}(\mathbb{X}, \epsilon_j)$ for $\epsilon_i < \epsilon_j$ was crucial.
Through this inclusion, the *tracking problem* becomes moot since there is a natural correspondence between vertices of TDA graphs for different values of ϵ, and edges *persist* with ϵ. This means that not only is homological information subject to the validation of *persistence*, but also other qualitative topological information like the existence of flares and partitioning into groupings we have sought out in TDA graphs.

The spirit of our approach has been essentially different, since we have focused on using the Nerve Theorem and our definition of *good discrete coverings* to build Čech complexes homotopically equivalent to the supposed underlying space X, independent from a parameter ϵ.
By using *coverings* and constructing $\check{C}(\mathfrak{U})$ instead of $\check{C}(\mathbb{X}, \epsilon)$, our complex becomes essentially *smaller*, and calculations are easier and less costly computationally.
However, in this context, there is no structural relationship between $\check{C}(\mathbb{X}, \mathfrak{U})$ and $\check{C}(\mathbb{X}, \mathfrak{B})$ for two different discrete coverings $\mathfrak{U}$ and $\mathfrak{B}$, as was the case for $\check{C}(\mathbb{X}, \epsilon_i)$ and $\check{C}(\mathbb{X}, \epsilon_j)$; so an extrapolation of the validation scheme from persistent homology is not possible.

The use of these approaches leaves us with a trade-off:

1. If we choose to use constructions of the form $\check{C}(\mathbb{X}, \epsilon)$, then the resulting complexes will be very high-dimensional, and computing their homologies is very costly. However, the fact that $\check{C}(\mathbb{X}, \epsilon_i) \subseteq \check{C}(\mathbb{X}, \epsilon_j)$ when $\epsilon_i \leq \epsilon_j$ allows us to formulate the notion of *persistence* and attempt to discover the true topological structure of the data, free from the *noise structure* for a single value ϵ.

2. If we choose to use constructions of the form $\check{C}(\mathbb{X}, \mathfrak{U})$, the resulting complexes will be considerably smaller (depending on the cover), so the computation of their homology should be less costly. Nevertheless, there is not a systematic relationship between the complexes for different coverings, so a validation scheme like the one designed in the theory of Persistent Homology is not applicable.

This reflection leaves us two main themes where improvements can be made: reducing computational cost of calculations of homological structure of data, and designing theoretical and practical enhancements to construct discrete covers and offer assurance of their *goodness*.
We are also in the awkward position of needing to provide the reader with justification as to why we choose the "*filter and cover*" approach over the "*ϵ-balls*" approach, which seemingly has the upper hand with it's validation possibility.

Addressing the first item, there are some alternative constructions to the Čech complex which result in computationally *easier* complexes. Lets take a look at these possibilities:

Definition 6.2. Let (X, d) be a metric space, and $\epsilon > 0$. The *Vietoris-Rips complex of X attached to ϵ*, denoted by $VR(X, \epsilon)$, is defined as the abstract simplicial complex with vertex set X and where $\{x_0, x_1, ..., x_k\}$ span a k-simplex if and only if $d(x_i, x_j) \leq \epsilon \ \forall i, j$ with $0 \leq i, j \leq k$.

The reason why the Vietoris-Rips complex provides a computationally cheaper option to the Čech complex is that it is completely determined by it's 1-skeleton.
In effect, for any subset $\{i_0, i_1, ..., i_l\} \subseteq \{0, 1, ..., k\}$, if $\{x_i, x_j\} \in VR(X, \epsilon)$ for any pair $i, j \in \{i_0, i_1, ..., i_l\}$, then by definition $\{x_{i_0}, ..., x_{i_l}\} \in VR(X, \epsilon)$.
This quality can be exploited in the algorithm that computes simplicial homology from a complex.

Additionally, even though we do not have a Nerve Theorem for the Vietoris-Rips complex, no homological information is lost. This fact is established by the following proposition.

Proposition 6.3. We have the following inclusions:

$$\check{C}(X, \epsilon) \subseteq VR(X, 2\epsilon) \subseteq \check{C}(X, 2\epsilon)$$

Proof. Let $\{x_0, ..., x_k\} \in \check{C}(X, \epsilon)$. By definition, $B_\epsilon(x_0) \cap ... \cap B_\epsilon(x_k) \neq \emptyset$, which means that $\exists \hat{x} \in X$ such that $d(\hat{x}, x_i) \leq \epsilon \ \forall i$.
Using triangular inequality we can therefore establish $\forall i, j$ that $d(x_i, x_j) \leq d(x_i, \hat{x}) + d(x_j, \hat{x}) \leq \epsilon + \epsilon = 2\epsilon$, and therefore $\{x_0, ..., x_k\} \in VR(X, 2\epsilon)$, establishing the first "$\subseteq$".

On the other hand, for any $i_0 \in \{0, 1, ..., k\}$, we have that $x_{i_0} \in B_{2\epsilon}(x_0) \cap ... \cap B_{2\epsilon}(x_k)$ since $d(x_{i_0}, x_j) \leq 2\epsilon$ for all j by definition. Therefore $B_{2\epsilon}(x_0) \cap ... \cap B_{2\epsilon}(x_k) \neq \emptyset$ and by definition $\{x_0, ..., x_k\} \in \check{C}(X, 2\epsilon)$, establishing the second "$\subseteq$".

□

The importance of this proposition lies in the fact that it allows for the following construction:

Let $0 < \epsilon_1 < \epsilon_2 < ... < \epsilon_n$ such that $\forall i$ we have that $2\epsilon_i < \epsilon_{i+1}$. Then we have the following structure:

$$\begin{array}{ccccccccc}
... & \xrightarrow{d} & \Delta_{k+2}^{\check{C}^i} & \xrightarrow{d} & \Delta_{k+1}^{\check{C}^i} & \xrightarrow{d} & \Delta_{k}^{\check{C}^i} & \xrightarrow{d} & \Delta_{k-1}^{\check{C}^i} \xrightarrow{d} ... \\
 & & \downarrow{\scriptstyle \psi} & & \downarrow{\scriptstyle \psi} & & \downarrow{\scriptstyle \psi} & & \downarrow{\scriptstyle \psi} \\
... & \xrightarrow{d} & \Delta_{k+2}^{VR^{2i}} & \xrightarrow{d} & \Delta_{k+1}^{VR^{2i}} & \xrightarrow{d} & \Delta_{k}^{VR^{2i}} & \xrightarrow{d} & \Delta_{k-1}^{VR^{2i}} \xrightarrow{d} ... \\
 & & \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle \varphi} \\
... & \xrightarrow{d} & \Delta_{k+2}^{\check{C}^{2i}} & \xrightarrow{d} & \Delta_{k+1}^{\check{C}^{2i}} & \xrightarrow{d} & \Delta_{k}^{\check{C}^{2i}} & \xrightarrow{d} & \Delta_{k-1}^{\check{C}^{2i}} \xrightarrow{d} ...
\end{array}$$

The maps φ and ψ are induced by the inclusions from Proposition 6.3, and are easily checked to be chain maps. It is also straightforward to see that the map ϕ from Section 2.2 is such that $\phi = \varphi \circ \psi$.

In the same spirit as in persistence homology, homological structure becomes tractable along different values of ϵ in the Vietoris-Rips complex.
Specifically, if we have $\epsilon_i < \epsilon_j$, and a sequence:

$$\check{C}(\mathbb{X}, \epsilon_i) \subseteq VR(\mathbb{X}, 2\epsilon_i) \subseteq VR(\mathbb{X}, 2\epsilon_j) \subseteq \check{C}(\mathbb{X}, 2\epsilon_j),$$

then for any k-homological class $[x]$ in $H_k^{\check{C}^i}(\mathbb{X})$ such that $\phi^*_{i \to 2j}([x]) \neq 0$, then it must be that $\psi([x]) \neq 0$ and $\varphi^{-1}(\phi^*_{i \to 2j}([x])) \neq 0$.
In other words, homological classes that *lived* in the simplicial homology of the Čech complex from ϵ_i through $2\epsilon_j$, must have also *lived* in the simplicial homology of the Vietoris-Rips complex from $2\epsilon_i$ to $2\epsilon_j$.

A word by word copy of the philosophy of persistence homology allows us to construct *barcode diagrams* for Vietoris-Rips complexes and make a judgment as to what barcodes

represent true underlying structure, and which are attributable to sampling noise.

There are many different ways to construct simplicial complexes from discrete samples of underlying spaces X which might also be useful in practice to discover topological information from X. Some examples include the *Delauney* complex, and the *strong* and *weak witness complex*. They can be referenced in [Car09].
The *Delauney* complex for example, produces simplices of much lower dimension than the Čech or Vietoris-Rips complexes, and in many practical applications have well related homotopy type to that of X. However, none of these constructions have the theoretical soundness of the two we have already explored.

Moving on to the second item, ensuring the *goodness* of discrete covers has been a central theme in this article. In Section 2.2 we discussed how a valuable validation scheme is checking the acyclicity of the data (in the persistence homology meaning of course).
This validation is very costly however. Its cost is contributed to by two factors: the inherent cost of calculating persistent homology, and the cost of going through all possible combinations of intersections of sets in the covering.
We just discussed how the Vietoris-Rips construction might alleviate the first cost.

For the second, it would be interesting to design an algorithm which exploits the repetitive nature of validating each intersection, to make this scheme viable.
For example, once $\mathfrak{U}_\alpha \cap \mathfrak{U}_\beta = \emptyset$, no intersection involving α and β must be considered.
Also, once we have constructed $\check{C}(\mathfrak{U}_\alpha, \epsilon)$ or $VR(\mathfrak{U}_\alpha, \epsilon)$, then $\check{C}(\mathfrak{U}_\alpha \cap \mathfrak{U}_\beta, \epsilon)$ and $VR(\mathfrak{U}_\alpha \cap \mathfrak{U}_\beta, \epsilon)$ are simply subcomplexes, and if this information is intelligently stored in the algorithm it can reduce the computational burden.

The problems we encountered in our application to football data also give further insight into how TDA can be improved.

Example 5.2 highlighted the importance of keeping track of appearing structure for different covers, since a single observation for a single parameter can be misleading.
This complication was further stressed in Example 5.4, where we discussed how the *grid* approach for generating target covers is flawed, since the *scale* of the problem is in all likelihood not uniform across the range of the filter function; and therefore important structure can fail to coincide for a single set of parameters.
As we explained in our examples, there is no true reason to pick a uniform grid. In areas with a high density of points, smaller parameters are needed to pick up structure than in less dense areas, so a uniform grid does nothing to balance these two needs out.

There are definite improvements to be made in determining the target cover through this reflection.
An idea (which we will not truly pursue yet) of this *non-uniform* grid applied to a 2-dimensional principal component filter function could be the following:

Project the points onto the first principal component, and through kernel density estimation obtain a function $f_1(x)$ which estimates the density of the filtered points onto this coordinate, and whose support is a slightly widened version of the support of the image of the filter.
The same can be done for the second principal component, obtaining a density function $f_2(y)$.
Next, for a fixed k_1, numerically find a point t_1 such that $\int_{-\infty}^{t_1} f_1(x)dx = \frac{1}{k_1}$ (recall that the support of $f_1(x)$ should be truncated). Once t_1 is found, t_2 can be found numerically by asking that $\int_{t_1}^{t_2} f_1(x)dx = \frac{1}{k_1}$.
This is done until t_{k_1-1} is found; and the equivalent construction is done on the second principal component for a fixed k_2, so that a sequence $s_1, ..., s_{k_2-1}$ is obtained.
Finally, our grid would be given by sets of the form $[t_i, t_{i+1}] \times [s_j, s_{j+1}]$ for $i \in \{1, ..., k_1\}$ and $j \in \{1, ..., k_2\}$.

This is not the only possible construction of course. Investigation into this problem should be encouraged.

This reflection also allows us to justify why the "*filter+cover*" approach is preferable to the "ϵ-*balls*" approach: density estimation is extremely vulnerable to the curse of dimensionality. The very nature of metrics in high-dimensional euclidean spaces pretty much ensures that a single parameter of ϵ will not be appropriate across the whole data set. To solve this problem in the original vectorization space in a similar fashion as we just proposed in the filter space would require density estimation in this space, and is easily unmanageable for all interesting data sets.

Another valuable advancement to be made in the field of **TDA Data Mining** is the elimination of the human component. We mentioned before how a valuable asset of our methodology was its *hypothesis free* approach; how it didn't require a previous hypothesis to gain *knowledge* of a data set.
However, this capacity has thus far relied on having a reasonably intelligent human observing the graph and interpreting the results.
The elimination of this human component would definitely be appealing.

To understand where this non-human systematization is needed, we should first identify where this component has so far been utilized. There are two main parts in the process:

1. Identifying the typical topological structure of **TDA Graphs**, like flares or loops
2. Tracking the stability of topological structure for different covers, and deciding on its *veracity*

There are some tools from graph theory which allow us to address the first process. In the specific case of flares, for example, the following definition comes in handy:

Definition 6.4. For a graph $G = (V, E)$, and a node $n \in V$, the *eccentricity* of n is defined

as:

$$e(n) = \sum_{m \in V} d(n, m),$$

where $d(n, m)$ is the standard distance notion in graphs, i.e. the minimum number of *edges* that must be crossed to get from n to m.

The reason this definition is useful is that nodes at the end of flares should obviously have higher eccentricity values.
An algorithm capable of recognizing flares systematically is as follows:

Calculate the *eccentricity* for all nodes in the graph, and produce a list of nodes ordered by this value form highest to lowest.
We iterate over the nodes of the list, at each step *processing* the node. By "*processing*" we mean integrate onto the set of already *processed* nodes by attaching it to any neighbors of it already *processed*.
In this system, a node being processed will result in one of three things:

1. It will become an isolated node if none of its neighbors have been processed.
2. It will be added to a connected component of already processed nodes.
3. It will merge together two connected components and create a new single one.

With this construction, each connected component of the process has a *birth time*, defined to be the step of the iteration on the list of nodes in which it appeared, and a *death time*, defined as the time at when this component was merged with another. A component's *lifespan* it the difference between these values.
Naturally, components with long lifespans should be flares. When running this flare detection algorithm, the programmer can simply fix a threshold for the lifespans of components after which these are considered as flares, or something of the sort.

Now, with this algorithm in place, we would like to track appearing flares for several parameters to verify that they are meaningful structural qualities of the data and not noise from a specific construction (for example we already saw on several occasions how a loop can be mistaken for two flares).
This could be done systematically in a *human-free* way with the following algorithm:

1. Fix the threshold on the lifespans for flares, so that the algorithm can decide which subgroups of nodes for a specific graph are flares
2. Fix a set of parameter values whose graphs will be constructed
3. Fix a threshold on the percentage of elements two components of two different graphs must share in order to be considered the same. For example, if 80% of the Holding

Midfielders appear in a flare for a different parameter, with the other maybe broken off for the specific graph, this would still be considered as the same structural element which led us to consider it a relevant grouping in the first place

4. Fix a threshold on the percentage of parameters for which a flare must appear in order to be considered *structural* and not *noise*.

5. Run the *eccentricity* algorithm on the graph for each parameter. Any *flare* which passes the test of each of the three thresholds is a significant grouping inside the data.

Remark 6.5. In [LSL$^+$13] the notion of eccentricity is used in a different but very valuable way:
Instead of identifying which components are flares as such, their algorithm was designed to *count* the number of flares.
They compared the outcomes for their data sets and compared with 1000 artificially generated data set, with the same dimensionality as the original data, with each entry being sampled by a Gaussian distribution. In this trial the artificial data never generated more than one flare, and so they conclude that flares are truly indicative of topological behavior.
This is valuable since we must be aware that **TDA Graphs** are by no means a sort of *picture* of the data. Their nature is essentially different, but we hope that in building and observing them information is to be gained. This reflection shows that flares, whatever they are in the original data vectorization, appear in what we empirically believe to be a *qualitative revelation* in *TDA Graphs* through filtering, therefore justifying our use of them. A similar analysis of the revelations of the ϵ-*balls* approach would have to be performed for this approach to even be appealing.
For more details consult [LSL$^+$13].

Finally, in what concerns the **TDA visualization** as such, like **TDA graphs**, a lot of visual potential is being lost by only plotting the 1-skeleton.
Deciding the best layout for a graph in $\mathbb{R}^2$ is sometimes an overlooked step in the process, but by no means is it a trivial one. In our methodology we used a computer program called Gephi, which had a list of distributions from which to choose from.
To our knowledge, there isn't a computer program available who can perform this task by representing the 2-skeleton in $\mathbb{R}^3$ (the Klein Bottle is an example that this isn't even always possible), but surely this would be an interesting development in the field of **TDA Data Mining**.

Appendix

1. **Full List of In-Game Statistics Available:**
   Appearances, Time Played, Starts, Substitutions On, Substitutions Off, Goals, First Goal, Winning Goal, Shots on target including goals, shots on target including woodwork, Blocked shots, Penalties taken, Penalty goals, Penalties saved, Penalties off-target, Penalties not scored, Direct Free-Kick goals, Direct free-kick on target, Direct

free-kick off target, Blocked direct free-kick, Goals from inside box, shots on from inside box, shots off from inside box, Blocked shots from inside box, Goals from outside box, shots on from outside box, shots off from outside box, Blocked shots from outside box, Headed goals, Headed shots on target, Headed shots of target, Headed Blocked Shots, Left Foot Goals, Left Foot Shots On Target, Left Foot Shots Off Target, Left Foot Blocked Shots, Right Foot Goals, Right Foot Shots On Target, Right Foot Shots Off Target, Right Foot Blocked Shots, Other Goals, Other Shots On Target, Other Shots Off Target, Other Blocked Shots Shots Cleared off Line, Shots Cleared off Line Inside Area, Shots Cleared off Line Outside Area, Goals Open Play, Goals from Corners Goals from Throws, Goals from Direct Free Kick, Goals from Set Play, Goals from penalties, Attempts Open Play on target, Attempts from Corners on target, Attempts from Throws on target, Attempts from Direct Free Kick on target, Attempts from Set Play on target, Attempts from Penalties on target, Attempts Open Play off target, Attempts from Corners off target, Attempts from Throws off target, Attempts from Direct Free Kick off target, Attempts from Set Play off target ,Attempts from Penalties off target, Goals as a substitute, Total Successful Passes All, Total Unsuccessful Passes All, Assists Key Passes, Total Successful Passes Excl Crosses Corners, Total Unsuccessful Passes Excl Crosses Corners, Successful Passes Own Half, Unsuccessful Passes Own Half, Successful Passes Opposition Half, Unsuccessful Passes Opposition Half, Successful Passes Defensive third, Unsuccessful Passes Defensive third, Successful Passes Middle third, Unsuccessful Passes Middle third, Successful Passes Final third, Unsuccessful Passes Final third, Successful Short Passes, Unsuccessful Short Passes, Successful Long Passes, Unsuccessful Long Passes, Successful Flick-Ons, Unsuccessful Flick-Ons, Successful Crosses Corners, Unsuccessful Crosses Corners, Corners Taken incl short corners, Corners Conceded, Successful Corners into Box, Unsuccessful Corners into Box, Short Corners, Throw Ins to Own Player, Throw Ins to Opposition Player, Successful Dribbles, Unsuccessful Dribbles, Successful Crosses Corners Left, Unsuccessful Crosses Corners Left, Successful Crosses Left, Unsuccessful Crosses Left, Successful Corners Left, Unsuccessful Corners Left, Successful Crosses Corners Right, Unsuccessful Crosses Corners Right, Successful Crosses Right, Unsuccessful Crosses Right, Successful Corners Right, Unsuccessful Corners Right, Successful Long Balls, Unsuccessful Long Balls, Successful Lay-Offs, Unsuccessful Lay-Offs, Through Ball, Successful Crosses Corners in the air, Unsuccessful Crosses Corners in the air, Successful crosses in the air, Unsuccessful crosses in the air, Successful open play crosses, Unsuccessful open play crosses, Touches, Goal Assist Corner, Goal Assist Free Kick, Goal Assist Throw In, Goal Assist Goal Kick, Goal Assist Set Piece, Key Corner, Key Free Kick, Key Throw In, Key Goal Kick, Key Set Pieces, Duels won, Duels lost, Aerial Duels won, Aerial Duels lost, Ground Duels won, Ground Duels lost, Tackles Won, Tackles Lost, Last Man Tackle, Total Clearances, Headed Clearances, Other Clearances, Clearances Off the Line, Blocks, Interceptions, Recoveries, Total Fouls Conceded, Fouls Conceded exc handballs pens, Total Fouls Won, Fouls Won in Danger Area inc pens, Fouls Won not in danger area, Foul Won Penalty, Handballs Conceded, Penalties Conceded, Offsides, Yellow Cards, Red Cards, Goals Conceded, Goals Con-

ceded Inside Box, Goals Conceded Outside Box, Saves Made, Saves Made from Inside Box, Saves Made from Outside Box, Saves from Penalty, Catches, Punches, Drops, Crosses not Claimed, GK Distribution, GK Successful Distribution, GK Unsuccessful Distribution, Clean Sheets, Team Clean sheet, Error leading to Goal, Error leading to Attempt, Challenge Lost, Shots On Conceded, Shots On Conceded Inside Box, Shots On Conceded Outside Box, Turnovers, Dispossessed, Big Chances, Big Chances Faced, Pass Forward, Pass Backward, Pass Left, Pass Right, Unsuccessful Ball Touch, Successful Ball Touch, Take-Ons, Overrun, Touches open play final third, Touches open play opp box, Touches open play opp six yards.

2. **Feature Selection for Player Problems:**
   The following is a list of the features that were conserved for the individual player problem data set:

   Time Played Starts, Substitute On, Substitute Off, Goals Shots On Target inc goals, Shots Off Target inc woodwork, Shots On from Inside Box, Shots Off from Inside Box, Shots On Target Outside Box, Shots Off Target Outside Box, Headed Goals, Headed Shots On Target, Headed Shots Off Target, Goals Open Play, Goals from Corners, Goals from Set Play, Attempts Open Play on target, Attempts from Corners on target, Attempts from Set Play on target, Attempts Open Play off target, Attempts from Corners off target, Attempts from Set Play off target, Total Successful Passes All, Total Unsuccessful Passes All, Assists, Key Passes, Total Successful Passes Excl Crosses Corners, Total Unsuccessful Passes Excl Crosses Corners, Successful Passes Own Half, Unsuccessful Passes Own Half, Successful Passes Opposition Half, Unsuccessful Passes Opposition Half, Successful Passes Defensive third, Unsuccessful Passes Defensive third, Successful Passes Middle third, Unsuccessful Passes Middle third, Successful Passes Final third, Unsuccessful Passes Final third, Successful Short Passes, Unsuccessful Short Passes, Successful Long Passes, Unsuccessful Long Passes, Successful Flick Ons, Unsuccessful Flick Ons, Successful Crosses Corners, Unsuccessful Crosses Corners, Successful Dribbles, Unsuccessful Dribbles, Successful Long Balls, Unsuccessful Long Balls, Successful Lay Offs, Unsuccessful Lay Offs, Through Ball, Successful open play crosses, Unsuccessful open play crosses, Touches, Duels won, Duels lost, Aerial Duels won, Aerial Duels Lost, Ground Duels won, Ground Duels lost, Tackles Won, Tackles Lost, Headed Clearances, Blocks, Interceptions, Recoveries, Total Fouls Conceded, Total Fouls Won, Fouls Won in Danger Area inc pens, Fouls Won not in danger area, Offsides, Dispossessed, Big Chances, Pass Forward, Pass Backward, Pass Left, Pass Right, Unsuccessful Ball Touch, Successful Ball Touch, Take Ons Overrun, Touches open play final third, Touches open play opp box, Touches open play opp six yards.

3. **Feature Selection for the Team Performance Problem**
   Recall that all features selected fell under one of two categories; their were either summed up for the whole squad, or were disaggregated between defenders, midfielders and strikers.

The list of features for each category is shown below.

(a) **Whole Squad Features:**
Goles a Favor, Goles en Contra, Penalty Goals, Penalties Saved, Goals from Inside Box, Shots On from Inside Box, Shots Off from Inside Box, Goals from Outside Box Shots On Target Outside Box, Shots Off Target Outside Box, Headed Goals, Goals from Corners, Goals from Set Play, Attempts from Corners on target, Attempts from Set Play on target, Attempts from Corners off target, Attempts from Set Play off target, Goals as a substitute, Successful Flick Ons, Unsuccessful Flick Ons, Successful Crosses Corners, Unsuccessful Crosses Corners, Corners Taken incl short corners, Corners Conceded, Successful Corners into Box, Unsuccessful Corners into Box, Key Corner, Key Set Pieces, Blocks, Total Fouls Conceded, Total Fouls Won, Fouls Won in Danger Area inc pens, Penalties Conceded, Offsides, Yellow Cards, Red Cards, Error leading to Goal, Error leading to Attempt, Challenge Lost, Take Ons Overrun, Goals Conceded, Goals Conceded Inside Box, Goals Conceded Outside Box, Saves Made, Saves Made from Inside Box, Saves Made from Outside Box, Saves from Penalty, Catches, Punches, Drops, Crosses not Claimed, Big Chances Faced

(b) **Disaggregated Features**
Time Played, Starts, Substitute On, Substitute Off, Goals, Shots On Target inc goals, Shots Off Target inc woodwork, Goals Open Play, Attempts Open Play on target, Attempts Open Play off target, Total Successful Passes All, Total Unsuccessful Passes All, Assists, Key Passes, Total Successful Passes Excl Crosses Corners, Total Unsuccessful Passes Excl Crosses Corners, Successful Passes Own Half, Unsuccessful Passes Own Half, Successful Passes Opposition Half, Unsuccessful Passes Opposition Half, Successful Passes Defensive third, Unsuccessful Passes Defensive third, Successful Passes Middle third, Unsuccessful Passes Middle third, Successful Passes Final third, Unsuccessful Passes Final third, Successful Short Passes, Unsuccessful Short Passes, Successful Long Passes, Unsuccessful Long Passes, Successful Dribbles, Unsuccessful Dribbles, Successful Long Balls, Unsuccessful Long Balls, Successful Lay Offs, Unsuccessful Lay Offs, Through Ball, Successful crosses in the air, Unsuccessful crosses in the air, Successful open play crosses, Unsuccessful open play crosses, Touches, Duels won, Duels lost, Aerial Duels won, Aerial Duels lost, Ground Duels won, Ground Duels lost, Tackles Won, Tackles Lost, Total Clearances, Headed Clearances, Interceptions, Recoveries, Dispossessed, Big Chances, Pass Forward, Pass Backward, Pass Left, Pass Right, Unsuccessful Ball Touch, Successful Ball Touch, Touches open play final third, Touches open play opp box, Touches open play opp six yards.

4. **Full List of *Holding* Midfielders**

Ashley Richards, Nigel de Jong, Leon Britton, Michael Carrick, Simon Lappin, Oriol Romeu, Jake Livermore, Michael Essien, Cheik Tiote, David Fox, Scott Parker, Kemy

Agustien, Mahamadou Diarra, Gareth Barry, Paul Scholes, Joe Allen, Danny Murphy, Yaya Toure.

5. **Full List of *Box to Box* Midfielders**

   Alexandre Song, Raul Meireles, Nenad Milijas, Anderson, James Milner, Yohan Cabaye, Jordan Henderson, Jonjo Shelvey, Jamie O'Hara, Luka Modric, Aaron Ramsey, Charlie Adam, Steven Gerrard, Tom Cleverley, Niko Kranjcar, Ross Barkley, Frank Lampard, Moussa Dembele, Joey Barton, Abou Diaby, Samir Nasri, Ryan Giggs, Sylvian Marveaux, Tomas Rosicky.

6. **Full List of *Wide* Midfielders**

   Jean Beausejour, Stewart Downing, James Morrison, Wes Hoolahan, Antonio Valencia, Wayne Routledge, Jermaine Pennant, Gylfi Sigurdsson, Chris Eagles, Craig Bellamy, Adam Johnson, Nathan Dyer, Gareth Bale, Shaun Maloney, Dirk Kuyt, Charles N'Zogbia, Alex Oxlade-Chamberlain, Andrey Arshavin, Ashley Young, Magaye Gueye, James McFadden, Nani, Andrea Orlandi.

7. **Full List of *Key* Midfielders**

   Adel Taarabt, Florent Malouda, Rafael van der Vaart, Juan Mata, David Silva, Steven Pienaar.

8. **Full List of Groups Found for Strikers**

   (a) **Flare Central Strikers 1:**
   Pappiss Demba Cisse, Aaron Wilbraham, Apostolos Vellios, Victor Anichebe, Peter Crouch, Jay Bothroyd, Nikica Jelavic, Leon Best, Steve Morison, Demba Ba, Kevin Davies, Leroy Lita, Pavel Pogrebnyak, Anthony Modeste, Pavel Pavlyuchenko, Nicklas Bendtner, Franco Di Santo, Connor Wickham, Djibril Cisse, Shane Long, Yakubu.

   (b) **Flare Central Strikers 2:**
   Dimitar Berbatov, Didier Drogba, Thierry Henry, Steven Fletcher, Orlando Sa, Denis Stracqualursi, Andy Carroll, Robin van Persie, Javier Hernandez, Edin Dzeko, Mario Balotelli, Emmanuel Adebayor, Marc-Antoine Fortune, David Ngog, Sergio Agüero.

   (c) **Flare Wingers:**
   Jermaine Defoe, Gervinho, Victor Moses, Theo Walcott, Daniel Sturridge, Hugo Rodallega, Somen Tchoyi, Sylvian Ebanks-Blake, James Vaughan, Peter Odemwingie, Clint Dempsey, Bobby Zamora, Danny Welbeck, Shola Ameobi, Fernando Torres, Asamoah Gyan, Gabriel Abgonlahor, Andrew Johnson, DJ Campbell, Fraizer Campbell, Andreas Weimann, Ivan Klasnic, Jonathan Walters, Sammy Ameobi, Jason Roberts.

9. **Full List of Group of Matches Between the Top 8:**

   Tottenham 0-0 Chelsea, Arsenal 0-0 Chelsea, Tottenham 0-1 Everton, Manchester City 1-0 Arsenal, Manchester United 2-1 Arsenal, Manchester United 3-3 Chelsea, Liverpool 2-1 Chelsea, Manchester United 1-1 Newcastle, Tottenham 1-3 Manchester United, Liverpool 3-0 Everton, Manchester United 3-1 Tottenham, Arsenal 5-3 Chelsea.

   Other matches in that flare which are not between top 8 teams are:

   Liverpool 0-1 Stoke, Swansea 3-2 Arsenal, Arsenal 1-1 Stoke, West Bromwich Albion 5-1 Wolverhampton, Wolverhampton 2-0 Fulham.

References

[Car09] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[Ghr08] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[LPM03] Ann B Lee, Kim S Pedersen, and David Mumford. The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54(1-3):83–103, 2003.

[LSL+13] PY Lum, G Singh, A Lehman, T Ishkanov, Mikael Vejdemo-Johansson, M Alagappan, J Carlsson, and G Carlsson. Extracting insights from the shape of complex data using topology. *Scientific reports*, 3, 2013.

[MC85] Glenn W Milligan and Martha C Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, 1985.

[RNQ+14] David Romano, Monica Nicolau, Eve-Marie Quintin, Paul K Mazaika, Amy A Lightbody, Heather Cody Hazlett, Joseph Piven, Gunnar Carlsson, and Allan L Reiss. Topological methods reveal high and low functioning neuro-phenotypes within fragile x syndrome. *Human brain mapping*, 35(9):4904–4915, 2014.

[TWH01] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[WBS05] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

[XJRN02] Eric P Xing, Michael I Jordan, Stuart Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 505–512, 2002.

[ZC05] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.