DEGREE PROJECT, IN MATHEMATICS , SECOND LEVEL
STOCKHOLM, SWEDEN 2015

A topological approach to data visualization

HENRIK LÖFBERG

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCI SCHOOL OF ENGINEERING SCIENCES

A topological approach to data visualization

HENRIK LÖFBERG

Master's Thesis in Mathematics (30 ECTS credits)
Master Programme in Mathematics (120 credits)
Royal Institute of Technology year 2015
Supervisor at KTH was Wojciech Chacholski
Examiner was Wojciech Chacholski

TRITA-MAT-E 2015:43
ISRN-KTH/MAT/E--15/43--SE

Royal Institute of Technology
School of Engineering Sciences

KTH SCI
SE-100 44 Stockholm, Sweden

URL: www.kth.se/sci

Abstract

Barcoding is a mathematical tool, to analyze data, which is based on the theory of persistent homology. In this thesis both Hierarchical Clustering and Barcoding are defined and analyzed according to three criterion: Continuity, Computability and Visualizability. It is also presented how the two methods, barcoding and hierarchical clustering, are connected and why barcoding, in some cases, is a generalized method of hierarchical clustering. Lastly some more question of interest, for better understanding barcoding, are stated.

Sammanfattning

Barcoding är ett matematiskt verktyg, för att analysera data, vilket bygger på teorin om ihållande homologi. I den här uppsatsen är både Hierarkisk Klustring och Barcoding definierade och analyserade med avseende på tre kriterier: Kontinuitet, Beräkningsbarhet och Visualiserbarhet. Det presenteras även hur de två metoderna, barcoding och hierarkisk klustring, är sammanlänkade och varför barcoding, i vissa fall, är en generaliserad metod av hierarkisk klustring. Tillsist är några fler frågor av intresse, för att bättre förstå barcoding, presenterad.

Acknowledgements

I would like to thank my supervisor Wojciech Chachólski for all the guidance and inspiring meetings. Every Wednesday I looked forward to the meetings not only to be able to throw some ideas out but also because of the energy in those meetings.

Since this thesis marks the end of my study at KTH I would like to thank everyone I got to know and who inspired me while studying here. I know a lot of us will keep in touch in the future.

Contents

Notation

Symbol	Meaning
$\mathbb{R}$	The real numbers.
$\mathbb{R}^+$	The positive real numbers including 0.
$\mathbb{N}$	The natural numbers including 0.
K	A simplicial complex.
$\pi_0(K)$	The set of connected components of K.
$\check{\mathrm{C}}(I, \sim)$	The Čech complex.
$\mathrm{VR}(I, \sim)$	The Vietoris-Rips complex.
$\mathrm{N}(I)$	The Nerve complex.
(X, d)	A metric space.
$\{B_r^{-1}(i)\}_{i \in f(I)}$	A sequence of subsets of X constructed from the balls of radius r in some pseudometric space over X.
$\check{\mathrm{C}}(I, \sim_r)$	The Čech complex constructed from $\{B_r^{-1}(i)\}_{i \in f(I)}$.
$\mathrm{VR}(I, \sim_r)$	The Vietoris-Rips complex constructed from $\{B_r^{-1}(i)\}_{i \in f(I)}$.
$\mathrm{N}(I, \sim_r)$	The Nerve complex constructed from $\{B_r^{-1}(i)\}_{i \in f(I)}$.
$H_n(K)$	The n:th homology group over the the simplicial complex K.
$d_{GH}(X, Y)$	The Gromow-Hausdorff distance between the two metric spaces X and Y.
$C(X)$	The collection of all non-empty subsets of X.
Π	A partition of X.
$\mathcal{B}$	An element of $C(X)$.
$\mathcal{P}(X)$	The collection of all partitions of a set X.
$\mathcal{D}(X)$	The collection of all dendrograms over a set X.
$\mathcal{U}(X)$	The collection of all ultrametric spaces over a set X.

$\theta(X)$	A dendrogram in $\mathcal{D}(X)$ for some set X.
$\mathcal{C}$	A category.
P	A partially ordered set.
$\mathcal{P}$	The category of a partially ordered set P.
$\mathcal{K}$	A category of simplicial complexes.
$\mathcal{V}$	A category of vector spaces.
$\Phi : \mathcal{P} \to \mathcal{C}$	A $\mathcal{P}$-persistence category.
$\Phi : \mathcal{P} \to \mathcal{K}$	A $\mathcal{P}$-persistence simplicial complex
$\Phi : \mathcal{P} \to \mathcal{V}$	A $\mathcal{P}$-persistence vector space
$Q(a, b)$	A bar from a to b.
$\mathcal{Q}$	A bar code.

Background

Now more than ever the need for tools to deal with big data is increasing. In a lot of cases the distribution behind the data is unknown and also it is rather a rule than an exception that the data is noisy. Therefore methods for analyzing big sets of high dimensional data which can deal with the noise are needed. In this thesis an algebraic approach to the problem will be presented.

In the first chapter the basics concepts used in the rest of the thesis are brought up. The second chapter explains hierarchical clustering. The third chapter explains how the method of barcoding is carried out. Both methods are discussed and examples are shown [4].

Almost all the examples in this thesis are generated by a package called javaplex for matlab (see [12]). It is a very useful tool for anyone who wants to learn more about barcoding and work with examples to understand the computational aspect better.

2.1 Simplicial Complexes

When trying to cluster or simplify data we want to end up with a structure that is easier to comprehend and visualize. The premise is that the data is sampled from a geometric object or a topological space. Since we are interested in the shape of the data we first need to build a structure that reflects the geometry of the space the data is sampled from. For this purpose the *simplicial complexes* are used.

Definition: Let $\mathcal{U}$ be a set (we call it a universe for our simplicial complexes). A simplicial complex K is a collection of non-empty finite subsets of $\mathcal{U}$ such that if $\sigma \subset \mathcal{U}$ belongs to K, then so does any non-epmty subset $\tau \subset \sigma$.

To be able to describe properties of simplicial complexes, we use the following dictionary. An element of simplicial complex K whose cardinality is $n+1$ is called a *simplex* of dimension n of K. A *vertex* of K is a simplex of K of dimension 0. A *face* of K is a simplex in K of maximal dimension. A *subcomplex* Q of K is a simplicial complex such that $Q \subset K$. For example the collection of all the subsets of a simplex σ in K is a subcomplex of K denoted by $\Delta[\sigma] \subset K$. The vertices in $\Delta[\sigma]$ are called the vertices of σ. A subcomplex $Q \subset K$ is called *full* if a simplex σ in K belongs to Q if and only if all its vertices belong to Q. Thus a full subcomplex $Q \subset K$ is uniquely determined by its set of vertices.

Two vertices $x, y \in K$ are said to be *directely connected* if $\{x, y\}$ is a simplex in K. Note that being directly connected is a symmetric and reflexive relation on the set of vertices of K. However it is not a transitive relation in general. Consider then the equivalence relation on the set of vertices of K generated by the relation of being directly connected. The set of equivalence classes of this equivalence relations is denoted by $\pi_0(K)$ and called the set of connected components of K. Let $[x] \in \pi_0(K)$ be an equivalence class. The full subcomplex on the set of vertices that belong to $[x]$ is called the connected component of K and is denoted by $K_{[x]}$. Note that $K = \coprod_{[x]\in\pi_0(K)} K_{[x]}$. We see later on how to use homology to determine the size of $\pi_0(K)$.

2.2 Forming Complexes

A computer cannot analyze a topological space directly but they can however analyze combinatorial objects e.g. simplicial complexes. The goal of this section is to introduce notations that allow us to connect a set of data drawn from a topological space to a simplicial complex.

A convenient starting point to form simplicial complexes is a reflexive and symmetric relation. Let us choose such a relation $\sim$ on a set I. We use it to construct two simplicial complexes, $\check{\mathrm{C}}(I, \sim)$ called *Čech complex* and $\mathrm{VR}(I, \sim)$ called *Vietoris Rips complex*:

$$\check{\mathrm{C}}(I, \sim) = \{\sigma \subset I \mid \text{there is } i \text{ in } I \text{ s.t., for any } j \text{ in } \sigma,\ i \sim j\}$$

$$\mathrm{VR}(I, \sim) = \{\sigma \subset \mathcal{U} \mid \text{for any } i \text{ and } j \text{ in } \sigma,\ i \sim j\}$$

Let us choose a sequence of subsets $\{U_i \subset X\}_{i\in I}$ of a given set X. Such a sequence leads to the following reflexive and symmetric relation of the set of indexes I:

$$i \sim j \text{ if and only if } U_i \cap U_j \neq \emptyset.$$

We can then use this relation to form two simplicial complexes $\check{\mathrm{C}}(I, \sim)$ and $\mathrm{VR}(I, \sim)$. Explicitly, $\sigma \in \check{\mathrm{C}}(I, \sim)$ if and only if there is U_j such that $U_j \cap U_i \neq \emptyset$ for any i in σ; and $\sigma \in \mathrm{VR}(I, \sim)$ if and only if $U_i \cap U_j \neq \emptyset$ for any i and j in σ. With a sequence of subsets $\{U_i \subset X\}_{i\in I}$ we can also form a so called the *nerve* complex:

$$N(I) := \{\sigma \subset I \mid \bigcap_{i\in\sigma} U_i \neq \emptyset\}$$

Assume that X is not just a set but a topological space. By choosing various sequences of subsets $\{U_i \subset X\}_{i\in I}$ we can then form three simplicial complexes. The natural questions is how much of the geometry of X these simplicial complexes capture? In this context the nerve complex is particularily important since the *nerve theorem* states that if the chosen sequence $\{U_i \subset X\}_{i \in I}$ consists of open subsets, it covers X, and all non-empty finite intersections of its elements are contractible, then the associated nerve complex is weakly equivalent to the space X [3]. Unfortunately the nerve complex tend to be very big in real life situations and hence computationally expensive. Same tends to be true also for the Čech complex. It is therefore often the Vietoris Rips complex which is the preferable choice, even though it may not capture the entire homotopy type of X. The Vietoris Rips complex is also easier to store in a computer since we only need to store the vertices and edges, the higher dimensional simplexes are constructed from this information. However this also means that it can be computationally heavy since we do not have all the simplexes stored directly. Also we will see that Vietoris-Rips does often capture so called persistance features of X.

We have seen that we can construct a simplicial complex from a sequence of subsets and therefor it is natural to discuss how a sequence like this can be constructed. The method used in this thesis needs three things for the construction.

1. A map $f : X \to (X, d)$ from the topological space X to a pseudometric space (X, d). Remember that a pseudometric space is a metric space but $d(x, y) = 0$ does not imply that $x = y$.

2. A finite subset $I \subseteq X$. We could choose $I = X$ but from a computational standpoint X might be too big.

3. Also we need to choose a *resolution*, sometimes called *clustering parameter*, $r \in \mathbb{R}^+$.

Now for each $i \in I$ we build an r-ball in the pseudometric space $B_r(f(i)) = \{x \in (X, d) \mid d(x, i) < r\}$. By taking the inverse $f^{-1}(B_r(f(i)))$ we end up with a sequence of subsets in X as wanted. With short notation we denote this sequence as $\{B_r^{-1}(i)\}_{i \in f(I)}$.

Example: Let X be a topological space, $f : X \to (X, d)$ some map to a pseudomateric space (X, d) and let $I = X$. Then the family of open balls $\{B_r^{-1}(x)\}_{x \in f(X)}$ is a covering of X for all $r > 0$.

For short hand notation we write $\mathrm{N}(I, \sim_r)$, $\mathrm{VR}(I, \sim_r)$ and $\check{\mathrm{C}}(I, \sim_r)$ when referring to the complexes built by the ball sequence. This section will be closed with an important theorem showing an intimate connection between the Nerve complex and the Vietoris Rips complex.

Theorem: The following inclusions are true $\mathrm{N}(I, \sim_r) \subseteq \mathrm{VR}(I, \sim_{2r}) \subseteq \mathrm{N}(I, \sim_{2r})$

Proof. We start by showing the first inclusion $\mathrm{N}(I, \sim_r) \subseteq \mathrm{VR}(I, \sim_{2r})$. Let the simplex $\sigma = (\alpha_0, \alpha_1, \ldots, \alpha_k) \in \mathrm{N}(I, \sim_r)$. Since σ is in the Nerve complex this means that for any two $\alpha_i, \alpha_j \in \sigma$ the intersection of $B_r(\alpha_i)$ and $B_r(\alpha_j)$ is non empty. This implies that the distance between α_i and α_j must be smaller than $2r$ and hence $\sigma \in \mathrm{VR}(I, \sim_{2r})$.

To show the second inclusion $\mathrm{VR}(I, \sim_{2r}) \subseteq \mathrm{N}(I, \sim_{2r})$ let the simplex $\sigma = (\alpha_0, \alpha_1, \ldots, \alpha_k) \in \mathrm{VR}(I, \sim_{2r})$. Then for any two $\alpha_i, \alpha_j \in \sigma$ the distance between the points is less than $2r$ and hence $\alpha_i \in B_{2r}(\alpha_j)$ for any $0 \leq i, j \leq n$. Then the intersection $\bigcap_i B_{2r}(\alpha_i) \neq \emptyset$ and hence $\sigma \in \mathrm{N}(I, \sim_{2r})$ which completes the proof. □

Note that the theorem does not guarantee us that the Nerve- and Vietoris Rips complex will be isomorphic for some choices of the clustering parameters of respective complex. Rather is shows that the clusterings are similar in how they form as we increase the clustering parameter.

2.3 Multiresolution

Let X be a topological space and $f : X \to (X, d)$ a map to some pseudometric space. Then we need to make a choice regarding which resolution r to use when constructing the sequence of subsets. In general there is however not a single unique resolution that will capture the geometry of the data in an optimal way. Therefore we need a method of varying the resolution to capture the general geometry. This concept is called *multiresolution.*

Definition: Let $U = \{U_i\}_{i \in I}$ and $V = \{V_j\}_{j \in J}$ be two coverings of X. A *map of coverings* is defined as a function $g : I \to J$ such that $U_i \subseteq V_{g(i)}$ for all $i \in I$.

Note that a map of coverings leads to an induced map on the nerve of the coverings $N(g) : N(U) \to N(V)$ of simplicial complexes given by g on the vertices of the simplicial complexes. In the same way given a family of coverings $\{U_\alpha\}$ and a family of map of coverings $\{g_\alpha\}$ we obtain the induced chain of simplicial complexes.

$$N(U_0) \overset{N(g_0)}{\hookrightarrow} N(U_1) \overset{N(g_1)}{\hookrightarrow} \ldots \overset{N(g_{N-1})}{\hookrightarrow} N(U_N)$$

Example: Let X be a topological space and let $\{B_r^{-1}(i)\}_{i \in f(I)}$ be a sequence of subsets for some map f to a pseudometric space and a resolution r. For any $r' \in \mathbb{R}^+$ such that $r < r'$ we get the map of coverings

$$g : B_r^{-1}(i) \mapsto B_{r'}^{-1}(i)$$

and hence the induced map of simplicial complexes.

$$N(g) : N(\bigcup_{i \in f(I)} B_r^{-1}(i)) \hookrightarrow N(\bigcup_{i \in f(I)} B_{r'}^{-1}(i))$$

Figure 2.1 illustrates the point that the geometry of data is hard to capture with a single resolution. In this case it could be solved by only keeping a percentage of the points that are in the densest area of the data. But such a choice would be arbitrary and hence we want another method of doing it. By using multiresolution we will not miss the circle feature of the data since at small resolutions the circle will be captured [9].

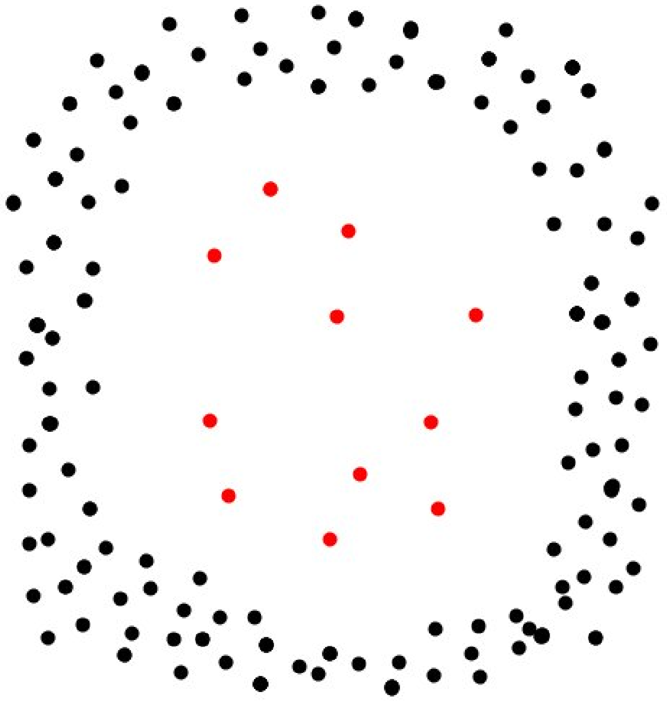

Figure 2.1: The data resembles a circle but the red data points in the middle would make clustering miss this feature at some choices of resolution.

2.4 Homology

Even though a computer is good at storing information needed to describe a simplicial complex, it is not good at comparing them as this process is computationally heavy. Instead of comparing complexes directly, one strategy is to first extract certain geometric invariants and then compare the invariants rather than complexes themselves. Homology is a convenient source of such invariants.

Let us fix a field which for computational purposes is often finite. Let K be a simplicial complex. Choose an ordering on the set of its vertices. Define Σ_n to be the set of *increasing* (with respect to the chose ordering) sequences $(x_0, \dots, x_n)$ of length $n+1$ of vertices of K. Let $S_n(K)$ to be the vector space generated by the set Σ_n. We use the ordering to define a boundary operator $\partial_n : S_n(K) \to S_{n-1}(K)$ as follows. For $\sigma = (x_0, \dots, x_n)$, $\partial_n(\sigma) := \Sigma_{i=0}^{n}(-1)^i(\sigma - \{x_i\})$, where $\sigma - \{x_i\}$ denotes the sequence obtained from σ by removing its i-th component.

One can observe that $\partial_n \circ \partial_{n+1} = 0$ and hence $Im(\partial_{n+1}) \subseteq Ker(\partial_n)$ which allows one to define $H_n(K) := Ker(\partial_n)/Im(\partial_{n+1})$. This vector space is called the n-th homology of K and it does not depend on the choice of ordering of vertices of K.

The dimension of the 0-th homology $H_0(K)$ is the number of connected components of K and hence coincides with the size $|\pi_0(K)|$. More generally, for $n > 0$, the vector space $H_n(K)$ counts the number of $n+1$ dimensional holes in the simplicial complex, see for example [2] for further explanation.

2.5 The Gromov-Hausdorff Distance

When working with metric spaces it is useful to be able to determine the degree of similarity between them. The *Gromov-Hausdorff distance* measures the similarity between metric spaces as seen in the definition.

Definition: Let (X, d_X) and (Y, d_Y) be two finite metric spaces. The *Gromov-Hausdorff distance* of X and Y is defined as:

$$d_{GH}(X,Y) = \min_{\substack{\varphi: X \to Y \\ \psi: Y \to X}} \max\{\text{dis}(\varphi), \text{dis}(\psi), \text{dis }(\varphi, \psi)\}.$$

where the *distortion* is defined in the following way

$$\text{dis}(\varphi) = \max_{x,x' \in X} |d_X(x,x') - d_Y(\varphi(x), \varphi(x'))|$$

$$\text{dis}(\psi) = \max_{y,y' \in Y} |d_Y(y,y') - d_X(\psi(y), \psi(y'))|$$

$$\text{dis}(\varphi, \psi) = \max_{x \in X, y \in Y} |d_Y(y, \varphi(x)) - d_X(x, \psi(y))|$$

Note that if dis(ψ, φ)=0 then $d_{GH} = 0$ but dis(ψ) = dis$(\varphi) = 0$ does not imply that dis(ψ, φ)=0. Informally the Gromov-Hausdorff distance measures how well the metric spaces can be embedded in each other and how well the functions ψ and φ are inverses of each other.

2.6 Dendrograms

Graphically a dendrogram is presented as a rooted tree and usually meant to present a hierarchical clustering. Formally a dendrogram is a finite set X together with a function $\theta[0, \infty) \to \mathcal{P}(X)$ where $\mathcal{P}(X)$ is the collection of all partitions, or clusterings, of X. There are four additional restrictions one puts on a dendrogram.

1. $\theta(0) = \{\{x_0\}, \{x_1\}, \cdot, \{x_n\}\}$, that is the space itself.

2. There exists a t_0 such that for every $t \geq t_0$ the partition is the single block.

3. if $r \leq s$ then $\theta(r)$ refines $\theta(s)$.

4. For each r there is an $\epsilon > 0$ such that $\theta(r) = \theta(t)$ where $t \in [r, r + \epsilon]$.

We denote a dendrogram over some set X by $\theta(X)$. Dendrograms are used when clustering data to visualize how the clusters are formed. Two examples of dendrograms can be seen in figure 2.2. However for clustering of big data sets dendrograms can be hard to interpret. Also it is computationally expensive to compare two dendrograms over sets with each other.

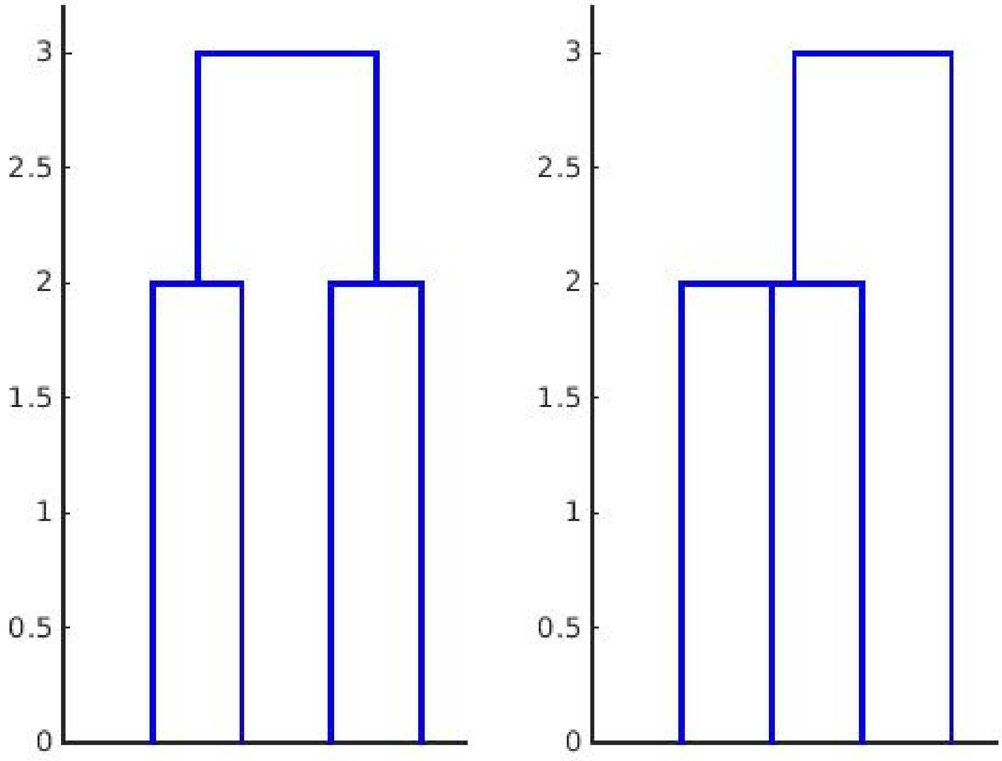

Figure 2.2: Two dendrograms visualizing two different clusterings of four data points.

Hierarchical Clustering

In this thesis we discuss the method of agglomerative hierarchical clustering which is when each data point starts out as its own cluster. We will however simply refer to it as *hierarchical clustering.*

3.1 Clustering Algorithm

A hierarchical clustering is a method for assigning to a metric space a dendrogram. This is done with a help of a so called *linkage function* which encodes a notion of a distance between two subsets of a given set. Recall that $C(X)$ denotes the collection all non-empty subsets of X. A linkage function on a set X is simply a bounded function of the form $l\colon C(X) \times C(X) \to \mathbb{R}_{\geq 0}$. Here are some of the most common and most used examples of linkage functions:

1. Single linkage: $l^{SL}(\mathcal{B}, \mathcal{B}') = \min_{x \in \mathcal{B}} \min_{y \in \mathcal{B}'} \mathrm{d}(x, y)$.

2. Complete linkage: $l^{CL}(\mathcal{B}, \mathcal{B}') = \max_{x \in \mathcal{B}} \max_{y \in \mathcal{B}'} \mathrm{d}(x, y)$.

3. Average linkage: $l^{AL}(\mathcal{B}, \mathcal{B}') = \frac{1}{|\mathcal{B}||\mathcal{B}'|} \Sigma_{x \in \mathcal{B}} \Sigma_{y \in \mathcal{B}'} \mathrm{d}(x, y)$

We recall now how a linkage function leads to a permutation invariant hierarchical clustering. Permutation invariance is important as it makes this process independent from how we enumerate and store the data. This method can be found in "Characterization, Stability and Convergence of Hierarchical Clustering Methods" (2009) by Gunnar Carlsson and Facundo Mémoli [6].

1. Fix a linkage function $l\colon C(X) \times C(X) \to \mathbb{R}_{\geq 0}$. For each resolution $r \geq 0$ form the equivalence relation $\sim_{l,r}$ on $C(X)$ where $\mathcal{B} \sim_{l,r} \mathcal{B}'$ if and only if there is a sequence $\mathcal{B} = \mathcal{B}_1, \ldots, \mathcal{B}_s = \mathcal{B}'$ of subsets of X such that $l(\mathcal{B}_i, \mathcal{B}_{i+1}) < r$ for $i = 1, \ldots, s-1$.

2. Define a sequence of partitions $\Pi_1, \Pi_2, \ldots$ of X as follows. We start with $\Pi_1 := \{\{x_1\}, \{x_2\}, \ldots, \{x_n\}\}$ and set recursively $\Pi_{i+1} = \Pi_i / \sim_{l,R_i}$ for $i \geq 1$ where

$$R_i := \min\{l(\mathcal{B}, \mathcal{B}') : \mathcal{B}, \mathcal{B}' \in \Pi_i, \mathcal{B} \neq \mathcal{B}'\}$$

3. Define $\theta^l : [0, \infty) \to \mathcal{P}(X)$ by $\theta^l(r) := \Pi_{i(r)}$ where $i(r) := \max\{i | R_i \leq r\}$.

3.2 Discussion of Properties

Even thought hierarchical clustering is a widely used method for analyzing data it has several drawbacks. To analyze data ultimately our method should:

1. have an output that is visualizable;
2. have outputs that are in some metric space with calculable distances,
3. be continuous and stable so its outcome is not sensitive to noise and small perturbations.

Unfortunately not all hierarchical clustering methods satisfy all the above requirements. In what follows we discuss shortly these aspects of hierarchical clustering.

3.2.1 Visualization

Small dendrograms provide a very convincing and useful visualization of the data. However, as the number of data points increases, so does the complexity of dendrograms. This makes it very hard to understand properties of such a dendrogaram, see for example figure 3.1.

3.2.2 Comparing Outputs: Dendrograms

To understand if it is hard to compare dendrograms some more theory is needed. The key observation is the identification of dendrograms over a set X with so called ultrametrics on X. This is the content of "Characterization, Stability and Convergence of Hierarchical Clustering Methods" (2009) by Gunnar Carlsson and Facundo Mémoli [6] and we recall below how it is done.

Theorem: Given a finite set X, there is a bijection $\psi : \mathcal{D}(X) \to \mathcal{U}(X)$ between the collection $\mathcal{U}(X)$ of all ultrametrics over X and the collection $\mathcal{D}(X)$ of all dendrograms over X, such that for any dendrogram $\theta \in \mathcal{D}(X)$ the ultrametric $\psi(\theta)$ gives the same hierarchical decomposition as θ. i.e

$$\text{for each r} \geq 0, \psi(\theta)(x,x') \leq r \iff x, x' \in \mathcal{B} \in \theta(r)$$

and the bijection is given by

$$\psi(\theta)(x,x') \mapsto \min\{r \geq 0 \mid \text{x and x' belong to the same block of } \theta(r)\}$$

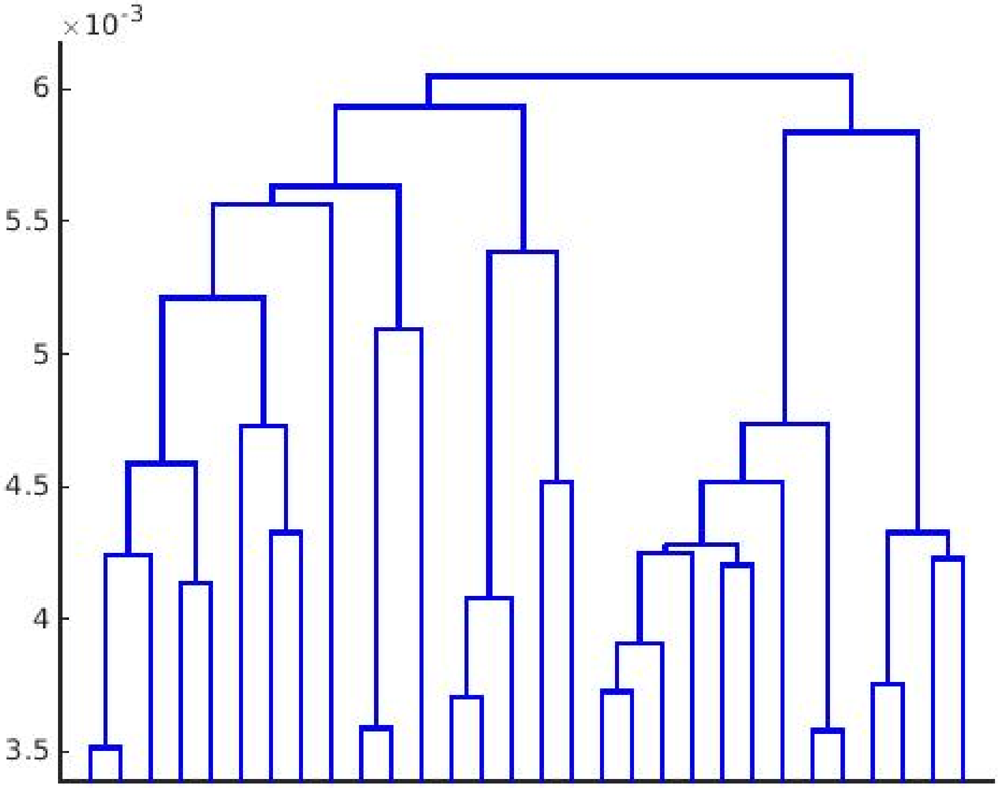

Figure 3.1: Already at 30 data points a dendrogram can be hard to interpret.

Proof. We first explain how to form an ultrametric on X given a dendrogram $\theta\colon [0,\infty) \to \mathcal{P}(X)$. Define the symmetric function $X \times X \to \mathbb{R}^+$

$$(x, x') \mapsto \min\{r \geq 0 \mid \text{x and x' belong to the same block of } \theta(r)\}$$

This map is well defined because of condition (4) in the definition of dendrograms. Also note that this is in fact an ultrametric.

With an ultramatric u, as with any other metric on X, we can use a single linkage to associate a dendrogram:

$$\theta(u)\colon [0,\infty) \to \mathcal{P}(X)$$

Recall that, for each $r \geq 0$, we use the single linkage to form the following relation:

$$x \sim_r x' \iff u(x, x') \leq r$$

The reflexivity and symmetry holds for any metric on X. However for an ultrametric this relation is also transitive and hence it is an equivalence relation on X. This implies that ψ is a bijection and that for any $\theta \in \mathcal{D}(X)$ the corresponding ultrametric $\psi(\theta)$ gives the same hierarchical decomposition. □

This theorem allows us to identify each dendrogram with its corresponding ultrametric space. The collection of all ultrametric spaces $\mathcal{U}(X)$ is a metric space itself with the Gromov-Hausdorff distance. Via the mentioned identification we can then import the Gromov-Hausdorff distance to define a metric on the collection of all dendrograms on X. This gives us a a way of measuring distances between dendrograms and understanding how far apart they can be. Unfortunately Gromov-Hausdorff distance is computationally very expensive and requires to perform $|X|!$ operations, where $|X|$ denotes the size of X. Thus even for a very small set X, it is basically computationally impossible to determine the Gromov-Hausdorff distance between dendrograms on X. We can use however this distance to understand the stability of a hierarchical clustering method which is the content of the next section.

3.2.3 Continuity

Among the three mentioned linkage functions above only the hierarchical clustering induced by the single linkage is continuous. This is unfortunate since single linkage has the *chaining phenomenon* which merges together separated clusters if there is even a very small path of points connecting them. This is illustrated in figures 3.2 and 3.3. In such cases complete linkage is a preferable methods as it clearly shows there are two separate clusters in the data set. Unfortunately complete linkage is not a stable hierarchical clustering method in general. Thus its usage for a particular data set has to be tested for continuity around that data set.

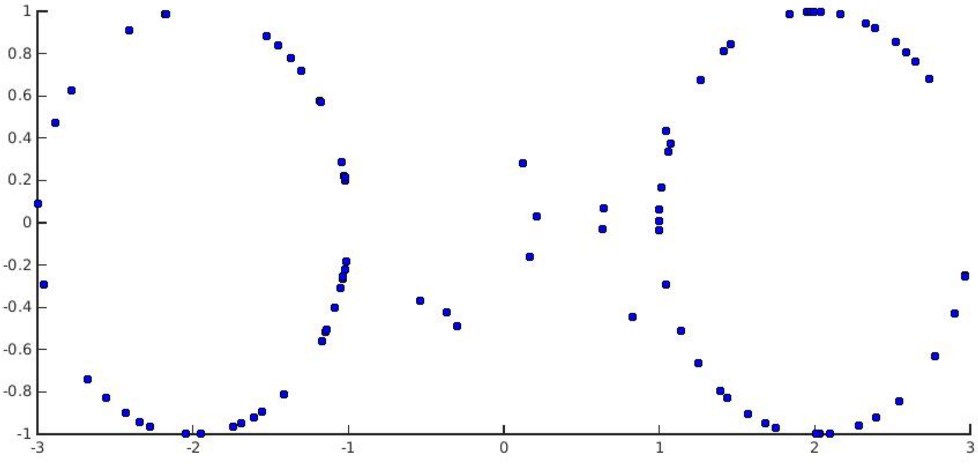

Figure 3.2: Some noisy data of two circles.

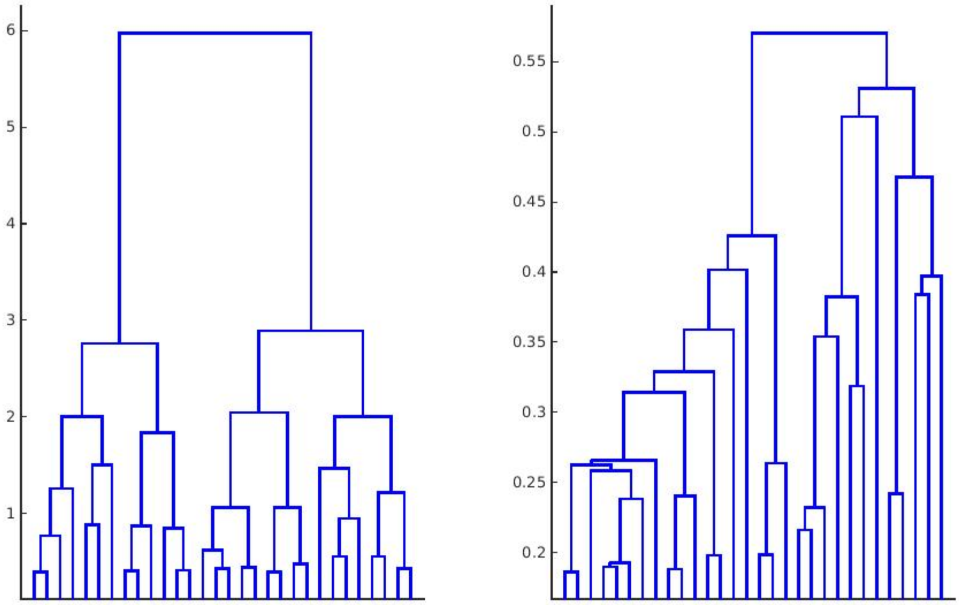

Figure 3.3: Left: The dendrogram produced from data with complete linkage. Right: The dendrogram produced from data with single linkage.

Barcoding

A more general approach of clustering data is by creating *bar codes.* This section describes the step by step process of going from a data set to a bar code. There are four main steps each described in the following order.

1. Data systems
2. Persistence simplicial complexes
3. Persistence vector spaces
4. Bar codes

A brief explanation of the four different steps is given here before further in depth explanation and results are discussed. A data system is a set of data X drawn from a topological space together with at least one function $f : X \to (X, d)$ from X to some pseudometric space (X, d) over X. Let $I \subseteq X$ and for each $r \in \mathbb{R}^+$ construct the sequence of subsets $\{B_r^{-1}(i)\}_{i \in I}$ which in turn allows us to define simplicial complexes, for example the nerve complex $N(I, \sim)$. The family of of simplicial complexes $\{N(I, \sim_r)\}_{r \in \mathbb{R}^+}$ together with the ordering on $\mathbb{R}^+$ is what we call an $\mathbb{R}^+$-persistence simplicial complex. Applying the homology functor on the family of simplicial complexes gives us a family of vector spaces. This family together with the ordering on $\mathbb{R}^+$ is what we call a $\mathbb{R}^+$-persistence vector space. From the $\mathbb{R}^+$-persistence vector space we can then draw the *bar code* which is the goal of the entire method.

4.1 Data Systems

Let X be a set which is drawn from some geometry. Then we can define a *data system* as follows.

Definition: A *data system* is a set X together with a family of functions, $f_i : X \to (X_i, d_i)$, where (X_i, d_i) is a pseudometric space. The functions f_i are called *measurements.*

The measurements $\{f_i\}$ should informally be though of as a way to capture something of interest in the data. A map f to pseudometric space could for example try to capture how similar two data points are. We will mainly focus on data systems with only one map in this thesis.

4.2 Persistence Simplicial Complexes

In this section we discuss how to construct a persistence simplical complex from a data systems. First the *persistence simplicial complexes* is defined and then some examples are given.

Definition: Let $\mathcal{C}$ be any category and P a partially ordered set. Define the category $\mathcal{P}$ with the object set P and the unique morphism $p_{x\leq y} : x \mapsto y$ whenever $x \leq y$. A $\mathcal{P}$*-persistence category* is a functor $\Phi : \mathcal{P} \to \mathcal{C}$ or in other words a set of objects $\{c_x\}_{x\in\mathcal{P}}$ together with a set of morphisms $\phi_{x\leq y} : c_x \to c_y$ whenever $x \leq y$. Also $\phi_{y\leq z} \circ \phi_{x\leq y} = \phi_{x\leq z}$ whenever $x \leq y \leq z$.

In the most examples of this thesis the partially ordered set P is the positive real line $\mathbb{R}^+$ and is called *resolution* [8]. Remember that given a data system we can construct a simplicial complex, for example the nerve complex, for each resolution $r \in \mathbb{R}^+$. Let $\mathcal{K}$ be the category of simplicial complexes with the objects the simplicial complexes constructed for each resolution $\{k_r\}_{r\in\mathbb{R}^+}$ and the trivial inclusion morphisms $\phi_{x\leq y} : k_x \hookrightarrow k_y$ if $x \leq y$. Note that then it is true that $\phi_{y\leq z} \circ \phi_{x\leq y} = \phi_{x\leq z}$ whenever $x \leq y \leq z$ and hence $\mathcal{K}$ is a category. We define a $\mathbb{R}$-*persistence simplicial complex* as a functor $\Phi : \mathcal{R}^+ \to \mathcal{K}$ which maps:

- **Objects:** $\Phi(x) = k_x$.
- **Morphisms:** $\Phi(p_{x\leq y}) = \phi_{x\leq y}$ if $x \leq y$.

Example: Let X together with a function $f : X \to (X, d)$ to a pseudometric space be a data system. Let $I \subseteq X$ be a subset of X and for each resolution $r \in \mathbb{R}^+$ construct the nerve complex $\mathrm{N}(I, \sim_r)$. Let $\Phi : \mathbb{R}^+ \to \mathcal{K}$ be a functor where $\mathbb{R}^+$ is the category where the objects are the real positive numbers and the morphisms are $p_{x\leq y}$ from x to y when $x \leq y$. Let $\mathcal{K}$ be the category where $\mathrm{N}(I, \sim_r)$ are the objects and the morphisms are the trivial inclusions $\phi_{x\leq y} : \mathrm{N}(I, \sim_x) \hookrightarrow \mathrm{N}(I, \sim_y)$ for $x \leq y$.

As seen in the example the functor Φ is a $\mathbb{R}^+$-persistence simplicial complex. We think about Φ as a collection of simplicial complexes together with an ordering on them. Also we have the trivial inclusion maps from a simplicial complex lower in the ordering to one higher in the ordering. Another example is when we have more than one map to a metric space. Then a relation can be defined in a similar manner:

Example: Let $f_n : X \to (X_n, d_n)$ where $0 \leq n \leq N$ for some $N \in \mathbb{N}$ be a data system. Define the partial ordering $(x_1, x_2, \dots, x_N) \leq (y_1, y_2, \dots, y_N)$ if and only if $f_n(x) \leq f_n(y)$ for every $0 \leq n \leq N$. Let $I \subseteq X$ and construct the sequences of subsets $\{B_{r_n}(i)\}_{i\in I}$ in X for each f_n and resolution $r_n \in r = (r_0, r_1, \dots, r_N)$. Construct $\mathrm{N}(I, \sim_r) = \bigcap_{0\leq n\leq N} \mathrm{N}_n(I, \sim_r)$ where $\mathrm{N}_n(I, \sim_r)$ denotes the nerve complex constructed from $\{B_{r_n}(i)\}_{i\in I}$ and define the category of simplical complexes $\mathcal{K}$ where the morphisms are the trivial inclusion morphisms $\phi_{x\leq y} : \mathrm{N}(I, \sim_x) \hookrightarrow \mathrm{N}(I, \sim_y)$ if and only if $x = (x_1, x_2, \dots, x_N) \leq y = (y_1, y_2, \dots, y_N)$ according to the above definition.

Then define the category $\mathcal{R}^N$ where the objects are elements of $\mathbb{R}^N$ and the morphisms are $p_{x^N\leq y^N} : x^N \to y^N$ from the object $x^N = (x_0, x_1, \dots, x_N)$ to the object $y^N = (y_0, y_1, \dots, y_N)$ if only if $x_n \leq y_n$ for $0 \leq n \leq N$.

Finally we construct the $\mathbb{R}^N$-persistence simplicial complex $\Phi : \mathcal{R}^N \to \mathcal{K}$ by $\Phi(x^N) \mapsto \mathrm{N}(I, \sim_{x^N})$ and $\Phi(p_{x^N\leq y^N}) \mapsto \phi_{x^N\leq y^N}$.

Note that the $\mathcal{P}$-persistence objects form their own category. Let Φ and Ψ be functors from the category of a partially ordered set $\mathcal{P}$ to a category $\mathcal{C}$. We first note that the functors Φ and Ψ can be thought of as a collection of objects and morphisms denoted $\{c_x, \phi_{x\leq y}\}_{x,y\in P}$ and $\{d_x, \psi_{x\leq y}\}_{x,y\in P}$ respectively. Then a functor $F : \Phi \to \Psi$, or more concretely $F : \{c_x, \phi_{x\leq y}\} \to \{d_x, \psi_{x\leq y}\}$, is a family of morphisms $\{f_x\}$ such that $f_x : c_x \mapsto d_x$ and the following diagram commutes: [5]

$$\begin{tikzcd}
c_x \arrow[r, "f_x"] \arrow[d, "\phi_{x\leq y}"'] & d_x \arrow[d, "\psi_{x\leq y}"] \\
c_y \arrow[r, "f_y"] & d_y
\end{tikzcd}$$

4.3 Persistence Vector Spaces

Even constructing persistence simplicial complexes is not enough to visualize data. These complexes are often of very high dimension and hence cannot be directly visualized. Also it is of high computational cost to compare two simplicial complexes. Hence we need further simplifications and the concept of *persistence vector spaces* is an intermediate step to be able to construct *bar codes.*

Definition: Let $\mathcal{P}$ be the category of a partially ordered set and let $\mathcal{V}$ be a category of vector space over some field k. The objects of $\mathcal{V}$ are vector space V_x where $x \in P$ and the morphisms $L(x, y) : V_x \to V_y$ are linear maps whenever $x \leq y$ such that

$$L(y, z) \circ L(x, y) = L(x, z) \text{ whenever } x \leq y \leq z$$

A *persistence vector space* is a functor $\Psi : \mathcal{P} \to \mathcal{V}$ from the category $\mathcal{P}$ of a partially ordered set to the category of vector spaces $\mathcal{V}$. Let $p_{x\leq y} \in \mathcal{P}$ be defined as earlier. Then $\Psi(p_{x\leq y}) = L(V_x, V_y)$ and $\Psi(x) = V_x$.

Let $\Phi : \mathcal{P} \to \mathcal{K}$ be a $\mathcal{P}$-persistence simplicial complex. We have already seen examples of how to construct Φ from a data system. Let $k_x \in \mathcal{K}$ be a simplicial complex then for any $n \in \mathbb{N}$ we can construct the vector space $V_x = H_n(k_x)$ where H_n denotes the homology functor.

To summarize where we are at before moving on to the next section. Given a data system we construct an $\mathbb{R}^+$-persistence simplicial complex

$$\Phi : \mathbb{R}^+ \to \mathcal{K}$$

and with the homology functor H_n construct an $\mathbb{R}^+$-persistence vector space:

$$\Psi : \mathbb{R}^+ \to H_n(\mathcal{K}) = \mathcal{V}$$

for some $n \in \mathbb{N}$. Remember that informally the dimension of a vector space $H_n(K_x)$ corresponds to the number of n-dimensional holes of K_x.

4.4 Bar Codes

In this section a way of creating *bar codes* from $\mathbb{R}$-persistence vector spaces is discussed. The reader should note that even though in the $\mathbb{R}$-persistence case bar codes will actually look like bar codes the goal of barcoding is simply to visualize data. It is still being worked on how to do bar codes for a $\mathbb{R}^n$-persistence system.

Let $\Phi : \mathbb{R}^+ \to \mathcal{V}$ be an $\mathbb{R}^+$-persistence system constructed from a finite data system. We form the equivalence classes

$$[V_x] = \{V_y \mid \text{either if } y \leq x \text{ then } L(y,x) \text{ is a bijection or else if } x < y \; L(x,y) \text{ is a bijection}\}$$

Since the data system is finite there is only a finite number of equivalence classes. Also, each class corresponds to a unique piece of the real line $\mathbb{R}^+$. Introduce the total ordering $[V_x] < [V_y]$ if and only if $x < y$. Since the data system is finite there is only a finite number of equivalence classes. Define the $\mathbb{N}$-persistence vector space with the objects V_n = the n:th equivalence class according to the ordering and the morphisms defined in the same manner. Now also note that since $L(i+1, i+2) \circ L(i, i+1) = L(i, i+2)$ every map can be decomposed $L(i, i+k) = \varphi^{i+k-1} \circ \varphi^{i+k-2} \circ \ldots \circ \varphi^i$ where $L(i, i+1) = \varphi^i$ is the unique map from V_i to V_{i+1}. Hence we only need to consider the objects $\{V_i\}_{i \in \mathbb{N}}$ together with the morphisms $\{\varphi^i\}_{i \in \mathbb{N}}$ [13]. We are now ready to simplify this system into what we call *bars*.

Definition: Let F be a 1-dimensional vector space over some field. We define a *bar* as an $\mathbb{N}$-persistence vector space $Q(a, b)$ by

- $Q(a,b)_t = 0$ for $t < a$
- $Q(a,b)_t = F$ for $a \leq t \leq b$
- $Q(a,b)_t = 0$ for $b < t$

where $a, b, t \in \mathbb{N}$. Let the linear maps $L(m,n) = Id_F$ for $a \leq m \leq n \leq b$ and else $L(m,n)$ is the zero map, $m, n \in \mathbb{N}$.

Informally a *bar* is a collection of 1-dimensional vector spaces with identity maps between them. Define $Q(a,+\infty)_t = F$ for all $a \leq t$ with the corresponding maps. Since an $\mathbb{N}$-persistence vector space is constructed from a finite data system each vector space V_n is finite dimensional. The fact that the data system if finite also implies that there is an $N \in \mathbb{N}$ such that all V_n are isomorphic for $n \geq N$. One way to easily visualize a bar is by simply drawing a line from a to b on the real line which is exactly the kind of simple visualization we are looking for. An example of how a bar code looks like can be seen in figure 4.1.

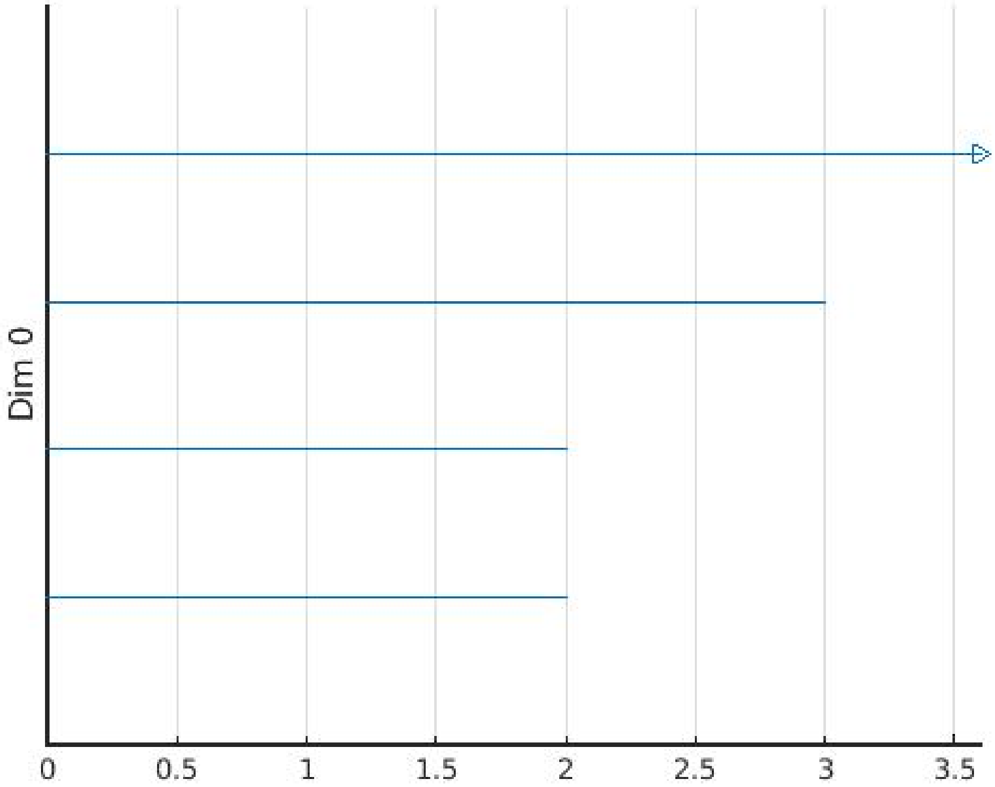

Figure 4.1: The bar code generated by the dendrograms in figure 2.2.

We first show that each $\mathbb{N}$-persistence vector space that is built from a finite data system can be decomposed into bars and then that this decomposition is unique.

Theorem: All $\mathbb{N}$-persistence F-vector spaces, $V = \{V_i, \varphi_i\}_{i \in \mathbb{N}}$, such that $dim(V_i) < \infty$ and such that there exists an $n \in \mathbb{N}$ such that φ_i is an isomorphism if $i \geq n$ can be decomposed into a sum of bars, i.e $V = \bigoplus_i Q(a_i, b_i)$.

Proof. This proof is done by contradiction. Note that given the smallest $n \in \mathbb{N}$ such that every map after that is an isomorphism the sum $dim(V) = \Sigma_{i=1}^{n} dim(V_i)$ is finite. Now assume it is not true that V can be decomposed. Then there is a V that cannot be decomposed such that $dim(V)$ is the smallest.

$$V_1 \overset{\varphi_1}{\to} V_2 \overset{\varphi_2}{\to} \ldots \overset{\varphi_{n-1}}{\to} V_n \cong V_{n+1} \ldots$$

The map φ_1 must be a monomorphism since if it is not one can decompose $V_1 = W \oplus ker(\varphi_1) = W \oplus Q(1,1)^{dim(ker(\varphi_1))}$ for some W with smaller dimension than V_1. This is a contradiction since then

$$W \overset{\varphi_1}{\to} V_2 \overset{\varphi_2}{\to} \ldots \overset{\varphi_{n-1}}{\to} V_n \cong V_{n+1} \ldots$$

would be non-decomposable but with a smaller sum of dimensions. Hence

$$V_1 \overset{\varphi_1}{\hookrightarrow} V_2 \overset{\varphi_2}{\to} \ldots \overset{\varphi_{n-1}}{\to} V_n \cong V_{n+1} \ldots$$

where $\hookrightarrow$ denotes a monomorphism.

Now assume that φ_i is a monomophism for $i \leq k$ where $k \in \mathbb{N}$

$$V_1 \overset{\varphi_1}{\hookrightarrow} V_2 \overset{\varphi_2}{\hookrightarrow} \ldots \overset{\varphi_k}{\hookrightarrow} V_k \overset{\varphi_{k+1}}{\to} \ldots \overset{\varphi_{n-1}}{\to} V_n \cong V_{n+1} \ldots$$

We want to show that φ_{k+1} is a monomorphism by contradiction and hence assume φ_{k+1} is not a monomorphism. Hence φ_{k+1} has a kernel $ker(\varphi_{k+1})$. If $ker(\varphi_{k+1}) \cap Im(\varphi_k) = \emptyset$ then we can decompose $V_k = W_k \oplus ker(\varphi_{k+1})$ and by the same argument as before this is a contradiction.

If $ker(\varphi_{k+1}) \cap Im(\varphi_k) \neq \emptyset$ we can assume, without loss of generality, that $dim(ker(\varphi_{k+1}) \cap Im(\varphi_k)) = 1$ and simply denote the intersection by F. Since φ_i, $i \leq k$, are monomorphisms we can write

$$W_1 \oplus F \overset{\varphi_1}{\hookrightarrow} W_2 \oplus F \overset{\varphi_2}{\hookrightarrow} \ldots \overset{\varphi_k}{\hookrightarrow} W_k \oplus F \overset{\varphi_{k+1}}{\to} \ldots \overset{\varphi_{n-1}}{\to} V_n \cong V_{n+1} \ldots$$

which means we can decompose $V = W \oplus Q(1,k)$ and hence W would be a $\mathbb{N}$-persistence vector space such that $dim(W) < dim(V)$, which is a contradiction, and hence φ_i is a monomophism for all i.

The last step of the proof is to note that since that each φ_i is a monomophism V_1 must be the zero vector space since else we could write $V = W \oplus Q(1, +\infty)^{dim(V_1)}$ which would be a contradiction. This must then be true also for V_2 and so on which gives

$$0 \overset{\varphi_1}{\hookrightarrow} 0 \overset{\varphi_2}{\hookrightarrow} \ldots \overset{\varphi_n}{\hookrightarrow} 0 \cong 0 \ldots$$

which is a contradiction and completes the proof. □

By studying this proof we can see exactly what the bars represent in the structure. A bar from a to b is simply a hole in the collection of simplicial complexes that exists from index a to index b in the sequence of simplicial complexes. The uniqueness of this decomposition is also important to show since it guarantees we do not interpret the same data in two different ways.

Theorem: The decomposition of an $\mathbb{N}$-persistence vector $V = \{V_i, \varphi_i\}_{i\in\mathbb{N}}$ space into bars is unique.

Proof. Assume there are two different decompositions

$$\bigoplus_{a_i, b_i \in \mathbb{N}} Q(a_i, b_i) \cong \bigoplus_{c_j, d_j \in \mathbb{N}} Q(c_j, d_j)$$

We want to show that these decompositions in fact must be identical in the sense that there exists a bijection f such that

$$f(Q(a_i, b_i)) = Q(c_j, d_j) \text{ if and only if } a_i = c_j \text{ and } b_i = d_j.$$

or more informally the sums are the same up to indexing. We will show this by induction on a_i. First let $a' = \min_{i\in I} a_i$ and pick the corresponding collection of bars $\{Q(a_k, b_k)\}_{k\in K}$ where $K = \{k \in I \; |a_k = a'\}$. Note that $\{Q(a_k, b_k)\}_{k\in K}$ must have a function f as defined above on it otherwise the decompositions would not be isomorphic.

Now assume f exists on a collection of bars, constructed as above, for all $a_i \leq n$ for some $n \in \mathbb{N}$. Let $a' = \min\ _{i\in I} a_i$ such that $a' > n$. We want to show that f also is defined on $\{Q(a_i, b_i)\}_{a_i \leq a'}$. By the exact same argument as for the base case we see that f must exist on $\{Q(a_i, b_i)\}_{a_i \leq a'}$ since otherwise the decompositions would not be isomorphic which completes the proof. □

4.5 Discussion of Properties

We analyze barcoding with respect to the same criteria as hierarchical clustering. As a remainder we want the method to:

1. have an output that is visualizable;
2. have outputs that are in some metric space with calculable distances,
3. be continuous and stable so its outcome is not sensitive to noise and small perturbations.

4.5.1 Visualization

In both processes, barcoding and hierarchical clustering, the first steps are the same and involve building a persistence simplicial complex. In the case of hierarchical clustering the simplicial compexes are discrete. By taking the homology we then get a persistence vector space. This can be illustrated as follows:

$$\text{Data} \xrightarrow{\text{hierarchical clustering}} \text{Dendrograms} \xrightarrow{H_0} \text{Persistance vector spaces}$$

The result can be then decompose into bars which is easily visualizable. Note that in this last step more information is removed as there are more isomorphisms of persistance vector spaces than isomprphisms of dendrograms. In this way we identify non-isomorphic dendrograms. Figure 2.2 gives an example of two non-isomorphic dendrograms which have the same bar code, seen in figure 4.1.

An example of how to plot barcodes is seen in figure 4.2. Note that a bar $Q(a, b)$ is represented by a line starting at a and ending at b. For a high number of bars this plot can get hard to read and we simplify it transposing all the bars to start at zero:

$$Q(a, b) \to Q(0, b - a).$$

Note that in doing this we assume the *length* of the bars contains most of the information. Additionally, since we do not care about the indexing of the bars we can sort them according to length to make it even easier to visualize. An example of this can be seen in figure 4.3.

4.5.2 Comparing Outputs: Bar Codes

To discuss the comparing of bar codes first a distance between them has to be defined. We will here use a special case of the bottleneck distance (see [10]). Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be

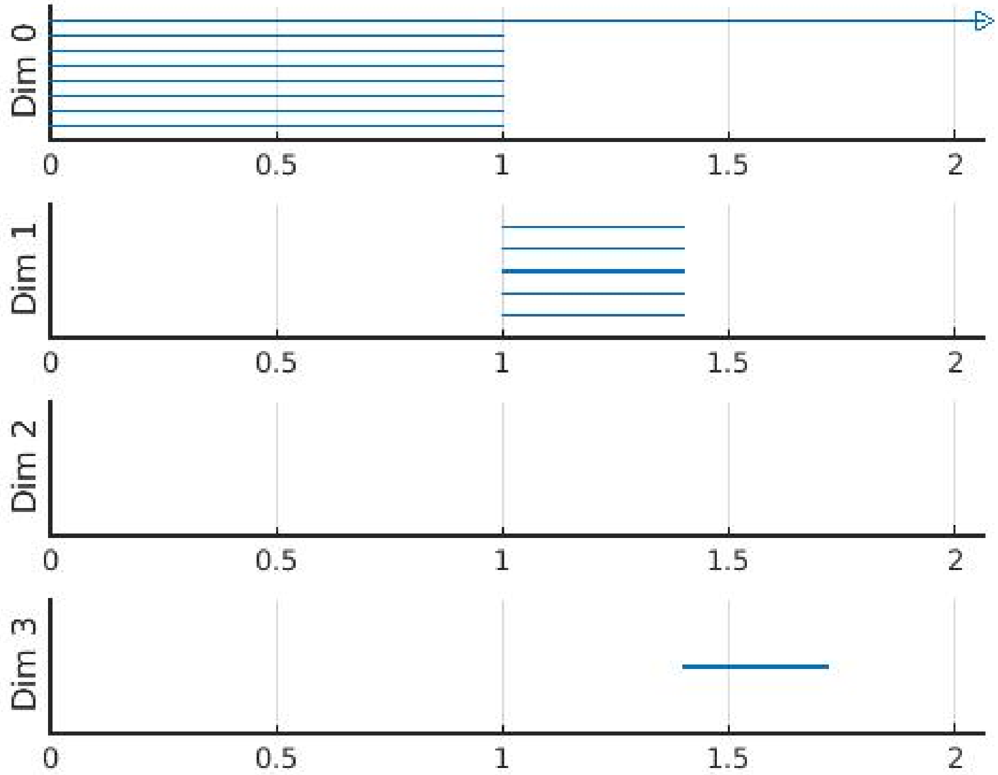

Figure 4.2: A plot of barcodes up to dimension 3. Dim 0 in the figure corresponds to H_0, that is the components. More generally Dim N corresponds to H_N or the $N+1$ dimensional holes.

bar codes and let the length of the bars be defined as $|Q(a,b)| = |b-a|$. We then define the distance between the bar codes $d_B(\mathcal{Q}_1, \mathcal{Q}_2)$ as the smallest $\epsilon > 0$ such that there is a bijection $f\colon \mathcal{Q}_1 \to \mathcal{Q}_2$ from all the bars in $\mathcal{Q}_1$ of length greater than ϵ to all the bars of $\mathcal{Q}_\in$ of length greater than ϵ and

$$|Q(a,b) - f(Q(a,b)) < \epsilon|.$$

Now given bar codes that are sorted according to length it is fairly easy to approximate ϵ even though no exact complexity can be stated here. Also we can choose which bar codes we want to analyze by restricting our comparison to the homology H_n of interest [1].

4.5.3 Continuity

The process of creating bar codes described in this thesis with the nerve complex is continuous with respect to the bottleneck distance [4] [10]. It should be noted thought that by looking at $\pi_0(K)$ of the persistence simplicial complex the dendrogram will correspond

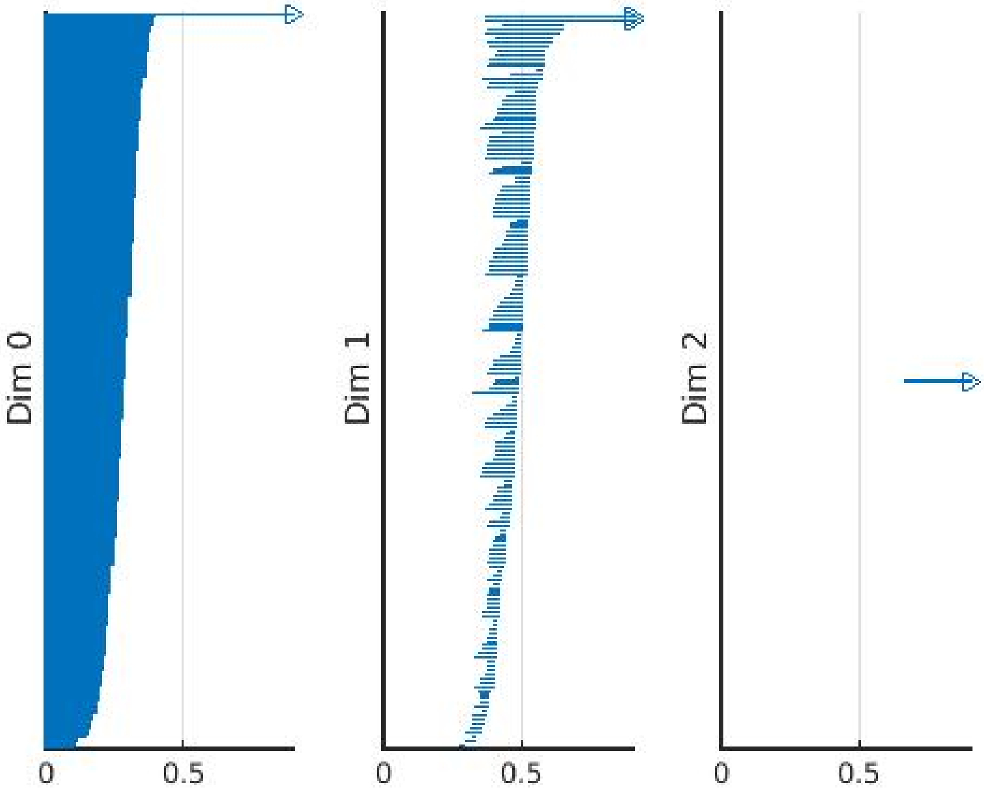

Figure 4.3: Barcodes obtained from Vietoris-Rips on a torus. They are not transposed to start at 0 but sorted by length.

to the one of single linkage hierarchical clustering.

Conclusions and Closing Thoughts

In general we can think of barcoding as the process of hierarchical clustering but more general since we capture higher dimensional geometry and not only the clustering. Of course there are some drawbacks like the fact that there is not obvious way of how the counterpart of complete linkage clustering would look like in the barcoding case.

In this section we will first conclude the the last two chapters. Then some ideas for further work will be presented together with an idea of how to combine complete linkage with bar coding. The idea is in no way analyzed in depth because of time constraints but it will be argued of why it could be of interest to pursue.

5.1 Conclusions

Barcoding is, in general, easier to visualize than hierarchical clustering and gives us higher dimensional geometric information. We have noted however that barcoding can interpret different dendrograms as the same seen as bar codes. This is however something we expect and that is desired since we want to make the output easier to visualize. Also it should be noted that the complexity is higher for barcoding ($O(n^3)$) (see [13]) than single- and complete linkage ($O(n^2)$) (see [11] [7]). On the other hand even though barcoding is a more expensive method the bars are cheaper to compare to one another.

Hierarchical clustering is continuous for single linkage and barcoding is continuous for all three methods presented in this thesis. This will be discussed a bit more in depth in the next section.

5.2 Combining Complete Linkage and Bar Codes

Let K be a simplicial complex constructed with any of the three methods presented in this thesis. We obtain an $\mathbb{R}^+$-persistence simplicial complex $\Phi : \mathbb{R}^+ \to \mathcal{K}$ by varying the resolution $r \in \mathbb{R}^+$. Construct the dendrogram θ by

$$\theta : r \mapsto \pi_0(K_r) = \pi_0(\Phi(r))$$

It is easily checked that this dendrogram corresponds to the single linkage hierarchical clustering algorithm dendrogram. To avoid the chaining phenomenon another approach is presented here.

1. Perform complete linkage at the data set.

2. Analyze the dendrogram and divide the data set in clusters.

3. Perform barcoding on the separate clusters.

The idea is to avoid chaining by first finding how the data clusters before we use barcoding. We want a method for analyzing the dendrogram that does not require an arbitrary choice of some parameter. One such method is the following method:

1. Find the cluster that persists the longest $\Pi_0 \neq \theta(+\infty)$, i.e. the longest leg in the dendrogram. (Note that we do not count the last cluster that exists to infinity.)

2. Find the cluster that persists the longest $\Pi_1 \neq \Pi_0$ and also overlaps with Π_0, i.e. there is some $r \in \mathbb{R}^+$ such that $\Pi_0, \Pi_1 \in \theta(r)$.

3. Repeat for Π_i until there are no more clusters.

By dividing the data sets in smaller parts before using barcoding the method should in general become less computational expensive as well ah avoiding the chaining phenomenon. The obvious drawback is of course that we have to either choose at what resolution we should divide the data into separate clusters or we end up with two resolutions to vary. There are three main questions that would be interesting to think a bit deeper about regarding this method.

1. Is there a good way to visualize the output?

2. What is the complexity?

3. How does this method change the higher homology captured compared to traditional barcoding?

5.3 Discussion Regarding Ultrametric Spaces

We have seen that hierarchical clustering is closely connected with ultrametric spaces. In this section we show that another class of metric rooms are of interest to better classify to understand barcoding better.

Preposition: The metric space (X, d) is an ultrametric if and only if $\mathrm{VR}(X, \sim_r)$ is a collection of disjoint simplexes for every $r > 0$.

Proof. First assume (X, d) is an ultrametric. Let y_r be the collection of all points $z \in X$ such that there exists a collection of points $x_i \in X$ such that $y = x_0, x_n = z$ and $d(x_i, x_{i+1}) < r$. Then note that $d(z, y) \leq$max $d(x_i, x_{i+1}) < r$, i.e. all vertices are connected to each other and the implication is done.
Now assume that $\mathrm{VR}(X, \sim_r)$ is a collection of disjoint simplexes for every r. This means that if the vertices $(x, y), (y, z)$ are in $\mathrm{VR}(X, \sim_r)$ then we have that (x, z) also must be in $\mathrm{VR}(X, \sim_r)$. Since (x, z) is in $\mathrm{VR}(X, \sim_r)$ as long as both (x, y) and (y, z) are in $\mathrm{VR}(X, \sim_r)$ it follows that $d(x, z) \leq \max(d(x, y), d(y, z))$ and the proof is complete. □

Corollary: If (X, d) is an ultrametric then $\mathrm{VR}(X, \sim_r) \to \pi_0(X)$ is a homotopy equivalence.

Proof. This is easy to see since all the clusters in $\mathrm{VR}(X, \sim_r)$ will be simplexes and hence they can be continuously retracted to points. □

Informally we can say that in an ultrametric space we will get no information by looking at the higher homology since everything is simple disconnected points.

An interesting question that now arises is for which metric rooms that are not ultrametrics that $\mathrm{VR}(X, \sim_r) \to \pi_0(X)$ will be a homotopy equivalence? It would be interesting if one could classify these rooms to better understand which rooms that contains higher dimensional geometrical data.

Bibliography

[1] Grzegorz Jabłoński Andrea Cerri, Barbara Di Fabio and Filippo Medri. Comparing shapes through multi-scale approximations of the matching distance. *Computer Vision and Image Understanding*, 121(0):43–56, 2014.

[2] M.A. Armstrong. *Basic Topology.* Springer, 1983.

[3] A. Björner. Topological methods. In M. Grötschel R. Graham and L. Lovátsz, editors, *Handbook of combinatorics*, page 1850. Elsevier Science, 1995.

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

[5] Gunnar Carlsson. Topological pattern recognition for point cloud data. *Acta Numerica*, 23:289–368, 5 2014.

[6] Gunnar Carlsson and Facundo Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *J. Mach. Learn. Res.*, 11:1425–1470, August 2010.

[7] D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.

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

[9] Facundo Mémoli Gurjeet Singh and Gunnar Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. *Eurographics Symposium on Point-Based Graphics*, 2007.

[10] Hyekyoung Lee, Hyejin Kang, M.K. Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *Medical Imaging, IEEE Transactions on*, 31(12):2267–2277, Dec 2012.

[11] R. Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[12] Andrew Tausz, Mikael Vejdemo-Johansson, and Henry Adams. JavaPlex: A research software package for persistent (co)homology, 2014.

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

TRITA -MAT-E 2015:43
ISRN -KTH/MAT/E--15/43--SE

www.kth.se