Persistent Homology: Interpretation and Stability

Tao Hou, University of Oregon

More on a point in PD

- Previously, we were able to formally define persistent homology and the topological summary it produces, aka. PD or barcode.
- We also looked at several common ways for building filtration out of data that are practical in different applications

More on a point in PD

- Previously, we were able to formally define persistent homology and the topological summary it produces, aka. PD or barcode.
- We also looked at several common ways for building filtration out of data that are practical in different applications
- Notice that our ultimate goal is to use persistent homology to infer the "shape" of data (i.e., homology inference)
- To do this, we need to fully understand the meanings of PD or barcode, from different aspects

More on a point in PD

- Previously, we were able to formally define persistent homology and the topological summary it produces, aka. PD or barcode.
- We also looked at several common ways for building filtration out of data that are practical in different applications
- Notice that our ultimate goal is to use persistent homology to infer the “shape” of data (i.e., homology inference)
- To do this, we need to fully understand the meanings of PD or barcode, from different aspects
- Moreover, we shall also see some properties that are critical to showing that persistent homology is a “reliable” way for inferring the shape

More on a point in PD

- Previously, we defined a persistence diagram (PD) as a set of points on the 2D plane above the diagonal

More on a point in PD

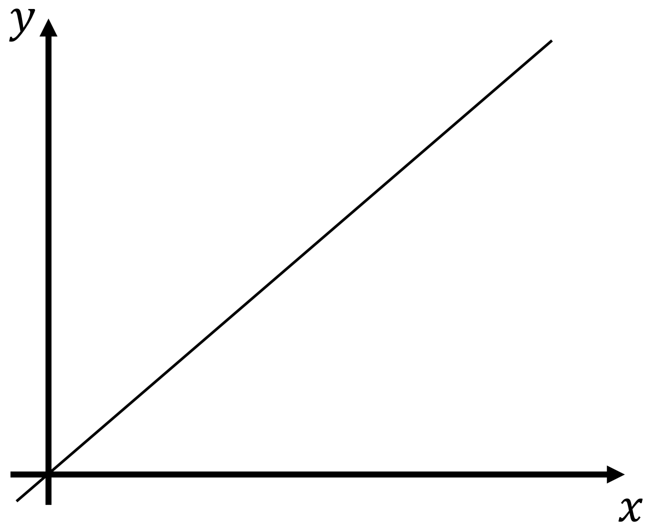
- Previously, we defined a persistence diagram (PD) as a set of points on the 2D plane above the diagonal
- But somehow, all we know about a point (b, d) in a PD is that
  - b is the "birth value" of a homological feature (hole)
  - d is the "death value" of it

More on a point in PD

- Previously, we defined a persistence diagram (PD) as a set of points on the 2D plane above the diagonal
- But somehow, all we know about a point (b, d) in a PD is that
  - b is the "birth value" of a homological feature (hole)
  - d is the "death value" of it
- Why above the diagonal?

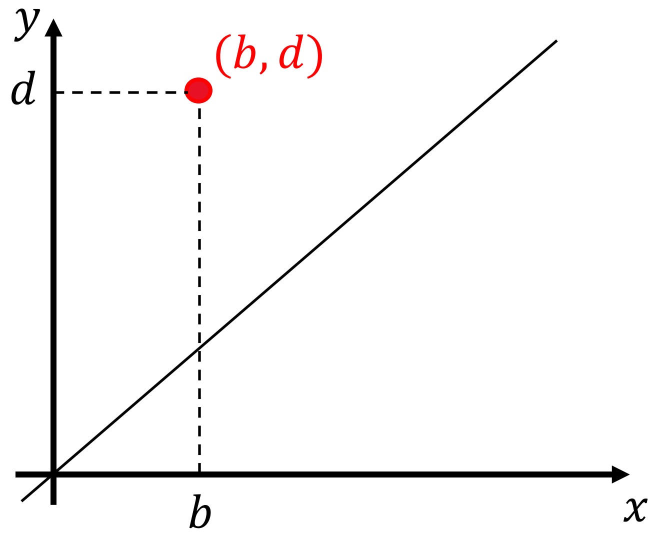
More on a point in PD

- Consider a point (b, d) in a PD



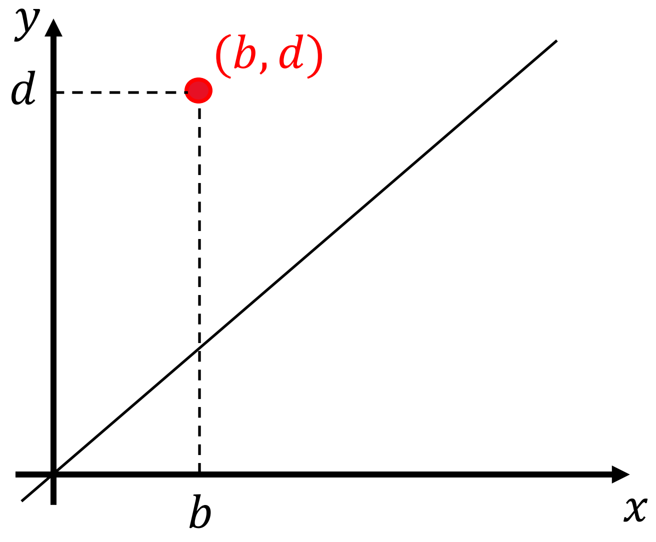
More on a point in PD

- Consider a point (b, d) in a PD



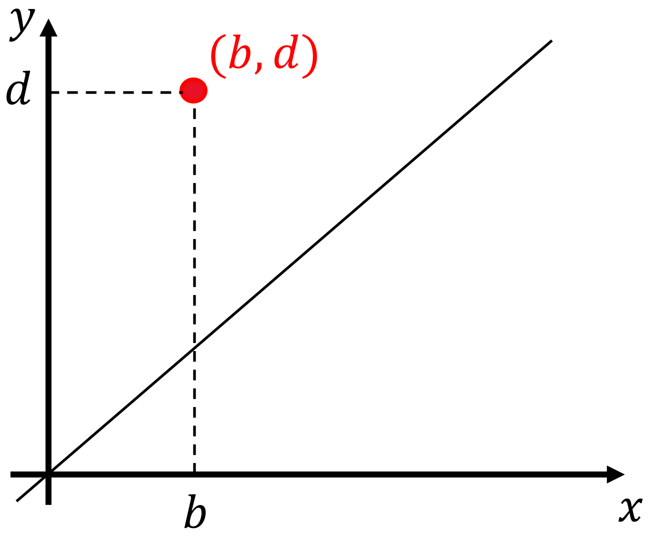
More on a point in PD

- Consider a point (b, d) in a PD
- We must have $b < d$ because something must be born before it can die



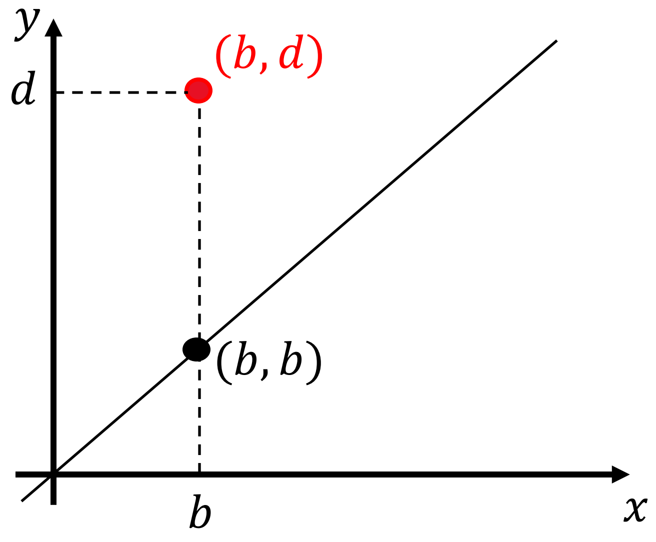
More on a point in PD

- Consider a point (b, d) in a PD
- We must have $b < d$ because something must be born before it can die
- If you look at the intersection of the vertical line passing (b, d) and and diagonal, which is (b, b)



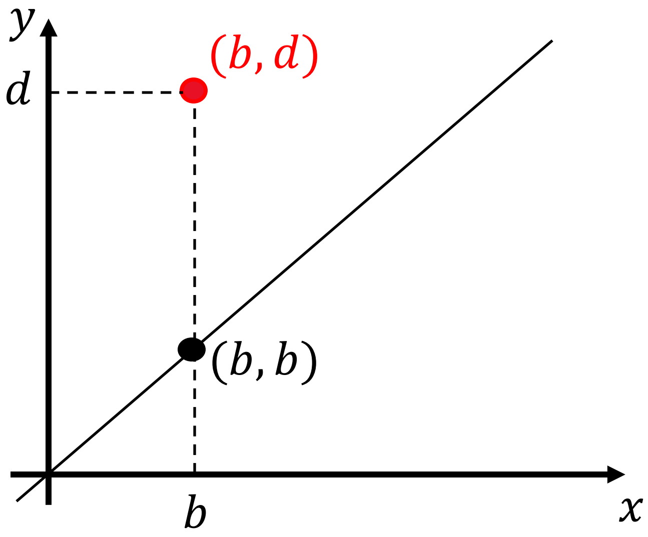
More on a point in PD

- Consider a point (b, d) in a PD
- We must have $b < d$ because something must be born before it can die
- If you look at the intersection of the vertical line passing (b, d) and and diagonal, which is (b, b)



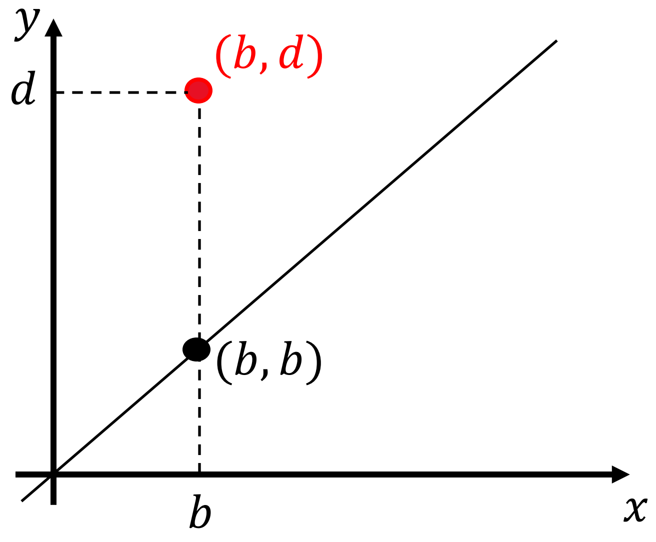
More on a point in PD

- Consider a point (b, d) in a PD
- We must have $b < d$ because something must be born before it can die
- If you look at the intersection of the vertical line passing (b, d) and and diagonal, which is (b, b)
- (b, d) must be above (b, b) because $b < d$



More on a point in PD

- Consider a point (b, d) in a PD
- We must have $b < d$ because something must be born before it can die
- If you look at the intersection of the vertical line passing (b, d) and and diagonal, which is (b, b)
- (b, d) must be above (b, b) because $b < d$
- So (b, d) is above the diagonal



Meaning of "**length**" of a PD interval

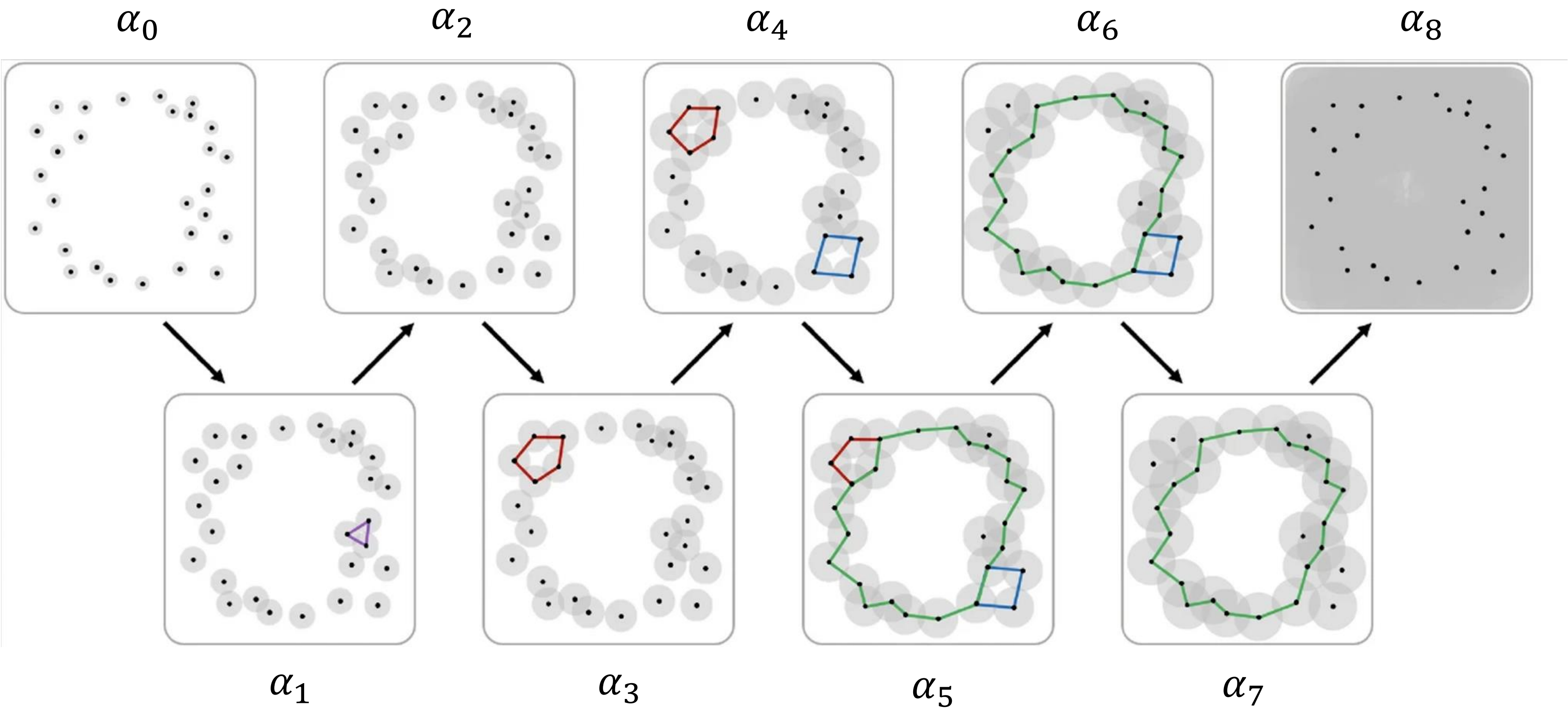
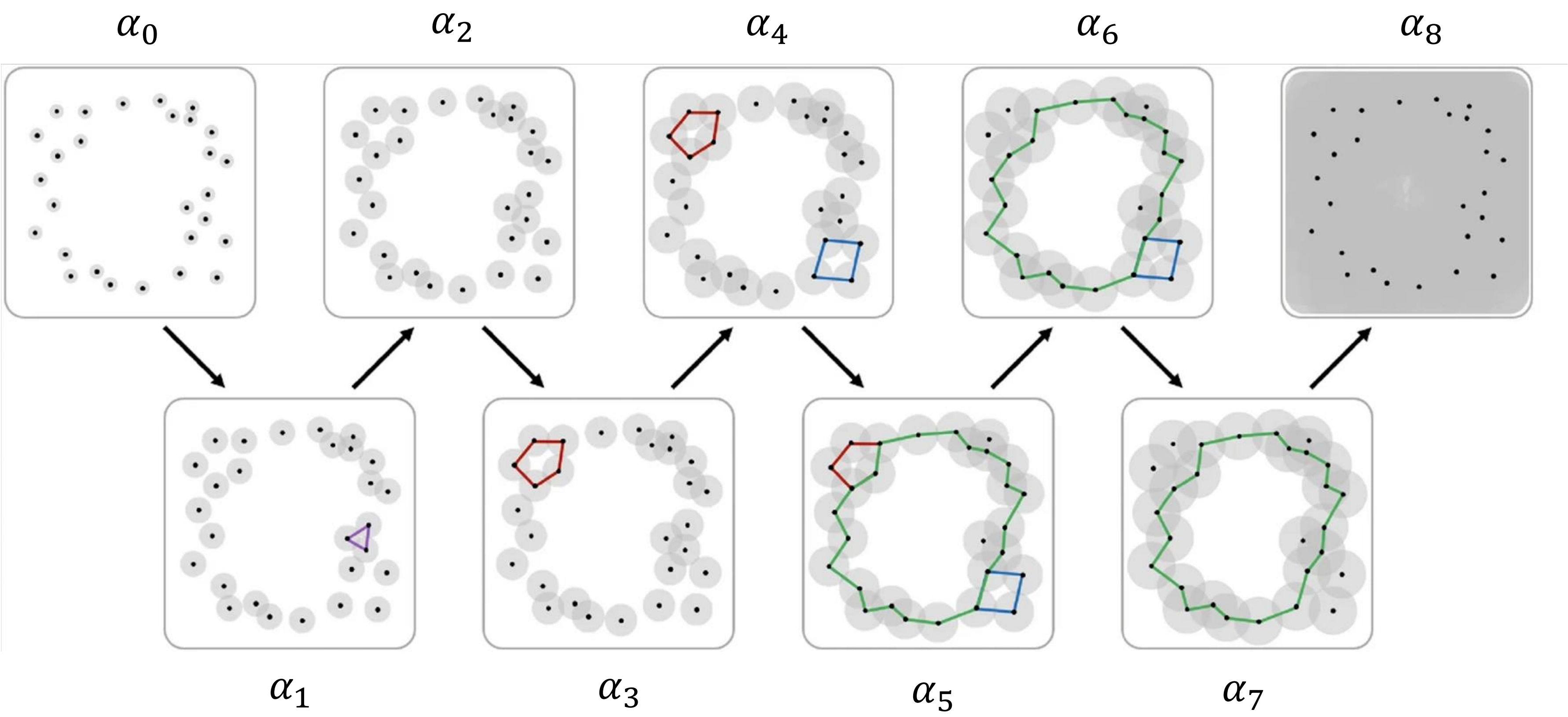


Image: Bobrowski, Skraba. A universal null-distribution for topological data analysis

Meaning of "**length**" of a PD interval



- $(\alpha_5, \alpha_8) \Rightarrow$ green cycle
- $(\alpha_4, \alpha_7) \Rightarrow$ blue cycle
- $(\alpha_3, \alpha_6) \Rightarrow$ red cycle
- $(\alpha_1, \alpha_2) \Rightarrow$ purple cycle

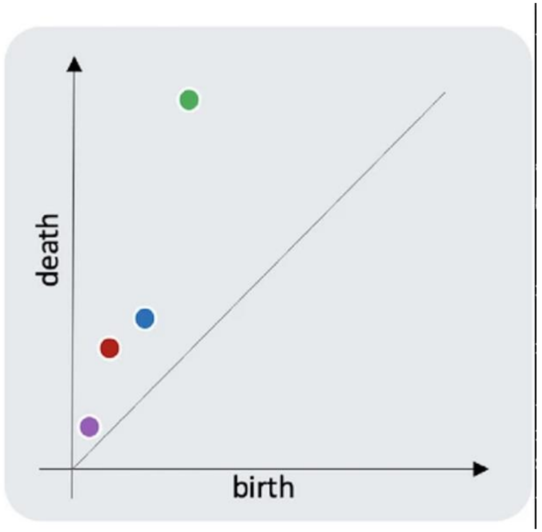
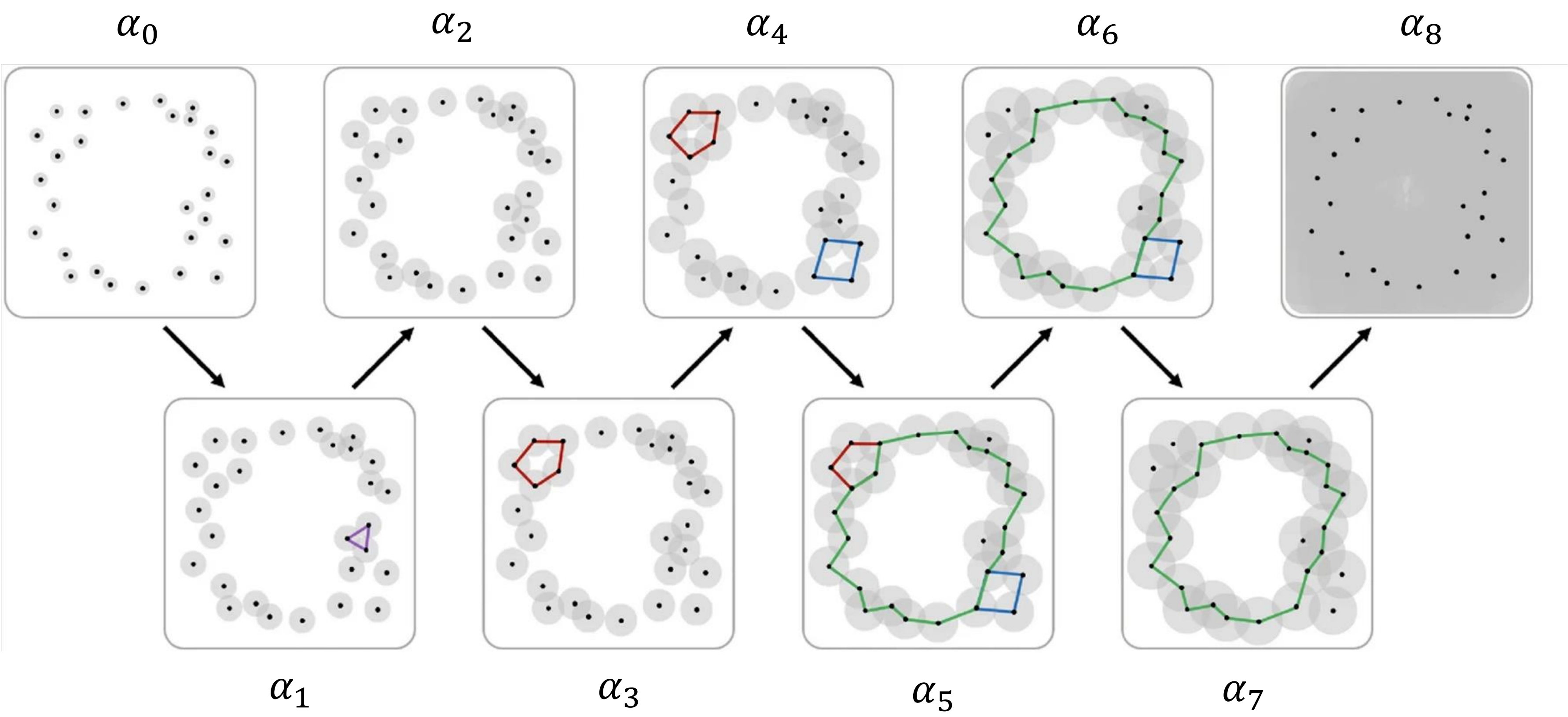


Image: Bobrowski, Skraba. A universal null-distribution for topological data analysis

Meaning of "**length**" of a PD interval

- Define length of a PD interval (point), (b, d), as $d - b$



- $(\alpha_5, \alpha_8) \Rightarrow$ green cycle
- $(\alpha_4, \alpha_7) \Rightarrow$ blue cycle
- $(\alpha_3, \alpha_6) \Rightarrow$ red cycle
- $(\alpha_1, \alpha_2) \Rightarrow$ purple cycle

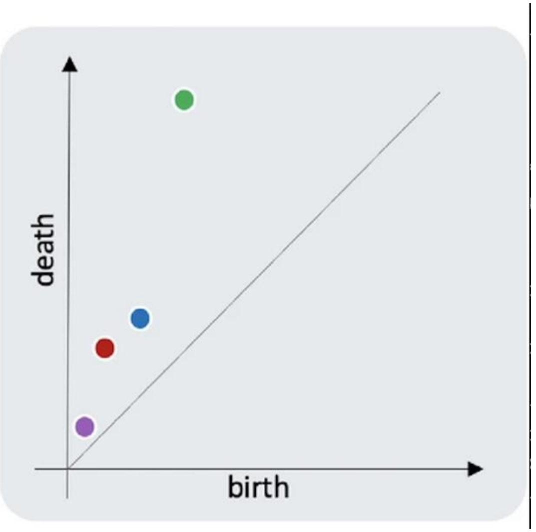
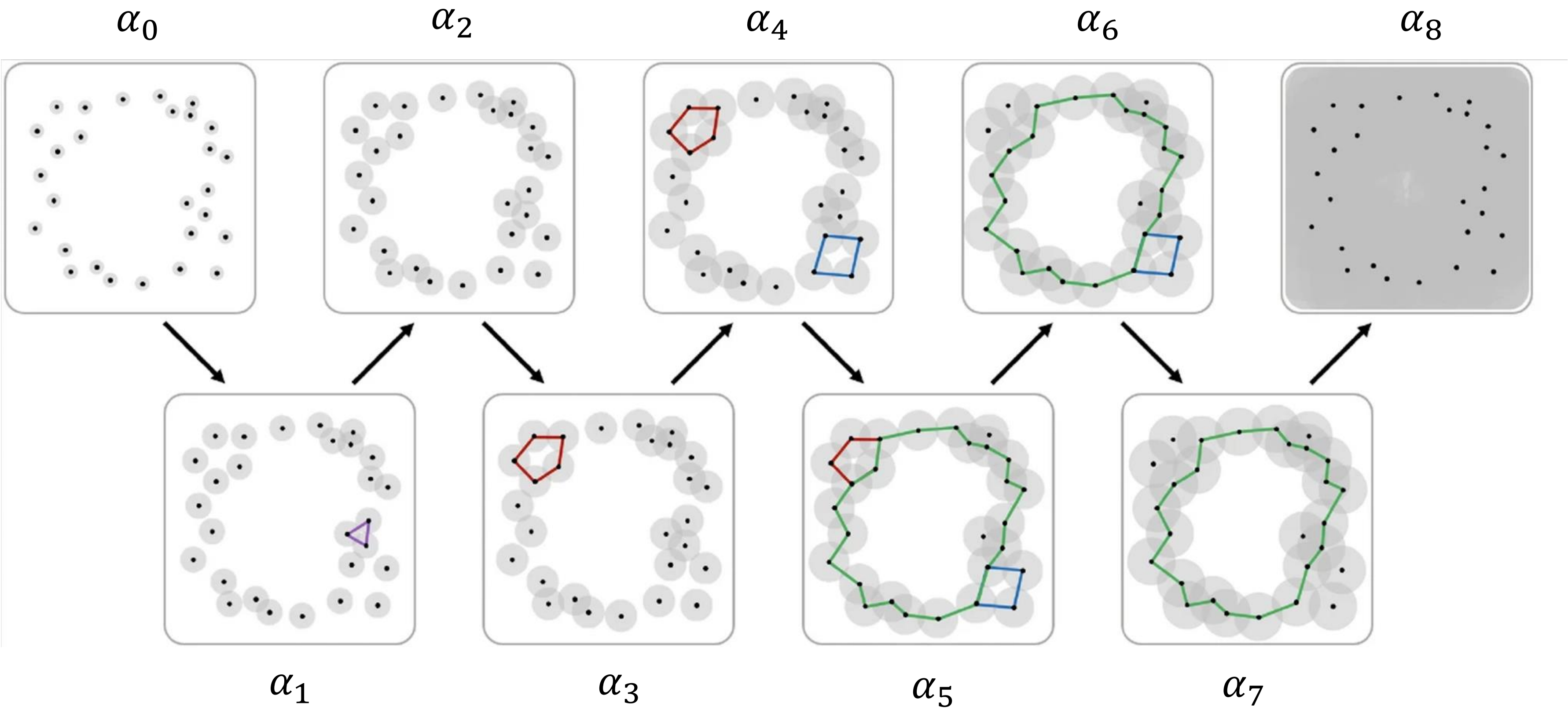


Image: Bobrowski, Skraba. A universal null-distribution for topological data analysis

Meaning of "**length**" of a PD interval

- Define length of a PD interval (point), (b, d), as $d - b$
- Observe the longer an interval is, the more significant its homology hole is



- $(\alpha_5, \alpha_8) \Rightarrow$ green cycle
- $(\alpha_4, \alpha_7) \Rightarrow$ blue cycle
- $(\alpha_3, \alpha_6) \Rightarrow$ red cycle
- $(\alpha_1, \alpha_2) \Rightarrow$ purple cycle

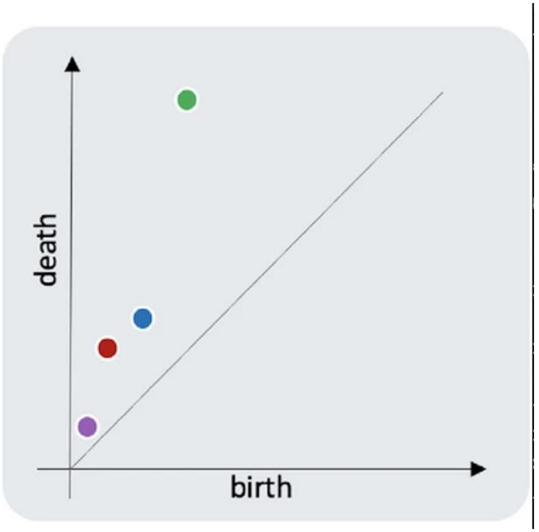
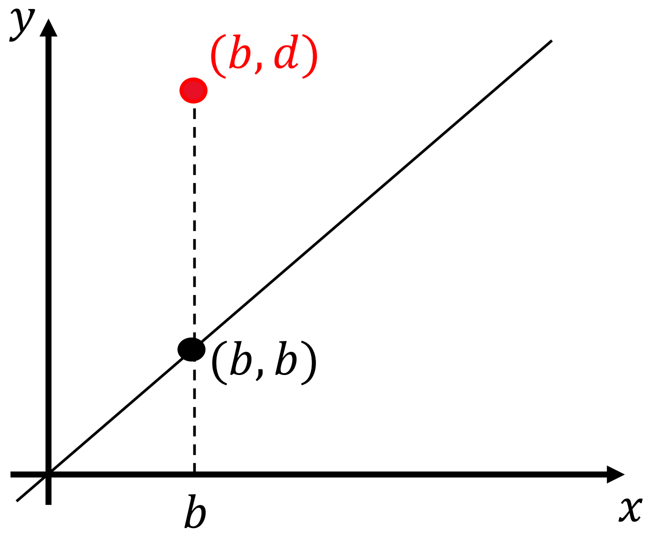


Image: Bobrowski, Skraba. A universal null-distribution for topological data analysis

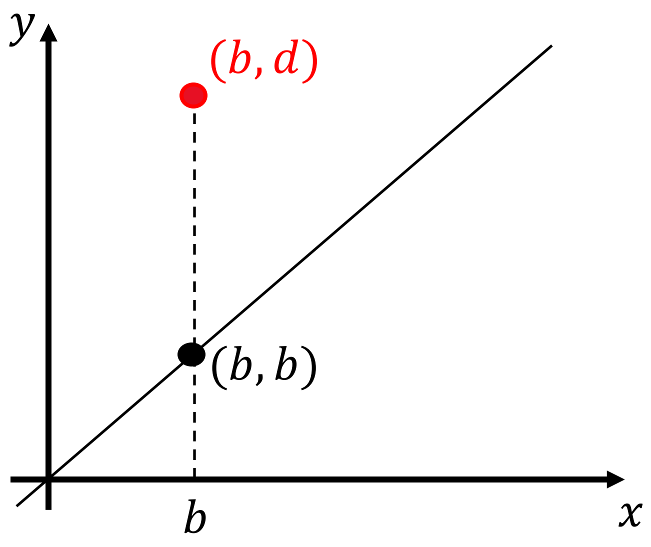
Length of an interval manifested on PD

- We observe that the length $d - b$ of a PD point (b, d) equals $1/\sqrt{2}$ times the distance of (b, d) to the diagonal



Length of an interval manifested on PD

- We observe that the length $d - b$ of a PD point (b, d) equals $1/\sqrt{2}$ times the distance of (b, d) to the diagonal
- This means that **the distance of a point in PD to the diagonal indicates the length of the corresponding interval**, and hence **the significance of the homological feature**



“Reliability” of PD

- Now, we are going to address an important problem in topology inference using PD: Arguing that PD is a “**reliable**” topological signature / indicator for the data

“Reliability” of PD

- Now, we are going to address an important problem in topology inference using PD: Arguing that PD is a “**reliable**” topological signature / indicator for the data
- For this, we argue that the PD you get is not some arbitrary set of points produced: their change w.r.t the data is “reasonable”

“Reliability” of PD

- Now, we are going to address an important problem in topology inference using PD: Arguing that PD is a “**reliable**” topological signature / indicator for the data
- For this, we argue that the PD you get is not some arbitrary set of points produced: their change w.r.t the data is “reasonable”
  - Aka. if the data changes a little, the PD also changes a little
  - If the two datasets correspond, then their PD’s also correspond

“Reliability” of PD

- Now, we are going to address an important problem in topology inference using PD: Arguing that PD is a “**reliable**” topological signature / indicator for the data
- For this, we argue that the PD you get is not some arbitrary set of points produced: their change w.r.t the data is “reasonable”
  - Aka. if the data changes a little, the PD also changes a little
  - If the two datasets correspond, then their PD’s also correspond
- The result that we are going to build is formally called the “**stability**” of persistence diagram, and is a corner stone of persistent homology and its applications

“Reliability” of PD

- Now, we are going to address an important problem in topology inference using PD: Arguing that PD is a “**reliable**” topological signature / indicator for the data
- For this, we argue that the PD you get is not some arbitrary set of points produced: their change w.r.t the data is “reasonable”
  - Aka. if the data changes a little, the PD also changes a little
  - If the two datasets correspond, then their PD’s also correspond
- The result that we are going to build is formally called the “**stability**” of persistence diagram, and is a corner stone of persistent homology and its applications
- For this, we need a way to measure:
  - Difference between the data
  - Difference between two PDs

“Reliability” of PD

- Now, we are going to address an important problem in topology inference using PD: Arguing that PD is a “**reliable**” topological signature / indicator for the data
- For this, we argue that the PD you get is not some arbitrary set of points produced: their change w.r.t the data is “reasonable”
  - Aka. if the data changes a little, the PD also changes a little
  - If the two datasets correspond, then their PD’s also correspond
- The result that we are going to build is formally called the “**stability**” of persistence diagram, and is a corner stone of persistent homology and its applications
- For this, we need a way to measure:
  - Difference between the data
  - Difference between two PDs
- Mathematically, the tool to measure the difference of two objects is called a **distance** function (or **metric**)

Distance function (as on Wikipedia)

Definition [edit]

Formally, a **metric space** is an ordered pair (M, d) where M is a set and d is a **metric** on M, i.e., a function

$$d : M \times M \to \mathbb{R}$$

satisfying the following axioms for all points $x, y, z \in M$:[4][5]

1. The distance from a point to itself is zero:

   $$d(x, x) = 0$$

2. (Positivity) The distance between two distinct points is always positive:

   $$\text{If } x \neq y \text{, then } d(x, y) > 0$$

3. (Symmetry) The distance from x to y is always the same as the distance from y to x:

   $$d(x, y) = d(y, x)$$

4. The triangle inequality holds:

   $$d(x, z) \leq d(x, y) + d(y, z)$$

   This is a natural property of both physical and metaphorical notions of distance: you can arrive at z from x by taking a detour through y, but this will not make your journey any shorter than the direct path.

If the metric d is unambiguous, one often refers by abuse of notation to "the metric space M".

Distance function

Examples:

- For two numbers x and y on the real line, $d(x, y) = |x - y|$

Distance function

Examples:

- For two numbers x and y on the real line, $d(x, y) = |x - y|$
- For two points from higher-dimensional space, say, two points

$$\boldsymbol{x} = (x_1, x_2) \text{ and } \boldsymbol{y} = (y_1, y_2)$$

from $\mathbb{R}^2$, we have:

Distance function

Examples:

- For two numbers x and y on the real line, $d(x, y) = |x - y|$
- For two points from higher-dimensional space, say, two points

$$\boldsymbol{x} = (x_1, x_2) \text{ and } \boldsymbol{y} = (y_1, y_2)$$

from $\mathbb{R}^2$, we have:

  - L_1-distance (Manhattan distance): $d_1(\boldsymbol{x}, \boldsymbol{y}) = |x_1 - y_1| + |x_2 - y_2|$

Distance function

Examples:

- For two numbers x and y on the real line, $d(x, y) = |x - y|$
- For two points from higher-dimensional space, say, two points

$$\boldsymbol{x} = (x_1, x_2) \text{ and } \boldsymbol{y} = (y_1, y_2)$$

from $\mathbb{R}^2$, we have:

  - L_1-distance (Manhattan distance): $d_1(\boldsymbol{x}, \boldsymbol{y}) = |x_1 - y_1| + |x_2 - y_2|$
  - L_2-distance (Euclidean distance): $d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$

Distance function

Examples:

- For two numbers x and y on the real line, $d(x, y) = |x - y|$
- For two points from higher-dimensional space, say, two points

$$\boldsymbol{x} = (x_1, x_2) \text{ and } \boldsymbol{y} = (y_1, y_2)$$

from $\mathbb{R}^2$, we have:

  - L_1-distance (Manhattan distance): $d_1(\boldsymbol{x}, \boldsymbol{y}) = |x_1 - y_1| + |x_2 - y_2|$
  - L_2-distance (Euclidean distance): $d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$
  - L_∞-distance: $d_\infty(\boldsymbol{x}, \boldsymbol{y}) = \max\{|x_1 - y_1|, |x_2 - y_2|\}$

Distance for functions

- Recall we need to measure:
  - Difference between the data
  - Difference between two PDs

Distance for functions

- Recall we need to measure:
  - <u>Difference between the data</u>
  - Difference between two PDs
- What is the data we try to measure here?

Distance for functions

- Recall we need to measure:
  - Difference between the data
  - Difference between two PDs
- What is the data we try to measure here?
- It turns out that "functions" are a quite universal type of data (reason will be made clear later)

Distance for functions

- Recall we need to measure:
  - Difference between the data
  - Difference between two PDs
- What is the data we try to measure here?
- It turns out that "functions" are a quite universal type of data (reason will be made clear later)
- Specifically, for measuring the difference of two functions, we assume the domain to be the same, aka. we measure two functions of the following form:
  - $f: X \to \mathbb{R}$
  - $g: X \to \mathbb{R}$

Distance for functions

- The idea of our distance $d(f, g)$ for the two functions f, g is to measure the maximum of difference of the function values at each point in the domain X

Distance for functions

- The idea of our distance $d(f, g)$ for the two functions f, g is to measure the maximum of difference of the function values at each point in the domain X
- The distance is also denoted $\parallel f - g \parallel_\infty$:

$$\parallel f - g \parallel_\infty = \max_{x \in X}\{|f(x) - g(x)|\}$$

Distance for functions

- Suppose $X = \mathbb{R}$ for f and g.

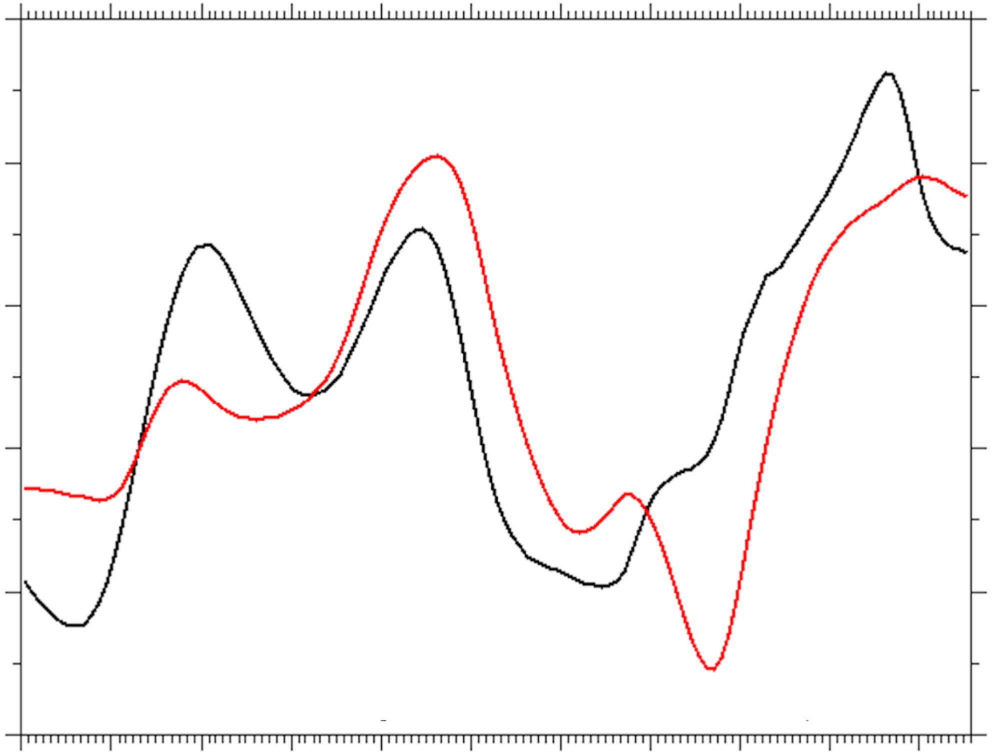


Image: https://faculty.washington.edu/kessler/salinity/165e-salinity-maximum.html

Distance for functions

- Suppose $X = \mathbb{R}$ for f and g.
- The maximum is achieved by the blue line

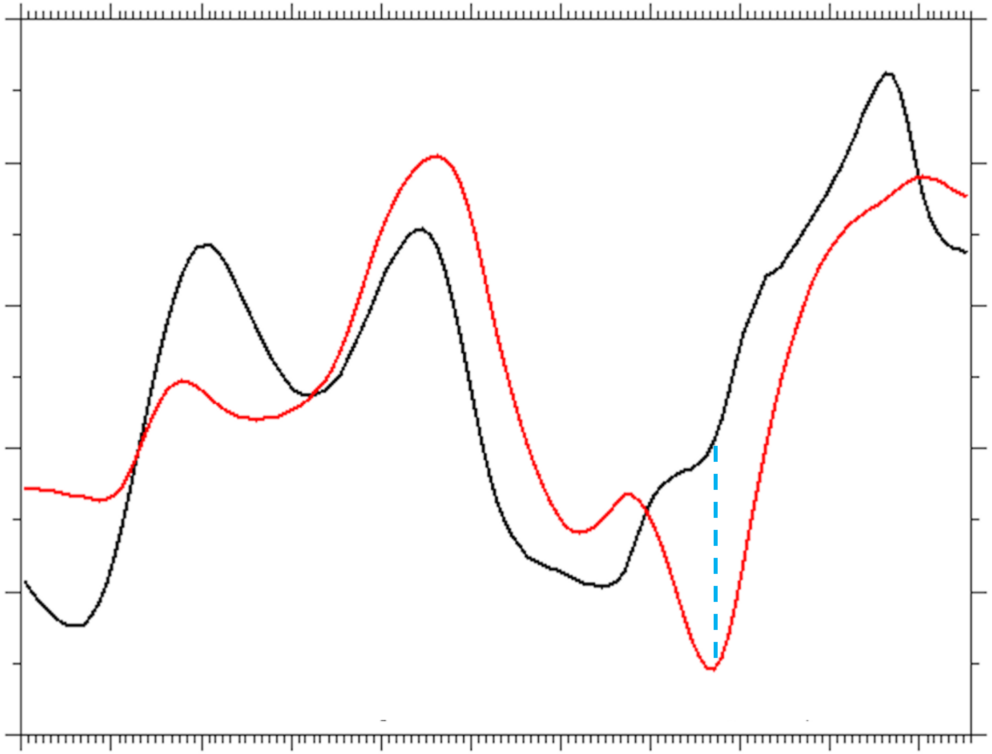


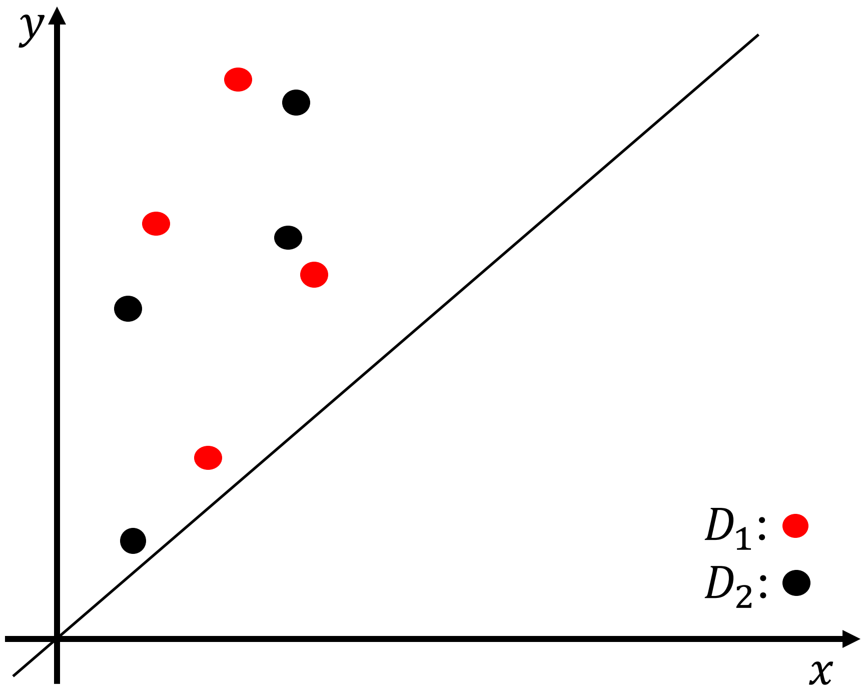
Image: https://faculty.washington.edu/kessler/salinity/165e-salinity-maximum.html

Distance for PDs

- Now we look at how to measure difference between two PDs, D_1 and D_2

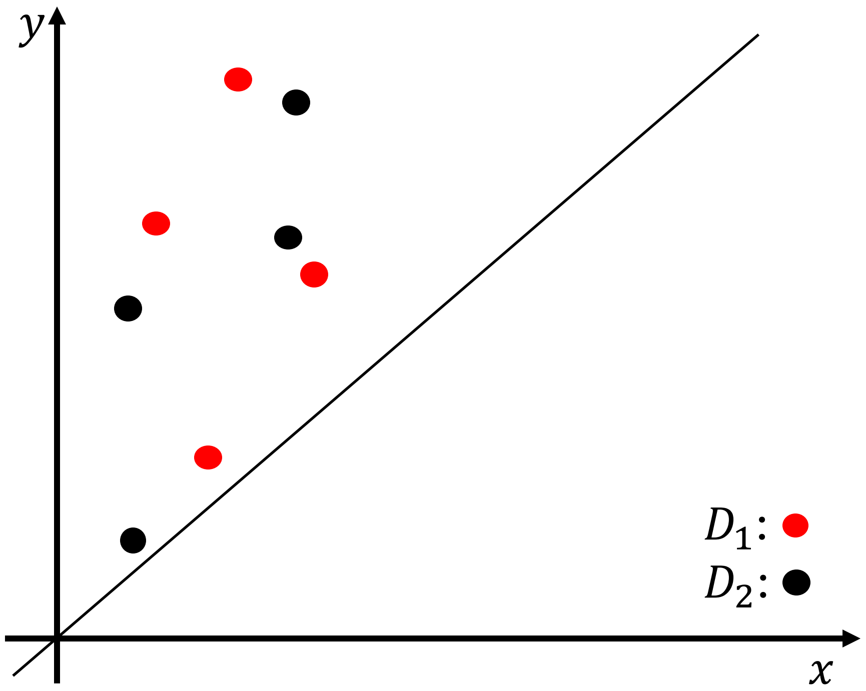
Distance for PDs

- Now we look at how to measure difference between two PDs, D_1 and D_2



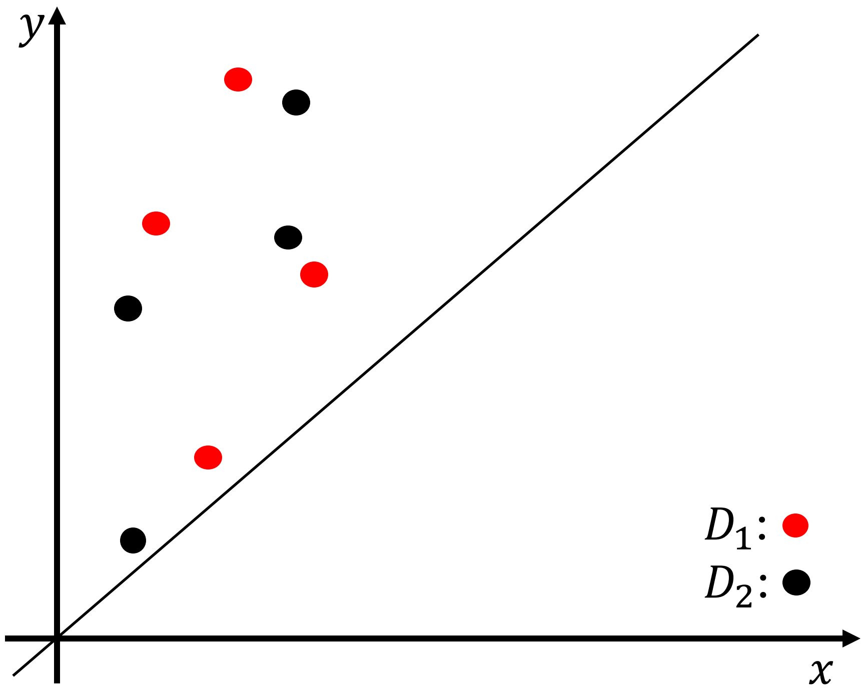
Distance for PDs

- Now we look at how to measure difference between two PDs, D_1 and D_2
- To measure the difference, we try to “match” points in D_1 and D_2 in a one-to-one manner



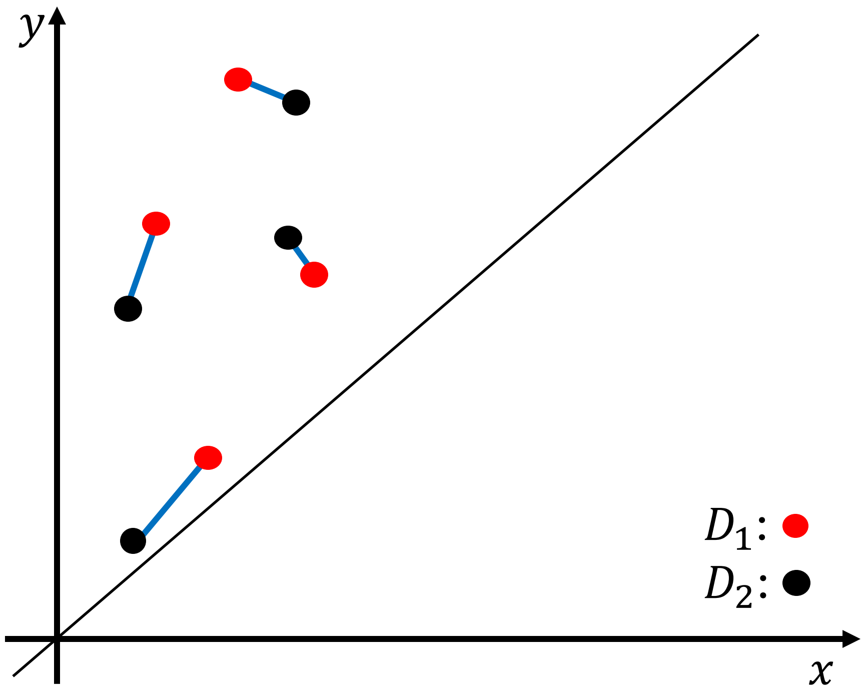
Distance for PDs

- Now we look at how to measure difference between two PDs, D_1 and D_2
- To measure the difference, we try to "match" points in D_1 and D_2 in a one-to-one manner
- Furthermore, we want the matched points in the two PDs to be as close as possible, which indicates that overall the two PDs are "close" to each other



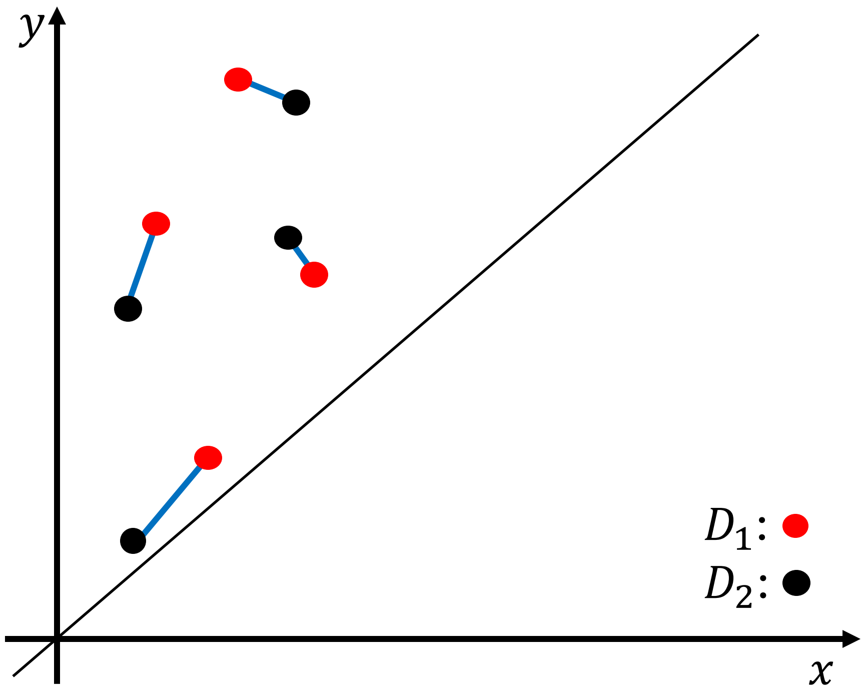
Distance for PDs

- Now we look at how to measure difference between two PDs, D_1 and D_2
- To measure the difference, we try to "match" points in D_1 and D_2 in a one-to-one manner
- Furthermore, we want the matched points in the two PDs to be as close as possible, which indicates that overall the two PDs are "close" to each other



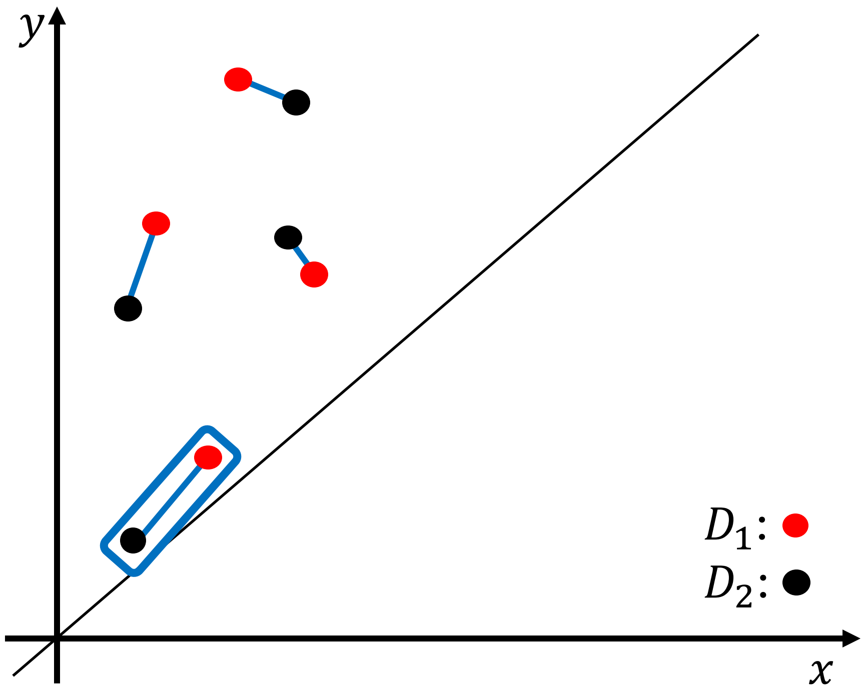
Distance for PDs

- Under the matching, the distance of the farthest two points (aka. "maximum" distances among the matched pairs) provides the distance for the overall two PDs



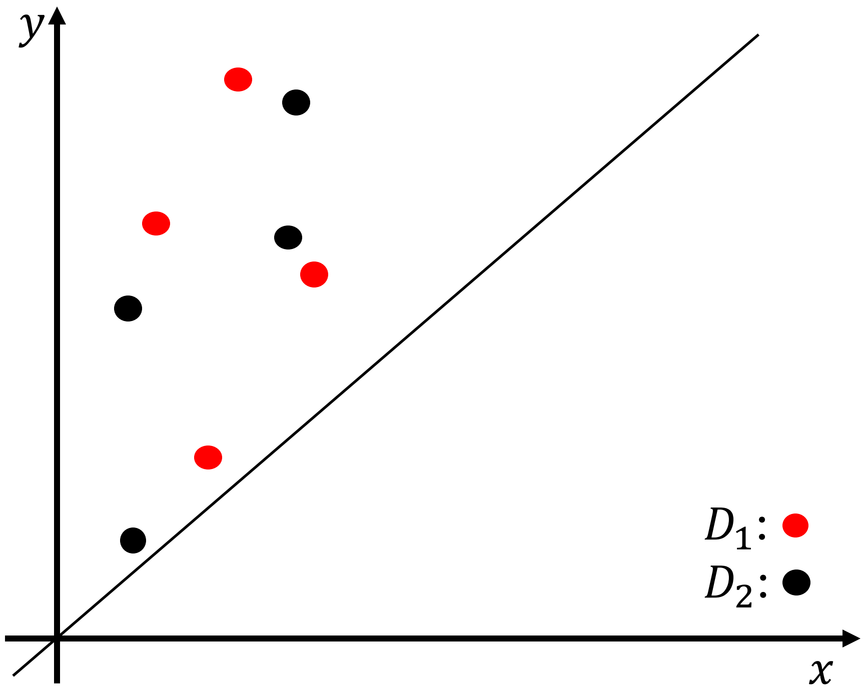
Distance for PDs

- Under the matching, the distance of the farthest two points (aka. "maximum" distances among the matched pairs) provides the distance for the overall two PDs



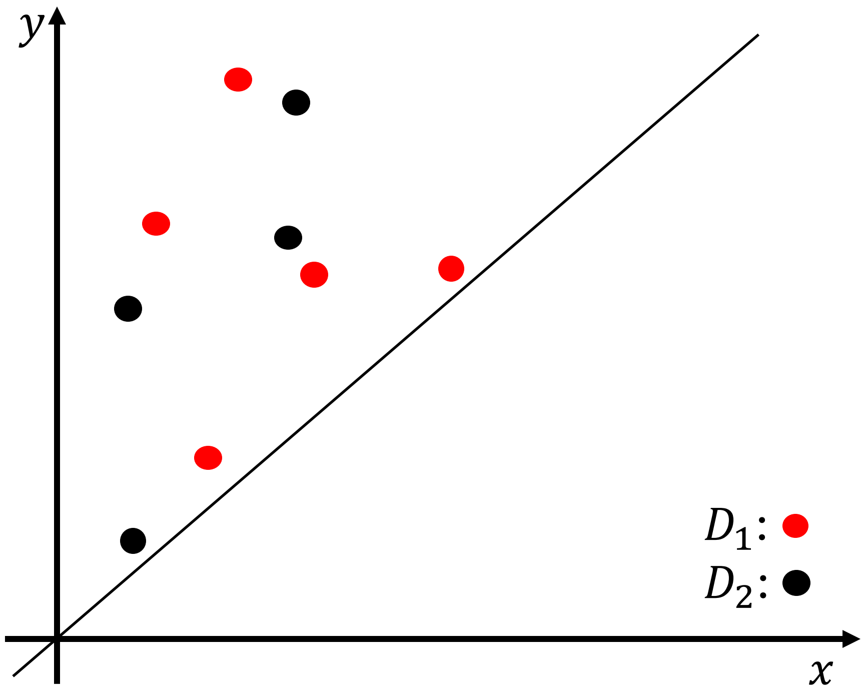
Distance for PDs

- However, D_1 and D_2 may not have the same number of points, making the "perfect matching" impossible (aka. suppose we add another point in D_1)



Distance for PDs

- However, D_1 and D_2 may not have the same number of points, making the "perfect matching" impossible (aka. suppose we add another point in D_1)



Distance for PDs

- We then introduce a "**partial matching**":
  1. Select the same number of points from D_1 and D_2: let these points "perfectly" match to each other

Distance for PDs

- We then introduce a "**partial matching**":
  1. Select the same number of points from D_1 and D_2: let these points "perfectly" match to each other
     - And of course, we want each matched points to be close to each other

Distance for PDs

- We then introduce a "**partial matching**":
  1. Select the same number of points from D_1 and D_2: let these points "perfectly" match to each other
     - And of course, we want each matched points to be close to each other
  2. For the remaining points, we let them be "unmatched", meaning we want to "ignore" them

Distance for PDs

- We then introduce a "**partial matching**":
  1. Select the same number of points from D_1 and D_2: let these points "perfectly" match to each other
     - And of course, we want each matched points to be close to each other
  2. For the remaining points, we let them be "unmatched", meaning we want to "ignore" them
     - However, we can only "ignore" then when they are not very "important", aka. being close to the diagonal

Distance for PDs

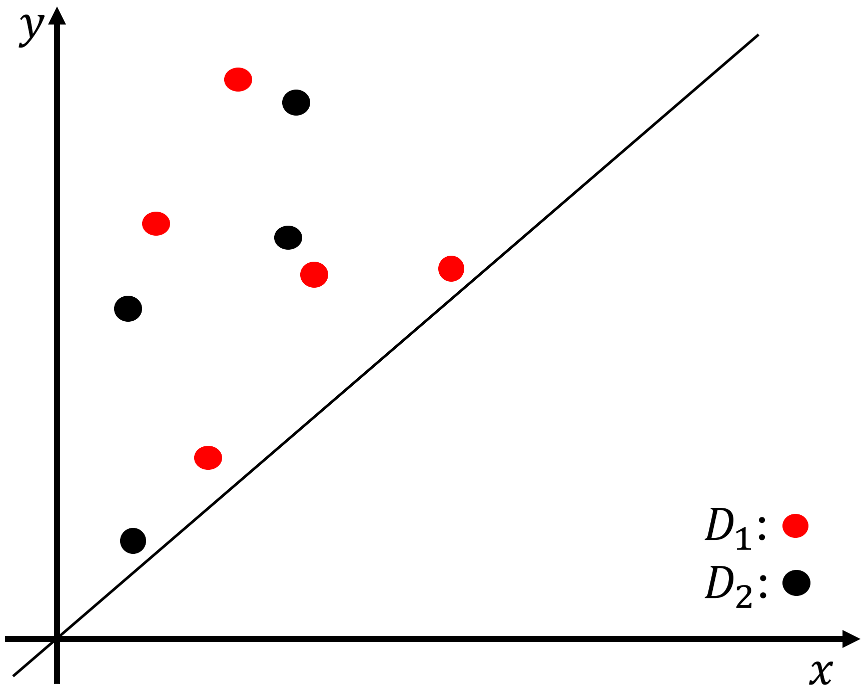
- The distance defined by such a “partial matching” is called the **bottleneck distance**, denoted $d_{\mathrm{B}}(D_1, D_2)$.

Distance for PDs

- The distance defined by such a "partial matching" is called the **bottleneck distance**, denoted $d_{\mathrm{B}}(D_1, D_2)$.
- $d_{\mathrm{B}}(D_1, D_2)$ being small means that we could find a close matching between those significant points in D_1, D_2 while those remaining points are "insignificant"

Distance for PDs

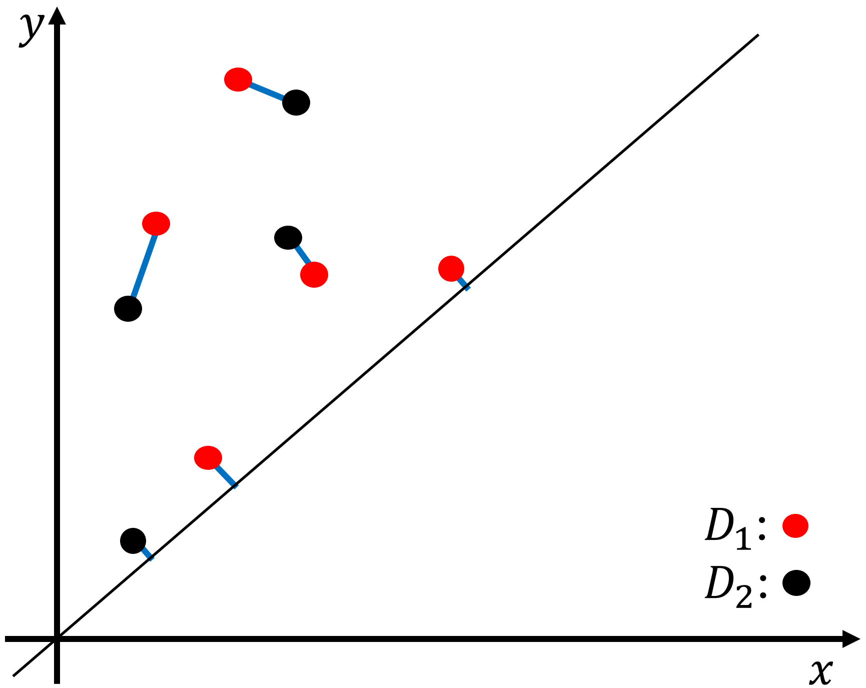
- The distance defined by such a "partial matching" is called the **bottleneck distance**, denoted $d_{\mathrm{B}}(D_1, D_2)$.
- $d_{\mathrm{B}}(D_1, D_2)$ being small means that we could find a close matching between those significant points in D_1, D_2 while those remaining points are "insignificant"



Distance for PDs

- The distance defined by such a "partial matching" is called the **bottleneck distance**, denoted $d_{\mathrm{B}}(D_1, D_2)$.
- $d_{\mathrm{B}}(D_1, D_2)$ being small means that we could find a close matching between those significant points in D_1, D_2 while those remaining points are "insignificant"

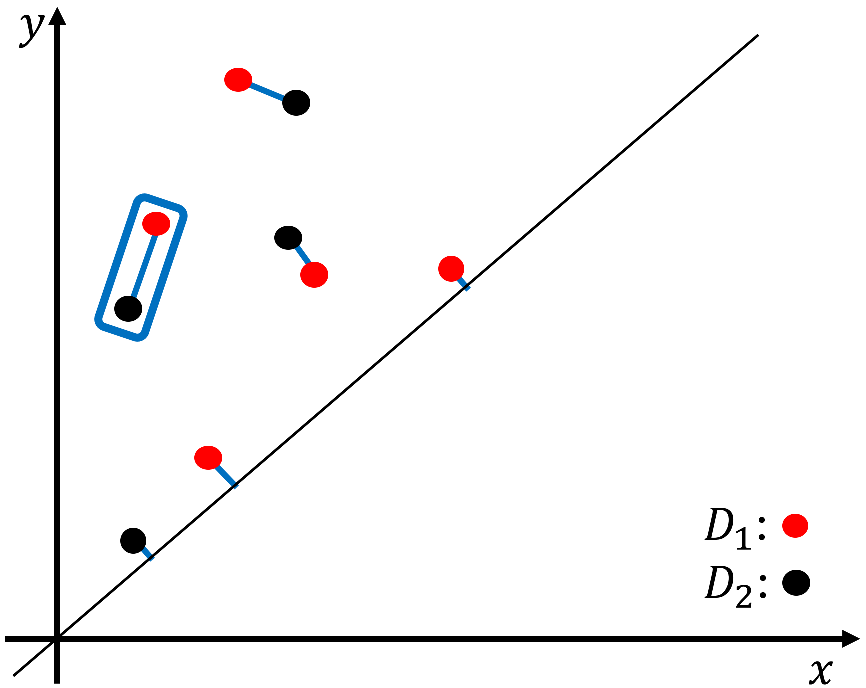
Blue lines: a "close" partial matching η



Distance for PDs

- The distance defined by such a "partial matching" is called the **bottleneck distance**, denoted $d_{\mathrm{B}}(D_1, D_2)$.
- $d_{\mathrm{B}}(D_1, D_2)$ being small means that we could find a close matching between those significant points in D_1, D_2 while those remaining points are "insignificant"

Blue lines: a "close" partial matching η



Bottleneck distance (more formally)

Let η be a partial matching between D_1 and D_2. We define $cost(\eta)$ as the maximum of the two below:

Bottleneck distance (more formally)

Let η be a partial matching between D_1 and D_2. We define $cost(\eta)$ as the maximum of the two below:

1. Maximum L_∞-distance of each two matched point p and q in D_1, D_2:

$$\max_{p,q \text{ matched in } \eta}\{d_\infty(p,q)\}$$

Bottleneck distance (more formally)

Let η be a partial matching between D_1 and D_2. We define $cost(\eta)$ as the maximum of the two below:

1. Maximum L_∞-distance of each two matched point p and q in D_1, D_2:

$$\max_{p,q \text{ matched in } \eta}\{d_\infty(p,q)\}$$

where $d_\infty(p,q) = \max\{|b_1 - d_1|, |b_2 - d_2|\}$ for $p = (b_1, d_1), q = (b_2, d_2)$

Bottleneck distance (more formally)

Let η be a partial matching between D_1 and D_2. We define $cost(\eta)$ as the maximum of the two below:

1. Maximum L_∞-distance of each two matched point p and q in D_1, D_2:

$$\max_{p,q \text{ matched in } \eta}\{d_\infty(p,q)\}$$

where $d_\infty(p,q) = \max\{|b_1 - d_1|, |b_2 - d_2|\}$ for $p = (b_1, d_1), q = (b_2, d_2)$

2. Maximum length $d - b$ (aka. distance to diagonal) of each unmatched point (b, d)

Bottleneck distance (more formally)

Let η be a partial matching between D_1 and D_2. We define $cost(\eta)$ as the maximum of the two below:

1. Maximum L_∞-distance of each two matched point p and q in D_1, D_2:

$$\max_{p,q \text{ matched in } \eta}\{d_\infty(p, q)\}$$

   where $d_\infty(p, q) = \max\{|b_1 - d_1|, |b_2 - d_2|\}$ for $p = (b_1, d_1), q = (b_2, d_2)$

2. Maximum length $d - b$ (aka. distance to diagonal) of each unmatched point (b, d)

The **bottleneck distance** is then defined as follows:

$$d_\text{B}(D_1, D_2) = \min_{\eta \text{ over all partial matchings}}\{cost(\eta)\}$$

aka. the minimum cost of all partial matchings

Bottleneck distance (more formally)

- The bottleneck distance between two PDs being small means that there is a “close” partial matching such that:

Bottleneck distance (more formally)

- The bottleneck distance between two PDs being small means that there is a “close” partial matching such that:
  - Each two matched points in the two PDs are close to each other (their L_∞-distance is small)

Bottleneck distance (more formally)

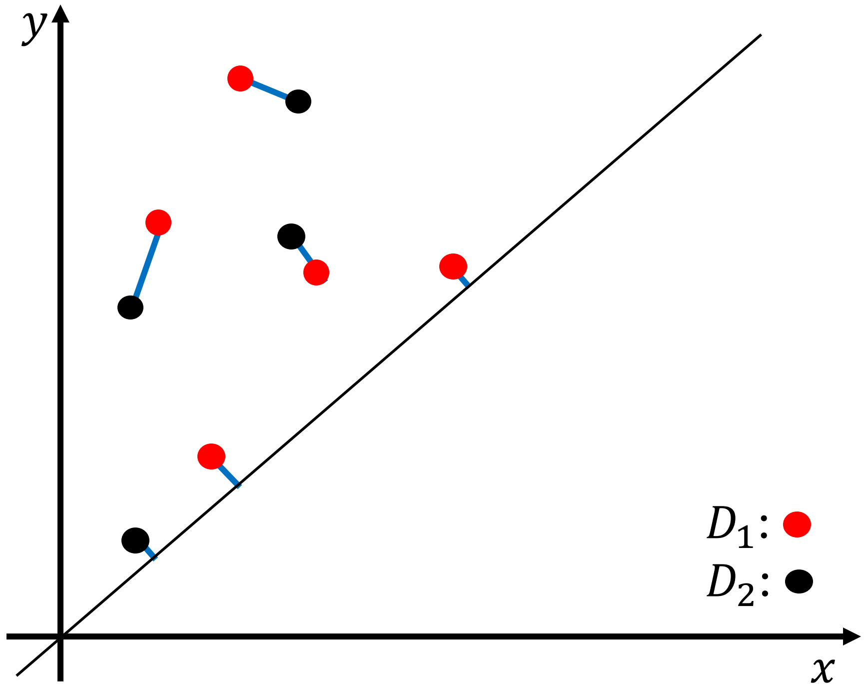
- The bottleneck distance between two PDs being small means that there is a “close” partial matching such that:
  - Each two matched points in the two PDs are close to each other (their L_∞-distance is small)
  - Each unmatched point is “insignificant” (length is small) so we can “ignore” them

Bottleneck distance (more formally)

- The bottleneck distance between two PDs being small means that there is a "close" partial matching such that:
  - Each two matched points in the two PDs are close to each other (their L_∞-distance is small)
  - Each unmatched point is "insignificant" (length is small) so we can "ignore" them
- So overall the two PDs are close

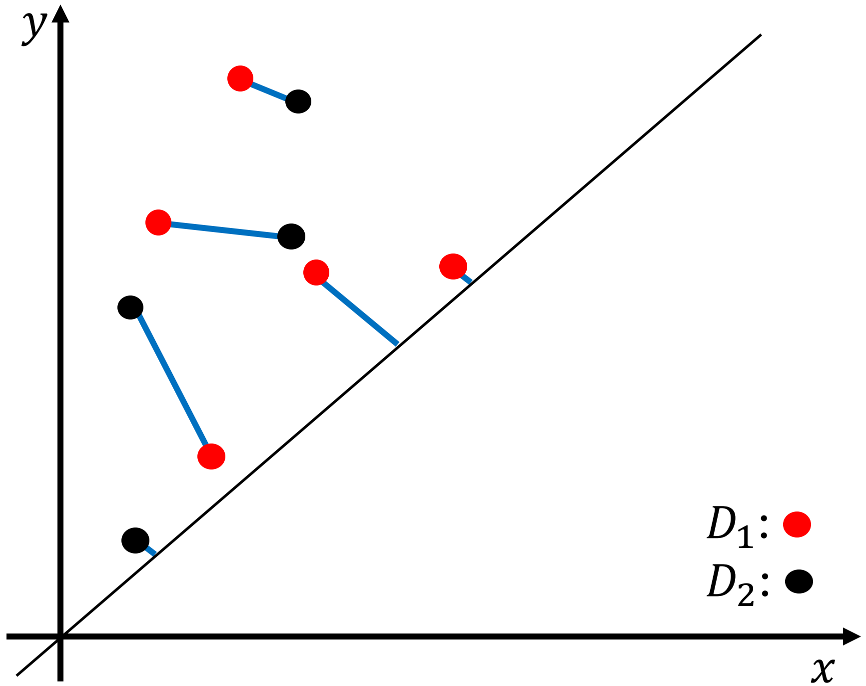
Distance for PDs

- The following is a "close" partial matching (with small cost) which achieves the bottleneck distance between D_1, D_2 (aka. $d_{\mathrm{B}}(D_1, D_2) = cost(\eta)$)



Distance for PDs

- The following is a partial matching where the max distance between matched points is high



Stability Theorem for Persistent Homology

- **Stability Theorem**: For any two functions $f, g: X \to \mathbb{R}$, one has

$$d_{\mathrm{B}}(PD(f), PD(g)) \leq \| f - g \|_{\infty}$$

Stability Theorem for Persistent Homology

- **Stability Theorem**: For any two functions $f, g: X \to \mathbb{R}$, one has

$$d_{\mathrm{B}}(PD(f), PD(g)) \leq \| f - g \|_{\infty}$$

- The $PD(f)$, $PD(g)$ above indicates the PD derived by taking the sublevelset filtration of the function

Stability Theorem for Persistent Homology

- **Stability Theorem**: For any two functions $f, g: X \to \mathbb{R}$, one has

$$d_{\mathrm{B}}(PD(f), PD(g)) \leq \| f - g \|_{\infty}$$

- The $PD(f)$, $PD(g)$ above indicates the PD derived by taking the sublevelset filtration of the function

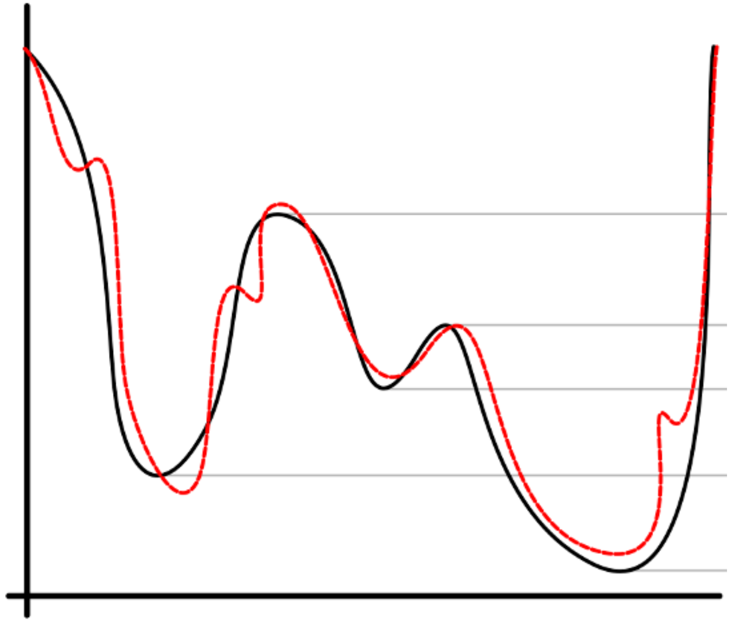


Figure from: Bendich et al. Topological and statistical behavior classifiers for tracking applications

Stability Theorem for Persistent Homology

- **Stability Theorem**: For any two functions $f, g: X \to \mathbb{R}$, one has

$$d_B(PD(f), PD(g)) \leq \| f - g \|_\infty$$

- The $PD(f)$, $PD(g)$ above indicates the PD derived by taking the sublevelset filtration of the function

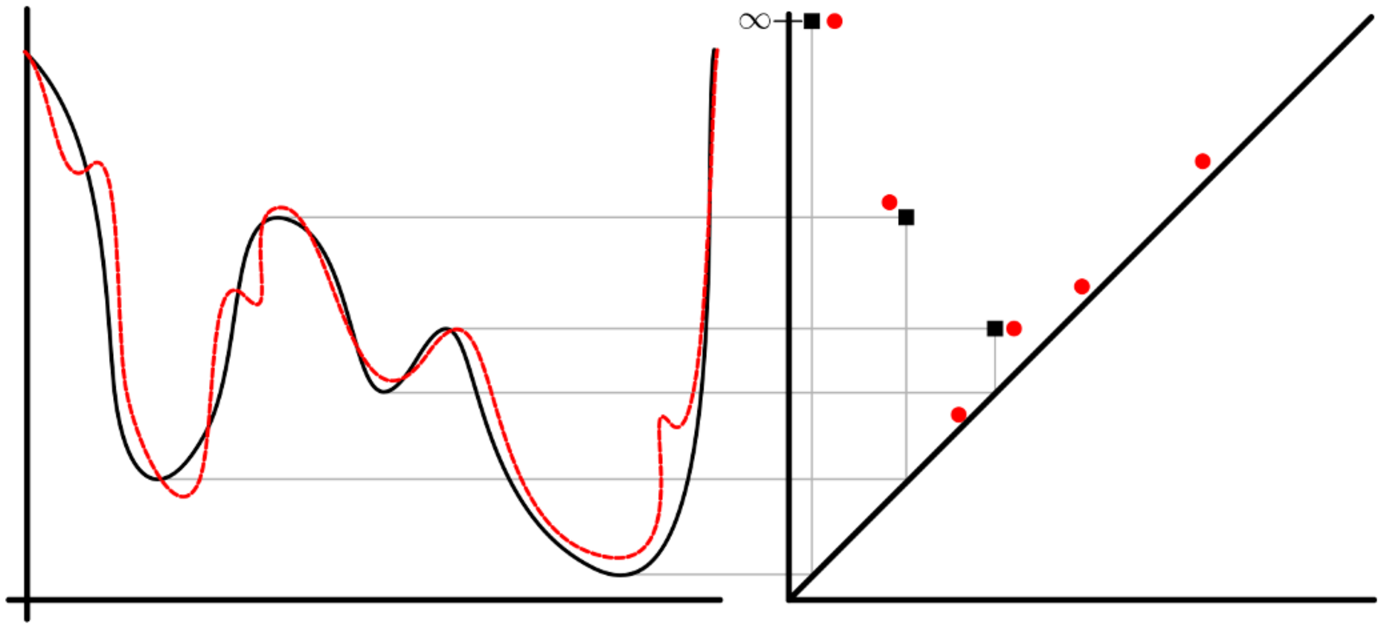


Figure from: Bendich et al. Topological and statistical behavior classifiers for tracking applications

Stability Theorem for Persistent Homology

- **Stability Theorem**: For any two functions $f, g: X \to \mathbb{R}$, one has

$$d_B(PD(f), PD(g)) \leq \| f - g \|_\infty$$

- The $PD(f)$, $PD(g)$ above indicates the PD derived by taking the sublevelset filtration of the function

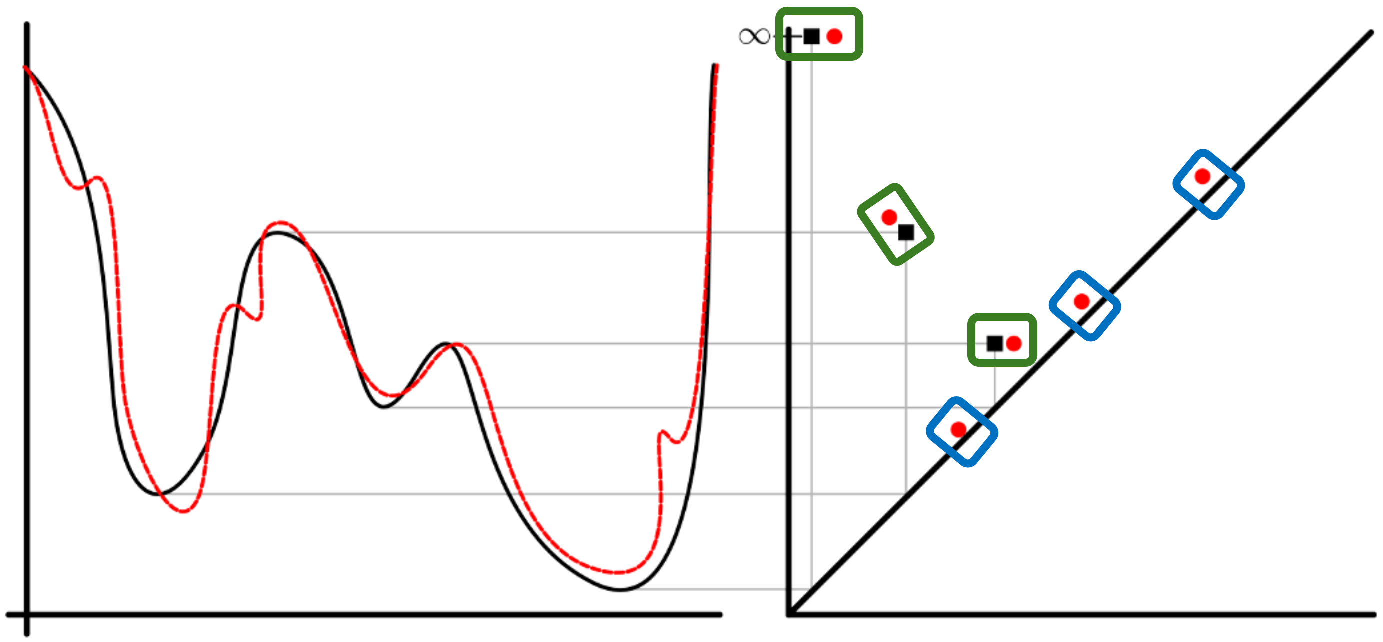


Figure from: Bendich et al. Topological and statistical behavior classifiers for tracking applications

Stability Theorem for Persistent Homology

- We also have that the local minimums and maximums that create and destroy the 0d gaps (holes) correspond in the two functions

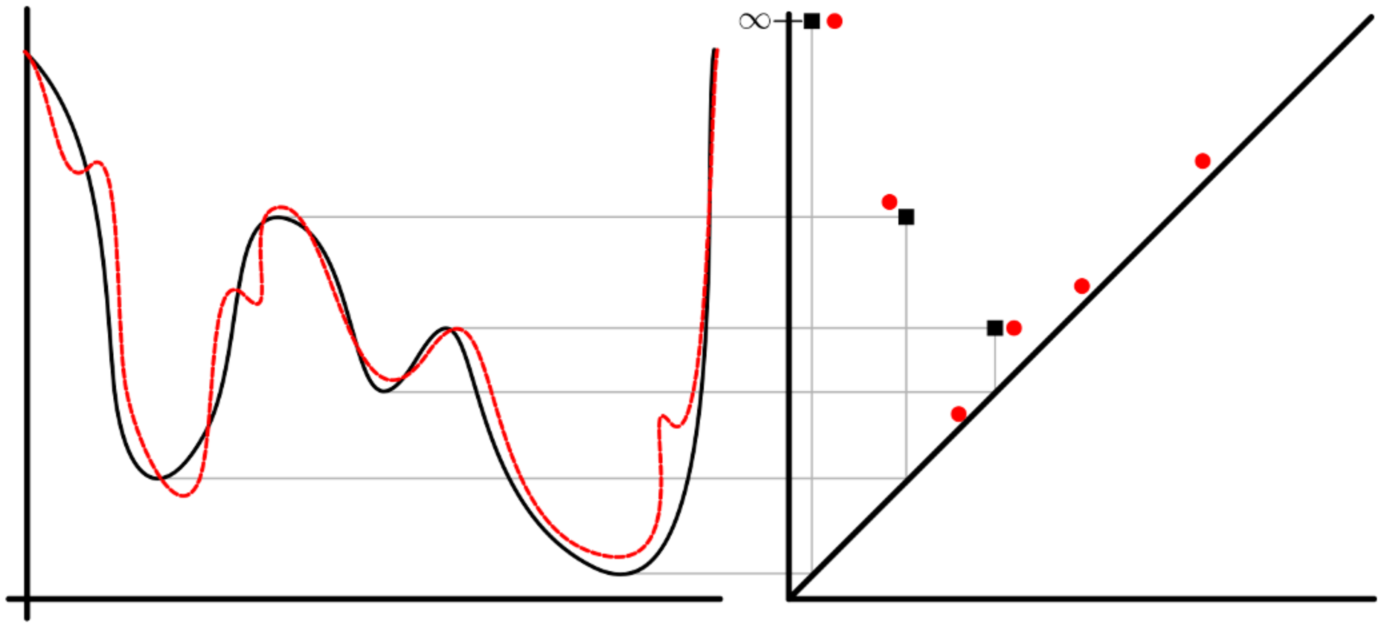


Figure from: Bendich et al. Topological and statistical behavior classifiers for tracking applications

Stability Theorem for Persistent Homology

- We also have that the local minimums and maximums that create and destroy the 0d gaps (holes) correspond in the two functions

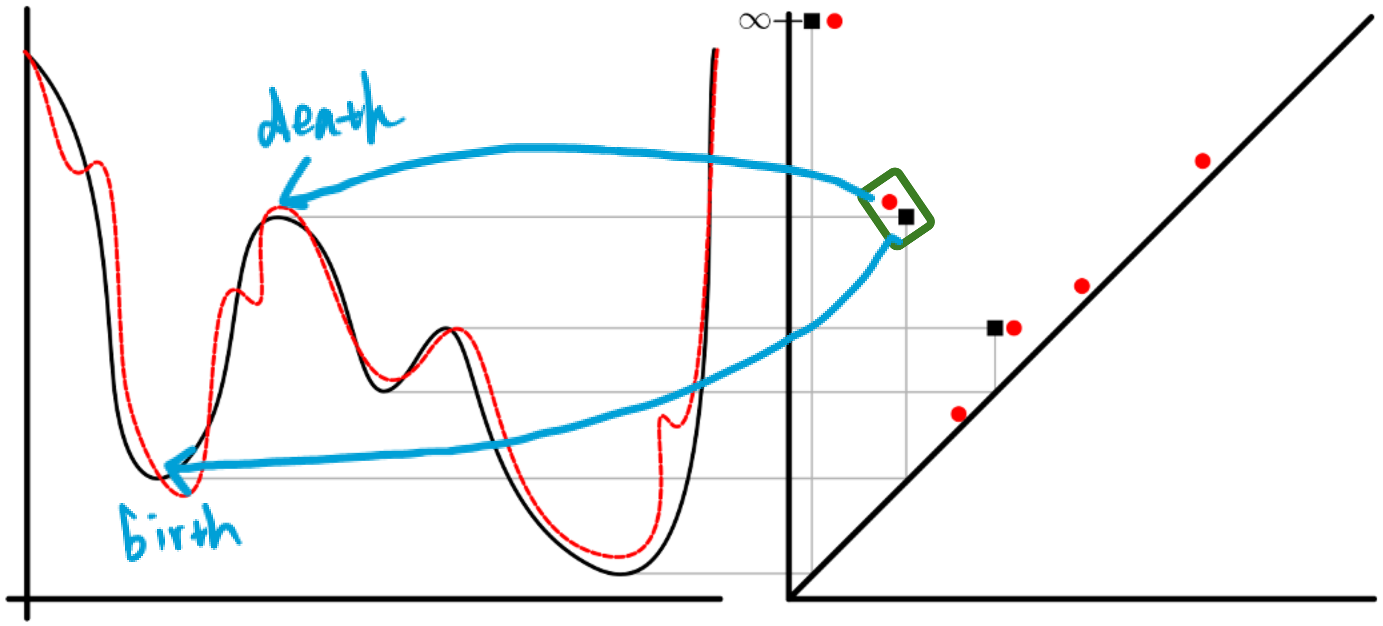


Figure from: Bendich et al. Topological and statistical behavior classifiers for tracking applications

Stability Theorem for Persistent Homology

- We also have that the local minimums and maximums that create and destroy the 0d gaps (holes) correspond in the two functions

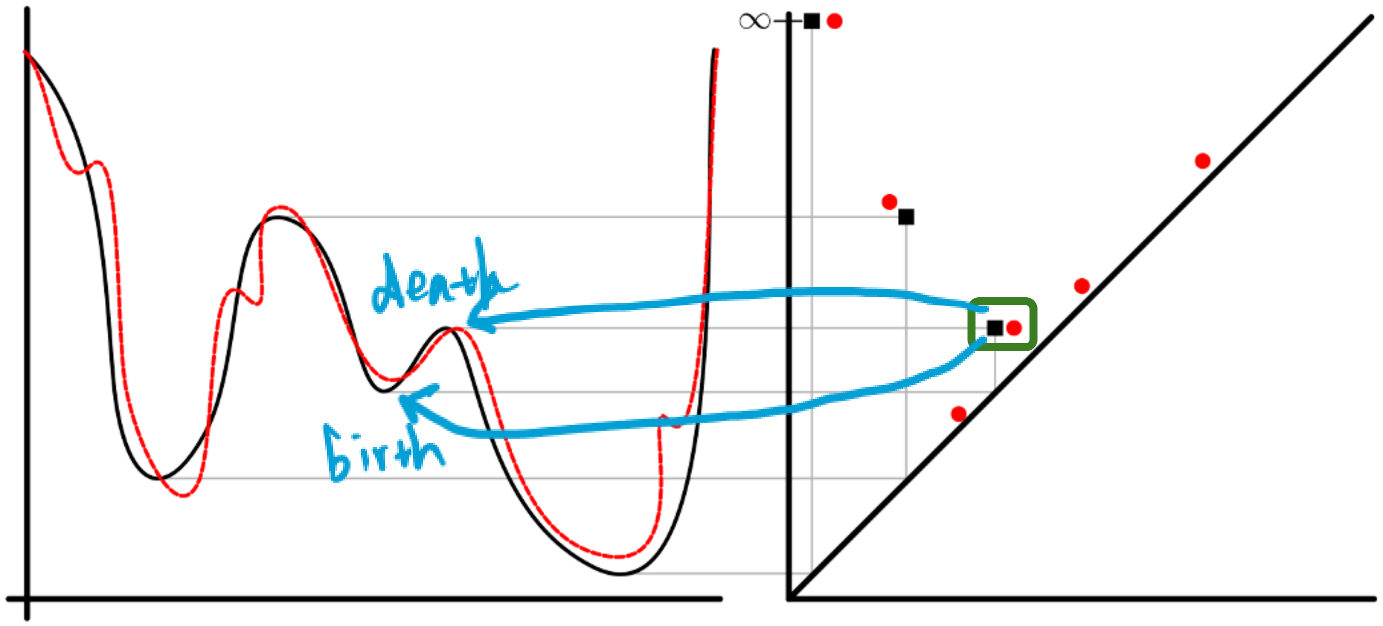


Figure from: Bendich et al. Topological and statistical behavior classifiers for tracking applications

Stability Theorem for Persistent Homology

- Furthermore, the three points in the PD of red curve are close to diagonal --- hence, they are "insignificant" and are typically **consider "noise" in TDA**

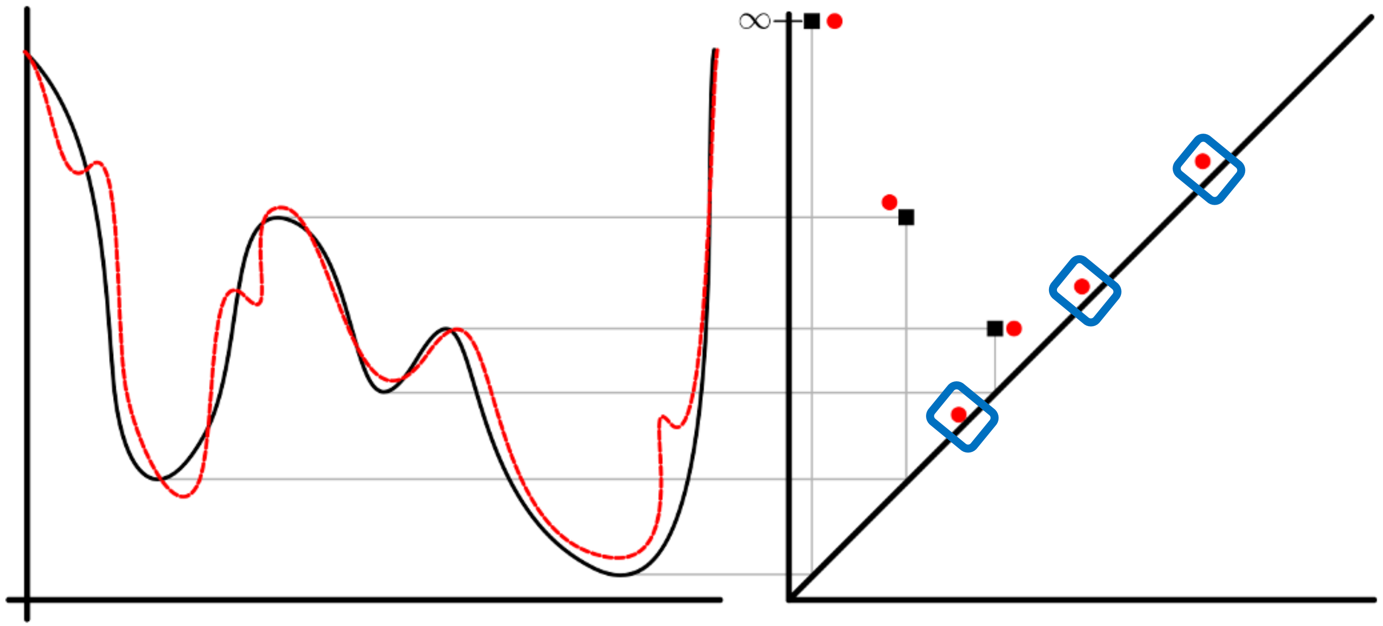


Figure from: Bendich et al. Topological and statistical behavior classifiers for tracking applications

Stability Theorem for Persistent Homology

- Furthermore, the three points in the PD of red curve are close to diagonal --- hence, they are “insignificant” and are typically **consider “noise” in TDA**
- This also corresponds to the fact that the red curve is more noisy than the black one

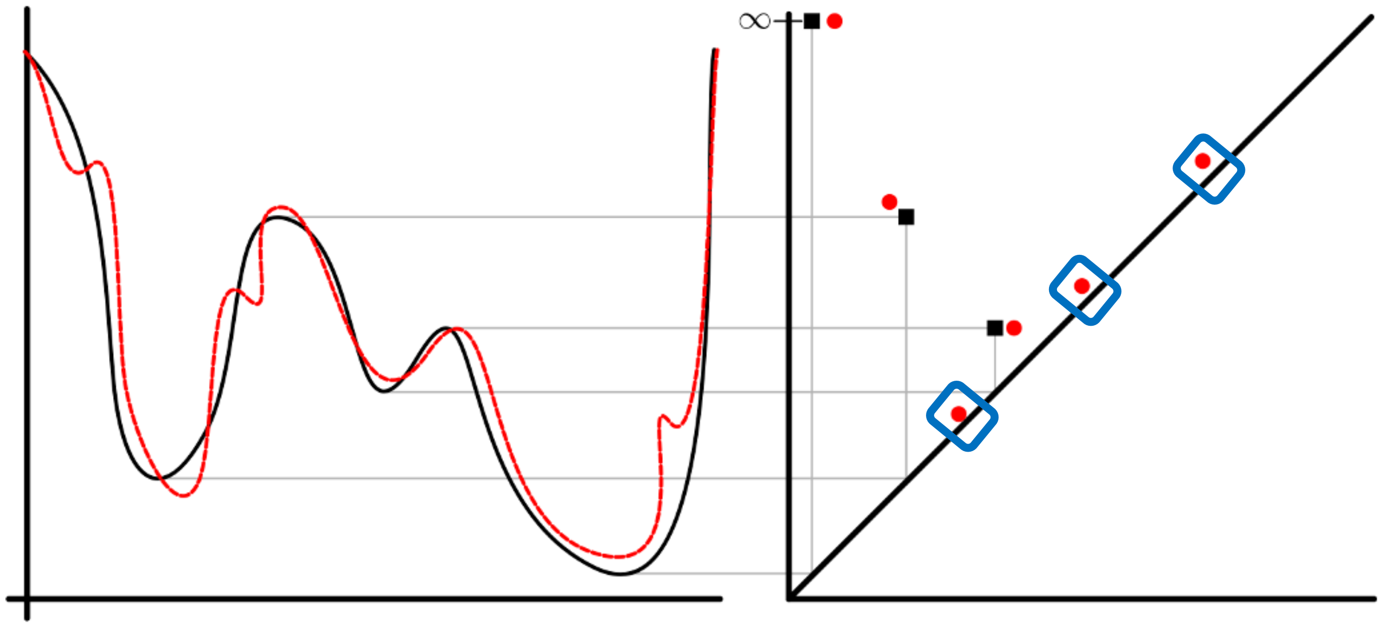


Figure from: Bendich et al. Topological and statistical behavior classifiers for tracking applications

Stability for point clouds

- So far, we have addressed the stability of PD for functions
- But there is another important type of data which is point cloud ---- a stability result for it would also be helpful
- We are going to frame the stability for point clouds in terms of the stability for functions

Stability for point clouds

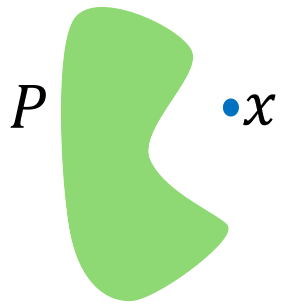
- So far, we have addressed the stability of PD for functions
- But there is another important type of data which is point cloud ---- a stability result for it would also be helpful
- We are going to frame the stability for point clouds in terms of the stability for functions
- First, given a point set $P \subseteq \mathbb{R}^d$ and an arbitrary point $x \in \mathbb{R}^d$, we define distance between x and P as the minimum distance of x with any point in P:

$$d(x, P) = \min_{p \in P}\{d(x, p)\}$$

Stability for point clouds

- So far, we have addressed the stability of PD for functions
- But there is another important type of data which is point cloud ---- a stability result for it would also be helpful
- We are going to frame the stability for point clouds in terms of the stability for functions
- First, given a point set $P \subseteq \mathbb{R}^d$ and an arbitrary point $x \in \mathbb{R}^d$, we define distance between x and P as the minimum distance of x with any point in P:

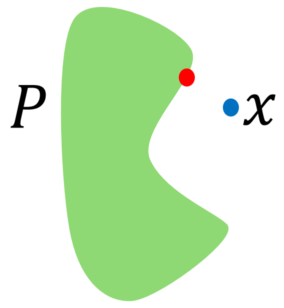
$$d(x, P) = \min_{p \in P}\{d(x, p)\}$$



Stability for point clouds

- So far, we have addressed the stability of PD for functions
- But there is another important type of data which is point cloud ---- a stability result for it would also be helpful
- We are going to frame the stability for point clouds in terms of the stability for functions
- First, given a point set $P \subseteq \mathbb{R}^d$ and an arbitrary point $x \in \mathbb{R}^d$, we define distance between x and P as the minimum distance of x with any point in P:

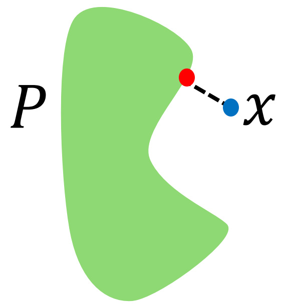
$$d(x, P) = \min_{p \in P}\{d(x, p)\}$$



Stability for point clouds

- So far, we have addressed the stability of PD for functions
- But there is another important type of data which is point cloud ---- a stability result for it would also be helpful
- We are going to frame the stability for point clouds in terms of the stability for functions
- First, given a point set $P \subseteq \mathbb{R}^d$ and an arbitrary point $x \in \mathbb{R}^d$, we define distance between x and P as the minimum distance of x with any point in P:

$$d(x, P) = \min_{p \in P}\{d(x, p)\}$$



Distance field

- Now, given a point set $P \subseteq \mathbb{R}^d$, we define a function $f_P: \mathbb{R}^d \to \mathbb{R}$ called the distance field of P as:

$$f_P(x) = d(x, P) \quad \forall x \in \mathbb{R}^d$$

Distance field

- Now, given a point set $P \subseteq \mathbb{R}^d$, we define a function $f_P \colon \mathbb{R}^d \to \mathbb{R}$ called the distance field of P as:
$$f_P(x) = d(x, P) \quad \forall x \in \mathbb{R}^d$$
- **Observation**: $f_P^{-1}(-\infty, \alpha] = \bigcup_{p \in P} B_\alpha(p)$, aka. the sublevelset of f_P with value α equals the union of the α-balls around points in P

Distance field

- Now, given a point set $P \subseteq \mathbb{R}^d$, we define a function $f_P\colon \mathbb{R}^d \to \mathbb{R}$ called the distance field of P as:
$$f_P(x) = d(x, P) \quad \forall x \in \mathbb{R}^d$$
- **Observation**: $f_P^{-1}(-\infty, \alpha] = \bigcup_{p \in P} B_\alpha(p)$, aka. the sublevelset of f_P with value α equals the union of the α-balls around points in P
- The reason is that: A point x is in $\bigcup_{p \in P} B_\alpha(p)$ if and only if x is of a distance no more than α to a point in P

Distance field

- Now, given a point set $P \subseteq \mathbb{R}^d$, we define a function $f_P: \mathbb{R}^d \to \mathbb{R}$ called the distance field of P as:

$$f_P(x) = d(x, P) \quad \forall x \in \mathbb{R}^d$$

- **Observation**: $f_P^{-1}(-\infty, \alpha] = \bigcup_{p \in P} B_\alpha(p)$, aka. the sublevelset of f_P with value α equals the union of the α-balls around points in P
- The reason is that: A point x is in $\bigcup_{p \in P} B_\alpha(p)$ if and only if x is of a distance no more than α to a point in P
- Hence, the distance of x to P is no more than α

Distance field

- Now, given a point set $P \subseteq \mathbb{R}^d$, we define a function $f_P: \mathbb{R}^d \to \mathbb{R}$ called the distance field of P as:

$$f_P(x) = d(x, P) \quad \forall x \in \mathbb{R}^d$$

- **Observation**: $f_P^{-1}(-\infty, \alpha] = \bigcup_{p \in P} B_\alpha(p)$, aka. the sublevelset of f_P with value α equals the union of the α-balls around points in P
- The reason is that: A point x is in $\bigcup_{p \in P} B_\alpha(p)$ if and only if x is of a distance no more than α to a point in P
- Hence, the distance of x to P is no more than α
- **Corollary**: The above observation also means that **$PD(f_P)$ equals the PD of the Čech filtration of P**

Hausdorff distance

- To frame the stability for PDs w.r.t to two different point clouds P and Q, we also need a distance between P and Q
- This is the famous **Hausdorff** distance

Hausdorff distance

- To frame the stability for PDs w.r.t to two different point clouds P and Q, we also need a distance between P and Q
- This is the famous **Hausdorff** distance
- We first define the "unsymmetric Hausdorff" distance from P to Q as follows:

$$d_H^{unsym}(P, Q) = \max_{p \in P}\{d(p, Q)\}$$

aka. it is the maximum distance of any point in P to the whole set Q

Hausdorff distance

- To frame the stability for PDs w.r.t to two different point clouds P and Q, we also need a distance between P and Q
- This is the famous **Hausdorff** distance
- We first define the "unsymmetric Hausdorff" distance from P to Q as follows:

$$d_H^{unsym}(P, Q) = \max_{p \in P}\{d(p, Q)\}$$

aka. it is the maximum distance of any point in P to the whole set Q

- As the name indicates, the previous distance is not symmetric, i.e.,

$$d_H^{unsym}(P, Q) \neq d_H^{unsym}(Q, P)$$

while a distance has to be symmetric

Hausdorff distance

- To frame the stability for PDs w.r.t to two different point clouds P and Q, we also need a distance between P and Q
- This is the famous **Hausdorff** distance
- We first define the "unsymmetric Hausdorff" distance from P to Q as follows:

$$d_H^{unsym}(P, Q) = \max_{p \in P}\{d(p, Q)\}$$

aka. it is the maximum distance of any point in P to the whole set Q

- As the name indicates, the previous distance is not symmetric, i.e.,

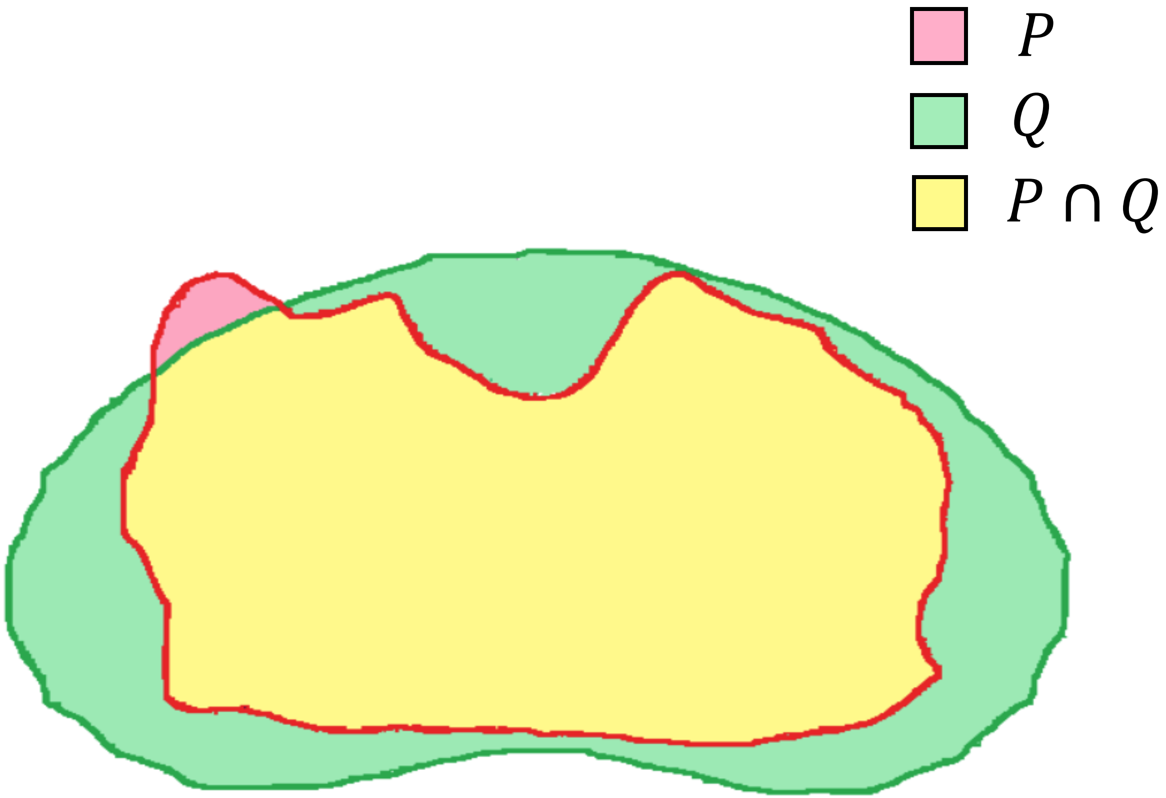
$$d_H^{unsym}(P, Q) \neq d_H^{unsym}(Q, P)$$

while a distance has to be symmetric

- The **Hausdorff** distance is then the maximum of the two:

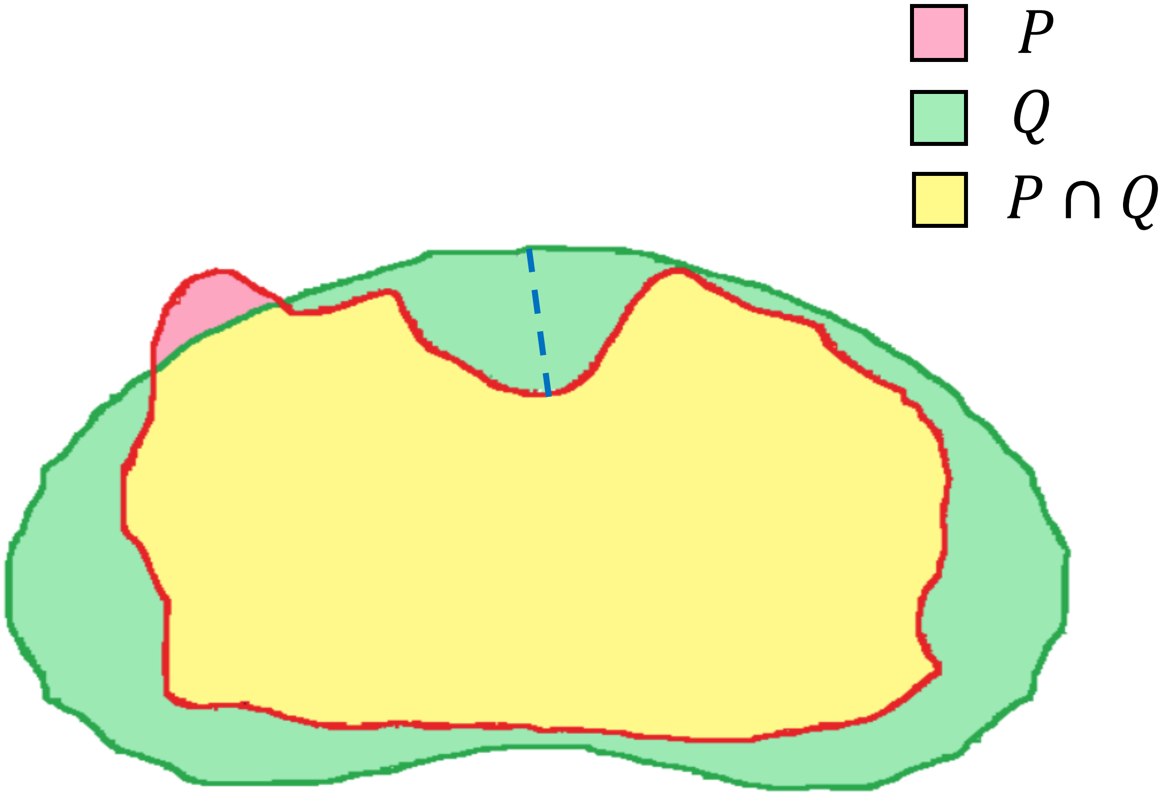
$$d_H(P, Q) = \max\{d_H^{unsym}(P, Q), d_H^{unsym}(Q, P)\}$$

Hausdorff distance



Img source: Karimi & Salcudean. Reducing the Hausdorff Distance in Medical Image Segmentation With Convolutional Neural Networks

Hausdorff distance



Img source: Karimi & Salcudean. Reducing the Hausdorff Distance in Medical Image Segmentation With Convolutional Neural Networks

Stability for point clouds

- **Fact:** For two point clouds P and Q, one has $\| f_P - f_Q \|_\infty \leq d_H(P, Q)$

Stability for point clouds

- **Fact:** For two point clouds P and Q, one has $\| f_P - f_Q \|_\infty \leq d_H(P, Q)$
- **Proof**: Since

$$\| f_P - f_Q \|_\infty = \max_{x\in\mathbb{R}^d}\{|f_P(x) - f_Q(x)|\} = \max_{x\in\mathbb{R}^d}\{|d(x, P) - d(x, Q)|\},$$

Stability for point clouds

- **Fact**: For two point clouds P and Q, one has $\| f_P - f_Q \|_\infty \leq d_H(P, Q)$
- **Proof**: Since

$$\| f_P - f_Q \|_\infty = \max_{x \in \mathbb{R}^d}\{|f_P(x) - f_Q(x)|\} = \max_{x \in \mathbb{R}^d}\{|d(x, P) - d(x, Q)|\},$$

- we only need to show that for any $x \in \mathbb{R}^d$, $|d(x, P) - d(x, Q)| \leq d_H(P, Q)$.

Stability for point clouds

- **Fact**: For two point clouds P and Q, one has $\| f_P - f_Q \|_\infty \leq d_H(P, Q)$
- **Proof**: Since

$$\| f_P - f_Q \|_\infty = \max_{x \in \mathbb{R}^d}\{|f_P(x) - f_Q(x)|\} = \max_{x \in \mathbb{R}^d}\{|d(x, P) - d(x, Q)|\},$$

- we only need to show that for any $x \in \mathbb{R}^d$, $|d(x, P) - d(x, Q)| \leq d_H(P, Q)$.
- To show this, we show

$$d(x, P) - d(x, Q) \leq d_H(P, Q) \text{ and } d(x, Q) - d(x, P) \leq d_H(P, Q)$$

Stability for point clouds

- **Fact**: For two point clouds P and Q, one has $\| f_P - f_Q \|_\infty \leq d_H(P, Q)$
- **Proof**: Since

$$\| f_P - f_Q \|_\infty = \max_{x \in \mathbb{R}^d}\{|f_P(x) - f_Q(x)|\} = \max_{x \in \mathbb{R}^d}\{|d(x, P) - d(x, Q)|\},$$

- we only need to show that for any $x \in \mathbb{R}^d$, $|d(x, P) - d(x, Q)| \leq d_H(P, Q)$.
- To show this, we show

$$d(x, P) - d(x, Q) \leq d_H(P, Q) \text{ and } d(x, Q) - d(x, P) \leq d_H(P, Q)$$

- We only show $d(x, P) - d(x, Q) \leq d_H(P, Q)$ and the other is similar

Stability for point clouds

- **Fact**: For two point clouds P and Q, one has $\| f_P - f_Q \|_\infty \leq d_H(P, Q)$
- **Proof**: Since

$$\| f_P - f_Q \|_\infty = \max_{x \in \mathbb{R}^d}\{|f_P(x) - f_Q(x)|\} = \max_{x \in \mathbb{R}^d}\{|d(x, P) - d(x, Q)|\},$$

- we only need to show that for any $x \in \mathbb{R}^d$, $|d(x, P) - d(x, Q)| \leq d_H(P, Q)$.
- To show this, we show

$$d(x, P) - d(x, Q) \leq d_H(P, Q) \text{ and } d(x, Q) - d(x, P) \leq d_H(P, Q)$$

- We only show $d(x, P) - d(x, Q) \leq d_H(P, Q)$ and the other is similar
- Let p be the point in P such that $d(x, P) = d(x, p)$.

Stability for point clouds

- **Fact**: For two point clouds P and Q, one has $\| f_P - f_Q \|_\infty \leq d_H(P, Q)$
- **Proof**: Since

$$\| f_P - f_Q \|_\infty = \max_{x \in \mathbb{R}^d}\{|f_P(x) - f_Q(x)|\} = \max_{x \in \mathbb{R}^d}\{|d(x, P) - d(x, Q)|\},$$

- we only need to show that for any $x \in \mathbb{R}^d$, $|d(x, P) - d(x, Q)| \leq d_H(P, Q)$.
- To show this, we show

$$d(x, P) - d(x, Q) \leq d_H(P, Q) \text{ and } d(x, Q) - d(x, P) \leq d_H(P, Q)$$

- We only show $d(x, P) - d(x, Q) \leq d_H(P, Q)$ and the other is similar
- Let p be the point in P such that $d(x, P) = d(x, p)$.
- Let q be the point in Q such that $d(p, Q) = d(p, q)$.

Stability for point clouds

- **Fact**: For two point clouds P and Q, one has $\| f_P - f_Q \|_\infty \leq d_H(P,Q)$
- **Proof**: Since

$$\| f_P - f_Q \|_\infty = \max_{x\in\mathbb{R}^d}\{|f_P(x) - f_Q(x)|\} = \max_{x\in\mathbb{R}^d}\{|d(x,P) - d(x,Q)|\},$$

- we only need to show that for any $x \in \mathbb{R}^d$, $|d(x,P) - d(x,Q)| \leq d_H(P,Q)$.
- To show this, we show

$$d(x,P) - d(x,Q) \leq d_H(P,Q) \text{ and } d(x,Q) - d(x,P) \leq d_H(P,Q)$$

- We only show $d(x,P) - d(x,Q) \leq d_H(P,Q)$ and the other is similar
- Let p be the point in P such that $d(x,P) = d(x,p)$.
- Let q be the point in Q such that $d(p,Q) = d(p,q)$.
- Based on the definition of Hausdorff distance, we have

$$d(p,q) = d(p,Q) \leq d_H(P,Q)$$

Stability for point clouds

- We then have

$$d(x,Q) \leq d(x,q) \leq d(x,p) + d(p,q) \leq d(x,P) + d_H(P,Q)$$

with the second inequality due to the triangle inequality

Stability for point clouds

- We then have

$$d(x,Q) \leq d(x,q) \leq d(x,p) + d(p,q) \leq d(x,P) + d_H(P,Q)$$

  with the second inequality due to the triangle inequality
- We then have $d(x,Q) - d(x,P) \leq d_H(P,Q)$

Stability for point clouds

- Define $PD_{\check{C}ech}(P) \coloneqq PD(\mathcal{C}(P))$ for a point cloud P and recall that $\mathcal{C}(P)$ is the Čech filtration of P

Stability for point clouds

- Define $PD_{\check{C}ech}(P) \coloneqq PD(\mathcal{C}(P))$ for a point cloud P and recall that $\mathcal{C}(P)$ is the Čech filtration of P
- **Stability Theorem for Point Clouds**: One has

$$d_{\mathrm{B}}(PD_{\check{C}ech}(P), PD_{\check{C}ech}(Q)) \leq d_H(P, Q)$$

Stability for point clouds

- Define $PD_{\check{C}ech}(P) \coloneqq PD(\mathcal{C}(P))$ for a point cloud P and recall that $\mathcal{C}(P)$ is the Čech filtration of P
- **Stability Theorem for Point Clouds**: One has

$$d_{\mathrm{B}}(PD_{\check{C}ech}(P), PD_{\check{C}ech}(Q)) \leq d_H(P, Q)$$

- **Proof**: First of all

$$PD_{\check{C}ech}(P) = PD(f_P) \text{ and } PD_{\check{C}ech}(Q) = PD\left(f_Q\right)$$

based on previous corollary

Stability for point clouds

- Define $PD_{\check{C}ech}(P) \coloneqq PD(\mathcal{C}(P))$ for a point cloud P and recall that $\mathcal{C}(P)$ is the Čech filtration of P
- **Stability Theorem for Point Clouds**: One has

$$d_{\mathrm{B}}(PD_{\check{C}ech}(P), PD_{\check{C}ech}(Q)) \le d_H(P, Q)$$

- **Proof**: First of all

$$PD_{\check{C}ech}(P) = PD(f_P) \text{ and } PD_{\check{C}ech}(Q) = PD\big(f_Q\big)$$

based on previous corollary

- We then have

$$d_{\mathrm{B}}(PD_{\check{C}ech}(P), PD_{\check{C}ech}(Q)) = d_{\mathrm{B}}(PD(f_P), PD(f_Q)) \le \| f_P - f_Q \|_\infty \le d_H(P, Q)$$

where the middle inequality is by previous stability theorem

Stability for point clouds

- Define $PD_{Rips}(P) \coloneqq PD(\mathcal{R}(P))$ for a point cloud P and recall that $\mathcal{R}(P)$ is the Rips filtration of P
- **Stability Theorem for Point Clouds (Vietoris-Rips):** One has

$$d_{\mathrm{B}}(PD_{Rips}(P), PD_{Rips}(Q)) \leq d_H(P, Q)$$

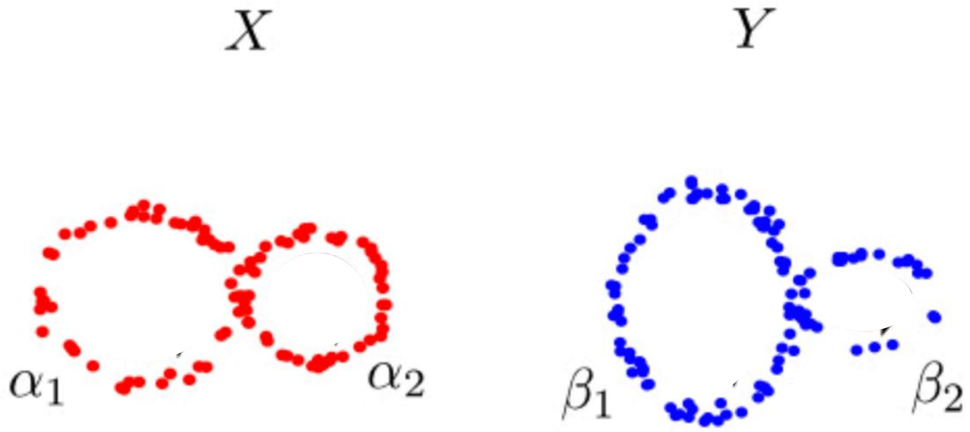
Stability for point clouds

- Define $PD_{Rips}(P) \coloneqq PD(\mathcal{R}(P))$ for a point cloud P and recall that $\mathcal{R}(P)$ is the Rips filtration of P
- **Stability Theorem for Point Clouds (Vietoris-Rips):** One has

$$d_{\mathrm{B}}(PD_{Rips}(P), PD_{Rips}(Q)) \leq d_H(P, Q)$$

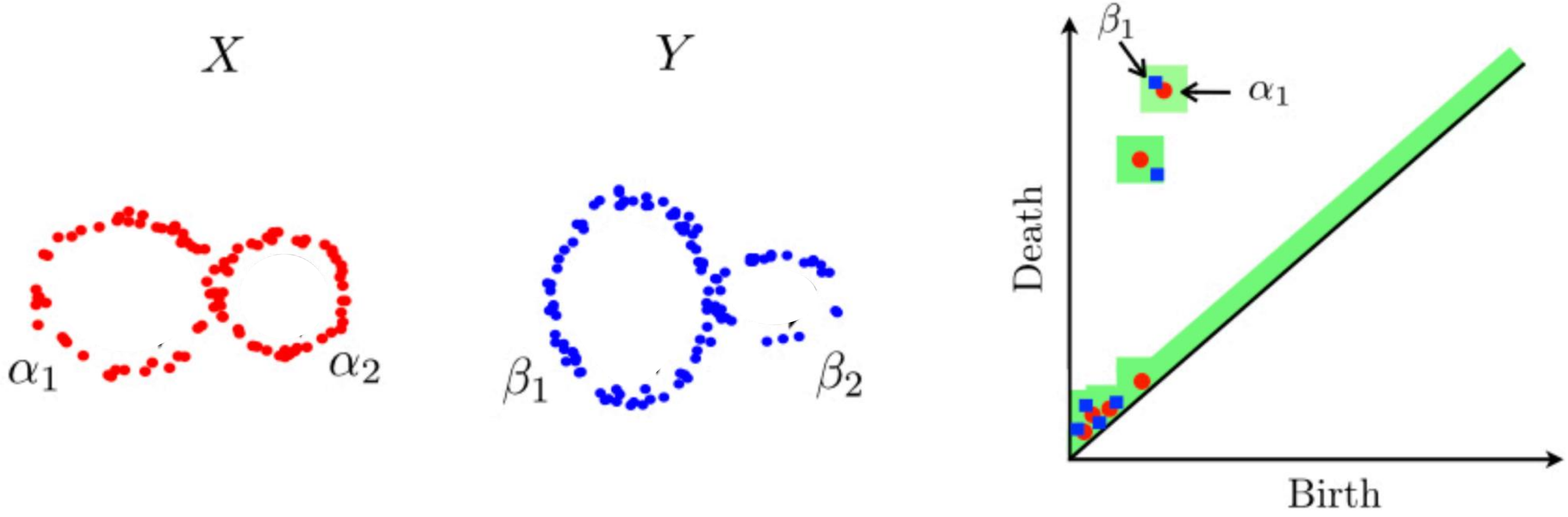
- Proof of this needs the more advanced notion of "interleaving" stability and is beyond scope

Stability for point clouds



Img source: Kusano, Fukumizu, Hiraoka. Kernel Method for Persistence Diagrams via Kernel Embedding and Weight Factor

Stability for point clouds



Img source: Kusano, Fukumizu, Hiraoka. Kernel Method for Persistence Diagrams via Kernel Embedding and Weight Factor

Stability for point clouds

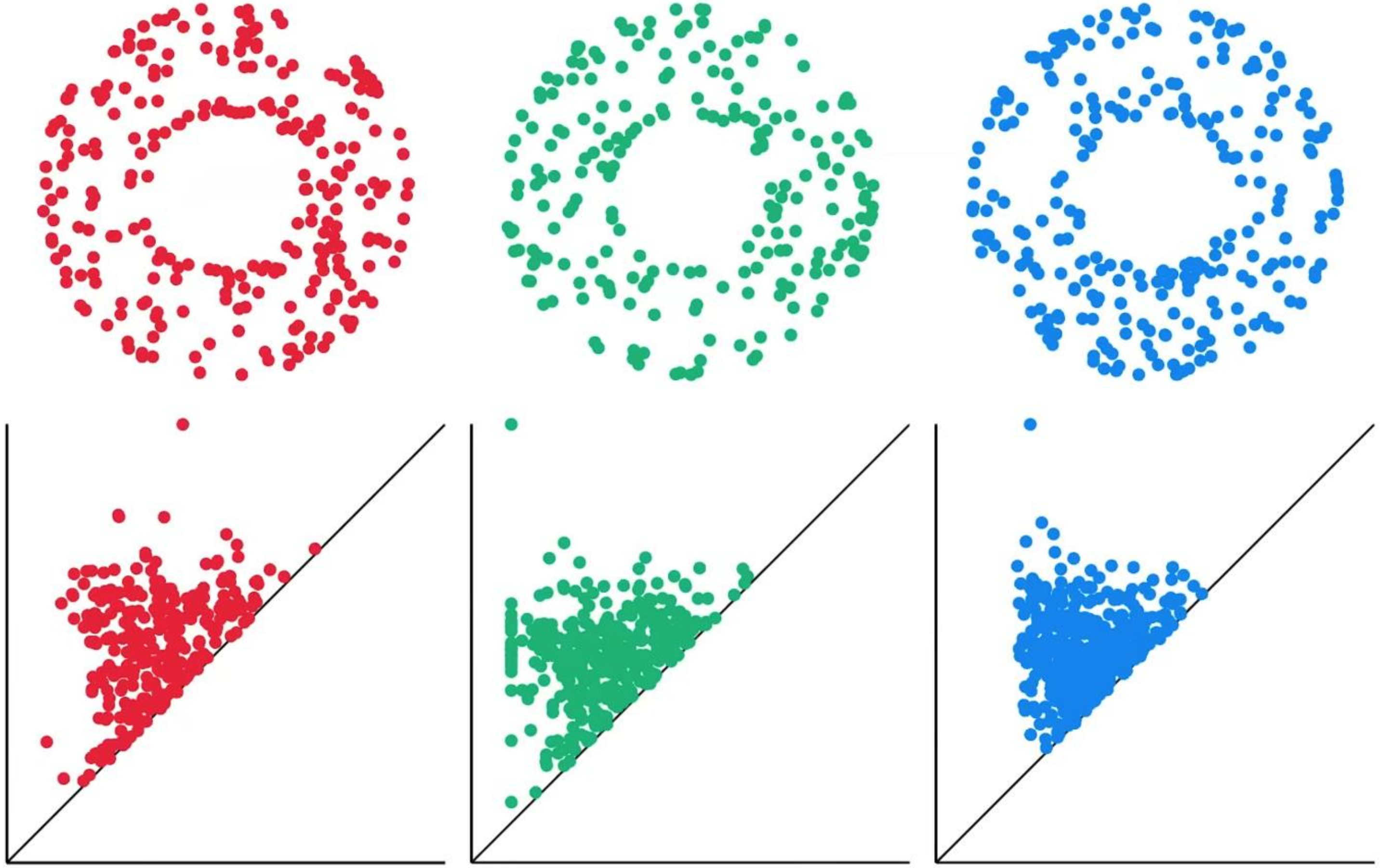


Figure from: Bastian Rieck: Topological Data Analysis for Machine Learning III: Topological Descriptors & How to Use Them (https://www.youtube.com/watch?app=desktop&v=7i1kabhl5IU)

Similarity of Čech and Rips PD

Similarity of Čech and Rips PD

- For a point cloud P, define $PD^{\log}_{\check{C}ech}(P)$ as the PD derived from $PD_{\check{C}ech}(P)$ by taking the logarithm of the birth and death value for each point (with an arbitrary but fixed base)

Similarity of Čech and Rips PD

- For a point cloud P, define $PD^{\log}_{\check{C}ech}(P)$ as the PD derived from $PD_{\check{C}ech}(P)$ by taking the logarithm of the birth and death value for each point (with an arbitrary but fixed base)
- Define $PD^{\log}_{Rips}(P)$ similarly

Similarity of Čech and Rips PD

- For a point cloud P, define $PD_{\check{C}ech}^{\log}(P)$ as the PD derived from $PD_{\check{C}ech}(P)$ by taking the logarithm of the birth and death value for each point (with an arbitrary but fixed base)
- Define $PD_{Rips}^{\log}(P)$ similarly
- **Theorem**: One has

$$d_{\mathrm{B}}(PD_{\check{C}ech}^{\log}(P), PD_{Rips}^{\log}(P)) \leq \log 2$$

Similarity of Čech and Rips PD

- For a point cloud P, define $PD^{\log}_{Čech}(P)$ as the PD derived from $PD_{Čech}(P)$ by taking the logarithm of the birth and death value for each point (with an arbitrary but fixed base)
- Define $PD^{\log}_{Rips}(P)$ similarly
- **Theorem**: One has

$$d_{\mathrm{B}}(PD^{\log}_{Čech}(P), PD^{\log}_{Rips}(P)) \leq \log 2$$

- The theorem follows from the previous "interleaving" between Čech and Rips complexes (but details omitted)

$$\mathbb{C}^{\alpha}(P) \subseteq \mathbb{VR}^{\alpha}(P) \subseteq \mathbb{C}^{2\alpha}(P)$$

Computing bottleneck distance

- https://github.com/nihell/tutorialathon/blob/master/BottleneckTutorial.ipynb
- https://www.youtube.com/watch?v=4WswT9snTjc

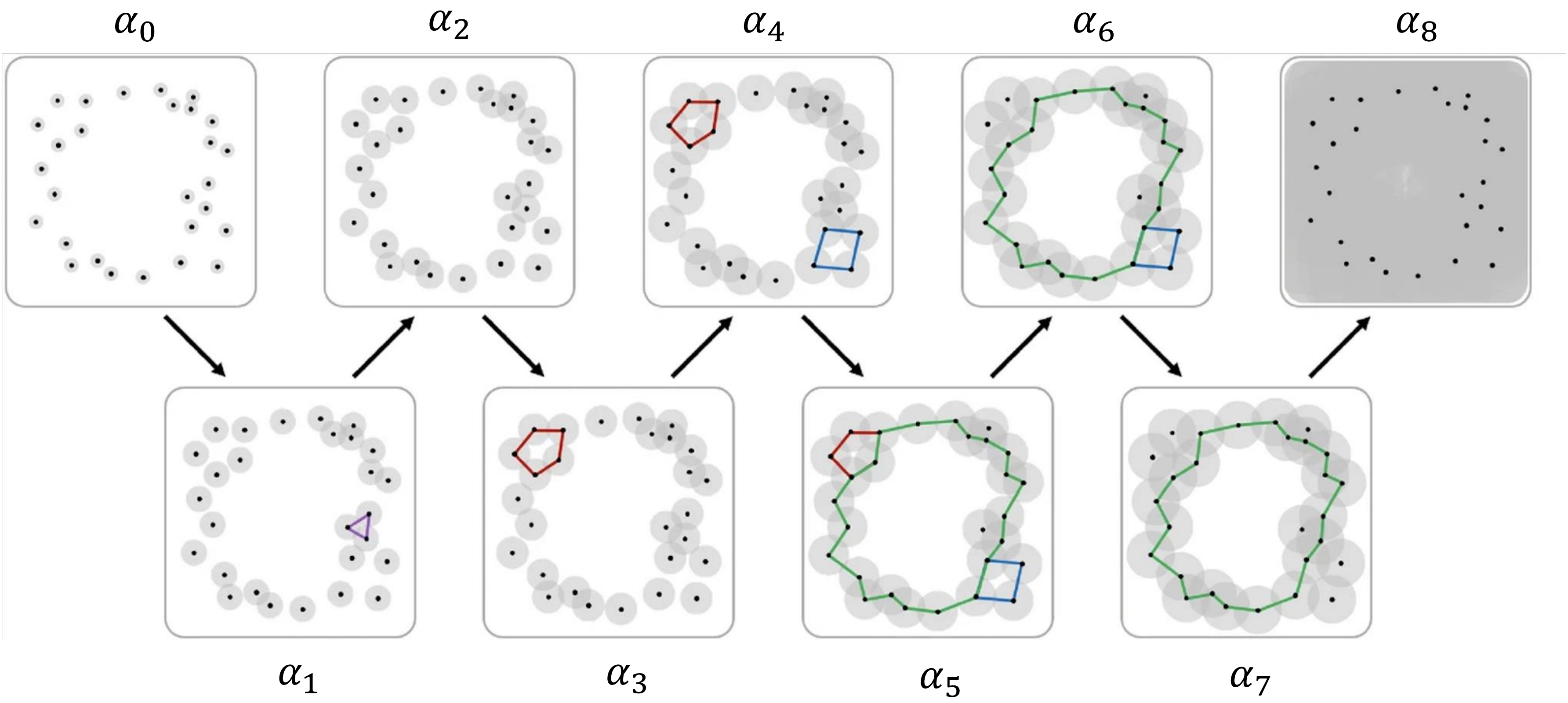
A practical use of persistence barcode

- A practical use of drawing a PD as a barcode is that barcode provides a way to “visualize” the Betti number across the different range (value α)

Image source: Bobrowski O, Skraba P. A universal null-distribution for topological data analysis.

A practical use of persistence barcode

- A practical use of drawing a PD as a barcode is that barcode provides a way to "visualize" the Betti number across the different range (value α)
- For the previous filtration on a point cloud and its 1d PD



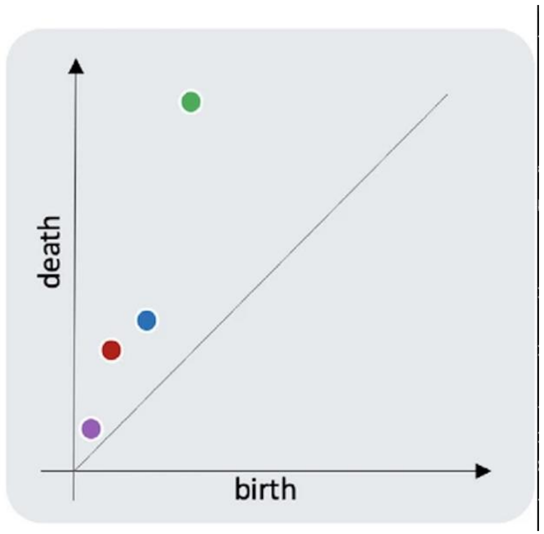
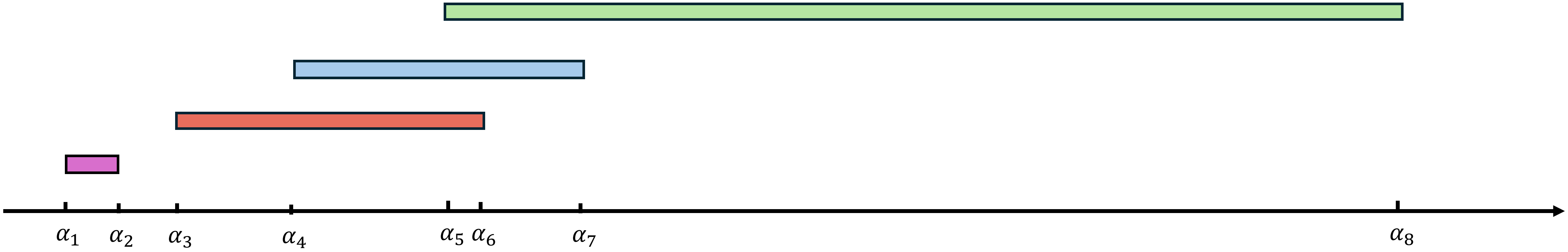


Image source: Bobrowski O, Skraba P. A universal null-distribution for topological data analysis.

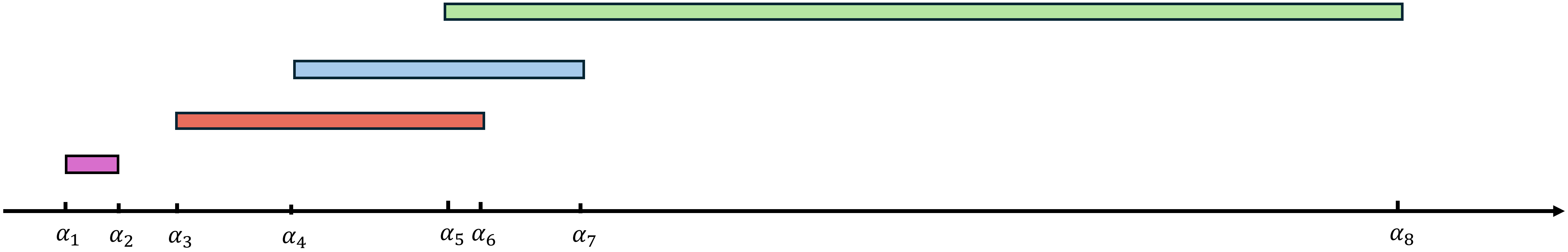
A practical use of persistence barcode

- The following is its 1d barcode



A practical use of persistence barcode

- The following is its 1d barcode
- Observe, if you count the number of intervals (bars) containing a certain value, then it gives you the 1st Betti for the complex corresponding to the range (value α) in the filtration



A practical use of persistence barcode

- The following is its 1d barcode
- Observe, if you count the number of intervals (bars) containing a certain value, then it gives you the 1st Betti for the complex corresponding to the range (value α) in the filtration

A practical use of persistence barcode

- The following is its 1d barcode
- Observe, if you count the number of intervals (bars) containing a certain value, then it gives you the 1st Betti for the complex corresponding to the range (value α) in the filtration

A practical use of persistence barcode

- The following is its 1d barcode
- Observe, if you count the number of intervals (bars) containing a certain value, then it gives you the 1st Betti for the complex corresponding to the range (value α) in the filtration

A practical use of persistence barcode

- The following is its 1d barcode
- Observe, if you count the number of intervals (bars) containing a certain value, then it gives you the 1st Betti for the complex corresponding to the range (value α) in the filtration

A practical use of persistence barcode

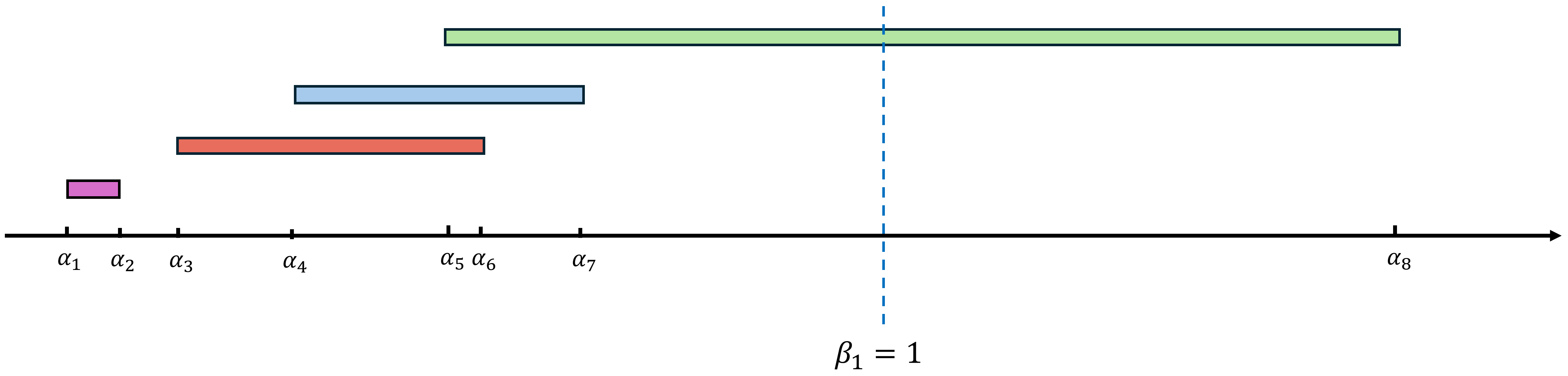
- The following is its 1d barcode
- Observe, if you count the number of intervals (bars) containing a certain value, then it gives you the 1st Betti for the complex corresponding to the range (value α) in the filtration

A practical use of persistence barcode

- The following is its 1d barcode
- Observe, if you count the number of intervals (bars) containing a certain value, then it gives you the 1st Betti for the complex corresponding to the range (value α) in the filtration

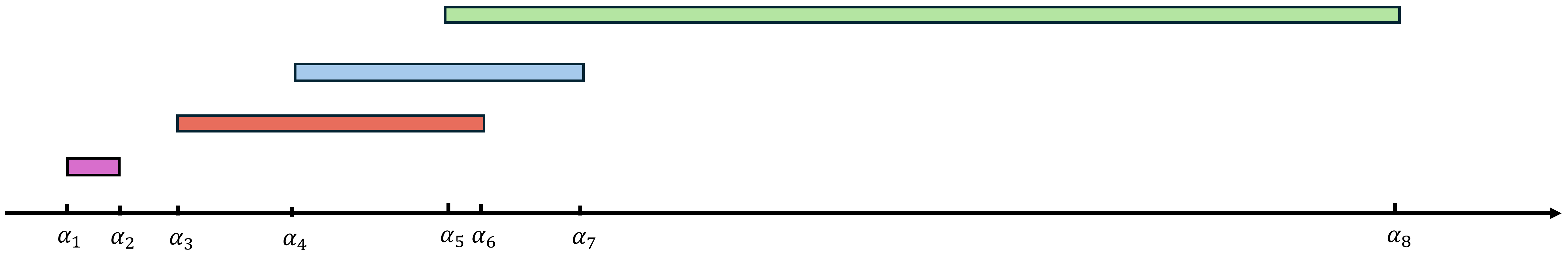
A practical use of persistence barcode

- The following is its 1d barcode
- Observe, if you count the number of intervals (bars) containing a certain value, then it gives you the 1st Betti for the complex corresponding to the range (value α) in the filtration



A practical use of persistence barcode

- More powerfully, if you look at the range of the values where the Betti number stays the same, and take the longest range, that would give you the most probably inference of the Betti number



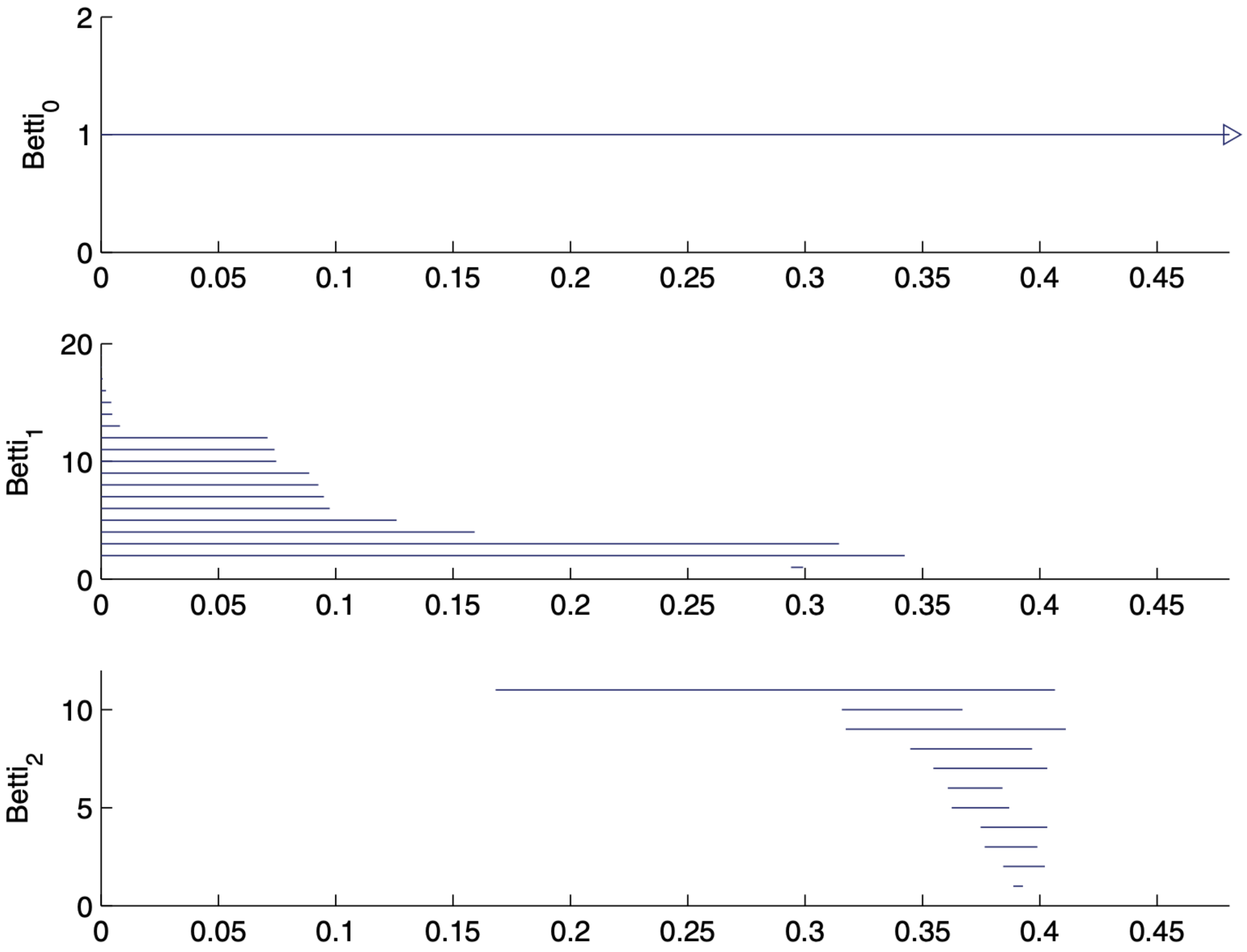
A practical use of persistence barcode

- More powerfully, if you look at the range of the values where the Betti number stays the same, and take the longest range, that would give you the most probably inference of the Betti number

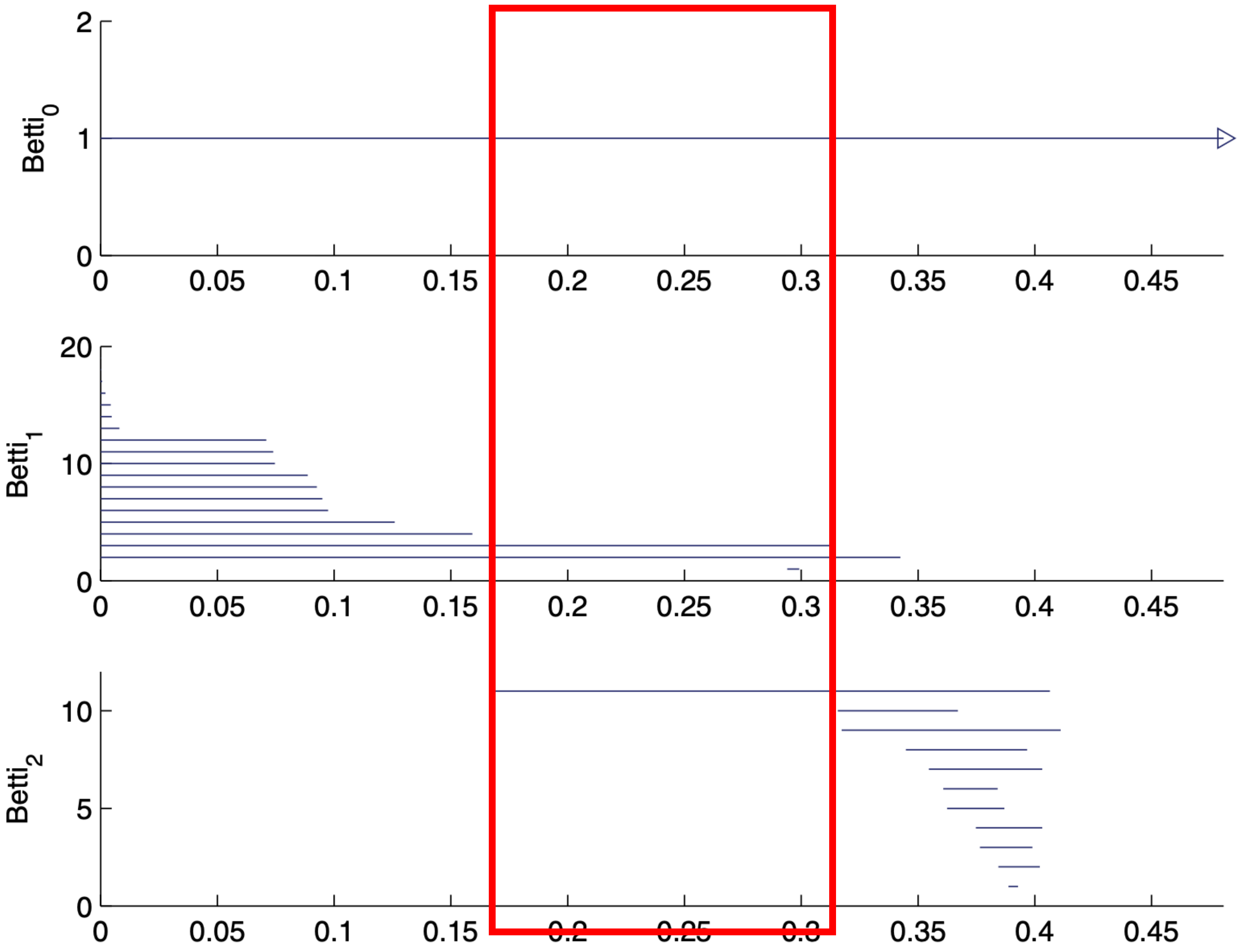
A practical use of persistence barcode

- More powerfully, if you look at the range of the values where the Betti number stays the same, and take the longest range, that would give you the most probably inference of the Betti number
- So most probably, for the previous point cloud, $\beta_1 = 1$ (complying out intuitions)

Another example



Another example



Counting Betti number using PD

- Now we know that taking the intersection of a vertical line with the intervals in the barcode gives you the Betti number at the value of the vertical line, what about PD? Can we do similar things in PD?

Image source: Bobrowski O, Skraba P. A universal null-distribution for topological data analysis.

Counting Betti number using PD

- It turns out that, the number of intervals intersecting a value α as manifested on the PD is the number of pointing in the upper-left quadrant of the point (α, α) on the diagonal

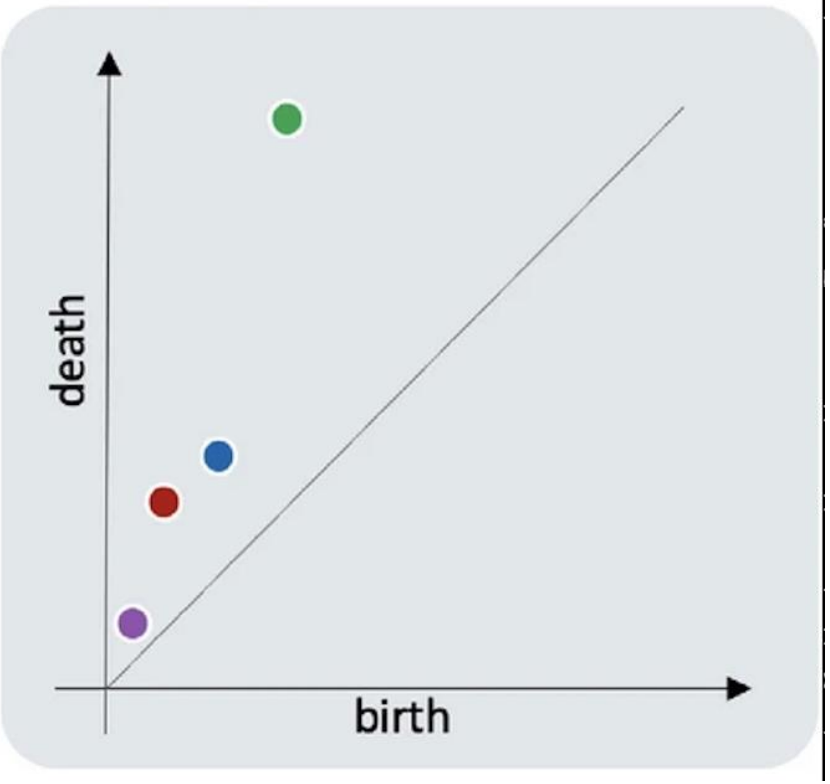


Image source: Bobrowski O, Skraba P. A universal null-distribution for topological data analysis.

Counting Betti number using PD

- It turns out that, the number of intervals intersecting a value α as manifested on the PD is the number of pointing in the upper-left quadrant of the point (α, α) on the diagonal

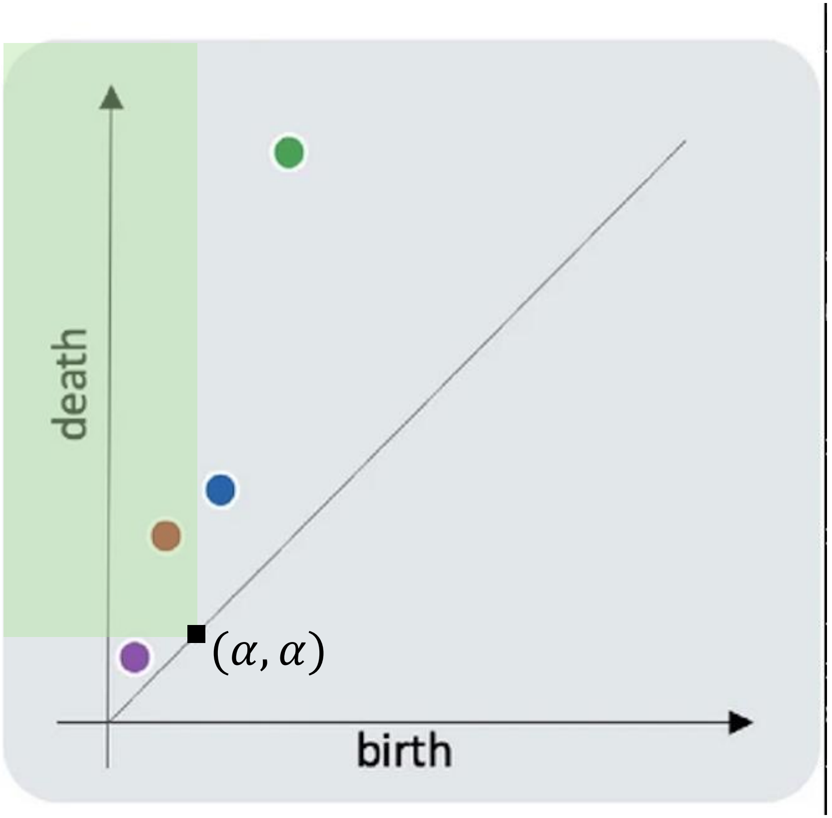


Image source: Bobrowski O, Skraba P. A universal null-distribution for topological data analysis.

On barcode

α_1
α_2
α_3
α_4
α_5
α_6
α_7
α_8
$\beta_1 = 1$

On PD

- $\alpha_3 < \alpha < \alpha_4$

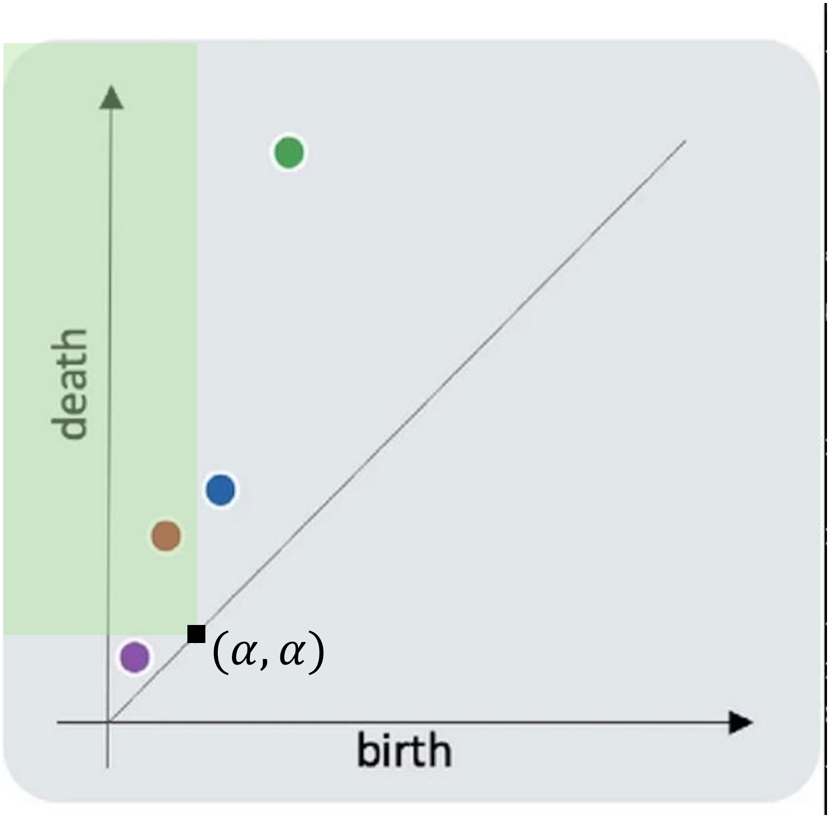


Image source: Bobrowski O, Skraba P. A universal null-distribution for topological data analysis.

On Barcode

α_1
α_2
α_3
α_4
α_5
α_6
α_7
α_8
$\beta_1 = 3$

On PD

- $\alpha_5 < \alpha < \alpha_6$

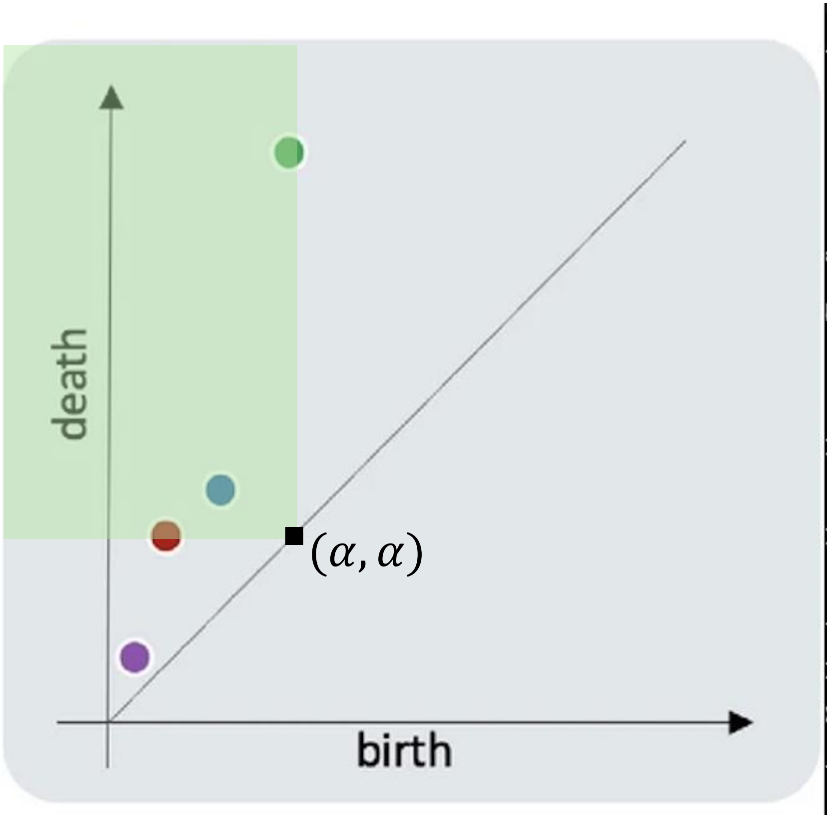


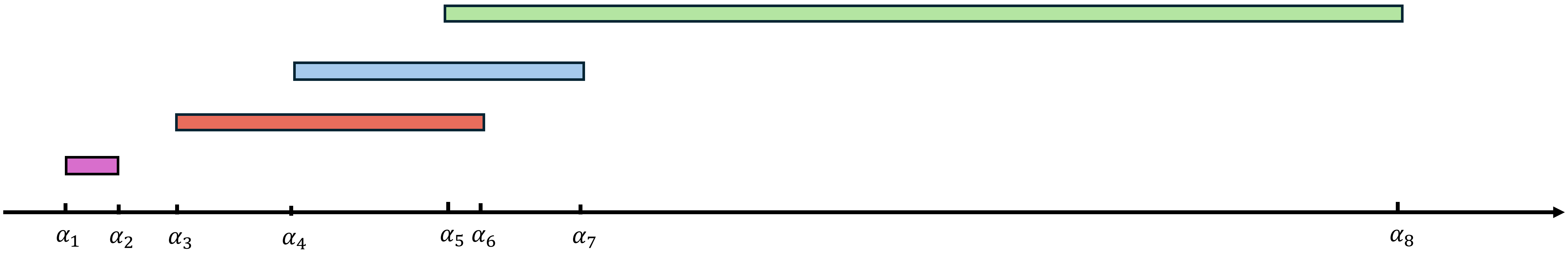
Image source: Bobrowski O, Skraba P. A universal null-distribution for topological data analysis.

Length “cut-off” for intervals

- A typically belief in TDA is that people often think of features (intervals) with long lifespans as robust, important features, whereas a short lifespan may be an indication that the feature is less essential and may in fact be due to noise in data
- A commonly adopted approach in using PD/barcode for inference is to find a “cut-off” for the short intervals (noise) , and focus on the long intervals (actual features) only

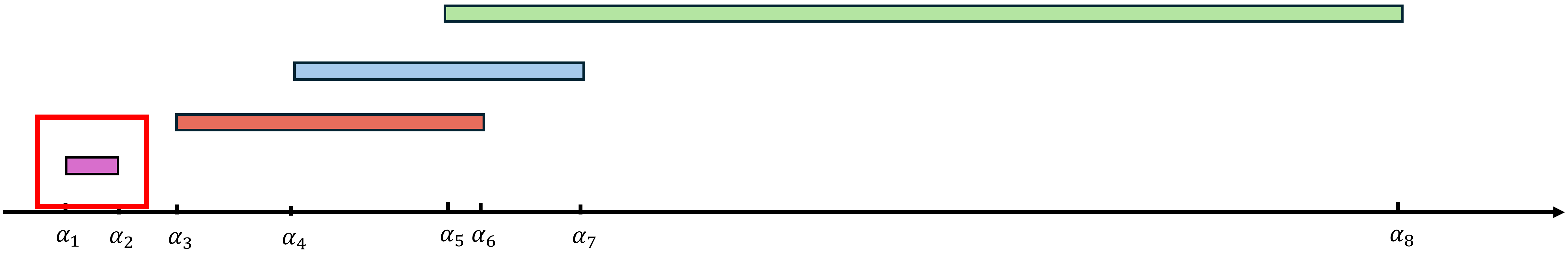
Length “cut-off” for intervals

- A typically belief in TDA is that people often think of features (intervals) with long lifespans as robust, important features, whereas a short lifespan may be an indication that the feature is less essential and may in fact be due to noise in data
- A commonly adopted approach in using PD/barcode for inference is to find a “cut-off” for the short intervals (noise) , and focus on the long intervals (actual features) only



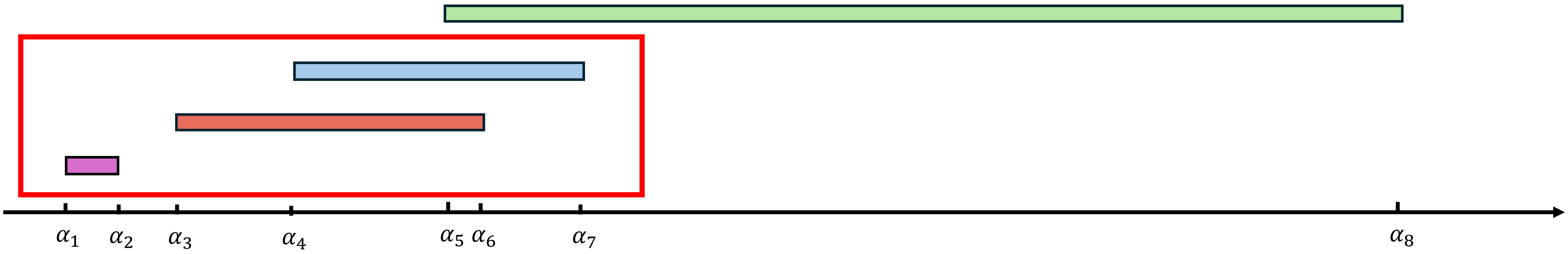
Length "cut-off" for intervals

- A typically belief in TDA is that people often think of features (intervals) with long lifespans as robust, important features, whereas a short lifespan may be an indication that the feature is less essential and may in fact be due to noise in data
- A commonly adopted approach in using PD/barcode for inference is to find a "cut-off" for the short intervals (noise) , and focus on the long intervals (actual features) only



Length "cut-off" for intervals

- A typically belief in TDA is that people often think of features (intervals) with long lifespans as robust, important features, whereas a short lifespan may be an indication that the feature is less essential and may in fact be due to noise in data
- A commonly adopted approach in using PD/barcode for inference is to find a "cut-off" for the short intervals (noise) , and focus on the long intervals (actual features) only



Length “cut-off” for intervals

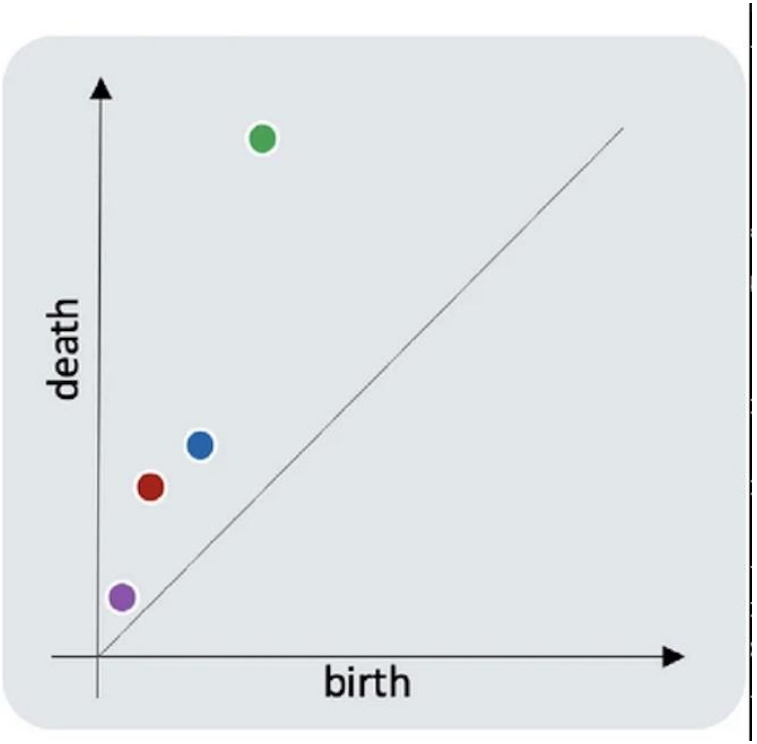
- The interpretation for the length “cut-off” in the PD is that we ignore all points which are within a certain distance from the diagonal

Length “cut-off” for intervals

- The interpretation for the length “cut-off” in the PD is that we ignore all points which are within a certain distance from the diagonal
- This is equivalent to take the lower-half space of a $45°$ degree line (parallel to disgonal)

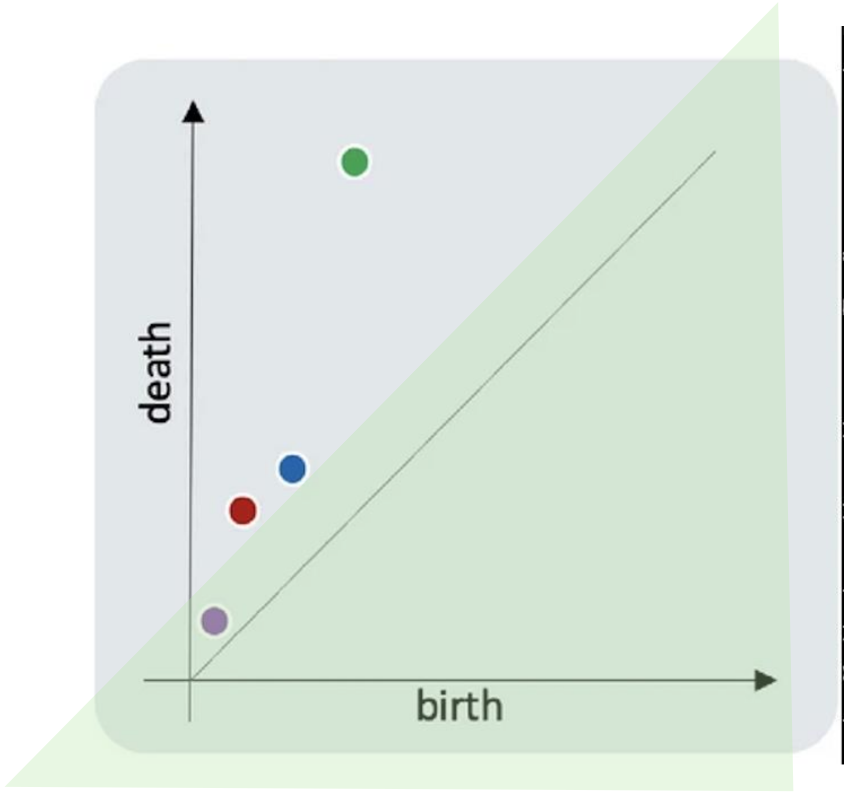
Length "cut-off" for intervals

- The interpretation for the length "cut-off" in the PD is that we ignore all points which are within a certain distance from the diagonal
- This is equivalent to take the lower-half space of a $45°$ degree line (parallel to disgonal)



Length "cut-off" for intervals

- The interpretation for the length "cut-off" in the PD is that we ignore all points which are within a certain distance from the diagonal
- This is equivalent to take the lower-half space of a $45°$ degree line (parallel to disgonal)



Length "cut-off" for intervals

- The interpretation for the length "cut-off" in the PD is that we ignore all points which are within a certain distance from the diagonal
- This is equivalent to take the lower-half space of a $45°$ degree line (parallel to disgonal)