The Datasaurus and Persistent Homology

Ryan Jensen

Stephen F. Austin State University

November 3 2019

Outline

Outline

- Creating of the Datasaurus

Outline

Outline

- Creating of the Datasaurus
- Persistent Homology

Outline

Outline

- Creating of the Datasaurus
- Persistent Homology
- Results for Datasaurus

Outline

Outline

- Creating of the Datasaurus
- Persistent Homology
- Results for Datasaurus
- Large Scale Geometry and Persistent Homology

The Datasaurus

Goal

Justin Matejka and George Fitzmaurice morphed the "Datasaurus" into various shapes, all of which have essentially the same statistics. Their point was to always visualize data. The goal is to use persistent homology to differentiate the morphed shapes.

Figure: https://www.autodeskresearch.com/publications/samestats

The Datasaurus

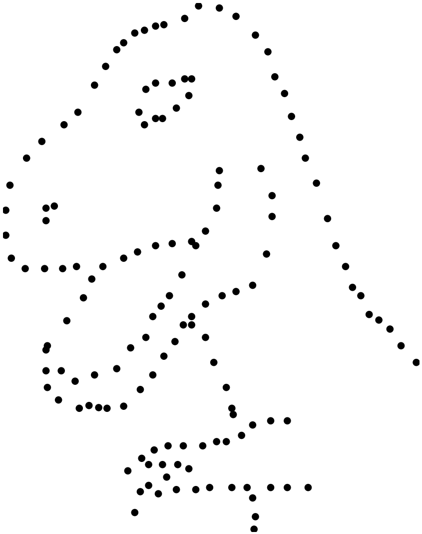
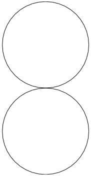


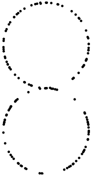
Figure: The Datasaurus

The Datasaurus



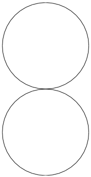
(a) Target Simplicial Complex

(b) Starting Point Cloud



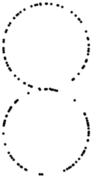
(c) Ending Point Cloud

The Datasaurus



(a) Target Simplicial Complex

(b) Starting Point Cloud



(c) Ending Point Cloud

Video Link

The Datasaurus

Morphing the Datasaurus

- Start with a point cloud, for example the Datasaurus points, and a target simplicial complex, such as a wedge of two circles.

The Datasaurus

Morphing the Datasaurus

- Start with a point cloud, for example the Datasaurus points, and a target simplicial complex, such as a wedge of two circles.
- Randomly select a point and move it a random amount.

The Datasaurus

Morphing the Datasaurus

- Start with a point cloud, for example the Datasaurus points, and a target simplicial complex, such as a wedge of two circles.
- Randomly select a point and move it a random amount.
- Check to see if the statistics stay close enough.

The Datasaurus

Morphing the Datasaurus

- Start with a point cloud, for example the Datasaurus points, and a target simplicial complex, such as a wedge of two circles.
- Randomly select a point and move it a random amount.
- Check to see if the statistics stay close enough.
  - Check to see if new point is closer to the target shape than the old point.

The Datasaurus

Morphing the Datasaurus

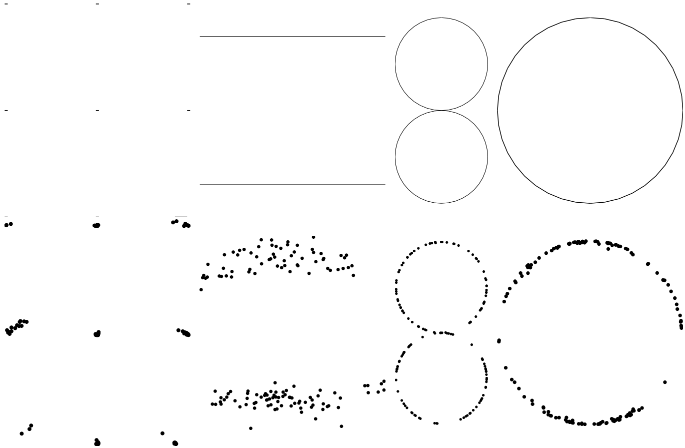
- Start with a point cloud, for example the Datasaurus points, and a target simplicial complex, such as a wedge of two circles.
- Randomly select a point and move it a random amount.
- Check to see if the statistics stay close enough.
    - Check to see if new point is closer to the target shape than the old point.
    - Check the "temperature" (random number), if a high temperature move the point even if it is farther away than the old point. This avoids getting stuck in a local optimization.

The Datasaurus

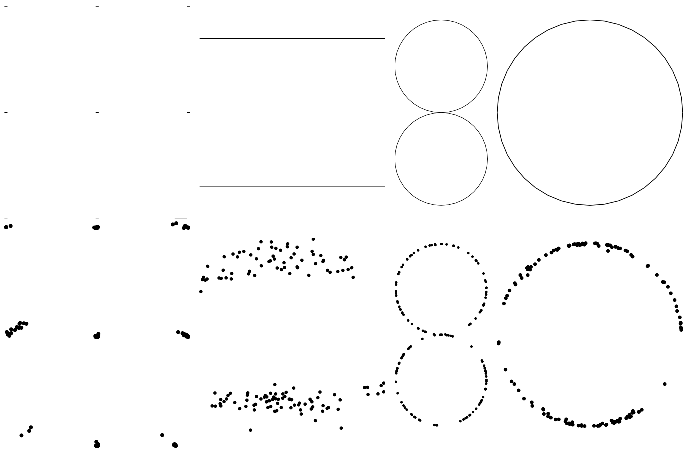
Morphing the Datasaurus

- Start with a point cloud, for example the Datasaurus points, and a target simplicial complex, such as a wedge of two circles.
- Randomly select a point and move it a random amount.
- Check to see if the statistics stay close enough.
  - Check to see if new point is closer to the target shape than the old point.
  - Check the “temperature” (random number), if a high temperature move the point even if it is farther away than the old point. This avoids getting stuck in a local optimization.
- 100,000 iterations of moving points.

The Datasaurus



The Datasaurus



Video Link

Persistent Homology

Homology is First

The idea of **homology** is to associate an algebraic object, such as an abelian group or a vector space, to another mathematical object, in this case a topological (large scale) space. The **betti number** is the rank/dimension of the algebraic object.

Homology is First

The idea of **homology** is to associate an algebraic object, such as an abelian group or a vector space, to another mathematical object, in this case a topological (large scale) space. The **betti number** is the rank/dimension of the algebraic object.

Even though homology is an algebraic characterization, we speak of the homology of a topological (large scale) space, and often by homology we mean the betti numbers of the associated algebraic objects.

Homology is First

The idea of **homology** is to associate an algebraic object, such as an abelian group or a vector space, to another mathematical object, in this case a topological (large scale) space. The **betti number** is the rank/dimension of the algebraic object.
Even though homology is an algebraic characterization, we speak of the homology of a topological (large scale) space, and often by homology we mean the betti numbers of the associated algebraic objects.

0. The zeroth homology, H_0, counts the number of connected components.

Homology is First

The idea of **homology** is to associate an algebraic object, such as an abelian group or a vector space, to another mathematical object, in this case a topological (large scale) space. The **betti number** is the rank/dimension of the algebraic object.

Even though homology is an algebraic characterization, we speak of the homology of a topological (large scale) space, and often by homology we mean the betti numbers of the associated algebraic objects.

0. The zeroth homology, H_0, counts the number of connected components.
1. The first homology, H_1, counts the number "holes".

Homology is First

The idea of **homology** is to associate an algebraic object, such as an abelian group or a vector space, to another mathematical object, in this case a topological (large scale) space. The **betti number** is the rank/dimension of the algebraic object.
Even though homology is an algebraic characterization, we speak of the homology of a topological (large scale) space, and often by homology we mean the betti numbers of the associated algebraic objects.

0. The zeroth homology, H_0, counts the number of connected components.
1. The first homology, H_1, counts the number "holes".
2. The second homology, H_2, counts the number 2 dimensional "voids".

Homology is First

The idea of **homology** is to associate an algebraic object, such as an abelian group or a vector space, to another mathematical object, in this case a topological (large scale) space. The **betti number** is the rank/dimension of the algebraic object.
Even though homology is an algebraic characterization, we speak of the homology of a topological (large scale) space, and often by homology we mean the betti numbers of the associated algebraic objects.

0. The zeroth homology, H_0, counts the number of connected components.
1. The first homology, H_1, counts the number "holes".
2. The second homology, H_2, counts the number 2 dimensional "voids".
3. H_n counts the number of n dimensional "voids".

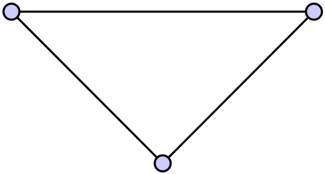
Simplicial Example

The homology of a simplicial complex counts the number of cycles which are not boundaries in each dimension.

Simplicial Example

The homology of a simplicial complex counts the number of cycles which are not boundaries in each dimension.

Here is an example:

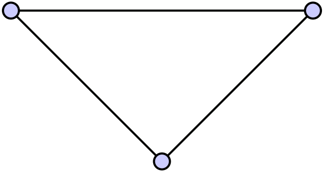


A 1-cycle which is not a boundary.

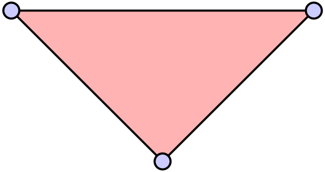
Simplicial Example

The homology of a simplicial complex counts the number of cycles which are not boundaries in each dimension.

Here is an example:



A 1-cycle which is not a boundary.



A 1-cycle which is a boundary.

Persistent Homology

- Persistent homology - Edelsbrunner & Harer 2005, Carlsson & Zomorodia 2005, Ghrist 2007.

Persistent Homology

- Persistent homology - Edelsbrunner & Harer 2005, Carlsson & Zomorodia 2005, Ghrist 2007.
- Persistent homology is concerned about the homology of a sequence of related spaces:

$$X_0 \xrightarrow{f_0} X_1 \xrightarrow{f_1} X_2 \xrightarrow{f_2} \cdots X_{n-1} \xrightarrow{f_{n-1}} X_n.$$

Persistent Homology

- Persistent homology - Edelsbrunner & Harer 2005, Carlsson & Zomorodia 2005, Ghrist 2007.
- Persistent homology is concerned about the homology of a sequence of related spaces:

$$X_0 \xrightarrow{f_0} X_1 \xrightarrow{f_1} X_2 \xrightarrow{f_2} \cdots X_{n-1} \xrightarrow{f_{n-1}} X_n.$$

- The spaces are related by the f_is, which are usually inclusion.

Persistent Homology

- Persistent homology - Edelsbrunner & Harer 2005, Carlsson & Zomorodia 2005, Ghrist 2007.
- Persistent homology is concerned about the homology of a sequence of related spaces:

$$X_0 \xrightarrow{f_0} X_1 \xrightarrow{f_1} X_2 \xrightarrow{f_2} \cdots X_{n-1} \xrightarrow{f_{n-1}} X_n.$$

- The spaces are related by the f_is, which are usually inclusion.
- The sequence of spaces yields a sequence of algebraic structures (vector spaces) for each dimension k:

$$H_k(X_0) \to H_k(X_1) \to H_k(X_2) \to \cdots \to H_K(X_n).$$

Persistent Homology

- Persistent homology - Edelsbrunner & Harer 2005, Carlsson & Zomorodia 2005, Ghrist 2007.
- Persistent homology is concerned about the homology of a sequence of related spaces:

$$X_0 \xrightarrow{f_0} X_1 \xrightarrow{f_1} X_2 \xrightarrow{f_2} \cdots X_{n-1} \xrightarrow{f_{n-1}} X_n.$$

- The spaces are related by the f_is, which are usually inclusion.
- The sequence of spaces yields a sequence of algebraic structures (vector spaces) for each dimension k:

$$H_k(X_0) \to H_k(X_1) \to H_k(X_2) \to \cdots \to H_K(X_n).$$

- A **bar code** is a visual representation of the betti numbers.

Persistent Homology Process

- Start with a metric (large scale) space X.

Persistent Homology Process

- Start with a metric (large scale) space X.
- From X, derive a filtration of complexes:

$$X \mapsto (X_0 \to X_1 \to X_2 \to \cdots X_{n-1} \to X_n).$$

Persistent Homology Process

- Start with a metric (large scale) space X.
- From X, derive a filtration of complexes:

$$X \mapsto (X_0 \to X_1 \to X_2 \to \cdots X_{n-1} \to X_n).$$

- Convert the filtration of complexes to a sequence of vector spaces (called a **persistence vector space**). For each dimension k:

$$(X_0 \to X_1 \to X_2 \to \cdots X_{n-1} \to X_n) \mapsto$$
$$H_k(X_0) \to H_K(X_1) \to \cdots H_k(X_{n-1}) \to H_k(X_n)$$

Persistent Homology Process

- Start with a metric (large scale) space X.
- From X, derive a filtration of complexes:

$$X \mapsto (X_0 \to X_1 \to X_2 \to \cdots X_{n-1} \to X_n).$$

- Convert the filtration of complexes to a sequence of vector spaces (called a **persistence vector space**). For each dimension k:

$$(X_0 \to X_1 \to X_2 \to \cdots X_{n-1} \to X_n) \mapsto$$
$$H_k(X_0) \to H_K(X_1) \to \cdots H_k(X_{n-1}) \to H_k(X_n)$$

- From the persistence vector space, give the barcode.

Persistent Homology Process

- Start with a metric (large scale) space X.
- From X, derive a filtration of complexes:

$$X \mapsto (X_0 \to X_1 \to X_2 \to \cdots X_{n-1} \to X_n).$$

- Convert the filtration of complexes to a sequence of vector spaces (called a **persistence vector space**). For each dimension k:

$$(X_0 \to X_1 \to X_2 \to \cdots X_{n-1} \to X_n) \mapsto$$
$$H_k(X_0) \to H_K(X_1) \to \cdots H_k(X_{n-1}) \to H_k(X_n)$$

- From the persistence vector space, give the barcode.
  - A homology class of $H_K(X_i)$ is **born** when it is not in the image of $H_k(X_{i-1})$.

Persistent Homology Process

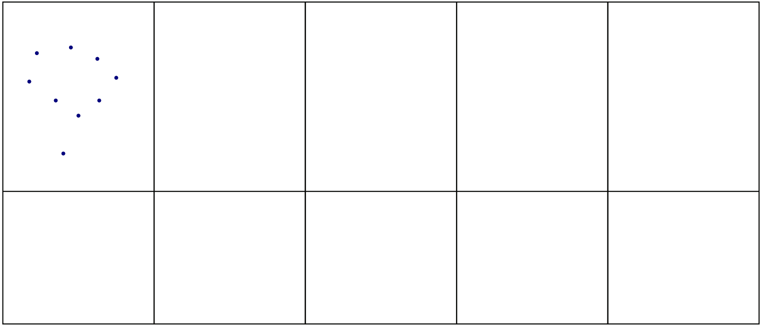
- Start with a metric (large scale) space X.
- From X, derive a filtration of complexes:

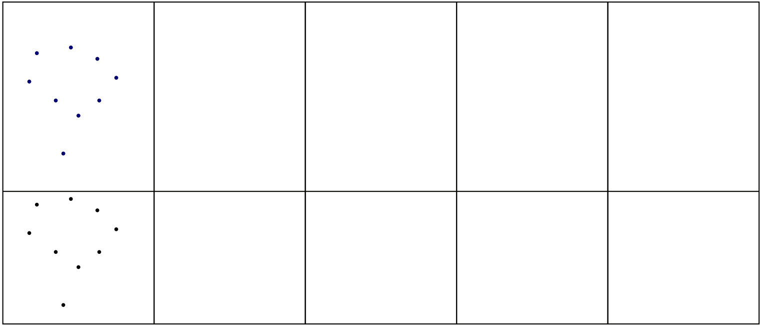
$$X \mapsto (X_0 \to X_1 \to X_2 \to \cdots X_{n-1} \to X_n).$$

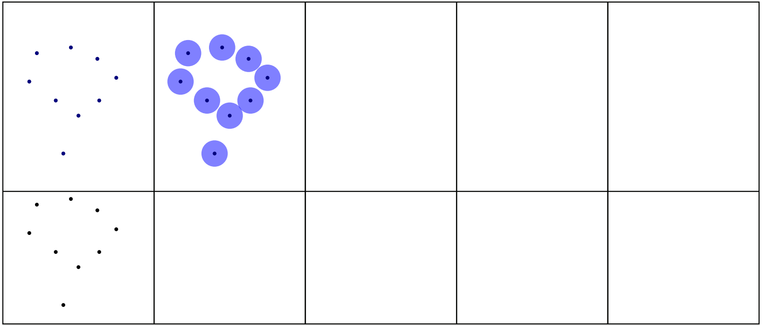
- Convert the filtration of complexes to a sequence of vector spaces (called a **persistence vector space**). For each dimension k:

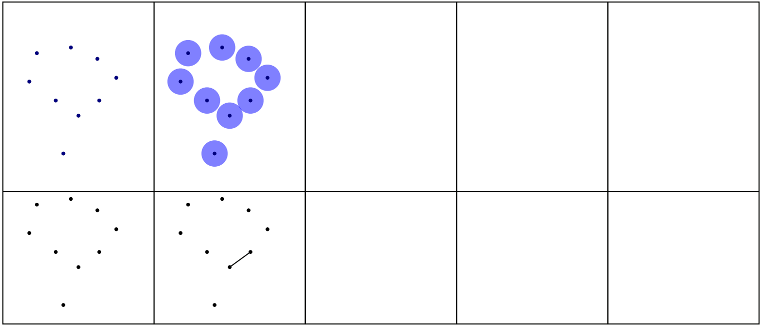
$$(X_0 \to X_1 \to X_2 \to \cdots X_{n-1} \to X_n) \mapsto$$
$$H_k(X_0) \to H_K(X_1) \to \cdots H_k(X_{n-1}) \to H_k(X_n)$$

- From the persistence vector space, give the barcode.
  - A homology class of $H_K(X_i)$ is **born** when it is not in the image of $H_k(X_{i-1})$.
  - A homology class of $H_K(X_i)$ **dies** when its image is 0, otherwise it **persists**.



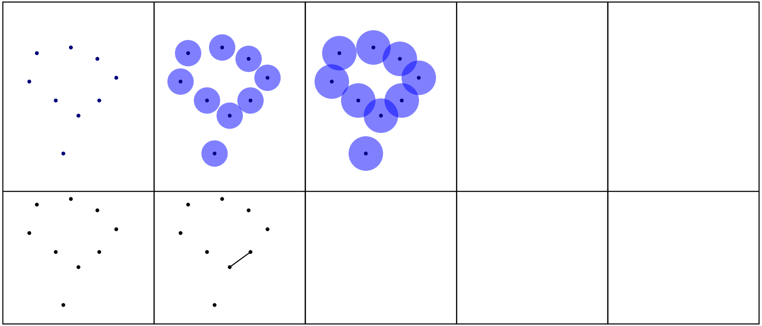


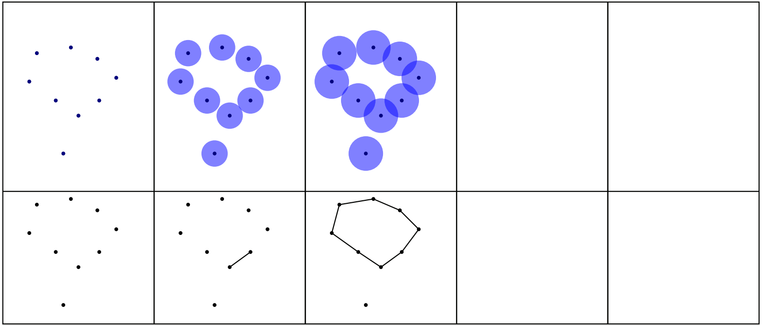


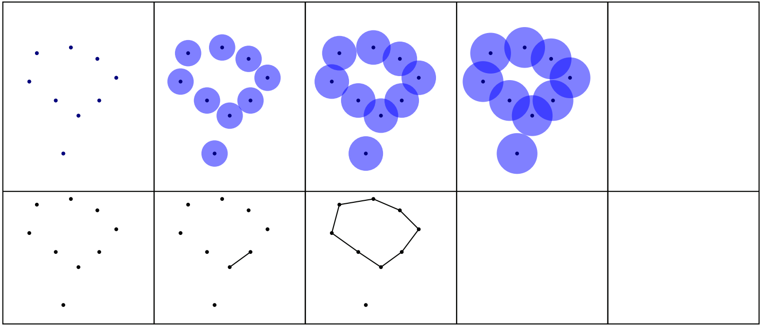


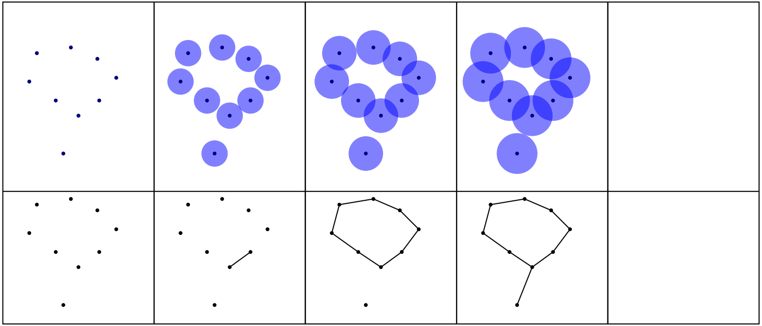
The Datasaurus and Persistent Homology

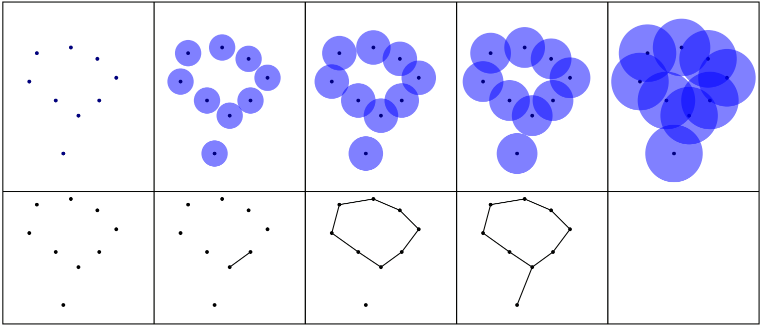
Ryan Jensen

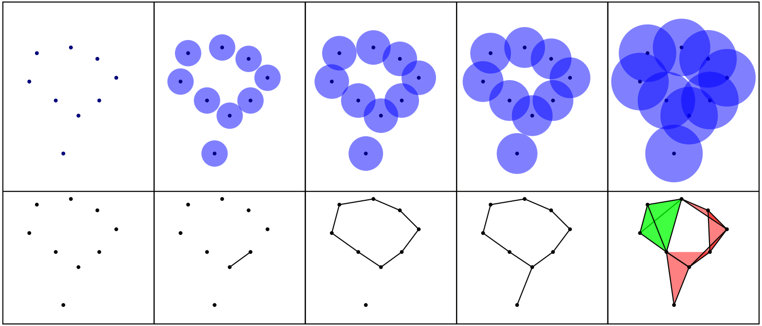


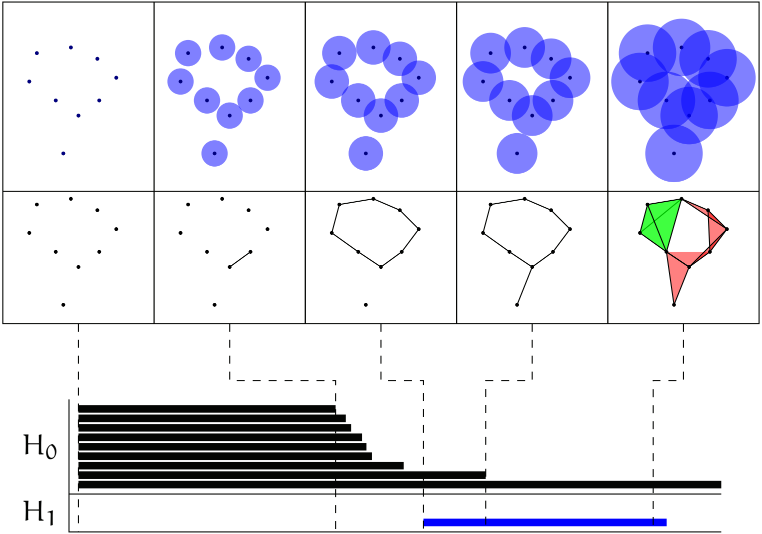











H_0
H_1

The Datasaurus

Visualizing Data

Data visualization is important, but what happens when it is not possible? For example how can high dimensional data be visualized?

The Datasaurus

Visualizing Data

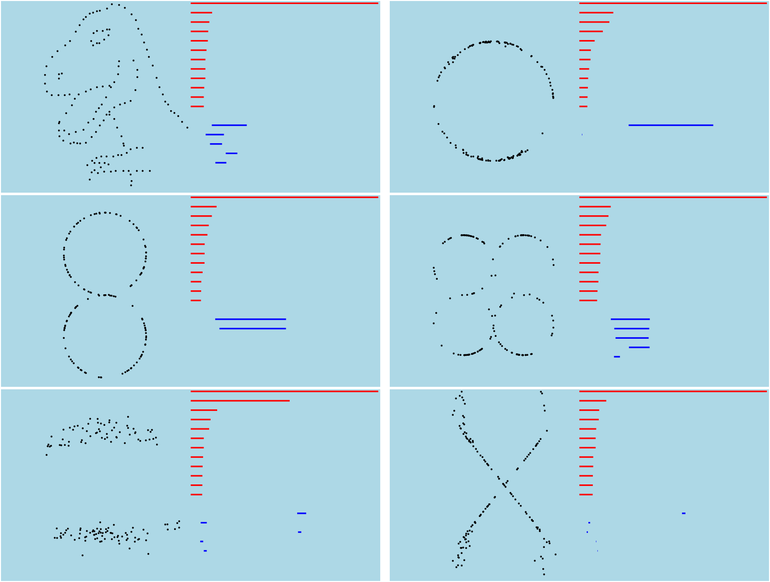
Data visualization is important, but what happens when it is not possible? For example how can high dimensional data be visualized?

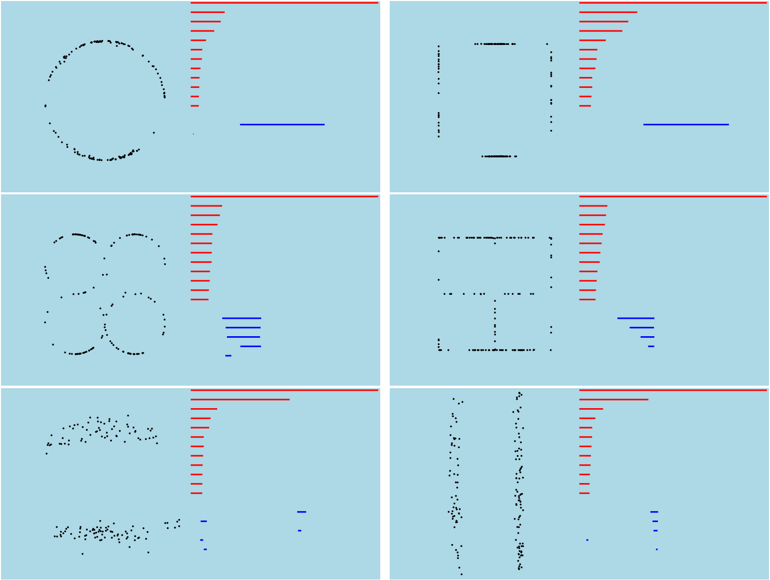
Persistent Homology

We can look at the persistent homology of the morphings and that may give us some idea that they are different.

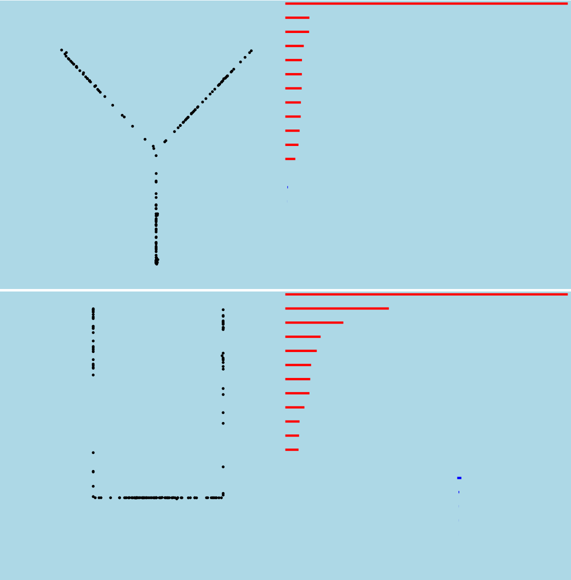
The Datasaurus

Video Link





The Y or The U?



Large Scale Geometry and Topology

Definition: Topology

A **topology** on a set X is a collection $\mathcal{T}$ of subsets of X, called **open set**, so that:

0. $\emptyset, X \in \mathcal{T}$.
1. $U, V \in \mathcal{T}$ implies $U \cap V \in \mathcal{T}$.
2. $\{U_\alpha\}_{\alpha \in A} \subset \mathcal{T}$, implies $\bigcup_{\alpha \in A} U_\alpha \in \mathcal{T}$.

Definition: Topology

A **topology** on a set X is a collection $\mathcal{T}$ of subsets of X, called **open set**, so that:

0. $\emptyset, X \in \mathcal{T}$.
1. $U, V \in \mathcal{T}$ implies $U \cap V \in \mathcal{T}$.
2. $\{U_\alpha\}_{\alpha \in A} \subset \mathcal{T}$, implies $\bigcup_{\alpha \in A} U_\alpha \in \mathcal{T}$.

Definition: Large Scale Structure (Dydak & Hoffland 2006)

A **large scale structure** on a set X is a collection $\mathcal{LS} \neq \emptyset$ of families $\mathcal{B}$ of subsets of X, called **uniformly bounded** so that:

1. $\mathcal{B}_1, \mathcal{B}_2 \in \mathcal{LS}$ implies $\mathrm{St}(\mathcal{B}_1, \mathcal{B}_2) \in \mathcal{LS}$.
2. $\mathcal{B}_1 \in \mathcal{LS}$ implies $\mathcal{B}_2 \in \mathcal{LS}$ if each element of $\mathcal{B}_2$ consisting of more than one point is contained in some element of $\mathcal{B}_1$.

Definition: Star

Let X be a set and $\mathcal{U}$ a collection of subsets of X, and $V \subset X$. The **star** of V with respect to $\mathcal{U}$, $\mathrm{St}(V, \mathcal{U})$, is the union of all elements of $\mathcal{U}$ which intersect V. That is

$$\mathrm{St}(V, \mathcal{U}) = \bigcup_{U \in \mathcal{U},\ U \cap V \neq \emptyset} U.$$

Definition: Star

Let X be a set and $\mathcal{U}$ a collection of subsets of X, and $V \subset X$. The **star** of V with respect to $\mathcal{U}$, $\mathrm{St}(V, \mathcal{U})$, is the union of all elements of $\mathcal{U}$ which intersect V. That is

$$\mathrm{St}(V, \mathcal{U}) = \bigcup_{U \in \mathcal{U},\ U \cap V \neq \emptyset} U.$$

If $\mathcal{V}$ is another collection of subsets of X, then the **star** of $\mathcal{V}$ with respect to $\mathcal{U}$ is

$$\mathrm{St}(\mathcal{V}, \mathcal{U}) = \{\mathrm{St}(V, \mathcal{U}) : V \in \mathcal{V}\}.$$

Definition: Star

Let X be a set and $\mathcal{U}$ a collection of subsets of X, and $V \subset X$. The **star** of V with respect to $\mathcal{U}$, $\mathrm{St}(V, \mathcal{U})$, is the union of all elements of $\mathcal{U}$ which intersect V. That is

$$\mathrm{St}(V, \mathcal{U}) = \bigcup_{U \in \mathcal{U},\ U \cap V \neq \emptyset} U.$$

If $\mathcal{V}$ is another collection of subsets of X, then the **star** of $\mathcal{V}$ with respect to $\mathcal{U}$ is

$$\mathrm{St}(\mathcal{V}, \mathcal{U}) = \{\mathrm{St}(V, \mathcal{U}) : V \in \mathcal{V}\}.$$

The n-**star** of V with respect to $\mathcal{U}$ is defined as

$$\begin{aligned} \mathrm{St}^0(V, \mathcal{U}) &= V \\ \mathrm{St}^n(V, \mathcal{U}) &= \mathrm{St}\left(\mathrm{St}^{n-1}(V, \mathcal{U}), \mathcal{U}\right) \quad \text{for } n \geqslant 1. \end{aligned}$$

Distance via Stars

Staring gives a discrete “distance” (at least enough of a distance for a filtration used in persistent homology). Let $\mathcal{U}$ be a collection of subsets of a space X. For points x and y

$$d_{\mathcal{U}}(x, y) = \text{smallest } n \text{ so that } y \in \mathsf{St}^n(\{x\}, \mathcal{U}),$$

or ∞ if no such n exists.

Distance via Stars

Staring gives a discrete "distance" (at least enough of a distance for a filtration used in persistent homology). Let $\mathcal{U}$ be a collection of subsets of a space X. For points x and y

$$d_{\mathcal{U}}(x,y) = \text{smallest } n \text{ so that } y \in \mathsf{St}^n(\{x\},\mathcal{U}),$$

or ∞ if no such n exists.
If V and W are subsets of X, then

$$d_{\mathcal{U}}(V,W) = \text{smallest } n \text{ so that } W \cap \mathsf{St}^n(V,\mathcal{U}) \neq \emptyset,$$

or ∞ if no such n exits.

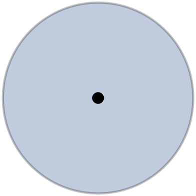
Example: Staring

Let $\mathbf{p}$ be a point in $\mathbb{R}^2$ and $V = \{\mathbf{p}\}$, and $\mathcal{U} = \{B(\mathbf{x}, 1)\}_{\mathbf{x} \in \mathbb{R}^2}$.

Example: Staring

Let $\mathbf{p}$ be a point in $\mathbb{R}^2$ and $V = \{\mathbf{p}\}$, and $\mathcal{U} = \{B(\mathbf{x}, 1)\}_{\mathbf{x} \in \mathbb{R}^2}$. Then

$$\mathrm{St}(V, \mathcal{U}) = B(\mathbf{p}, 2)$$



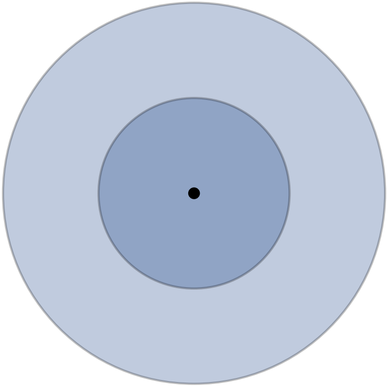
Example: Staring

Let $\mathbf{p}$ be a point in $\mathbb{R}^2$ and $V = \{\mathbf{p}\}$, and $\mathcal{U} = \{B(\mathbf{x}, 1)\}_{\mathbf{x} \in \mathbb{R}^2}$. Then

$$\mathrm{St}(V, \mathcal{U}) = B(\mathbf{p}, 2)$$

and

$$\mathrm{St}^2(V, \mathcal{U}) = B(\mathbf{p}, 4)$$



Large Scale Geometry vs Topology

- Large scale geometry is (almost) the dual of topology.

Large Scale Geometry vs Topology

- Large scale geometry is (almost) the dual of topology.
  0. $\mathcal{LS} \neq \emptyset$ and the fact that point-sets cover a space correspond to $\emptyset, X \in \mathcal{T}$.

Large Scale Geometry vs Topology

- Large scale geometry is (almost) the dual of topology.
  0. $\mathcal{LS} \neq \emptyset$ and the fact that point-sets cover a space correspond to $\emptyset, X \in \mathcal{T}$.
  1. Finite starrings (enlarging) covers corresponds to finite intersections.

Large Scale Geometry vs Topology

- Large scale geometry is (almost) the dual of topology.
  0. $\mathcal{LS} \neq \emptyset$ and the fact that point-sets cover a space correspond to $\emptyset, X \in \mathcal{T}$.
  1. Finite starrings (enlarging) covers corresponds to finite intersections.
  2. Arbitrary refinement corresponds to arbitrary unions.

Large Scale Geometry vs Topology

- Large scale geometry is (almost) the dual of topology.
  0. $\mathcal{LS} \neq \emptyset$ and the fact that point-sets cover a space correspond to $\emptyset, X \in \mathcal{T}$.
  1. Finite starrings (enlarging) covers corresponds to finite intersections.
  2. Arbitrary refinement corresponds to arbitrary unions.
- A metric space induces a large scale structure, just as it induces a topology. For example a large scale structure on $\mathbb{R}^n$ has uniformly bounded families $\mathcal{B}_r$ for $r \geqslant 0$, where

$$\mathcal{B}_r = \{B(\mathbf{x}, r)\}_{\mathbf{x} \in \mathbb{R}^n}.$$

Large Scale Geometry vs Topology

- We think of the families in a large scale structure as covers of a space.

Large Scale Geometry vs Topology

- We think of the families in a large scale structure as covers of a space.
- Topology concerns itself with properties holding at smaller and smaller scales.

Large Scale Geometry vs Topology

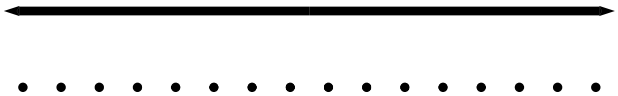
- We think of the families in a large scale structure as covers of a space.
- Topology concerns itself with properties holding at smaller and smaller scales.
- Large scale geometry concerns itself with properties holding at larger and larger scales.

Large Scale Geometry vs Topology

- We think of the families in a large scale structure as covers of a space.
- Topology concerns itself with properties holding at smaller and smaller scales.
- Large scale geometry concerns itself with properties holding at larger and larger scales.
- Topologically $\mathbb{Z}$ and $\mathbb{R}$ are very different.

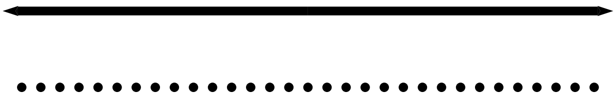
Large Scale Geometry vs Topology

- We think of the families in a large scale structure as covers of a space.
- Topology concerns itself with properties holding at smaller and smaller scales.
- Large scale geometry concerns itself with properties holding at larger and larger scales.
- Topologically $\mathbb{Z}$ and $\mathbb{R}$ are very different.
- In large scale geometry $\mathbb{Z}$ and $\mathbb{R}$ are the same (coarsely equivalent).



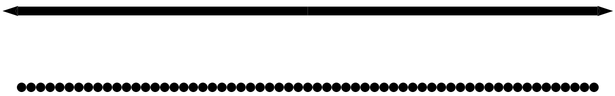
Large Scale Geometry vs Topology

- We think of the families in a large scale structure as covers of a space.
- Topology concerns itself with properties holding at smaller and smaller scales.
- Large scale geometry concerns itself with properties holding at larger and larger scales.
- Topologically $\mathbb{Z}$ and $\mathbb{R}$ are very different.
- In large scale geometry $\mathbb{Z}$ and $\mathbb{R}$ are the same (coarsely equivalent).



Large Scale Geometry vs Topology

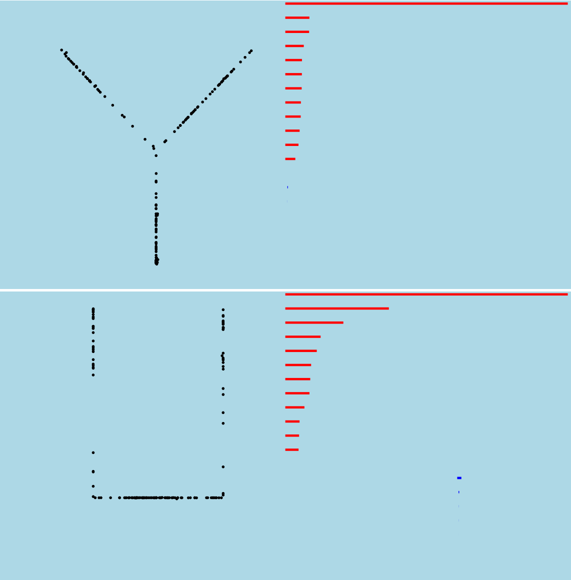
- We think of the families in a large scale structure as covers of a space.
- Topology concerns itself with properties holding at smaller and smaller scales.
- Large scale geometry concerns itself with properties holding at larger and larger scales.
- Topologically $\mathbb{Z}$ and $\mathbb{R}$ are very different.
- In large scale geometry $\mathbb{Z}$ and $\mathbb{R}$ are the same (coarsely equivalent).



Large Scale Geometry vs Topology

- We think of the families in a large scale structure as covers of a space.
- Topology concerns itself with properties holding at smaller and smaller scales.
- Large scale geometry concerns itself with properties holding at larger and larger scales.
- Topologically $\mathbb{Z}$ and $\mathbb{R}$ are very different.
- In large scale geometry $\mathbb{Z}$ and $\mathbb{R}$ are the same (coarsely equivalent).

The Y or The U?



The Y or The U

Gunnar Carlsson developed the following algorithm, called mapper.

The Y or The U?

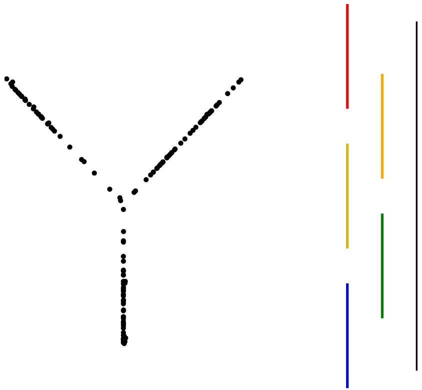


Figure: Height Function

The Y or The U?

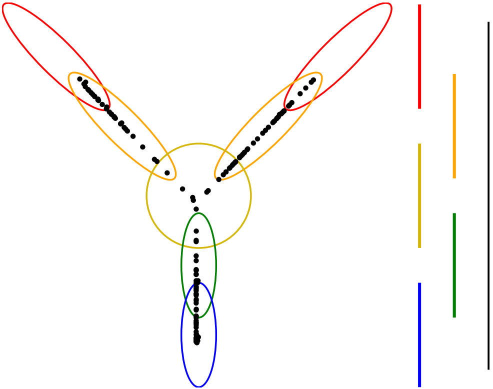


Figure: Clustering

The Y or The U?

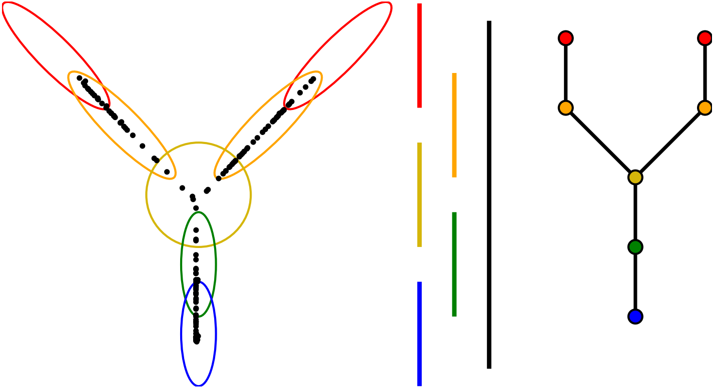


Figure: Graph

The Y or The U?

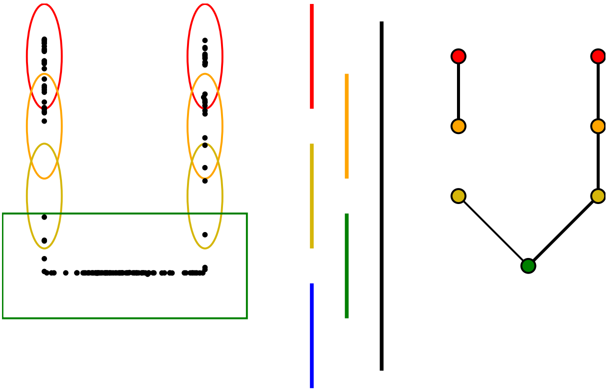


Figure: Graph

The Y or The U?

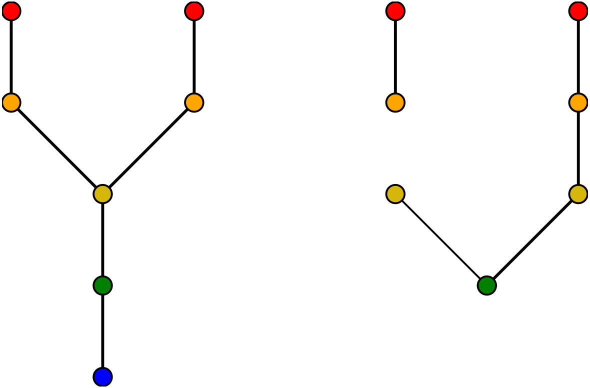


Figure: Graph Comparison

The Y or The U?

An approach using large scale geometry

- A cover need not be generated from a height function, any cover will do.

The Y or The U?

An approach using large scale geometry

- A cover need not be generated from a height function, any cover will do.
- For example take a cover to be closed squares of the form $[i, i+1] \times [j, j+1]$ for $i, j \in \mathbb{Z}$.

The Y or The U?

An approach using large scale geometry

- A cover need not be generated from a height function, any cover will do.
- For example take a cover to be closed squares of the form $[i, i+1] \times [j, j+1]$ for $i, j \in \mathbb{Z}$.
- This amounts to putting the data points on a grid.

The Y or The U?

An approach using large scale geometry

- A cover need not be generated from a height function, any cover will do.
- For example take a cover to be closed squares of the form $[i, i+1] \times [j, j+1]$ for $i, j \in \mathbb{Z}$.
- This amounts to putting the data points on a grid.
- Coarser or finer covers may also be used.

The Y or The U?

An approach using large scale geometry

- A cover need not be generated from a height function, any cover will do.
- For example take a cover to be closed squares of the form $[i, i+1] \times [j, j+1]$ for $i, j \in \mathbb{Z}$.
- This amounts to putting the data points on a grid.
- Coarser or finer covers may also be used.
- In this case there is no need for a clustering algorithm.

The Y or The U?

An approach using large scale geometry

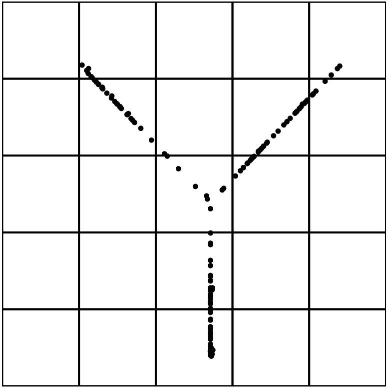
- A cover need not be generated from a height function, any cover will do.
- For example take a cover to be closed squares of the form $[i, i+1] \times [j, j+1]$ for $i, j \in \mathbb{Z}$.
- This amounts to putting the data points on a grid.
- Coarser or finer covers may also be used.
- In this case there is no need for a clustering algorithm.
- A graph can be produced as in the previous example.

The Y or The U?

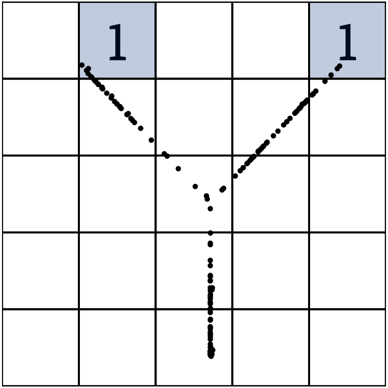
An approach using large scale geometry

- A cover need not be generated from a height function, any cover will do.
- For example take a cover to be closed squares of the form $[i, i+1] \times [j, j+1]$ for $i, j \in \mathbb{Z}$.
- This amounts to putting the data points on a grid.
- Coarser or finer covers may also be used.
- In this case there is no need for a clustering algorithm.
- A graph can be produced as in the previous example.
- Cubical homology can also begin used, with a filtration by sub or super level sets.

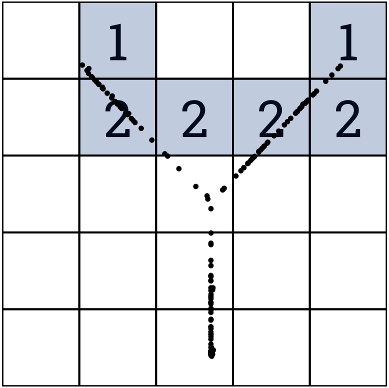
The Y or The U?



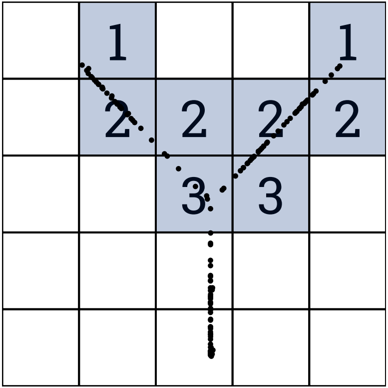
The Y or The U?



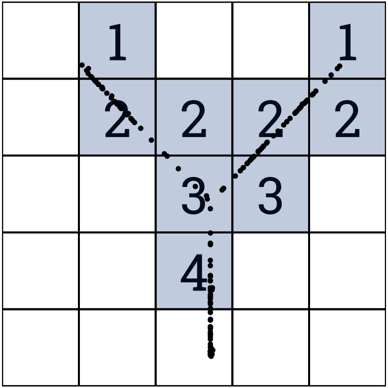
The Y or The U?



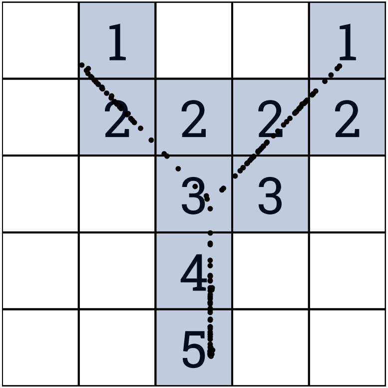
The Y or The U?



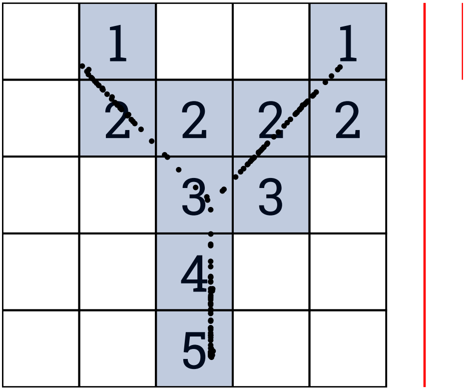
The Y or The U?



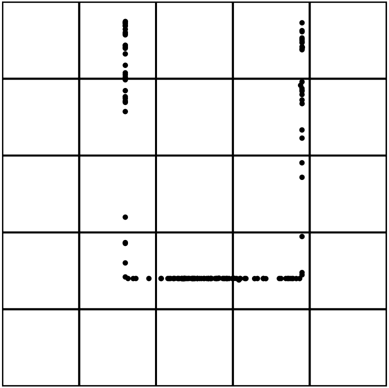
The Y or The U?



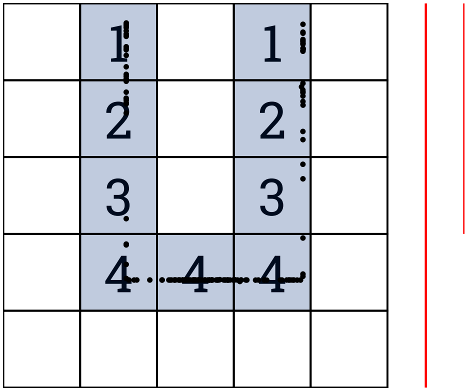
The Y or The U?



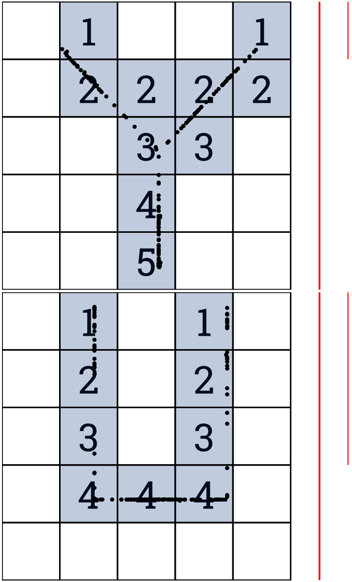
The Y or The U?



The Y or The U?



The Y or The U?



Thank You!