EMT: LOCATING EMPTY TERRITORIES OF HOMOLOGY GROUP GENERATORS IN A DATASET

XIN XU* AND JESSI CISIEWSKI-KEHE

Department of Statistics and Data Science, Yale University
New Haven, CT 06511, USA

Abstract. Persistent homology is a tool within topological data analysis to detect different dimensional holes in a dataset. The boundaries of the empty territories (i.e., holes) are not well-defined and each has multiple representations. The proposed method, Empty Territory (EmT), provides representations of different dimensional holes with a specified level of complexity of the territory boundary. EmT is designed for the setting where persistent homology uses a Vietoris-Rips complex filtration, and works as a post-analysis to refine the hole representation of the persistent homology algorithm. In particular, EmT uses alpha shapes to obtain a special class of representations that captures the empty territories with a complexity determined by the size of the alpha balls. With a fixed complexity, EmT returns the representation that contains the most points within the special class of representations. This method is limited to finding 1D holes in 2D data and 2D holes in 3D data, and is illustrated on simulation datasets of a homogeneous Poisson point process in 2D and a uniform sampling in 3D. Furthermore, the method is applied to a 2D cell tower location geography dataset and 3D Sloan Digital Sky Survey (SDSS) galaxy dataset, where it works well in capturing the empty territories.

1. Introduction. Given a point cloud dataset, e.g., galaxy positions in astronomy or plant locations in forestry, it can be of interest to investigate the geometrical or topological structure of the underlying data. Topological data analysis (TDA) is an approach for learning about the underlying structure of a dataset, and one of the popular tools of TDA is persistent homology [9, 3, 41, 6, 10, 13, 36]. Intuitively, homology groups refer to different dimensional holes in a manifold: the 0th-dimensional homology group ($H_0$) is generated by connected components, the 1st-dimensional homology group ($H_1$) is generated by loops, the 2nd-dimensional homology group ($H_2$) is generated by voids (the boundary of an empty region in 3D, e.g. the boundary of a soccer ball), and so on. For a point cloud dataset, all the data points are separate connected components so the dimension of $H_0$ would be the number of points in the dataset, and there would not be any higher dimensional homology groups. In order to find higher dimensional homology groups in a point cloud dataset, an intermediate structure is needed. There are different approaches for defining this structure, and one method is to sequentially connect points based on their distances from each other. In this process, edges and loops form, along with potentially higher dimensional homology features. For reasons that will be

2010 Mathematics Subject Classification. Primary: 62H30, 55U99; Secondary: 62-07.

Key words and phrases. Alpha shapes, astrostatistics, data segmentation, persistent homology, topological data analysis.

* Corresponding author: Xin Xu.
discussed in more detail later, certain features appear and disappear during this process, and how long a feature exists is called its persistence. Persistent homology is a framework for building a sequence of homology groups from a data set. It provides a graphical summary called a persistence diagram, which summarizes the persistence of each feature that appears and disappears along the sequence.

Persistent homology has been used in many different settings. In [39], $H_1$ generators are used to detect ‘tie-backs’ structures in natural language processing; in particular, the work proposed a similarity filtration with the time skeleton (SIFTS) algorithm, which is a nested sequence of sets that takes text ordering into consideration in its construction. [2] use persistent homology for brain artery trees by analyzing persistences of $H_0$ and $H_1$ generators; principal component analysis and permutation test are used on persistences to understand the influences of age, sex, and artery length on artery tree structures. In [15], they use barcode persistence diagrams, which is another graphical presentation of persistence diagram using a collection of horizontal segments with the segment lengths indicating the feature persistences, to detect periodic structure in dynamical systems and achieved high accuracy in detecting abnormal human lung sounds. [37] introduced molecular topological fingerprint (MTF) as a summary statistic to analyze protein structure, flexibility and folding stability; moreover, an accumulated bar length is proposed to study the structure composition of protein evolution and rigidity.

Persistent homology detects homology group generators and summarizes them in a persistence diagram, but there is not a rigorous framework to locate a desirable representation of the homology group generators back in the original dataset, which can be useful information (e.g., [38]). One major issue with finding representations of homology group generators in original dataset is that there is not a unique representation. For example, in Fig. 1 all the three loops are representations of the same hole and they are equivalent in the sense that they all encircle the same hole. The widely used algorithm of [9] does provide locations of homology group generators as a subset of data points around the hole by tracking cycles that represents various homology groups during the computation. Unfortunately, the output locations are not necessarily a desirable representation of the homology group generators. In Fig. 1, although the three loops are equivalent in representing the hole, they look very different geometrically. In this paper, the goal is to construct a homology group generator representation with a certain complexity (explained later) that outlines the empty region; that is, the region that does not contain any data point inside the representation. The orange loop on the right of Fig. 1 illustrates a possible representation of the hole with a certain level of complexity, while the red and blue loops contain data points inside them. As current algorithms do not constrain the representation of the hole that is returned, we propose a post analysis method called Empty Territory (EmT) to refine the representations of homology group generators reported from the algorithm of [9]. The proposed representation is useful when the goal is to find the truly empty territories. However, this is not necessarily always going to be the desirable representation. Note that requiring emptiness restricts the method to locating $H_1$ generators in 2D data and $H_2$ generators in 3D data since, for example, there is not a general notion of emptiness of an $H_1$ generator in 3D data.

2. Background of persistent homology. Next we introduce some basic ideas of persistent homology; additional details can be found in [11]. Given a point
The three loops are equivalent in the sense that they all encircle the same hole. However, it can be desirable to find a representation that captures the empty territory, with no data point inside. The first two loops contain data points inside them, while the third one is empty. EmT is a method for finding the third type of representation at a specified level of complexity.

cloud dataset, in order to gain meaningful homological structure, an intermediate structure is needed. A common structure used for this purpose is a simplicial complex, which is built to compute the homology on varying scales. A $k$-simplex $\sigma$ is the convex hull of $k + 1$ affinely independent points $X = \{x_0, x_1, \ldots, x_k\} \in \mathbb{R}^d$, denoted as $\sigma = \text{conv}X$. A face of $\sigma$ is a simplex generated by a subset of the set of points that generate $\sigma$, denoted as $\text{conv}S$, where $S \subset X$. A simplicial complex $K$ is a collection of simplices of different dimensions such that, (i) every face of a simplex in $K$ is also in $K$, and (ii) any intersection of two simplices is either empty or a face of both.

Persistent homology computes homology groups on data by building simplicial complexes from the data along a sequence of filtration parameters, which starts at 0 and as it increases, it defines the changing simplicial complexes in the filtration. For example, let the filtration parameter $\delta$ be a threshold for pairwise distances between data points. At the beginning, $\delta = 0$ and there are only the original data points, which are 0-simplexes. As $\delta$ increases, when there is a pair of points whose distance is less than $\delta$, an edge between these two points is added to the simplicial complex. If a set of $k$ points whose pairwise distances are less than $\delta$, the corresponding $k$-simplex, i.e. triangles, polyhedrons, etc, is added to the simplicial complex. Different dimensional holes in the simplicial complex appear and disappear during the process of adding more simplexes into the simplicial complex, e.g. loops formed by edges in the simplicial complex can manifest as $H_1$ generators. Eventually all the data points become interconnected and the whole simplicial complex becomes a single connected component ($H_0$ generator). $H_p$ generators (for some homology dimension $p$) appearing and disappearing at different $\delta$ values during the process can be thought of as having a birth and death at different times in the filtration, with the change of $\delta$ value between its birth and death time being its persistence. One way to interpret a homology group generator that has higher persistence is as topological signal, while a homology group generator with smaller persistence can be considered topological noise [16].

Given a point-set of data, there are several ways to build a simplicial complex [8, 35, 4]. A common approach in TDA is to use the Vietoris-Rips (VR) complex
construction [34], which requires pairwise intersections to form a simplex [18, 11], as was used in the previous example.

**Definition (Vietoris-Rips Complex).** Given a point set \( X = \{x_1, x_2, \ldots, x_n\} \) in \( \mathbb{R}^D \) and a filtration parameter \( \delta > 0 \), let \( VR(X, \delta) \) be the VR complex formed from \( X \) for this \( \delta \). For any \( k \)-point subset, the corresponding \( k \)-simplex \( \sigma_i = \text{conv}\{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\} \) is in \( VR(X, \delta) \) if and only if for every two-point pair in \( \{x_{i_j}, x_{i_k}\} \), \( d(x_{i_j}, x_{i_k}) \leq \delta \), where \( d \) is a distance function.

For example, consider a point cloud dataset \( X = \{a, b, c\} \) in \( \mathbb{R}^2 \) as in Fig. 2a, and a filtration parameter \( \delta > 0 \). First, \( VR(X, \delta) \) contains all 0-simplices: \( a, b \) and \( c \). Next, three circles centered at \( a, b, c \) with radius \( \frac{\delta}{2} \) are drawn in Fig. 2a. As the three circles intersect pairwise, we have \( d(a, b) \leq \delta \), \( d(b, c) \leq \delta \), and \( d(a, c) \leq \delta \). Thus \( VR(X, \delta) \) contains edges \( ab, ac, bc \), and also the triangle \( abc \). Thus \( VR(X, \delta) = \{a, b, c, ab, ac, bc, abc\} \), as displayed in Fig. 2b.

![Figure 2](image-url)

(a) Dataset \( X \): three points (b) \( VR(X, \delta) \)

Different \( \delta \) values result in different \( VR(X, \delta) \) complexes, with \( VR(X, 0) = X \) and \( VR(X, \infty) = \{\text{conv}\ S \subset X\} \). An increasing sequence of \( \delta \): \( 0 < \delta_1 < \delta_2 < \cdots < \delta_k < \infty \), lead to a filtration of complexes:

\[
VR(X, 0) \subset VR(X, \delta_1) \subset VR(X, \delta_2) \subset \cdots \subset VR(X, \delta_k) \subset VR(X, \infty). \tag{1}
\]

A sequence of the \( p \)-dimensional homology groups corresponding to this filtration can be determined as below:

\[
H_p(\text{conv} X) \rightarrow H_p(VR(X, 0)) \rightarrow H_p(VR(X, \delta_1)) \rightarrow H_p(VR(X, \delta_2)) \rightarrow \cdots \rightarrow H_p(VR(X, \infty)). \tag{2}
\]

Along the sequence of homology groups, an \( H_p \) generator has a birth time \( \delta_b \) and a death time \( \delta_d \) if it starts to appear in \( H_p(VR(X, \delta_b)) \) and first disappears in \( H_p(VR(X, \delta_d)) \), denoted as \((b, d, p)\). For all the homology group generators ever present in (2), we have a collection of birth and death times and their corresponding homology group dimension, \((b_1, d_1, p_1), (b_2, d_2, p_2), \ldots, (b_m, d_m, p_m)\), where \( m \) is the total number of generators. These triples can be used to produce a persistence diagram. Fig. 3a displays the same point cloud used in Fig. 1, and Fig. 3b shows...
3. Method. After computing a persistence diagram, it can be useful to consider where the homology group generators appear in the data. However, this goal is not well-defined because there is an equivalence class of representations for each homology group generator (see Fig. 1). The algorithm of [9] can provide a representation for each homology group generator by tracking compositions of simplicial complexes during the computation. The order the data points are input can affect the representation that gets reported. Therefore, it may be desirable to find a representation of homology group generators that is more robust to the order of input data.

The proposed method, EmT, is a post-analysis approach for generating representations and is designed to find the empty regions of homology group generators at a specified level of complexity. EmT uses alpha shapes [12], which is a generalization of the convex hull, in such a way that the inner boundary of a homology group generator can be detected, and maps the boundary to original data points as a representation of the homology group generator. Before introducing our algorithm, we first introduce some necessary concepts and definitions.

3.1. Concave hull and inner shell. In order to find a representation of the hole, it is necessary to determine where the actual empty region is located. This work proposes a framework that uses a concave hull and inner shell to detect the outer and inner empty regions given a collection of data points associated with a homology group generator. For a set of points, the external boundary of the points can be captured by the convex hull of the point set. To obtain a representation of the
hole inside the set of points, an alpha shape is used to separate the inner empty regions from the region outside the collection of points. As noted, an alpha shape is a generalization of the convex hull of a set of points, which is a subgraph of the Delaunay triangulation [12]. Alpha shapes can be intuitively explained by an example using a collection of small balls taking up volume (or area in 2D). Consider the space in $\mathbb{R}^3$ around a set of points. Small balls (or circles in 2D) of a fixed radius fill the volume of the space without enclosing any data point inside the balls. In particular, the balls can fill in any empty space among data points. After filling the space as much as possible, the remaining unoccupied volume is an alpha hull [12]. If we straighten all the round faces carved by spheres to flat faces (or straighten the arcs to segments in 2D), it becomes an alpha shape. A formal definition of an alpha shape is given next [14], where $S$ is a point set in $\mathbb{R}^2$ or $\mathbb{R}^3$.

**Definition.** An $\alpha$-ball is an open ball $b$ with radius $\alpha$, with $\partial b$ representing the boundary of the ball.

**Definition.** A $k$-simplex $\sigma$ ($k = 0, 1, \text{ or } 2$), which is the convex hull of a subset $T \subset S$, is called $\alpha$-exposed if there exists an empty $\alpha$-ball $b$ without any point of $S$ inside, such that $T = \partial b \cap S$.

**Definition.** Let $S_\alpha$ be an alpha shape with radius $\alpha$, then the boundary $\partial S_\alpha$ consists of all the $\alpha$-exposed simplices of $S$.

An example of an alpha hull and alpha shape for the same dataset as in Fig. 1 is displayed in Fig. 4a. The green circles illustrate the $\alpha$-balls occupying the area around the dataset. The remaining part outlined by red curves is the alpha hull, which generates the alpha shape by straightening the arcs, as drawn in blue lines. If we only consider the outer part of the alpha shape, it becomes a *concave hull*. Intuitively, a concave hull is a polygon that encloses all the points with less area than a convex hull, but the concave hull’s perimeter is always longer [17]. There is not a universal definition of the concave hull, and multiple approaches have been proposed to generate concave hulls [25, 1]. Generally, there is a smoothness parameter to adjust the shape of the concave hull, e.g. a value to bound the internal angle degrees of the polygon [1]. We define a concave hull using ideas related to alpha shapes except here the balls cannot arbitrarily fit in any empty space among the data points, but only the external space around the set of points. To determine which region is outside the set, an additional step is needed, which is presented next.

To define the concave hull, we first define an *external point* and an *external path*. An external point, denoted as $p_e$ is defined as a point whose distance to any point in $S$ is larger than $D$, where $D$ is the largest distance between any pair of points in $S$. That is, $p_e$ is at an external location away from any data point in $S$. An external path is a chain of $\alpha$-balls connecting a point to $p_e$, formally defined in below.

**Definition.** For any point $u$, if there exists a set of pairwise intersected $\alpha$-balls (not necessarily with the same radius) denoted as $P = \bigcup_{i=1,\ldots,n_P} b_i$, where $b_i$ is the $i$-th $\alpha$-ball, such that $\{u, p_e\} \subset P$, then $P$ is called an *external path* of $u$.

**Definition.** A $k$-simplex $\sigma$ ($k = 0, 1, \text{ or } 2$), which is the convex hull of a subset $T \subset S$, is $\alpha$-outer-exposed if (i) there exists an empty $\alpha$-ball $b$ without any point of $S$ inside, such that $T = \partial b \cap S$; (ii) there exists a point $u$ in the $\alpha$-ball $b$, and an external path $P$ of $u$, such that $P \cap S_\alpha = \emptyset$, where $S_\alpha$ is the alpha shape of $S$. 
**Definition.** Let $CH_\alpha$ be a concave hull with radius $\alpha$, then the boundary $\partial CH_\alpha$ consists of all the $\alpha$-outer-exposed simplices of $S$.

Fig. 4b shows a concave hull in blue lines. Since the goal is to obtain the representation of a hole, the inner boundary of an alpha shape is of more interest, which can be used to generate a representation of the hole. Similar to the definition of concave hull, we consider the inner part of an alpha shape and define it as an inner shell, as shown in Fig. 4c.

**Definition.** A $k$-simplex ($k = 0, 1, or 2$) $\sigma = \text{conv}T$ of $S$ is $\alpha$-inner-exposed if (i) there exists an empty $\alpha$-ball $b$ without any point of $S$ inside, such that $T = \partial b \cap S$; (ii) for any point $u$ in the $\alpha$-ball $b$, there does not exist an external path $P$ of $u$, such that $P \cap S_\alpha = \emptyset$, where $S_\alpha$ is the alpha shape of $S$.

**Definition.** Let $IS_\alpha$ be an inner shell with radius $\alpha$, then the boundary $\partial IS_\alpha$ consists of all the $\alpha$-inner-exposed simplices of $S$.

As displayed in Fig. 1, a single homology group generator can have multiple topologically equivalent representations. To find a particular representation, we consider a special class of representations for a hole, analogical to the idea of the inner shell. Still using balls to occupy the empty regions, but they only occupy regions from the interior, but do not occupy as much as possible - they can occupy less space than the inner shell. The data points touched by these circles are the resulting representation set. (If the balls do not touch any data point, the representation set is empty.) This defines a particular class of representations for any fixed $\alpha$ value. Any representation within this class has no data point inside and captures the empty region. The parameter $\alpha$ works as a complexity parameter for each representation class, since it controls the radius of $\alpha$-balls and so the maximum number of points in each representation within this class.

**Definition.** Let $R_{\alpha,i}$ be the $i$th representation of a homology group generator with $\alpha$ as radius of $\alpha$-balls, then $R_{\alpha,i}$ consists of (not necessarily all) 0-simplices of $S$, which are $\alpha$-inner-exposed. $R_\alpha = \{R_{\alpha,i}, i = 1, \ldots, n_\alpha\}$, where $n_\alpha$ is the number of possible representations.
Fig. 5 shows four different $R_{\alpha,i}$ with $\alpha$ as 0.8 and 0.6. Fig. 5c and Fig. 5d are representations corresponding to the $I S_{0.8}$ and $I S_{0.6}$: balls take up as much space as possible, and so the resulting representation contains as many points as possible. The vertex set of $I S_{\alpha}$, denoted as $\tilde{I S}_{\alpha}$, is the “largest” representation in the sense that the balls take up the most space and the representation contains the most points.

3.2. Algorithm. As discussed above, for a certain complexity $\alpha$, the representation set derived from $I S_{\alpha}$ has the most points among all the representations in $R_{\alpha}$, denoted as $\tilde{I S}_{\alpha}$. Therefore, $\tilde{I S}_{\alpha}$ is used as the output representation set of the EmT algorithm. For a VR complex, the EmT approach is summarized in Algorithm 1. As mentioned before, persistent homology algorithm of [9] reports data points that are associated with the homology group generators on a persistence diagram, then $I S_{\alpha}$ is obtained based on those locations. Details of the algorithm are contained in Algorithm 1, and summarized next.

Let the group generator location points returned by persistent homology algorithm for any group generator in point cloud $S$ be set $S_X$. $S_X$ does not necessarily
Algorithm 1 EmT Algorithm

Step 1: Build a convex hull on the representation point set reported by persistent homology algorithm of [9], and denote points inside the convex hull as set $S_X'$. 
Step 2: Build an alpha shape $S_\alpha$ on $S_X'$ and get the circle (or sphere in 3D) centers $C_\alpha$ of its corresponding alpha hull.  
Step 3: Keep the ball centers in $C_\alpha$ which are inside $CH_\alpha$ as $C_\alpha'$ and delete the others which are outside $CH_\alpha$. 
Step 4: Select the vertices of arcs (or spheres) touched by circles (or spheres) whose centers are in $C_\alpha'$ to be $\tilde{IS}_\alpha$, the representation points of the hole. 

provide a good representation of the homology group generator, but can be used as a starting point to find a more desirable representation. Build a convex hull on the set $S_X$ and denote it as $convS_X$. Then select all the data points inside $convS_X$ to be set $S_X'$, as shown in Fig. 6a, where the same dataset from Fig. 1 is used to illustrate the EmT algorithm. In Fig. 6a, the red circle points are the set $S_X'$, the green lines are $convS_X$, and the green exes are set $S_X'$. We will only use set $S_X'$ in the following steps.

On the set $S_X'$, build an alpha shape $S_\alpha$ with alpha value $\alpha$, which is the radius of the balls as mentioned above. The $IS_\alpha$ gives a specific boundary shape of the empty region with the complexity level $\alpha$. To obtain the $IS_\alpha$, consider the centers of the balls of the alpha hull, denoted as $C_\alpha$. The green points in Fig. 6b are the set $C_\alpha$.

If a ball is on the inner boundary of the alpha hull, its center is inside $CH_\alpha$. For each point in $C_\alpha$ that is inside $CH_\alpha$, assign it to a set $C_\alpha'$, drawn as the red pluses in Fig. 6c. Let the vertices in $S_X'$ touched by the circles (or spheres) in $C_\alpha'$ be set $\tilde{IS}_\alpha$, drawn as the blue diamonds in Fig. 6d, which is the output of the EmT algorithm.

Remark. $H_1$ generators can have points inside a loop (i.e., a loop is not necessarily empty). The capability of persistent homology detecting the inner points or discovering the correct loop depends on specific situations. Four examples are displayed in Fig. 7, showing the same loop with different point arrangement inside. In Fig. 7a, the raw representation of persistent homology is not able to detect the inner points, while EmT does include the inner points in its representation. In Fig. 7b, both the raw representation and EmT have the inner points in their representations. In Fig. 7c, there are more points inside the loop and the points are more spread out. While the raw representation contains the bigger loop with the inner points, the EmT representation includes two separate loops, as indicated by the corresponding alpha hulls. In Fig. 7d, the raw representation is not able to identify the bigger loop and so the EmT representation is not able to find the bigger loop either because EmT uses the raw representation as input.

Remark. The $\alpha$ can be selected based on specific scenario and the desired complexity of the representation. As a starting point, we suggest using a value between $\frac{\text{birth time}}{2}$ and $3 \times \frac{\text{birth time}}{2}$ (where birth time is the filtration parameter value when this homology group first appears). Note that if the $\alpha$ value is too large, there will not be a hole in alpha shape, and if it is too small a loop may not be found.

4. Simulation examples. To illustrate the proposed EmT, two simulated examples are provided. The first example is in 2D, with observations generated from
Figure 6. The same dataset from previous examples is used here to illustrate EmT. (a) Convex hull of the dataset in green lines. The red circles are set $S_X$ and green crosses are set $S_X'$. (b) The alpha hull in red lines and alpha shape in blue lines. Green circles are $\alpha$ balls and green points are $C_\alpha$. (c) Blue lines are $CH_\alpha$ and red pluses are circle centers that are inside $CH_\alpha$. (d) The comparison between the EmT output $\tilde{IS}_\alpha$ as blue diamonds and the original representation point set $S_X$ returned by persistent homology as red circles.

4.1. 2D example. We generate a dataset from the homogeneous Poisson point process with intensity 300 in $[0,1] \times [0,1]$, shown in Fig. 8. A homogeneous Poisson point process has complete spatial randomness (CSR), which means that the locations of points have no influence on one another [5]. For each $i = 1, \ldots, N$ ($N$ is a homogeneous Poisson point process. There is no true underlying holes since the homogeneous Poisson point process has complete spatial randomness [5], but some holes may appear randomly. EmT is used to locate the boundaries of these random holes. The second example is an uniform distribution in a 3D cube, with two embedded empty regions of different shapes, and EmT is used to recover the boundaries of the empty regions.
from a Poisson distribution), let $X_i(A)$ denote the event that point $i$ is located in region $A$. Under CSR, the random variables $\{X_i(A) : i = 1, \ldots, N\}$ are independent for each region $A$. As the points are completely random, there is no underlying hole in the dataset and the shapes of empty regions are arbitrary. The proposed
Fig. 8. A dataset from the homogeneous Poisson point process with intensity 300 in \([0, 1] \times [0, 1]\). (a) Persistence diagram of the dataset. (b) The eight \(H_1\) generators with the largest persistences reported by persistent homology in different colors. (c) The representations of the eight \(H_1\) generators obtained by the EmT approach, where the boundary of the empty regions are better captured.

4.2. 3D example. In order to investigate the effect of the EmT algorithm in finding empty holes in 3D, an illustrating example in a 3D cube is presented. We first generate a dataset of 1800 points from the uniform distribution in \([0, 1] \times [0, 1] \times [0, 1]\). Then we define one sphere of radius 0.2 and one cube of edge length 0.36 inside \([0, 1] \times [0, 1] \times [0, 1]\), drawn as red regions in Fig. 9a. We delete all the data points inside these two regions and there are 1648 points left. Then we use EmT to see how accurately it detects the boundary shapes of the empty regions. Fig. 9b shows the resulting persistence diagram and Fig. 9c shows the raw representations from the persistent homology algorithm. The representations are displayed in blue and green shading, where the red points are inside the shaded volumes. Similarly, Fig. 9d displays the representations from the EmT approach in blue and green shading, but there is no point inside the shading volumes. We can generally see similar shapes of the EmT representations and the original shapes defined in red in Fig. 9a while the raw representations from the persistent homology algorithm of Fig. 9c do not represent the regions well.

5. Real data examples. In Sec. 4, the examples illustrated EmT in a simulation setting. More examples are needed to illustrate the performance of this approach on realistic datasets. In this section, two real data examples are presented: a cell tower location dataset in 2D and a subset of the Sloan Digital Sky Survey (SDSS) galaxy dataset in 3D. EmT works well in producing the desired representations in both examples.
5.1. Cell tower. A geographical dataset of cell tower locations in Minnesota\(^1\) is used as a 2D example in this section. Each cell tower has a working range affected by many conditions, such as signal frequency and cell tower height. A region on the map with no cell towers potentially suggests limited cell phone signal within this region. We apply the EmT algorithm on the cell tower location dataset and the resulting \(H_1\) generator representations are the areas with potentially limited cell phone signal. The dataset is shown in Fig. 10a with a geographic view generated from Google Maps Static API using R package ggmap \(^7\). The resulting persistence diagram and the raw representations of the eight \(H_1\) generators with the largest persistences reported by the persistent homology algorithm are displayed in Fig. 10c.

\(^{1}\)https://gisdata.mn.gov/dataset/util-fcc
and Fig. 10d, respectively. Fig. 10e shows the representations of the same $H_1$ generators from EmT. In particular, the yellow representation on the right of Fig. 10d has one winding loop containing many cell towers inside. The corresponding representation Fig. 10e has two small loops, each circling a small empty region. This is because EmT is only going to find the empty territories, and so for a certain $\alpha$ value, the alpha shape of this $H_1$ generator contains two holes inside, which results in two small loops. In the bottom-left corner of Fig. 10d, there are two generators (orange and cyan) overlapping with each other. It is because the orange generator forms first and then the smaller cyan loop forms with several edges overlapped with the orange one. In Fig. 10e, the orange representation contains two loops, among which one is exactly the same as the cyan representation. The top-left blue raw representation loop in Fig. 10d contains a larger area than is uncovered by the corresponding representation by EmT in Fig. 10e. The blue raw representation loop throughout the filtration appears to segment into potentially three smaller loops, including the identified green loop. The number of loops found by the EmT algorithm depends on the complexity parameter $\alpha$; if a smaller complexity parameter $\alpha$ was used, the additional empty regions of the blue raw loop could have been uncovered. The corresponding geographic view of the blue loop in Fig. 10e results in an interesting finding: the loop encloses a large portion of the Red Lake Indian Reservation, displayed in Fig. 10b.

5.2. SDSS galaxy dataset. In the universe, regions of low matter density are called cosmic voids. Cosmic voids are considered useful in studying the universe structure and galaxy evolution [27, 29, 28]. Many methods have been developed to detect cosmic voids in the universe [26, 30, 32, 22, 20, 38]. In this section, we use persistent homology to find cosmic voids similarly to the work in [38]. While [38] used a smoothed distance function to construct the filtration and the reported representations are not necessarily empty inside, this work builds the filtration using the VR complex as introduced in Sec. 2 and outputs representations with empty regions inside. The EmT approach is applied to get the representations of cosmic voids detected by persistent homology. The Sloan Digital Sky Survey (SDSS) main galaxy redshift survey\textsuperscript{2} [31] is used. The 3D dataset is recorded in equatorial coordinate system, which is a celestial coordinate system used to describe the positions of celestial objects such as galaxies. It takes the Earth as the origin and projects the Earth’s equator onto the celestial sphere to form the celestial equator. In this work, a subset of the catalog called dim1 [32] is used, with $0.0 < z < 0.05$ ($z$ is redshift), containing 57795 galaxies. Then we transform the dataset into Cartesian coordinates as inputs to the persistent homology algorithm.

When the size of a dataset is large, the VR complex becomes computationally prohibitive, as the computation time grows exponentially with number of observations [40]. In addition to the computational time, large datasets also require substantial memory. For the SDSS dataset, 57795 data points are computationally prohibitive. To mitigate this issue, we consider a k-means clustering approach [24, 19, 23] to reduce the size of the data.

To investigate the impact of k-means clustering on 3D data with a structure similar to the large-scale structure of the Universe, an example is presented to see how well $H_2$ generators are detected as $k$ changes. Fig. 11a displays a Voronoi foam simulation dataset, which is used as an approximation of the universe structures.

\textsuperscript{2}http://www.cosmicvoids.net/documents
A Voronoi foam simulation dataset is generated by a set of randomly sampled void seeds (which specify the locations of the voids) and their Voronoi tessellation; the simulation procedure is explained in detail in [38]. Since the void location is known, the matching of $H_2$ generators to void seeds can be calculated by comparing distances of the $H_2$ generators to each void seed. We use eight void seeds to generate a Voronoi foam dataset of 750 points in a box of $[0, 10] \times [0, 10] \times [0, 10]$. Then we add boundaries on each of the six cube surfaces by uniformly sampling points, 450 points in total, making the whole dataset 1200 points.

The other parameters are $p_{\text{cluster}} = 7\%$, $p_{\text{filament}} = 15\%$, $p_{\text{wall}} = 75\%$, $p_{\text{clutter}} = 3\%$, and $\sigma_{\text{perturb}} = 0.1$ where $p_{\text{cluster}}$, $p_{\text{filament}}$, $p_{\text{wall}}$, and $p_{\text{clutter}}$ are proportions of cluster, filament, wall, and clutter noise points, respectively, associated with different cosmological structures. The $\sigma_{\text{perturb}}$ is the standard deviation of Gaussian noise added to the sampled points.
In Fig. 11a, the eight large red points with blue labels are void seeds and void regions are generated around each of the seed points. The green pluses illustrate one of the eight void regions, which is corresponding to the void seed with label 1. Since we know the locations of the eight void seeds, we can check whether persistent homology\textsuperscript{4} could find each of the eight voids. Using the original 1200 data points, the eight $H_2$ generators with the largest persistences match seven of the eight void regions (there are seven $H_2$ generators that have obviously larger persistence than the other $H_2$ generators). The void region with label 2 in Fig. 11a is not detected, which is smaller in volume than the others.

Next, the numbers of detected voids are compared among different $k$ values for k-means. Consider using k-means clustering with $k$ equal to 5%, 10%, 15%, 20%, 25% and 30% of the total number of points in the original dataset (denoted as $n$). Use only the $k$ centers obtained by k-means clustering as the input for persistent homology. The resulting persistence diagrams are shown in Fig. 12. By checking the eight $H_2$ generators with the largest persistences of each persistence diagram, we can see how the results differ for different $k$ values. Tab. 1 summarizes the performance for each $k$ value. When $\frac{k}{n} = 5\%$, only five of the eight voids are identified, while $\frac{k}{n} = 10\%$ and $15\%$ find six of the eight voids. When $\frac{k}{n} = 20\%$, the same $H_2$ generators are detected as the original dataset; when $\frac{k}{n} = 25\%$ and $\frac{k}{n} = 30\%$, all the eight voids are detected, which is better than the original dataset. This improved performance may be due to the noise-reducing effect of k-means clustering, because the sparse noise points are removed after k-means clustering. From this experiment, it seems a range from 5% to 30% locates most of the voids, therefore any of these values may be reasonable for the SDSS dataset.

Using a $k=5000$, which is approximately 10% of the SDSS dataset, the number of input points for persistent homology is reduced to 5000. Applying persistent homology on the reduced dataset, seven cosmic voids ($H_2$) with the highest persistences

\textsuperscript{4}The standard persistent homology algorithm and the raw representations are used here since we are interested in how the k-means dimension reduction approach affects the persistence diagram.
are selected for EmT to find improved representations. The raw representations reported by the persistent homology algorithm and representations from EmT are displayed in Fig. 13a and Fig. 13b, respectively. Since it is a real dataset with the true \( H_2 \) generators unknown, we cannot obtain any quantitative conclusion. Qualitatively, Fig. 13b appears to have more accurate representations of the empty regions as expected from EmT, while representations in Fig. 13a do not define the empty regions clearly. For example, the cosmic void colored cyan is highlighted in Fig. 13c. The volume inside the raw representation from persistent homology is shaded in blue and galaxies inside the representation are displayed as red points.
In Fig. 13d, the representation from EmT is shaded in blue and there is no galaxy point inside the volume.

6. Concluding remarks. In this work, we proposed EmT, a post-analysis approach for persistent homology to find a particular type of representation of each homology group generator that traces the empty territory. Persistent homology
using a VR complex was introduced Sec. 2. To obtain the EmT representations that capture empty regions, a concave hull and inner shell are used as defined in Sec. 3.1 and motivated by alpha shapes. The concave hull is used to find the external boundary of the alpha shape with a complexity parameter $\alpha$ (i.e., the radius of the ball), while inner shell defines interior empty territory of the alpha shape with the same complexity parameter. With a fixed complexity parameter $\alpha$, a special class of representations using the space-filling balls is considered. Within this class of representations, the vertex set of the corresponding inner shell has the most points and is the output of EmT. The EmT algorithm for finding a desirable representation of $H_1$ generators in 2D data and $H_2$ generators in 3D data is proposed in Sec. 3.2. Two simulation examples are displayed to illustrate the EmT algorithm: a Poisson point process example in 2D and a uniform distribution in a cube in 3D. Furthermore, two real data examples are presented to illustrate EmT: cell tower locations in Minnesota in 2D and SDSS galaxy dataset in 3D.

For large datasets, persistent homology becomes computationally intractable due to both time and memory limits. An approach to deal with large datasets is proposed in this work using $k$-means clustering to reduce the size of a dataset. An illustration example of a Voronoi foam dataset is shown in Sec. 5.2. The performances of $\frac{k}{n}$ as 5%, 10%, 15%, 20%, 25%, and 30% are compared in Tab. 1. Based on this example, a $k$ larger than 5% of the total number of data points is good to use for galaxy datasets.

The proposed algorithm is designed for locating $H_1$ generators in 2D and $H_2$ generators in 3D as alpha shape do not have a counterpart for $H_1$ generators in 3D. Finding, for example, $H_1$ features in 3D point cloud data remains an open problem. Moreover, this work does not consider functional filtrations (e.g. [4] and [16]) of persistent homology, but an approach was proposed in [38]. However, the method of [38] does not find representations that are empty, although using a functional filtration is computationally faster than the VR complex filtration. Moreover, a functional filtration typically requires a grid that loses information about the data in its representation of homology group generators. Though not always the case, it can be desirable to account for small scale features in the data (and sometimes even outliers), which can be done with a VR complex filtration, while a functional filtration generally washes out these features.

Acknowledgments. The authors thank Jisu Kim, Alessandro Rindalo, and Larry Wasserman for helpful discussions in the early stages of this work. The authors thank the Yale Center for Research Computing for guidance and use of the research computing infrastructure.

REFERENCES

[1] S. Asaeedi, F. Didehvar and A. Mohades, $\alpha$-concave hull, a generalization of convex hull, Theoretical Computer Science, 702 (2017), 48–59.
[2] P. Bendich, J. S. Marron, E. Miller, A. Pieloch and S. Skwerer, Persistent homology analysis of brain artery trees, The annals of applied statistics, 10 (2016), 198–218.
[3] G. Carlsson, A. Zomorodian, A. Collins and L. J. Guibas, Persistence barcodes for shapes, International Journal of Shape Modeling, 11 (2005), 149–187.
[4] F. Chazal, B. Fasy, F. Lecci, B. Michel, A. Rinaldo, A. Rinaldo and L. Wasserman, Robust topological inference: Distance to a measure and kernel distance, The Journal of Machine Learning Research, 18 (2017), Paper No. 159, 40 pp.
[5] S. N. Chiu, D. Stoyan, W. S. Kendall and J. Mecke, Stochastic Geometry and Its Applications, John Wiley & Sons, 2013.
[6] D. Cohen-Steiner, H. Edelsbrunner and J. Harer, Stability of persistence diagrams, *Discrete & Computational Geometry*, 37 (2007), 105–120.

[7] D. Kahle and H. Wickham, ggmap: Spatial Visualization with ggplot2, *The R Journal*, 5 (2013), 144–161.

[8] V. De Silva and G. E. Carlsson, Topological estimation using witness complexes., *SPBG*, 4 (2004), 157–166.

[9] Edelsbrunner, Letscher and Zomorodian, Topological persistence and simplification, *Discrete & Computational Geometry*, 28 (2002), 511–533, URL https://doi.org/10.1007/s00454-002-2885-2.

[10] H. Edelsbrunner and J. Harer, Persistent homology-a survey, *Contemporary Mathematics*, 453 (2008), 257–282.

[11] H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction*, American Mathematical Soc., 2010.

[12] H. Edelsbrunner, D. Kirkpatrick and R. Seidel, On the shape of a set of points in the plane, *IEEE Transactions on information theory*, 29 (1983), 551–559.

[13] H. Edelsbrunner and D. Morozov, Persistent homology: Theory and practice, *European Congress of Mathematics*, 31–50, Eur. Math. Soc., Zürich, 2013.

[14] H. Edelsbrunner and E. P. Mücke, Three-dimensional alpha shapes, *ACM Transactions on Graphics (TOG)*, 13 (1994), 43–72.

[15] S. Emrani, T. Gentimis and H. Krim, Persistent homology of delay embeddings and its application to wheeze detection, *IEEE Signal Processing Letters*, 21 (2014), 459–463.

[16] B. T. Fasy, F. Lecci, A. Rinaldo, L. Wasserman, S. Balakrishnan, A. Singh et al., Confidence sets for persistence diagrams, *The Annals of Statistics*, 42 (2014), 2301–2339.

[17] A. Galton, Pareto-optimality of cognitively preferred polygonal hulls for dot patterns, in *Spatial Cognition VI. Learning, Reasoning, and Talking about Space* (eds. C. Freksa, N. S. Newcombe, P. Gärdenfors and S. Wölf), Springer Berlin Heidelberg, Berlin, Heidelberg, 45 (2008), 409–425.

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

[19] J. A. Hartigan and M. A. Wong, Algorithm as 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28 (1979), 100–108.

[20] Y. Hoffman, O. Metuki, G. Yepes, S. Gottlöber, J. E. Forero-Romero, N. I. Libeskind and A. Knebe, A kinematic classification of the cosmic web, *Monthly Notices of the Royal Astronomical Society*, 425 (2012), 2049–2057.

[21] V. Icke, R. Weygaert et al., The galaxy distribution as a voronoi foam, *Quarterly Journal of the Royal Astronomical Society*, 32 (1991), 85.

[22] F.-S. Kitaura and R. E. Angulo, Linearization with cosmological perturbation theory, *Monthly Notices of the Royal Astronomical Society*, 425 (2012), 2443–2454.

[23] S. Lloyd, Least squares quantization in pcm, *IEEE Transactions on Information Theory*, 28 (1982), 129–137.

[24] J. MacQueen et al., Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, Oakland, CA, USA, 1967, 281–297.

[25] A. Moreira and M. Y. Santos, Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points.

[26] M. C. Neyrinck, ZOBOV: A parameter-free void-finding algorithm, *Monthly Notices of the Royal Astronomical Society*, 386 (2008), 2101–2109.

[27] P. Peebles, The void phenomenon, *The Astrophysical Journal*, 557 (2001), 495.

[28] A. Pisani, P. Sutter, N. Hamaus, E. Alizadeh, R. Biswas, B. D. Wandelt and C. M. Hirata, Counting voids to probe dark energy, *Physical Review D*, 92 (2015), 083531.

[29] R. R. Rojas, M. S. VogeleY, F. Hoyle and J. Brinkmann, Photometric properties of void galaxies in the sloan digital sky survey, *The Astrophysical Journal*, 617 (2004), 50.

[30] T. Sousbie, The persistent cosmic web and its filamentary structure–i. theory and implementation, *Monthly Notices of the Royal Astronomical Society*, 414 (2011), 350–383.

[31] M. A. Strauss, D. H. Weinberg, R. H. Lupton, V. K. Narayanan, J. Annis, M. Bernardi, M. Blanton, S. Burles, A. Connolly, J. Dalcanton et al., Spectroscopic target selection in the Sloan Digital Sky Survey: The main galaxy sample, *The Astronomical Journal*, 124 (2002), 1810.
[32] P. Sutter, G. Lavaux, B. D. Wandelt and D. H. Weinberg, A public void catalog from the sdss dr7 galaxy redshift surveys based on the watershed transform, The Astrophysical Journal, 761 (2012), 44.

[33] R. van de Weygaert, Fragmenting the universe. 3: The constructions and statistics of 3-d voronoi tessellations, Astronomy and Astrophysics, 283 (1994), 361–406.

[34] L. Vietoris, Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen, Mathematische Annalen, 97 (1927), 454–472.

[35] H. Wagner, C. Chen and E. Vučini, Efficient computation of persistent homology for cubical data, in Topological Methods in Data Analysis and Visualization II, Springer, 2012, 91–106.

[36] L. Wasserman, Topological data analysis, Annual Review of Statistics and Its Application, 5 (2018), 501–535.

[37] K. Xia and G.-W. Wei, Persistent homology analysis of protein structure, flexibility, and folding, International Journal for Numerical Methods in Biomedical Engineering, 30 (2014), 814–844.

[38] X. Xu, J. Cisewski-Kehe, S. B. Green and D. Nagai, Finding cosmic voids and filament loops using topological data analysis, Astronomy and Computing.

[39] X. Zhu, Persistent homology: An introduction and a new text representation for natural language processing., in IJCAI, 2013, 1953–1959.

[40] A. Zomorodian, Fast construction of the Vietoris-Rips complex, Computers & Graphics, 34 (2010), 263–271.

[41] A. Zomorodian and G. Carlsson, Computing persistent homology, Discrete & Computational Geometry, 33 (2005), 249–274.

E-mail address: xin.xu@yale.edu
E-mail address: jessica.cisewski@yale.edu