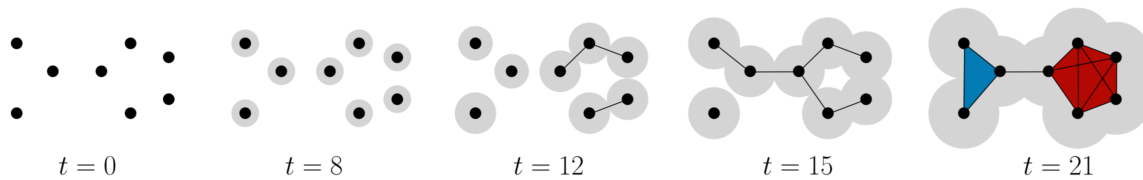

Figure 5: Vietoris–Rips filtration at time values $t \in \{0, 8, 12, 15, 21\}$ for a point cloud P equipped with the Euclidean distance $d = \|\cdot\|_2$ in the plane. For $t < 0$, $\mathrm{VR}_t(P, d)$ is the empty set, as points appear in the filtration at $t = 0$. For $t \in \{12, 15\}$, only edges are added as there are no three vertices with pairwise distances lower than t. For $t = 21$, one triangle and two tetrahedra are added to the filtration. Eventually, for all t after a threshold, $\mathrm{VR}_t(P, d)$ becomes a simplex of dimension equal to the number of points of P minus one.

This allows one to compute a Vietoris–Rips filtration in the same way as for point clouds. We denote the Vietoris–Rips filtration of a weighted graph (G, w_V, w_E) by $\mathrm{VR}(G, w_V, w_E)$, and we denote by $\mathrm{VR}^{k_{\max}}(G, w_V, w_E)$ the dimension-limited version.

Sometimes, weighted graphs are provided only with weights on their edges, that is, a function w_V is not provided. In such cases, there is no canonical way to define a distance between a vertex and itself. One solution is to define w_V for a vertex $v \in V(G)$ as the minimum weight of all its incident edges, and another possible solution is to define w_V as the global minimum weight among all edge weights.

Vietoris–Rips and Čech filtrations add simplices with lower diameter first, that is, the lower the values of t, the lower the diameter of the simplices inside VR_t and $\check{C}_t$. However, it is usual in the graph realm to add edges with higher weights first. Thus, by giving a descending order $e_1, \ldots, e_n$ of the edges by their weights, where $n = |E(G)|$, the edge e_i is added at $t = i - 1$ to the filtration for all $i \in [n]$. In this case, the vertices can be added either at $t = 0$, or at the moment the first incident edge enters the filtration, or at $t = w_V(v)$ if the vertices are weighted by a function w_V that satisfies $w_V(v) \geq w_E(e)$ for all vertices v and all edges incident to v. We can replicate the three approaches also using Vietoris–Rips filtrations. For the first two, we do this by defining the following dissimilarity functions:

$$d^0_\downarrow(v, w) = \begin{cases} 0 & \text{if } v = w, \\ i - 1 & \text{if } \{v, w\} = e_i, \\ +\infty & \text{otherwise,} \end{cases} \quad d_\downarrow(v, w) = \begin{cases} \min\{i : v \in e_i\} & \text{if } v = w, \\ i - 1 & \text{if } \{v, w\} = e_i, \\ +\infty & \text{otherwise.} \end{cases}$$

For the third one, by defining $\mathfrak{S} = E(G) \cup \{\{v\} : v \in V(G)\}$ and $w \colon \mathfrak{S} \to \mathbb{R}$ as

$$w(\{v, w\}) = \begin{cases} w_V(v) & \text{if } |\{v, w\}| = 1, \\ w_E(\{v, w\}) & \text{otherwise,} \end{cases}$$

and giving a descending order $s_1, \ldots, s_{|\mathfrak{S}|}$ of $\mathfrak{S}$ induced by w, we use the dissimilarity function

$$d^V_\downarrow(v, w) = \begin{cases} i - 1 & \text{if } \{v, w\} = s_i, \\ +\infty & \text{if } \{v, w\} \notin \mathfrak{S}. \end{cases}$$

For points (b, d) in persistence diagrams associated with Vietoris–Rips filtrations, b and d are values of the corresponding dissimilarity function for some pair of points in the point cloud. This means that points of Vietoris–Rips persistence diagrams coming from functions $d_\downarrow^0$, $d_\downarrow$, and $d_\downarrow^V$ are tuples of indices of edges in the graph. Therefore, an alternative persistence diagram $D^w(\mathbb{V}_k(\mathrm{VR}(\cdot)))$ containing the weights of the edges can be obtained simply by taking the weight values of the edges associated with the indices in the persistence diagram. That is,

$$D^w(\mathbb{V}_k(\mathrm{VR}(V(G), d))) = \{(w_E(e_{i+1}), w_E(e_{j+1})) : (i, j) \in D(\mathbb{V}_k(\mathrm{VR}(V(G), d)))\} \tag{7}$$

for $d \in \{d_\downarrow^0, d_\downarrow\}$, and

$$D^w(\mathbb{V}_k(\mathrm{VR}(V(G), d_\downarrow^V))) = \{(w(s_{i+1}), w(s_{j+1})) : (i, j) \in D(\mathbb{V}_k(\mathrm{VR}(V(G), d)))\}\,, \tag{8}$$

where we define $w_E(e_\infty) = +\infty = w(s_\infty)$. We refer to this persistence diagram as the *weighted persistence diagram* of the persistence modules of the Vietoris–Rips family of filtrations for graphs induced by $d_\downarrow^0$, $d_\downarrow$ or $d_\downarrow^V$. This also holds for dimension-limited Vietoris–Rips filtrations.

There are many useful filtrations that can be employed in persistent homology for finite sets of points or graphs. In this section, we have presented only some of them. For a broader perspective on point cloud and graph filtrations, we refer the reader to the works by Rieck (2023), by Edelsbrunner and Harer (2022), and by Chazal et al. (2014).

Manifolds also yield persistence diagrams computed from Morse functions. A *Morse function* is a smooth function $f\colon \mathbb{M} \to \mathbb{R}$ on a smooth manifold M such that all critical points are non-degenerate, that is, the Hessian of f at each critical point is non-singular. Similarly to the simplicial case, given $n \in \mathbb{N}$ and a field $\mathbb{F}$, one can build persistence modules from the homology of sublevel sets of a Morse function at different levels $t \in \mathbb{R}$. Thus, the persistence module $\mathbb{V}_n^{\mathbb{F}}(f)$ for a Morse function f on a smooth manifold is defined as

$$V_n^{\mathbb{F}}(f)_t = H_n(f^{-1}(-\infty, t]),$$

where singular homology is meant with coefficients in $\mathbb{F}$. If M is compact, then $\mathbb{V}_n^{\mathbb{F}}(f)$ is of finite type and hence a barcode and a persistence diagram can be extracted from it.

We briefly mention an extension of the usual persistence modules computed with homology that also produce persistence diagrams, known as *zigzag persistence*, introduced by Carlsson and de Silva (2010). Zigzag persistence modules are analogous to persistence modules, but the inclusions are not necessarily given by the order of the real numbers indexing the vector spaces. Zigzag persistence modules share a good amount of properties with ordinary persistence modules, such as their unique representation by persistence diagrams or their stability (Cohen-Steiner et al., 2007; Botnan and Lesnick, 2018), under mild assumptions.

For many purposes, it is useful to have a notion of distance between persistence diagrams. The most frequently used distances between persistence diagrams are the bottleneck distance and the q-Wasserstein distances, for a given $q \in \mathbb{N}$. The *bottleneck distance* between two persistence diagrams D_1 and D_2 is defined as

$$W_\infty(D_1, D_2) = \inf_{\eta\colon D_1^\Delta \to D_2^\Delta} \sup_{x \in D_1^\Delta} \|x - \eta(x)\|_\infty,$$

where D_1^Δ and D_2^Δ denote the persistence diagrams D_1 and D_2 supplemented with their diagonals, that is, $D_1^\Delta = D_1 \cup \Delta$ and $D_2^\Delta = D_2 \cup \Delta$, where $\Delta = \{(x,x) : x \in \bar{\mathbb{R}}\}$, and $\eta\colon D_1^\Delta \to D_2^\Delta$ are bijections of multisets, where points in Δ have countably infinite multiplicity. The *Wasserstein distance* W_q is defined as

$$W_q(D_1, D_2) = \inf_{\eta : D_1^\Delta \to D_2^\Delta} \left(\sum_{x \in D_1^\Delta} \|x - \eta(x)\|_\infty^q \right)^{1/q}$$

although a p-norm $\|\cdot\|_p$ may be used instead of the supremum norm $\|\cdot\|_\infty$. The Wasserstein distances induce norms on persistence diagrams given by the distance to the empty diagram, namely $\|D\|_q = W_q(D, \emptyset)$.

Persistence diagrams are not used directly for data analysis due to their lack of a suitable structure for statistical inference. Instead, persistence summaries are used, which are real-valued functions or vector-valued functions derived from persistence diagrams. Examples of persistence summaries are *total persistence*, which is the sum of the lifetimes of all the non-infinity points of a persistence diagram; persistence landscapes (Bubenik, 2015), and persistence images (Adams et al., 2017), among others. A *landscape* λ of a persistence diagram D is a sequence of piecewise linear functions $(\lambda_1, \lambda_2, \ldots)$ defined as

$$\lambda_i(t) = \max\nolimits^i_{(b,d)\in D} \left\{ f_{(b,d)}(t) \right\}, \quad f_{(b,d)}(t) = \max\left(0, \min\left(b+t, d-t\right)\right),$$

where $\max^i$ returns the i-th maximum value of a multiset.

2.3.2 Mapper and GTDA

Mapper is an algorithm, introduced by Singh et al. (2007), that extracts visual descriptions of high-dimensional datasets in the form of simplicial complexes (usually graphs). One key concept behind Mapper is the nerve of a covering. Given a topological space X and a covering $\mathcal{U} = \{\mathcal{U}_i\}_{i\in I}$ of X, the *nerve* of $\mathcal{U}$ is defined as the simplicial complex $N(\mathcal{U})$ whose set of vertices is I and where a family $\{i_1, \ldots, i_k\} \subseteq I$ is a simplex of $N(\mathcal{U})$ if and only if $\bigcap_{j=1}^k \mathcal{U}_{i_j} \neq \emptyset$. For good coverings (open coverings whose sets and their non-empty finite intersections are contractible), the geometric realization of $\mathcal{N}(\mathcal{U})$ is homotopy equivalent to X. Hence, $\mathcal{N}(\mathcal{U})$ encodes relevant features of the shape of X into a combinatorial object.

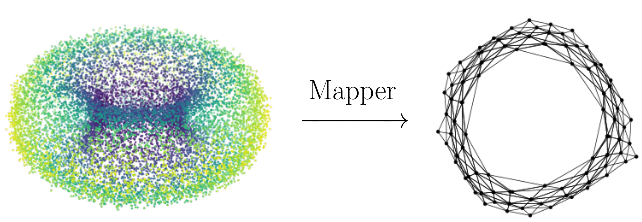

Figure 6: Mapper graph generated from a noisy sample of a torus with 14,400 points using Kepler Mapper (van Veen et al., 2019). The filter function used is given by the projections to the X and Y axes. Covers are taken with an overlap of $p = 0.2$ using 100 squares, by means of the standard cover implementation of the software.