Notes on the Mapper Algorithm

Daniel Edmiston

Abstract

These notes summarize the Mapper algorithm of [Singh et al., 2007] and its analysis as found in [Carrière et al., 2018]. They are to help me organize my thoughts with regard to the algorithm itself, as well as help me acquaint myself with its uses. Note: These notes were written primarily for myself, and may contain errors.

1 Introduction

The Mapper algorithm [Singh et al., 2007] is an algorithm designed for the extraction of global features from high-dimensional data. It enables the simple descriptions of point clouds as simplicial complexes, abstracting away from exact distances/angles, and even individual data points. The output of Mapper is a simplicial complex which provides a compact, global representation of the data. These notes are broken into three parts: mathematical preliminaries, the algorithm itself, and then practical application. In the first part, consisting of sections 2 and 3, I review the preliminary mathematical notions underpinning the algorithm, as well as some material necessary to analyze it. The second part, consisting of Sections 4 and 5, describes the algorithm itself, mostly following the presentation in [Chazal and Michel, 2017], and then summarizes key points in the analysis of [Carrière et al., 2018]. The third part consists of Section 6, and provides a brief example of Mapper in practice.

2 Preliminary Mathematical Definitions: Algorithm

This section lists definitions for the mathematical concepts involved in the Mapper algorithm.

Definition 1. Given a set of $k+1$ affinely independent points $\mathbb{X} = \{x_0, ..., x_k\}$, the *k-dimensional simplex* $\sigma = [x_0, ..., x_k]$ spanned by $\mathbb{X}$ is the convex hull of $\mathbb{X}$. The points $x_i \in \mathbb{X}$ are the *vertices* of σ, and the simplices spanned by the subsets of $\mathbb{X}$ are the *faces* of σ.

Definition 2. Given a vertex set $\mathcal{V}$, the *abstract simplicial complex* $\mathcal{K}$ is a set of finite subsets of $\mathcal{V}$, its simplices. For any $\sigma \in \mathcal{K}$, any subset of σ also belongs

to $\mathcal{K}$. The dimension of an abstract simplicial complex is the dimension of its highest-dimension simplex.[1]

Definition 3. Given subset $\mathcal{A}$ of topological space $(\mathcal{X}, \mathcal{T})$[2], an *open cover* of $\mathcal{A}$ with indexing set I is a subset $\mathcal{U} = \bigcup_{i \in I} U_i, U_i \subseteq \mathcal{T}$, such that $\mathcal{A} \subseteq \mathcal{U}$. That is, an *open cover* of a subset of a topological space is a family of open sets in that topological space such that the subset is contained in the union of those open sets.

Definition 4. Given a cover $\mathcal{U} = \bigcup_{i \in I} U_i$ of some topological space, the *nerve* of $\mathcal{U}$ is the abstract simplicial complex $\mathcal{K}_\mathcal{U}$ whose vertices are the U_i's, and whose simplices are given by $\sigma = [U_{i_0}, ..., U_{i_k}] \in \mathcal{K}_\mathcal{U}$ iff $\bigcap_{j=0}^{k} U_{i_j} \neq \emptyset$. That is, the simplices are the subsets of $\mathcal{U}$ with non-empty intersection.

Definition 5. Given continuous function $f : \mathcal{X} \to \mathbb{R}^d$ and open cover $\mathcal{U} = \bigcup_{i \in I} U_i$ of $\mathbb{R}^d$, the *pull-back cover* of $\mathcal{X}$ with respect to $(f, \mathcal{U})$ is the collection of open sets $(f^{-1}(U_i))_{i \in I}$. The *refined pull-back* is the collection of connected components of the *pull-back cover*.[3]

3 Preliminary Mathematical Definitions: Analysis

This section lists definitions for the mathematical concepts involved in analyzing the Mapper algorithm.

Definition 6. Let $\mathcal{X}$ be a topological space and let $f : \mathcal{X} \to \mathbb{R}$ be a continuous function, here called a *filter*. Define the equivalence relation $\sim_f$ as follows: for all $x, x' \in \mathcal{X}$, $x \sim_f x'$ *iff* x, x' belong to the same connected component of $f^{-1}(y)$ for some $y \in f(\mathcal{X})$. The Reeb graph $R_f(\mathcal{X})$ of $\mathcal{X}$ computed with f is then the quotient space $\mathcal{X}/\sim$ endowed with the quotient topology.[4]

Definition 7. Let f be a continuous real-valued function defined on compact space $\mathcal{X}$[5]. Then f is Morse-type if it meets the following criteria.

[1] Given an abstract simplicial complex, one can embed it in a sufficiently high-dimensional Euclidean space so as to realize a *geometric simplicial complex*, assuming the embedding follows certain constraints. Geometric simplicial complexes are unnecessary for understanding the Mapper algorithm, and as such will not be defined here.

[2] I have other notes which define topological spaces, and many other basic notions from point-set topology.

[3] The refined pull-back cover is a subcover of the pull-back cover, making it also a *refinement* of the pull-back cover in the formal sense, hence its name.

[4] Given topological space $\mathcal{X}$ and equivalence relation $\sim$ on $(\mathcal{X}, \mathcal{T})$, one defines the quotient topology as the pair $(\mathcal{X}', \mathcal{T}')$ where $\mathcal{X}'$ is defined as the set of all equivalence classes $[x] = \{y \in \mathcal{X} \mid x \sim y\}$, and set $U' \in \mathcal{T}'$ *iff* $U \in \mathcal{T}$, where $U = \bigcup_{[x]_i \in U'} [x]_i$, the union of the members of the equivalence classes in U'.

[5] This amounts to saying it is closed and bounded since we're dealing with Euclidean space.

1. There is a finite set $Crit(f) = \{a_1, ..., a_n\}$ called the set of critical values, s.t. over every open interval $(a_0 = -\infty, a_1), ..., (a_i, a_{i+1}), ..., (a_n, a_{n+1} = \infty)$, there is a compact and locally connected space $\mathcal{Y}_i$ and homeomorphism $\mu_i : \mathcal{Y}_i \times (a_i, a_{i+1}) \to f^{-1}((a_i, a_{i+1}))$ s.t. $\forall i = 0, ..., n$, $f|_{f^{-1}((a_i,a_{i+1}))} = \pi_2 \circ \mu_i^{-1}$, where π_2 is projection onto the second factor.

2. $\forall i = 1, ..., n-1$, μ_i extends to a continuous function $\bar{\mu}_i : \mathcal{Y}_i \times [a_i, a_{i+1}] \to f^{-1}([a_i, a_{i+1}])$, and similarly $\bar{\mu}_0 : \mathcal{Y}_0 \times (-\infty, a_1] \to f^{-1}((-\infty, a_1])$ and $\bar{\mu}_n : \mathcal{Y}_n \times [a_n, \infty) \to f^{-1}([a_n, \infty))$

3. Each level set has a finitely generated homology.

NB: This definition alone is highly unintuitive without at least a passing knowledge of Morse theory. In essence, a Morse-type function f is one which allows us to recover topological structure of the space on which it is defined. See `https://www.youtube.com/watch?v=780MJ8JKDqI` for a good crash-course introduction to Morse theory; I will have notes on this in the (near?) future.

Definition 8. Given any graph $\mathcal{G} = (V, E)$ and a function defined on its nodes $f : V \to \mathbb{R}$, the Extended Persistence Diagram $Dg(\mathcal{G}, f)$ is a multi-set of points in $\mathbb{R}^2$ that is computed with extended persistence theory [Oudot, 2015]. Each point in the diagram has one of four specific types: Ord_0, Rel_1, Ext_0^+, Ext_1^-.[6]

Definition 9. Given two (extended) persistence diagrams D, D', the bottleneck distance between them is:

$$W_\infty(D, D') \coloneqq \inf_{\varphi: D \to D'} \sup_{p \in D} \|p - \varphi(p)\|_q$$

where φ ranges over bijections from D to D'.[7]

Using W_∞ to denote bottleneck distance reflects the fact that bottleneck distance is in fact a special case of the Wasserstein distance. Though here the bottleneck distance is defined on diagrams, below I write $W_\infty(X, Y)$, where X, Y are spaces, as a shorthand to denote $W_\infty(Dg(X, f), Dg(Y, g))$.

In words, the bottleneck distance is measured as the maximum distance between matching points (i.e. $p, \varphi(p)$) under the optimal bijection.

Definition 10. A modulus of continuity is a function $\omega : \mathbb{R}^+ \to \mathbb{R}^+$ s.t.

1. $\omega(0) = 0$.

2. ω is monotonically non-decreasing.

3. ω is sub-additive, i.e. $\omega(x + y) \leq \omega(x) + \omega(y)$.

[6] This is not enough background to understand this. In short, extended persistence provides a means of describing (some of the) topological structure of a space by means of points in the plane. An explanation of (extended) persistence will follow in future notes.

[7] To D and D', we add infinitely many points on the diagonal, each with infinite multiplicity; this allows us to define bijections φ when in general D and D' have different cardinalities [Chazal et al., 2016].

4. ω is everywhere continuous.

A modulus of continuity for a function $f : \mathcal{X} \to \mathbb{R}$ is a modulus of continuity which in addition satistifies the following:

$$|f(x) - f(x')| \leq \omega(\|x - x'\|)$$

for any $x, x' \in \mathcal{X}$. This essentially defines an upper bound on how quickly values can change in the co-domain relative to the domain.

Definition 11. Given two metric spaces $(X, \rho_X), (Y, \rho_Y)$, function $f : X \to Y$ is *Lipschitz Continuous* if there exists a real constant $K \geq 0$ s.t. for all $x, x' \in X$,

$$\rho_Y(f(x), f(x')) \leq K\rho_X(x, x')$$

K here is referred to as *a* Lipschitz constant, with the smallest such K being referred to as *the* Lipschitz constant. Such a function f is bounded in how fast values in the co-domain can change relative to values in the domain. Given a Lipschitz constant K for function f, one says that function f is K-Lipschitz, and one can define a modulus of continuity for f by $\omega(\Delta) = K\Delta$, where $\Delta = \|x - x'\|, x, x' \in X$.

Definition 12. Hausdorff distance is a distance metric over non-empty compact subsets of a metric space. It is defined in the following way:

$$d_H(X, Y) = max\{\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\}$$

In words, given subsets non-empty compact subsets $X, Y \subset M$, where M is a metric space with metric d, the Hausdorff distance between X and Y is measured such that it takes the maximum of the largest distances in the sets of closest pairwise distances between X and Y. See [Chazal and Michel, 2017] for a more detailed discussion, as well as intuitive illustrations.

4 Algorithm

The following is the algorithm for Mapper as shown in [Chazal and Michel, 2017].

Input: data set $\mathbb{X}$, function $f : \mathbb{X} \to \mathbb{R}^d$, cover $\mathcal{U} = \bigcup_{i \in \mathcal{I}} U_i$ of image $f(\mathbb{X})$
(1) For each $U \in \mathcal{U}_i$, cluster $f^{-1}(U_i)$ into k_{U_i} clusters $C_{U_{i,1}}, ..., C_{U_{i,k_{U_i}}}$
(2) $C_{U_{i,1}}, ..., C_{U_{i,k_{U_i}}}$ for each $U_i \in \mathcal{U}$ now define a cover of $\mathbb{X}$; calculate the nerve of this cover
Output: Simplicial complex with vertex set v_{U_i} for each cluster C_{U_i}, edge between v_{U_i} and $v_{U'_j}$ *iff* $C_{U_i} \cap C_{U'_j} \neq \emptyset$

Algorithm 1: The Mapper Algorithm

The algorithm as described here is quite succinct. In words, step (1) clusters the preimage of $f^{-1}(U_i) = X_{U_i} \subseteq \mathbb{X}$ for each $U_i \in \mathcal{U}$. It's worth noting that

certain implementations of Mapper give one the option of clustering either the points in the original space or the projected space. This results in k_{U_i} clusters for each U_i, which represent the vertices in the output. Each cluster C_{U_i} represents a connected component in $f^{-1}(U)$ (see definition 5 above). In step (2), if the intersection of any group of clusters is non-empty, the simplex consisting of the representative vertices of these clusters is added to the output simplicial complex. This simplicial complex is the final result of the algorithm, providing a high-level, combinatorial description of the data set.

For the purposes of analysis, [Carrière et al., 2018] define Mapper in a slightly different manner. Here, Mapper is in essence a statistical version of the Reeb graph $R_f(\mathcal{X})$ computed with some filter f. Assume point-cloud $\mathbb{X}_n = \{x_1, ..., x_n\} \subset \mathcal{X}$ with known pairwise distances. Then one computes the Mapper algorithm on $\mathbb{X}_n$ with filter $\hat{f}$ (which is either f or some approximation of f) in the following way.

Input: dataset $\mathbb{X}_n$, function $\hat{f} : \mathbb{X}_n \to \mathbb{R}$, $r \in \mathbb{R}$, $g \in (0, 0.5)$, $\delta \in \mathbb{R}^+$
(1) Compute δ-neighborhood graph on $\mathbb{X}_n$.
(2) Compute $\mathbb{Y}_n = \hat{f}(\mathbb{X}_n)$, the one dimensional image of $\mathbb{X}_n$ under $\hat{f}$.
(3) Define a cover of $\mathbb{Y}_n$ with a set of consecutive intervals $\{I_s\}_{1 \leq s \leq S}$, where each interval is of length r and consecutive intervals overlap with proportion g.
(4) Clustering of $\hat{f}^{-1}(I_s)$ is done by taking the connected components induced by the δ-neighborhood graph. The combined clusterings for each $\hat{f}^{-1}(I_s)$ now define a cover of $\mathbb{X}_n$.
Output: The nerve of the cover defined in (4).

The output of $Mapper(\mathbb{X}_n, \hat{f}, r, g, \delta)$ is then an approximation of $R_f(\mathcal{X})$.

5 Analysis of Mapper

The high-level purpose of [Carrière et al., 2018] is to produce upper bounds on the dissimilarity between a Reeb graph $R_f(\mathcal{X})$ and its Mapper approximation $Mapper(\mathbb{X}_n, \hat{f}, r, g, \delta)$, as well as providing effective parameter choices for r, g, and δ with theoretical guarantees under multiple scenarios.

5.1 Upper Bounds

The means of comparing Reeb graphs with their Mapper approximations is comparing their extended persistence signatures. Specifically, for Morse-type function f on $\mathcal{X}$ we can consider $Dg(R_f(\mathcal{X})) = Dg(R_f(\mathcal{X}), f_R)$, where $f_R : R_f(\mathcal{X}) \to \mathbb{R}$ s.t. $f = f_R \circ \pi$, where π is the quotient map $\mathcal{X} \to R_f(\mathcal{X})$. For a Mapper approximation M_n calculated on $\mathbb{X}_n \subset \mathcal{X}$, we can consider $Dg(M_n) = Dg(M_n, f_{\mathcal{I}})$, where $f_{\mathcal{I}}$ is a function on the nodes of v of graph M_n s.t. if v represents a connected component in the preimage of I_s, then $f_{\mathcal{I}}(v) = mid(\tilde{I}_s)$, where $\tilde{I}_s = I_s \backslash (I_{s-1} \cup I_{s+1})$ and $mid(\tilde{I}_s)$ denotes the midpoint of the interval $\tilde{I}_s$.

The following then is Theorem 7 presented in [Carrière et al., 2018].

Theorem 1. Assume that $\mathcal{X}$ has positive reach rch and convexity radius ρ.[8] Let $\mathbb{X}_n$ be a point-cloud of n points drawn from $\mathcal{X}$, and assume filter f is Morse-type on $\mathcal{X}$, with ω being a modulus of continuity for f. Finally let r, g, and δ be parameters of Mapper as defined above. Then if the following conditions hold:

1. $\delta \leq \frac{1}{4} min\{rch, \rho\}$
2. $max\{|\ f(x) - f(x')| : x, x' \in \mathbb{X}_n, \|x - x'\| \leq \delta\} < gr$
3. $4d_H(\mathcal{X}, \mathbb{X}_n) \leq \delta$

where d_H denotes the Hausdorff distance, then the $Mapper(\mathbb{X}_n, f, r, g, \delta)$ is s.t.

$$W_\infty(R_f(\mathcal{X}), M_n) \leq r + 2\omega(\delta)$$

It stands to reason that approximation error would be constrained by resolution (as modulated by r) and f's regularity (as bounded by ω). Specifically, it is clear that given sufficiently dense sampling, the finer the resolution in the co-domain the better the approximation will be. The same is true of ω, which one will recall represents an upper bound on how quickly values in the co-domain change relative to the domain. The lower this value, the more "well-behaved" f is and the better the approximation will be.

Foruntanately for the sake of analysis, many of the filters commonly used for f in practice are Lipschitz, i.e. one can define for f a modulus of continuity $\omega(\Delta) = K\Delta$ for positive constant K. For example, PCA projections and coordinate filters (i.e. projecting directly onto coordinates) are both 1-Lipschitz.[9]

In many instances in practice, one will have to approximate the filter function. While not the case in, say, coordinate projection, in cases where the filter is estimated from data (e.g. PCA or regression estimators) one gets slightly different theoretical guarantees. In this case, if the following hold:

1. $\delta \leq \frac{1}{4} min\{rch, \rho\}$
2. $max\{max\{|f(x) - f(x')|, |\hat{f}(x) - \hat{f}(x')|\} : x, x' \in \mathbb{X}_n, \|x - x'\| \leq \delta\} < gr$
3. $4d_H(\mathcal{X}, \mathbb{X}_n) \leq \delta$

then the following upper bound holds:

$$W_\infty(R_f(\mathcal{X}), M_n) \leq 2r + 2\omega(\delta) + \underset{x \in \mathbb{X}_n}{max} |f(x) - \hat{f}(x)|$$

Note that the first and third conditions are the same as above when f was known exactly, and the second simply takes the maximum difference of images between true filter f and approximation $\hat{f}$ for points no further away than δ in the domain, stating that this value is less than gr.

[8] These quantities serve to measure the curvedness of a space.

[9] Moduli of continuity for filter functions such as regression estimators are also well defined, if less obvious.

5.2 Parameter Selection

Recall that the mapper algorithm has many parameters, specifically filter $\hat{f}$, interval length r, proportion of overlap g, and neighborhood parameter for graph construction δ. [Carrière et al., 2018] provide means of inferring effective parameters under three scenarios: (i) known filter and generative model, (ii) known filter but unknown generative model, and (iii) unknown filter and unknown generative model.

In practice, we often estimate the filter through some dimensionality reduction algorithm such as PCA, and the generative model is likewise unknown exactly. As such, scenario (iii) is the most pertinent, and I will content myself to explain parameter estimation under this scenario.

At first glance, the prospect of inferring effective parameter choices for δ, r, and g in a principled fashion seems hopeless, as we don't have direct access to the true f. Nonetheless, we may still calculate meaningful parameter estimates if we know the type of function that f is. For instance, if f is PCA, then even though we can only infer f with $\hat{f}$ through a finite dataset, we still know that PCA projectors are 1-Lipschitz, and as such we can define a modulus of continuity for them.

Let $\hat{V}_n(\delta_n) = max\{|\hat{f}(x) - \hat{f}(x')| : x, x' \in \mathbb{X}_n, \|x - x'\| \leq \delta_n\}$, and let ω be a modulus of continuity for f. Then the following provide principled estimates for parameters such that theoretical guarantees exist.

$$g \in (\frac{1}{3}, \frac{1}{2}), \text{The choice of which is arbitrary}$$

$$\delta_n = d_H(\hat{\mathbb{X}}_n^{s_n}, \mathbb{X}_n), \text{ Where } d_H \text{ is Hausdorff distance}$$

$$r_n = \frac{max\{\omega(\delta_n), \hat{V}_n(\delta_n)\}}{g}$$

Here, $\hat{\mathbb{X}}_n^{s_n}$ denotes a sample of size s_n taken from $\mathbb{X}_n$.

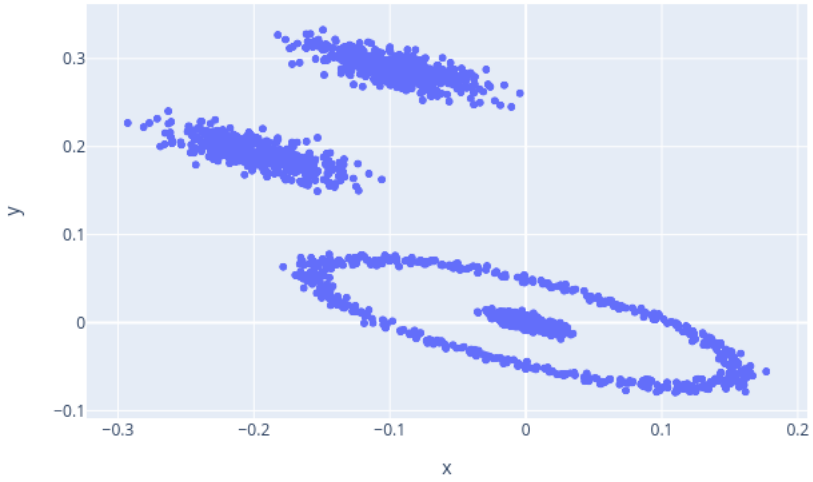
6 Exploring a HD dataset

This section details a brief experiment designed to better acquaint myself with Mapper and one of its implementations. In this experiment, I will generate low-dimensional data of a familiar shape, embed it into a high-dimensional space, and attempt to recapture the shape using Mapper. I will actually be aware of the generative model in this case, but will pretend as if I don't for parameter estimation.

I will try two different ways of estimating parameters. The first will be the principled method of [Carrière et al., 2018], and the second will be by grid-search, where evaluation will take place via comparing persistence diagrams.

6.1 Data Generation

For data generation, I've used *sklearn*'s make-blobs and make-circs functions to generate the data, and used *tadasets*' embed function to embed the data in $\mathbb{R}^2$ into $\mathbb{R}^{100}$. Before embedding, the data can be visualized as follows.

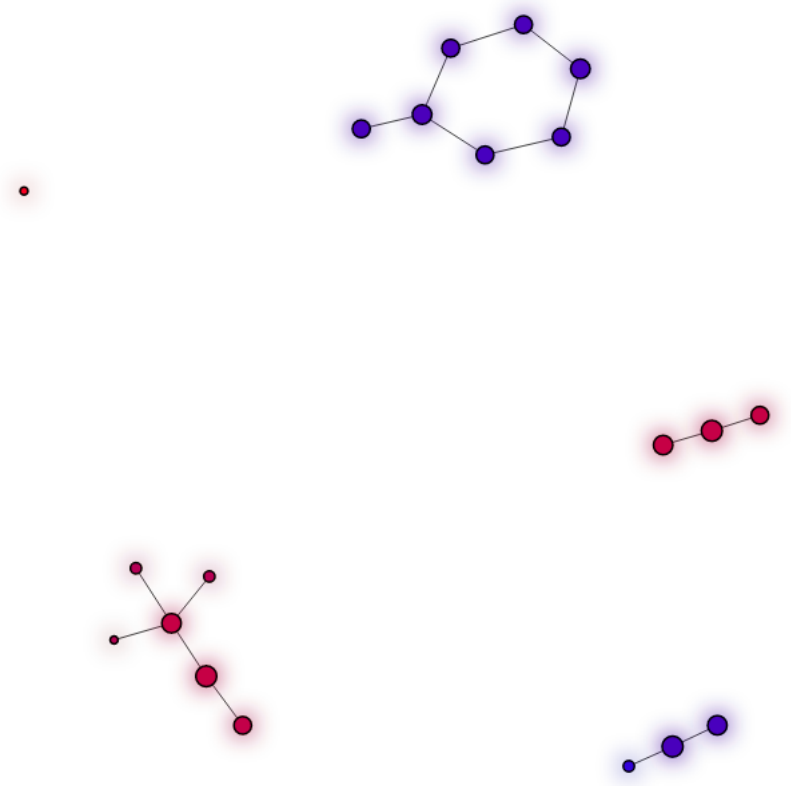


We see that ideally the returned Mapper estimate will have four connected components, with one loop.[10]

6.2 Implementation and Output

To estimate the topology of the dataset, I've estimated the parameters via the method of [Carrière et al., 2018] using the Kepler-Mapper package [van Veen and Saul, 2019]. The following was the result.

[10]There would be two loops if the make-circles data were more spread out.



Here, the data generated by the circles is in blue, and the blobs in red. It can be seen that other than a stray red node, Mapper has roughly inferred the correct topology of the data. There are four large-scale connected components, and one loop.

References

[Carrière et al., 2018] Carrière, M., Michel, B., and Oudot, S. (2018). Statistical analysis and parameter selection for mapper. *The Journal of Machine Learning Research*, 19(1):478–516.

[Chazal et al., 2016] Chazal, F., De Silva, V., Glisse, M., and Oudot, S. (2016). *The structure and stability of persistence modules*. Springer.

[Chazal and Michel, 2017] Chazal, F. and Michel, B. (2017). An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists. arXiv preprint arXiv:1710.04019.

[Oudot, 2015] Oudot, S. Y. (2015). *Persistence theory: from quiver representations to data analysis*, volume 209. American Mathematical Society Providence.

[Singh et al., 2007] Singh, G., Mémoli, F., and Carlsson, G. E. (2007). Topological Methods for the Analysis of High-Dimensional Data Sets and 3D Object Recognition. In *SPBG*, pages 91–100.

[van Veen and Saul, 2019] van Veen, H. J. and Saul, N. (2019). Keplermapper. http://doi.org/10.5281/zenodo.1054444.