Abstract

In topological data analysis (TDA), persistence diagrams have been a successful tool. To compare them, Wasserstein and Bottleneck distances are commonly used. We address the shortcomings of these metrics and show a way to investigate them in a systematic way by introducing bottleneck profiles. This leads to a notion of discrete Prokhorov metrics for persistence diagrams as a generalization of the Bottleneck distance. These metrics satisfy a stability result and can be used to bound Wasserstein metrics from above and from below. We provide algorithms to compute the newly introduced quantities and end with an discussion about experiments.

1 Introduction

The field of topological data analysis (TDA) is becoming a popular tool to study the structure of complex data. One of the major tools of TDA is persistent homology (PH) [EH10]. Its pipeline takes a (often highly complex) point cloud in Euclidean space as input and produces a point cloud in the plane, the persistence diagram (PD), as output. Intuitively, persistence diagrams serve as a summary of the shape of the input data. As a consequence, one can compare different shapes indirectly, by comparing their PDs. The need for a robust and computationally efficient notion of distance for PDs arises. Classically, one uses the Bottleneck and Wasserstein distances to this end [KMN17]. However, the Bottleneck distance only picks up the single biggest difference and the Wasserstein distance is prone to noise, as it picks up every difference no matter how small.

This fact motivates our work to search for new metrics. Starting from the investigation of bottlenecks, we introduce the notion of the bottleneck profile of two PDs, which is a map $\mathbb{R}_{\geq 0} \to \mathbb{N} \cup \{\infty\}$ (Definition 3.1). This tool summarizes metric information at varying scales and generalizes the Bottleneck
distance. Also, the Wasserstein distance can be, in special cases, computed from the bottleneck profile; in general, it can be bounded given a bottleneck profile. The bottleneck profiles arise naturally in a discrete version of the Prokhorov distance, which is a classical tool in probability theory. It turns out that the Bottleneck and the Prokhorov distance are just two instances of a whole family of Prokhorov-style metrics discussed in this paper (Definition 4.1). This family is parameterised by subclass of functions \( f: [0, \infty) \to [0, \infty) \). Not every function \( f \) gives in fact rise to a genuine metric; we examine the conditions on \( f \) in which cases it does (Definition 4.2, such \( f \) are called *admissible*).

**Theorem 4.7.** Fix an admissible function \( f: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \). The discrete \( f \)-Prokhorov metric is an extended pseudometric.

In addition to theoretical development, we discuss algorithms to compute the bottleneck profile and various Prokhorov-type distances.

**Proposition 4.21.** Let \( f: [0, \infty) \to [0, \infty) \) be monotonically increasing. Assume that the values and preimages of \( f \) can be computed in \( O(1) \). Then \( \pi_f(X,Y) \) can be computed in \( O(n^2 \log(n)) \).

We provide a run-time analysis and experiments on a number of data sets. The algorithms are provided as an open source implementation.

## 2 Background

### 2.1 Measure Theory

Let \((X,d)\) be a metric space. It is *complete* if every Cauchy sequence has a limit in \(X\). It is *separable* if it has a countable dense subset. A complete separable metrizable topological space is called a *Polish space*. For example, all Euclidean spaces \( \mathbb{R}^n \) are Polish. Polish spaces are a convenient setting for measure or probability theory.

In general, we endow \(X\) with the Borel \(\Sigma\)-algebra \(\mathcal{B}(M)\) and denote the set of probability measures by \(\mathcal{P}(X)\).

Let us recall an important inequality [Gra14, p. 6]:

**Lemma 2.1** (Chebychev’s inequality). Let \((X, \Sigma, \mu)\) be a measure space and let \(f: X \to \mathbb{R}\) be a measurable function. Then for any \(p > 0\) and \(t > 0\),

\[
\mu\{|f(x)| > t\} \leq \frac{1}{t^p} \int_{|f| \geq t} |f|^p \, d\mu.
\]

**Metrics for Probability Measures**

There are various ways to compare different probability measures.
Definition 2.2. For $p \geq 1$, the $p$-Wasserstein metric is 

$$W_p: \mathcal{P}(X) \times \mathcal{P}(X) \to [0, \infty]$$

$$W_p(\mu, \nu) = \inf_{\gamma} \left( \int d(x, y)^p d\gamma(x, y) \right)^{\frac{1}{p}},$$

where the infimum is taken over all couplings $\gamma$ with marginals $\mu$ and $\nu$.

The 1-Wasserstein metric is also known as Kantorovich metric or earth mover’s distance. The latter name is motivated by the idea of thinking about $\gamma$ as a transport plan for moving a pile of earth $\mu$ into the pile $\nu$. The cost of transportation equals the distance by which the earth is moved.

Intuitively, there are two different ways to “slightly change” a measure. The first one is to move all the mass by a tiny distance. The second one is to move a tiny part of the mass arbitrarily, possibly very far away. While the Wasserstein metric is stable under perturbations of the first kind, small changes of the second kind can result in large differences in the metric. The Prokhorov metric [Pro56] seeks to resolve this problem. It is constructed in such a way that an $\varepsilon$-neighborhood of a measure is characterized as follows: One may move $\varepsilon$ of the mass arbitrarily and the rest by at most $\varepsilon$, see Figure 1 for an illustration. We now formalize

![Illustration of two Prokhorov-close measures which are not Wasserstein-close.](image)

For a Borel set $A \subset X$, the (open) $\varepsilon$-ball around $A$ is

$$A_\varepsilon = \{ x \in X : d(x, A) < \varepsilon \}$$

The Prokhorov metric $\pi$ for two probability measures $\mu, \nu$ is defined as follows:

$$\pi: \mathcal{P}(X) \times \mathcal{P}(X) \to [0, \infty]$$

$$\pi(\mu, \nu) = \inf \{ \varepsilon > 0 : \forall A \in \mathfrak{B}(X) : \mu(A) \leq \nu(A_\varepsilon) + \varepsilon \text{ and } \nu(A) \leq \mu(A_\varepsilon) + \varepsilon \}.$$
By Strassen’s Theorem (cf. Remark 1.29 in [Vil03], Appendix 1.4), an alternative characterization of the Prokhorov metric is given in terms of couplings $\gamma$ which marginalize to $\mu$ and $\nu$ (compare Figure 2),

$$\pi(\mu, \nu) = \inf\{\inf\{\epsilon > 0: \gamma(\{(x, y): d(x, y) > \epsilon\}) < \epsilon\}: \gamma \text{ has marginals } \mu \text{ and } \nu\}. \quad (1)$$

This allows for a discretization suitable for persistence diagrams, see Section 4.

**Example 2.3.** Let $x_1, x_2 \in X$ with $d(x_1, x_2) < 1$ and consider the Dirac measures $\delta_{x_1}, \delta_{x_2}$. We claim that $\pi(\delta_{x_1}, \delta_{x_2}) = d(x_1, x_2)$. The only coupling with correct marginals is $\delta_{(x_1, x_2)}$. Then we have

$$\delta_{(x_1, x_2)}(\{(x, y): d(x, y) > \epsilon\}) = \begin{cases} 1 & \text{if } \epsilon < d(x_1, x_2), \\ 0 & \text{otherwise}; \end{cases}$$

we write $1_{\epsilon < d(x_1, x_2)}$ as a shorthand notation for the right hand side. Consequently, we have

$$\inf\{\epsilon > 0: 1_{\epsilon < d(x_1, x_2)} < \epsilon\} = d(x_1, x_2).$$

![Figure 2: Illustration of two measures $\mu$ and $\nu$ and a coupling $\gamma$ of them.](image)

In their survey [GS02], Gibbs and Su show that the 1-Wasserstein can be related to the Prokhorov metric via

$$\pi^2 \leq W_1 \leq (1 + \text{diam}(X)) \pi,$$
where diam($X$) is the diameter of the underlying space. We provide discrete analogues of this estimate in Propositions 4.11 and 4.16. For more on metrics of probability measures, see the book [Rac91]; references for optimal transport include [PC+19] which takes on a computational perspective.

2.2 Persistent Homology

**Definition 2.4.** The category $\text{PersMod}$ is the functor category

$$\text{PersMod} = \text{Fun}(\mathbb{R}, k\text{-mod}^{fd})$$

from the reals as a poset category to finite dimensional vector spaces. Its objects are called pointwise finite dimensional (p.f.d.) persistence modules. A p.f.d. persistence module $A = (A_t)_{t \in \mathbb{R}}$ comes with transition maps

$$A_{s \leq t} : A_s \to A_t$$

for $s \leq t$.

**Definition 2.5.** An interval module for an interval $J \subset \mathbb{R}$ is a p.f.d persistence module with

$$\mathbb{I}_{J,t} = \begin{cases} k & \text{if } t \in J, \\ 0 & \text{otherwise}, \end{cases}$$

and

$$\mathbb{I}_{J,s} \to \mathbb{I}_{J,t} = \begin{cases} \text{id} & \text{if } s, t \in J, \\ 0 & \text{otherwise}, \end{cases}$$

for $s \leq t$. The start and endpoint of $J$ are referred to as birth time $b(J)$ and death time $d(J)$, respectively. Their difference $d(J) - b(J)$ is called the persistence or lifetime of an interval.

Note that we do not specify whether the endpoints are contained in the interval; they may be $\pm \infty$. Interval modules are of special interest because p.f.d. persistence modules admit an interval decomposition.

**Theorem 2.6** ([Cra15], Theorem 1.1). Let $A \in \text{PersMod}$. Then there exists a collection of intervals $\mathcal{J}$ such that

$$A \cong \bigoplus_{J \in \mathcal{J}} \mathbb{I}_J.$$ 

Such an interval decomposition (sometimes called barcode) can be visualized via a persistence diagram.

**Definition 2.7.** A persistence diagram (PD) is multiset of points in $\mathbb{R}^2$, consisting of

- points above the diagonal $(b, d), b < d$, each with finite multiplicity and
• each point on the diagonal \( \Delta = \{(s, s) \in \mathbb{R}^2\} \) with countable multiplicity.

The convention to include diagonal points with infinite multiplicity will be useful for the construction of distances between persistence diagrams. To obtain a PD from the above interval decomposition, collect the birth and death times of the intervals

\[
\text{Dgm}(A) = \{(b, d) \in (\mathbb{R} \cup \{\infty\})^2 : \langle b, d \rangle \in \mathcal{J}\}
\]

(where the angled brackets indicate that the endpoints may or may not be included): add all the points on the diagonal with countable multiplicities. Off-diagonal points have finite multiplicities since the persistence module is pointwise finite dimensional. We will freely identify off-diagonal points in the diagram with the corresponding interval. Points close to the diagonal have a short lifetime and are often regarded as noise. To compare persistence diagrams, we consider one-to-one correspondences between them. To take care of different cardinalities of off-diagonal points and to get rid of noisy, short-lifetime points, we allow them to be mapped to the diagonal. This explains the inclusion of the diagonal with infinite multiplicity in the above definition.

**Definition 2.8.** A matching \( \eta \) between persistence diagrams \( X \) and \( Y \) is a bijection which fixes all but finitely many diagonal points. The cardinality or size of a matching \( \eta \), denoted by \(|\eta|\) is the number of points which are not fixed.

**Definition 2.9.** The bottleneck distance between two persistence diagrams \( X, Y \) is

\[
W_\infty(X, Y) = \inf_{\eta : X \to Y} \sup \{d(x, \eta(x)) : x \in X\},
\]

where \( \eta \) ranges over all matchings.

**Definition 2.10.** Let \( 1 \leq p < \infty \). The p-Wasserstein distance between two persistence diagrams \( X, Y \) is

\[
W_p(X, Y) = \inf_{\eta : X \to Y} \left( \sum_{x \in X} d(x, \eta(x))^p \right)^{\frac{1}{p}},
\]

where \( \eta \) ranges over all matchings.

The notation of Definition 2.10 has the advantage of being compact, but note that we have uncountably many summands. Although usually, only finitely many, namely \(|\eta|\) for the optimal matching \( \eta \), will be non-zero. Similarly, also only finitely many elements of the uncountable set of which we take the supremum in Definition 2.9 are non-zero.

**Definition 2.11.** Let \( p \geq 1 \). We say a persistence diagram \( X \) has finite pth moment, if the p-Wasserstein distance to the empty diagram is finite: \( W_p(X, \emptyset) < \infty \).
Except from section 4.2, the persistence diagrams in this paper are assumed to have finitely many off-diagonal points. Therefore, the infima in definitions 2.9 and 2.10 are actually minima.

Notice the analogy between Definitions 2.2 and 2.10. We replace probability measures by counting measures and hence turn the integral into a sum. The infimum is taken over all matchings instead of all couplings. This observation will serve as a blueprint for the construction of the discrete Prokhorov metric for persistence diagrams in the Section 4.

The motivation to compare persistence diagrams comes from topological data analysis, where they serve as a summary statistic of topological information.

**Example 2.12.** Given a finite subset $X$ of some metric space, we can consider the Vietoris-Rips complex, cf. [EH10] III.2. This is a filtered simplicial complex; for filtration value $r > 0$ it is given by

$$\text{VR}(X)[r] = \{ S \subset X : \text{diam}(S) < 2r \}.$$

Applying the homology functor gives rise to a persistence module: For $r \leq s$, the inclusion

$$\text{VR}(X)[r] \hookrightarrow \text{VR}(X)[s]$$

induces a module homomorphism

$$H_*(\text{VR}(X)[r]) \rightarrow H_*(\text{VR}(X)[s]).$$

This is called the (Vietoris-Rips) persistent homology of $X$, denoted $\text{PH}_*(X)$.

Summands in its interval decomposition are interpreted as topological features which are “born” at a certain point in the filtration and “persist” for some time. They are regarded to be more significant the longer the intervals are. The following theorem ascertains that this is a useful tool.

**Theorem 2.13 ([Cha+09, Theorem 3.1]).** Let $X, Y$ be finite metric spaces, fix some $k \geq 0$. Then we have

$$W_\infty(\text{Dgm}(\text{PH}_k(X)), \text{Dgm}(\text{PH}_k(Y))) \leq 2d_{\text{GH}}(X, Y),$$

where $d_{\text{GH}}$ is the Gromov-Hausdorff distance [Gro07].

In other words, if we change the input point cloud by $\varepsilon$ in the Gromov-Hausdorff metric, the resulting PDs differ by at most $\varepsilon$ in the Bottleneck distance.

## 3 Bottleneck Profiles

The bottleneck distance $W_\infty$ has a major drawback: It only captures the single most extreme difference between two persistence diagrams. This implies that the same bottleneck distance can be realized by different pairs of persistence diagrams, cf Figure 3. We introduce the notion of the bottleneck profile to address the topic of secondary, tertiary,... bottlenecks and their multiplicities.

---

1Note that [Cha+09] uses a different notion of Gromov-Hausdorff distance, which is equal to two times $d_{\text{GH}}$ of [Gro07].
Definition 3.1. Given two persistence diagrams $X, Y$, define their bottleneck profile to be

$$D_{X,Y} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{N} \cup \{\infty\}, \quad t \mapsto \inf_{\eta : X \rightarrow Y} |\{x : d(x, \eta(x)) > t\}|;$$

where $| \cdot |$ denotes the cardinality of the set.

For $d : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}_{\geq 0}$ we take an $\ell^p$-metric $d(x, y) = \|x - y\|_p$, where the choice of $p$ might depend on the setting. For example, when comparing with the $p$-Wasserstein distance, one might like to choose this same $p$.

Since the infimum is taken over a subset of the natural numbers, it is actually a minimum. To be consistent with the notation in definitions 2.9 and 2.10, we choose to adhere to the use of infimum.

The following observation is immediate:

Lemma 3.2. The bottleneck profile $D_{X,Y}$ is monotonically decreasing.

Proof. Let $\eta : X \rightarrow Y$ be any matching realizing $D_{X,Y}(s)$ for some $s$. Let now $t > s$, then every distance longer than $t$ is in particular longer than $s$ and consequently

$$|\{x : d(x, \eta(x)) > t\}| \leq |\{x : d(x, \eta(x)) > s\}| = D_{X,Y}(s).$$

Taking the infimum over all matchings decreases the left hand side and yields $D_{X,Y}(t)$. □

Knowing this, it is interesting when the bottleneck profile becomes zero.

Lemma 3.3. $D_{X,Y}(t) = 0 \iff t \geq W_\infty(X, Y)$. 
Proof. By definition, the bottleneck distance is the smallest \( t > 0 \) such that there is a matching mapping all points within distance \( t \). In formulas,

\[
W_\infty(X, Y) = \inf \{ t > 0 : \inf_{\eta : X \to Y} |\{ x : d(x, \eta(x)) > t \}| = 0 \}
= \inf \{ t > 0 : D_{X,Y}(t) = 0 \}.
\]

Thus we recover the bottleneck distance from the bottleneck profile. The bottleneck cost of a matching is the longest distance over which two points are matched. Minimizing the bottleneck cost over all matchings yields the bottleneck distance, which we can think of as the primary bottleneck. Similarly, the secondary bottleneck cost of a matching is the second longest distance over which two points are matched. Taking the minimum over all matchings here gives a notion of a secondary bottleneck, which equals \( \inf \{ t > 0 : D_{X,Y}(t) \leq 1 \} \) by an argument analogous to the previous proof. This motivates the name bottleneck profile.

Example 3.4. Let \( X = \{x\} \) and \( Y = \{y\} \) both consist of one point each and assume that \( d(x, y) < d(x, x') + d(y, y') \), where the prime denotes the projection to the diagonal. That means that \( x \mapsto y \) is an optimal matching. Consequently, the bottleneck profile looks as follows:

\[
D_{X,Y}(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq d(x, y), \\ 0 & \text{if } t > d(x, y). \end{cases}
\]

Example 3.5. If we take one of the persistence diagrams to be the empty one, there is only one choice of matching: everything is paired with the diagonal. As a consequence,

\[
D_{X,\emptyset}(t) = |\{ x = (x_1, x_2) : x_2 - x_1 > t \}| = |\{ x = (x_1, x_2) : x_1 + 2t < x_2 \}|.
\]

This is also known as the stable rank function corresponding to the contour \( C(a, \varepsilon) = a + 2\varepsilon \), introduced in [CR20], which counts the bars of \( X \) of length \( > 2t \).

Example 3.6. Consider some particular simple persistence diagrams. The first three parts of Figure 4 each show a base diagram ("Diagram X", in blue) with four points and perturbations of it: The orange diagram ("Diagram Y") in the first image is obtained by shifting the blue one by three. The green diagram ("Diagram Z") shifts the top point of \( X \) by three, the next point by two, the third by one and leaves the lowest point unchanged. For the yellow diagram ("Diagram W") in the third image, we only shift two points from \( X \) by three and leave the other two untouched. Clearly, the bottleneck distance between the base diagram and each of the shifted versions is three. But the amount of shifted points is reflected in the bottleneck profile: While \( D_{X,Y}(t) \) is four, \( D_{X,Z}(t) \) is two (i.e. the multiplicity of the bottleneck) for \( 0 < t < 3 \). And \( D_{X,W} \) displays more steps, reflecting the fact that there are secondary and tertiary bottlenecks.
Figure 4: The PD $X$ has bottleneck distance 3 to each of the PDs $Y, Z, W$ (first three images). However, it is attained with different multiplicities, which one can read off from the bottleneck profile (right-most image).

Note that the function $D$ enjoys some properties reminiscent of a metric (hence the notation $D$): It is obviously symmetric. The triangle inequality does not hold pointwise but in a scaled version, that is:

**Lemma 3.7.** For all persistence diagrams $X, Y, Z$ and all real numbers $s, t \geq 0$, $D_{X, Z}(s + t) \leq D_{X, Y}(s) + D_{Y, Z}(t)$.

**Proof.** This follows from the triangle inequality on $\mathbb{R}^2$. Fix $s, t \geq 0$, let $\eta_{X, Y}: X \to Y$ and $\eta_{Y, Z}: Y \to Z$ denote optimal matchings realizing $D_{X, Y}(s)$ and $D_{Y, Z}(t)$, respectively. Let $\eta = \eta_{Y, Z} \circ \eta_{X, Y}: X \to Z$ be the matching obtained by composition. It suffices to show that

$$|\{x: d(x, \eta(x)) > s + t\}| \leq |\{x: d(x, \eta_{X, Y}(x)) > s\}| + |\{y: d(y, \eta_{Y, Z}(y)) > t\}|,$$

because the left hand side only decreases if we take the infimum over all matchings. Hence we have to investigate what happens when a point $x$ is matched to $\eta(x)$ which is farther apart than $s + t$. Note that $\eta(x) = \eta_{Y, Z}(\eta_{X, Y}(x))$, so we compare the distances of the matched points using the triangle inequality,

$$s + t < d(x, \eta(x)) \leq d(x, \eta_{X, Y}(x)) + d(\eta_{X, Y}(x), \eta(x)).$$

Therefore, it cannot be that both $d(x, \eta_{X, Y}(x)) \leq s$ and $d(\eta_{X, Y}(x), \eta(x)) \leq t$ (compare Figure 5). That means, we have $d(x, \eta_{X, Y}(x)) > s$ or $d(\eta_{X, Y}(x), \eta(x)) > t$ or both. Using the principle of inclusion-exclusion, conclude

$$|\{x: d(x, \eta(x)) > s + t\}| = |\{x: d(x, \eta_{X, Y}(x)) > s\}| + |\{y: d(y, \eta_{Y, Z}(y)) > t\}|$$

$$ - |\{x \in: d(x, \eta_{X, Y}(x)) > s \text{ and } d(\eta_{X, Y}(x), \eta_{Y, Z}(\eta_{X, Y}(x))) > t\}|$$

$$\leq |\{x: d(x, \eta_{X, Y}(x)) > s\}| + |\{y: d(y, \eta_{Y, Z}(y)) > t\}|.$$
Note that $D_{X,Y}(t) = 0$ for all $t > 0$ implies $X = Y$ only under some finiteness assumptions. For example, consider a converging sequence $(a_n)_{n \in \mathbb{N}} \subset \mathbb{R}^2$ above the diagonal with limit $a \notin (a_n)$, which is also above the diagonal. Set $X$ to consist of all elements of the sequence $\{a_n: n \in \mathbb{N}\}$. Set $Y$ to be $X \cup \{a\}$. Then for all $\varepsilon > 0$ there exists $\eta: X \to Y$ such that $d(x, \eta(x)) < \varepsilon$ for all $x \in X$. Therefore, $D_{X,Y}(t) = 0$ for every $t > 0$, but $X \neq Y$.

Following [Blu+14], we denote by $\bar{\mathcal{B}}$ the set of persistence diagrams such that for each $\varepsilon > 0$ there are finitely many points of persistence $> \varepsilon$. The next lemma is an immediate consequence of [Blu+14, Lemma 3.4].

**Lemma 3.8.** The bottleneck profile satisfies $D_{X,X}(t) = 0$ for all persistence diagrams $X$ and $t > 0$. Moreover, $D_{X,Y}(t) = 0$ for all $t > 0$ implies $X = Y$ for $X,Y \in \bar{\mathcal{B}}$.

**Proof.** If $D_{X,Y}(t) = 0$ for all $t > 0$, then $W_\infty(X,Y) = 0$ by Lemma 3.3. Now for $X,Y \in \bar{\mathcal{B}}$, this only happens if $X = Y$ by [Blu+14, Lemma 3.4].

### 3.1 Relation to Wasserstein distances

We have already seen how the bottleneck profile is related to the bottleneck distance. This is actually part of a more general result comparing it to $p$-Wasserstein metrics.

**Lemma 3.9.** Let $X,Y$ be two persistence diagrams, and let $p > 0$. Then

$$D_{X,Y}(t) \leq \frac{1}{tp} W_p(X,Y)^p.$$ 

**Proof.** This follows from the Chebychev inequality (Lemma 2.1) for counting
measures. To spell out the details, estimate that for every bijection $\eta$

$$\left| \{ x : d(x, \eta(x)) > t \} \right| = \sum_{\{ x : d(x, \eta(x)) > t \}} 1 \leq \sum_{\{ x : d(x, \eta(x)) > t \}} \frac{d(x, \eta(x))^p}{t^p} \leq \sum_{x \in X} \frac{d(x, \eta(x))^p}{t^p} = \frac{1}{t^p} \sum_{x \in X} d(x, \eta(x))^p.$$ 

Now choosing $\eta$ to minimize the right hand side, we have by definition of the Wasserstein distance an estimate for $D_{X,Y}$:

$$D_{X,Y}(t) \leq \left| \{ x : d(x, \eta(x)) > t \} \right| \leq \frac{1}{t^p} W_p(X,Y)^p.$$ 

This is illustrated by Figure 6. Note that we recover Lemma 3.3 in the limit for

\[ D_{X,Y}(t) \leq \left| \{ x : d(x, \eta(x)) > t \} \right| \leq \frac{1}{t^p} W_p(X,Y)^p. \]

\[ \square \]

This is illustrated by Figure 6. Note that we recover Lemma 3.3 in the limit for

\[ D_{X,Y}(t) \leq \left| \{ x : d(x, \eta(x)) > t \} \right| \leq \frac{1}{t^p} W_p(X,Y)^p. \]

Figure 6: An example for the relation between $D_{X,Y}$ and the Wasserstein distance.
Figure 7: Illustrating the proof of Lemma 3.10: Decomposing the area under the graph into rectangles.

Proposition 3.11. If the bottleneck profile $D_{X,Y}(t)$ can be realized by the same matching $\eta$ for all $t > 0$, then $\eta$ realizes $W_1(X,Y)$.
Proof. If \( \eta \) realizes \( D_{X,Y}(t) \) for all \( t > 0 \), then the inequality in the proof of the previous lemma becomes an equality

\[
\int_0^\infty D_{X,Y}(t)dt = \int_0^\infty |\{x: d(x, \eta(x)) > t\}|dt = \sum_{x \in X} d(x, \eta(x)) \overset{(*)}{=} W_1(X,Y).
\]

Combining this with Lemma 3.10, we obtain

\[
\int_0^\infty D_{X,Y}(t)dt = W_1(X,Y).
\]

Consequently, the inequality (\( \ast \)) is actually an equality, which is what we wanted to prove.

\[\square\]

3.2 Algorithms

Recall the definition

\[
D_{X,Y}(t) = \inf_{\eta} |\{x: d(x, \eta(x)) > t\}|,
\]

and let \( \eta \) be the matching realizing the infimum. Then \( \eta \) also realizes the following supremum:

\[
\sup_{\eta} |\{x: d(x, \eta(x)) \leq t\}|,
\]

and consequently

\[
D_{X,Y}(t) = |\eta| - \sup_{\eta} |\{x: d(x, \eta(x)) \leq t\}|.
\]

Here, \(|\eta|\) denotes the number of matched pairs which involve at least one off-diagonal point. The computation of \(\sup_{\eta} |\{x: d(x, \eta(x)) \leq t\}|\) is a version of the unweighted maximum cardinality bipartite matching problem. First, set up the following notation (following [EH10, chapter VIII.4]). Denote by \( X_0 \) the off-diagonal points of \( X \) and by \( X_0' \) their projections to the diagonal (and analogously for \( Y \)).w Set \( U = X_0 \cup Y_0' \) and \( V = Y_0 \cup X_0' \) and consider the bipartite graph \( G = (U \cup V, E) \) with \( e = \{u, v\} \in E \) if either of the following holds:

- \( u \in X_0, v \in Y_0 \) and \( d(u, v) \leq t \),
- \( u \in X_0, v \in X_0' \) is its projection to the diagonal and \( d(u, v) \leq t \),
- \( v \in Y_0, u \in Y_0' \) is its projection to the diagonal and \( d(u, v) \leq t \),
- \( u \in Y_0' \) and \( v \in X_0' \).

Let \( M \subset E \) be a matching of maximal cardinality. Observe that such a matching corresponds to a bijection \( \eta: X \to Y \) maximizing \( |\{x: d(x, \eta(x)) \leq t\}|\).
To estimate the run-time of this algorithm, let $n = |X| + |Y|$. We solve the unweighted maximum cardinality bipartite matching problem using the Hopcroft-Karp algorithm [HK73]. Let us briefly recall this classical algorithm. The algorithm extends a partial matching $M$ until it reaches a maximum one. It achieves this by augmenting paths: A path $p$ that starts at an unmatched vertex in $U$ and ending at an unmatched vertex in $V$ such that edges from $U$ to $V$ are not in $M$ but edges from $V$ to $U$ are. Removing edges from $p \cap M$ from the matching and instead inserting edges from $p \cap (E \setminus M)$ increases the size of $M$ by one. The Hopcroft-Karp algorithm finds vertex-disjoint augmenting paths in $O(n^2)$ via the so-called layer subgraph, which is constructed via a depth-first search in $O(n^2)$. After extending the matching using all these augmenting paths, the algorithm starts over. The algorithm terminates after $O(\sqrt{n})$ of these iterations.

While this consequently takes $O(n^{2.5})$ in the worst case, we perform a variant which exploits the geometric nature of the setting, as suggested in [EIK01]. Instead of building the layer graph explicitly, one can use a geometric data structure that allows for querying neighbors within a given distance, as well as removing points. Following [KMN17], k-d trees achieve this requiring $O(\sqrt{n})$ for either of the two operations. Consequently, as noted by [KMN17] and [EIK01], our variant of the Hopcroft-Karp algorithm runs in $O(n^2)$. Summarizing, we find the following:

**Proposition 3.12.** Let $X, Y$ be finite persistence diagrams and denote $n = |X| + |Y|$. The value of the bottleneck profile at $t$, $D_{X,Y}(t)$, can be computed in $O(n^2)$.

**Remark 3.13.** Using k-d trees is useful in practice, but does not yield optimal theoretical run-times. Indeed, the more sophisticated data structure from [EIK01], Section 5.1, can be constructed in $O(n \log(n))$. The two relevant operations on it require $O(\log(n))$, so that the bottleneck profile could be evaluated in $O(n^{1.5} \log(n))$ using this method.

**Remark 3.14.** Instead of using Hopcroft-Karp, one can regard the matching problem as a linear program. For each $x \in X$ and $y \in Y$, we have a binary variable $f_{xy}$ indicating whether the edge from $x$ to $y$ is in the matching. The coefficients (the cost of the edge) are given by

$$c_{xy} = \begin{cases} 1 & \text{if } d(x, y) > t, \\ 0 & \text{otherwise.} \end{cases}$$

The objective is

$$\minimize \sum_{x,y} c_{xy} f_{xy}$$

subject to $\forall x \in X$: $\sum_y f_{xy} = 1$, $\forall y \in Y$: $\sum_x f_{xy} = 1$. 

15
4 Discrete Prokhorov Metrics for Persistence Diagrams

A straight-forward discretization of the coupling characterization of the probabilistic Prokhorov metric (1) gives the main notion of this section.

Definition 4.1. Given two persistence diagrams \( X, Y \), consider matchings \( \eta: X \to Y \) to define their Prokhorov distance as

\[
\pi(X, Y) = \inf\{t > 0: D_{X, Y}(t) < t\} = \inf\{t > 0: \inf_{\eta: X \to Y} |\{x: d(x, \eta(x)) > t\}| < t\}.
\]

Informally, we look at the intersection of the bottleneck profile with the diagonal. Similarly, we have already seen that the bottleneck distance arises as the intersection of \( D_{X, Y} \) with the horizontal axis. This motivates the question, what functions we can intersect the bottleneck profile with to obtain a sensible notion of distance.

Definition 4.2. Consider a function \( f: [0, \infty) \to [0, \infty] \). We say \( f \) is superadditive if for any \( s, t \geq 0 \) we have \( f(s + t) \geq f(s) + f(t) \). A superadditive function \( f \) is called admissible if \( \lim_{t \to 0} f(t) = 0 \). Furthermore, the function \( f \equiv 1 \) is also said to be admissible.

Notice that such superadditive functions are monotonically non-decreasing. For example, any linear function with non-negative slope is admissible. Moreover, increasing convex functions \( f \) with \( f(0) = 0 \) are admissible. For instance polynomials with non-negative coefficients and absolute term zero fulfill this criterion.

Definition 4.3. Given a fixed admissible function \( f: \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \), define for any two PDs \( X, Y \) their \( f \)-Prokhorov distance to be

\[
\pi_f(X, Y) = \inf\{t > 0: D_{X, Y}(t) < f(t)\} = \inf\{t > 0: \inf_{\eta} |\{x: d(x, \eta(x)) > t\}| < f(t)\}.
\]

Plugging in \( f = \text{id} \) gives the Prokhorov distance, plugging in \( f \equiv 1 \) recovers the bottleneck distance (this is why this function is admissible even though it is not superadditive).

Intuitively, for \( n \in \mathbb{N} \), plugging in \( f \equiv n \) (although this is not an admissible function) gives the \( n \)th bottleneck.

For two Prokhorov-close PDs, we require the number (=counting measure) of unmatched points to be small. Points with small persistence get matched to the diagonal and thus do not blow up the Prokhorov distance. Hence it is robust with respect to noise.

Example 4.4. Assume \( f \) is invertible. Recall the situation of Example 3.4: \( X = \{x\} \) and \( Y = \{y\} \) both consist of one point each and we assume that...
d(x,y) < d(x,x') + d(y,y'), where the prime denotes the projection to the diagonal. We saw that the bottleneck profile looks as follows:

\[ D_{X,Y}(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq d(x,y), \\ 0 & \text{if } t > d(x,y). \end{cases} \]

It follows that

\[ \pi_f(X,Y) = \min(f^{-1}(1), d(x,y)). \]

**Lemma 4.5.** For f admissible, \( D_{X,Y}(\pi_f(X,Y)) \leq f(\pi_f(X,Y)) \).

**Proof.** Note that \( D_{X,Y} \) is right-continuous by construction. \( \square \)

The triangle inequality follows from Lemma 3.7.

**Lemma 4.6.** Fix an admissible function \( f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} \). For any three persistence diagrams \( X, Y, Z \), we have

\[ \pi_f(X,Z) \leq \pi_f(X,Y) + \pi_f(Y,Z). \]

**Proof.** We make the following estimates:

\[
D_{X,Z}(\pi_f(X,Y) + \pi_f(Y,Z)) \\
\leq D_{X,Y}(\pi_f(X,Y)) + D_{Y,Z}(\pi_f(Y,Z)) \\
\leq f(\pi_f(X,Y)) + f(\pi_f(Y,Z)) \\
\leq f(\pi_f(X,Y) + \pi_f(Y,Z)).
\]

Here we used Lemma 3.7 for the first inequality, Lemma 4.5 for the second and superadditivity of \( f \) for the final one. Therefore,

\[
\inf\{ t > 0 : D_{X,Z}(t) < f(t) \} \leq \pi_f(X,Y) + \pi_f(Y,Z);
\]

the left hand side is the definition of \( \pi_f(X,Z) \), as desired. \( \square \)

As the symmetry is clear, we have shown:

**Theorem 4.7.** Fix an admissible function \( f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} \). The discrete \( f \)-Prokhorov metric is an extended pseudometric.

Just like for the bottleneck distance, we need some finiteness property for the \( \pi_f \) to be a genuine metric. Let \( \mathcal{B} \) denote the persistence diagrams which for every \( \varepsilon > 0 \) have only finitely many points of persistence \( > \varepsilon \). Then Lemma 3.8 implies:

**Lemma 4.8.** Let \( f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0} \) be admissible. For \( X, Y \in \mathcal{B} \), we have \( \pi_f(X,Y) = 0 \) only if \( X = Y \).

**Proof.** If \( \pi_f(X,Y) = 0 \), then \( D_{X,Y}(t) < f(t) \) for all \( t > 0 \). As the bottleneck profile is monotonically decreasing and \( \lim_{t \downarrow 0} f(t) = 0 \), this implies \( D_{X,Y}(t) = 0 \) for all \( t > 0 \). By Lemma 3.8, this happens only if \( X = Y \). \( \square \)
Our next task is to investigate how \( \pi_f \) depends on the function \( f \). While from a metric point of view, we need to fix \( f \), the context of data science suggests a different perspective: For given training data (a fixed set of persistence diagrams) adjust \( f \) to obtain a metric that performs well on it (e.g. in a classification problem, cf. section 5).

**Lemma 4.9.** Let \( f, g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \) such that \( f(t) \leq g(t) \) for all \( t \geq 0 \). Then for any two persistence diagrams \( X, Y \), we have \( \pi_g(X, Y) \leq \pi_f(X, Y) \).

**Proof.** If \( t > 0 \) satisfies \( D_{X,Y}(t) < f(t) \), then also \( D_{X,Y}(t) < g(t) \). Therefore,

\[
\inf\{t > 0 : D_{X,Y}(t) < g(t)\} \leq \inf\{t > 0 : D_{X,Y}(t) < f(t)\}
\]

and by definition \( \pi_g(X, Y) \leq \pi_f(X, Y) \).

For fixed persistence diagrams, the Prokhorov metric is continuous with respect to the functions in supremum metric.

**Proposition 4.10.** Fix two persistence diagrams \( X, Y \). Let \( f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \) be admissible. Then for all \( \varepsilon > 0 \) there is \( \delta > 0 \) such that for each admissible \( g : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} \), we have

\[
\|f - g\|_{\infty} < \delta \Rightarrow |\pi_f(X,Y) - \pi_g(X,Y)| < \varepsilon.
\]

**Proof.** Without loss of generality, assume that \( f(\pi_f(X,Y)) \leq g(\pi_g(X,Y)) \) (otherwise exchange \( f \) and \( g \) below). This implies \( \pi_f(X,Y) \geq \pi_g(X,Y) \) by monotonicity of \( D_{X,Y} \). We choose \( \delta < f(\varepsilon) \) and estimate

\[
f(\pi_g(X,Y) + \varepsilon) \geq f(\pi_g(X,Y)) + f(\varepsilon) \quad \text{by superadditivity}
\]

\[
> f(\pi_g(X,Y)) + \delta \quad \text{by choice of } \delta
\]

\[
> f(\pi_g(X,Y)) + \|f - g\|_{\infty} \quad \text{by choice of } g
\]

\[
\geq g(\pi_g(X,Y)) \quad \text{by definition of the sup-norm}
\]

\[
\geq f(\pi_f(X,Y)) \quad \text{by assumption.}
\]

By monotonicity of \( f \) we find that

\[
\pi_f(X,Y) - \pi_g(X,Y) = |\pi_f(X,Y) - \pi_g(X,Y)| < \varepsilon.
\]

From a data science perspective, the preceding Lemma allows us to tune the parameter function \( f \) on a fixed training set of persistence diagrams.

### 4.1 Comparison with Wasserstein

Fix a persistence diagram \( X \) and consider Wasserstein metrics and Prokhorov distances to some other diagram \( Y \). We can perturb \( Y \) by adding more “noise”. More precisely, we add \( k \) points whose distance to the diagonal is less than
\( \pi_f(X,Y) \) and denote this diagram by \( Y_k \). This does not affect the Prokhorov metric at all, while for all \( p \in [1, \infty) \), the value of \( W_p(X,Y_k) \) goes to infinity when \( k \) does. This is what we mean when we say that the Prokhorov metric is more robust with respect to noise compared to the Wasserstein metric. In other (more mathematical) words, the identity map \( \text{id}: (Dgm, \pi_f) \to (Dgm, W_p) \), where \( Dgm \) is the set of all persistence diagrams, is nowhere continuous for \( p \in [1, \infty)^2 \). In this section, we further explore the relation between Prokhorov and Wasserstein distances.

Similarly to the proofs in [GS02] for the measure-theoretic variants, we can bound our metric in terms of the Wasserstein distance. As we will explain, the metrics \( \pi_{t \to t^q} \) are of special interest.

**Proposition 4.11.** Let \( p \geq 1, q \geq 0, c > 0 \) and \( f(t) = c \cdot t^q \). For two persistence diagrams \( X, Y \) we have

\[
\pi_f(X,Y) \leq W_p(X,Y)^{\frac{q}{p+q}} \cdot c^{\frac{1}{p+q}}.
\]

**Proof.** Recall from Lemma 3.9 that

\[
D_{X,Y}(t) = \inf_{\eta} \{ x : d(x, \eta(x)) > t \} \leq \frac{1}{tp} W_p(X,Y)^p.
\]

We now want to find a suitable value of \( t \) such that \( D_{X,Y}(t) < c \cdot t^q \) to infer that \( \pi_f(X,Y) \leq t \). Plugging in \( t = W_p(X,Y)^{\frac{q}{p+q}} \cdot c^{\frac{1}{p+q}} \), one obtains

\[
\inf_{\eta} \{ x : d(x, \eta(x)) > W_p(X,Y)^{\frac{q}{p+q}} \cdot c^{\frac{1}{p+q}} \} \leq \frac{W_p(X,Y)^p}{W_p(X,Y)^{\frac{q}{p+q}} \cdot c^{\frac{1}{p+q}}}
\]

Now if \( q = 0 \), the right hand side simplifies to \( c = f(W_p(X,Y) \cdot c^{-1/p}) \). If \( q > 0 \), we compute

\[
\frac{W_p(X,Y)^p}{W_p(X,Y)^{\frac{q}{p+q}} \cdot c^{\frac{1}{p+q}}} = W_p(X,Y) = \frac{\frac{q}{p+q} - \frac{q}{p+q} \cdot \frac{p+q-q}{p+q}}{\frac{q}{p+q}} = c \cdot \left( W_p(X,Y)^{\frac{q}{p+q}} \cdot c^{\frac{1}{p+q}} \right)^q.
\]

Therefore,

\[
\inf_{\eta} \{ x : d(x, \eta(x)) > W_p(X,Y)^{\frac{q}{p+q}} \cdot c^{\frac{1}{p+q}} \} \leq c \cdot \left( W_p(X,Y)^{\frac{q}{p+q}} \cdot c^{\frac{1}{p+q}} \right)^q
\]

and we conclude \( \pi_f(X,Y) \leq W_p(X,Y)^{\frac{q}{p+q}} \cdot c^{\frac{1}{p+q}} \) as desired. \( \square \)

**Corollary 4.12.** Let \( p \geq 1, q \geq 0 \) and \( c > 0 \). The map \( \text{id}: (Dgm, W_p) \to (Dgm, \pi_{c \cdot t^q}) \) is continuous.

\(^2\)To avoid such problems, one usually restricts to a subset of \( Dgm \) of diagrams with “finite \( p \)th moment” [MMH11] when using \( p \)-Wasserstein distances.

19
When comparing with the bottleneck distance, i.e. $p = \infty$ in the above setting, we can say even more:

**Proposition 4.13.** For all admissible $f$ and all persistence diagrams we have $\pi_f(X, Y) \leq W_\infty(X, Y)$.

**Proof.** We recall by Lemma 3.3,

$$\inf_{\eta} \{x : d(x, \eta(x)) > W_\infty(X, Y)\} = 0 \leq f(W_\infty(X, Y)),$$

and therefore $\pi_f(X, Y) \leq W_\infty(X, Y)$.

Specializing to $c = 1$ and $p \in \{1, \infty\}$ or $q \in \{0, 1\}$, we obtain:

**Corollary 4.14.** The following inequalities hold:

| $p$ | $q$ | $0$ | $1$ | $q$ |
|-----|-----|-----|-----|-----|
| 1   | $d_B \leq W_1$ | $\pi \leq \sqrt{W_1}$ | $\pi_{tv} \leq W_1^{1+q}$ |
| $\infty$ | $d_B \leq d_B$ | $\pi \leq d_B$ | $\pi_{tv} \leq d_B$ |
| $p$ | $d_B \leq W_p$ | $\pi \leq W_p^{\frac{p}{1+q}}$ | $\pi_{tv} \leq W_p^{\frac{p+q}{1+q}}$ |

In particular, the Bottleneck Stability Theorem 2.13 implies stability for the new metrics by Proposition 4.13:

**Theorem 4.15.** Let $X, Y$ be finite metric spaces, fix some admissible function $f$ and $k \in \mathbb{N}$. Then we have

$$\pi_f(Dgm(PH_k(X)), Dgm(PH_k(Y))) \leq 2d_{GH}(X, Y),$$

where $d_{GH}$ is the Gromov-Hausdorff distance.

We can provide not only lower but also upper bounds for Wasserstein distances in terms of the Prokhorov distance.

**Proposition 4.16.** $W_q(X, Y)^q \leq \pi_{tv}(X, Y)^q(\max(d(x, \eta(x)))^q + |\eta|)$, where $\eta : X \to Y$ is any matching realizing $\pi_{tv}(X, Y)$.

**Proof.** For an arbitrary bijection $\eta : X \to Y$, consider any $t > 0$ such that
\[
\{|\{d(x, \eta(x)) > t\} \} \leq t^q. \text{ We estimate:}
\]
\[
W_p^q(X,Y)^q \leq \sum_x d(x, \eta(x))^q
\]
\[
= \sum_{d(x, \eta(x)) > t} d(x, \eta(x))^q + \sum_{d(x, \eta(x)) \leq t} d(x, \eta(x))^q
\]
\[
\leq |\{d(x, \eta(x)) > t\}| \max(d(x, \eta(x)))^q + t^q|\{d(x, \eta(x)) \leq t\}|
\]
\[
= |\{d(x, \eta(x)) > t\}| \max(d(x, \eta(x)))^q + t^q|\eta| - |\{d(x, \eta(x)) > t\}|
\]
\[
= |\{d(x, \eta(x)) > t\}||\max(d(x, \eta(x)))^q - t^q| + t^q|\eta|
\]
\[
\leq t^q \max(d(x, \eta(x)))^q - t^{2q} + t^q|\eta|
\]

Taking the infimum over all matchings and all such \( t \) we obtain the desired inequality
\[
W_p^q(X,Y)^q \leq \pi_t^q(X,Y)^q |\max(d(x, \eta(x)))^q + |\eta|). \]

Combining the two inequalities from Propositions 4.11 and 4.16, we obtain a comparison for different Wasserstein metrics.

**Corollary 4.17.** \( W_p^q(X,Y)^q \leq W_p(X,Y)^{\frac{p}{p+q}} (\max(d(x, \eta(x)))^q + |\eta|). \)

**Remark 4.18.** Another inequality relating Wasserstein distances for different \( p \) and \( q \) originates from the Hölder inequality, given in [AGS20, Lemma 3.5]: For finite persistence diagrams \( X, Y \) and real numbers \( 1 \leq q < p < \infty \), we have
\[
W_p^q(X,Y)^q \leq |\eta|^{\frac{1}{p} - \frac{1}{q}} W_p(X,Y),
\]
where \( \eta \) is the matching realizing \( W_p(X,Y) \). Our inequality above yields a lower exponent for \( W_p(X,Y) \) at the cost of multiplying with the largest distance in the matching. In particular, for \( q = 1, p = 2 \), our formula reads
\[
W_1(X,Y) \leq W_2^q(X,Y)^{\frac{2}{q}} (\max(d(x, \eta(x)))^q + |\eta|),
\]
with \( \eta \) realizing \( \pi_t^q(X,Y) \), whereas the one of [AGS20] reads (with \( \eta \) realizing \( W_2(X,Y) \))
\[
W_1(X,Y) \leq W_2^q(X,Y)|\eta|^{\frac{1}{q}}.
\]
Depending on the size of \( W_p(X,Y) \) relative to the size of \( X \) and \( Y \), our inequality can provide sharper bounds than the one of [AGS20]. To investigate the size of \( \max(d(x, \eta(x))) \) remains an interesting question for future work. One possible application of such inequalities is that they allow to infer stability results for vectorizations with respect to \( W_p \) for \( p > 1 \) from the stability with respect to \( W_1 \). Another use of Propositions 4.11 and 4.16 is that the bounds they provide for Wasserstein distances are easily computed, as we will see in Section 4.3 below.
4.2 Metric and Topological Properties

Using the comparison with Wasserstein (Section 4.1) and the results from [MMH11], we address questions of convergence and separability. We run into similar issues as [BV18, Theorems 4.20, 4.24, 4.25] and [Blu+14, section 3]. In this section, we explicitly allow diagrams with a countably infinite number of off-diagonal points under certain finiteness assumptions specified below.

**Theorem 4.19.** Let $p \geq 1$. The space of persistence diagrams with finite $p$th moment endowed with the $c \cdot t^q$-Prokhorov metric is separable.

**Proof.** Let $\varepsilon > 0$, $X$ a persistence diagram and $p \geq 1$. Let $S$ be a countable dense subset for the $p$-Wasserstein metric; this exists by [MMH11, Theorem 12]. In fact, they show that we can take $S$ to be the set of finite diagrams whose points have rational coordinates. Let $X_S \in S$ be a persistence diagram such that $W_p(X, X_S) < \varepsilon^p \cdot c^\frac{1}{p}$. Then by Proposition 4.11, we have

$$\pi_{c \cdot t^q}(X, X_S) \leq W_p(X, X_S)^\frac{p}{p+q} \cdot c^\frac{1}{p} < \varepsilon^p \cdot c^\frac{1}{p} \cdot c^\frac{1}{p+q} = \varepsilon.$$ 

Note that the assumptions in the previous Theorem are weaker than the ones usually considered for the bottleneck distance, compare [BV18, Theorem 4.18]. Recall that $\bar{\mathcal{B}}$ denotes the persistence diagrams which for all $\varepsilon > 0$ have finitely many points of persistence $> \varepsilon$. The next Theorem is a consequence of [Blu+14, Theorem 3.5], which asserts that the bottleneck distance makes $\bar{\mathcal{B}}$ into a Polish space.

**Theorem 4.20.** The space $\bar{\mathcal{B}}$ endowed with the Prokhorov metric $\pi_f$ is Polish for all admissible $f$.

**Proof.** Let $(X_n) \subset \bar{\mathcal{B}}$ be a Cauchy sequence with respect to the Prokhorov metric $\pi_f$. Let $\varepsilon > 0$ such that $f(\varepsilon) \leq 1$. Then the inequality $\pi_f(X_m, X_n) < \varepsilon$ implies by definition of $\pi_f$ that

$$D_{X_m, X_n}(\varepsilon) < f(\varepsilon) \leq 1.$$ 

As the bottleneck profile takes values in the integers, we conclude that $D_{X_m, X_n}(\varepsilon) = 0$ and hence, by Lemma 3.3, we have $\varepsilon \geq W_\infty(X_m, X_n)$. In particular, $X_n$ is a Cauchy sequence with respect to the bottleneck distance. By completeness of $\bar{\mathcal{B}}$ with the bottleneck distance, there is a limit diagram $X \in \bar{\mathcal{B}}$ to which the sequence converges. Finally by Lemma 4.13, convergence in bottleneck implies convergence in Prokhorov.

Now for separability, consider a subset $A \subset \bar{\mathcal{B}}$ which is dense with respect to the bottleneck distance. Let $X \in \bar{\mathcal{B}}$ and $\varepsilon > 0$. Then by assumption, there is $Y \in A$ with $W_\infty(X, Y) < \varepsilon$. Then, since by Proposition 4.13 $\pi_f(X, Y) \leq W_\infty(X, Y)$, we also have $\pi_f(X, Y) < \varepsilon$. Therefore, $A$ is dense in $\bar{\mathcal{B}}$ with respect to $\pi_f$ as well. 

\[ \square \]
4.3 Algorithms

In this section, all persistence diagrams are finite. Now we will provide an algorithm to compute \( \pi_f(X, Y) \) for continuous monotonically increasing functions \( f \). In this case, there is always a single value \( t_0 \in [0, \infty) \) such that \( D_{X,Y}(t) < f(t) \) for \( t > t_0 \) and \( D_{X,Y}(t) > f(t) \) for \( t < t_0 \). We can find its location by bisection. Recall that we set \( n = |X| + |Y| \).

**Proposition 4.21.** Let \( f : [0, \infty) \to [0, \infty) \) be monotonically increasing. Assume that the values and preimages of \( f \) can be computed in \( O(1) \). Then \( \pi_f(X, Y) \) can be computed in \( O(n^2 \log(n)) \).

**Proof.** First, observe that the Prokhorov distance takes its value among the pairwise distances of points in the persistence diagrams (if \( f \) crosses the bottleneck profile at one of its vertical gaps) or among preimages of integers under \( f \) (if \( f \) crosses the bottleneck profiles at one of its constant pieces), in formulas
\[
\pi_f(X, Y) \in \{d(x, y) : x \in X, y \in Y\} \cup f^{-1}(N_{\leq |X|+|Y|}) =: T_1.
\]

To perform a binary search, we sort the elements in \( T_1 \) as a preprocess, which has runtime complexity \( O(n^2 \log(n)) \). In each iteration of the binary search we pick the median \( t \in T_1 \). Next we compute the value of the bottleneck profile \( D_{X,Y}(t) \) using Proposition 3.12, taking \( O(n^2) \). Then we compute \( f(t) \), which by assumption takes \( O(1) \). Now if \( D_{X,Y}(t) > f(t) \) set \( T_{i+1} \) to be the right half, if \( D_{X,Y}(t) \leq f(t) \) set \( T_{i+1} \) to be the left half of \( T_i \). Hence we obtain a runtime of \( O(n^2 \log(n)) \) for the binary search as well.

---

**Procedure 1** The binary search to compute \( \pi_f(X, Y) \)

**Input:** Persistence diagrams \( X, Y \); function \( f \)

**Output:** \( \pi_f(X, Y) \)

1. \( T = \{d(x, y) : x \in X, y \in Y\} \cup f^{-1}(N_{\leq |X|+|Y|}) \)
2. sort \( T \)
3. \( L = 0; \; R = \text{length}(T) \)
4. while \( L < R \) do
5. \( m = \lfloor \frac{R+L}{2} \rfloor \)
6. \( t = T[m] \)
7. if \( D_{X,Y}(t) > f(t) \) then
8. \( L = m + 1 \)
9. else
10. \( R = m \)
11. end if
12. end while
13. return \( T[L] \)

In particular, if one uses a more efficient geometric data structure to improve the runtime of the matching algorithm, the sorting preprocessing dominates the
There is an easy modification to the above algorithm to approximate $\pi_f$ up to an additive error of $\varepsilon$. Instead of performing the binary search on the indicated discrete set (which needs to be sorted or otherwise pre-processed in a costly way, as noted), one can run it on an interval $[0, M]$. Here, $M$ is some upper bound, for example the sum of the longest lifespans of points in $X$ and $Y$ respectively (which is computed in $O(n)$). We bisect the interval until we arrive at one of length less than $2\varepsilon$. Its midpoint is guaranteed to be less than $\varepsilon$ away from the true value of $\pi_f(X,Y)$.

5 Experiments

A simple application of the bottleneck profile, based on simple synthetic persistence diagrams, was already presented in Example 3.6.

5.1 Highlighting Geometric Intuition

This experiment is a toy example, showing how the Prokhorov distance can capture our geometric intuition more accurately than bottleneck or Wasserstein. Consider three different shapes in $\mathbb{R}^2$: a) a big circle ($r = 6$), b) a big ($r = 6$) and a medium circle ($r = 4$), c) a big ($r = 6$), a medium ($r = 4$) and small circle ($r = 2$). We take five samples with noise from each shape according to Table 1.

| shape | number of circles | radii | samples | noise | colour in the figures |
|-------|-------------------|-------|---------|-------|-----------------------|
| a     | 1                 | 6     | 120     | uniform from $[-0.2, 0.2]^2$ | blue |
| b     | 2                 | 6, 4  | 300     | uniform from $[-0.23, 0.23]^2$ | red |
| c     | 3                 | 6, 4, 2 | 120     | uniform from $[-0.2, 0.2]^2$ | green |

Table 1: The three shapes: one two and three circles.

For each point cloud we compute the first persistent homology modules of it alpha complex filtration and represent them as PDs (see Figure 8). We can look at the averaged $D$-function for each pair of shapes (Figure 9). After careful inspection of this figure and some trial and error, we come up with the choice of $f(t) = t^3 \cdot 20^t$ to separate three bottleneck profiles in a most efficient way: Between around 0.55 and 0.65, the averaged bottleneck profiles involving shape c) with the small circle decrease, while the one comparing a) and b) stays constant. Intersecting with a function in this interval will provide a good choice for the Prokhorov distance: It puts the two and three circles closest to each other and one and three circles the farthest apart. In data science tasks, we will of course need an automated way to find a good parameter function $f$, we will discuss this in more detail below.

Now we want to compare the Bottleneck, Prokhorov and Wasserstein distances.
Figure 8: One two and three noisy circles and their PDs for the first persistent homology.

Figure 9: The averaged bottleneck profile for the three circles.
Figure 10: MDS plots of the dataset in Section 5.1.

Figure 11: Distance matrices of the dataset in Section 5.1.
The bottleneck distance between shapes a) and both b) and c) is roughly the same. This distance does not take the presence of the additional small circle in shape c). By blowing up the sample size and the noise in shape b), the Wasserstein distance from a) and c) to it are artificially blown up (Figures 10 and 10). The Prokhorov distance is built to avoid these pitfalls and nicely captures the geometry of the setting. The MDS plot for Prokhorov agrees with our intuition and places b) between a) and c) (Figures 10).

5.2 Classification Experiments

We now turn to more sophisticated data sets to illustrate the usage and advantages of the Prokhorov distance. In particular, we consider persistence diagrams that actually arise in applications of TDA. We use the library [Ped+11] for standard machine learning algorithms (in particular K-Neighbors). For the Bottleneck and Wasserstein metrics we use the Gudhi library [God21] and [DCR21]. To score the different metrics, we use K-neighbors classification accuracy as well as classification accuracy based on K-Medoids clustering with the “build” initialization [Sch], [SR20]. In the latter case, points are assigned to the class of the medoid of their cluster. We split the data sets into training and testing with 50% of the points each. All computations were carried out on a laptop with an Intel i5-8265U CPU with 1.60 GHz and 8 GB memory. The code to reproduce the experiments is available online.

Parameter tuning – choosing \( f \)

One needs to specify an admissible function \( f \) as a parameter for the Prokhorov distance \( \pi_f \). The set off all such functions is vast, therefore it is sensible to restrict to a smaller subset. In the experiments below, we choose \( f \) from linear functions with integer slope \( \in [10, 100] \). We do this by performing a grid search over the parameters and evaluating them by five-fold cross-validation. By selecting this subset of parameters, we reduce the risk of overfitting and are able to run the parameter selection in reasonable time. We leave it as a problem for further investigation to find better means to run the parameter selection, but note that the fact that the bottleneck profile is piecewise constant obstructs the use of gradient descent.

Prokhorov Distance for Cubical Complexes with Outlier Pixels

We generate\(^4\) 100 \( \times \) 100 pixel greyscale images according to the following procedure, cf. Figure 12. Initializing every pixel with 0, we choose \( n \) points at random, at which we add a Gaussian with \( \sigma = 3 \). We normalize the values to [0, 2] and then shift them up by 64. The goal is to distinguish images with \( n = 15 \) from images with \( n = 20 \). The obstacle is that we superimpose a particular kind

\(^3\)https://github.com/nihell/ProkhorovExamples/blob/master/Experiments.ipynb

\(^4\)Code available at https://github.com/nihell/ProkhorovExamples/blob/master/GenerateCubicalNoise.ipynb
noise, similar to salt-and-pepper noise. We choose $k$ pixels randomly at which we set the value to a random integer from $[1, 128]$; the eight surrounding pixels are set to zero. For each of the four combinations $n \in \{15, 20\}$ and $k \in \{3, 5\}$ we sample 50 greyscale images. We then create a cubical complex from each using the pixels as top-dimensional cells (lower-star filtration) and compute persistent homology in dimensions 0 and 1. We proceed as indicated at the beginning of this section to assess the accuracy of the different metrics. The results are summarized in Table 2. Both in dimension 0 and 1, the K-Neighbors classifier is inconclusive in the setting of Bottleneck and Wasserstein. With a suitable Prokhorov metric, we are able to achieve an accuracy of more than 80%. In the K-Medoids approach, the story is similar but less pronounced: Bottleneck and Wasserstein are inconclusive, but Prokhorov achieves around 60% accuracy.

![Figure 12: The underlying Gaussians, the superimposed noise and the resulting persistence diagram](image)

| dim | $f(t)$ | Prokhorov | Bottleneck | 1-Wasserstein | 2-Wasserstein |
|-----|--------|-----------|------------|---------------|--------------|
| K neighbors training score | 0 | 49t | 0.8425 | 0.515 | 0.58 | 0.525 |
| K neighbors test score | | | 0.8575 | 0.485 | 0.535 | 0.4925 |
| computation time [s] | | | 26.69 | 36.61 | 44.56 | 125.2 |
| parameter tuning time [s] | | | 1059 | | | |
| K medoids training score | 0 | 18t | 0.5975 | 0.5325 | 0.515 | 0.51 |
| K medoids test score | | | 0.62 | 0.51 | 0.485 | 0.5125 |
| computation time [s] | | | 70.14 | 108.6 | 127.6 | 379.4 |
| parameter tuning time [s] | | | 1082 | | | |
| K neighbors training score | 1 | 92t | 0.8625 | 0.5025 | 0.545 | 0.4825 |
| K neighbors test score | | | 0.825 | 0.5375 | 0.575 | 0.495 |
| computation time [s] | | | 26.57 | 36.68 | 44.43 | 125.9 |
| parameter tuning time [s] | | | 1025 | | | |
| K medoids training score | 1 | 16t | 0.62 | 0.4925 | 0.49 | 0.485 |
| K medoids test score | | | 0.5975 | 0.5125 | 0.4975 | 0.515 |
| computation time [s] | | | 77.08 | 113.9 | 132.6 | 401.9 |
| parameter tuning time [s] | | | 1098 | | | |

Table 2: Classification scores for the synthetic dataset.
3D Segmentation

We adapt an example from [Car] and [DCR21], which is based on the dataset [CGF09]. The task is to classify 3D-meshes based on the persistence diagrams of certain functions defined on them. The shapes are for example airplanes, hands, chairs ... The results of classification are presented in the Tables 3. All the considered metrics yield a similar accuracy. Prokhorov is the fastest, however at the cost of first having to find the suitable parameter, which took moore than ten hours in this case.

|                      | $f(t)$ | Prokhorov | Bottleneck | 1-Wasserstein | 2-Wasserstein |
|----------------------|--------|-----------|------------|---------------|---------------|
| K neighbors training score | 10t    | 0.9101    | 0.9098     | 0.9042        | 0.9059        |
|                      |        | 795.2     | 1252       | 838.5         | 1740          |
| K neighbors test score | 40440  | 0.9270    | 0.9245     | 0.9312        | 0.9298        |
| computation time [s]   |        | 838.5     | 2057       | 1740          |               |
| parameter tuning time [s] |        | 1740      |            |               |               |
| K medoids training score | 13t    | 0.4792    | 0.4905     | 0.4021        | 0.4592        |
|                      |        | 1946      | 3467       | 2057          | 5009          |
| K medoids test score   |        | 0.4985    | 0.4891     | 0.4126        | 0.5125        |
| computation time [s]   |        |           |            |               |               |
| parameter tuning time [s] |        |           |            |               |               |

Table 3: Classification scores for the 3d segmentation dataset.

Synthetic Dataset

Finally, we consider the dataset introduced by [Ada+17, Section 6.1]. It contains six shape classes: A sphere, a torus, clusters, clusters within clusters, a circle and the unit cube. From each class take 25 samples of 500 points. Then add two levels of Gaussian noise ($\eta = 0.05, 0.1$) and the zeroth and first persistent homology of the Vietoris-Rips filtration are computed. We compute the distance matrices and evaluate them based on the $K$-neighbors and $K$-medoids classifiers. The results are displayed in Table 4. We find that Prokhorov performs better Bottleneck and only slightly worse than Wasserstein. Prokhorov takes at most similarly long as 1-Wasserstein; Bottleneck is faster and 2-Wasserstein is slower.

5.3 Discussion

First and foremost, we found that Prokhorov is able to produce good results in situations where the classical tools of Bottleneck and Wasserstein fail. In order to explain the differences in the computation time, we note the size of the persistence diagrams in the various settings: By inspecting Table 5 we see that the 3D segmentation dataset contains way smaller diagrams, on which the Prokhorov metric seems to perform well, both in terms of runtime and score. On the bigger diagrams from the synthetic dataset, the Wasserstein metrics yield the highest scores. Prokhorov outperforms
|                  | dim | noise | $f(t)$ | Prokhorov | Bottleneck | 1-Wasserstein | 2-Wasserstein |
|------------------|-----|-------|--------|-----------|------------|---------------|---------------|
| K neighbors      | 0   | 0.05  | 42t    | 0.9067    | 0.8133     | 1.0           | 0.9867        |
| training score   |     |       | 4218   | 0.84      | 0.7867     | 0.96          | 0.9467        |
|                  |     |       |        | 144.2     | 45.39      | 252.8         | 1063          |
| computation time |     |       |        |           |            |               |               |
| [s]              |     |       |        |           |            |               |               |
| parameter tuning |     |       |        |           |            |               |               |
| time [s]         |     |       |        |           |            |               |               |
| K medoids        | 0   | 0.05  | 93t    | 0.8       | 0.68       | 0.9733        | 0.8933        |
| training score   |     |       | 4507   | 0.9067    | 0.6        | 0.88          | 0.8933        |
|                  |     |       |        | 465.3     | 156.1      | 801.3         | 3207          |
| computation time |     |       |        |           |            |               |               |
| [s]              |     |       |        |           |            |               |               |
| parameter tuning |     |       |        |           |            |               |               |
| time [s]         |     |       |        |           |            |               |               |
| K neighbors      | 1   | 0.05  | 51t    | 0.9733    | 0.7867     | 0.9867        | 0.96          |
| training score   |     |       | 736.2  | 1.0       | 0.7467     | 1.0           | 0.96          |
|                  |     |       |        | 145.8     | 44.22      | 267.0         | 1018          |
| computation time |     |       |        |           |            |               |               |
| [s]              |     |       |        |           |            |               |               |
| parameter tuning |     |       |        |           |            |               |               |
| time [s]         |     |       |        |           |            |               |               |
| K medoids        | 1   | 0.05  | 98t    | 0.8       | 0.6        | 0.9867        | 0.96          |
| training score   |     |       | 779.6  | 0.9067    | 0.56       | 0.96          | 0.9733        |
|                  |     |       |        | 465.3     | 161.0      | 791.4         | 3195          |
| computation time |     |       |        |           |            |               |               |
| [s]              |     |       |        |           |            |               |               |
| parameter tuning |     |       |        |           |            |               |               |
| time [s]         |     |       |        |           |            |               |               |
| K neighbors      | 1   | 0.1   | 61t    | 0.9333    | 0.92       | 0.92          | 0.9333        |
| training score   |     |       | 809.1  | 0.9467    | 0.9333     | 0.9867        | 0.9867        |
|                  |     |       |        | 28.17     | 22.28      | 26.98         | 138.4         |
| computation time |     |       |        |           |            |               |               |
| [s]              |     |       |        |           |            |               |               |
| parameter tuning |     |       |        |           |            |               |               |
| time [s]         |     |       |        |           |            |               |               |
| K medoids        | 1   | 0.1   | 50t    | 0.88      | 0.6933     | 0.8133        | 0.8133        |
| training score   |     |       | 832.2  | 0.8133    | 0.7067     | 0.8533        | 0.8533        |
|                  |     |       |        | 88.91     | 75.50      | 80.01         | 413.8         |
| computation time |     |       |        |           |            |               |               |
| [s]              |     |       |        |           |            |               |               |
| parameter tuning |     |       |        |           |            |               |               |
| time [s]         |     |       |        |           |            |               |               |

Table 4: Classification scores for the synthetic dataset from [Ada+17].

|                  | 3D-Segmentation | Synthetic data $H_0, \eta = 0.05$ | Synthetic data $H_0, \eta = 0.1$ | Synthetic data $H_1, \eta = 0.05$ | Synthetic data $H_1, \eta = 0.1$ |
|------------------|-----------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Mean size        | 11.84           | 500                               | 500                               | 177.7                             | 189.9                             |
| standard deviation| 4.893           | 0                                 | 0                                 | 40.53                             | 38.84                             |

Table 5: Cardinalities of the persistence diagrams for the considered experiments.
Bottleneck in the scores at the cost of higher runtimes. The difference in the computation time is caused by the evaluation of \( f(t) \), which is the only difference between the Bottleneck and Prokhorov implementations. Bottleneck – and to some extend also Prokhorov – work less well on zero-dimensional PDs. There, every class is born at time zero, hence the PD is intrinsically one-dimensional and points are matched in linear order. The bottleneck distance is less meaningful in this setting. Moreover, the Prokhorov (and even more the Bottleneck) distance do not take points matched over a small distance into account. This is a consequence of being designed to be robust against noise. However, this data can actually contain meaningful information, which is picked up by the Wasserstein distances. This is a possible explanation for the fact that Wasserstein yields better scores in the synthetic dataset. Hence, the Prokhorov metric works best on rather small diagrams and runs fastest with simple (e.g. linear) parameter functions \( f \). Even then, one needs to take the additional time for tuning the parameter \( f \) into account.

6 Discussion and Outlook

Summarizing the results from the previous section, we find that the Prokhorov metric is well-suited for small persistence diagrams. Large scale computations can be improved by the technique of entropic regularization from the theory of optimal transport [LCO18]. As the classical Prokhorov metric admits an optimal transport characterization, our discrete variant might be tractable using similar techniques.

A major aspect of the importance of the Bottleneck distance is its algebraic formulation in terms of interleavings. This theory generalizes to incorporate the family of Prokhorov metrics. An algebraic formulation would also provide a perspective on generalizations to multiparameter persistence.

Our results in section 4.2 establish that our construction yields a Polish space. This makes it suitable for statistical inference. In a similar vein, one can also investigate bottleneck profiles persistence diagrams arising from random geometric complexes. What kind of limit objects appear in this context? Can they be used to perform statistical testing?

Morally, stability theorems should involve related metrics on the input point cloud and on the persistence diagram side. This motivates to investigate Prokhorov-type distances for point clouds in \( \mathbb{R}^n \). Such distances might be useful throughout data science.

Code availability

We provide an implementation as a part of a custom gudhi fork at \url{https://github.com/nihell/persistence-prokhorov}. It is a modification of the GUDHI implementation of the Bottleneck distance [God21]. Let us first illustrate how to use it before we come to runtime considerations. The algorithm is implemented in C++ and comes with Python bindings.
It asks for three inputs: \texttt{diagram\_1}, \texttt{diagram\_2} and \texttt{coef}. The two diagrams need to be presented as 2D numpy arrays. The third parameter is a 1D numpy array representing the coefficients of a polynomial to be used as \( f \). Note that the zeroth entry needs to be zero in order to obtain a metric, compare Lemma 4.8. However, setting the polynomial to be a constant integer one recovers the values of \( D_{X,Y} \), which is a feature. In the technical details, our approach follows [God21], which follows [KMN17].

In addition, we also add the Prokhorov metric to [DCR21], allowing for parallel computations of distance matrices and integration with \texttt{sklearn}.

\textbf{Acknowledgement}

This work was in part supported by the Centre for Topological Data Analysis, EPSRC grant EP/R018472/1, and by the Dioscuri program initiated by the Max Planck Society, jointly managed with the National Science Centre (Poland), and mutually funded by the Polish Ministry of Science and Higher Education and the German Federal Ministry of Education and Research. We thank Gesine Reinert and Sayan Mukherjee for valuable discussions and Davide Gurnari for providing help with the experiments. Finally, we thank the anonymous referees for their helpful suggestions.

\textbf{References}

[Ada+17] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta and Lori Ziegelmeier. ‘Persistence images: a stable vector representation of persistent homology’, English. In: \textit{J. Mach. Learn. Res.} 18 (2017). Id/No 8, p. 35. issn: 1532-4435; 1533-7928/e.

[AGS20] Nieves Atienza, Rocio Gonzalez-Díaz and Manuel Soriano-Trigueros. ‘On the Stability of Persistent Entropy and New Summary Functions for Topological Data Analysis’. In: \textit{Pattern Recognition} 107 (Nov. 2020), p. 107509. issn: 0031-3203. doi: 10.1016/j.patcog.2020.107509.

[Blu+14] Andrew J Blumberg, Itamar Gal, Michael A Mandell and Matthew Pancia. ‘Robust statistics, hypothesis testing, and confidence intervals for persistent homology on metric measure spaces’. In: \textit{Foundations of Computational Mathematics} 14.4 (2014), pp. 745–789.

[BV18] Peter Bubenik and Tane Vergili. ‘Topological Spaces of Persistence Modules and Their Properties’. In: \textit{J Appl. and Comput. Topology} 2.3-4 (Dec. 2018), pp. 233–269. issn: 2367-1726, 2367-1734. doi: 10.1007/s41468-018-0022-4. arXiv: 1802.08117.
Mathieu Carrière. 3D shape segmentation using TDA. IPython notebook. URL: https://github.com/MathieuCarriere/sklearn-tda/tree/master/example/3DSeg.

Xiaobai Chen, Aleksey Golovinskiy and Thomas Funkhouser. ‘A Benchmark for 3D Mesh Segmentation’. In: ACM Transactions on Graphics (Proc. SIGGRAPH) 28.3 (Aug. 2009).

Frédéric Chazal, David Cohen-Steiner, Leonidas J Guibas, Facundo Memoli and Steve Y Oudot. ‘Gromov-Hausdorff stable signatures for shapes using persistence’. In: Computer Graphics Forum. Vol. 28. 5. Wiley Online Library. 2009, pp. 1393–1403.

Wojciech Chachólski and Henri Riihimaki. ‘Metrics and stabilization in one parameter persistence’. In: SIAM Journal on Applied Algebra and Geometry 4.1 (2020), pp. 69–98.

William Crawley-Boevey. ‘Decomposition of pointwise finite-dimensional persistence modules.’ English. In: J. Algebra Appl. 14.5 (2015). Id/No 1550066, p. 8. issn: 0219-4988; 1793-6829/e.

Pawel Dlotko, Mathieu Carrière and Martin Royer. ‘Persistence representations’. In: GUDHI User and Reference Manual. 3.4.1. GUDHI Editorial Board, 2021.

Herbert Edelsbrunner and John L. Harer. Computational topology. An introduction. English. 2010.

A. Efrat, A. Itai and M. J. Katz. ‘Geometry helps in bottleneck matching and related problems.’ English. In: Algorithmica 31.1 (2001), pp. 1–28. issn: 0178-4617; 1432-0541/e.

François Godi. ‘Bottleneck distance’. In: GUDHI User and Reference Manual. 3.4.1. GUDHI Editorial Board, 2021.

Loukas Grafakos. Classical Fourier Analysis. Vol. 249. Graduate Texts in Mathematics. New York, NY: Springer New York, 2014. isbn: 978-1-4939-1193-6 978-1-4939-1194-3. doi: 10.1007/978-1-4939-1194-3.

Misha Gromov. Metric structures for Riemannian and non-Riemannian spaces. Transl. from the French by Sean Michael Bates. With appendices by M. Katz, P. Pansu, and S. Semmes. Edited by J. LaFontaine and P. Pansu. English. Ed. by J. LaFontaine and P. Pansu. 2007.

Alison L Gibbs and Francis Edward Su. ‘On choosing and bounding probability metrics’. In: International statistical review 70.3 (2002), pp. 419–435.

John E. Hopcroft and Richard M. Karp. ‘An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs’. In: SIAM J. Comput. 2.4 (1st Dec. 1973). Publisher: Society for Industrial and Applied Mathematics, pp. 225–231. issn: 0097-5397. doi: 10.1137/0202019.
