Curvature of point clouds through principal component analysis

Yasuhiko Asao∗ Yuichi Ike†

January 10, 2022

Abstract
In this article, we study curvature-like feature value of data sets in Euclidean spaces. First, we formulate such curvature functions with desirable properties under the manifold hypothesis. Then we make a test property for the validity of the curvature function by the law of large numbers, and check it for the function we construct by numerical experiments. These experiments also suggest the conjecture that the mean of the curvature of sample manifolds coincides with the curvature of the mean manifold. Our construction is based on the dimension estimation by the principal component analysis and the Gaussian curvature of hypersurfaces. Our function depends on provisional parameters ε, δ, and we suggest dealing with the resulting functions as a function of these parameters to get some robustness. As an application, we propose a method to decompose data sets into some parts reflecting local structure. For this, we embed the data sets into higher dimensional Euclidean space using curvature values and cluster them in the embedding space. We also give some computational experiments that support the effectiveness of our methods.

Keywords: curvature, principal component analysis, manifold hypothesis, Gaussian random fields

1 Introduction

1.1 Motivation
Most of data we deal with take the form of point clouds in Euclidean spaces. As many data analysts propose, they are often distributed around an embedded manifold in the Euclidean space, which is now called the manifold hypothesis (see, for example, [MF11, NSW11, FMN16]). To extract some feature values from them is a pivotal task for data analysis. From the standpoint of the manifold hypothesis, estimating its curvature seems effective and natural. Several authors study the curvature of data from the view point of manifold learning or 3D shape study. See Subsection 1.4 for these related works.

∗Fukuoka University, 8-19-1 Nanakuma, Jonan-ku, Fukuoka city, Fukuoka, Japan. asao@fukuoka-u.ac.jp
†Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan. yuichi.ike.1990@gmail.com, ike@mist.i.u-tokyo.ac.jp
In this article, we first formulate a curvature function of data to satisfy some desirable properties under the manifold hypothesis, and we construct a candidate for such a function. Then we make a test property for the validity of the curvature function based on the law of large numbers, and check it for the function we construct. Our construction is based on the dimension estimation by the principal component analysis and the Gaussian curvature of hypersurfaces. We also give a computation algorithm for it as outlined in the next subsection.

1.2 Summary of our construction

Suppose that the data set $X$ we deal with is a finite subset of the Euclidean space $\mathbb{R}^n$ for some $n$. We estimate the curvature of $X$ at $p \in X$ by the following procedure.

(i) Fix thresholds $\varepsilon > 0, \delta > 0$.

(ii) Calculate the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ of the covariance matrix of the $\varepsilon$-neighborhood of $p$. Let $u_1, \cdots, u_n$ be the corresponding eigenvectors.

(iii) Look at the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_K \geq \delta$ that are greater than the threshold $\delta$. We regard the affine space $p + \langle u_1, \cdots, u_K \rangle$ as the tangent space of the data at the point $p$, and also regard $u_{K+1}$ as the normal direction at $p$.

(iv) We give a frame and an orientation of the tangent space. We project all the points around $p$ to the affine space spanned by the tangent space and the normal direction. Then we fit the resulting points by quadratic hypersurface in the affine space, and compute its Gaussian curvature at $p$ with respect to the fixed frame.

The steps (i)–(iii) are essentially the same as principal component analysis. See Subsection 3.1 for the mathematical background of (i)–(iii), and see Proposition 3.2 for that of (iv). For the detail of the algorithm, see Section 4.

1.3 Contributions

Our contributions are the following.

(1) In Section 2, we formulate curvature functions of data under the manifold hypothesis probability theoretically. Based on this formulation, we make a test property which is desirable for such a function. This property is verifiable by computations.

(2) In Section 3, we define the “curvature” of point sets and give theoretical foundation that supports the validity of the definition. For that purpose, we investigate a modified version of the principal component analysis and an estimation method of Gaussian curvature.

(3) In Section 4, we propose a novel algorithm to compute the curvature based on the theoretical background. We also propose a clustering algorithm relying on the curvature values of a point set.

(4) In Section 5, we experimentally show the effectiveness of our method using artificial and real-world data sets.
1.4 Related work

Our method first computes the tangent space at each point, which is related to studies in manifold learning (see, for example, [LV07, MF11, Wan12]). There one usually use $k$-nearest neighbor graph instead of a ball centered at each point as in our algorithm. Locally Linear Embedding (LLE) [RS00] estimates tangent spaces of a point set relying on $k$-nearest neighbor graph. [BM05] proposes a method to estimate tangent spaces by using neural networks whose input is global information of a point set, while LLE only uses local information. In most of the studies in manifold learning, it is often sufficient to estimate the tangent space for machine learning tasks, hence there are few studies on non-linear approximation of the manifold as in our method.

In 3D shape studies, curvature-like feature values have been studied from various point of views. [Tau95] proposes estimating curvature via polyhedral approximation. [LP05] computes curvature of a point set by solving some minimization problems to estimate tangent spaces. [CLS+19] studies the Weingarten map from a point set and estimates its convergence rate. In this article, we do not only focus on 3D shapes, but also explore a method that can be applied to broader type of data satisfying the manifold hypothesis.

A mathematical framework of ours related to the manifold hypothesis is close to one considered in [NSWT11]. They formalize the hypothesis by considering a distribution on the normal bundle of a submanifold of $\mathbb{R}^n$, where the submanifold is conceptualized as geometric core of the distribution, and the normal direction is considered for noise distribution. On the other hand, our formulation does not have any restriction on the direction of noise. Another formulation is discussed in [FMN16], where they consider a distribution supported on a unit sphere in a Hilbert space. In [NSW08, NSW11], the authors discuss how to recover the homology from observations by utilizing a quantity $1/\tau$ which is closely related to the curvature of manifolds. The relationship between $1/\tau$ and the curvature function we construct is not clear and we plan to investigate it in future work. In [AKTW18], they study the limit behavior of $1/\tau$ as the dimension of ambient space gets large. Subsequent research such as [FMR+14, CFL+17] studies the problem combining methods in topological data analysis with statistics. Our theoretical setting is somewhat similar to theirs (see Remark 2.1), but our method focuses on geometrical features rather than topological ones.

Acknowledgements

The first author was supported by RIKEN Center for Advanced Intelligence Project. The second author was partially supported by JST ACT-X (JPMJAX1903).

2 Our framework involving manifold hypothesis

In this section, we explain the hypothetical framework under which we work.

Let $\Omega$ be a probability space. We consider a stochastic process $\Theta: \Omega \times \mathbb{R}^m \to \mathbb{R}^n$ centered at a manifold, that is, the restriction $\Theta(\cdot, p): \Omega \to \mathbb{R}^n$ is measurable for
every \( p \in \mathbb{R}^m \), and the map \( \int \Theta : \mathbb{R}^m \to \mathbb{R}^n \) defined by

\[
\int \Theta(p) = \int \Theta(\omega, p)d\omega
\]
is an embedding. We call such a stochastic process a stochastic embedding. We regard stochastic embeddings as a source of data obeying the manifold hypothesis. We will later formulate data as some subset of the image of samples of stochastic embeddings.

Remark 2.1. If we have a restriction that \( \Theta(\omega, p) \) varies (with respect to \( \omega \)) only in the normal direction of the mean manifold at \( p \), this formulation is very close to that of Niyogi–Smale–Weinberger’s [NSW11]. They consider that the distribution of data is induced from that on the normal bundle of a manifold. One of the merits of our approach using stochastic process is that we can deal with multiple points simultaneously which are generated with some noise from distinct points of the manifold. This is convenient to consider the curvature of the sample points. On the other hand, Niyogi–Smale–Weinberger consider a quantity \( 1/\tau \) associated to an embedded manifold that is a curvature-like quantity. However, it seems difficult to estimate \( 1/\tau \) from the data.

We denote by \( \text{Emb}(\mathbb{R}^m, \mathbb{R}^n) \) the set of embeddings \( \mathbb{R}^m \to \mathbb{R}^n \), and also denote by \( \text{Emb}(\Omega \times \mathbb{R}^m, \mathbb{R}^n) \) the set of stochastic embeddings \( \Omega \times \mathbb{R}^m \to \mathbb{R}^n \). We also denote by \( \text{SP}(\Omega \times \mathbb{R}^m, \mathbb{R}^n) \) the set of stochastic processes \( \Omega \times \mathbb{R}^m \to \mathbb{R}^n \), that is a collection of maps \( \Omega \times \mathbb{R}^m \to \mathbb{R}^n \) satisfying that \( \Theta(-, p) : \Omega \to \mathbb{R}^n \) is measurable for every \( p \in \mathbb{R}^m \). Assume that we have a map \( c : \text{Emb}(\mathbb{R}^m, \mathbb{R}^n) \to \text{Map}(\mathbb{R}^m, \mathbb{R}) \) that extracts geometric feature of each point of embeddings, and we call such a map curvature function. Here \( \text{Map}(A, B) \) denotes the set of maps from \( A \) to \( B \). For example, we can take \( c \) as the sectional curvature of the embedded manifold equipped with a metric induced from the standard metric of the ambient Euclidean space. We further assume that, corresponding to a curvature function \( c \), there is a map \( c_{\Omega} : \text{Emb}(\Omega \times \mathbb{R}^m, \mathbb{R}^n) \to \text{SP}(\Omega \times \mathbb{R}^m, \mathbb{R}) \) that plays as a stochastic version of curvature function. We may expect that \( c_{\Omega} \) is a ‘blurring’ of the curvature map \( c \), and the expectation of \( c_{\Omega} \) coincides with \( c \). That is, the maps \( c \) and \( c_{\Omega} \) are expected to satisfy the following commutative diagram:

\[
\begin{array}{ccc}
\text{Emb}(\Omega \times \mathbb{R}^m, \mathbb{R}^n) & \xrightarrow{c_{\Omega}} & \text{SP}(\Omega \times \mathbb{R}^m, \mathbb{R}) \\
\downarrow{f} & & \downarrow{f} \\
\text{Emb}(\mathbb{R}^m, \mathbb{R}^n) & \xrightarrow{c} & \text{Map}(\mathbb{R}^m, \mathbb{R})
\end{array}
\] (2.1)

where \( f \) denotes the integration over \( \Omega \). We do not prove the existence of such a map \( c_{\Omega} \) (it will be studied in our future work), but we can check the existence if we restrict \( \text{Emb}(\Omega \times \mathbb{R}^m, \mathbb{R}^n) \) to the set of stochastic embeddings with the following conditions:

(i) \( m = 2, n = 3 \), and \( f \Theta \) is a surface,

(ii) \( \Theta(\omega, -) : \mathbb{R}^m \to \mathbb{R}^n \) is smooth for every \( \omega \in \Omega \),
\( \partial_{ij} \Theta(-, p) : \Omega \to \mathbb{R}^n \), the second derivatives at \( p \in \mathbb{R}^m \), are uncorrelated for every \( p \). That is, the covariance of each pair of the differentials is 0.

If \( \Theta \) satisfies the above conditions, then the mean of the Gaussian curvature coincides with the Gaussian curvature of the mean manifold \( \int \Theta \). We consider that such a restriction makes sense when we deal with real-world data. For example, a Gaussian process \( \Theta : \Omega \times \mathbb{R}^2 \to \mathbb{R}^3 \) with \( \Theta(\omega, p) = (p, t) \), \( t \sim N(0, 1) \) whose covariance is determined the kernel \( K(p, q) = p^Tq \) satisfies the above. It is checked by calculating

\[
\int \partial_{ij} \Theta(\omega, p) \partial_{i'j'} \Theta(\omega, p) d\omega = \partial_{ij} \partial_{i'j'} K(p, q)|_{p=q} = 0.
\]

For another example, we can take the RBF kernel \( K(p, q) = e^{-\|p-q\|^2} \) for the same process so that mean of the curvature coincides with the curvature of the mean. We can check it by calculating

\[
\int \partial_{11} \Theta(\omega, p) \partial_{22} \Theta(\omega, p) d\omega = \partial_{11} \partial_{22} K(p, q)|_{p=q} = 4,
\]

and

\[
\int \partial_{12} \Theta(\omega, p) \partial_{12} \Theta(\omega, p) d\omega = \partial_{12} \partial_{12} K(p, q)|_{p=q} = 4.
\]

We also check it numerically in Subsection 5.1.4.

Next we formulate the term “data” under the manifold hypothesis. Let \( X \) be a finite subset of \( \mathbb{R}^m \). For a fixed \( \omega \in \Omega \), we define a map \( (\omega, X) : \text{Emb}(\Omega \times \mathbb{R}^m, \mathbb{R}^n) \to \text{Map}(X, \mathbb{R}^n) \) by

\[
(\omega, X) \Theta = \Theta(\omega, -)|_{X}.
\]

We define a data as an element in \( \text{Map}(X, \mathbb{R}^n) \) which is contained in the image of \( (\omega, X) \) for some \( \omega \). For a data \( D \in \text{Map}(X, \mathbb{R}^n) \), we call each element \( D(x) \) for \( x \in X \) a datum. We denote \( \text{Data}_\Omega(X, \mathbb{R}^n) \) the set of data. A pivotal task of data analysis is to extract some feature value from a data which is a finite point cloud. Hence we formulate such task including curvature estimation as a map \( \text{Data}_\Omega(X, \mathbb{R}^n) \to \text{Map}(X, \mathbb{R}) \). In contrast to such maps, we regard \( c_\Omega \) as an ‘ideal feature value function’, and we expect that extraction from a data is an approximation of that by ideal function. Hence we define a curvature function of data \( c_d : \text{Data}_\Omega(X, \mathbb{R}^n) \to \text{Map}(X, \mathbb{R}) \) as a function with the following commutative diagram:

\[
\begin{array}{ccc}
\text{Emb}(\Omega \times \mathbb{R}^m, \mathbb{R}^n) & \xrightarrow{f \circ c_\Omega} & \text{Map}(\mathbb{R}^m, \mathbb{R}) \\
X^* \downarrow & & \downarrow X^* \\
\text{Map}(\Omega \times X, \mathbb{R}^n) & \xrightarrow{\tilde{c}_d} & \text{Map}(\Omega, \text{Map}(X, \mathbb{R})) \\
& \xrightarrow{f} & \text{Map}(X, \mathbb{R}),
\end{array}
\]

where \( X^* \) denotes the restriction to \( X \), \( \tilde{c}_d \) is only defined on \( \text{Im}X^* \) by \( \tilde{c}_d(X^*\Theta)(\omega) = c_d(\omega, X)\Theta \), and the map \( f \) in the bottom is defined by \( f \varphi(x) = \int \varphi(x)(\omega)d\omega \) for
any $\varphi \in \text{Map} (\Omega, \text{Map} (X, \mathbb{R}))$ and $x \in X$. If such a map $c_d$ exists, then we have the
convergence
\[
\sum_i c_d(\omega) \circ X^* \Theta_i \rightarrow \int c_d(\omega) X^* \Theta_i d\omega = c |_X \int \Theta_i,
\]
(2.2)
where $\Theta_i$’s in $\text{Emb} (\Omega \times \mathbb{R}^m, \mathbb{R}^n)$ by the law of large numbers. In this article, we
construct a candidate of the curvature function of data $c_d$ and show its validity by
checking (2.2) in Subsection 5.1.4.

3 Theoretical background

In this section, we give the theoretical background that supports our method. In
the first subsection, we review principal component analysis with slight modification
according to our usage. For the usual principal component analysis, we refer to,
for example, [Jol86, AW10]. This subsection details the theoretical background of
(i)–(iii) in Subsection 1.2. In the second subsection, we explain how to estimate
curvature of point sets from Gaussian curvature of hypersurface by using principal
components. This subsection details the background of (iv) in Subsection 1.2.

3.1 Our usage of principal component analysis

Let $X = \{p_1, \ldots, p_N\} \subset \mathbb{R}^n$ be a finite subset and $p \in \mathbb{R}^n$. For an inner product
space $(W, \cdot)$, we denote by $S(W)$ the set of all unit vectors in $W$. Let $u, v \in S(\mathbb{R}^n)$. We define the covariance of $X$ from $p$ along $u$ and $v$ by
\[
V_X(p, u, v) := \frac{1}{N} \sum_{i=1}^{N} ((p - p_i) \cdot u) ((p - p_i) \cdot v).
\]
We write $V_X(p, v, v) = V_X(p, v)$ and call it the variance of $X$ from $p$ along $v$. The variance $V_X(p, v)$ indicates how the data $X$ is scattered in the direction $v$ from the
point $p$. We define the covariance matrix of $X$ from $p$ by $V_X(p) := (V_X(p, e_i, e_j))_{i,j} \in M_n(\mathbb{R})$, where $e_i$’s are the standard basis of $\mathbb{R}^n$. Using $V_X(p)$, we can write $V_X(p, v)$ as
\[
V_X(p, v) = \frac{1}{N} \sum_{i=1}^{N} ((p - p_i) \cdot v)^2 \\
= \frac{1}{N} \sum_{i=1}^{N} v^T (p - p_i)(p - p_i)^T v \\
= v^T \left( \frac{1}{N} \sum_{i=1}^{N} (p - p_i)(p - p_i)^T \right) v \\
= v^T V_X(p) v.
\]
Hence all the eigenvalues of $V_X(p)$ are non-negative since $V_X(p, v) \geq 0$ for all $v$.
The principal component analysis method gives an orthogonal decomposition $\mathbb{R}^n \cong W_1 \oplus \cdots \oplus W_m$ such that any unit vector in $S(W_i)$ maximizes $V_X(p, v)$ under the
restriction $v \in S(W_i \oplus \cdots \oplus W_m)$. More precisely, we have the following proposition.
Proposition 3.1. Let $\lambda_1 > \cdots > \lambda_m$ \((1 \leq m \leq n)\) be the eigenvalues of the covariance matrix $V_X(p)$, and let $W_1, \ldots, W_m$ be the corresponding eigenspaces, respectively. Then for $1 \leq i \leq m$ and any $v \in S(W_i)$,

\[
v = \arg\max_{u \in S(W_1 \oplus \cdots \oplus W_m)} V_X(p, u).
\]

Proof. For $u_i \in W_i$ and $v_j \in W_j$ with $i \neq j$, we have $u_i \cdot v_j = 0$ since $\lambda_i(u_i \cdot v_j) = u_i^T V_X(p) v_j = \lambda_j(u_i \cdot v_j)$. Hence $W_i$’s are orthogonal to each other. By taking an orthonormal basis of each $W_i$’s, we have an orthonormal basis $u_1, \ldots, u_m$ of $W_1 \oplus \cdots \oplus W_m$. We denote the eigenvalue corresponding to $u_i$ by $\lambda_i'$. Obviously, we have $\lambda_1' \geq \cdots \geq \lambda_m'$. Let $v = \sum_{j=k}^{n} a_j u_j \in S(W_1 \oplus \cdots \oplus W_n)$. Since $u_j$’s are orthonormal, we have $\sum_{j=k}^{n} a_j^2 = 1$. Then we have

\[
V_X(p, v) = v^T V_X(p) v = \left( \sum_{j=k}^{n} a_j u_j \right)^T V_X(p) \left( \sum_{j=k}^{n} a_j u_j \right)
= \left( \sum_{j=k}^{n} a_j u_j \right)^T \left( \sum_{j=k}^{n} \lambda_j' a_j u_j \right)
= \sum_{j=k}^{n} \lambda_j' a_j^2 \leq \lambda_i.
\]

Hence, for any $v \in S(W_1 \oplus \cdots \oplus W_n)$, we obtain $V_X(p, v) \leq \lambda_i$. On the other hand, for $v \in S(W_i)$, we have $V_X(p, v) = v^T V_X(p) v = \lambda_i v^T v = \lambda_i$. This completes the proof. \hfill \square

Let us introduce a terminology for later use. Consider an embedding of a manifold $M \to \mathbb{R}^n$. If it factors through an embedding $\iota: M \to W$ for some affine subspace $W \subset \mathbb{R}^n$, we call the embedding $\iota$ an efficient embedding. From the standpoint of the manifold hypothesis that every point set we consider is distributed around an embedded manifold, the variance from a point on the manifold in the direction of its tangent vector should be large, while it should be very small in the other direction normal to the tangent space. Hence if we decompose $\mathbb{R}^n_p$, the set of all vectors from $p$, as $\mathbb{R}^n_p \cong W_1 \oplus \cdots \oplus W_m$ by Proposition 3.1, it is reasonable to regard its subspace $W_1 \oplus \cdots \oplus W_k$ as the tangent space at $p$. Here we fix some threshold $\delta > 0$, and $k$ is determined by the conditions $\lambda_k \geq \delta$ and $\lambda_{k+1} < \delta$. By adding an eigenvector in $W_{k+1}$ to this affine space, we regard it an ambient space in which the underlying manifold is efficiently embedded.

### 3.2 Estimation of Gaussian curvature

To estimate the curvature of a point set, we use the notion of Gaussian curvature of hypersurfaces. See [Pet06] for Gaussian curvature of Riemannian manifolds. It is known that if a hypersurface in $\mathbb{R}^n$ is expressed by an explicit function as $x_n = f(x_1, \cdots, x_{n-1})$ with $\partial f / \partial x_i = 0$ for $1 \leq i \leq n-1$, then its Gaussian curvature at the origin is expressed by the determinant of the Hessian $H_f = (\partial^2 f / \partial x_i \partial x_j)_{i,j}$. If a
manifold is efficiently embedded in codimension 1, then we can see the manifold as a hypersurface and compute its Gaussian curvature. Let \( \iota: M \to \mathbb{R}^n \) be an embedding of \( d \)-manifold, and \( a \in M \). Even if \( \iota \) is not codimension 1, under the assumption that the linear space \( W \) spanned by the first and the second derivatives of \( \iota \) at \( a \) is \((d+1)\)-dimensional, the following proposition guarantees that the orthogonal projection to \( W \) sufficiently approximates its Gaussian curvature at \( a \) by regarding the projected manifold as an embedding in \( W \). Here we estimate the Gaussian curvature at \( a \) by regarding the direction in \( W \) normal to the tangent space of \( \iota(a) \) as the codimensional direction.

**Proposition 3.2.** Let \( \iota: \mathbb{R}^m \to \mathbb{R}^n \) be an embedding of class \( C^2 \), and let \( a \in \mathbb{R}^m \). Consider the \( \mathbb{R} \)-vector space \( W_a \) spanned by
\[
\left\{ \frac{\partial \iota}{\partial x_1}(a), \ldots, \frac{\partial \iota}{\partial x_m}(a), \frac{\partial^2 \iota}{\partial x_1 \partial x_1}(a), \frac{\partial^2 \iota}{\partial x_1 \partial x_2}(a), \ldots, \frac{\partial^2 \iota}{\partial x_m \partial x_m}(a) \right\}. \tag{3.1}
\]
Then the affine space \( \iota(a) + W_a \subset \mathbb{R}^n \) has the following property. Let \( P: \mathbb{R}^n \to \mathbb{R}^n \) be the projection to \( \iota(a) + W_a \). Then for any \( \varepsilon > 0 \), there exist \( \varepsilon' > 0 \) such that the map \( \iota|_{B(a;\varepsilon')} \) is \( \varepsilon \)-close to \( P \circ \iota|_{B(a;\varepsilon')} \) in \( C^2 \), where \( B(a;\varepsilon') \) denotes the \( \varepsilon' \)-ball centered at \( a \).

**Proof.** We have \( \iota|_{B(a;\varepsilon')} = P \circ \iota|_{B(a;\varepsilon')} \). Furthermore, since the affine space \( \iota(a) + W_a \) is spanned by the vectors \( \{3.1\} \), we have
\[
\frac{\partial}{\partial x_i} P \circ \iota(a) = P \left( \frac{\partial \iota}{\partial x_i}(a) \right) = \frac{\partial \iota}{\partial x_i}(a),
\]
\[
\frac{\partial^2}{\partial x_i \partial x_j} P \circ \iota(a) = P \left( \frac{\partial^2 \iota}{\partial x_i \partial x_j}(a) \right) = \frac{\partial^2 \iota}{\partial x_i \partial x_j}(a)
\]
for every \( 1 \leq i, j \leq m \). Hence the statement follows from the continuity of the map and the derivatives. \( \square \)

Now we provide our definition of “curvature” of embeddings and point sets, which is not the standard notion of curvature, but enough for extracting a feature value of embeddings or point sets. For an embedding \( \iota \), we choose a normal direction \( n_a \) of tangent space at \( \iota(a) \) for each point \( \iota(a) \), and denote by \( W_a \) the affine space spanned by the tangent space at \( \iota(a) \) and \( n_a \). We project the embedding \( \iota \) to \( W_a \) and regard its image around \( \iota(a) \) as a hypersurface in \( W_a \). We define the curvature of the embedding at \( \iota(a) \) as the Gaussian curvature at \( \iota(a) \). By Proposition 3.2, this sufficiently approximates the Gaussian curvature in the situation explained right before the proposition. Let \( X \) be a point set and \( p \in X \). If they are distributed around an embedding of a manifold, it is easy to understand the following, but our procedure is irrelevant to the assumption. As explained in Subsection 3.1, we regard the affine space \( W_p = W_1 \oplus \cdots \oplus W_k \) determined by some fixed \( \delta > 0 \) as the tangent space of the underlying manifold. We choose an eigenvector in \( W_{k+1} \) in the manner explained below, and regard it as the normal direction. We choose an orientation of the tangent space as explained later, and estimate the Gaussian curvature at \( p \) by fitting a polynomial surface around \( p \) regarded as a hypersurface. Note that, when the tangent space is even dimensional, the Gaussian curvature is independent of the choice of an orientation. It is checked by calculating the Hessian with respect to
the square error. The choice of the normal direction and the orientation is done as follows. Let \( u_1, \ldots, u_K \) be orthonormal basis of \( W_p = W_1 \oplus \cdots \oplus W_k \) such that each \( u_i \) belongs to some \( W_j \). Such a basis exists since \( W_j \)'s are orthogonal to each other. See also the proof of Proposition 3.1. Consider \( u_i \)'s as points on the sphere \( S^{n-1} \). We give a decomposition \( U \sqcup S^{n-1} \setminus U \) such that \( v \in U \) or \( -v \in U \) for every point \( v \in S^{n-1} \), then we choose \( u_i \)'s so that they belong to \( U \). Let \( S^{i-1}_+ = \{(x_1, \ldots, x_i) \in S^{i-1} \mid x_i > 0\} \) be the upper semi-sphere for \( i \geq 0 \). Inductively we define \( U_0 = \{1\} \subset S^0 \) and \( U_i = S^n_+ \cup U_{i-1} \) for \( 1 \leq i \leq n-1 \). Then the obtained \( U_{n-1} \subset S^{n-1} \) satisfies the desired condition. We adjust the signatures of \( u_i \)'s so that they satisfy that \( u_i \in U_{n-1} \). This determines a frame and an orientation of \( W_p \), hence those of the tangent space at \( p \). It also determines \( u_{K+1} \), which is the normal direction.

### 4 Algorithm for computing curvatures of point clouds

In this section, we give our algorithm to compute the curvature of a given point set. We also provide a clustering algorithm of point sets depending on the curvature values. Until the end of this section, let \( X \) be a finite set in \( \mathbb{R}^n \).

#### 4.1 Curvature of point clouds

Our aim is to calculate a curvature-like quantity for each point \( p \in X \). In order to work locally near \( p \in X \), we fix a threshold \( \varepsilon > 0 \) and consider the subset \( B := X \cap B(p; \varepsilon) \) of \( X \), where \( B(p; \varepsilon) \) denotes the open ball of radius \( \varepsilon \) centered at \( p \). First we apply the principal component analysis described in Subsection 3.1 to \( B \). That is, we diagonalize the covariance matrix \( \text{Var}_B(p) \) as \( U^T DU \), where \( D = \text{diag}(\lambda_1, \ldots, \lambda_n) \) with \( \lambda_1 \geq \cdots \geq \lambda_n \) and \( U \) is an orthogonal matrix. Write \( U = [u_1, \ldots, u_n] \). We give an orientation for \( u_1, \ldots, u_n \) as in Subsection 3.2. It is pursued for each \( u_i = (s^i_1, \ldots, s^n_i) \) as follows. Look at the \( n \)-th coordinate \( s^n_i \), and set \( u_i = -u_i \) if \( s^n_i < 0 \), and remain if \( s^n_i > 0 \). If \( s^n_i = 0 \), look at the \( (n-1) \)-th coordinate \( s^{n-1}_i \) and repeat this procedure.

Now, set another threshold \( \delta > 0 \) and let \( \lambda_1 \geq \cdots \geq \lambda_K \geq \delta \) be the eigenvalues at least \( \delta \). We consider the \( K \)-dimensional affine space \( p + \langle u_1, \ldots, u_K \rangle \) that is centered at \( p \) and spanned by vectors \( u_1, \ldots, u_K \). We regard \( K \) as the dimension of \( X \) around \( p \), since the affine space would approximate the tangent space of the centered true manifold as explained in Section 3. The pseudocode for computing the dimension \( K \) and vectors \( u_1, \ldots, u_K, u_{K+1} \) is shown in Algorithm 4.

Next, we shall compute the curvature of \( X \) at \( p \). For a point \( q \in \mathbb{R}^n \), we set \( x_i(q) = (q - p) \cdot u_i \) for \( i = 1, \ldots, K + 1 \). We fit a quadratic hypersurface \( x_{K+1} = \frac{1}{2} \sum_{i,j} a_{i,j} x_i x_j =: f(x_1, \ldots, x_K) \) with \( a_{i,j} = a_{j,i} \) to \( X \). For this, we minimize the square error

\[
E(a) = \sum_{q \in B} \left( \frac{1}{2} \sum_{i,j} a_{i,j} x_i(q) x_j(q) - x_{K+1}(q) \right)^2.
\]

Solving the equations \( \partial E/\partial a_{i,j} = 0 \) gives \( a_{i,j} \)'s. We define the curvature of \( X \) at \( p \) as
by the Hessian $H_f$ of $f$, which is computed as

$$H_f = \det \begin{bmatrix} a_{1,1} & \cdots & a_{1,K} \\ \vdots & \ddots & \vdots \\ a_{K,1} & \cdots & a_{K,K} \end{bmatrix}.$$ 

The pseudocode for computing the curvature is shown in Algorithm 2.

Combining the procedures described above, we obtain an algorithm to compute the dimension and the curvature of $X$ at $p$ given thresholds $\varepsilon, \delta > 0$. The pseudocode for the overall algorithm is given in Algorithm 3. We can use the parameter $\varepsilon$ as a persistent parameter as in topological data analysis, which encodes a multi-scale topological feature of a given point set. That is, varying $\varepsilon$ from 0 to $+\infty$ we track change of values of dimension and curvature. A small $\varepsilon$ reflects the local structure of $X$ near $p$ while a large reflects the global one. Moreover, if the value is close to a fixed constant for different $\varepsilon$, that is, if the value “persists” for a long time, we can regard it as the intrinsic one. In Section 5, we show change of dimension and curvature depending on $\varepsilon$ in various examples.

**Algorithm 1:** Compute dimension

```plaintext
input : X: point set, p ∈ X: point, ε, δ: positive numbers
output: dim(X, p; ε, δ), (u_1, ..., u_{dim(X, p; ε, δ)}): non-negative integer and orthonormal vectors
1 COMPUTE_DIMENSION(X, p; ε, δ)
2 B ← X ∩ B(p; ε) ;
3 A ← Var_B(p) ;
4 diagonalize A as U^T D U with D = diag(λ_1, ..., λ_n), λ_1 ≥ ... ≥ λ_n,
   U = [u_1, ..., u_n] ;
5 K ← #{i ∈ {1, ..., n} | λ_i ≥ δ} ;
6 return K, (u_1, ..., u_K, u_{K+1})
```

**Algorithm 2:** Compute curvature

```plaintext
input : B: point set, p ∈ B: point, u_1, ..., u_K, u_{K+1}: orthonormal vectors
output: curv(B, p; u_1, ..., u_K, u_{K+1}): real number
1 COMPUTE_CURVATURE(X, p; u_1, ..., u_K, u_{K+1})
2 foreach q ∈ B, i = 1, ..., k + 1 do
3   x_i(q) ← (q - p) · u_i;
4   solve \[ \frac{\partial}{\partial a_{i,j}} \left( \sum_{q ∈ B} \left( \frac{1}{2} \sum_{i,j} a_{i,j} x_i(q) x_j(q) - x_{K+1}(q) \right)^2 \right) = 0 \] with
   a_{i,j} = a_{j,i} ;
5 return det \[ \begin{bmatrix} a_{1,1} & \cdots & a_{1,K} \\ \vdots & \ddots & \vdots \\ a_{K,1} & \cdots & a_{K,K} \end{bmatrix} \]
```
Algorithm 3: Compute dimension and curvature

\textbf{input} : $X$: point set, $p \in X$: point, $\varepsilon, \delta$: positive numbers
\textbf{output}: $\dim(X, p; \varepsilon, \delta)$, $\text{curv}(X, p; \varepsilon, \delta)$: non-negative integer and real number

1 $\text{ComputeDimensionCurvature}(X, p; \varepsilon, \delta)$
2 $K, (u_1, \ldots, u_K, u_{K+1}) \leftarrow \text{ComputeDimension}(X, p; \varepsilon, \delta)$;
3 $B \leftarrow X \cap B(p; \varepsilon)$;
4 $c \leftarrow \text{ComputeCurvature}(B, p; u_1, \ldots, u_K, u_{K+1})$;
5 return $K, c$

4.2 Clustering based on curvatures

Using the curvature values, we also propose a clustering algorithm. First we cluster the obtained curvature values, then add the cluster information as the $(n + 1)$th coordinate, and finally apply single-linkage clustering algorithm. More precisely, our clustering algorithm is the following:

(i) set a scaling parameter $t > 0$ and a threshold $d, d' > 0$;

(ii) compute the curvature value $c(p)$ for $p \in X$ depending on thresholds $\varepsilon, \delta > 0$ and set

$$a(p) := \begin{cases} -t & (c(p) < -d) \\ 0 & (|c(p)| \leq d) \\ t & (c(p) > d) \end{cases}$$

(iii) embed $X$ into $\mathbb{R}^{n+1}$ by $p \mapsto (p, a(p))$ and write $X'$ for the image;

(iv) connect every two point in $X'$ with distance less than $d'$ by an edge and output the resultant connected components.

If we vary the threshold $d'$ in the step (iv), this corresponds to computing hierarchical clustering or 0-th persistence homology in persistent homology theory. When we use a fixed threshold $d'$, one need to be set $d'$ less than $t$ so that the curvature coordinate effectively works. Namely, we want to distinguish two points $p$ and $p + \varepsilon'$ that are very near but have quite different curvature value, hence the threshold should be smaller than $\lim_{\varepsilon' \to 0} d((p, 0), (p + \varepsilon', t)) = t$. In our experiments presented in Section 5 we set the distance threshold $d'$ to be $t/2$. We remark that in the step (iv) we can use other methods than single-linkage clustering, such as $k$-means or mean-shift clustering. Applying a clustering algorithm to the embedded point set instead of the original $X$ enables us to separate convex and concave parts in the point set. The experiment with artificial and real-world data sets will be shown in the next section.

5 Experiments

In this section, we showcase our method to compute curvature-like quantity from artificial and real-world point sets. Although our method can be applied to point
sets in any dimensional Euclidean space, we restrict ourselves to point sets in 3-dimensional space for visualization.

In practical use, the density around a point in a given point set $X$ is different from point to point in general. Hence the number of points in the intersection $X \cap B(p; \varepsilon)$ heavily depends on the point $p$ if we use a fixed $\varepsilon$. For this reason, we adjust $\varepsilon$ for each point, from the viewpoint of density. More precisely, for the radius of the ball centered at $p$ we use $\varepsilon(p)$ computed as follows.

(i) Fix a positive number $\eta > 0$ and set $r = 1/10 \times (\text{the diameter of } X)$.
(ii) For $p \in X$, set $N(p) := \#(X \cap B(p; r))$, the number of points in $X \cap B(p; r)$.
(iii) Define $\varepsilon(p) := 2\eta/N(p)$ for each $p \in X$.

In all the experiment in this section, we set $\eta$ and compute $\varepsilon$ as described above. Moreover, the orientation was always taken as explained in Subsection 4.1.

5.1 Artificial data sets

In this subsection, we show the result of computation of the curvatures for artificial point sets.

5.1.1 Simple examples

First, we observe the behavior of our curvature value for two simple point sets. We generated two point sets consisting of 3000 points on the hypersurfaces $z = -x^2 - y^2$ and $z = x^2 - y^2$, respectively. The parameter $\varepsilon$ was set as explained above with $\eta$ equal to 3 times the diameter of the point set and $\delta = 0.001$. The results are shown in Figure 5.1.

![Curvature values for surfaces]

Figure 5.1: The curvature values for the surfaces $z = -x^2 - y^2$ and $z = x^2 - y^2$.

In these results, we can see that for the surface $z = -x^2 - y^2$ (left) the curvature values are positive while for $z = x^2 - y^2$ (right) the curvature values are negative, which coincide with the usual Gaussian curvatures $4/(1 + 4x^2 + 4y^2)^2$ and $-4/(1 + 4x^2 + 4y^2)^2$, respectively. As explained in Subsection 3.2 the curvature values are independent from the choice of frames since they are 2-dimensional.
Next we showcase two point sets that are difficult to distinguish by persistent homology but can be easily distinguished by our method. Roughly speaking, persistent homology captures the topology of point set \( X \) in \( \mathbb{R}^n \) and summarizes the information as a persistence diagram, which is a multiset in \((\mathbb{R} \cup \{\infty\})^2\). Here, we briefly explain what they are (see, for example [EH10], for more details). The idea is to consider the homology of the union \( X_r = \bigcup_{p \in X} \overline{B}(p; r) \) for \( r \geq 0 \), where \( \overline{B}(p; r) \) denotes the closed ball of radius \( r \) centered \( p \), and the family \( (H_n(X_r), i_{r,s})_{r \leq s \in \mathbb{R}_{\geq 0}} \) of homology and homomorphisms induced by inclusions \( i_{r,s} : X_r \hookrightarrow X_s \). The family is called the persistent homology. Any homology class \( \alpha \), which represents a connected component, a loop, or a cavity, etc., is generated at \( r = d_\alpha \) and vanishes at \( r = d_\alpha \), where \( d_\alpha = \infty \) if \( \alpha \) does not vanish. The collection \( \{(b_\alpha, d_\alpha)\}_{\alpha} \) for all homology classes is called the persistence diagram. A homology class \( \alpha \) can be seen as a topological noise if \( d_\alpha - b_\alpha \) is small, which means it vanishes just after the generation. Thus, in a persistence diagram, a point near the diagonal is regarded as a noise and can be ignored.

In the experiment, we used the two point sets displayed in the upper row of Figure 5.2. The first one is a finite set of 3000 points on the union of \( S^1 \times [0, 1] \) and two discs (upper left), where we randomly sampled 1500 point from \( S^1 \times [0, 1] \) and 750 points from discs, respectively. The second one is a finite set of 3000 points from the sphere with radius 0.5 (upper right). We computed the persistence homology of each point set and output the corresponding persistence diagram, which is shown in the middle row of Figure 5.2. Here, we used alpha complex for reducing the computation complexity, with the use of GUDHI library\[1\] in Python. We also computed the curvature values at each point in the point sets with \( \eta = 3 \times (\text{the diameter of the point set}) \), \( \delta = 0.001 \) and made the histogram of the values, which is shown in the third row of Figure 5.2.

In these results, we can see the two point sets have similar persistence diagrams while they have quite different histograms of curvature values. When the topological feature of data sets is not sufficient and more geometric information is needed for a task in question, our curvature values could be helpful in practical situation.

Next, we observe the behavior of curvature values depending on the choices of \( \varepsilon \). For this purpose, we generated two artificial point sets as follows. The first one is the union of \( S^1 \times [0, 1] \) and two discs same as in Figure 5.2 in Subsection 5.1.2. Here we randomly sample 1500 point from \( S^1 \times [0, 1] \) and 750 points from each disc. The second one is a union of \( S^1 \times [0, 1] \) and two hemi-ellipsoids (Figure 5.4). Similarly to the above, we randomly sample 1500 point from \( S^1 \times [0, 1] \) and 750 points from each hemi-ellipsoid. We applied our algorithm to these point clouds consisting of 3000 points. Here the threshold \( \delta \) was set to be 0.001 and where \( \eta \) was set to be

\[
k \times (\text{the diameter of a given point cloud}) \quad (k = 1, 2, 3, 4). \tag{5.1}
\]

The results are shown in Figures 5.3 and 5.4. The left columns show the estimated dimensions and the right ones indicate the curvatures.

\[1\]https://gudhi.inria.fr
Figure 5.2: The two point sets that have similar persistence diagrams and different histograms of curvature values. The major difference between the diagrams appear only near the diagonal and can be ignored from the view of persistent homology. While the two diagrams are similar apart from a neighborhood of the diagonal, the histograms of curvature values are quite different.

In Figure 5.3, apart from points near the edges, the dimension is 2 for most of the points, which coincides with the true dimension. Moreover, most of the points on the flat parts have curvature value close to 0 for different $\eta$. On the other hand, a point on the edges has non-zero value because the fitted quadratic surface has high curvature at the point. For a larger $\eta$, more points are used for the calculation of the principal component analysis and the eigenvalues of the variance matrix tends to be larger near the edges. This explains why enlarging $\eta$ increases the number of points whose curvature values are 3 near the edges. In Figure 5.4, we can observe that
points near the two poles have non-zero curvature values while those on the other parts have value close to 0. For this point sets, enlarging $\eta$ increases the number of points with estimated dimension 3.

We also applied our clustering algorithm explained in Subsection 4.2 to these point sets. We set the scaling parameter $t$ to be 4, $d' = t/2$, and the threshold $d$ to be 0.5. The results for clustering algorithm are showed in Figure 5.5 depending on different choices of $\varepsilon$. Since the curvature values were added as the fourth coordinates, the clustering algorithm separated points with different curvature values as well as coordinate values in $\mathbb{R}^3$. For $S^1 \times [0, 1]$ union two flat discs, the algorithm clustered the points sets into three parts, that is, the edges with high curvatures, the flat discs union the middle part of the column with curvature close to 0, and the rest. Adjusting the scaling parameter $t$ and $d'$, one could also cluster the point sets into neighborhoods of two edges on both sides and the rest. For $S^1 \times [0, 1]$ union two hemi-ellipsoids, the algorithm clustered the point sets into three parts, two regions near the poles and the rest, as expected. This is because the regions near the poles had higher curvature values and these regions were far enough in $\mathbb{R}^3$.

![Figure 5.5: The clustering results of the artificial data sets.](image)

5.1.4 Noisy datasets

Here, we show the validity of our construction of curvature function by checking (2.2), that is, mean of the curvature coincides with the curvature of the mean manifold. We consider the behavior of our algorithm on noisy data.

We generate point sets by the following procedure. First, we generate randomly 1000 points $a_1, \ldots, a_{1000}$ on the plane $\mathbb{R}^2$. Next, we consider a set of random variables $(X_1, \ldots, X_{1000})$ that follows $N(0, C)$, the normal distribution with mean 0 and covariance matrix $C = (C_{ij})$ defined by

$$C_{ij} = \exp(-\|a_i - a_j\|^2).$$

Consider a surface $z = f(x, y)$ with $f \equiv 0$ or $f(x, y) = \pm \sqrt{1 - x^2 - y^2}$. We generate 1000 points by setting $p_i := f(a_i) + (\sigma X_i)n_i$, where $\sigma > 0$ and $n_i$ is a unit normal vector of the surface at $f(a_i)$. The first distribution is same as the one explained in the fourth paragraph of Section 2. As mentioned in Remark 2.1, this modeling can also be seen as the model proposed by Niyogi–Smale–Weingberger [NSW11].
We compute the curvature values for the point set \( X = \{ p_1, \ldots, p_{1000} \} \) for 50 times with \( \sigma = 0.1, \eta = 1, \delta = 0.005 \). Then we compute the mean of the curvature values for each \( i \), which are displayed in Figure 5.6. We can see that mean of the curvature is close to the curvature of the mean manifold.

![Figure 5.6: The distributions of the curvature values of noisy planes (left) and noisy spheres (right) with \( \sigma = 0.1 \). We generated point sets 50 times and computed the averages.](image)

Now we conjecture that for stochastic embeddings with suitable kernel function and suitable curvature function, the diagram (2.1) commutes. It has already verified for the distribution with \( f(x, y) \equiv 0 \) in the fourth paragraph of Section 2.

### 5.2 Real-world data sets

Now we show the results for real-world data sets. For these experiments, we used **G-PCD: Geometry Point Cloud Dataset**\(^2\), from which we pick up Stanford Bunny and Stanford Dragon point sets. We randomly took 3000 points from each point set and applied our algorithm, setting \( \delta \) to be 0.001 and \( \eta \) varies as in (5.1). The results are shown in Figures 5.7 and 5.8.

In Figure 5.7, we can find that the estimated dimension fits well the true dimension 2 for most of the points when \( \eta = 3.816 \) or 5.088. For \( \eta = 1.272 \), only few points were used for the calculation of the principal component analysis and the algorithm could not estimate the covariance matrix well at each point. As \( \eta \) gets larger, the estimated dimension at each point tends to be bigger. This is again because more points are used for the principal component analysis and the eigenvalues of the variance matrix tends to be larger near the edges. Indeed, the points near the base and the ears have estimated dimension 3 for \( \eta = 5.088 \). For curvature values, most of the points on the flat body parts have values close to 0 for each \( \eta \), thus can be seen flat with our curvature. On the other hand, when \( \eta = 3.816 \) and 5.088, the curvature values at convex or concave part, e.g., the ears and the legs, have large absolute values. Figure 5.8 indicates a similar result for the dragon point sets. In this case, the high convexity and concavity causes the large absolute values of curvature for most of the points when \( \eta = 3.070 \) or 4.093.

Using the curvature values, we also applied the clustering algorithm presented in Subsection 4.2 to these point sets. Here, we set the scaling parameter \( t \) to be

\(^2\)https://www.epfl.ch/labs/mmspg/downloads/geometry-point-cloud-dataset/
4, \( d' = t/2 \), and the threshold \( d \) to be 0.5. The results are shown in Figure 5.9. The algorithm clusters the bunny point set into three parts, the flat body part, the legs plus the ears that have large absolute value of curvature, and outliers. For the dragon point set, the result is not satisfactory as the bunny set. The method clusters the point set into three parts, depending on curvature value is sufficiently positive, close to 0, and sufficiently negative. In this case, one of the cluster can be seen as the flat body parts while the meaning of the others is not so clear.

6 Conclusion

Curvature is a essential quantity for studying spaces and would be helpful for studying point sets in data analysis. In this article, we have proposed a mathematical framework that indicates how the curvature of data should behave. Then we have given a construction of curvature function for data, and have shown its validity by numerical experiments. Here we conjecture that our framework is appropriate in the sense that the diagram (2.1) commutes in suitable situations. We also have proposed a method to compute the dimensions and the curvatures of point sets with respect to our definitions. We have also given some experimental results which supports effectiveness of our method. Fixing the parameter \( \varepsilon \) in our algorithm would take unexpected values due to noises, thus we proposed computing the curvature values as functions of \( \varepsilon \). We expect that such function-like treatment would give robustness against noise, and we would build mathematical foundation for them in future work.
Figure 5.3: The estimated dimensions (left) and the curvature values (right) of $S^1 \times [0, 1]$ union two flat discs.
Figure 5.4: The estimated dimensions (left) and the curvature values (right) of \( S^1 \times [0, 1] \) union two hemi-ellipsoids.
Figure 5.7: The estimated dimensions (left) and the curvature values (right) of the Stanford bunny data set.
Figure 5.8: The estimated dimensions (left) and the curvature values (right) of the Stanford dragon data set.
References

[AKTW18] Robert J Adler, Sunder Ram Krishnan, Jonathan E. Taylor, and Shmuel Weinberger. Convergence of the reach for a sequence of gaussian-embedded manifolds. *Probability Theory and Related Fields*, 171(3):1045–1091, 2018.

[AW10] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[BM05] Yoshua Bengio and Martin Monperrus. Non-local manifold tangent learning. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2005.

[CFL+17] Frédéric Chazal, Brittany Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, Alessandro Rinaldo, and Larry Wasserman. Robust topological inference: Distance to a measure and kernel distance. *The Journal of Machine Learning Research*, 18(1):5845–5884, 2017.

[CLS+19] Yueqi Cao, Didong Li, Huafei Sun, Amir H Assadi, and Shiqiang Zhang. Efficient weingarten map and curvature estimation on manifolds. *arXiv e-prints*, arXiv:1905.10725 [stat.ML] 2019.

[EH10] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[FLR+14] Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, Aarti Singh, et al. Confidence sets for persistence diagrams. *Annals of Statistics*, 42(6):2301–2339, 2014.

[FMN16] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[Jol86] Ian T Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986.

[LP05] Carsten Lange and Konrad Polthier. Anisotropic smoothing of point sets. *Computer Aided Geometric Design*, 22(7):680–692, 2005.

[LV07] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[MF11] Yunqian Ma and Yun Fu. *Manifold learning theory and applications*. CRC press, 2011.

[NSW08] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

[NSW11] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. A topological view of unsupervised learning from noisy data. *SIAM Journal on Computing*, 40(3):646–663, 2011.
[Pet06] Peter Petersen. *Riemannian geometry*, volume 171. Springer, 2006.

[RS00] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[Tau95] Gabriel Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proceedings of IEEE International Conference on Computer Vision*, pages 902–907. IEEE, 1995.

[Wan12] Jianzhong Wang. *Geometric structure of high-dimensional data and dimensionality reduction*, volume 5. Springer, 2012.