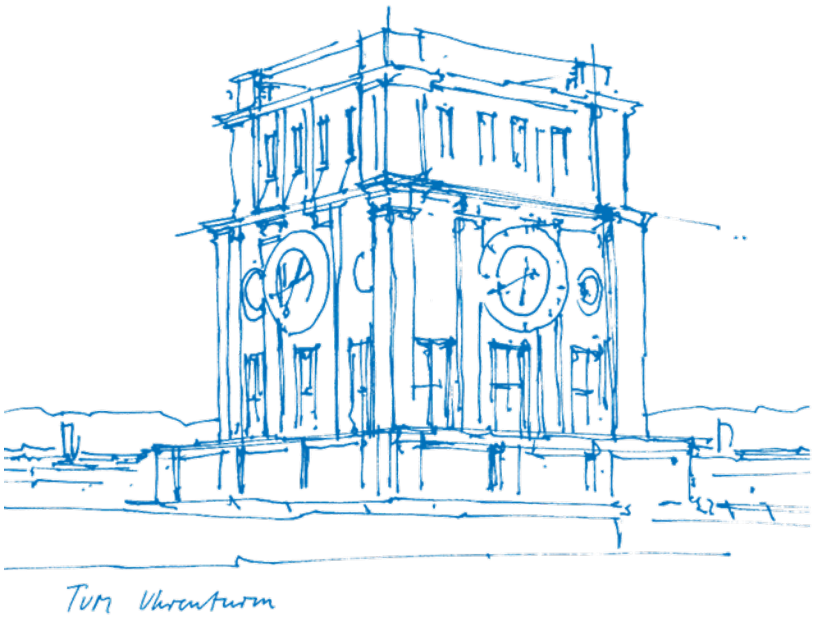
Representation of data:
Manifold learning with Diffusion Maps

Dr. Felix Dietrich



Today: Manifold Learning

Representation of data with Diffusion Maps

1. Definition: manifold
2. Topology and geometry
3. Manifold learning
4. Laplace-Beltrami operator
5. Diffusion Maps algorithm

Representation of data

High-dimensional data with low-dimensional structure - general idea

1. Given input: data matrix $X \in \mathbb{R}^{N \times n}$ with N data points in n-dimensional space.
2. ...algorithm...
3. Output: new representation of the data, e.g. as another coordinate matrix $U \in \mathbb{R}^{N \times p}$.

Ideally: $p \ll n$, so that the dimension of the data is reduced (manifold learning, compression).
For visualization, $p = 2, 3, (4)$ is necessary.
Example for a low-dimensional structure: $U \in \mathbb{R}^{1000 \times 3}$ with rows $u_i \in \mathbb{R}^3$, $\|u_i\| = 1$:

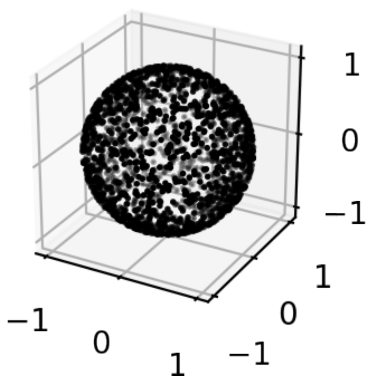


Figure: Data set where the points u_i (black) are distributed on a sphere.

Representation of data

High-dimensional data with low-dimensional structure - manifolds

[Manifolds are] generalizations of curves and surfaces to arbitrarily many dimensions [and] provide the mathematical context for understanding "space" in all of its manifestations. [Lee, 2012]

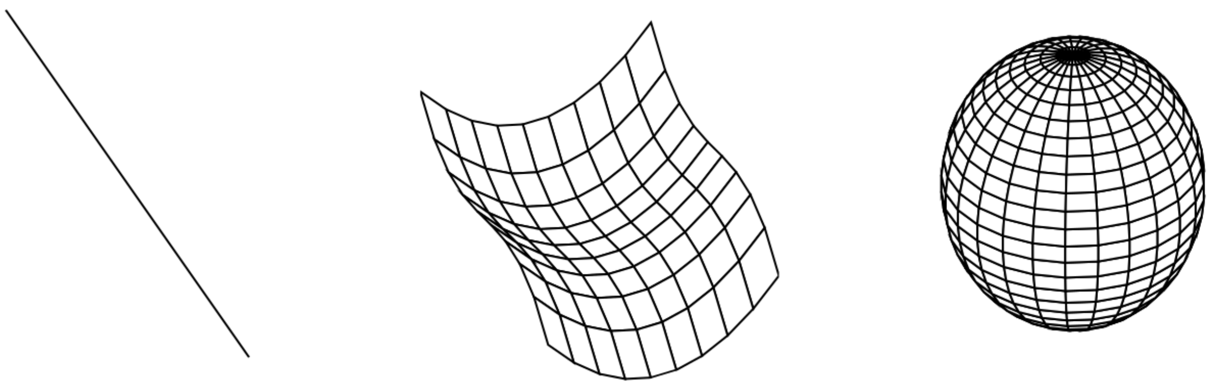


Figure: From [Dietrich, 2017]: Examples for manifolds with different geometries and intrinsic dimensions. The line segment is of intrinsic dimension one, the center surface is a two-dimensional manifold, curved and embedded in three-dimensional space. The sphere has intrinsic dimension two, but cannot be deformed through any homeomorphism into the surface in the center. Remark regarding last lecture: there are also geometric bifurcations!

Representation of data

High-dimensional data with low-dimensional structure - manifolds

Definition: Manifold, shortened. A topological space M is a topological manifold of dimension d if M is locally Euclidean: each point of M has a neighborhood that is homeomorphic to an open subset of R^d. [Lee, 2012]
[To be precise: M has to be Hausdorff and second-countable, too.]

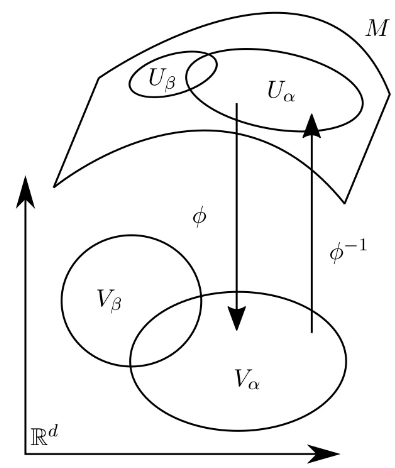
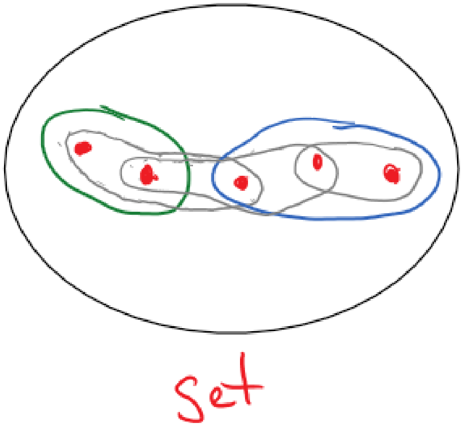
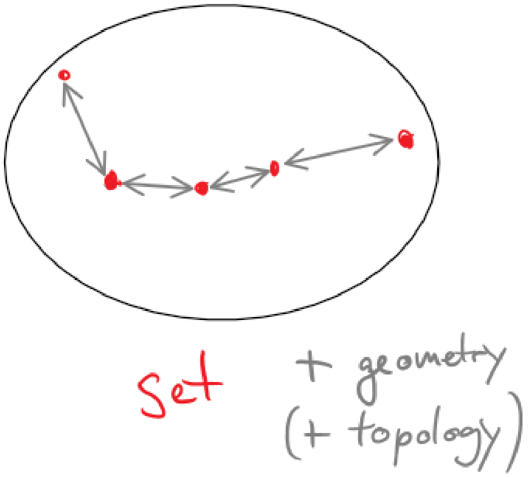


Figure: Visualization of a manifold M. The subsets $U_\alpha, U_\beta \subset M$ and $V_\alpha, V_\beta \subset \mathbb{R}^d$ are open sets, ϕ is a homeomorphism.

Representation of data

Topology versus geometry





Let X be a set. A ***topology on*** X is a collection $\mathcal{T}$ of subsets of X, called ***open subsets***, satisfying

(i) X and $\varnothing$ are open.
(ii) The union of any family of open subsets is open.
(iii) The intersection of any finite family of open subsets is open.

A pair $(X, \mathcal{T})$ consisting of a set X together with a topology $\mathcal{T}$ on X is called a ***topological space***.

A ***metric space*** is a set M endowed with a ***distance function*** (also called a ***metric***) $d : M \times M \to \mathbb{R}$ satisfying the following properties for all $x, y, z \in M$:

(i) POSITIVITY: $d(x, y) \geq 0$, with equality if and only if $x = y$.
(ii) SYMMETRY: $d(x, y) = d(y, x)$.
(iii) TRIANGLE INEQUALITY: $d(x, z) \leq d(x, y) + d(y, z)$.

Representation of data

Topology versus geometry

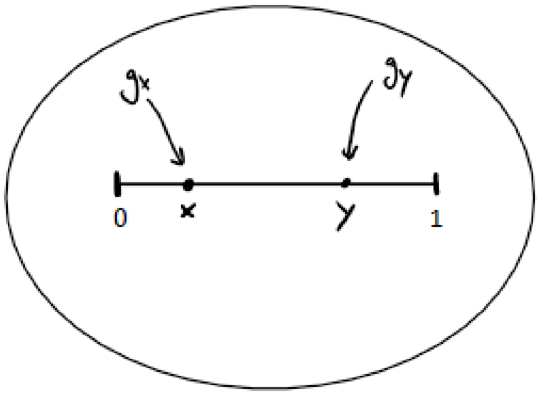
SAME topology

DIFFERENT geometry

https://upload.wikimedia.org/wikipedia/commons/2/26/Mug_and_Torus_morph.gif
Author: Lucas Vieira

Representation of data

Riemannian manifolds

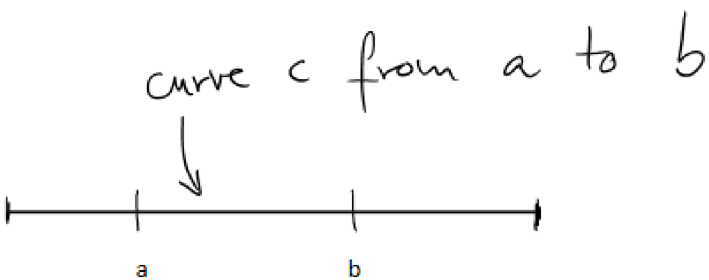


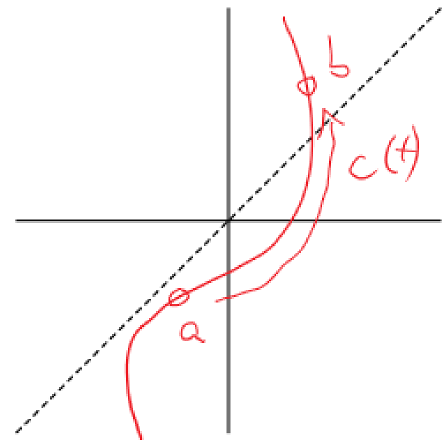
Riemannian manifold

A ***Riemannian metric on M*** is a smooth symmetric covariant 2-tensor field on M that is positive definite at each point. A ***Riemannian manifold*** is a pair (M, g), where M is a smooth manifold and g is a Riemannian metric on M. One sometimes simply says "M is a Riemannian manifold" if M is understood to be endowed with a specific Riemannian metric.

Representation of data

Curves on Riemannian manifolds

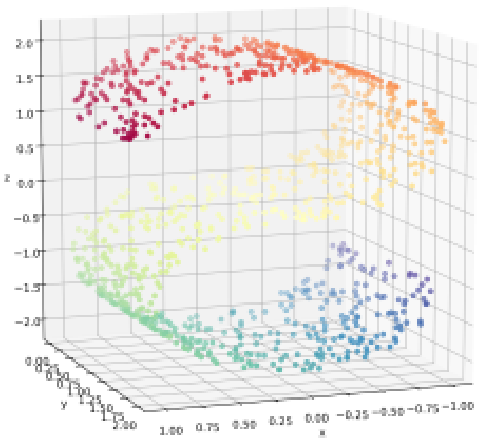


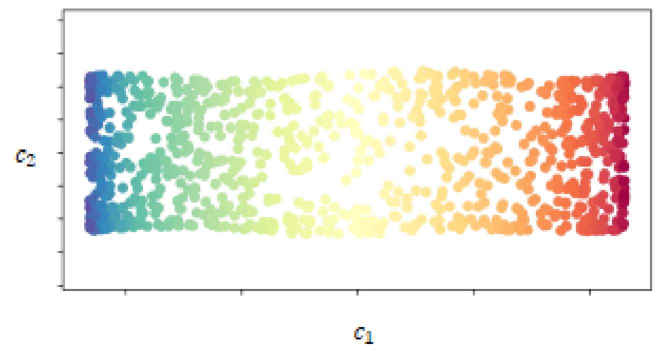


$$L_a^b(c) := \int_a^b \sqrt{g(c'(t), c'(t))}\,\mathrm{d}t = \int_a^b \|c'(t)\|\,\mathrm{d}t.$$

Representation of data

Manifold learning - in general





	A	B	C	D
1	x	y	z	
2	0.579699	0.418081	0.049091	
3	0.918333	0.819384	0.889048	
4	0.641222	0.453401	0.587756	
5	0.827659	0.808729	0.108997	
6	0.320323	0.698129	0.04561	
7	0.223276	0.876384	0.149571	
8	0.01184	0.972112	0.989366	
9	0.087981	0.895696	0.379578	
10	0.491548	0.171433	0.018083	
11	0.453172	0.083112	0.907416	
12	0.797321	0.076609	0.912459	

	A	B	C	D
1	c1	c2		
2	0.008613	0.964437		
3	0.4912	0.477741		
4	0.912283	0.650359		
5	0.964879	0.159634		
6	0.254983	0.316284		
7	0.241464	0.989014		
8	0.892201	0.011716		
9	0.682857	0.478074		
10	0.973667	0.323248		
11	0.357255	0.258019		
12	0.181788	0.141262		
13	0.882955	0.9977		

Representation of data

Nonlinear manifold learning: Diffusion Maps

1. Basic idea: eigenfunctions of the diffusion operator Δ embed the manifold with data X [Coifman et al., 2005, Coifman and Lafon, 2006].
2. Algorithm: compute a few eigenfunctions evaluated on the data, use them as new coordinates U [Nadler et al., 2006, Berry et al., 2013].
3. Challenge: how to define a diffusion operator on a point cloud X?

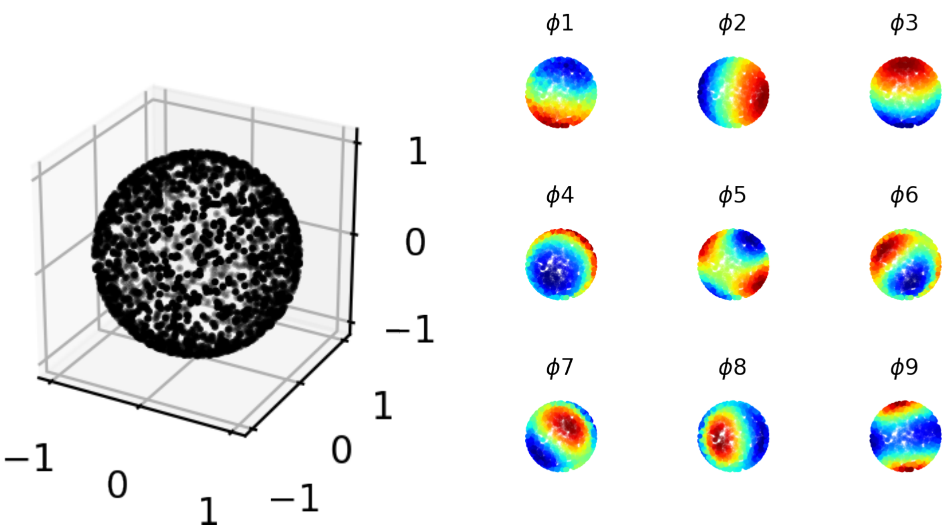


Figure: Spherical data set and eigenfunctions of the Laplace-Beltrami (Diffusion) operator.

Representation of data

Nonlinear manifold learning: Diffusion Maps

Challenge: how to define a diffusion operator on a point cloud X?

Diffusion equation: find a function $f : T \times M \to \mathbb{R}$, with specified initial data $f(0, x) = g(x)$, solve

$$\frac{\partial}{\partial t} f = \Delta f. \tag{1}$$

Note: if $M = \mathbb{R}$, the real line, $\Delta = \frac{\partial^2}{\partial x^2}$.

Representation of data

Nonlinear manifold learning: Diffusion Maps

Challenge: how to define a diffusion operator on a point cloud X?
Diffusion equation: find a function $f : T \times M \to \mathbb{R}$, with specified initial data $f(0, x) = g(x)$, solve

$$\frac{\partial}{\partial t} f = \Delta f. \tag{1}$$

Note: if $M = \mathbb{R}$, the real line, $\Delta = \frac{\partial^2}{\partial x^2}$.
Main idea: the solution of equation (1) with initial condition $f(0, x) = \delta_x$ is

$$f(t, x) = \exp(t\Delta)\delta_x. \tag{2}$$

Locally and for small t, that solution is a “bump function” centered at x, of the form

$$k(t, y) = \exp(-\|x - y\|^2/t) \tag{3}$$

where x is the center point and y is another point in the neighborhood of x.

Representation of data

Nonlinear manifold learning: Diffusion Maps

Challenge: how to define a diffusion operator on a point cloud X?
Diffusion equation: find a function $f : T \times M \to \mathbb{R}$, with specified initial data $f(0,x) = g(x)$, solve

$$\frac{\partial}{\partial t} f = \Delta f. \tag{1}$$

Note: if $M = \mathbb{R}$, the real line, $\Delta = \frac{\partial^2}{\partial x^2}$.
Main idea: the solution of equation (1) with initial condition $f(0,x) = \delta_x$ is

$$f(t,x) = \exp(t\Delta)\delta_x. \tag{2}$$

Locally and for small t, that solution is a "bump function" centered at x, of the form

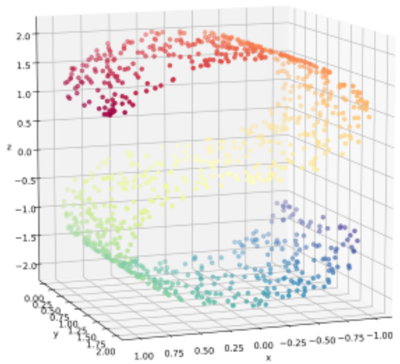
$$k(t,y) = \exp(-\|x-y\|^2/t) \tag{3}$$

where x is the center point and y is another point in the neighborhood of x.
Algorithm: compute k for all pairs of N points in the data set, with a small value of t. This results in a "kernel matrix" $K \in \mathbb{R}^{N \times N} \approx \exp(t\Delta)$. Then, solve the eigenproblem

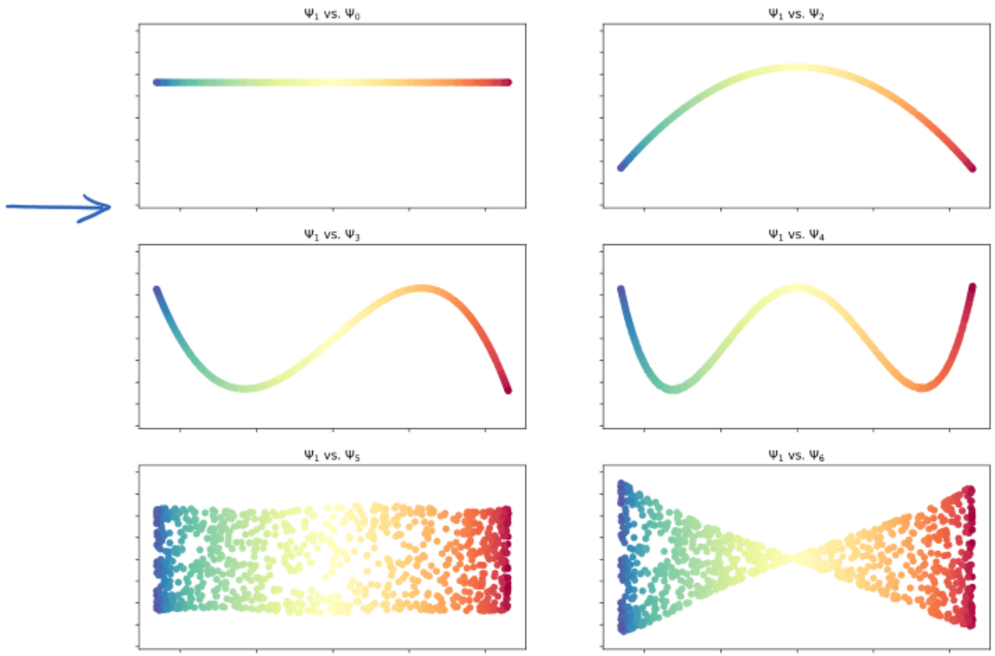
$$\exp(t\Delta)\phi_l = \lambda_l \phi_l. \tag{4}$$

Representation of data

Manifold learning - S-curve with Diffusion Maps



	A	B	C	D
1	x	y	z	
2	0.579699	0.418081	0.049091	
3	0.918333	0.819384	0.889048	
4	0.641222	0.453401	0.587756	
5	0.827659	0.808729	0.108997	
6	0.320323	0.698129	0.04561	
7	0.223276	0.876384	0.149571	
8	0.01184	0.972112	0.989366	
9	0.087981	0.895696	0.379578	
10	0.491548	0.171433	0.018083	
11	0.453172	0.083112	0.907416	
12	0.797321	0.076609	0.912459	



Also see here: https://datafold-dev.gitlab.io/datafold/tutorial_basic_dmap_scurve.html

Representation of data

Nonlinear manifold learning: Diffusion Maps

Given a data set $\{y_i \in \mathbb{R}^n\}_{i=1}^N$ [Berry et al., 2013]:

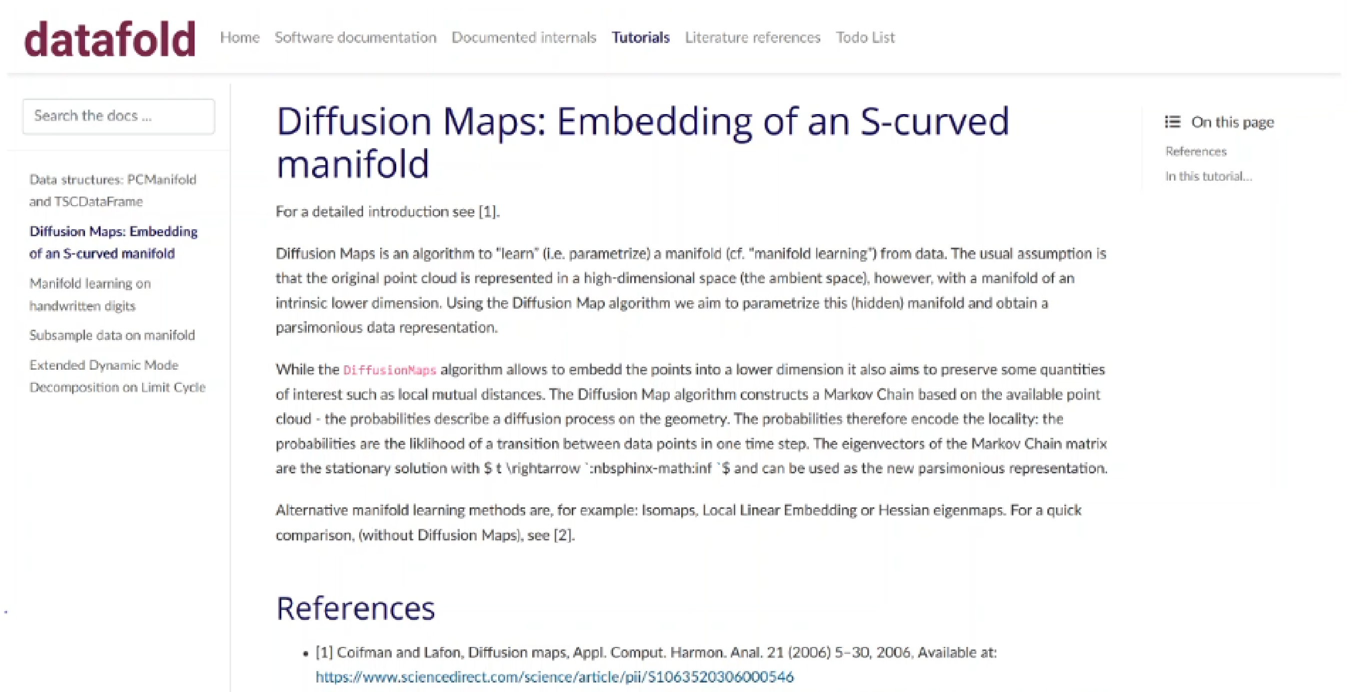
1. Form a distance matrix D with entries
$$D_{ij} = \|y_i - y_j\|,$$
where $i = 1, \ldots, N$ are the rows, $j = 1, \ldots, N$ are the columns, and y_i, y_j are the data points.
2. Set ε to 5% of the diameter of the dataset: $\varepsilon = 0.05(\max_{i,j} D_{i,j})$.
3. Form the kernel matrix W with $W_{ij} = \exp\left(-D_{ij}^2/\varepsilon\right)$.
4. Form the diagonal normalization matrix $P_{ii} = \sum_{j=1}^N W_{ij}$.
5. Normalize to form the kernel matrix $K = P^{-1} W P^{-1}$.
6. Form the diagonal normalization matrix $Q_{ii} = \sum_{j=1}^N K_{ij}$.
7. Form the symmetric matrix $\hat{T} = Q^{-1/2} K Q^{-1/2}$.
8. Find the $L+1$ largest eigenvalues a_l and associated eigenvectors v_l of $\hat{T}$.
9. Compute the eigenvalues of $\hat{T}^{1/\varepsilon}$ by $\lambda_l^2 = a_l^{1/\varepsilon}$.
10. Compute the eigenvectors of the matrix $T = Q^{-1} K$ by $\phi_l = Q^{-1/2} v_l$.

Steps 1-3 form the ambient kernel, 4-7 normalize it, 8-10 compute the eigenvalues and -vectors.

Representation of data

The datafold software

https://pypi.org/project/datafold/


datafold Home Software documentation Documented internals Tutorials Literature references Todo List

Search the docs ...

Data structures: PCManifold and TSCDataFrame
Diffusion Maps: Embedding of an S-curved manifold
Manifold learning on handwritten digits
Subsample data on manifold
Extended Dynamic Mode Decomposition on Limit Cycle

On this page
References
In this tutorial...

Diffusion Maps: Embedding of an S-curved manifold

For a detailed introduction see [1].

Diffusion Maps is an algorithm to "learn" (i.e. parametrize) a manifold (cf. "manifold learning") from data. The usual assumption is that the original point cloud is represented in a high-dimensional space (the ambient space), however, with a manifold of an intrinsic lower dimension. Using the Diffusion Map algorithm we aim to parametrize this (hidden) manifold and obtain a parsimonious data representation.

While the `DiffusionMaps` algorithm allows to embedd the points into a lower dimension it also aims to preserve some quantities of interest such as local mutual distances. The Diffusion Map algorithm constructs a Markov Chain based on the available point cloud - the probabilities describe a diffusion process on the geometry. The probabilities therefore encode the locality: the probabilities are the liklihood of a transition between data points in one time step. The eigenvectors of the Markov Chain matrix are the stationary solution with $ t \rightarrow `:nbsphinx-math:inf `$ and can be used as the new parsimonious representation.

Alternative manifold learning methods are, for example: Isomaps, Local Linear Embedding or Hessian eigenmaps. For a quick comparison, (without Diffusion Maps), see [2].

References

- [1] Coifman and Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (2006) 5–30, 2006, Available at: https://www.sciencedirect.com/science/article/pii/S1063520306000546

See documentation here: `https://datafold-dev.gitlab.io/datafold/index.html`

Literature I

Berry, T., Cressman, J. R., Gregurić-Ferenĉek, Z., and Sauer, T. (2013).
Time-scale separation from diffusion-mapped delay coordinates.
SIAM Journal on Applied Dynamical Systems, 12(2):618–649.

Coifman, R. R. and Lafon, S. (2006).
Diffusion maps.
Applied and Computational Harmonic Analysis, 21(1):5–30.

Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S. W. (2005).
Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps.
Proceedings of the National Academy of Sciences of the United States of America, 102(21):7426–7431.

Dietrich, F. (2017).
Data-Driven Surrogate Models for Dynamical Systems.
phdthesis.

Lee, J. M. (2012).
Introduction to Smooth Manifolds.
Springer New York.

Nadler, B., Lafon, S., Coifman, R. R., and Kevrekidis, I. G. (2006).
Diffusion maps, spectral clustering and reaction coordinates of dynamical systems.
Applied and Computational Harmonic Analysis, 21(1):113–127.