A tutorial on Manifold Learning for real data

The Fields Institute Workshop on Manifold and Graph-based learning

Marina Meilă

Department of Statistics
University of Washington

19-20 May, 2022

Outline

Outline

What is manifold learning good for?

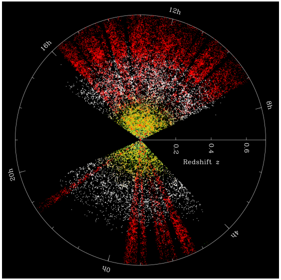
- Principal Component Analysis (PCA). What is it good for? = Linear Dim. Reduction

· High → low dim (save space, processing time, ...)

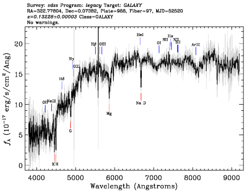
· understand ↔ more "relevant" features

Spectra of galaxies measured by the Sloan Digital Sky Survey (SDSS)

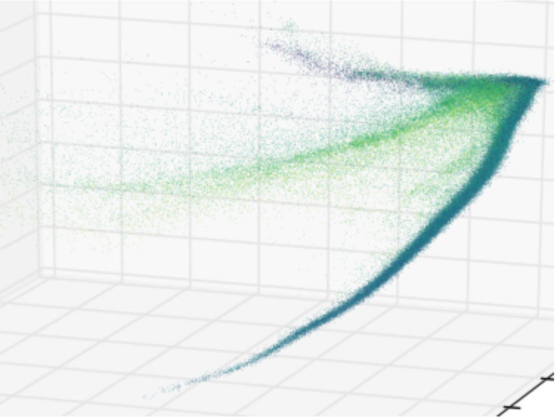
www.sdss.org



www.sdss.org



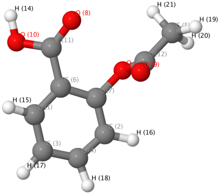
- Preprocessed by Jacob VanderPlas and Grace Telford
- $n = 675,000$ spectra $\times$ $D = 3750$ dimensions

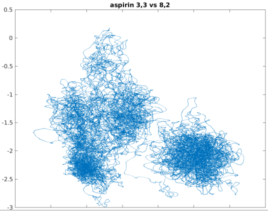


embedding by James McQueen

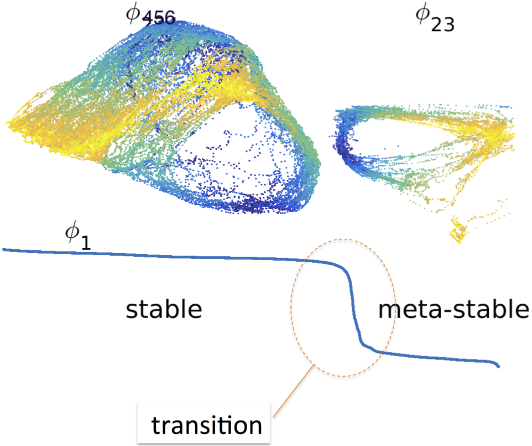
Molecular configurations

aspirin molecule



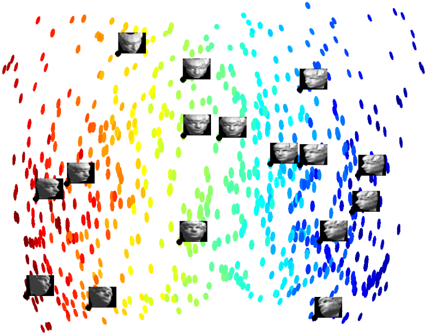


- Data from Molecular Dynamics (MD) simulations of small molecules by [Chmiela et al. 2016]
- $n \approx 200,000$ configurations $\times$ $D \sim 20 - 60$ dimensions



When to do (non-linear) dimension reduction

- $n = 698$ gray images of faces in $D = 64 \times 64$ dimensions
- head moves up/down and right/left
- With only two degrees of freedom, the faces define a 2D manifold in the space of all 64×64 gray images

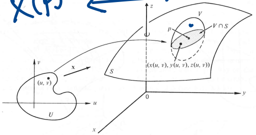


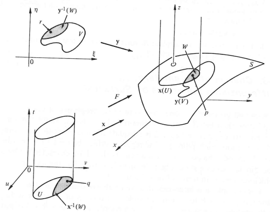
Outline

Manifold. Basic definitions

- **manifold** $\mathcal{M}$ = set that can locally be "$\approx$" $\mathbb{R}^d$ (like)
- **chart** $\mathcal{M} \supset U \xrightarrow{x} V \subseteq \mathbb{R}^d$ — U neighborhood — x = coordinate chart; x diffeomorphism, x^{-1} exists, differentiable
- **atlas** = {all charts}
- d is called **intrinsic dimension** of $\mathcal{M}$
- If the original data $p \in \mathbb{R}^D$, call D the **ambient dimension**.

$x(p) \in \mathbb{R}^d \longleftarrow p$





Intrinsic dimension. Tangent subspace

$p \in \mathcal{M}$

$T_p\mathcal{M} \cong \mathbb{R}^d$ vector space

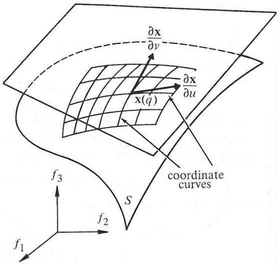
$T\mathcal{M} = \{ T_p\mathcal{M},\ p \in \mathcal{M} \}$ tangent bundle

$T_p\mathcal{M} = \{$ tangents to curves in $\mathcal{M} \}$

$\underbrace{\left\{ \frac{\partial x}{\partial u}(p), \frac{\partial x}{\partial v}(p) \right\}}_{\text{basis}} \in T_p\mathcal{M}$

$x(p) \in \mathbb{R}^d$

$x = \begin{bmatrix} u \\ v \end{bmatrix}$



Embeddings

- One can circumvent using multiple charts by mapping the data into $m > d$ dimensions.
- Let $\phi : \mathcal{M} \rightarrow \mathbb{R}^m$ be a smooth function, and let $\mathcal{N} = \phi(\mathcal{M})$.
- ϕ is an **embedding** if the inverse $\phi^{-1} : \mathcal{N} \rightarrow \mathcal{M}$ exists and is differentiable (a diffeormorphism).

data $\mathbb{R}^D \xrightarrow{\varphi} \mathbb{R}^m$

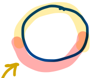
- Whitney's Embedding Theorem (**?**) states that any d-dimensional smooth manifold can be embedded into $\mathbb{R}^{2d}$.
- Hence, if $d \ll D$, very significant dimension reductions can be achieved with a single map $\phi : \mathcal{M} \rightarrow \mathbb{R}^m$.
- **Manifold learning algorithms** aim to construct maps ϕ like the above from finite data sampled from $\mathcal{M}$.

Embedding algorithm

Examples of manifolds and coordinate charts

John Lee – Smooth Manifolds
Riem. Manifold

- $\mathbb{R}^d$ $\quad \dim \mathbb{R}^d = d$

- 

  S^1
  $\dim S^1 = 1$
  circle
  needs 2 charts

  2 charts
  sphere S^2

  S^d sphere of $\dim = d$

- embedded in $\mathbb{R}^{d+1}$
  $m \geq d+1$

- 2 charts

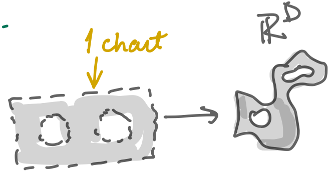
  T^2
  torus
  generated by 2 circles

  T^3
  3-torus
  $\downarrow$
  3 circles . . .

  , . . .

  $\dim T^d = d$
  $m = d+1$

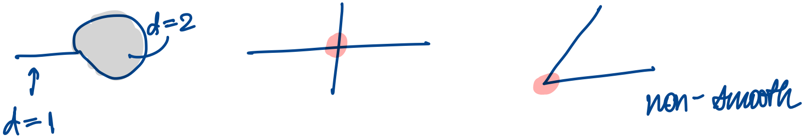
- subset of $\mathbb{R}^d$ mapped in $\mathbb{R}^q$



Examples of manifolds and coordinate charts

$\xi \in \mathbb{R}^D$ $\qquad S^d = \{ \|\xi_{1:d+1}\| = 1 \}$

Examples of manifolds and coordinate charts



Not manifolds

- dimension not constant
- unions of manifolds that intersect
- sharp corners (non-smooth)
- many/most neural network embeddings
- manifolds can have border

Outline

Non-linear dimension reduction: Three principles

Algorithm given $\mathcal{D} = \{\xi_1, \ldots \xi_n\}$ from $\mathcal{M} \subset \mathbb{R}^D$, map them by Algorithm f to $\{y_1, \ldots y_n\} \subset \mathbb{R}^m$
Assumption if points from $\mathcal{M}$, $n \to \infty$, f is embedding of $\mathcal{M}$ (f "recovers" $\mathcal{M}$ of arbitrary shape).

1. Local (weighted) PCA (lPCA)
2. Principal Curves and Surfaces (PCS)
3. Embedding algorithms (Diffusion Maps/Laplacian Eigenmaps, Isomap, LTSA, MVU, Hessian Eigenmaps,...)
4. [Other, heuristic] t-SNE, UMAP, LLE

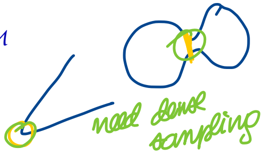
What makes the problem hard?

- Intrinsic dimension d
  - must be estimated (we assume we know it) (Lecture 3)
  - sample complexity is exponential in d – **NONPARAMETRIC** (upcoming)
- non-uniform sampling
- **volume** of $\mathcal{M}$ (we assume volume finite; larger volume requires more samples)
- **injectivity radius/reach** of $\mathcal{M}$ (next page)
- curvature
- ESSENTIAL smoothness parameter: the **neighborhood radius** (Lecture 3)

Non-linear dimension reduction: Three principles

~ sampling distribution $p()$

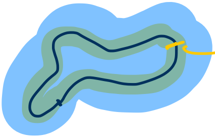
Algorithm given $\mathcal{D} = \{\xi_1, \dots \xi_n\}$ from $\mathcal{M} \subset \mathbb{R}^D$, map them by Algorithm f to $\{y_1, \dots y_n\} \subset \mathbb{R}^m$
Assumption if points from $\mathcal{M}$, $n \to \infty$, f is embedding of $\mathcal{M}$
(f "recovers" $\mathcal{M}$ of arbitrary shape).



need dense sampling!!

1. Local (weighted) PCA (lPCA)
2. Principal Curves and Surfaces (PCS)
3. Embedding algorithms (Diffusion Maps/Laplacian Eigenmaps, Isomap, LTSA, MVU, Hessian Eigenmaps,...)

4. [Other, heuristic] t-SNE, UMAP, LLE



r of this tube $= \tau$

What makes the problem hard?

- Intrinsic dimension d
  - must be estimated (we assume we know it) (Lecture 3)
  - sample complexity is exponential in d – **NONPARAMETRIC** (upcoming)
- non-uniform sampling
- **volume** of $\mathcal{M}$ (we assume volume finite; larger volume requires more samples)
- **injectivity radius/reach** of $\mathcal{M}$ $= \tau$ (next page)
- curvature

- ESSENTIAL smoothness parameter: the **neighborhood radius** (Lecture 3)

Parametric vs. non-parametric

An example of density estimation with data $x_{1:n} \in \mathbb{R}$.

1. Gaussian $N(\mu, \sigma^2)$ **parametric**.
   - $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $\hat{\sigma^2} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$
   - Error $\mu - \hat{\mu}$ has mean 0 and standard deviation $\sigma_{\hat{\mu}} = \frac{\sigma}{\sqrt{n}} \propto n^{-1/2}$
   - To increase accuracy $\times 10$, n must increase $\times 10^2 = 100$
2. Kernel density estimation (KDE), **non-parametric**

$$p_h(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}\kappa\left(\frac{x_i - x}{h}\right)$$

   - $\kappa = N(0,1)$ the kernel, $h > 0$ is the kernel width
   - Accuracy for KDE $\propto n^{-2/5}$
   - To increase accuracy $\times 10$, n must increase $\times 10^{5/2} \approx 316$

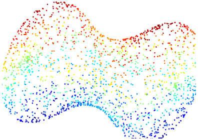
Model	e.g.	distribution shape	error rate	to decrease err. by 10 we need samples $\times$	
Parametric	$N(\mu, \sigma^2))$	fixed	$n^{-1/2}$	$n \times 10^2$	100
Non-parametric	KDE in $\mathbb{R}$	any	$n^{-2/5}$	$n \times 10^{5/2}$	316
	KDE in $\mathbb{R}^d$	any	$n^{-2/(d+4)}$	$n \times 10^{(d+4)/2}$	1000 ($d = 2$)
					3163 ($d = 3$)
					10,000 ($d = 4$)

Neighborhood graphs

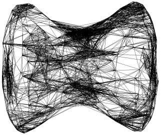
- All ML algorithms start with a **neighborhood graph** over the data points
  - neigh_i denotes the neighbors of ξ_i, and $k_i = |\text{neigh}_i|$.
  - $\Xi_i = [\xi_{i'}]_{i' \in \text{neigh}_i} \in \mathbb{R}^{D \times k_i}$ contains the coordinates of ξ_i's neighbors
- In the **radius-neighbor** graph, the neighbors of ξ_i are the points within distance r from ξ_i, i.e. in the ball $B_r(\xi_i)$.
- In the **k-nearest-neighbor (k-nn)** graph, they are the k nearest-neighbors of ξ_i.

- k-nn graph has many computational advantages
  - constant degree k (or $k-1$)
  - connected for any $k > 1$
  - more software available

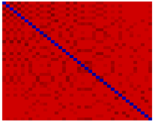
  - but much more difficult to use for consistent estimation of manifolds (see later, and)



data $\xi_1, \ldots \xi_n \subset \mathbb{R}^D$



neighborhood graph



A (sparse) matrix of distances between neighbors