Efficient manifold and subspace approximations with spherelets

Didong Li

Departments of Mathematics
Duke University

didongli@math.duke.edu

July 6, 2018

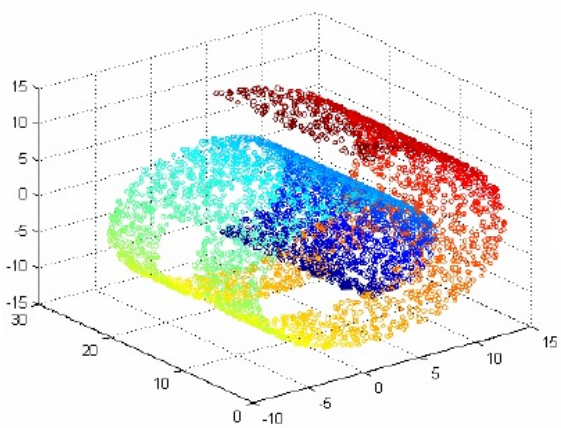
Joint work with Minerva Mukhopadhyay and David Dunson

Duke
UNIVERSITY

Overview

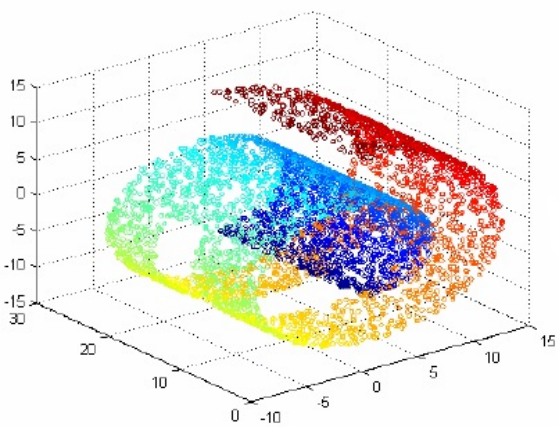
1. Background and Motivation
2. Low dimensional geometric object: spherelets
   - New Dictionary
   - Main Theorem
   - Spherical principal component analysis (SPCA)
   - Convergence Analysis
   - Spherelets Algorithm & Examples
3. Bayesian approach: mixture of spherelets

Background



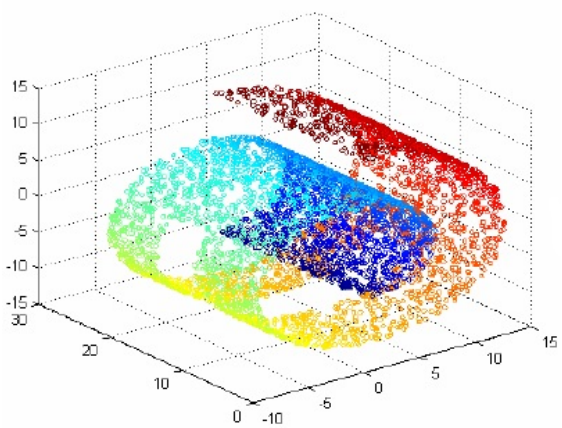
- (Of course) very common to collect high-dimensional data

Background



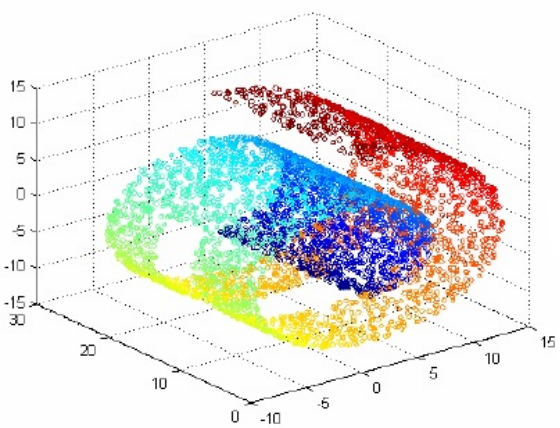
- (Of course) very common to collect high-dimensional data
- Let p = ambient dimension of data & n = sample size

Background



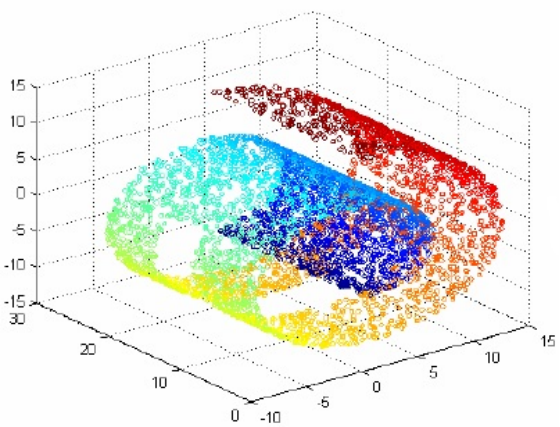
- (Of course) very common to collect high-dimensional data
- Let p = ambient dimension of data & n = sample size
- If $p \gg n$, we need to exploit lower-dimensional structure in the data

Background



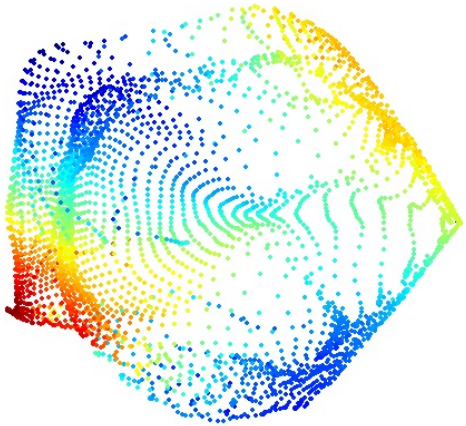
- (Of course) very common to collect high-dimensional data
- Let p = ambient dimension of data & n = sample size
- If $p \gg n$, we need to exploit lower-dimensional structure in the data
- Common to suppose data do not live everywhere in p-dimensional space

Background



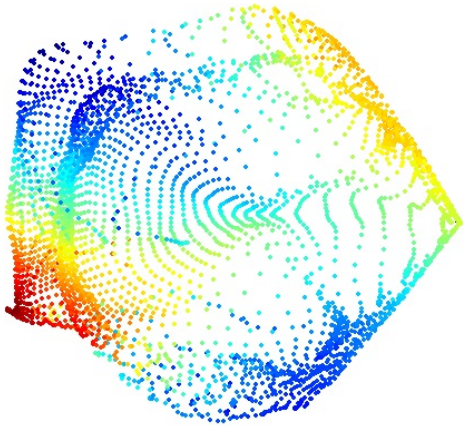
- (Of course) very common to collect high-dimensional data
- Let p = ambient dimension of data & n = sample size
- If $p \gg n$, we need to exploit lower-dimensional structure in the data
- Common to suppose data do not live everywhere in p-dimensional space
- May be concentrated near a *subspace* $\mathcal{M}$ having dimension d with $d \ll p$

Motivation



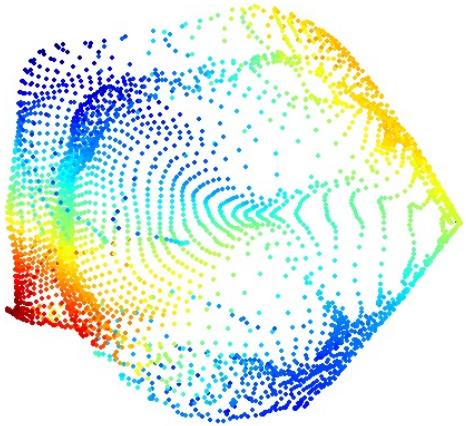
- Suppose $X_i = (X_{i1}, \ldots, X_{ip})^T \in \mathcal{M} \subset \mathbb{R}^p$, X_i are i.i.d. samples from density ρ, where $\text{supp}(\rho) = \mathcal{M}$, $\dim(\mathcal{M}) = d \ll p$

Motivation



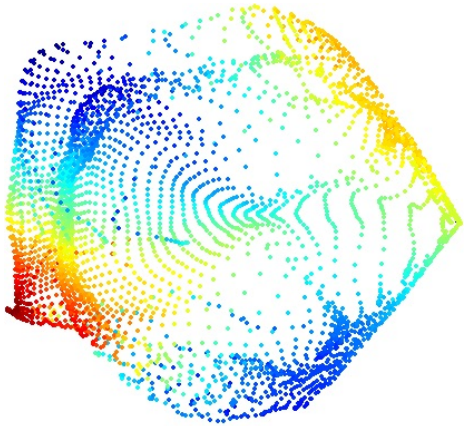
- Suppose $X_i = (X_{i1}, \ldots, X_{ip})^T \in \mathcal{M} \subset \mathbb{R}^p$, X_i are i.i.d. samples from density ρ, where $\text{supp}(\rho) = \mathcal{M}$, $\dim(\mathcal{M}) = d \ll p$
- $\mathcal{M}$ = *unknown* support having intrinsic dimension d

Motivation



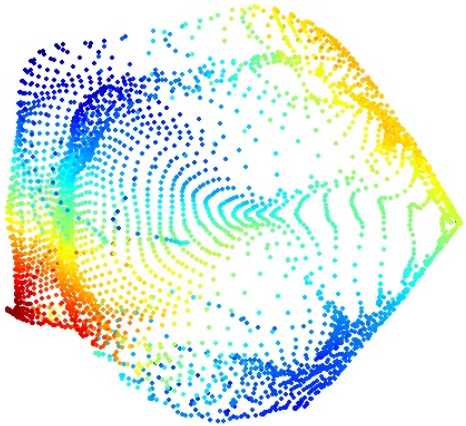
- Suppose $X_i = (X_{i1}, \ldots, X_{ip})^T \in \mathcal{M} \subset \mathbb{R}^p$, X_i are i.i.d. samples from density ρ, where $\text{supp}(\rho) = \mathcal{M}$, $\dim(\mathcal{M}) = d \ll p$
- $\mathcal{M}$ = *unknown* support having intrinsic dimension d
- Hence, we have a doubly nasty problem

Motivation



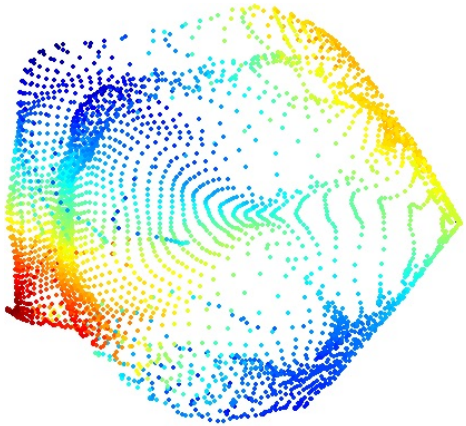
- Suppose $X_i = (X_{i1}, \ldots, X_{ip})^T \in \mathcal{M} \subset \mathbb{R}^p$, X_i are i.i.d. samples from density ρ, where $\text{supp}(\rho) = \mathcal{M}$, $\dim(\mathcal{M}) = d \ll p$
- $\mathcal{M}$ = *unknown* support having intrinsic dimension d
- Hence, we have a doubly nasty problem
- We don't know the density of the data (*density estimation in high-dimensions*)

Motivation



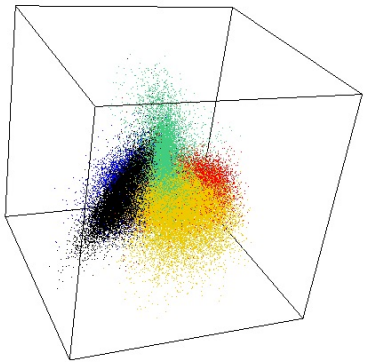
- Suppose $X_i = (X_{i1}, \ldots, X_{ip})^T \in \mathcal{M} \subset \mathbb{R}^p$, X_i are i.i.d. samples from density ρ, where $\text{supp}(\rho) = \mathcal{M}$, $\dim(\mathcal{M}) = d \ll p$
- $\mathcal{M}$ = *unknown* support having intrinsic dimension d
- Hence, we have a doubly nasty problem
- We don't know the density of the data (*density estimation in high-dimensions*)
- We also don't know the support of this density (*subspace learning*)

Motivation



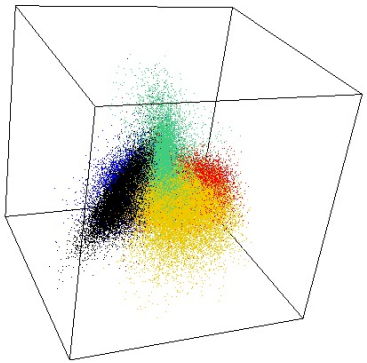
- Suppose $X_i = (X_{i1}, \ldots, X_{ip})^T \in \mathcal{M} \subset \mathbb{R}^p$, X_i are i.i.d. samples from density ρ, where $\text{supp}(\rho) = \mathcal{M}$, $\dim(\mathcal{M}) = d \ll p$
- $\mathcal{M}$ = *unknown* support having intrinsic dimension d
- Hence, we have a doubly nasty problem
- We don't know the density of the data (*density estimation in high-dimensions*)
- We also don't know the support of this density (*subspace learning*)
- Many relevant algorithms

Common Approach



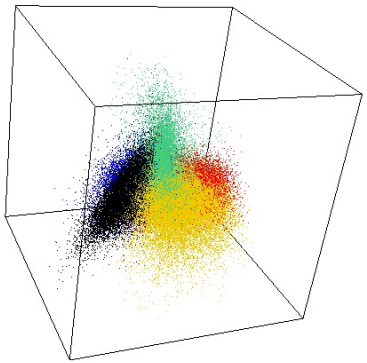
- First estimate coordinates on a low-dimensional subspace $X_i \to \eta_i$

Common Approach



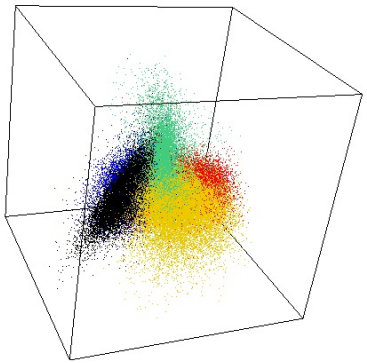
- First estimate coordinates on a low-dimensional subspace $X_i \rightarrow \eta_i$
- Often PCA is applied to estimate η_i

Common Approach



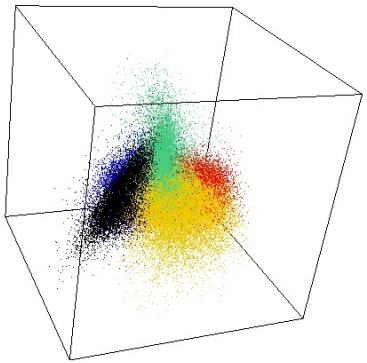
- First estimate coordinates on a low-dimensional subspace $X_i \to \eta_i$
- Often PCA is applied to estimate η_i
- Then in a second stage one can estimate the density of η_i

Common Approach



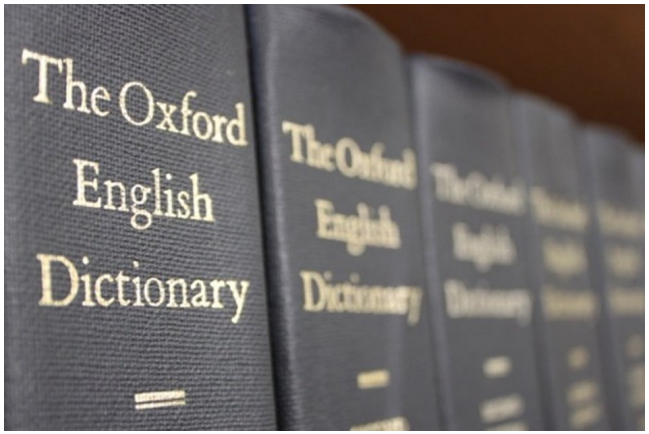
- First estimate coordinates on a low-dimensional subspace $X_i \to \eta_i$
- Often PCA is applied to estimate η_i
- Then in a second stage one can estimate the density of η_i
- The first stage is commonly referred to as *manifold learning*

Common Approach



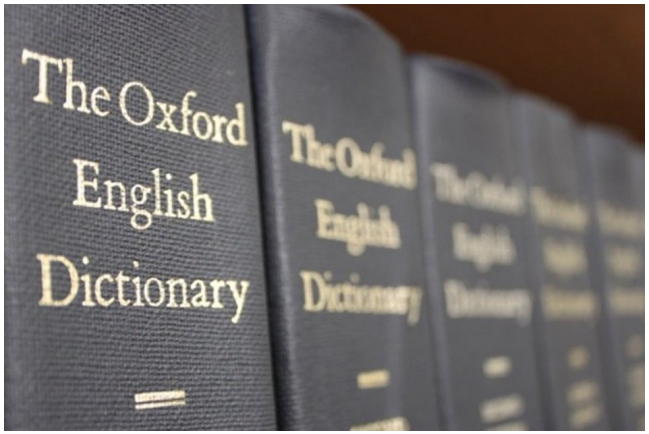
- First estimate coordinates on a low-dimensional subspace $X_i \to \eta_i$
- Often PCA is applied to estimate η_i
- Then in a second stage one can estimate the density of η_i
- The first stage is commonly referred to as *manifold learning*
- Assume that the subspace is either a smooth manifold or a collection of such manifolds

Dictionaries for Subspaces



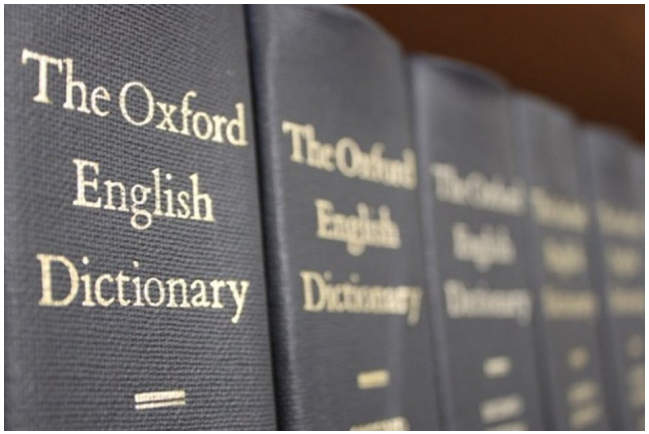
- Machine learning algorithms usually require some sort of *dictionary* to use in approximating the subspace $\mathcal{M}$

Dictionaries for Subspaces



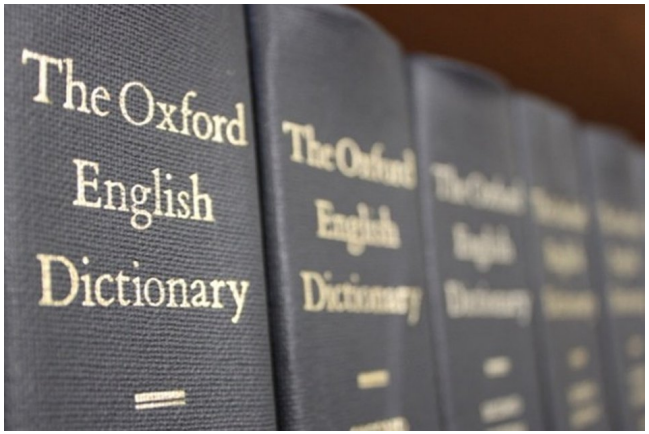
- Machine learning algorithms usually require some sort of *dictionary* to use in approximating the subspace $\mathcal{M}$
- If $\mathcal{M}$ is linear, then methods such as PCA, SVD, ICA & factor analysis can be used

Dictionaries for Subspaces



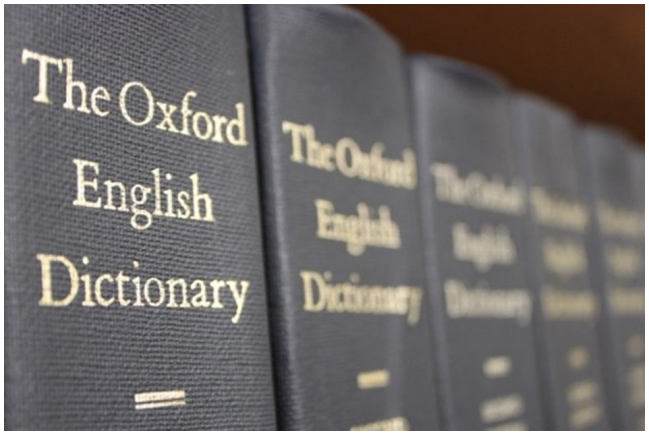
- Machine learning algorithms usually require some sort of *dictionary* to use in approximating the subspace $\mathcal{M}$
- If $\mathcal{M}$ is linear, then methods such as PCA, SVD, ICA & factor analysis can be used
- Of course linear $\mathcal{M}$ is much too restrictive in many applications

Dictionaries for Subspaces



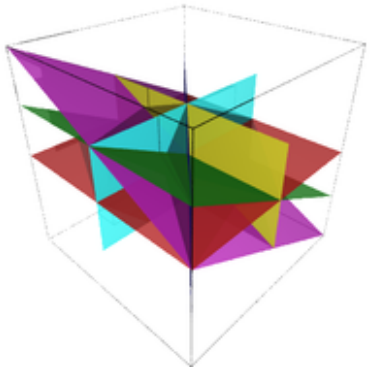
- Machine learning algorithms usually require some sort of *dictionary* to use in approximating the subspace $\mathcal{M}$
- If $\mathcal{M}$ is linear, then methods such as PCA, SVD, ICA & factor analysis can be used
- Of course linear $\mathcal{M}$ is much too restrictive in many applications
- $\mathcal{M}$ may have substantial curvature, potentially even with the curvature varying over $\mathcal{M}$

Dictionaries for Subspaces



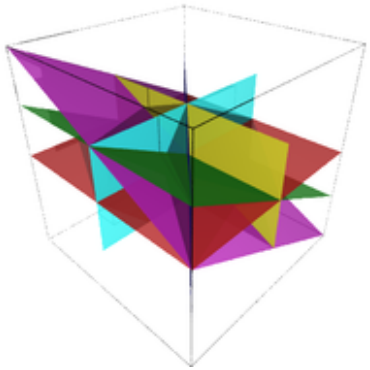
- Machine learning algorithms usually require some sort of *dictionary* to use in approximating the subspace $\mathcal{M}$
- If $\mathcal{M}$ is linear, then methods such as PCA, SVD, ICA & factor analysis can be used
- Of course linear $\mathcal{M}$ is much too restrictive in many applications
- $\mathcal{M}$ may have substantial curvature, potentially even with the curvature varying over $\mathcal{M}$
- How to approximate arbitrary non-linear subspaces?

Locally Linear Approaches



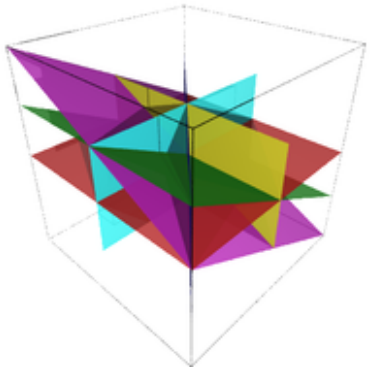
- It is extremely common in this setting to use locally linear approaches

Locally Linear Approaches



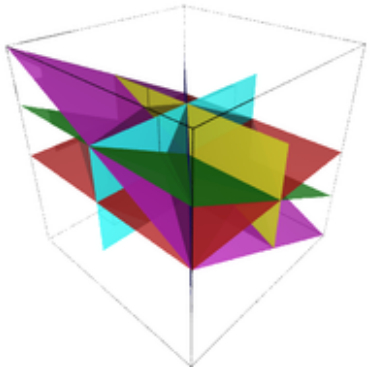
- It is extremely common in this setting to use locally linear approaches
- If $\mathcal{M}$ is a Riemannian manifold, can be motivated by thinking of a collection of tangent plane approximations

Locally Linear Approaches



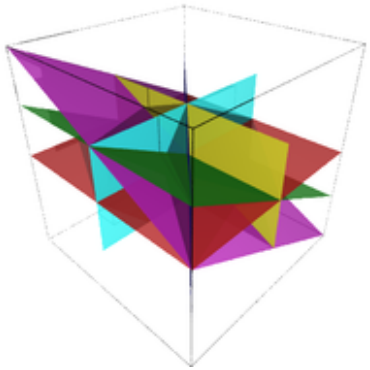
- It is extremely common in this setting to use locally linear approaches
- If $\mathcal{M}$ is a Riemannian manifold, can be motivated by thinking of a collection of tangent plane approximations
- Locally linear embeddings (LLE), Diffusion Map, EigenMap, tSNE, etc

Locally Linear Approaches



- It is extremely common in this setting to use locally linear approaches
- If $\mathcal{M}$ is a Riemannian manifold, can be motivated by thinking of a collection of tangent plane approximations
- Locally linear embeddings (LLE), Diffusion Map, EigenMap, tSNE, etc
- Local PCA, including Multiscale analysis of plane arrangements and Geometric Multi-Resolution Analysis (GMRA)

Locally Linear Approaches



- It is extremely common in this setting to use locally linear approaches
- If $\mathcal{M}$ is a Riemannian manifold, can be motivated by thinking of a collection of tangent plane approximations
- Locally linear embeddings (LLE), Diffusion Map, EigenMap, tSNE, etc
- Local PCA, including Multiscale analysis of plane arrangements and Geometric Multi-Resolution Analysis (GMRA)

Pros and Cons of Current Approaches

Pros

- Use simple linear pieces so conceptually easy
- Can potentially have good computational efficiency

Pros and Cons of Current Approaches

Pros

- Use simple linear pieces so conceptually easy
- Can potentially have good computational efficiency

Cons

- Tend to find too many pieces when the manifold has large curvature

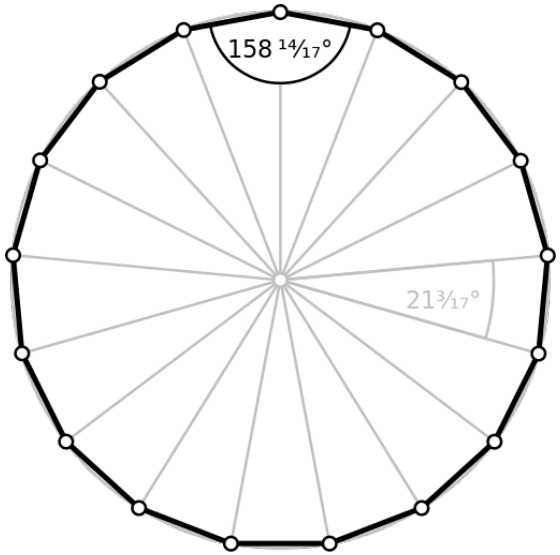
Pros and Cons of Current Approaches

Pros

- Use simple linear pieces so conceptually easy
- Can potentially have good computational efficiency

Cons

- Tend to find too many pieces when the manifold has large curvature



New dictionary

- First order $\longrightarrow$ second order: $x^\top Hx + f^\top x + c = 0$.

New dictionary

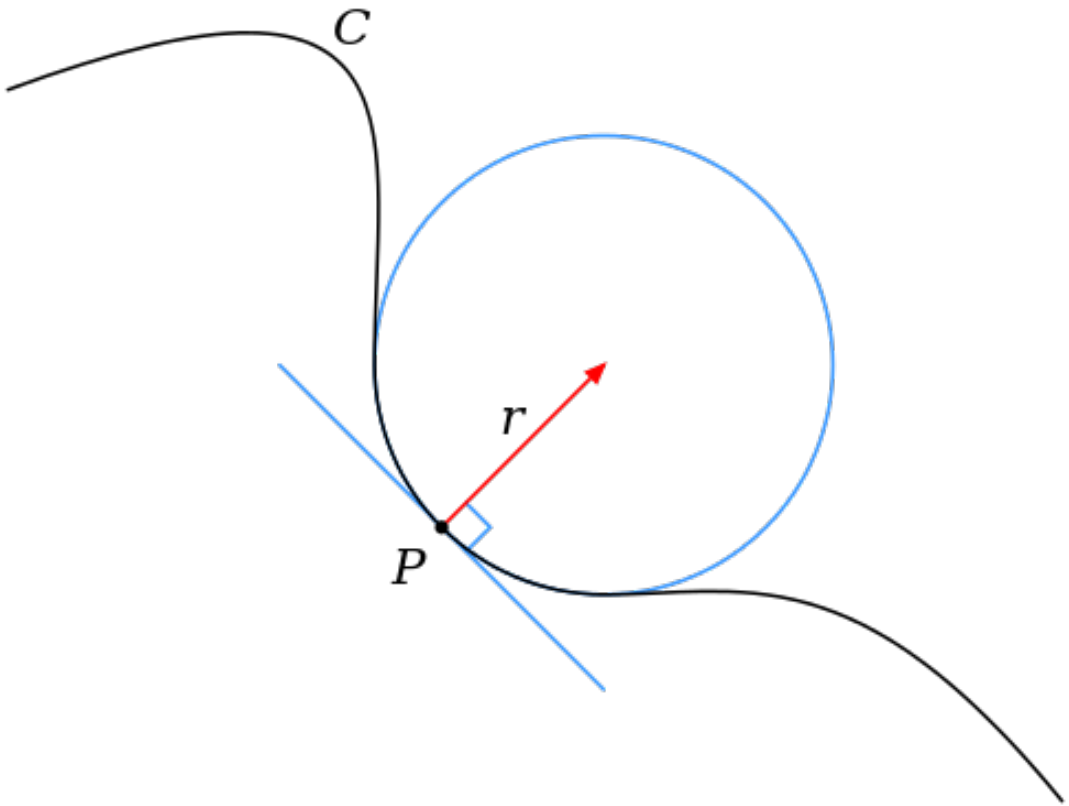
- First order $\longrightarrow$ second order: $x^\top Hx + f^\top x + c = 0$.
- Number of unknown parameters = $\frac{p(p+1)}{2} + p + 1 = O(p^2)$.

New dictionary

- First order $\longrightarrow$ second order: $x^\top Hx + f^\top x + c = 0$.
- Number of unknown parameters = $\frac{p(p+1)}{2} + p + 1 = O(p^2)$.
- Trades one problem (*too many pieces*) for another (*too many parameters*)

New dictionary

- First order $\longrightarrow$ second order: $x^\top Hx + f^\top x + c = 0$.
- Number of unknown parameters = $\frac{p(p+1)}{2} + p + 1 = O(p^2)$.
- Trades one problem (*too many pieces*) for another (*too many parameters*)
- An alternative is osculating circles/spheres



Using spheres to locally approximate subspaces

Why spheres?

Using spheres to locally approximate subspaces

Why spheres?

- Compactness

Using spheres to locally approximate subspaces

Why spheres?

- Compactness
- Hyperplane=sphere with infinite radius (compactification)

Using spheres to locally approximate subspaces

Why spheres?

- Compactness
- Hyperplane=sphere with infinite radius (compactification)
- Projection to sphere is easy to compute

Using spheres to locally approximate subspaces

Why spheres?

- Compactness
- Hyperplane=sphere with infinite radius (compactification)
- Projection to sphere is easy to compute
- Cell complex structure: $S^d = S^{d-1} \cup e_1^d \cup e_2^d$

Using spheres to locally approximate subspaces

Why spheres?

- Compactness
- Hyperplane=sphere with infinite radius (compactification)
- Projection to sphere is easy to compute
- Cell complex structure: $S^d = S^{d-1} \cup e_1^d \cup e_2^d$

Spherelets - A dictionary for subspaces

- We propose to use pieces of spheres or *spherelets* as a dictionary

Spherelets - A dictionary for subspaces

- We propose to use pieces of spheres or *spherelets* as a dictionary
- Often *many* fewer spheres than planes to obtain the same approximation error

Spherelets - A dictionary for subspaces

- We propose to use pieces of spheres or *spherelets* as a dictionary
- Often *many* fewer spheres than planes to obtain the same approximation error
- Each sphere has few parameters & they are simple geometric objects that are easy to fit

Spherelets - A dictionary for subspaces

- We propose to use pieces of spheres or *spherelets* as a dictionary
- Often *many* fewer spheres than planes to obtain the same approximation error
- Each sphere has few parameters & they are simple geometric objects that are easy to fit
- Before considering algorithms for fitting spherelets, we studied their approximation properties

Notation and concepts

- $\mathcal{M}$ is a compact C^3, d-dimensional orientable manifold embedded in $\mathbb{R}^p$

Notation and concepts

- $\mathcal{M}$ is a compact C^3, d-dimensional orientable manifold embedded in $\mathbb{R}^p$
- Trivial to extend our results to a collection of such manifolds

Notation and concepts

- $\mathcal{M}$ is a compact C^3, d-dimensional orientable manifold embedded in $\mathbb{R}^p$
- Trivial to extend our results to a collection of such manifolds
- We want to bound # pieces needed to obtain approximation error ϵ

Notation and concepts

- $\mathcal{M}$ is a compact C^3, d-dimensional orientable manifold embedded in $\mathbb{R}^p$
- Trivial to extend our results to a collection of such manifolds
- We want to bound # pieces needed to obtain approximation error ϵ
- $N_H(\epsilon, \mathcal{M})$ = minimal # hyperplanes, $N_S(\epsilon, \mathcal{M})$ = minimal # spheres

Notation and concepts

- $\mathcal{M}$ is a compact C^3, d-dimensional orientable manifold embedded in $\mathbb{R}^p$
- Trivial to extend our results to a collection of such manifolds
- We want to bound # pieces needed to obtain approximation error ϵ
- $N_H(\epsilon, \mathcal{M})$ = minimal # hyperplanes, $N_S(\epsilon, \mathcal{M})$ = minimal # spheres
- K=max curvature, T=maximum rate of change in curvature, V = Vol($\mathcal{M}$).

Main Theorem

Theorem

1. *The bound on the hyperplane covering number is*

$$N_H(\epsilon, \mathcal{M}) \leq V\left(\frac{2\epsilon}{K}\right)^{-\frac{d}{2}}$$

Main Theorem

Theorem

1. *The bound on the hyperplane covering number is*

$$N_H(\epsilon, \mathcal{M}) \leq V\left(\frac{2\epsilon}{K}\right)^{-\frac{d}{2}}$$

2. *Let* $F_\epsilon := \{p \in \mathcal{M} : |k_1(p) - k_d(p)| \leq (\frac{2\epsilon}{K})^{\frac{1}{2}}\}$, *where* $k_1(p)$ *and* $k_d(p)$ *are the max & min principal curvature of* $\mathcal{M}$ *at p. Let* $\mathcal{M}_\epsilon := \bigcup_{p \in F_\epsilon} B\left(p, \left(\frac{6\epsilon}{3+T}\right)^{\frac{1}{3}}\right)$ *and* $V_\epsilon := \mathrm{Vol}(\mathcal{M}_\epsilon)$, *then*

$$N_S(\epsilon, \mathcal{M}) \leq V_\epsilon\left(\frac{6\epsilon}{3+T}\right)^{-\frac{d}{3}} + (V - V_\epsilon)\left(\frac{2\epsilon}{K}\right)^{-\frac{d}{2}}$$

Implications of the Theorem

- Since $\epsilon \approx 0$, $\epsilon^{-d/2}$ is very large showing the *curse of dimensionality*

Implications of the Theorem

- Since $\epsilon \approx 0$, $\epsilon^{-d/2}$ is very large showing the *curse of dimensionality*
- Even if an oracle could perfectly choose the pieces to best approximate $\mathcal{M}$, we need lots of pieces as d increases for small ϵ

Implications of the Theorem

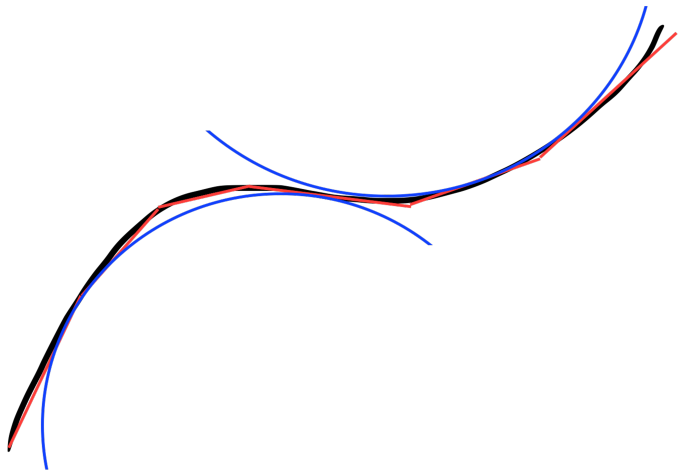
- Since $\epsilon \approx 0$, $\epsilon^{-d/2}$ is very large showing the *curse of dimensionality*
- Even if an oracle could perfectly choose the pieces to best approximate $\mathcal{M}$, we need lots of pieces as d increases for small ϵ
- Spherelets can decrease the impact of the curse to $\epsilon^{-d/3}$ IF

Implications of the Theorem

- Since $\epsilon \approx 0$, $\epsilon^{-d/2}$ is very large showing the *curse of dimensionality*
- Even if an oracle could perfectly choose the pieces to best approximate $\mathcal{M}$, we need lots of pieces as d increases for small ϵ
- Spherelets can decrease the impact of the curse to $\epsilon^{-d/3}$ IF
- There aren't too many locations $p \in \mathcal{M}$ having big changes in principal curvature

Implications of the Theorem

- Since $\epsilon \approx 0$, $\epsilon^{-d/2}$ is very large showing the *curse of dimensionality*
- Even if an oracle could perfectly choose the pieces to best approximate $\mathcal{M}$, we need lots of pieces as d increases for small ϵ
- Spherelets can decrease the impact of the curse to $\epsilon^{-d/3}$ IF
- There aren't too many locations $p \in \mathcal{M}$ having big changes in principal curvature



Spherical principal component analysis (SPCA)

Definition

$X \in \mathbb{R}^{n \times p}$,

Spherical principal component analysis (SPCA)

Definition

$X \in \mathbb{R}^{n \times p}$, $d \ll p$,

Spherical principal component analysis (SPCA)

Definition

$X \in \mathbb{R}^{n\times p}$, $d \ll p$, $Y_i = \bar{X} + \widehat{V}\widehat{V}^\top(X_i - \bar{X})$, $\widehat{V} = (v_1, \cdots, v_{d+1})$, $v_i = \mathrm{evec}_i\{(X - 1\bar{X}^\top)^\top(X - 1\bar{X}^\top)\}$, where $\mathrm{evec}_i(S)$ is the ith eigenvector of S in decreasing order.

Spherical principal component analysis (SPCA)

Definition

$X \in \mathbb{R}^{n\times p}$, $d \ll p$, $Y_i = \bar{X} + \widehat{V}\widehat{V}^\top(X_i - \bar{X})$, $\widehat{V} = (v_1, \cdots, v_{d+1})$, $v_i = \mathrm{evec}_i\{(X - 1\bar{X}^\top)^\top(X - 1\bar{X}^\top)\}$, where $\mathrm{evec}_i(S)$ is the ith eigenvector of S in decreasing order. $Z_i = \widehat{c} + \frac{\hat{r}}{\|Y_i - \widehat{c}\|}(Y_i - \widehat{c})$ is the d-dimensional spherical component of X,

Spherical principal component analysis (SPCA)

Definition

$X \in \mathbb{R}^{n\times p}$, $d \ll p$, $Y_i = \bar{X} + \widehat{V}\widehat{V}^\top(X_i - \bar{X})$, $\widehat{V} = (v_1, \cdots, v_{d+1})$, $v_i = \mathrm{evec}_i\{(X - 1\bar{X}^\top)^\top(X - 1\bar{X}^\top)\}$, where $\mathrm{evec}_i(S)$ is the ith eigenvector of S in decreasing order. $Z_i = \widehat{c} + \frac{\hat{r}}{\|Y_i - \widehat{c}\|}(Y_i - \widehat{c})$ is the d-dimensional spherical component of X, where $\hat{r} = \frac{1}{n}\sum_{i=1}^n \|Y_i - \widehat{c}\|$,

$$\widehat{c} = -\frac{1}{2}\left(\sum_{i=1}^n (\bar{Y} - Y_i)(\bar{Y} - Y_i)^\top\right)^{-1} \sum_{i=1}^n \left(\|Y_i^\top Y_i\| - \frac{1}{n}\sum_{j=1}^n \|Y_j^\top Y_j\|\right)(\bar{Y} - Y_i).$$

Spherical principal component analysis (SPCA)

Definition

$X \in \mathbb{R}^{n\times p}$, $d \ll p$, $Y_i = \bar{X} + \widehat{V}\widehat{V}^\top(X_i - \bar{X})$, $\widehat{V} = (v_1, \cdots, v_{d+1})$, $v_i = \mathrm{evec}_i\{(X - 1\bar{X}^\top)^\top(X - 1\bar{X}^\top)\}$, where $\mathrm{evec}_i(S)$ is the ith eigenvector of S in decreasing order. $Z_i = \widehat{c} + \frac{\hat{r}}{\|Y_i - \widehat{c}\|}(Y_i - \widehat{c})$ is the d-dimensional spherical component of X, where $\hat{r} = \frac{1}{n}\sum_{i=1}^n \|Y_i - \widehat{c}\|$,

$$\widehat{c} = -\frac{1}{2}\left(\sum_{i=1}^n (\bar{Y} - Y_i)(\bar{Y} - Y_i)^\top\right)^{-1} \sum_{i=1}^n \left(\|Y_i^\top Y_i\| - \frac{1}{n}\sum_{j=1}^n \|Y_j^\top Y_j\|\right)(\bar{Y} - Y_i).$$

- d-PSPCA = the projection of X to the "best" d dimensional sphere centered at c with radius r

Spherical principal component analysis (SPCA)

Definition

$X \in \mathbb{R}^{n\times p}$, $d \ll p$, $Y_i = \bar{X} + \widehat{V}\widehat{V}^\top(X_i - \bar{X})$, $\widehat{V} = (v_1, \cdots, v_{d+1})$, $v_i = \mathrm{evec}_i\{(X - 1\bar{X}^\top)^\top(X - 1\bar{X}^\top)\}$, where $\mathrm{evec}_i(S)$ is the ith eigenvector of S in decreasing order. $Z_i = \widehat{c} + \frac{\hat{r}}{\|Y_i - \widehat{c}\|}(Y_i - \widehat{c})$ is the d-dimensional spherical component of X, where $\hat{r} = \frac{1}{n}\sum_{i=1}^n \|Y_i - \widehat{c}\|$,

$$\widehat{c} = -\frac{1}{2}\left(\sum_{i=1}^n (\bar{Y} - Y_i)(\bar{Y} - Y_i)^\top\right)^{-1} \sum_{i=1}^n \left(\|Y_i^\top Y_i\| - \frac{1}{n}\sum_{j=1}^n \|Y_j^\top Y_j\|\right)(\bar{Y} - Y_i).$$

- d-PSPCA = the projection of X to the "best" d dimensional sphere centered at c with radius r
- Let (V^*, c^*, r^*) denote the values of $(\widehat{V}, \widehat{c}, \widehat{r})$ obtained plugging in exact moments of the population distribution in place of sample values.

Loss function

- SPCA minimizes the loss function

$$\sum_{i=1}^{n}(X_i^\top X_i + f^\top X_i + b)^2$$

where $\hat{f} = -2\hat{c}$ and $\hat{b} = \|\hat{c}\|^2 - \hat{r}^2$.

Loss function

- SPCA minimizes the loss function

$$\sum_{i=1}^{n}(X_i^\top X_i + f^\top X_i + b)^2$$

where $\hat{f} = -2\hat{c}$ and $\hat{b} = \|\hat{c}\|^2 - \hat{r}^2$.

- PCA minimizes the loss function

$$\sum_{i=1}^{n}(f^\top X_i + b)^2,$$

where $\hat{f}$ is the unit normal vector of the best d-dimensional affine subspace, or the eigenvector of covariance matrix corresponding to the smallest eigenvalue.

Loss function

- SPCA minimizes the loss function

$$\sum_{i=1}^{n}(X_i^\top X_i + f^\top X_i + b)^2$$

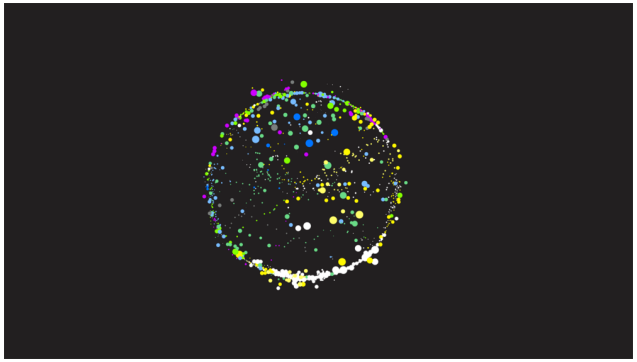
where $\hat{f} = -2\hat{c}$ and $\hat{b} = \|\hat{c}\|^2 - \hat{r}^2$.

- PCA minimizes the loss function

$$\sum_{i=1}^{n}(f^\top X_i + b)^2,$$

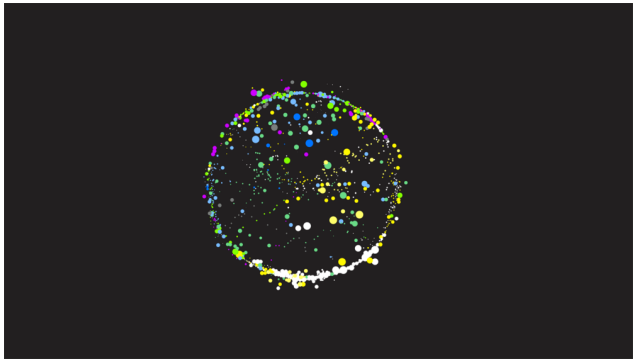
where $\hat{f}$ is the unit normal vector of the best d-dimensional affine subspace, or the eigenvector of covariance matrix corresponding to the smallest eigenvalue.

Spherical projection



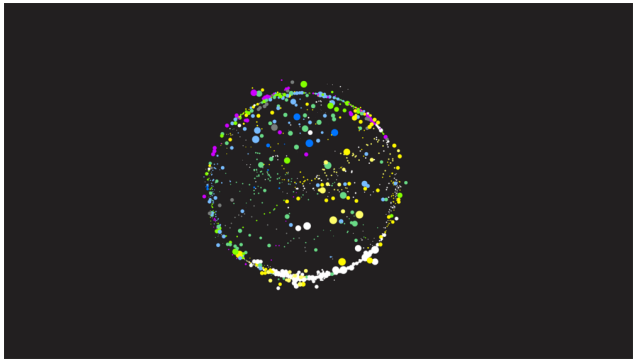
- $\widehat{\mathrm{Proj}}_n(x) := \hat{c} + \frac{\hat{r}}{\|\widehat{V}\widehat{V}^\top(x-\hat{c})\|}\widehat{V}\widehat{V}^\top(x-\hat{c})$ is the spherical projection to $S_{\widehat{V}}(\hat{c},\hat{r})$, where n is the sample size

Spherical projection



- $\widehat{\mathrm{Proj}}_n(x) := \hat{c} + \frac{\hat{r}}{\|\widehat{V}\widehat{V}^\top(x-\hat{c})\|}\widehat{V}\widehat{V}^\top(x-\hat{c})$ is the spherical projection to $S_{\widehat{V}}(\hat{c},\hat{r})$, where n is the sample size
- $\mathrm{Proj}^*(x) := c^* + \frac{r^*}{\|V^*V^{*\top}(x-c^*)\|}V^*V^{*\top}(x-c^*)$ is the population version

Spherical projection



- $\widehat{\mathrm{Proj}}_n(x) := \hat{c} + \frac{\hat{r}}{\|\widehat{V}\widehat{V}^\top(x-\hat{c})\|}\widehat{V}\widehat{V}^\top(x-\hat{c})$ is the spherical projection to $S_{\widehat{V}}(\hat{c},\hat{r})$, where n is the sample size
- $\mathrm{Proj}^*(x) := c^* + \frac{r^*}{\|V^*V^{*\top}(x-c^*)\|}V^*V^{*\top}(x-c^*)$ is the population version
- $\widehat{\mathrm{Proj}}_n$ converges to Proj^* in probability under some mild conditions

Convergence of empirical SPCA

(A) *Distributional Assumption:* $X = V\Lambda^{1/2}Z$ where $Z = ((z_{i,j}))$ is a $n \times p$ matrix whose elements $z_{i,j}$'s are i.i.d. non-degenerate random variables with $E(z_{i,j}) = 0$, $E(z_{i,j}^2) = 1$ and $E(z_{i,j}^6) < \infty$.

Convergence of empirical SPCA

(A) *Distributional Assumption:* $X = V\Lambda^{1/2}Z$ where $Z = ((z_{i,j}))$ is a $n \times p$ matrix whose elements $z_{i,j}$'s are i.i.d. non-degenerate random variables with $E(z_{i,j}) = 0$, $E(z_{i,j}^2) = 1$ and $E(z_{i,j}^6) < \infty$.

(B) *Spike Population Model:* $\Lambda = \text{diag}\{\lambda_1, \cdots, \lambda_p\}$, then $\exists m > d$ s.t. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m > \lambda_{m+1} = \ldots = \lambda_p = 1$, .

Convergence of empirical SPCA

(A) *Distributional Assumption:* $X = V\Lambda^{1/2}Z$ where $Z = ((z_{i,j}))$ is a $n \times p$ matrix whose elements $z_{i,j}$'s are i.i.d. non-degenerate random variables with $E(z_{i,j}) = 0$, $E(z_{i,j}^2) = 1$ and $E(z_{i,j}^6) < \infty$.

(B) *Spike Population Model:* $\Lambda = \text{diag}\{\lambda_1, \cdots, \lambda_p\}$, then $\exists m > d$ s.t. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m > \lambda_{m+1} = \ldots = \lambda_p = 1$, .

Theorem

Under the assumptions A and B, for any x, we have

$$\widehat{\text{Proj}}_n(x) \xrightarrow{p} \text{Proj}^*(x).$$

Error bound

Theorem

There exists $\theta > 0$ *that depends only on* (M, ρ) *such that*

$$\mathbb{E}_{\rho_U} \|x - \mathrm{Proj}^*(x)\|^2 \leq \theta \alpha^4,$$

where $\alpha = \mathrm{diam}(U) = \sup_{x,y \in U} d(x, y)$ *is the diameter of* U.

Error bound

Theorem

There exists $\theta > 0$ *that depends only on* (M, ρ) *such that*

$$\mathbb{E}_{\rho_U} \|x - \mathrm{Proj}^*(x)\|^2 \leq \theta \alpha^4,$$

where $\alpha = \mathrm{diam}(U) = \sup_{x,y \in U} d(x, y)$ *is the diameter of* U.

Corollary

Under assumptions A, B, there exists $\theta \in \mathbb{R}$ *that depends only on* (M, ρ) *such that for any* x, *for any* $\epsilon > 0$,

$$\lim_{n \to \infty} \mathbb{P}(\|x - \widehat{\mathrm{Proj}}_n(x)\|^2 > \theta \alpha^4 + \epsilon) = 0.$$

Error bound

Theorem

There exists $\theta > 0$ *that depends only on* (M, ρ) *such that*

$$\mathbb{E}_{\rho_U} \|x - \mathrm{Proj}^*(x)\|^2 \leq \theta \alpha^4,$$

where $\alpha = \mathrm{diam}(U) = \sup_{x,y \in U} d(x, y)$ *is the diameter of* U.

Corollary

Under assumptions A, B, there exists $\theta \in \mathbb{R}$ *that depends only on* (M, ρ) *such that for any* x, *for any* $\epsilon > 0$,

$$\lim_{n \to \infty} \mathbb{P}(\|x - \widehat{\mathrm{Proj}}_n(x)\|^2 > \theta \alpha^4 + \epsilon) = 0.$$

- In some multi-scale methods, $\alpha = 2^{-j}$ where j is the partition level.

Analyzing data using spherelets

- The main theorem suggests that we should see big gains in practical performance

Analyzing data using spherelets

- The main theorem suggests that we should see big gains in practical performance
- Spherelets provide a general dictionary for manifolds and subspaces–Local SPCA vs Local PCA

Analyzing data using spherelets

- The main theorem suggests that we should see big gains in practical performance
- Spherelets provide a general dictionary for manifolds and subspaces–Local SPCA vs Local PCA
- For any (locally) linear algorithm, we can replace PCA by spherical PCA and get the spherical version–denoising & visualization

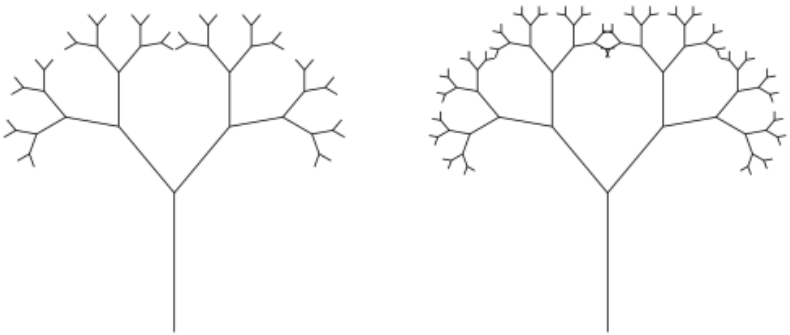
Analyzing data using spherelets

- The main theorem suggests that we should see big gains in practical performance
- Spherelets provide a general dictionary for manifolds and subspaces–Local SPCA vs Local PCA
- For any (locally) linear algorithm, we can replace PCA by spherical PCA and get the spherical version–denoising & visualization
- Given new (test) data, we don't need to retrain the spherelets–allow us to use CV to choose tuning parameters

Analyzing data using spherelets

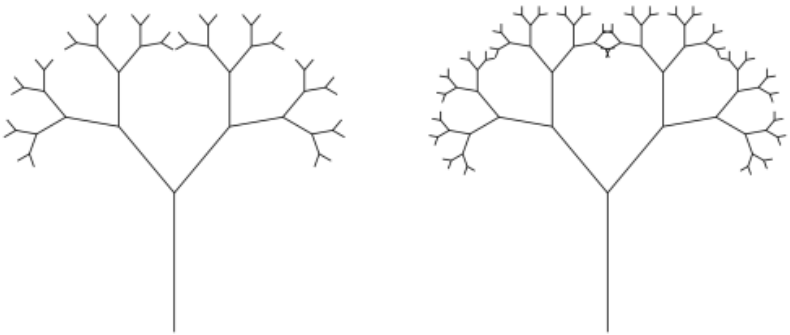
- The main theorem suggests that we should see big gains in practical performance
- Spherelets provide a general dictionary for manifolds and subspaces–Local SPCA vs Local PCA
- For any (locally) linear algorithm, we can replace PCA by spherical PCA and get the spherical version–denoising & visualization
- Given new (test) data, we don't need to retrain the spherelets–allow us to use CV to choose tuning parameters
- We also develop a mixtures of spherelets model for probabilistic inference (*Nonparametric Bayes*)

Local (S)PCA



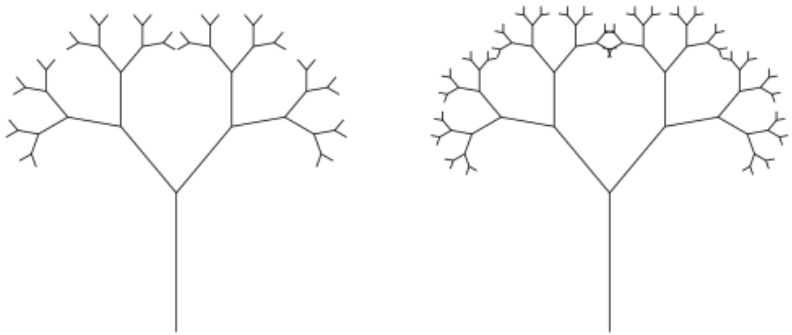
- Construct a partition $\{C_k\}_{k=1}^K$ where $\bigcup_{k=1}^K C_k = \mathbb{R}^p$

Local (S)PCA



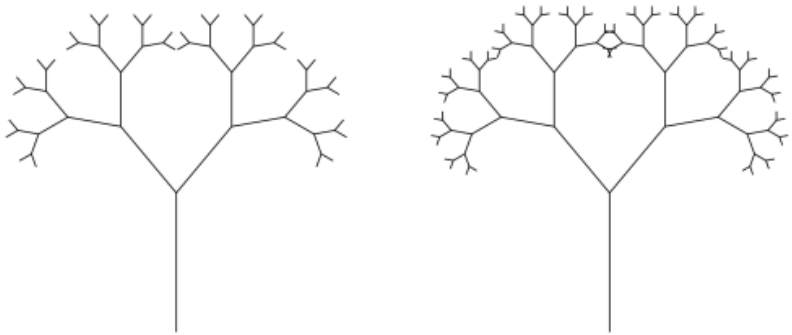
- Construct a partition $\{C_k\}_{k=1}^K$ where $\bigcup_{k=1}^K C_k = \mathbb{R}^p$
- Perform local (S)PCA on each C_k

Local (S)PCA



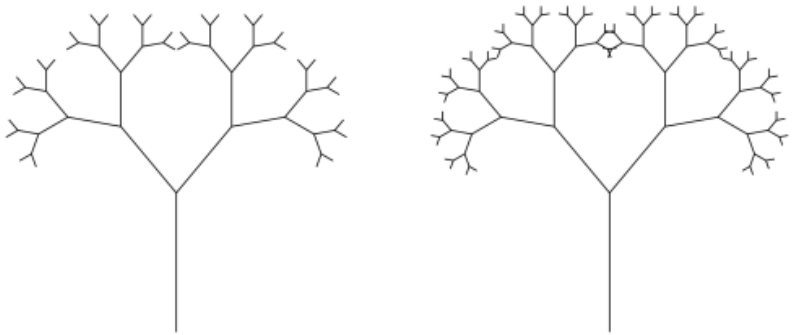
- Construct a partition $\{C_k\}_{k=1}^K$ where $\bigcup_{k=1}^K C_k = \mathbb{R}^p$
- Perform local (S)PCA on each C_k
- $\mathcal{M}$ could be approximated by its projection onto the family of linear subspaces(spherelets) obtained by local (S)PCA

Local (S)PCA



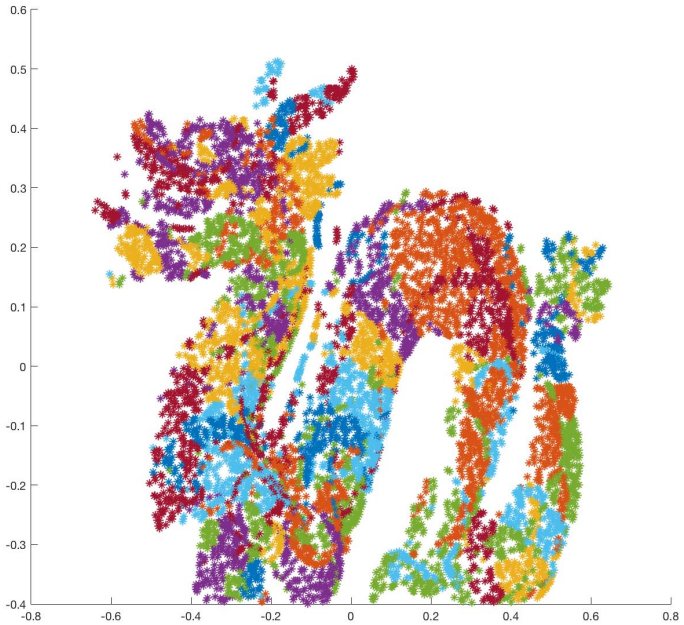
- Construct a partition $\{C_k\}_{k=1}^K$ where $\bigcup_{k=1}^K C_k = \mathbb{R}^p$
- Perform local (S)PCA on each C_k
- $\mathcal{M}$ could be approximated by its projection onto the family of linear subspaces(spherelets) obtained by local (S)PCA
- Many partitioning algorithms: cover tree, METIS, kNN, etc

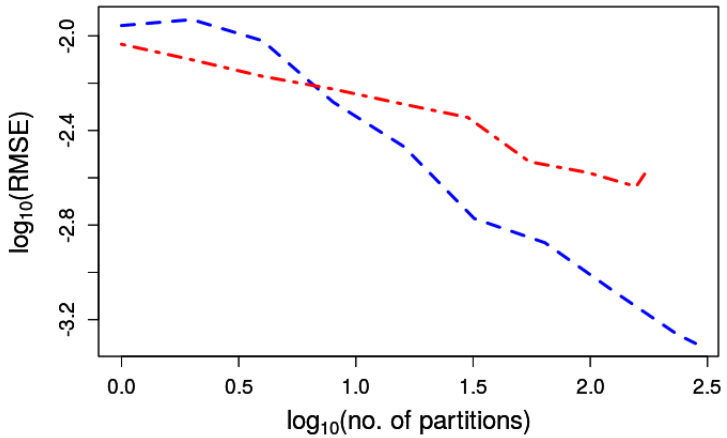
Local (S)PCA



- Construct a partition $\{C_k\}_{k=1}^K$ where $\bigcup_{k=1}^K C_k = \mathbb{R}^p$
- Perform local (S)PCA on each C_k
- $\mathcal{M}$ could be approximated by its projection onto the family of linear subspaces(spherelets) obtained by local (S)PCA
- Many partitioning algorithms: cover tree, METIS, kNN, etc

Dragon


0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8


log10(RMSE)
-2.0
-2.4
-2.8
-3.2
0.0
0.5
1.0
1.5
2.0
2.5
log10(no. of partitions)

Some real data apps (*'datasets' package in R*) [$d = 1$]

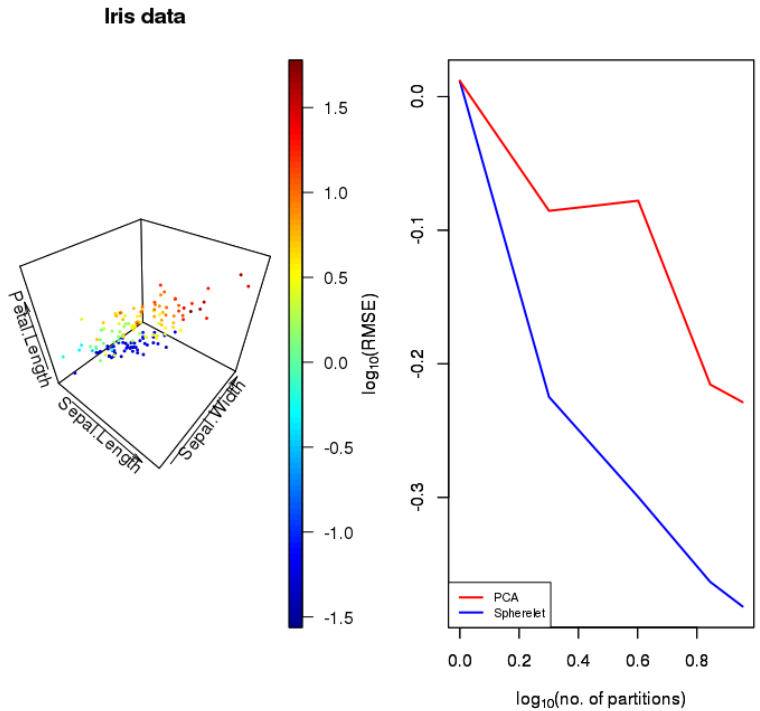
a. **Iris data**: measurements of sepal length & width + petal length & width, for 50 flowers from each of 3 species of iris.

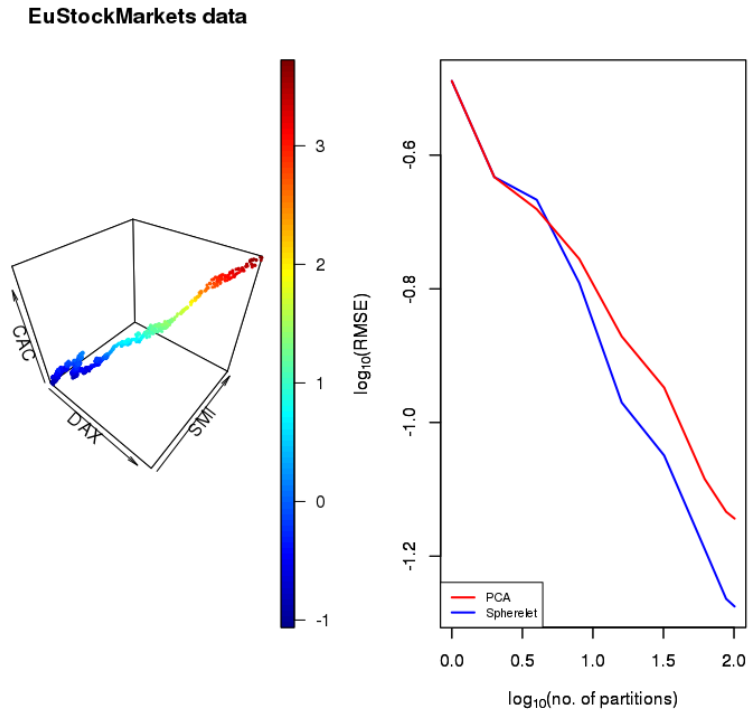
Some real data apps (*'datasets' package in R*) [$d = 1$]

a. **Iris data**: measurements of sepal length & width + petal length & width, for 50 flowers from each of 3 species of iris.
b. **EuStockMarkets data**: daily closing prices of major European stock indices: Germany DAX (Ibis), Switzerland SMI, France CAC & UK FTSE.

Some real data apps (*'datasets' package in R*) [$d = 1$]

a. **Iris data**: measurements of sepal length & width + petal length & width, for 50 flowers from each of 3 species of iris.
b. **EuStockMarkets data**: daily closing prices of major European stock indices: Germany DAX (Ibis), Switzerland SMI, France CAC & UK FTSE.





Some (more) real data apps [$d = 2$]

c. **Seals data**: vector field of seal movement from Brillinger et al., 2004 (*'ggplot2' R package*).

Some (more) real data apps [$d = 2$]

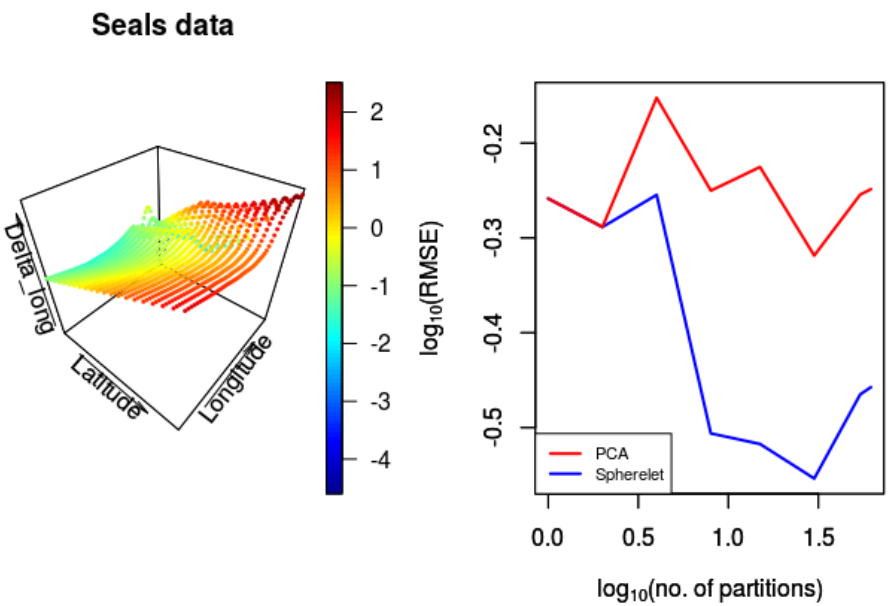
c. **Seals data**: vector field of seal movement from Brillinger et al., 2004 (*'ggplot2' R package*).

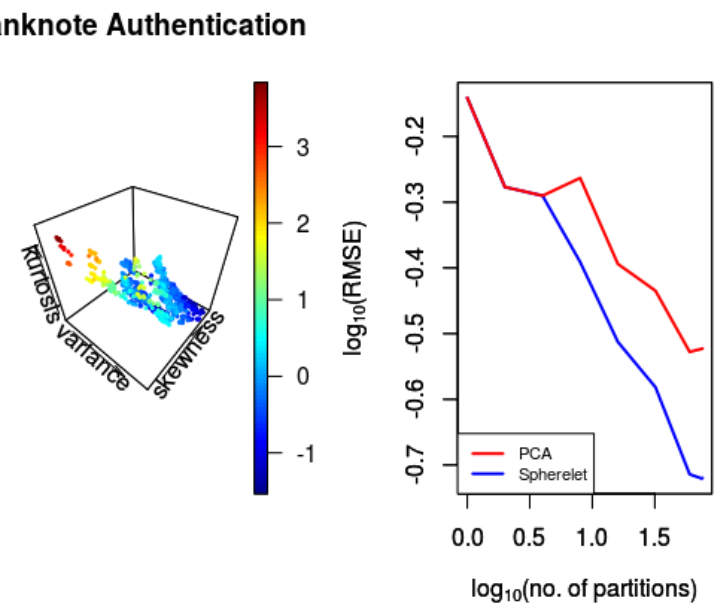
d. **Banknote authentication data**: images from genuine & forged banknote-like specimens (*UCL ML repository*)

Some (more) real data apps [$d = 2$]

c. **Seals data**: vector field of seal movement from Brillinger et al., 2004 (*'ggplot2' R package*).

d. **Banknote authentication data**: images from genuine & forged banknote-like specimens (*UCL ML repository*)



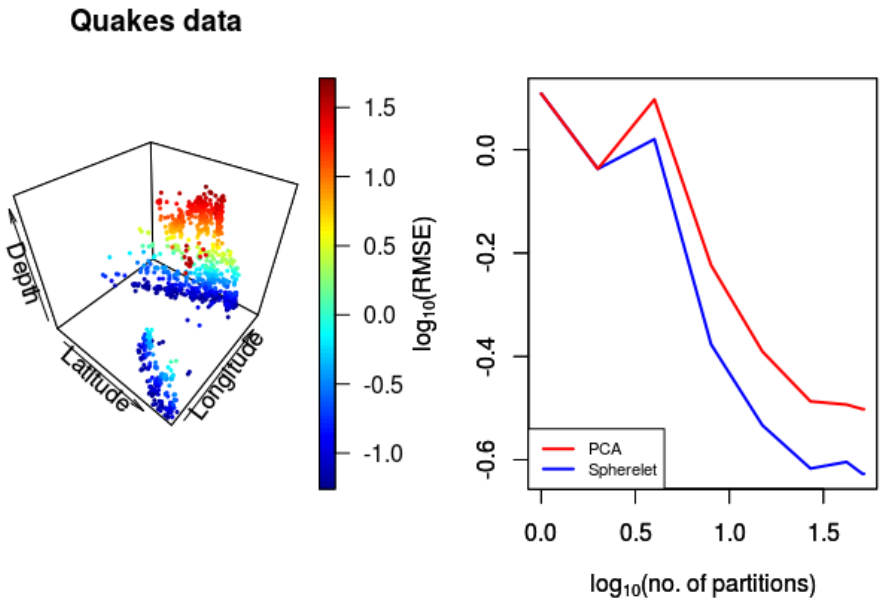


Yet another app (*'datasets' R package*) [$d = 1$]

e. **Quakes data**: locations of 1000 seismic events of $MB > 4.0$ occurring in a cube near Fiji since 1964.

Yet another app (*'datasets' R package*) [$d = 1$]

e. **Quakes data**: locations of 1000 seismic events of $MB > 4.0$ occurring in a cube near Fiji since 1964.



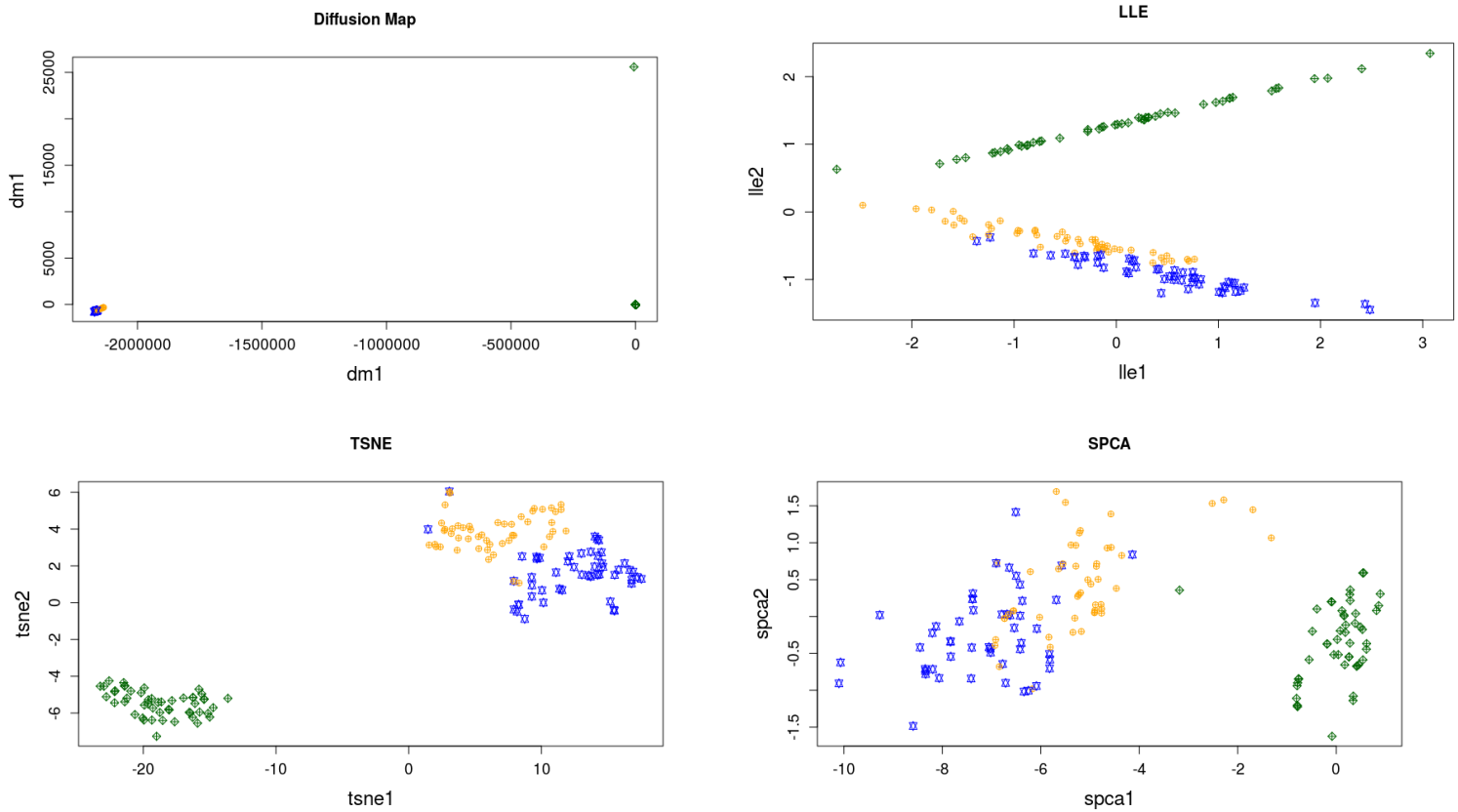
All datasets are standardized. In each case, we randomly select 1/2 samples as training & remaining as test.

Visualization

- **Iris data, d=1**: measurements of sepal length & width + petal length & width, for 50 flowers from each of 3 species of iris.

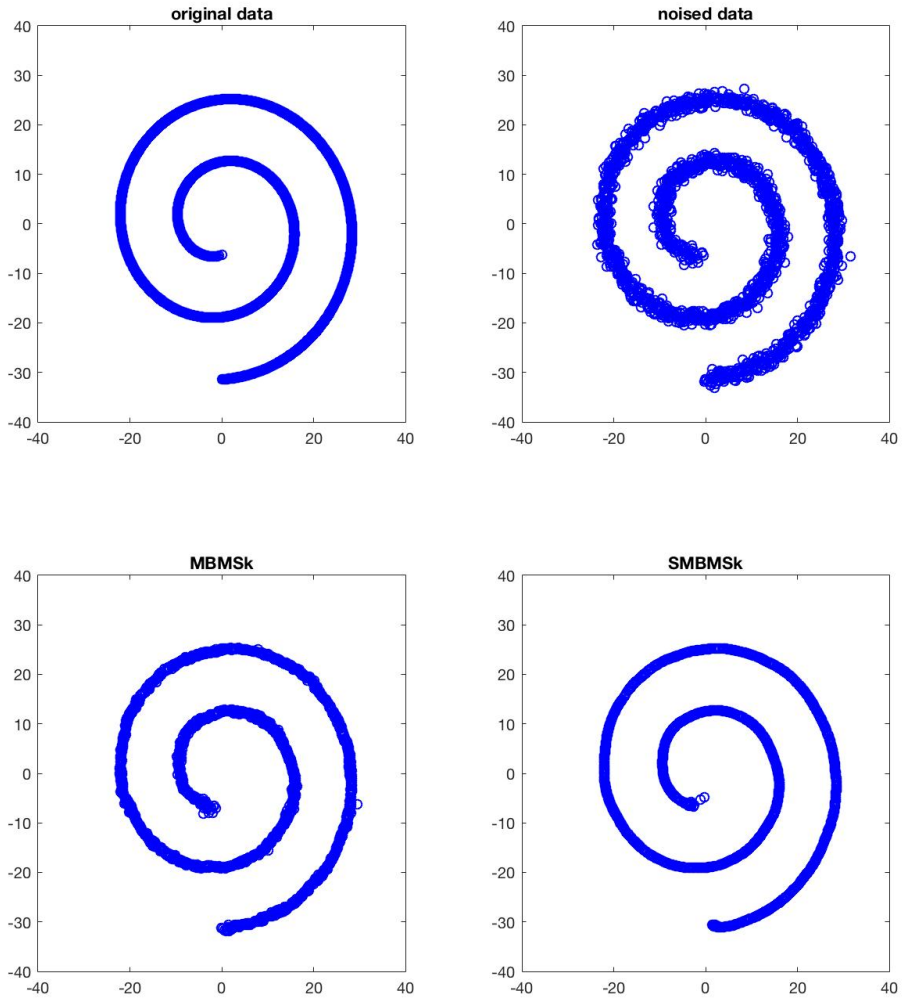
Visualization

- **Iris data, d=1**: measurements of sepal length & width + petal length & width, for 50 flowers from each of 3 species of iris.



Denoising

Manifold Blurring Mean Shift (MBMS) vs SMBMS



Nonparametric subspace & density estimation

- We can also take a likelihood-based approach

Nonparametric subspace & density estimation

- We can also take a likelihood-based approach
- *Mixture of spherelets* model

Nonparametric subspace & density estimation

- We can also take a likelihood-based approach
- *Mixture of spherelets* model
- ith data point is generated from the hth sphere with probability π_h

Nonparametric subspace & density estimation

- We can also take a likelihood-based approach
- *Mixture of spherelets* model
- ith data point is generated from the hth sphere with probability π_h
- Data in component h drawn from location-scale mixture of von Mises-Fisher distributions on sphere h

Nonparametric subspace & density estimation

- We can also take a likelihood-based approach
- *Mixture of spherelets* model
- ith data point is generated from the hth sphere with probability π_h
- Data in component h drawn from location-scale mixture of von Mises-Fisher distributions on sphere h
- Gaussian noise added to allow data to not fall exactly on a particular sphere

Mixture of spherelets : Model

Let $\{x_i\}_{i=1}^n$ be the observations with

$$x_i = y_i + \epsilon_i,$$

where y_i is exactly on some sphere & $\epsilon_i \sim N(0, \sigma^2 I_p)$.

Mixture of spherelets : Model

Let $\{x_i\}_{i=1}^n$ be the observations with

$$x_i = y_i + \epsilon_i,$$

where y_i is exactly on some sphere & $\epsilon_i \sim N(0, \sigma^2 I_p)$.

- $$f(y_i|\Pi, \Theta) = \sum_{k=1}^{K} \pi_k f(y_i|\Theta_k), \text{ with } \Pi = (\pi_1, \cdots, \pi_K),$$

  $f(y|\Theta_k)$ = density on kth sphere, $\Theta_k = (\Lambda_k, V_k, \mathbf{c}_k, r_k, M_k, T_k)$.

Mixture of spherelets : Model

Let $\{x_i\}_{i=1}^n$ be the observations with

$$x_i = y_i + \epsilon_i,$$

where y_i is exactly on some sphere & $\epsilon_i \sim N(0, \sigma^2 I_p)$.

- $$f(y_i|\Pi, \Theta) = \sum_{k=1}^{K} \pi_k f(y_i|\Theta_k), \text{ with } \Pi = (\pi_1, \cdots, \pi_K),$$

  $f(y|\Theta_k)$ = density on kth sphere, $\Theta_k = (\Lambda_k, V_k, \mathbf{c}_k, r_k, M_k, T_k)$.
- $$f\left(\frac{V_k V_k'(y_i - \mathbf{c}_k)}{r_k} \middle| M_k, T_k, \Lambda_k\right) = \sum_{l_k=1}^{L} \lambda_{l_k} f_{vMF}\left(\frac{y_i - \mathbf{c}_k}{r_k} \middle| \boldsymbol{\mu}_{l_k}, \tau_{l_k}\right),$$

  where $f_{vMF}(\cdot|\boldsymbol{\mu}, \tau)$ = Von-Mises Fisher density, and $\Lambda_k = (\lambda_{l_1}, \lambda_{l_2}, \cdots, \lambda_{l_k})$.

Mixture of spherelets : Priors

The priors of different parameters are as follows:

a. $\Pi = (\pi_1, \pi_2, \cdots, \pi_K) \sim \text{Dirichlet}(1/K, \ldots, 1/K)$.

Mixture of spherelets : Priors

The priors of different parameters are as follows:

a. $\Pi = (\pi_1, \pi_2, \cdots, \pi_K) \sim \text{Dirichlet}(1/K, \ldots, 1/K)$.

b. $\Lambda_k = (\lambda_{l_1}, \cdots, \lambda_{l_k}) \sim \text{Dirichlet}(1/L, \ldots, 1/L)$.

Mixture of spherelets : Priors

The priors of different parameters are as follows:

a. $\Pi = (\pi_1, \pi_2, \cdots, \pi_K) \sim \text{Dirichlet}(1/K, \ldots, 1/K)$.

b. $\Lambda_k = (\lambda_{l_1}, \cdots, \lambda_{l_k}) \sim \text{Dirichlet}(1/L, \ldots, 1/L)$.

c. $\mathbf{c}_k \sim N\left(\hat{\mathbf{c}}_k, \sigma_1^2 I_p\right), \quad r_k \sim \textit{Inverse–Gamma}(a_r, b_r)$, where a_r, b_r and σ_1 are hyper-parameters, $\hat{\mathbf{c}}_k$ is the empirical estimate of $\mathbf{c}_k$.

Mixture of spherelets : Priors

The priors of different parameters are as follows:

a. $\Pi = (\pi_1, \pi_2, \cdots, \pi_K) \sim \text{Dirichlet}(1/K, \ldots, 1/K)$.

b. $\Lambda_k = (\lambda_{l_1}, \cdots, \lambda_{l_k}) \sim \text{Dirichlet}(1/L, \ldots, 1/L)$.

c. $\mathbf{c}_k \sim N\left(\hat{\mathbf{c}}_k, \sigma_1^2 I_p\right), \quad r_k \sim \textit{Inverse–Gamma}(a_r, b_r)$, where a_r, b_r and σ_1 are hyper-parameters, $\hat{\mathbf{c}}_k$ is the empirical estimate of $\mathbf{c}_k$.

d. $\boldsymbol{\mu}_{l_k} \sim \text{vMF}((1/\sqrt{d}, \ldots, 1/\sqrt{d}), \kappa)$, and $\tau_{l_k} \sim \textit{Gamma}(a_\tau, b_\tau)$.

Mixture of spherelets : Priors

The priors of different parameters are as follows:

a. $\Pi = (\pi_1, \pi_2, \cdots, \pi_K) \sim \text{Dirichlet}(1/K, \ldots, 1/K)$.

b. $\Lambda_k = (\lambda_{l_1}, \cdots, \lambda_{l_k}) \sim \text{Dirichlet}(1/L, \ldots, 1/L)$.

c. $\mathbf{c}_k \sim N\left(\hat{\mathbf{c}}_k, \sigma_1^2 I_p\right), \quad r_k \sim \textit{Inverse–Gamma}(a_r, b_r)$, where a_r, b_r and σ_1 are hyper-parameters, $\hat{\mathbf{c}}_k$ is the empirical estimate of $\mathbf{c}_k$.

d. $\boldsymbol{\mu}_{l_k} \sim \text{vMF}((1/\sqrt{d}, \ldots, 1/\sqrt{d}), \kappa)$, and $\tau_{l_k} \sim \textit{Gamma}(a_\tau, b_\tau)$.

e. $\sigma^2 \sim \text{Inverse-Gamma}(a_\sigma, b_\sigma)$.

Mixture of spherelets : Priors

The priors of different parameters are as follows:

a. $\Pi = (\pi_1, \pi_2, \cdots, \pi_K) \sim \text{Dirichlet}(1/K, \ldots, 1/K)$.

b. $\Lambda_k = (\lambda_{l_1}, \cdots, \lambda_{l_k}) \sim \text{Dirichlet}(1/L, \ldots, 1/L)$.

c. $\mathbf{c}_k \sim N\left(\hat{\mathbf{c}}_k, \sigma_1^2 I_p\right), \quad r_k \sim \textit{Inverse–Gamma}(a_r, b_r)$, where a_r, b_r and σ_1 are hyper-parameters, $\hat{\mathbf{c}}_k$ is the empirical estimate of $\mathbf{c}_k$.

d. $\boldsymbol{\mu}_{l_k} \sim \text{vMF}((1/\sqrt{d}, \ldots, 1/\sqrt{d}), \kappa)$, and $\tau_{l_k} \sim \textit{Gamma}(a_\tau, b_\tau)$.

e. $\sigma^2 \sim \text{Inverse-Gamma}(a_\sigma, b_\sigma)$.

f. The matrix V_k is the empirical Bayes estimate.

Computation - Mixture of spherelets model

- For a finite mixture model, an EM algorithm or MCMC algorithm can be easily implement for computation

Computation - Mixture of spherelets model

- For a finite mixture model, an EM algorithm or MCMC algorithm can be easily implement for computation
- We initially take a fully Bayesian approach, using default priors & running MCMC

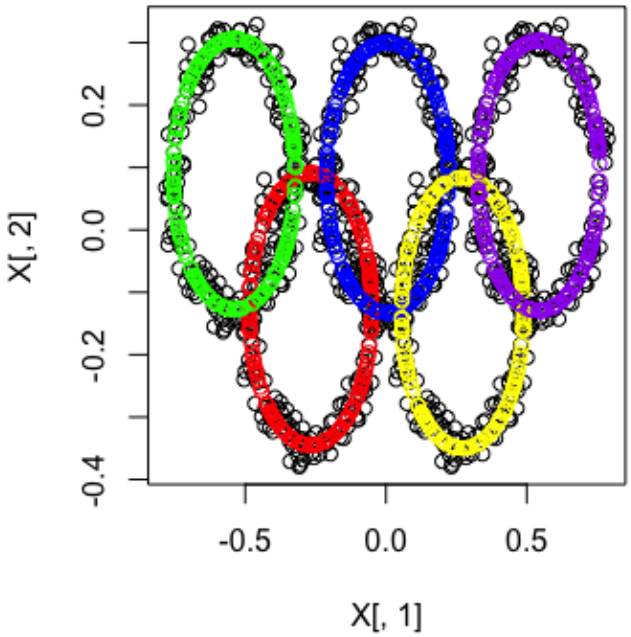
Computation - Mixture of spherelets model

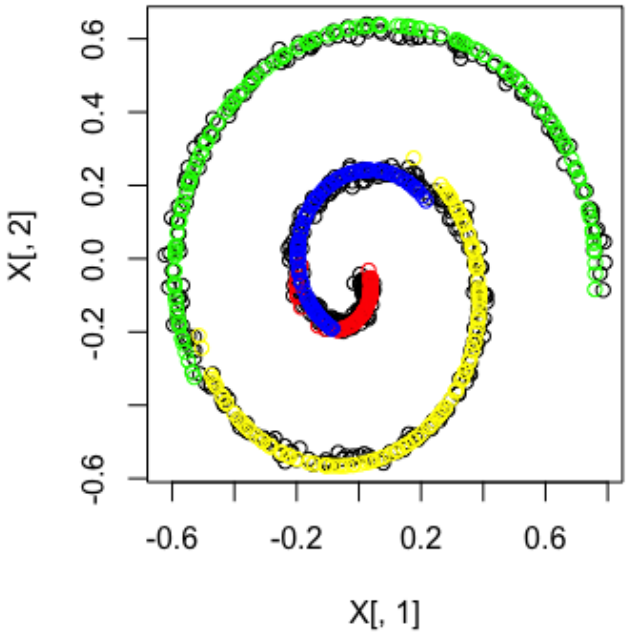
- For a finite mixture model, an EM algorithm or MCMC algorithm can be easily implement for computation
- We initially take a fully Bayesian approach, using default priors & running MCMC
- A simple data augmentation Gibbs sampler can be defined - starting the chain at the output of our initial algorithm

Computation - Mixture of spherelets model

- For a finite mixture model, an EM algorithm or MCMC algorithm can be easily implement for computation
- We initially take a fully Bayesian approach, using default priors & running MCMC
- A simple data augmentation Gibbs sampler can be defined - starting the chain at the output of our initial algorithm
- Over-fitted mixtures (Rousseau & Mengerson 2011) allow uncertainty in # of mixture components/clusters

Olympic Rings and Spiral-Bayesian version





Discussion

- Based on our theory & initial results, spherelets provide a promising alternative to linear approach (PCA)

Discussion

- Based on our theory & initial results, spherelets provide a promising alternative to linear approach (PCA)
- There are a lot of potential applications including manifold learning, denoising, visualization, manifold regression, clustering, etc

Discussion

- Based on our theory & initial results, spherelets provide a promising alternative to linear approach (PCA)
- There are a lot of potential applications including manifold learning, denoising, visualization, manifold regression, clustering, etc
- In the Bayesian case, we would like to estimate both $\mathcal{M}$ & $f(y)$ - obtaining minimax optimal posterior concentration rates

Discussion

- Based on our theory & initial results, spherelets provide a promising alternative to linear approach (PCA)
- There are a lot of potential applications including manifold learning, denoising, visualization, manifold regression, clustering, etc
- In the Bayesian case, we would like to estimate both $\mathcal{M}$ & $f(y)$ - obtaining minimax optimal posterior concentration rates
- Using the model-based approach straightforward to extend the approach to broad & complex data structures

Discussion

- Based on our theory & initial results, spherelets provide a promising alternative to linear approach (PCA)
- There are a lot of potential applications including manifold learning, denoising, visualization, manifold regression, clustering, etc
- In the Bayesian case, we would like to estimate both $\mathcal{M}$ & $f(y)$ - obtaining minimax optimal posterior concentration rates
- Using the model-based approach straightforward to extend the approach to broad & complex data structures

Acknowledgments & References

- D. Li and D. Dunson, Efficient Manifold and Subspace Approximations with Spherelets, https://arxiv.org/abs/1706.08263.
- W. Liao and M. Maggioni, Adaptive Geometric Multiscale Approximations for Intrinsically Low-dimensional Data, arXiv:1611.011, 2016.
- J. Rousseau and K. Mengersen, Asymptotic behaviour of the posterior distribution in overfitted mixture models, JRSS-B, 2011.
- M. Maggioni, S Minsker and N. Strawn, Multiscale Dictionary Learning: Non-Asymptotic Bounds and Robustness, Journal of Machine Learning Research, 2016.