Spherelets

Stat 185 Term Paper

Caleb Ren

December 14, 2019

Contents

1 Introduction

Whereas principal component analysis (PCA) is an eigenvalue/eigenvector problem from an inherently *linear* dimension reduction problem,

2 Method

2.1 Spherical PCA

Given a set of data $\vec{x}_1, \ldots, \vec{x}_N \in \mathbb{R}^D$, we find the best approximating sphere $S_V(c, r)$, where c is the center, r is the radius, and $V \in \mathbb{R}^{(d+1)\times(d+1)}$ is the $(d+1)$th dimensional affine subspace the sphere lives on. For any point in the dataset $\vec{x}_i$, the closest point $\vec{y}_i$ lying on the sphere $S_V(c, r)$ is the point that minimizes Euclidean distance $\|x, y\|^2$ between x and y. The optimal subspace V is given by $\hat{V} = (\vec{v}_1, \ldots, \vec{v}_{d+1})$, where $\vec{v}_i, i \in \{1, \ldots, d+1\}$ is the ith eigenvector ranked in descending order of $(\mathbf{X} - 1_N\bar{\mathbf{X}})^T(\mathbf{X} - 1_N\bar{\mathbf{X}})$.

If $\vec{z}_i = \bar{\mathbf{X}} + \hat{V}\hat{V}^T(\vec{x}_i - \bar{\mathbf{X}})$ are a change of basis to affine subspace V, then it can be shown that the minimizing pair $(\vec{\eta}^\star, \vec{\xi}^\star)$ of loss function $g(\vec{\eta}, \vec{\xi}) = \sum_{k=1}^N (\vec{z}_i^T\vec{z}_i + \vec{\eta}^T\vec{x}_i + \vec{\xi})^2$ is:

$$\vec{\eta} = -H^{-1}\omega$$

$$\vec{\xi} = -\frac{1}{N}\sum_{k=1}^N (\vec{z}_i^T\vec{z}_i + \vec{\eta}^T\vec{z}_i)$$

where H and ω are defined as:

$$H = \sum_{k=1}^N (\vec{z}_i - \bar{z})(\vec{z}_i - \bar{z})^T$$

$$\vec{\omega} = \sum_{k=1}^N \left(\|\vec{z}_i^T\vec{z}_i\| - \frac{1}{N}\sum_{j=1}^N \|\vec{z}_j^T\vec{z}_j\| \right)(\vec{z}_i - \bar{z})$$

The optimal parametrization $(\hat{V}, \hat{c}, \hat{r})$ of the projection of $\mathbf{X} \in \mathbb{R}^{N\times D}$ onto the sphere $S_V(c, r)$ is:

$$\hat{V} = (\vec{v}_1, \ldots, \vec{v}_{d+1})$$

$$\hat{c} = -\frac{\vec{\eta}^\star}{2}$$

$$\hat{r} = \frac{1}{N}\sum_{k=1}^N \|\vec{z}_i - \hat{c}\|$$

The projection map $\hat{\Psi}$ of data matrix $\mathbf{X}$ onto sphere $S_{\hat{V}}(\hat{c}, \hat{r})$ is the projection map onto affine subspace $\hat{c} + \hat{V}$, given by:

$$\hat{\Psi}(\vec{x}_i) = \hat{c} + \frac{\hat{r}}{\|\hat{V}\hat{V}^T(\vec{x}_i - \hat{c})\|}\hat{V}\hat{V}^T(\vec{x}_i - \hat{c})$$

2.2 Local SPCA

We have now defined spherical PCA (SPCA) to project the data $\mathbf{X}$ down to single sphere S_V. However, this single sphere will typically not be a sufficient approximation for the inherent manifold M. Instead, we partition the space $\mathbb{R}^D$ into k disjoint subsets $C_1, \ldots, C_k$. For the kth disjoint subset, we can define a data matrix $\mathbf{X}_k = \{X_i : X_i \in C_k\}$ that is a partition of the original data that lies within C_k. After applying SPCA to $\mathbf{X}_k$, we obtain spherical volume, center, and radius $(\hat{V}_k, \hat{c}_k, \hat{r}_k)$ alongside projection map Φ_k as a map from $x \in C_k$ to $y \in S_{\hat{V}_k}(\hat{c}_k, \hat{r}_k)$. A spherelets estimation $\hat{M}$ of the manifold M can be obtained by setting $\hat{M} = \bigcup_{k=1}^{K} \hat{M}_k$, where $\hat{M}_k$ is the local SPCA in the kth region and $\hat{M}_k = S_{\hat{V}_k}(\hat{c}_k, \hat{r}_k) \cap C_k$

2.3 Assumptions

There are two main

2.4 Method

The algorithm is as follows:

Algorithm 1 Spherelets

Input: Data matrix $\mathbf{X}$; intrinsic dimension d; partition $\{C_k\}_{k=1}^{K}$

Output: Local estimated manifolds $\hat{M}_k$ and projection map $\hat{\Psi}_k, k \in \{1, \ldots, K\}$; global estimated manifold $\hat{M}$ of intrinsic manifold M and projection map $\hat{\Psi}$

1: **for** (k = 1 : K) **do**
2: Define $\mathbf{X}_{[k]} = \mathbf{X} \cap C_k$
3: Calculate $\hat{V}_k, \hat{c}_k, \hat{r}_k$
4: Calculate $\hat{\Psi}_k(x) = \hat{c}_k + \frac{\hat{r}_k}{\|\hat{V}_k \hat{V}_k^T (x - \hat{c}_k)\|}(x - \hat{c}_k)$
5: Calculate $\hat{M}_k = S_{\hat{V}_k}(\hat{c}_k, \hat{r}_k) \cap C_k$
6: **end for**
7: Calculate $\hat{\Psi}(x) = \sum_{k=1}^{K} \mathbf{1}_{\{x \in C_k\}} \hat{\Psi}_k(x)$, and $\hat{M} = \bigcup_{k=1}^{K} \hat{M}_k$.

3 Strengths and Weaknesses

3.1 Strengths

- Performs well in areas with high curvature that local PCA can't approximate
- Can perform OOS assessments and returns the underlying manifold

3.2 Weaknesses

- Struggles with areas of non-uniform curvature
- Struggles with non-uniform dimensions
- Must specify inherent dimension d
- Computationally expensive
- Dependent on choice of manifold subsetting

4 Examples

To generate numerical examples, I used the `SPCA` and `SS_calc` functions written by co-author Minerva Mukhopadhyay (mmukhopadhyay 2019). The `SPCA` function takes in a matrix of N observations $\vec{x}_i \in \mathbb{R}^D, i \in 1, \ldots, N$ and returns the error given by spherical and local PCA (`SS` and `SS_new`), as well as the projected values `Y_D`.

4.1 Euler Spiral

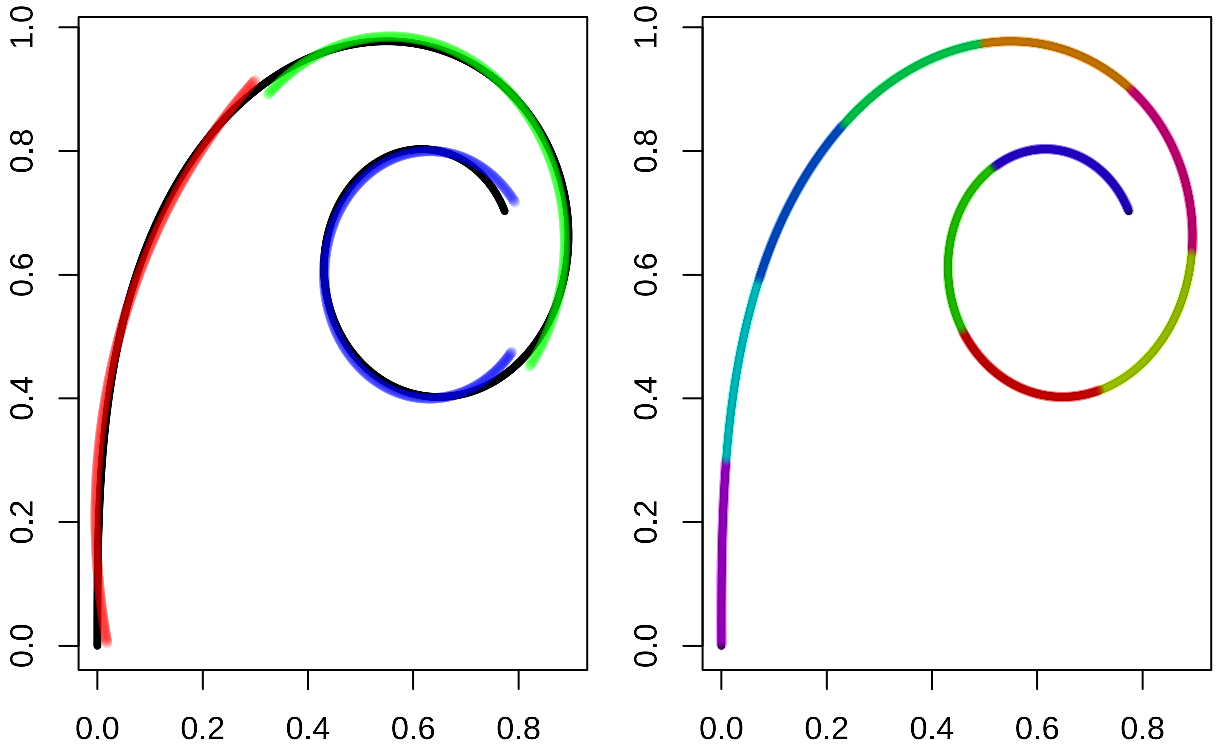

Figure 1: Spherical PCA performed on an Euler spiral with $k = 3, 8$.

4.2 Helix

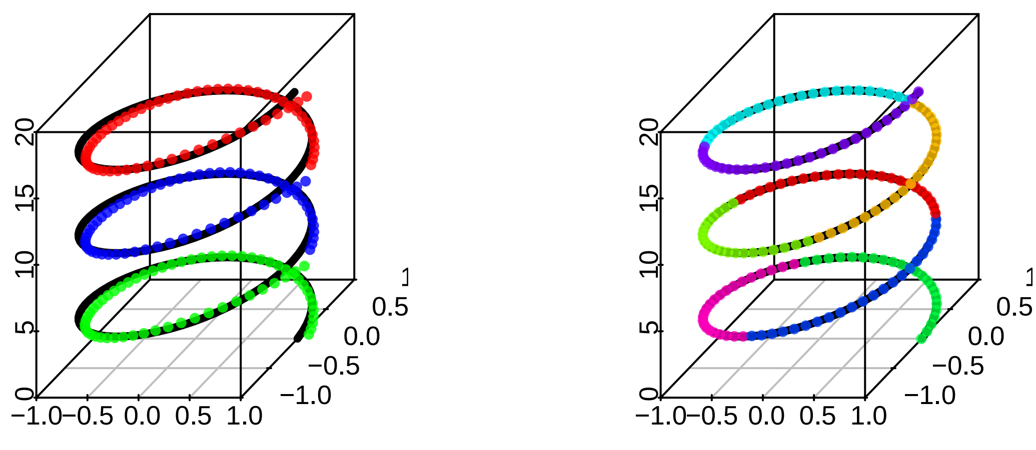

Figure 2: Spherical PCA performed on a helix with $k = 3, 8$.

4.3 Cylinder

```
## [1] 0.00269493
## [1] 7.353559e-05
```

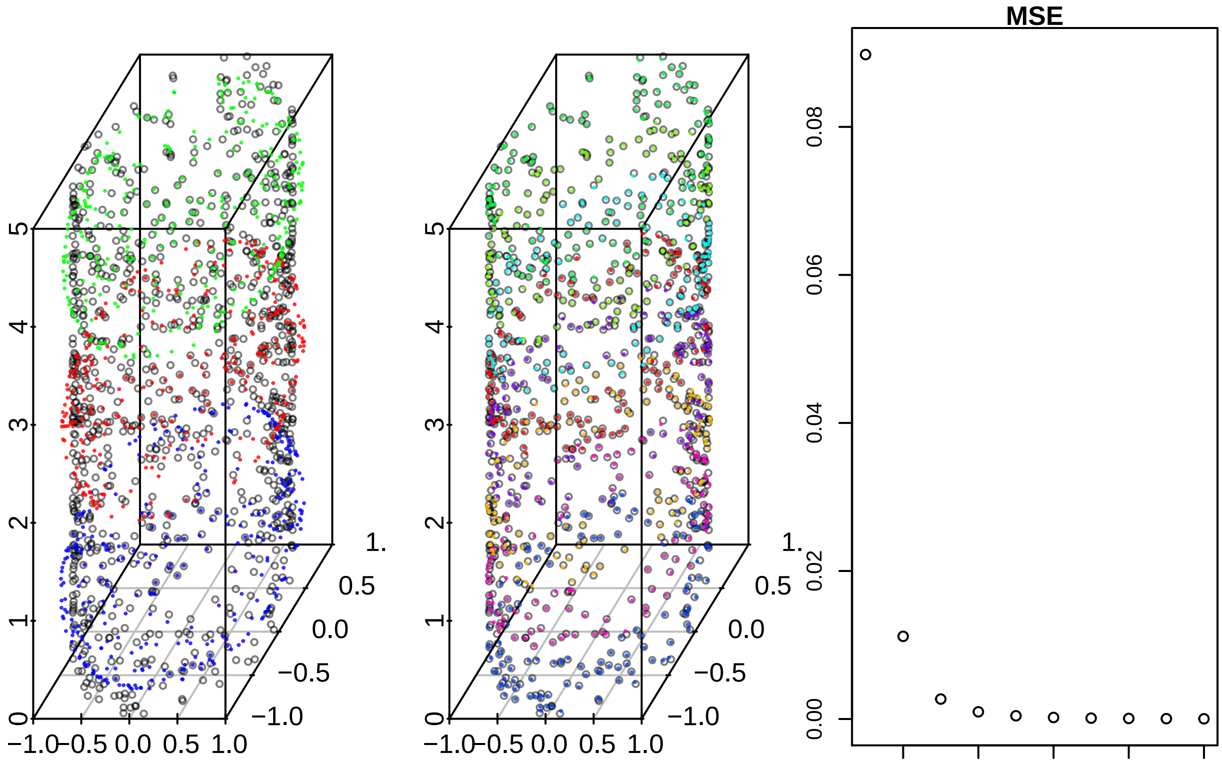

Figure 3: Spherical PCA performed on a cylinder with $k = 3, 8$.

We see that SPCA is not fully capable of handling a cylinder.

5 References

mmukhopadhyay. 2019. "Efficient Manifold Learning Using Spherelets." Github. April 9.