Spectral Methods for Dimensionality Reduction

A Literature Review

Jay Paek

UCSD Mathematics Directed Reading Program Project Presentation,
Spring 2024
Mentor: Qihao Ye

Goals

We have the following goals for this presentation:

- **Motivation:** Explain the curse of dimensionality in data science and classical methods in feature extraction and dimension-reduction techniques.
- **Main Algorithm:** Provide an intuitive understanding of the theory behind Laplacian eigenmaps and diffusion maps.
- **Numerical Experiments:** Present results from simulations done on datasets.
- **Theoretical Guarantees:** Introduce theoretical aspects of these techniques and some of the fundamental theorem in the papers.

Table of Contents

Table of Contents

Curse of Dimensionality

- Data points are in high dimensions but could lie in lower dimensional manifold.
- Behavior of these manifolds are not easily predictable in higher dimensions [Motivating Example 1].
- How to learn the manifold?

Motivating Examples

Example 1: Uniform sampling from an ℓ_∞-ball

Consider the unit-norm ball (defined by the ℓ_∞ norm) in $\mathbb{R}^d$. Let $[\mathbf{x}]_i$ denote the ith entry of $\mathbf{x}$.

$$S = \{\mathbf{x} \in \mathbb{R}^d : \max_{1 \leq i \leq d} |[\mathbf{x}]_i| < 1\}$$

Let us sample from this sample space under a uniform distribution. Each coordinate is independent.

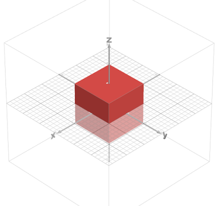

Figure: An ℓ_∞-ball in $\mathbb{R}^3$.

What is the probability of sampling a point such that $|[\mathbf{x}]_i| < 0.99, \forall 1 \leq i \leq d$?

Motivating Examples

Example 1: Uniform sampling from an ℓ_∞-ball

Let $X : \mathcal{F} \to \mathbb{R}^d$ be a random vector.

$$P(||X||_\infty < 0.99)$$
$$= \prod_{i=1}^{d} P(|[X]_i| < 0.99) = 0.99^d$$

Notice if $d \gg 0$, then $P(||X||_\infty < 0.99) \to 0$

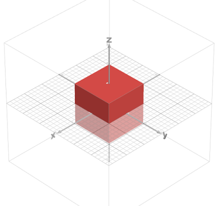

Figure: An ℓ_∞-ball in $\mathbb{R}^3$.

"High-dimensional orange is just the peel!" - Mikhail Belkin

Motivating Examples

Example 2: Principle Component Analysis

Consider a clustering task with two clusters with sample mean and covariance $\bar{x}_1, \bar{x}_2$ and $\bar{\Sigma}_1, \bar{\Sigma}_2$, respectively. Which direction should we project to perform most optimal classification?

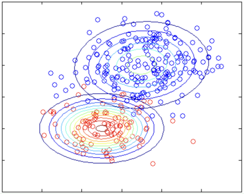

Figure: 2D Gaussian mixture.

"Some traits are easier to discriminate than others."
- Nuno Vasconcelos

Table of Contents

Algorithm Preparation

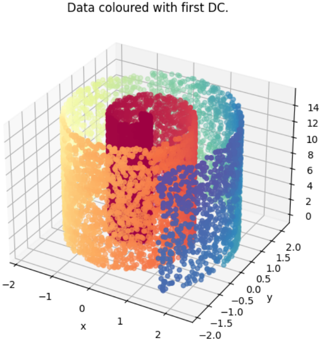

Figure: Swiss roll dataset $\subset \mathbb{R}^3$

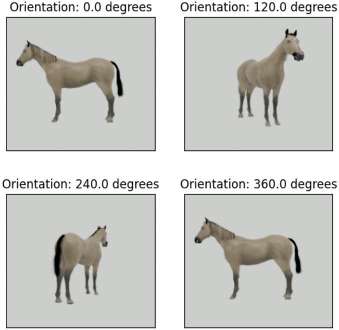

Figure: Horse dataset $\subset \mathbb{R}^{180\times200\times3}$ [2]

Graph Construction

Laplacian Eigenmap [4]	$[W]_{i,j} = \begin{cases} \exp\left\{-\frac{\lVert \mathbf{x}_i - \mathbf{x}_j \rVert^2}{\epsilon}\right\} & \mathbf{x}_i, \mathbf{x}_j \text{ connected} \\ 0 & \mathbf{x}_i, \mathbf{x}_j \text{ disconnected} \end{cases}$
Diffusion Map [1]	$W = P^t, \qquad [P]_{i,j} = \frac{K(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{\mathbf{z} \in X} K(\mathbf{x}_i, \mathbf{z})}$

Spectral Decomposition

Laplacian Eigenmap	Construct D s.t. $[D]_{i,i} = \sum_{j=1}^{n} W_{j,i}$ $L\psi = \lambda D\psi$ where $L = D - W$ $\psi_1, \ldots, \psi_m$ where $0 \neq \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_m \leq \ldots \leq \lambda_n$,
Diffusion Map	$\{\psi_l\}_{l \geq 1}$ with eigenvalues $1 = \lambda_0 > \lvert\lambda_1\rvert \geq \ldots \geq \lvert\lambda_n\rvert$. $W\psi_k = \lambda_k \psi_k$

then for every sample:

$$\mathbf{x}_i \mapsto [\psi_1(i), \ldots, \psi_m(i)]$$

Table of Contents

Experiment 1: Laplacian eigenmap for swiss roll dataset

Construct the following toy dataset with 10000 points.

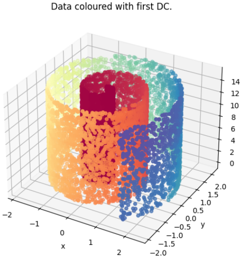

Figure: Swiss roll dataset

With the Gaussian similarity kernel that applies to the nearest 200 points, we can recover the following:

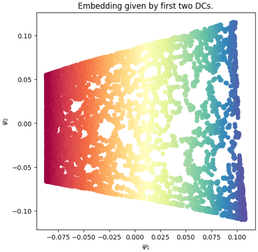

Figure: Projection to 2 diffusion coordinates

Experiment 1: Laplacian eigenmap for swiss roll dataset

Construct the following toy dataset with 6000 points.

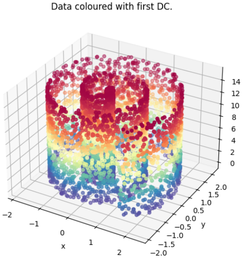

Figure: Swiss roll dataset

With the Gaussian similarity kernel that applies to the nearest 200 points, we can recover the following:

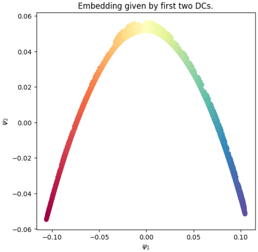

Figure: Projection to 2 diffusion coordinates

Experiment 2: Diffusion map for orientation learning

Consider the following dataset with 1000 image that are $(180 \times 200 \times 3)$

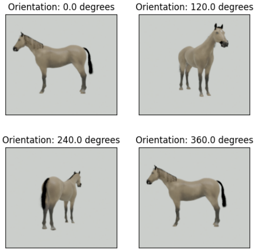

Figure: Horse orientation data

We construct the transition probability matrix with respect to a Gaussian kernel.

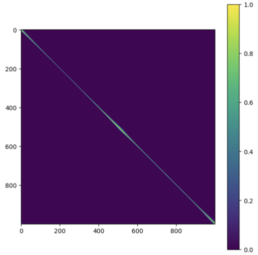

Figure: Transition matrix P for horse dataset

Experiment 2: Diffusion map for orientation learning

With the Gaussian similarity kernel, we can recover the embedding:

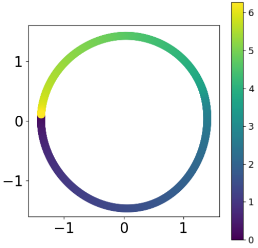

Figure: Projection to 2 diffusion coordinates. Color denotes the orientation in radians.

Graph the diffusion distance w.r.t. the 0° orientation sample

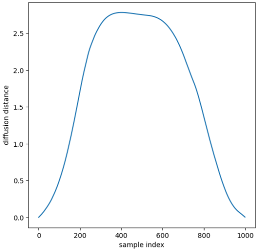

Figure: Graph of sample index vs. diffusion distance.

Additional Comments

- Feature extraction is orientation invariant w.r.t feature and distance invariant w.r.t. data points.
- Computing eigenvectors for an $N \times N$ matrix is not optimal.
- Large dataset needed for optimal learning.
- Possible applications:
  - Known/semi-known environment SLAM.
  - Facial recognition
  - Geometric data interpretation

Table of Contents

Adjacency graph for samples

Form an edge between $\mathbf{x}_i, \mathbf{x}_j$ if they are "close" to each other.

- Neighborhood relation: connect $\mathbf{x}_i, \mathbf{x}_j$ if $||\mathbf{x}_i - \mathbf{x}_j|| < \epsilon$, for a chosen ϵ and distance metric.
  - *Advantages:* Makes geometric sense, forms an equivalence relation between points.
  - *Disadvantages:* Selection of ϵ, can form too many edges or isolate points.
- k-nearest neighbors: connect $\mathbf{x}_i, \mathbf{x}_j$ if $\mathbf{x}_j$ is the within the kth closest sample.
  - *Advantages:* Easier computationally, will never have a disconnected graph.
  - *Disadvantages:* Less geometric intuition.

Attach weights to edges

For weights, the paper proposes two options:

- Gaussian kernel with parameter $t \in \mathbb{R}$. Let W be a matrix such that $[W]_{i,j} = w_{ij}$ (ith row, jth column) is the the weight of the edge connecting $\mathbf{x}_i, \mathbf{x}_j$. Then

$$w_{ij} = \begin{cases} \exp\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma}\} & \mathbf{x}_i, \mathbf{x}_j \text{ connected} \\ 0 & \mathbf{x}_i, \mathbf{x}_j \text{ disconnected} \end{cases}$$

  - *Intuition:* Farther the point, the less correlation between two points.

- Simple: taking $\sigma \to \infty$ results in the following kernel instead:

$$w_{ij} = \begin{cases} 1 & \mathbf{x}_i, \mathbf{x}_j \text{ connected} \\ 0 & \mathbf{x}_i, \mathbf{x}_j \text{ disconnected} \end{cases}$$

Spectral Analysis of Laplacian

We have now constructed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Assume this graph is connected, then construct a diagonal matrix D such that i.e. the column sums of W. The graph Laplacian is $L = D - W$. Solve for vectors $\Phi \in \mathbb{R}^n$ such that

$$L\Phi = \lambda D\Phi$$

Then take $\Phi_1, \ldots, \Phi_m$ where $0 \neq \lambda_1 \leq \ldots \leq \lambda_m \leq \ldots \leq \lambda_n$, then for every sample we encode it as:

$$\mathbf{x}_i \mapsto \begin{bmatrix} \Phi_1(i) & \ldots & \Phi_m(i) \end{bmatrix}$$

The Φs are known as the Laplacian eigenmap [3][4].

Remarks

Remark: Intuition of graph Laplacian

Laplacian is a matrix representation of a graph that encodes the "similarity" between the data points into each entry.

Remark: Isometry is not perfect

The general structure of the data is preserved in a local sense, but not in a global sense.

Spectral analysis of Markov chain

First we need some assumptions:

- Graph is connected then the Markov chain admits a unique stationary distribution:

$$\pi(\mathbf{y}) = \frac{d(\mathbf{y})}{\sum_{\mathbf{x}\in X} d(\mathbf{x})}$$

- The Markov chain is reversible:

$$\pi(\mathbf{x})p(\mathbf{x},\mathbf{y}) = \pi(\mathbf{y})p(\mathbf{y},\mathbf{x})$$

- X is finite.

Then P admits vectors $\{\psi_l\}_{l\geq 1}$ with eigenvalues $1 = \lambda_0 > |\lambda_1| \geq \ldots \geq |\lambda_n|$.

Why eigenvectors?

Markov chain with transition probability matrix P where:

$$P_{i,j} = p(\mathbf{x}_i, \mathbf{x}_j)$$

$P^t_{i,j}$ is the probability of transition from the i to j in t steps.
Analyzing entries of repeated matrix multiplication $\implies$ analysis of eigenvalues and eigenvectors.

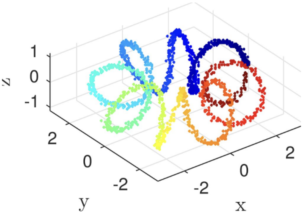

Figure: Toy data set (Wikipedia)

Diffusion distances and coordinates

The diffusion distance between two samples for a given t is

$$D_t(x,y) := ||p_t(x,.) - p_t(y,.)||^2_{L^2(X, d\mu/\pi)} = \int_X (p_t(x,u) - p_t(y,u))^2 \frac{d\mu(u)}{\pi(u)}$$

Thankfully, we can solve for this distance exactly [1]:

$$D_t(\mathbf{x},\mathbf{y}) = \Big(\sum_{l=1}^{n} \lambda_l^{2t} (\psi_l(\mathbf{x}) - \psi_l(\mathbf{y}))^2 \Big)^{\frac{1}{2}}$$

But we can get at least δ-close by choosing $n_{t,\delta} < n$ sufficiently large:

$$D_t(\mathbf{x},\mathbf{y}) = \Big(\sum_{l=1}^{n_{t,\delta}} \lambda_l^{2t} (\psi_l(\mathbf{x}) - \psi_l(\mathbf{y}))^2 \Big)^{\frac{1}{2}}$$

Diffusion distances and coordinates

Recall similarity kernel $K(\mathbf{x}_i, \mathbf{x}_j)$. With this kernel, construct a new kernel A

$$A(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sqrt{\pi(\mathbf{x}_i)}}{\sqrt{\pi(\mathbf{x}_j)}} p(\mathbf{x}_i, \mathbf{x}_j)$$

We're working in a finite measure space, so the map is compact. Encode this kernel into a matrix. This is a symmetric linear map, therefore we can make a spectral decomposition of this map.

$$A(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l \geq 1} \lambda_l \phi_l(\mathbf{x}_i) \phi_l(\mathbf{x}_j)$$

Diffusion distances and coordinates

$$\frac{\sqrt{\pi(\mathbf{x}_i)}}{\sqrt{\pi(\mathbf{x}_j)}} p(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l \geq 1} \lambda_l \phi_l(\mathbf{x}_i) \phi_l(\mathbf{x}_j)$$

$$p(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l \geq 1} \lambda_l \frac{\phi_l(\mathbf{x}_i)}{\sqrt{\pi(\mathbf{x}_i)}} \left(\sqrt{\pi(\mathbf{x}_j)} \phi_l(\mathbf{x}_j) \right)$$

$$p^t(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l \geq 1} \lambda_l^t \frac{\phi_l(\mathbf{x}_i)}{\sqrt{\pi(\mathbf{x}_i)}} \left(\sqrt{\pi(\mathbf{x}_j)} \phi_l(\mathbf{x}_j) \right)$$

Diffusion distances and coordinates

$$
\begin{aligned}
D_t(x,y) &= \int_X \left(p_t(x,u) - p_t(y,u)\right)^2 \frac{d\mu(u)}{\pi(u)} \\
&= \int_X \Bigg(\sum_{l\geq 1} \lambda_l^t \frac{\phi_l(x)}{\sqrt{\pi(x)}}\left(\sqrt{\pi(u)}\phi_l(u)\right) - \sum_{l\geq 1} \lambda_l^t \frac{\phi_l(y)}{\sqrt{\pi(y)}}\left(\sqrt{\pi(u)}\phi_l(u)\right)\Bigg)^2 d \\
&= \int_X \Bigg(\sum_{l\geq 1} \lambda_l^{2t} \left(\frac{\phi_l(x)}{\sqrt{\pi(x)}} - \frac{\phi_l(y)}{\sqrt{\pi(y)}}\right)^2 \left(\sqrt{\pi(u)}\phi_l(u)\right)^2 \frac{d\mu(u)}{\pi(u)} \\
&= \sum_{l\geq 1} \lambda_l^{2t} \left(\frac{\phi_l(x)}{\sqrt{\pi(x)}} - \frac{\phi_l(y)}{\sqrt{\pi(y)}}\right)^2 \int_X \left(\phi_l(u)\right)^2 d\mu(u) \\
&= \Bigg(\sum_{l\geq 1} \lambda_l^{2t} \left(\psi_l(x) - \psi_l(y)\right)\Bigg)^2
\end{aligned}
$$

Remarks

Remark: Computing the eigenvalues

By the construction of ϕ_l, it is left as an exercise to the viewer to show that $\psi(x) = \frac{\phi(x)}{\sqrt{\pi(x)}}$

Remark: Optimal embedding

ϕ_l are eigenfunctions that send the datapoints to the diffusion coordinate space.

References

[1] Ronald R. Coifman and Stéphane Lafon. “Diffusion maps”. In: *Applied and Computational Harmonic Analysis* 21.1 (2006). Special Issue: Diffusion Maps and Wavelets, pp. 5–30. ISSN: 1063-5203. DOI: https://doi.org/10.1016/j.acha.2006.04.006. URL: https://www.sciencedirect.com/science/article/pii/S1063520306000546.

[2] Roy R. Lederman and Bogdan Toader. “On Manifold Learning in Plato’s Cave: Remarks on Manifold Learning and Physical Phenomena”. In: *International Conference on Sampling Theory and Applications (SampTA 2023)* (2023).

[3] Partha Niyogi Mikhail Belkin. “Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering”. In: *NeurIPS Vol. 14 Issue 14* (2001), pp. 585–591.

[4] Partha Niyogi Mikhail Belkin. “Laplacian eigenmaps for dimensionality reduction and data representation”. In: *Neural*