Dimension reduction and manifold learning

From MDS to dimension reduction

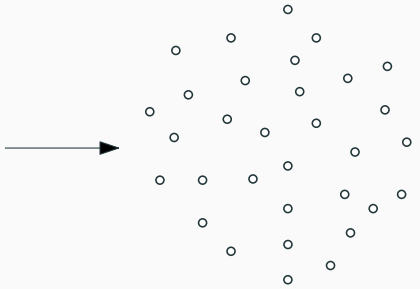
Eddie Aamari
Département de mathématiques et applications
CNRS, ENS PSL

Master MASH — Dauphine PSL

Recap on Multidimensional Scaling

The problem of multidimensional scaling

$$\begin{pmatrix} 0 & \delta_{1,2} & \cdots & \delta_{1,n} \\ \delta_{2,1} & 0 & \cdots & \delta_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n,1} & \delta_{n,2} & \cdots & 0 \end{pmatrix}$$



Given a weighted graph $(\mathcal{V}, \mathcal{E}, \delta)$ and embedding dimension d, find $y_1, \ldots, y_n \in \mathbb{R}^d$ such that

$$\|y_i - y_j\| \approx \delta_{ij}$$

for all (or most) $(i, j) \in \mathcal{E}$.

Classical Scaling

The main method for MDS is classical scaling (CS).

CS requires that all the dissimilarities be available, meaning, that the graph be *complete*.

The main method for MDS is classical scaling (CS).

CS requires that all the dissimilarities be available, meaning, that the graph be *complete*.

In that case, the input data can be gathered in a dissimilarity matrix

$$\Delta := \begin{pmatrix} \delta_{1,1} & \delta_{1,2} & \cdots & \delta_{1,n} \\ \delta_{2,1} & \delta_{2,2} & \cdots & \delta_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ \delta_{n,1} & \delta_{n,2} & \cdots & \delta_{n,n} \end{pmatrix} \in \mathbb{R}^{n \times n}$$

Step 1: Double-centering the matrix of squared dissimilarities

Form the matrix

$$\Delta_2^c = -\tfrac{1}{2} H \Delta^{\circ 2} H, \quad \text{where } H = I_{n\times n} - \tfrac{1}{n} 1 1^\top$$

Step 1: Double-centering the matrix of squared dissimilarities

Form the matrix

$$\Delta_2^c = -\tfrac{1}{2} H \Delta^{\circ 2} H, \quad \text{where } H = I_{n\times n} - \tfrac{1}{n} 1 1^\top$$

Step 2: Eigendecomposition

Let $\lambda_1 \geq \cdots \geq \lambda_d$ be the top d eigenvalues and $u_1, \ldots, u_d$ be corresponding unit eigenvectors of Δ_2^c

Step 1: Double-centering the matrix of squared dissimilarities

Form the matrix

$$\Delta_2^c = -\tfrac{1}{2} H \Delta^{\circ 2} H, \quad \text{where } H = I_{n\times n} - \tfrac{1}{n} 1 1^\top$$

Step 2: Eigendecomposition

Let $\lambda_1 \geq \cdots \geq \lambda_d$ be the top d eigenvalues and $u_1, \ldots, u_d$ be corresponding unit eigenvectors of Δ_2^c

Step 3: Embedding

Form the output matrix

$$Y_{\text{CS}} := \left(\sqrt{\max(\lambda_1, 0)} u_1 \mid \quad \cdots \quad \mid \sqrt{\max(\lambda_d, 0)}\, u_d \right) \in \mathbb{R}^{n\times d}$$

CS on a realizable graph

The graph $(\mathcal{V}, \mathcal{E}, \delta)$ is realizable in dimension p when there are some $x_1, \ldots, x_n \in \mathbb{R}^d$ such that $\delta_{ij} = \|x_i - x_j\|$ for all $(i, j) \in \mathcal{E}$.

Consider the Euclidean case and let $x_1, \ldots, x_n$ be a centered (w.l.o.g.) realizing point set, so that

$$\delta_{ij} = \|x_i - x_j\| \text{ and } \mathbb{E}_x[x] := \tfrac{1}{n} \textstyle\sum_i x_i = 0.$$

CS on a realizable graph

The graph $(\mathcal{V}, \mathcal{E}, \delta)$ is realizable in dimension p when there are some $x_1, \ldots, x_n \in \mathbb{R}^d$ such that $\delta_{ij} = \|x_i - x_j\|$ for all $(i, j) \in \mathcal{E}$.

Consider the Euclidean case and let $x_1, \ldots, x_n$ be a centered (w.l.o.g.) realizing point set, so that

$$\delta_{ij} = \|x_i - x_j\| \text{ and } \mathbb{E}_x[x] := \tfrac{1}{n} \textstyle\sum_i x_i = 0.$$

In that case, the key idea is to convert the dissimilarities (here Euclidean distances) into inner products to obtain a Gram matrix.

Double centering ≡ Transformation into Gram matrix

When $(\mathcal{V}, \mathcal{E}, \delta)$ is realizable in dimension p through centered point cloud

$$X = (x_1 | \cdots | x_n)^\top \in \mathbb{R}^{n \times p},$$

polarization yields

$$\begin{aligned}\langle x_i, x_j \rangle &= -\frac{1}{2}(\delta_{ij}^2 - \langle x_i, x_i \rangle - \langle x_j, x_j \rangle) \\ &= -\frac{1}{2}\Big(\delta_{ij}^2 - \frac{1}{n}\sum_l \delta_{il}^2 - \frac{1}{n}\sum_k \delta_{kj}^2 + \frac{1}{n^2}\sum_k \sum_l \delta_{kl}^2\Big).\end{aligned}$$

Hence, the matrix form of the above is

$$XX^\top = -\tfrac{1}{2} H \Delta^{\circ 2} H,$$

so the doubly centered matrix is the Gram matrix of X.

Matrix Form for Classical Scaling

Write $X = USV^\top$ for the singular value decomposition of $X \in \mathbb{R}^{n\times p}$:

- $U = (u_1|\cdots|u_n) \in \mathbb{R}^{n\times n}$ is orthogonal $(U^\top U = UU^\top = I_p)$
- $S \in \mathbb{R}^{n\times p}$ is diagonal (entries $\mu_1 \geq \ldots \mu_{\min\{n,p\}} \geq 0$)
- $V = (v_1|\cdots|v_p) \in \mathbb{R}^{p\times p}$ is orthogonal $(V^\top V = VV^\top = I_p)$

With these notation,

$$XX^\top = USS^\top U^\top.$$

Hence, the output of classical scaling writes as

$$Y_{\mathrm{CS}} = \begin{pmatrix} \mu_1 u_1 \mid & \cdots & \mid \mu_d u_d \end{pmatrix} = US_{*,[d]},$$

where $S_{*,[d]} := S \begin{pmatrix} I_{d\times d} \\ 0_{(p-d)\times d} \end{pmatrix} \in \mathbb{R}^{n\times d}$ is the first d columns of S.

Dimensionality Reduction

Given points $x_1, \ldots, x_n \in \mathbb{R}^p$ and embedding dimension d, find $y_1, \ldots, y_n \in \mathbb{R}^d$ such that

$$\|y_i - y_j\| \approx \|x_i - x_j\|$$

for all (or most) $i, j \in \{1, \ldots, n\}$.

Dimensionality Reduction

Motivations

- Computational challenges (time complexity)
- Generalization ability (curse of dimensionality)
- Data interpretation (meaningful structure, visualization)

Dimensionality Reduction

Motivations

- Computational challenges (time complexity)
- Generalization ability (curse of dimensionality)
- Data interpretation (meaningful structure, visualization)

Wilderness

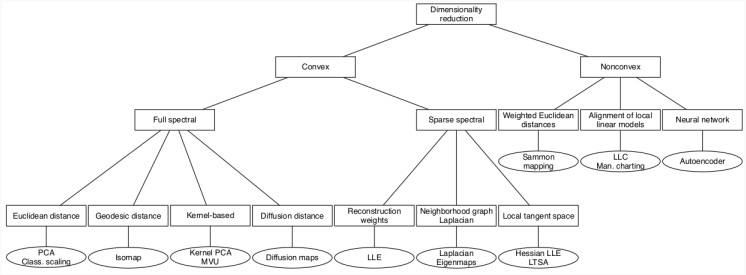


Figure 1: from Van Der Maaten, Postma, Herik, et al. 2009

Random Projections

Johnson Lindenstrauss

Define the distortion of an embedding $\mathbb{R}^{n\times n} \ni \Delta \mapsto Y \in \mathbb{R}^{n\times d}$ as

$$\operatorname{distorsion}(\Delta \mid Y) := \max_{i\neq j} \frac{\delta_{ij} \vee \|y_i - y_j\|}{\delta_{ij} \wedge \|y_i - y_j\|}.$$

Recall the following embeddability result.

Johnson Lindenstrauss

Define the distortion of an embedding $\mathbb{R}^{n\times n} \ni \Delta \mapsto Y \in \mathbb{R}^{n\times d}$ as

$$\operatorname{distorsion}(\Delta \mid Y) := \max_{i\neq j} \frac{\delta_{ij} \vee \|y_i - y_j\|}{\delta_{ij} \wedge \|y_i - y_j\|}.$$

Recall the following embeddability result.

Theorem (Bourgain 1985)

There exists a universal constant $C > 0$ such that any finite metric space of cardinality n can be embedded into $(\mathbb{R}^d, \|\cdot\|_2)$ with $d \leq C \log n$ and with distortion at most $C \log n$.

Johnson Lindenstrauss

Define the distortion of an embedding $\mathbb{R}^{n\times n} \ni \Delta \mapsto Y \in \mathbb{R}^{n\times d}$ as

$$\operatorname{distorsion}(\Delta \mid Y) := \max_{i\neq j} \frac{\delta_{ij} \vee \|y_i - y_j\|}{\delta_{ij} \wedge \|y_i - y_j\|}.$$

Recall the following embeddability result.

Theorem (Bourgain 1985)

There exists a universal constant $C > 0$ such that any finite metric space of cardinality n can be embedded into $(\mathbb{R}^d, \|\cdot\|_2)$ with $d \leq C \log n$ and with distortion at most $C \log n$.

A simple adaptation of his arguments show that the same is true for $(\mathbb{R}^d, \|\cdot\|_p)$, this time with $d \leq C(\log n)^2$ (Matoušek 2013).

Theorem (Johnson and Lindenstrauss 1984)

Let $\mathcal{X} \subset \mathbb{R}^p$ be a finite point cloud with $|\mathcal{X}| = n$, and $A \in \mathbb{R}^{d \times p}$ be a random matrix with entries $(A_{i,j})_{\substack{i \leq p \\ j \leq d}}$ are iid $\mathcal{N}(0, 1/d)$. If $d \geq 16\varepsilon^{-2} \log(n/\sqrt{t})$, then with probability at least $1 - t$, for all $x, x' \in \mathcal{X}$,

$$(1 - \varepsilon)\|x - x'\|_2^2 \leq \|Ax - Ax'\|_2^2 \leq (1 + \varepsilon)\|x - x'\|_2^2.$$

The embedding $\mathbb{R}^p \ni x \mapsto Ax \in \mathbb{R}^d$ is an ε-isometry on $\mathcal{X}$.

Theorem (Johnson and Lindenstrauss 1984)

Let $\mathcal{X} \subset \mathbb{R}^p$ be a finite point cloud with $|\mathcal{X}| = n$, and $A \in \mathbb{R}^{d \times p}$ be a random matrix with entries $(A_{i,j})_{\substack{i \leq p \\ j \leq d}}$ are iid $\mathcal{N}(0, 1/d)$. If $d \geq 16\varepsilon^{-2} \log(n/\sqrt{t})$, then with probability at least $1 - t$, for all $x, x' \in \mathcal{X}$,

$$(1 - \varepsilon)\|x - x'\|_2^2 \leq \|Ax - Ax'\|_2^2 \leq (1 + \varepsilon)\|x - x'\|_2^2.$$

The embedding $\mathbb{R}^p \ni x \mapsto Ax \in \mathbb{R}^d$ is an ε-isometry on $\mathcal{X}$.

The requirement on the reduced dimension d to be

$$d \gtrsim \varepsilon^{-2} \log(n/\sqrt{t})$$

does not depend on the original dimension p.

Proof outline for Johnson Lindenstrauss

- For all $Q \in \mathcal{O}(\mathbb{R}^d)$, AQ has the same distribution as A. Hence, if $\|x\| = 1$,

$$Ax \sim Ae_1 = (A_{1,1}A_{2,1} \cdots A_{d,1})^\top.$$

- On average, A is an exact isometry, in the sense that

$$\forall x \in \mathbb{R}^p, \quad \mathbb{E}\left[\|Ax\|_2^2\right] = \mathbb{E}\left[\sum_{i=1}^{d} A_{i,1}^2\right] \|x\|_2^2 = \|x\|_2^2.$$

  In fact, if $x \neq 0$, one has $\dfrac{\|Ax\|_2^2}{\|x\|_2^2} \sim \chi_d^2$.

- If $Z \sim \chi_d^2$, then $\mathbb{P}\left(|Z/d - 1| > t\right) \leq 2\exp(-dt^2/8)$.

Recovery from Random Projections

Johnson-Lindenstrauss' lemma is related to compressed sensing.

Theorem (Shalev-Shwartz and Ben-David 2014, Theorem 23.7)

If $\varepsilon < 2/5$ and $s < p$, then with high probability, for all $x \in \mathbb{R}^p$ such that $\|x\|_0 \leq s$,

$$x \in \underset{\substack{u \in \mathbb{R}^p \\ Au = Ax}}{\arg\min} \|u\|_1,$$

where $\|x\|_0 := \sum_{k=1}^{p} \mathbf{1}_{x^{(k)} \neq 0}$ and $\|u\|_1 := \sum_{k=1}^{p} |u^{(k)}|$.

Refinements of Johnson Lindenstrauss

- A fully-fledged line of research studies ways to construct A and to compute Ax faster than $O(pd)$, by including:
  - specific product structure (Ailon and Chazelle 2006)
  - sparsity (Garivier and Pilliat 2024; Kane and Nelson 2014)
  - tensor structure (Kasiviswanathan et al. 2010)
- Refined analyses when $\mathcal{X}$ lies on a manifold:
  - General upper and lower bounds in Iwen, Tavakoli, and Schmidt 2021 and Eftekhari and Wakin 2015
  - Accelerated versions in Iwen, Schmidt, and Tavakoli 2021

Coding and Decoding

What low-dimensional structure "best approximates" the data $\mathcal{X} = \{x_1, \ldots, x_n\}$?

Coding and Decoding

What low-dimensional structure "best approximates" the data $\mathcal{X} = \{x_1, \dots, x_n\}$?

Consider the square loss $\|\cdot\|_2^2$, and the generic reconstruction error

$$E_{\text{codec}} := \mathbb{E}_x \left[\|x - \text{dec}(\text{cod}(x))\|_2^2 \right],$$

where

$$\text{cod} : \mathbb{R}^p \to \mathbb{R}^d \quad \text{and} \quad \text{dec} : \mathbb{R}^d \to \mathbb{R}^p$$

This formulation suggests

- The existence of latent variables $y = \text{cod}(x)$ of low dimension.
- Otherwise, a lossy compression of the signal $x_1, \dots, x_n \in \mathbb{R}^p$.

Principal Component Analysis

Principal Component Analysis

Choosing linear maps as encoders and decoders leads to Principal Component Analysis (PCA).

This foundational approach dates back to Pearson 1901. Here,

$$\mathrm{cod} : \mathbb{R}^p \ni x \mapsto Ax \in \mathbb{R}^d \quad \text{and} \quad \mathrm{dec} : \mathbb{R}^d \ni x \to Bx \in \mathbb{R}^p,$$

where $x \in \mathbb{R}^p$ is a column vector, $A \in \mathbb{R}^{d \times p}$ and $B \in \mathbb{R}^{p \times d}$.

We are brought to the minimization problem

$$E_{\mathrm{PCA}} = \min_{A \in \mathbb{R}^{d \times p}, B \in \mathbb{R}^{p \times d}} \mathbb{E}_x \|x - BAx\|_2^2.$$

For simplicity, we assume that $\mathbb{E}_x[x] = 0$.

$$\min_{A\in\mathbb{R}^{d\times p},B\in\mathbb{R}^{p\times d}} \mathbb{E}_x\|x - BAx\|_2^2 \quad \min_{A\in\mathbb{R}^{d\times p},B\in\mathbb{R}^{p\times d}} \|X^\top - BAX^\top\|_{\mathrm{F}}^2.$$

Lemma

If $p \geq d$, then there exists $B \in \mathbb{R}^{p\times d}$ such that $B^\top B = I_{d\times d}$ and $(A, B) = (B^\top, B)$ is solution.

$$\min_{A\in\mathbb{R}^{d\times p},B\in\mathbb{R}^{p\times d}} \mathbb{E}_x\|x - BAx\|_2^2 \quad \min_{A\in\mathbb{R}^{d\times p},B\in\mathbb{R}^{p\times d}} \|X^\top - BAX^\top\|_{\mathrm{F}}^2.$$

Lemma

If $p \geq d$, then there exists $B \in \mathbb{R}^{p\times d}$ such that $B^\top B = I_{d\times d}$ and $(A, B) = (B^\top, B)$ is solution.

$BA = BB^\top \in \mathbb{R}^{p\times p}$ is a d-dimensional orthogonal projection.

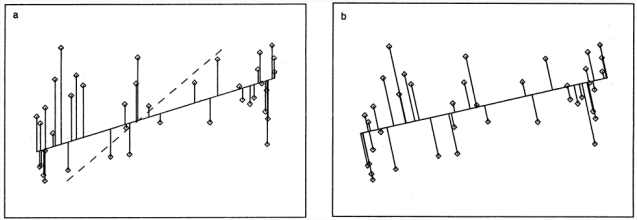


Figure 2: PCA vs linear regression (Hastie and Stuetzle 1989)

PCA vs Linear Regression

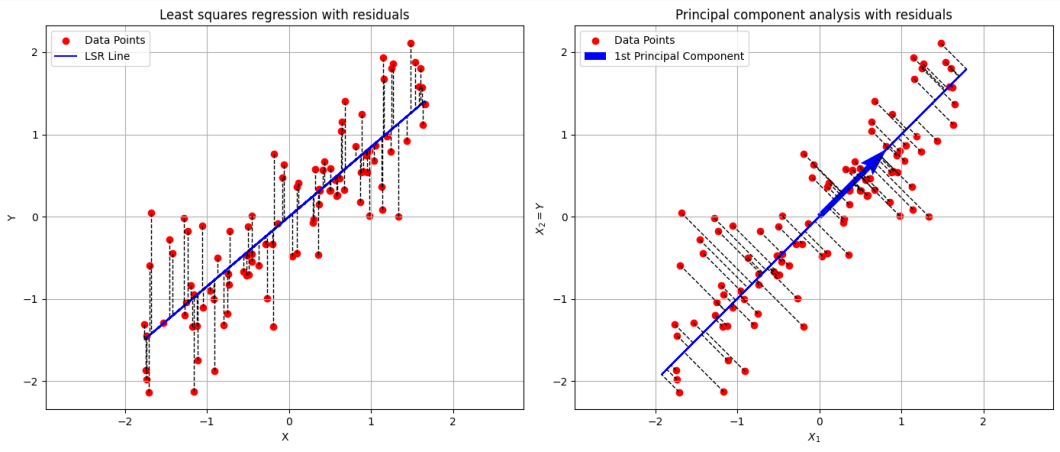


Figure 3: PCA and linear regression applied on the same dataset.

Reformulating PCA

For all $B \in \mathbb{R}^{p \times d}$ such that $B^\top B = I_{d \times d}$,

$$\begin{aligned}
\mathbb{E}_x \left[\|x - BB^\top x\|_2^2 \right] &= \mathbb{E}_x \left[x^\top x - 2x^\top BB^\top x + x^\top BB^\top BB^\top x \right] \\
&= \mathbb{E}_x[x^\top x] - \mathbb{E}_x \left[x^\top BB^\top x \right] \\
&= \mathbb{E}_x[x^\top x] - \mathbb{E}_x \left[\mathrm{Tr}(x^\top BB^\top x) \right] \\
&= \mathbb{E}_x[x^\top x] - \mathrm{Tr} \left(B^\top \mathbb{E}_x[xx^\top] B \right).
\end{aligned}$$

Output of PCA

PCA amounts to the optimization problem

$$\max_{\substack{B\in\mathbb{R}^{p\times d}\\ B^\top B=I_{d\times d}}} \operatorname{Tr}\left(B^\top \mathbb{E}_x[xx^\top]B\right).$$

Writing $X := (x_1|\cdots|x_n)^\top \in \mathbb{R}^{n\times p}$, we recognize covariance matrix

$$\mathbb{E}_x[xx^\top] = X^\top X,$$

Output of PCA

PCA amounts to the optimization problem

$$\max_{\substack{B\in\mathbb{R}^{p\times d}\\ B^\top B=I_{d\times d}}} \operatorname{Tr}\left(B^\top \mathbb{E}_x[xx^\top]B\right).$$

Writing $X := (x_1|\cdots|x_n)^\top \in \mathbb{R}^{n\times p}$, we recognize covariance matrix

$$\mathbb{E}_x[xx^\top] = X^\top X,$$

The optimizer $B = V_{*,[d]} = (v_1|\cdots|v_d) \in \mathbb{R}^{p\times d}$ is the matrix of (normalized) top eigenvectors of $X^\top X$.

The dimension-reduced data is then given by

$$Y_{\mathrm{PCA}} = XV_{*,[d]}.$$

(i.e. optimal coding is $\mathrm{cod}_{\mathrm{PCA}} : x \mapsto V_{*,[d]}^\top x$)

Matrix Form for Principal Component Analysis

Write $X = USV^\top$ for a singular value decomposition of $X \in \mathbb{R}^{n \times p}$:

- $U = (u_1 | \cdots | u_n) \in \mathbb{R}^{n \times n}$ is orthogonal $(U^\top U = UU^\top = I_p)$
- $S \in \mathbb{R}^{n \times p}$ is diagonal (entries $\mu_1 \geq \ldots \mu_{\min\{n,p\}} \geq 0$)
- $V = (v_1 | \cdots | v_p) \in \mathbb{R}^{p \times p}$ is orthogonal $(V^\top V = VV^\top = I_p)$

With these notation,

$$X^\top X = VS^\top SV^\top.$$

Hence, the output of principal component analysis writes as

$$Y_{\mathrm{PCA}} = XV_{*,[d]},$$

where $V_{*,[d]} := V \begin{pmatrix} I_{d \times d} \\ 0_{(p-d) \times d} \end{pmatrix} \in \mathbb{R}^{p \times d}$ is the first d columns of V.

PCA vs CS

Classical Scaling with $\delta_{i,j} = \|x_i - x_j\|_2$ works with the Gram matrix $XX^\top$ of data, and outputs

$$Y_{\mathrm{CS}} = US_{*,[d]} = US \begin{pmatrix} I_{d\times d} \\ 0_{(p-d)\times d} \end{pmatrix}.$$

Principal Component Analysis works with the covariance matrix $X^\top X$ of data, and outputs reduced variables

$$Y_{\mathrm{PCA}} = XV_{*,[d]} = USV^\top V \begin{pmatrix} I_{d\times d} \\ 0_{(p-d)\times d} \end{pmatrix} = US \begin{pmatrix} I_{d\times d} \\ 0_{(p-d)\times d} \end{pmatrix}.$$

PCA vs CS

Classical Scaling with $\delta_{i,j} = \|x_i - x_j\|_2$ works with the Gram matrix $XX^\top$ of data, and outputs

$$Y_{\mathrm{CS}} = US_{*,[d]} = US \begin{pmatrix} I_{d\times d} \\ 0_{(p-d)\times d} \end{pmatrix}.$$

Principal Component Analysis works with the covariance matrix $X^\top X$ of data, and outputs reduced variables

$$Y_{\mathrm{PCA}} = XV_{*,[d]} = USV^\top V \begin{pmatrix} I_{d\times d} \\ 0_{(p-d)\times d} \end{pmatrix} = US \begin{pmatrix} I_{d\times d} \\ 0_{(p-d)\times d} \end{pmatrix}.$$

Classical Scaling computed with $\delta_{i,j} = \|x_i - x_j\|_2$

$\Leftrightarrow$

Principal Component Analysis

Principal Component Analysis can be defined via the optimization of the:

- reconstruction error (Pearson 1901)
- variance preservation
- distance preservation
- decorrelation

From the calculations above, the optimal reconstruction error is

$$E_{\mathrm{PCA}} = \sum_{k=p-d+1}^{p} \lambda_k.$$

Principal Component Analysis can be defined via the optimization of the:

- reconstruction error (Pearson 1901)
- variance preservation
- distance preservation
- decorrelation

From the calculations above, the optimal reconstruction error is

$$E_{\mathrm{PCA}} = \sum_{k=p-d+1}^{p} \lambda_k.$$

Other names in other fields

- Principal component analysis (statistics)
- Karhunen–Loève decomposition (stochastic processes)
- Proper orthogonal decomposition (mechanics)
- Truncated Schmidt decomposition / SVD (signal processing)

Exact Recovery

PCA allows exact recovery if (equivalently)

- $\dim(\mathrm{Span}(x_1, \ldots, x_n)) \leq d$, or
- $x_i = By_i$ for some $B \in \mathbb{R}^{p \times d}$ and $y_1, \ldots, y_n \in \mathbb{R}^d$.

PCA vs JL

They have both specific recovery properties.

PCA guarantees exact recovery whenever the variables $x_1, \ldots, x_n$ lie in a d-plane of $\mathbb{R}^p$,

Random projections guarantee exact recovery whenever the original data is sparse (in a given orthogonal basis).

Axes Interpretation for PCA

The columns of $Y_{\mathrm{PCA}} = (Y^{(1)}|\cdots|Y^{(d)}) \in \mathbb{R}^{n\times d}$ can be interpreted based on those of $X = (X^{(1)}|\cdots|X^{(p)}) \in \mathbb{R}^{n\times p}$ through a correlation circle.

For all $k \in \{1,\dots,p\}$ and $\ell \in \{1,\dots,d\}$, write

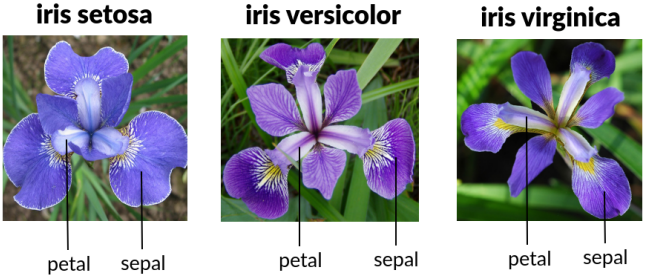
$$\mathrm{Corr}(X^{(k)}, Y^{(\ell)}) := \frac{\langle X^{(k)}, Y^{(\ell)}\rangle}{\|X^{(k)}\|\|Y^{(\ell)}\|}.$$

Each initial feature $k \in \{1,\dots,p\}$ can then be represented through the d-dimensional vector

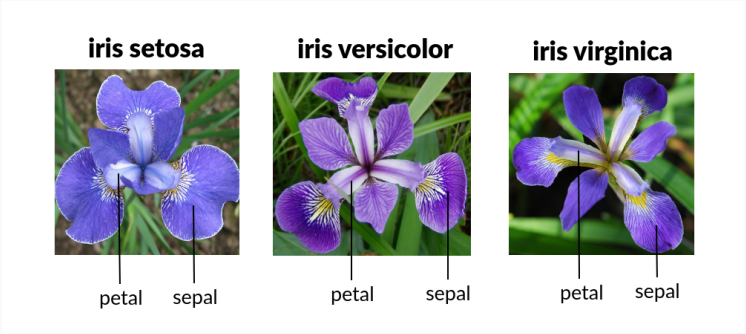
$$C^{(k)} := \big(\mathrm{Corr}(X^{(k)}, Y^{(\ell)})\big)_{\ell\leq d},$$

which lies in the unit ball of $\mathbb{R}^d$.

Iris dataset

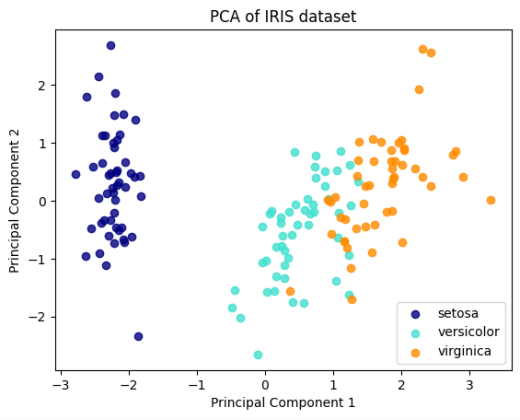


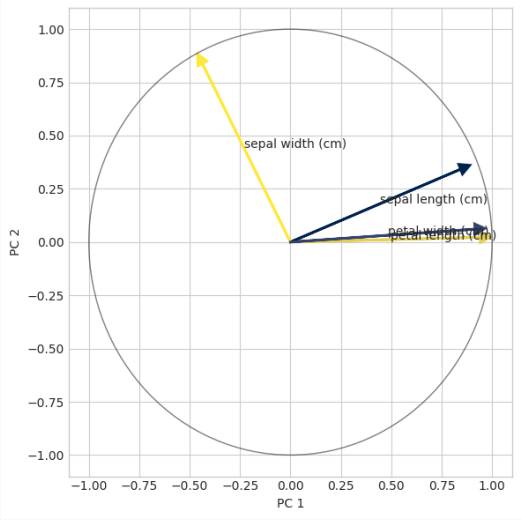
Iris dataset



To python!

Iris PCA axes





Sparse PCA

Limitations of PCA

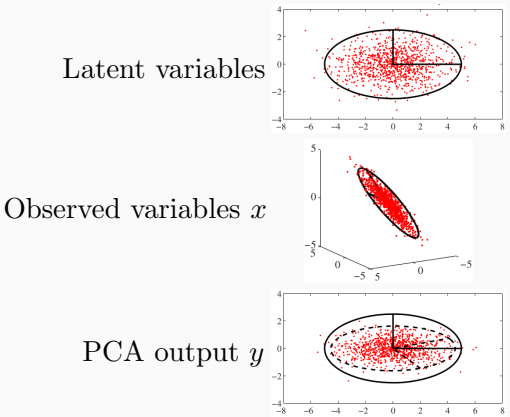


Figure 4: from Lee and Verleysen 2007

Limitations of PCA

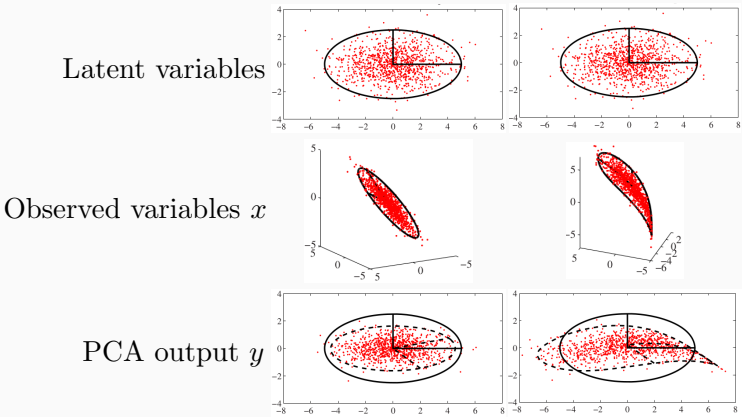


Figure 4: from Lee and Verleysen 2007

Nonlinear Dimension Reduction via Multidimensional Scaling

Dimensionality should try to unroll the curve.

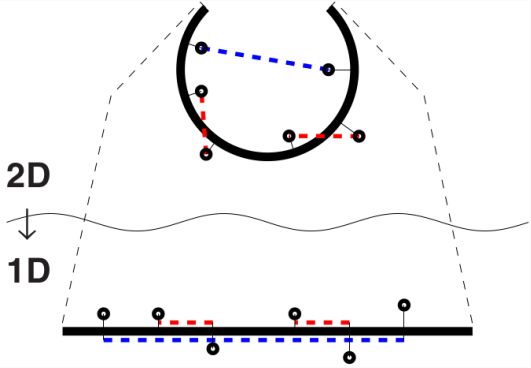


Figure 5: from Lee and Verleysen 2007

Only Trust Short Distances!

To unroll, we can try to mimic geodesic distances.

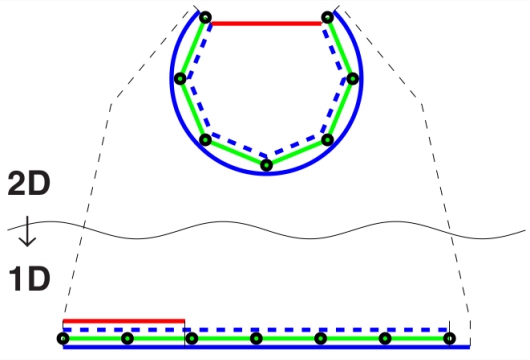


Figure 6: from Lee and Verleysen 2007

Isomap

Isomap stands for *isometric feature mapping*.

It originates from Tenenbaum 1997, for image data.

The idea is to use graph distances as an approximation of the geodesic distances.

Reminiscent of MDS-Diagram (Kruskal and Seery 1980)

Isomap

We presented a variant of the original (Tenenbaum 1997).

Step 1: Graph

Construct the r-neighborhood graph $(\mathcal{V} = \{1, \dots, n\}, \mathcal{E}, \delta)$ of $x_1, \dots, x_n \in \mathbb{R}^p$:

$$(x_i, x_j) \in \mathcal{E} \Leftrightarrow \delta_{i,j} := \|x_i - x_j\|_2 \leq r$$

Isomap

We presented a variant of the original (Tenenbaum 1997).

Step 1: Graph

Construct the r-neighborhood graph $(\mathcal{V} = \{1, \dots, n\}, \mathcal{E}, \delta)$ of $x_1, \dots, x_n \in \mathbb{R}^p$:

$$(x_i, x_j) \in \mathcal{E} \Leftrightarrow \delta_{i,j} := \|x_i - x_j\|_2 \leq r$$

Step 2: Manifold distance measure

Augment it to the complete graph $(\mathcal{V}, \bar{\mathcal{E}} = \binom{n}{2}, \bar{\delta})$, with

$$\bar{\delta}_{i,j} := \mathrm{d}_{(\mathcal{V},\mathcal{E},\delta)}(i, j).$$

Isomap

We presented a variant of the original (Tenenbaum 1997).

Step 1: Graph

Construct the r-neighborhood graph $(\mathcal{V} = \{1, \dots, n\}, \mathcal{E}, \delta)$ of $x_1, \dots, x_n \in \mathbb{R}^p$:

$$(x_i, x_j) \in \mathcal{E} \Leftrightarrow \delta_{i,j} := \|x_i - x_j\|_2 \leq r$$

Step 2: Manifold distance measure

Augment it to the complete graph $(\mathcal{V}, \bar{\mathcal{E}} = \binom{n}{2}, \bar{\delta})$, with

$$\bar{\delta}_{i,j} := \mathrm{d}_{(\mathcal{V},\mathcal{E},\delta)}(i, j).$$

Step 3: Isometric Euclidean embedding

Apply classiscal scaling to $(\mathcal{V}, \bar{\mathcal{E}}, \bar{\delta})$.

Fifty Shapes of Isomap

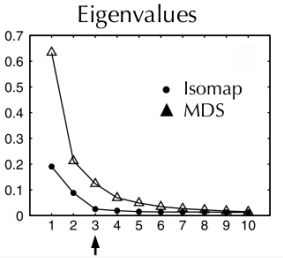
Building the weight matrix can be done

- Starting from a r-neighborhood graph
- Using k-nearest neighbors, oriented or non-oriented
- Plugging any other geodesic distance estimator

The embedding can be done using

- Other losses than the stress
- Incremental approaches

Isomap Examples



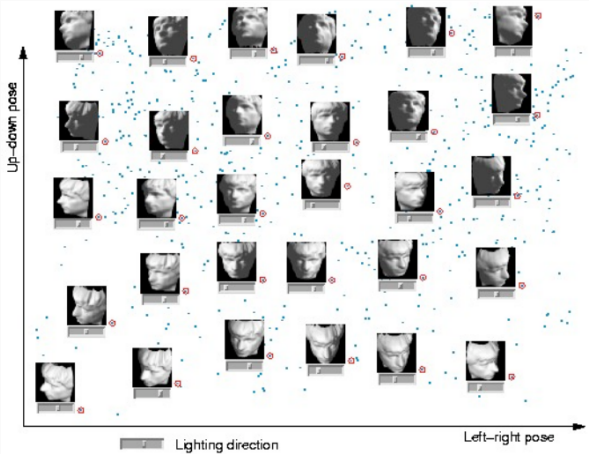


Figure 7: from Tenenbaum, Silva, and Langford 2000. Here, $p = 64 \times 64$, $n = 698$ and $k = 6$.

Isomap Examples

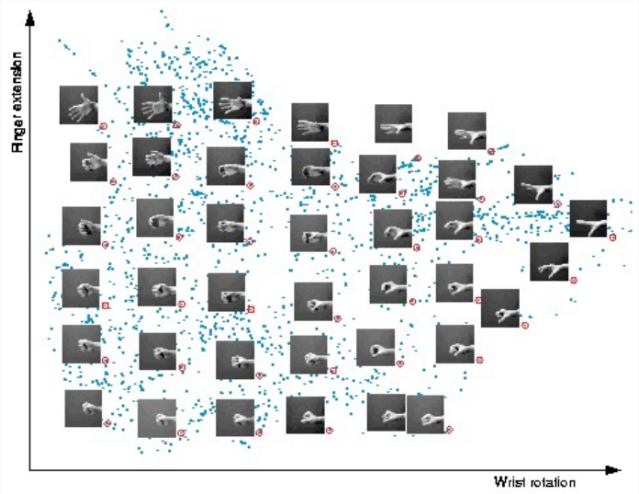


Figure 7: from Tenenbaum, Silva, and Langford 2000. $p = 64 \times 64$, $n = 2000$ and $k = 6$.

Applications of Isomap

Interpolations in the embedding space along straight lines.

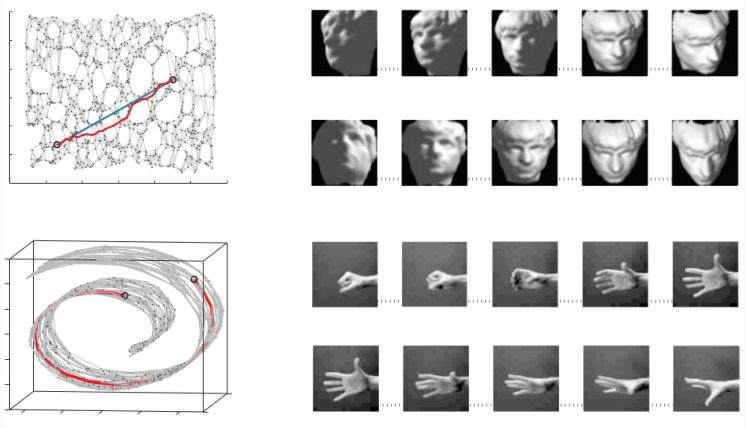


Figure 8: from Tenenbaum, Silva, and Langford 2000.

Isomap Strengths & Weaknesses

Strengths inherited from Classical Scaling

- Polynomial time algorithm
- No local optima
- Non-iterative
- Indicator for intrinsic dimensionality estimate

Isomap has a bandwidth parameter

- Neighborhood size r or k

Isomap Strengths & Weaknesses

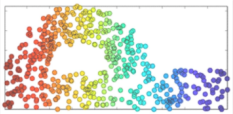
Strengths inherited from Classical Scaling

- Polynomial time algorithm
- No local optima
- Non-iterative
- Indicator for intrinsic dimensionality estimate

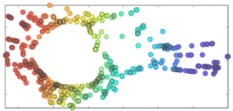
Isomap has a bandwidth parameter

- Neighborhood size r or k

Isomap cannot handle "non-convex" manifolds (i.e. with holes)



Input



IsoMap embedding

e.g. periodic motion

When can Isomap work without distortion?

When data lies on a d-submanifold M isometric to a convex. Indeed,

- Classical Scaling returns a lossless embedding only if there exists $\mathcal{Y} \subset \mathbb{R}^d$ such that $\bar{\delta}_{i,j} = \|y_i - y_j\|_2$ for all i, j.
- In the limit $r \to 0$ and $n \to \infty$, we expect that the graph shortest-path distance $\bar{\delta}_{i,j}$ will converge to the geodesic distance $\mathrm{d}_M(x_i, x_j)$ over M.

When can Isomap work without distortion?

When data lies on a d-submanifold M isometric to a convex. Indeed,

- Classical Scaling returns a lossless embedding only if there exists $\mathcal{Y} \subset \mathbb{R}^d$ such that $\bar{\delta}_{i,j} = \|y_i - y_j\|_2$ for all i, j.
- In the limit $r \to 0$ and $n \to \infty$, we expect that the graph shortest-path distance $\bar{\delta}_{i,j}$ will converge to the geodesic distance $\mathrm{d}_M(x_i, x_j)$ over M.

In the limit, this leads to the existence of a chart $\mathrm{cod} : M \to \Omega \subset \mathbb{R}^d$ such that for all $x, x' \in M$,

$$\mathrm{d}_M(x, x') = \|\mathrm{cod}(x) - \mathrm{cod}(x')\|_2.$$

When Ω is not convex, Isomap can be biased.

(think of $M = \Omega$ being an annulus)

Definition

We say that $M \subset \mathbb{R}^p$ is isometric to a convex if there exists

- a convex domain $\Omega \subset \mathbb{R}^d$ and
- a chart $\mathrm{cod} : M \to \Omega$ such that for all $x, x' \in M$,

$$\mathrm{d}_M(x, x') = \|\mathrm{cod}(x) - \mathrm{cod}(x')\|_2.$$

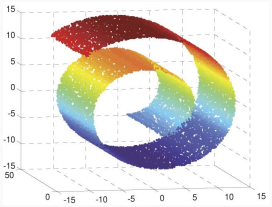


Figure 9: The (in)famous *Swiss roll* is isometric to a 2-rectangle.

Convergence of Isomap

Given a sample $\mathcal{X} \subset M$, we write

$$\varepsilon = \mathsf{d}_{\mathrm{H}}(M|\mathcal{X}) := \sup_{p \in M} \min_{y \in \mathcal{X}} \|y - p\|_2.$$

Theorem (Arias-Castro, Javanmard, and Pelletier 2020)

Assume that M and ∂M are compact and $\mathcal{C}^2$ smooth with reach $\mathrm{rch} > 0$, and that M is isometric to a convex. Write $z_1, \ldots, z_n \in \mathbb{R}^d$ for some (exact) embedding of $x_1, \ldots, x_n \in M$.

If $\varepsilon \lesssim r \lesssim \mathrm{rch}$, then Isomap outputs points $y_1, \ldots, y_n \in \mathbb{R}^d$ such that

$$\min_{Q \in \mathcal{O}(\mathbb{R}^d)} \left(\frac{1}{n} \sum_{i=1}^{n} \|z_i - Q y_i\|_2^2 \right)^{1/2} \lesssim \max\left\{ \left(\frac{r}{\mathrm{rch}}\right)^2, \left(\frac{\varepsilon}{r}\right)^2 \right\}.$$

Convergence of Isomap

Given a sample $\mathcal{X} \subset M$, we write

$$\varepsilon = \mathrm{d_H}(M|\mathcal{X}) := \sup_{p \in M} \min_{y \in \mathcal{X}} \|y - p\|_2.$$

Theorem (Arias-Castro, Javanmard, and Pelletier 2020)

Assume that M and ∂M are compact and $\mathcal{C}^2$ smooth with reach $\mathrm{rch} > 0$, and that M is isometric to a convex. Write $z_1, \ldots, z_n \in \mathbb{R}^d$ for some (exact) embedding of $x_1, \ldots, x_n \in M$.

If $\varepsilon \lesssim r \lesssim \mathrm{rch}$, then Isomap outputs points $y_1, \ldots, y_n \in \mathbb{R}^d$ such that

$$\min_{Q \in \mathcal{O}(\mathbb{R}^d)} \left(\frac{1}{n} \sum_{i=1}^n \|z_i - Q y_i\|_2^2 \right)^{1/2} \lesssim \max\left\{ \left(\frac{r}{\mathrm{rch}}\right)^2, \left(\frac{\varepsilon}{r}\right)^2 \right\}.$$

Other results in Arias-Castro and Le Gouic 2019; Bernstein et al. 2000.

Geodesic distance estimation

For all $x, x' \in \mathcal{X}$, write $\Delta_r(x, x')$ for the geodesic distance between x and x' in the r-neighborhood graph of $\mathcal{X}$, i.e. shortest path distance associated with the metric

$$\delta(x, x') = \begin{cases} \|x - x'\|_2 & \text{if } \|x - x'\|_2 \leq r; \\ \infty & \text{otherwise.} \end{cases}$$

Theorem (Arias-Castro and Le Gouic 2019)

If M is compact with reach bounded by rch, *and $\varepsilon \leq r/4$ with $r \leq c\,\mathrm{rch}$, then for all $x, x' \in \mathcal{X}$,*

$$(1 + Cr^2)^{-1} \mathrm{d}_M(x, x') \leq \Delta_r(x, x') \leq (1 + 4\varepsilon/r) \mathrm{d}_M(x, x').$$

Locally Linear Embedding (LLE)

Locally Linear Embedding is an alternative to Isomap.

It originates from Roweis and Saul 2000.

Tends to preserve the geometry via barycentric coordinates.

Locally Linear Embedding (LLE)

Locally Linear Embedding is an alternative to Isomap.

It originates from Roweis and Saul 2000.

Tends to preserve the geometry via barycentric coordinates.

Idea

- If the manifold is locally linear, we expect each data point to lie near a locally linear patch of the manifold.
- Characterize each point y_i as a convex linear combination of its k-nearest neighbors.
- Build an embedding that preserves these weights.

Locally Linear Embedding (LLE)

Step 1: Neighborhood. Find the neighbors $\mathcal{N}_i$ of each point.

Locally Linear Embedding (LLE)

Step 1: Neighborhood. Find the neighbors $\mathcal{N}_i$ of each point.

Step 2: Barycentric coordinates. Compute weights $W^* = (w^*_{i,j})_{1 \le i,j \le n}$ that best reconstruct the y_i's as convex sums of their neighbors.

$$W^* \in \underset{\substack{w_{i,j} \ge 0 \\ \sum_j w_{i,j} = 1}}{\arg\min} \sum_{i=1}^{n} \left\| x_i - \sum_{j \in \mathcal{N}_i} w_{i,j} x_j \right\|_2^2$$

Locally Linear Embedding (LLE)

Step 1: Neighborhood. Find the neighbors $\mathcal{N}_i$ of each point.

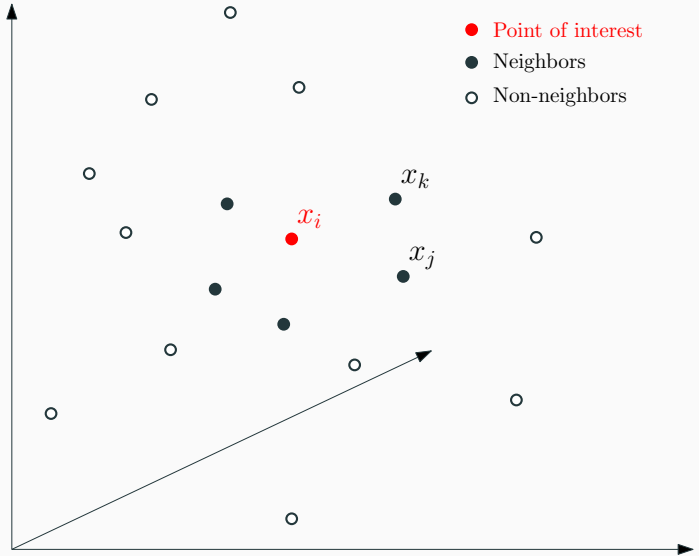
Step 2: Barycentric coordinates. Compute weights $W^* = (w^*_{i,j})_{1 \le i,j \le n}$ that best reconstruct the y_i's as convex sums of their neighbors.

$$W^* \in \underset{\substack{w_{i,j} \ge 0 \\ \sum_j w_{i,j} = 1}}{\arg\min} \sum_{i=1}^{n} \left\| x_i - \sum_{j \in \mathcal{N}_i} w_{i,j} x_j \right\|_2^2$$

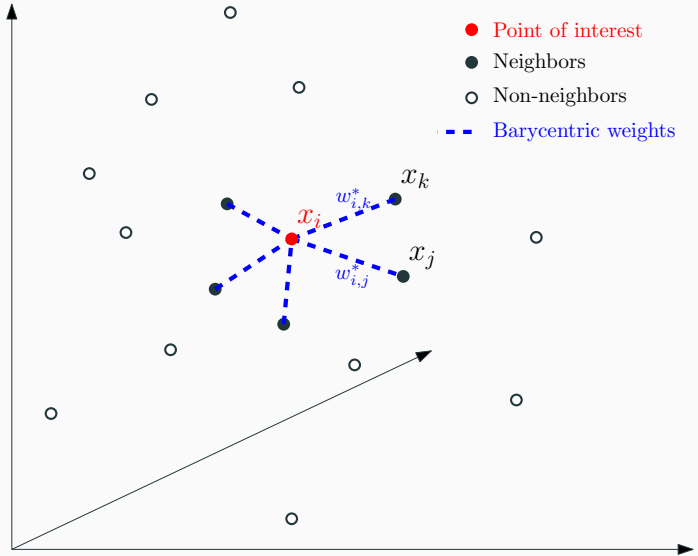
Step 3: Embedding. Embed using the previously computed weights W^*.

$$\mathbb{R}^d \supset \{y_1, \dots, y_n\} \in \underset{\substack{\sum_i y_i = 0 \\ \sum_i y_i y_i^\top = I_{n \times n}}}{\arg\min} \sum_{i=1}^{n} \left\| y_i - \sum_{j \in \mathcal{N}_i} w^*_{i,j} y_j \right\|_2^2$$

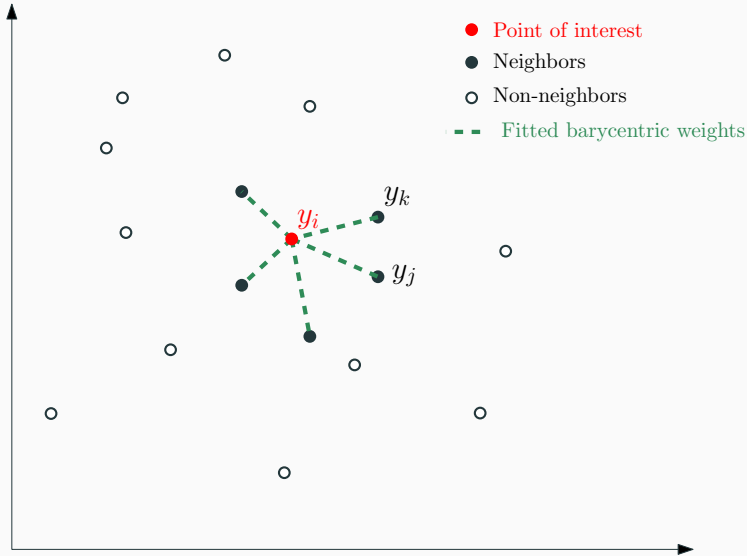
Visualizing LLE



Visualizing LLE



Visualizing LLE



A Few Remarks

When computing weights:

$$W^* \in \underset{\substack{w_{i,j}\geq 0 \\ \sum_j w_{i,j}=1}}{\arg\min} \sum_{i=1}^{n} \left\| x_i - \sum_{j\in\mathcal{N}_i} w_{i,j} x_j \right\|_2^2$$

- The loss and constraint are convex, with explicit optimum W^*.

When embedding:

$$\{y_1, \ldots, y_n\} \in \underset{\substack{\sum_i y_i = 0 \\ \sum_i y_i y_i^\top = I_{n\times n}}}{\arg\min} \sum_{i=1}^{n} \left\| y_i - \sum_{j\in\mathcal{N}_i} w^*_{i,j} y_j \right\|_2^2$$

- The embedding is only defined up to arbitrary affine maps.
- Constraints $\sum_i y_i = 0$ and $\sum_i y_i y_i^\top = I_{n\times n}$ for well-posedness.
- Explicit solution given by eigenvectors of $(I - W)^\top (I - W)$.

LLE Strengths & Weaknesses

As Isomap:

- Graph-base, spectral (eigenvector) method
- Polynomial time algorithm
- No local optima
- Non-iterative
- Single heuristic parameter (neighborhood size k)
- Can work with distances only

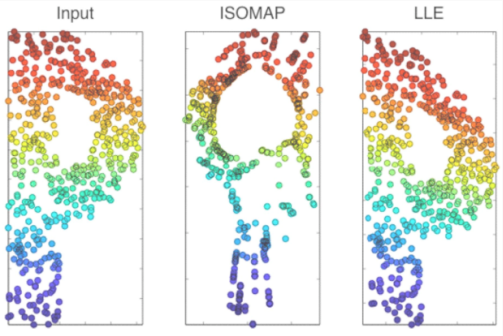
Additional weaknesses:

- Intrinsic dimension is an actual parameter
- Likely to tend to be a linear projector for large n (Goldberg and Ritov 2012; Wu and Hu 2006)

LLE Strengths & Weaknesses

Additional strengths:

- Better at handling non-convex parametrization domains



- Fast computations, taking advantage of the sparsity of

$$(I_{n\times n} - W)^\top (I_{n\times n} - W).$$

Related methods in MDS

LLE is reminiscent of MDS methods based on patches.

- FastMDS (Yang et al. 2006)
- split-and-combine MDS (Tzeng, Lu, and Li 2008)

Convergence of LLE

Not much positive theoretical results known for LLE, because its performance depends crucially on how W^* (not unique) is chosen.

Goldberg and Ritov 2012 prove the following convergence of Low-Dimensional neighborhood Representation (LDR-LLE).

(= LLE with neighborhoods $\mathcal{N}_i = B(x_i, r)$ and W^* minimizing $\|W\|_{\mathrm{F}}^2$)

Theorem (Goldberg and Ritov 2012, Theorem 3)

Assume that $M = \mathrm{dec}(\Omega)$ is isometric to a convex and that $\mathcal{X}$ is a iid uniform n-sample $x_i = \mathrm{dec}(z_i)$ from M. If $nr^d \to \infty$, and let ρ be such that $\rho/r \to 0$, then,

$$\frac{1}{n} \sum_{i \atop \mathrm{dist}(z_i, \partial D) \geq 2r+\rho} \max_{j \atop \|z_i - z_j\|_2 < \rho} \|y_i - y_j\|_2^2 = O_P(\rho/r).$$

References

Ailon, Nir and Bernard Chazelle (2006). **"Approximate nearest neighbors and the fast johnson-lindenstrauss transform".** In: *Proceedings of the thirty-eighth annual acm symposium on theory of computing*, pp. 557–563.

Arias-Castro, Ery, Adel Javanmard, and Bruno Pelletier (2020). **"Perturbation bounds for procrustes, classical scaling, and trilateration, with applications to manifold learning".** In: *Journal of machine learning research* 21, pp. 1–37.

Arias-Castro, Ery and Thibaut Le Gouic (2019). **"Unconstrained and curvature-constrained shortest-path distances and their approximation".** In: *Discrete & computational geometry* 62.1, pp. 1–28.

Bernstein, M., V. De Silva, J.C. Langford, and J.B. Tenenbaum (2000). **Graph approximations to geodesics on embedded manifolds.** Tech. rep. Department of Psychology, Stanford University.

Bourgain, Jean (1985). **"On lipschitz embedding of finite metric spaces in hilbert space".** In: *Israel journal of mathematics* 52.1, pp. 46–52.

Eftekhari, Armin and Michael B Wakin (2015). **"New analysis of manifold embeddings and signal recovery from compressive measurements".** In: *Applied and computational harmonic analysis* 39.1, pp. 67–109.

Garivier, Aurélien and Emmanuel Pilliat (2024). **"On sparsity and sub-gaussianity in the johnson-lindenstrauss lemma".** In: *Arxiv preprint arxiv:2409.06275*.

Goldberg, Yair and Ya'acov Ritov (2012). **"Theoretical analysis of lle based on its weighting step".** In: *Journal of computational and graphical statistics* 21.2, pp. 380–393.

Hastie, Trevor and Werner Stuetzle (1989). **"Principal curves".** In: *Journal of the american statistical association* 84.406, pp. 502–516.

Iwen, Mark, Arman Tavakoli, and Benjamin Schmidt (2021). **"Lower bounds on the low-distortion embedding dimension of submanifolds of $\mathbb{R}^n$".** In: *Arxiv preprint arxiv:2105.13512*.

Iwen, Mark A, Benjamin Schmidt, and Arman Tavakoli (2021). **"On fast johnson-lindernstrauss embeddings of compact submanifolds of rn with boundary".** In: *Arxiv preprint arxiv:2110.04193* 2.

Johnson, William B and Joram Lindenstrauss (1984). **"Extensions of lipschitz mappings into a hilbert space".** In: *Contemp. math.* 26, pp. 189–206.

Kane, Daniel M and Jelani Nelson (2014). **"Sparser johnson-lindenstrauss transforms".** In: *Journal of the acm (jacm)* 61.1, pp. 1–23.

Kasiviswanathan, Shiva Prasad, Mark Rudelson, Adam Smith, and Jonathan Ullman (2010). **"The price of privately releasing contingency tables and the spectra of random matrices with correlated rows".** In: *Proceedings of the forty-second acm symposium on theory of computing*, pp. 775–784.

Kruskal, Joseph B and Judith B Seery (1980). **"Designing network diagrams".** In: *Conference on social graphics*, pp. 22–50.

Lee, John A and Michel Verleysen (2007). **Nonlinear dimensionality reduction.** Vol. 1. Springer.

Matoušek, Jiří (2013). **"Lecture notes on metric embeddings".** Available from https://kam.mff.cuni.cz/~matousek/.

Pearson, Karl (1901). **"Liii. on lines and planes of closest fit to systems of points in space".** In: *The london, edinburgh, and dublin philosophical magazine and journal of science* 2.11, pp. 559–572.

Roweis, S. and L. Saul (2000). **"Nonlinear dimensionality reduction by locally linear embedding".** In: *Science* 290.5500, pp. 2323–2326.

Shalev-Shwartz, Shai and Shai Ben-David (2014). **Understanding machine learning: from theory to algorithms.** Cambridge university press.

Tenenbaum, J. B., V. de Silva, and J. C. Langford (2000). **"A global geometric framework for nonlinear dimensionality reduction".** In: *Science* 290.5500, pp. 2319–2323.

Tenenbaum, Joshua (1997). **"Mapping a manifold of perceptual observations".** In: *Advances in neural information processing systems* 10.

Tzeng, Jengnan, Henry Horng-Shing Lu, and Wen-Hsiung Li (2008). **"Multidimensional scaling for large genomic data sets".** In: *Bmc bioinformatics* 9.1, pp. 1–17.

Van Der Maaten, Laurens, Eric Postma, Jaap Van den Herik, et al. (2009). **"Dimensionality reduction: a comparative".** In: *J mach learn res* 10.66-71, p. 13.

Wu, FC and ZY Hu (2006). **"The lle and a linear mapping".** In: *Pattern recognition* 39.9, pp. 1799–1804.

Yang, Tynia, Jinze Liu, Leonard McMillan, and Wei Wang (2006). **"A fast approximation to multidimensional scaling".** In: *Ieee workshop on computation intensive methods for computer vision*.