NONLINEAR DIMENSIONALITY REDUCTION

The goal of dimensionality reduction is to map a high-dimensional dataset to a low-dimensional one in such a way that ——— and/or ——— geometric and topological properties are preserved.

The most common method for DR, PCA, is ———.

However, many high-dimensional datasets have ——— structure that is not captured by this method.
Isomap

Isometric feature mapping is the application of MDS to dissimilarities derived from shortest path lengths based on a local proximity graph such as a k-nearest neighbor graph.

This path length is viewed as an approximation to distance on the underlying data.

**Example 1** Swiss roll data

---

**Fig. 3.** The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. **A** For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). **B** The neighborhood graph $G$ constructed in step one of Isomap (with $K = 7$ and $N = 1000$ data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in $G$. **C** The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).
Laplacian eigenmaps

Given: \( x_1, \ldots, x_n \)

- desired embedding dimension \( p \)
- compute similarity graph
- form graph Laplacian
- compute eigenvectors \( u_2, \ldots, u_{p+1} \)
- set \( y_i = (u_{2i}, \ldots, u_{(p+1)i}) \in \mathbb{R}^p \)

Example

Randomly positioned horizontal or vertical bar

Unlike Isomap, Laplacian eigenmaps do not preserve global geodesic distances, just local neighborhood relationships.

Example | Swiss roll

![Swiss roll](image)

N = 10  t = 25.0

LLE capitalizes on the intuition that a data manifold that is globally nonlinear will appear linear locally.

Let high-dimensional data \( x_1, \ldots, x_n \in \mathbb{R}^8 \) be given. Let \( y_1, \ldots, y_n \in \mathbb{R}^p, \ p < 8, \) be the desired embedding. LLE has 3 steps.
1. For each $x_i$, define a local neighborhood $N_i$, e.g., the $k$-nearest neighbors

2. Solve

$$\min \{w_{ij}\}$$

s.t.

3. Now fix $\{w_{ij}\}$ and solve

$$\min \{g_{ij}\}$$

Computationally,

Step 2 $\Rightarrow$

Step 3 $\Rightarrow$
Even though LLE does not explicitly model global geodesic distances, it can still (hopefully) reconstruct the entire manifold by "patching together" local linear pieces.

**Isomap vs. LLE**

* ISOMAP: emphasizes global distance preservation $\Rightarrow$ can distort local geometry

* LLE: emphasizes local neighborhood preservation $\Rightarrow$ can lose global information, e.g. far away points mapped to nearby points.

Kernel principal component analysis (KPCA) extends PCA in the same way that SVMs extend soft-margin hyperplane classifiers:

1. Map data \( x_1, \ldots, x_n \) into a high-dimensional feature space \( \mathcal{H} \):

\[
\mathbf{x} = \begin{bmatrix}
    x^{(1)} \\
    \vdots \\
    x^{(d)}
\end{bmatrix} \longrightarrow \mathbf{I}(\mathbf{x}) = \begin{bmatrix}
    \phi^{(1)}(\mathbf{x}) \\
    \vdots \\
    \phi^{(d)}(\mathbf{x})
\end{bmatrix}
\]

where \( \phi^{(j)} \) are ________

and \( d \ll \)

2. Apply PCA to \( \mathbf{I}(x_1), \ldots, \mathbf{I}(x_n) \)
The principal components in $\mathcal{H}$ are linear in $\mathcal{V}(x_1), \ldots, \mathcal{V}(x_n)$, but nonlinear in $x_1, \ldots, x_n$. To make the procedure computationally tractable, we will rely on ____________.

Recall that a function $K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is an inner product kernel iff there exists an inner product space $\mathcal{H}$ and a mapping $\Phi: \mathbb{R}^d \rightarrow \mathcal{H}$ such that

$$K(u,v) = \langle \Phi(u), \Phi(v) \rangle$$

for all $u, v \in \mathbb{R}^d$. 
Derivation

We need to show that all operations can be represented in terms of inner products.

For simplicity, assume $\frac{1}{n} \sum_{i=1}^{n} I(x_i) = 0$ (This will not be the case typically, but the general case can be reduced to this one).

The sample covariance matrix is

$\mathbf{C} =$

We need to find all $\lambda > 0$ and $\mathbf{w} \in \mathcal{H}$, $\mathbf{w} \neq \mathbf{0}$, such that

In fact, we only care about $\lambda > 0$. 
Suppose \( v \) is an eigenvector of \( C \) with eigenvalue \( \lambda \). Consider the following:

1) For each \( i = 1, \ldots, n \)

\[ \lambda \langle \varpi(x_i), v \rangle = \langle \varpi(x_i), Cv \rangle \]

2) If \( \lambda > 0 \), then

\[ u = \frac{1}{\lambda} Cv \]

\[ = \frac{1}{\lambda} \sum_{j=1}^{n} \varpi(x_j) \cdot \varpi(x_j)^T v \]

\[ \in \text{span} \{ \varpi(x_1), \ldots, \varpi(x_n) \} \]

which means there exist \( \alpha_1, \ldots, \alpha_n \in \mathbb{R} \) such that

\[ u = \]

\( 0 \)
Combining 1) and 2) we conclude: If \( v \in \mathcal{H} \) is an eigenvector of \( \mathbf{C} \) with eigenvalue \( \lambda > 0 \), then there exists \( \alpha = (\alpha_1, \ldots, \alpha_n)^T \in \mathbb{R}^n \) such that

\[
v = \sum_{i=1}^{n} \alpha_i \mathbf{V}(x_i)\]

and for each \( l = 1, \ldots, n \)

\[
\lambda \left( \mathbf{V}(x_0), \sum_{i=1}^{n} \alpha_i \mathbf{V}(x_i) \right) = \frac{1}{n} \left( \mathbf{V}(x_0), \sum_{j=1}^{n} \mathbf{V}(x_j) \mathbf{V}(x_j)^T \sum_{i=1}^{n} \alpha_i \mathbf{V}(x_i) \right)
\]

where

\[
K :=
\]
**Claim** Let $\lambda > 0$. Then
\[ n\lambda K\alpha = K^2 \alpha \iff n\lambda \alpha = K\alpha \text{ or } K\alpha = 0 \]

**Proof** ($\iff$) obvious ($\implies$) Assume $n\lambda K\alpha = K^2 \alpha$.

We need to show $\alpha$ is an e-vec of $K$ with e-val $\lambda$ or $0$.

Since $K$ is symmetric & PSD, we can write $K = UDU^T$

where $U^TU = UU^T = I$, and $D$ is diagonal with elements $\geq 0$.

\[ n\lambda K\alpha = K^2 \alpha \iff n\lambda UDU^T\alpha = UD^2U^T\alpha \]

Set $w = U^T\alpha$. (change of coordinates), and left multiply by $U^T$, to yield $n\lambda Dw = D^2w$

\[
\begin{bmatrix}
d_1 \\
d_2 \\
\vdots \\
d_m \\
0
\end{bmatrix}
\begin{bmatrix}
\lambda d_1^2 \\
\lambda d_2^2 \\
\vdots \\
\lambda d_m^2 \\
0
\end{bmatrix}
= 
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_m
\end{bmatrix}
\]

Assuming (for simplicity) $d_1 > d_2 > \cdots > d_m > 0$, we have either

1. $w_1 = w_2 = \cdots = w_m = 0 \implies \alpha = Uw \in 0$-eigenspace of $K$
2. $w = e_j$ (jth standard basis vector) for some $j$, $1 \leq j \leq m$, in which case $n\lambda = d_j$ and $\alpha = Uw \in n\lambda$-eigenspace of $K$

If some eigenvalues have multiplicity $> 1$, the above conclusions can be easily seen to hold (if you understand eigenspaces).
If \( K\alpha = 0 \), and \( \nu = \sum_{i=1}^{n} \alpha_i \Phi(x_i) \), then

\[
K\alpha = \begin{bmatrix}
\langle \Phi(x_1), \nu \rangle \\
\vdots \\
\langle \Phi(x_n), \nu \rangle
\end{bmatrix} = 0
\]

\( \Rightarrow \nu = 0 \).

Thus, we only need to find solutions of

\[ n\lambda\alpha = K\alpha \]

with \( \lambda > 0 \). That is, if \( n_1 \geq \cdots \geq n_m \) are the nonzero eigenvalues of \( K \), with corresponding eigenvectors \( \alpha_1, \ldots, \alpha_m \), then \( \lambda_1 \geq \cdots \geq \lambda_m \) are the nonzero eigenvalues of \( C = \frac{1}{n} \sum \Phi(x_i)\Phi(x_i)^\top \), with corresponding

\[
\nu_i = \sum_{i} \alpha_{ij} \Phi(x_i).
\]

To ensure \( ||\nu_i|| = 1 \) we normalize \( \alpha_j \) so that

\( \sum \alpha_{ij} = 1 \).
If \( x \in \mathbb{R}^d \) is a test point, then the \( j \)th component of \( x \) is

\[
\langle \mathcal{F}(x), v_j \rangle = \frac{1}{n} \sum_{i=1}^{n} \mathcal{F}(x_i)
\]

Recall we assumed \( \sum_{i=1}^{n} \mathcal{F}(x_i) = 0 \). To eliminate this assumption, just apply the above steps to

This leads to needing to diagonalize \( \tilde{K} \) where

\[
\tilde{K}_{ij} = (K_{ij} - \frac{1}{n} K - K \mathbb{1}_n + \frac{1}{n} K \mathbb{1}_n)_{ij}
\]

and where \( \mathbb{1}_n \) is \( n \times n \) with \((\mathbb{1}_n)_{ij} = \frac{1}{n}\) for all \( i, j \).
Input: $x_1, \ldots, x_n \in \mathbb{R}^d$, kernel $k$, dimension $p \leq m$

- Form $\tilde{K}$ and compute $\hat{K} = U \cdot D \cdot U^T$ where
  
  $U = [u_1 \ldots u_n]$, \quad $D = \text{diag}(d_1, \ldots, d_m, 0, \ldots, 0)$

- Set $\alpha_j = \frac{1}{\sqrt{d_j}} u_j$, \quad $1 \leq j \leq p$

Output: mapping $x \mapsto y = (y^{(1)}, \ldots, y^{(p)})^T \in \mathbb{R}^p$ where

$$y^{(j)} = \sum_{i=1}^{n} \alpha_{ji} k(x, x_i)$$

Remarks:

1. Unlike PCA, KPCA can return potentially more than $d$ principal components.

2. Unlike MDS, Isomap, Laplacian eigenmaps, and LLE, KPCA defines a mapping that applies to arbitrary test points, not just the training data.

3. MDS and LLE can be related to KPCA.

4. See Smola and Schölkopf, Learning with Kernels, for more information.
A. global, local
B. linear, nonlinear
C. geodesic, manifold

D. \[ \min \sum_{i=1}^{n} \left\| x_i - \sum_{j \in N_i} w_{ij} x_j \right\|^2 \]
\[ \text{subject to } \sum_{j \in N_i} w_{ij} = 1, \quad w_{ij} \geq 0 \]

E. nonlinear
F. inner product kernel

G. \[ C = \frac{1}{n} \sum_{i=1}^{n} \hat{\varphi}(x_i) \hat{\varphi}(x_i)^T, \quad C \psi = \lambda \psi \]

H. \[ \psi = \sum_{i=1}^{n} \alpha_i \hat{\varphi}(x_i) \]

I. \[ \lambda(K\alpha)_k = \frac{1}{n} (K^2 \alpha)_k \iff n \lambda K \alpha = K^2 \alpha \]

\[ K = \left[ k(x_i, x_j) \right]_{i,j} \]

J. \[ \| v_j \|^2 = \sum_{i} \sum_{k} \alpha_{ij} \alpha_{jk} k(x_i, x_k) = \alpha_j^T K \alpha_j = n \lambda_j \| \alpha_j \|^2 \]

K. \[ \langle \hat{\varphi}(x), v_j \rangle = \langle \hat{\varphi}(x), \sum_{i} \alpha_{ji} \hat{\varphi}(x_i) \rangle = \sum_{i} \alpha_{ji} k(x, x_i) \]

L. \[ \hat{\varphi} (x_i) := \hat{\varphi}(x_i) - \frac{1}{n} \sum_{k} \hat{\varphi}(x_k) \]