# Dimensionality Reduction

Ziping Zhao

School of Information Science and Technology
ShanghaiTech University, Shanghai, China

CS182: Introduction to Machine Learning (Fall 2022)
http://cs182.sist.shanghaitech.edu.cn

Ch. 6 of I2ML (Secs. 6.4, 6.6, and 6.12 – 6.13 excluded)

# Outline

## Canonical Correlation

- CCA (a.k.a. canonical variates analysis) is an unsupervised problem for two sets of variables $\mathcal{X} = \{\mathbf{x}^t, \mathbf{y}^t\}_{t=1}^N$ with $\mathbf{x}^t \in \mathbb{R}^d$ and $\mathbf{y}^t \in \mathbb{R}^e$.
- Define $\mathbf{S}_{xx} = \text{Cov}(\mathbf{x}) = \text{Var}(\mathbf{x})$, $\mathbf{S}_{yy} = \text{Cov}(\mathbf{y}) = \text{Var}(\mathbf{y})$, $\mathbf{S}_{xy} = \text{Cov}(\mathbf{x}, \mathbf{y})$, and $\mathbf{S}_{yx} = \text{Cov}(\mathbf{y}, \mathbf{x}) = \mathbf{S}_{xy}^T$.
- We want to find two projections $\mathbf{w}$ and $\mathbf{v}$ s.t. when $\mathbf{x}$ is projected along $\mathbf{w}$ (i.e., $a = \mathbf{w}^T\mathbf{x}$) and $\mathbf{y}$ is projected along $\mathbf{v}$ (i.e., $b = \mathbf{v}^T\mathbf{y}$), the correlation is maximized, i.e.,

$$\underset{\mathbf{w},\mathbf{v}}{\text{maximize}} \quad \rho_{ab} = \text{Corr}(\mathbf{w}^T\mathbf{x}, \mathbf{v}^T\mathbf{y})$$

with

$$\begin{aligned}\text{Corr}(\mathbf{w}^T\mathbf{x}, \mathbf{v}^T\mathbf{y}) &= \frac{\text{Cov}(\mathbf{w}^T\mathbf{x}, \mathbf{v}^T\mathbf{y})}{\sqrt{\text{Var}(\mathbf{w}^T\mathbf{x})}\sqrt{\text{Var}(\mathbf{v}^T\mathbf{y})}} \\ &= \frac{\mathbf{w}^T\text{Cov}(\mathbf{x}, \mathbf{y})\mathbf{v}}{\sqrt{\mathbf{w}^T\text{Var}(\mathbf{x})\mathbf{w}}\sqrt{\mathbf{v}^T\text{Var}(\mathbf{y})\mathbf{v}}} = \frac{\mathbf{w}^T\mathbf{S}_{xy}\mathbf{v}}{\sqrt{\mathbf{w}^T\mathbf{S}_{xx}\mathbf{w}}\sqrt{\mathbf{v}^T\mathbf{S}_{yy}\mathbf{v}}}\end{aligned}$$

## Optimization

- The problem is equivalent to

$$\begin{aligned} \underset{\mathbf{w},\mathbf{v}}{\text{maximize}} \quad & \mathbf{w}^T \mathbf{S}_{xy} \mathbf{v} \\ \text{subject to} \quad & \mathbf{w}^T \mathbf{S}_{xx} \mathbf{w} = 1 \\ & \mathbf{v}^T \mathbf{S}_{yy} \mathbf{v} = 1 \end{aligned}$$

- The Lagrangian:

$$\mathcal{L}(\mathbf{w}, \mathbf{v}, \alpha, \beta) = -\mathbf{w}^T \mathbf{S}_{xy} \mathbf{v} + \alpha(\mathbf{w}^T \mathbf{S}_{xx} \mathbf{w} - 1) + \beta(\mathbf{v}^T \mathbf{S}_{yy} \mathbf{v} - 1)$$

- Taking the derivative of the Lagrangian w.r.t. $\mathbf{w}$ and $\mathbf{v}$, and setting it to $\mathbf{0}$, we get

$$\begin{aligned} \mathbf{S}_{xy}\mathbf{v} - 2\alpha \mathbf{S}_{xx}\mathbf{w} &= \mathbf{0} \\ \mathbf{S}_{yx}\mathbf{w} - 2\beta \mathbf{S}_{yy}\mathbf{v} &= \mathbf{0} \end{aligned} \implies \begin{aligned} \mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{w} &= 4\alpha\beta\mathbf{w} = \lambda\mathbf{w} \\ \mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{v} &= 4\alpha\beta\mathbf{v} = \lambda\mathbf{v} \end{aligned}$$

  indicating $\mathbf{w}$ is an eigenvector of $\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}$ corresponding to eigenvalue $\lambda$ and similarly $\mathbf{v}$ is an eigenvector of $\mathbf{S}_{yy}^{-1}\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$ corresponding to eigenvalue $\lambda$.

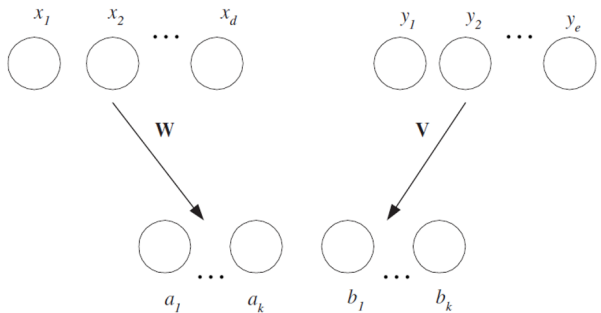
- To maximize $\rho_{ab}$, we choose the eigenvectors with the highest eigenvalue.

# Canonical Correlation Analysis

- Like PCA, we can find $k \leq \min\{d, e\}$ vectors of $\mathbf{w}_i$ and $\mathbf{v}_i$ based on the PoV measure.
- We can obtain

$$\mathbf{a} = \mathbf{W}^T(\mathbf{x} - \mathbf{m}_x), \quad \mathbf{b} = \mathbf{V}^T(\mathbf{y} - \mathbf{m}_y)$$

which constitute the new, lower-dimensional representation with values of $a_i$ uncorrelated and each $a_i$ uncorrelated with all $b_j$, $j \neq i$.



# Outline

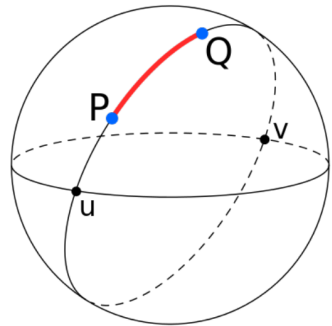
# Isometric Feature Mapping I

- PCA works when the data lies in a linear subspace.
- In many applications, the similarity between two features cannot be measured via the Euclidean distance.
- Isometric feature mapping (IsoMap) is MDS combined with a special metric, called geodesic distance , for reducing the dimensionality of data sampled from a smooth manifold.
- Instead of preserving the Euclidean distance, IsoMap preserves the geodesic distance.
- IsoMap is related to the manifold learning methods.

# Isometric Feature Mapping II

- Given a sample $\mathcal{X}$, IsoMap uses the geodesic distances between all pairs of data points.

- The geodesic distance of two data points that live in a manifold is the shortest distance along the manifold.
- On a sphere, it is just the great-circle distance.
- In practice, where we are only given a sample $\mathcal{X}$ sampled from an unknown manifold, we can approximate the true geodesic distances by the shortest-path distances.

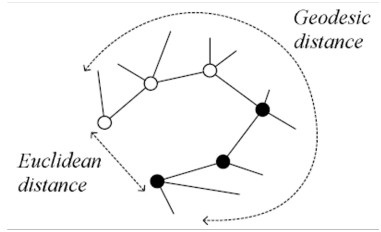


# Isometric Feature Mapping III

- For neighboring points that are close in the input space, Euclidean distance can be used (i.e., geodesic distance is locally linear)

$$d_{rs} = \|\mathbf{x}^{(r)} - \mathbf{x}^{(s)}\|_2$$

  - $\epsilon$-ball approach: for $\mathbf{x}^{(r)}$, $\mathbf{x}^{(s)}$ is close to $\mathbf{x}^{(r)}$ if $\|\mathbf{x}^{(r)} - \mathbf{x}^{(s)}\|_2 \leq \epsilon$, or
  - $k$NN approach: for $\mathbf{x}^{(r)}$, $\mathbf{x}^{(s)}$ is close to $\mathbf{x}^{(r)}$ if it is among the the $k$ nearest neighbors of $\mathbf{x}^{(r)}$.




- For faraway points, geodesic distance is approximated by the sum of the distances between the points along the way over the manifold (shortest-path distance), say, via Dijkstra's algorithm.
- Points that are far apart in the manifold are also far apart in the new $k$-dim. space after MDS even if they are close in terms of Euclidean distance in the original $d$-dim. space.

# Outline

## Kernel Methods for Dimensionality Reduction

- The kernel trick:
  - Choose a kernel $k(\cdot, \cdot)$.
  - Take any algorithm which can be computed purely using dot products $\mathbf{x}^{t^T}\mathbf{x}^{t'}$.
  - Replace each instance of $\mathbf{x}^{t^T}\mathbf{x}^{t'}$ with $k(\mathbf{x}^t, \mathbf{x}^{t'})$.
- Since $k(\mathbf{x}^t, \mathbf{x}^{t'}) = \phi(\mathbf{x}^t)^T \phi(\mathbf{x}^{t'})$, this procedure results in carrying out the original algorithm inside of $\mathbf{z} = \phi(\mathbf{x})$ space.
  - kernel PCA
  - kernel LDA
  - ...
- The result will be non-linear in the original data space.
- Similar idea to support vector machines.