Dimension Reduction

AI534

Algorithms

Supervised dimension reduction:

- Linear discriminant analysis (LDA)

Unsupervised dimension reduction:

- Principal Component Analysis (PCA)
- Nonlinear dimension reduction (ISOMAP)

Why dimension reduction?

- High dimensionality places significant burden on storage and computation
  - E.g., documents represented by tens of thousands of words, millions of bigrams
  - E.g., Images represented by millions of pixels
- Features often contain high redundancy
- Remove noise or irrelevant features (e.g., not all words are relevant for classifying documents)
- Difficult to interpret and visualize high dimensional data

Linear methods for dimension reduction

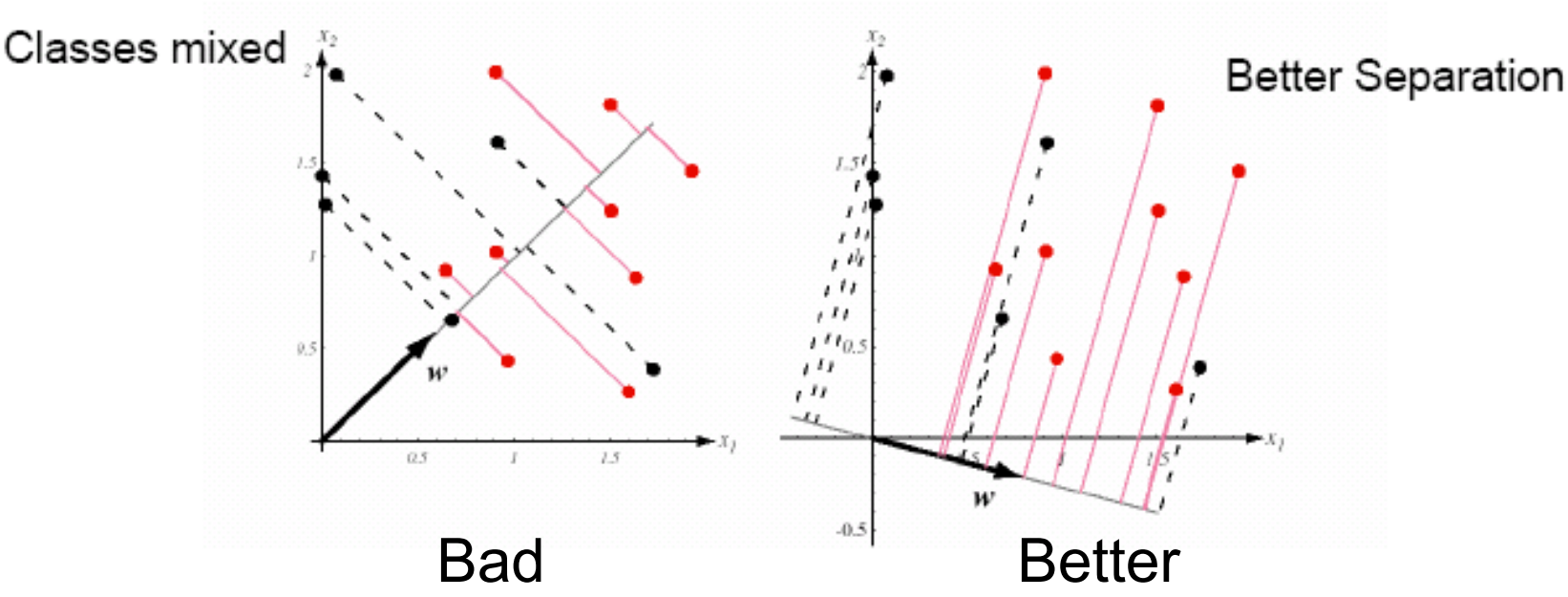
- Linearly project n-d data onto a k-d space
  - e.g., project space of 10^4 words into 3-dimensions
- There are infinitely many k-d subspaces that we can project the data into, which one should we choose
- This depends on the task at hand and what data we have available:
  - If supervised learning: maximize the separation among classes, i.e., Linear Discriminant Analysis (LDA)
  - If unsupervised, we may wish to retain as much variance as possible, i.e., Principal Component Analysis (PCA)

LDA: linear discriminant analysis

- Also named Fisher Discriminant Analysis
- It can be viewed as
  - *a dimension reduction* method
  - a generative classifier p(x|y): Gaussian with distinct $\boldsymbol{\mu}$ for each class but a shared Σ
- We will look at its **dimension reduction** interpretation and derive it that way

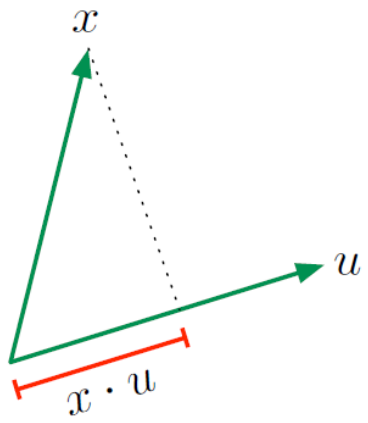
Intuition

- Find a projection direction to maximize the separation between classes



Projection

- What is the projection of $x \in R^d$ in the direction $u \in R^d$? Assume u is unit length vector (i.e., $|\mathbf{u}| = 1$)



Projection is:

$$\mathbf{x} \cdot \mathbf{u} = \mathbf{u} \cdot \mathbf{x} = \mathbf{u}^{\mathbf{T}}\mathbf{x}$$

If $\mathbf{u}$ is not unit length, then projection in the direction of $\mathbf{u}$ is:

$$\frac{\mathbf{x} \cdot \mathbf{u}}{|\mathbf{u}|} = \frac{\mathbf{u}^{\boldsymbol{T}}\mathbf{x}}{|\mathbf{u}|}$$

Objectives of LDA

- One way to measure separation is to look at the class means

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{\mathbf{x}\in c_1}\mathbf{x} \qquad \mathbf{m}_2 = \frac{1}{N_2}\sum_{\mathbf{x}\in c_2}\mathbf{x}$$

Original means

$$m'_1 = \frac{1}{N_1}\sum_{\mathbf{x}\in c_1}\mathbf{w}^T\mathbf{x} \qquad m'_2 = \frac{1}{N_2}\sum_{\mathbf{x}\in c_2}\mathbf{w}^T\mathbf{x}$$

Projected means

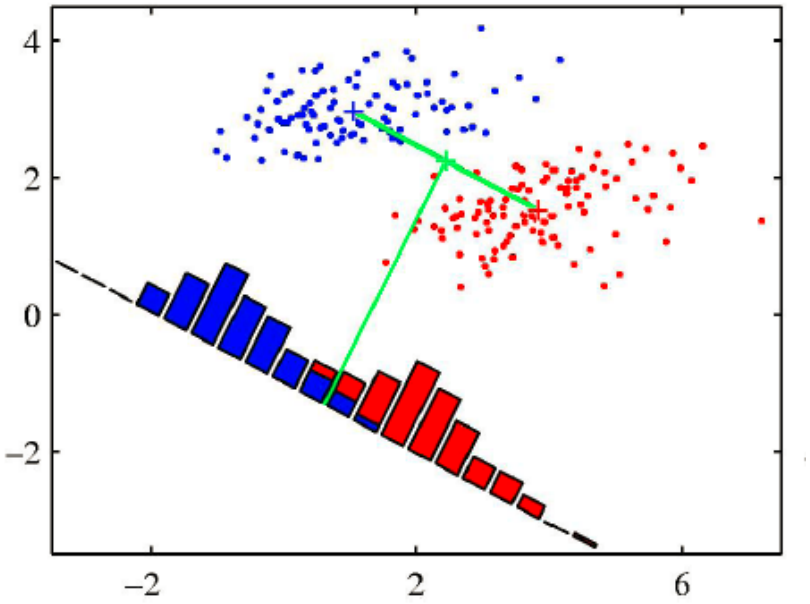
A possible goal: find the projection that maximizes

$$\left|m'_1 - m'_2\right|^2 = \left|\mathbf{w}^T\mathbf{m}_1 - \mathbf{w}^T\mathbf{m}_2\right|^2$$

subject to $|\mathbf{w}|^2 = 1$?

The maximizing solution: $\mathbf{w} \propto \mathbf{m}_1 - \mathbf{m}_2$

Maximizing mean separation is not sufficient



Objectives of LDA

- Maximizing the mean separation is insufficient
- We also want the data points from the same class to be as close as possible
- This can be measured by the within-class (or intraclass) ***scatter*** *(i.e., variance within the class)*

$$s_i^2 = \sum_{x \in c_i} (\mathbf{w}^T \mathbf{x} - m'_i)^2$$

Total within-class scatter for projected class i, where m'_i is the mean of class i after projection

$$s_1^2 + s_2^2$$

Total within-class scatter considering both classes

Combining the two sides

- There are a number of different ways to combine these two sides of the objective
- LDA seeks to optimize the following objective:

$$\underset{\mathbf{w}}{\operatorname{argmax}} \frac{|\mathbf{m}_1' - \mathbf{m}_2'|^2}{S_1^2 + S_2^2}$$

$$|\mathbf{m}_1' - \mathbf{m}_2'|^2 = |\mathbf{w}^T\mathbf{m}_1 - \mathbf{w}^T\mathbf{m}_2|^2$$

$$= \mathbf{w}^T(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T\mathbf{w} \quad \boxed{= \mathbf{w}^T\Sigma_B\mathbf{w}}$$

$$s_1^2 + s_2^2 = \mathbf{w}^T(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)\mathbf{w}$$

$$\boxed{= \mathbf{w}^T\Sigma_w\mathbf{w}}$$

$$s_1^2 = \sum_{\mathbf{x}\in C_1}(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T\mathbf{m}_1)^2 = \sum_{\mathbf{x}\in C_1}\mathbf{w}^T(\mathbf{x} - \mathbf{m}_1)(\mathbf{x} - \mathbf{m}_1)^T\mathbf{w}$$

$$= \mathbf{w}^T\left(\sum_{\mathbf{x}\in C_1}(\mathbf{x} - \mathbf{m}_1)(\mathbf{x} - \mathbf{m}_1)^T\right)\mathbf{w} = \mathbf{w}^T\Sigma_1\mathbf{w}$$

The LDA Objective

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \Sigma_B \mathbf{w}}{\mathbf{w}^T \Sigma_w \mathbf{w}}$$

$\Sigma_B = (\mathbf{m_1} - \mathbf{m2})(\mathbf{m1} - \mathbf{m2})^T$
the between-class scatter matrix

$\Sigma_w = \Sigma_1 + \Sigma_2$
the total within-class scatter matrix, where

$$\Sigma_i = \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T$$

- The above objective is known as generalized Reyleigh quotient, and it's easy to show a w that maximizes $J(w)$ must satisfy $\Sigma_B w = \lambda \Sigma_w w$

- Noticing that $\Sigma_B \mathbf{w} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{w}$ always take the direction of $\mathbf{m}_1 - \mathbf{m}_2$
- Ignoring the scalars, this leads to:

Scalar

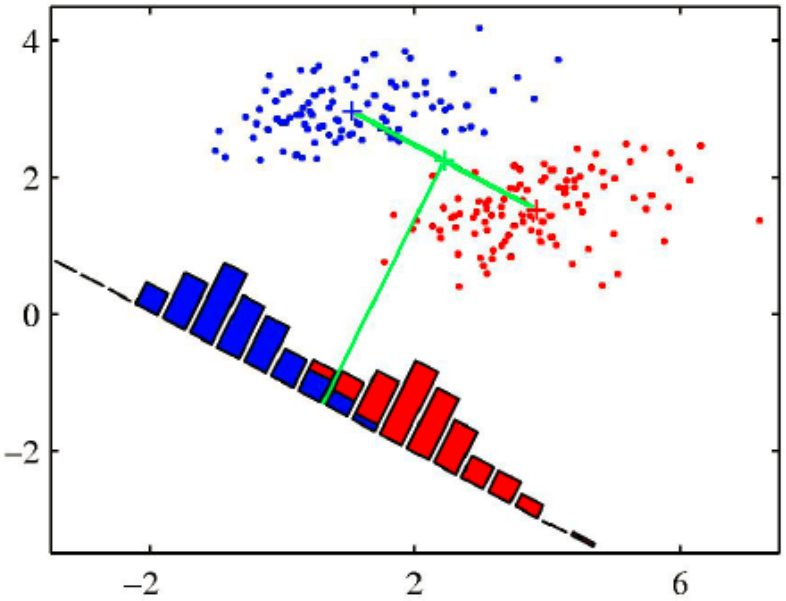
$$(\mathbf{m}_1 - \mathbf{m}_2) \propto \Sigma_w \mathbf{w}$$

$$\mathbf{w} \propto \Sigma_w^{-1} (\mathbf{m}_1 - \mathbf{m}_2)$$

LDA for two classes

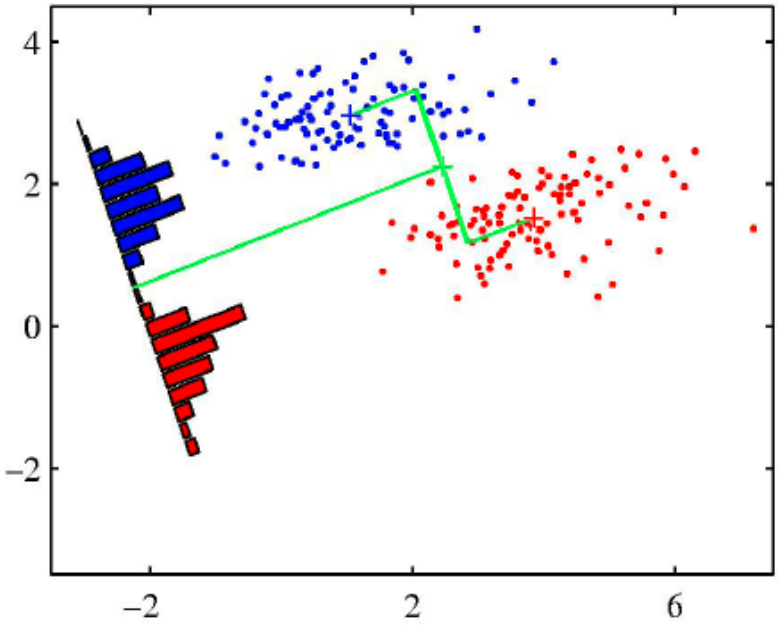
$$\mathbf{w} \propto \Sigma_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

Maximize the distance between projected mean



$\mathbf{w} \propto \mathbf{m}_1 - \mathbf{m}_2$

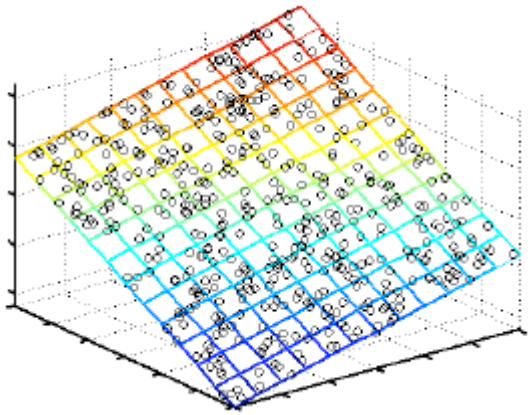
Maximize $\dfrac{\text{between scatter}}{\text{within scatter}}$



$\mathbf{w} \propto \Sigma_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$

Unsupervised Dimension Reduction

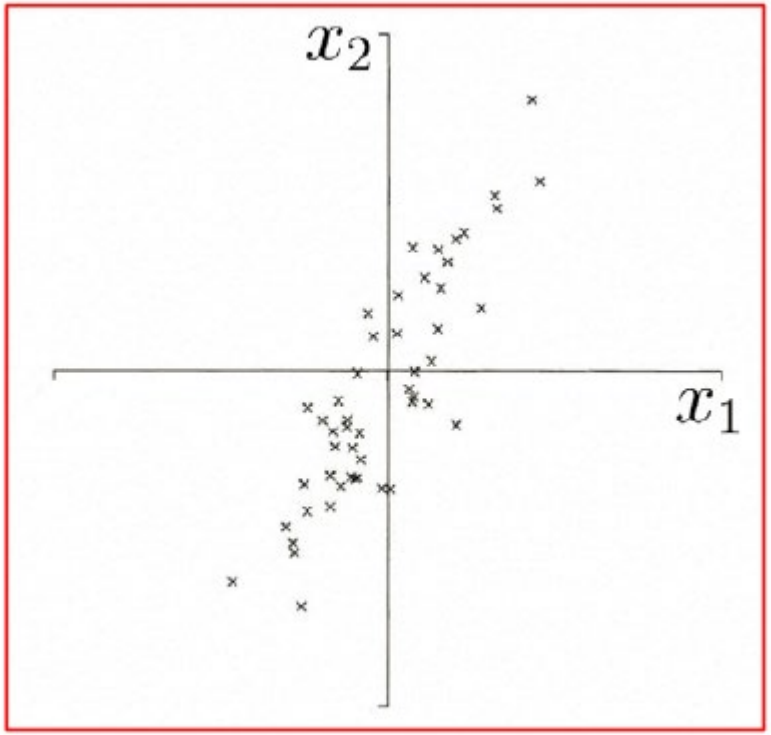
- Consider data without class labels
- Try to find a more compact representation of the data

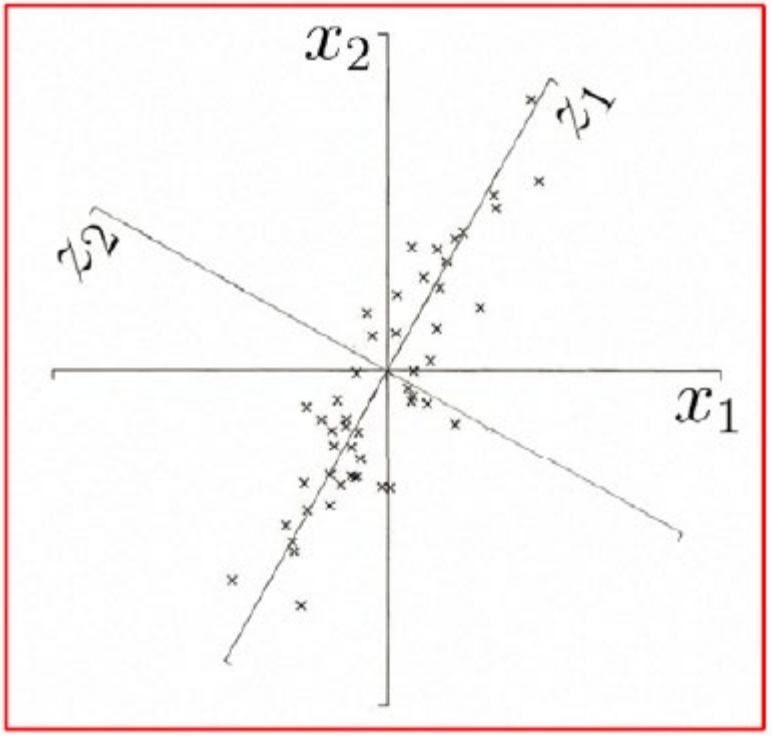


$3d \Rightarrow 2d$

- Assume that the high dimensional data actually resides in a inherent low-dimensional space
- Additional dimensions are just random noise
- Goal is to recover these inherent dimensions and discard noise dimensions

PCA: Geometric picture of principal components (PCs)





- The 1st PC is the projection direction that maximizes **the variance** of the projected data
- The 2nd PC is the projection direction that is orthogonal to the 1st PC and maximizes the variance …

PCA: variance maximization

- Given n data points: $\mathbf{x}_1, \cdots, \mathbf{x}_n$
- Consider a linear projection specified by $\mathbf{v}$
- The projection of $\mathbf{x}$ onto $\mathbf{v}$ is $z = \mathbf{v}^T\mathbf{x}$
- The variance of the projected data is $var(z) = var(\mathbf{v}^T\mathbf{x}) = \mathbf{v}^T Cov(\mathbf{x})\mathbf{v} = \mathbf{v}^T\Sigma\mathbf{v}$
- We seek to maximize the variance $\mathbf{v}^T\Sigma\mathbf{v}$ subject to the constraint $\mathbf{v}^T\mathbf{v} = 1$, where

$$\Sigma = Cov(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

Maximizing Projected Variance

- Goal: $\max \mathbf{v}^T \Sigma \mathbf{v}$, s.t. $\mathbf{v}^T \mathbf{v} = 1$
- Lagrange:

$$L = \mathbf{v}^T \Sigma \mathbf{v} - \lambda(\mathbf{v}^T \mathbf{v} - 1)$$

$$\nabla L = \Sigma \mathbf{v} - \lambda \mathbf{v} = 0 \rightarrow \Sigma \mathbf{v} = \lambda \mathbf{v}$$

- Thus the solution $\mathbf{v}$ must be an eigen-vector of Σ

- The variance of the projected data is:

$$\mathbf{v}^T \Sigma \mathbf{v} = \lambda \mathbf{v}^T \mathbf{v} = \lambda$$

- Variance captured by $\mathbf{v}$ = its eigen-values λ

Principal Component Analysis

- Let Σ be the $d \times d$ covariance matrix of $\mathbf{x}$
- Compute its eigen-decomposition:
  - Eigenvalues: $\lambda_1 \geq \lambda_2 \geq \ ... \geq \lambda_d$
  - Corresponding eigenvectors: $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_d \in R^d$ that are orthonormal (unit length and orthogonal to one another)
- To reduce to k dimensions while capturing as much variance of the data as possible, the projection would be: $\left(\mathbf{v}_1^T\mathbf{x}, \mathbf{v}_2^T\mathbf{x}, ..., \mathbf{v}_k^T\mathbf{x}\right)$
  - Or if centering ($\boldsymbol{\mu}$ is the center):

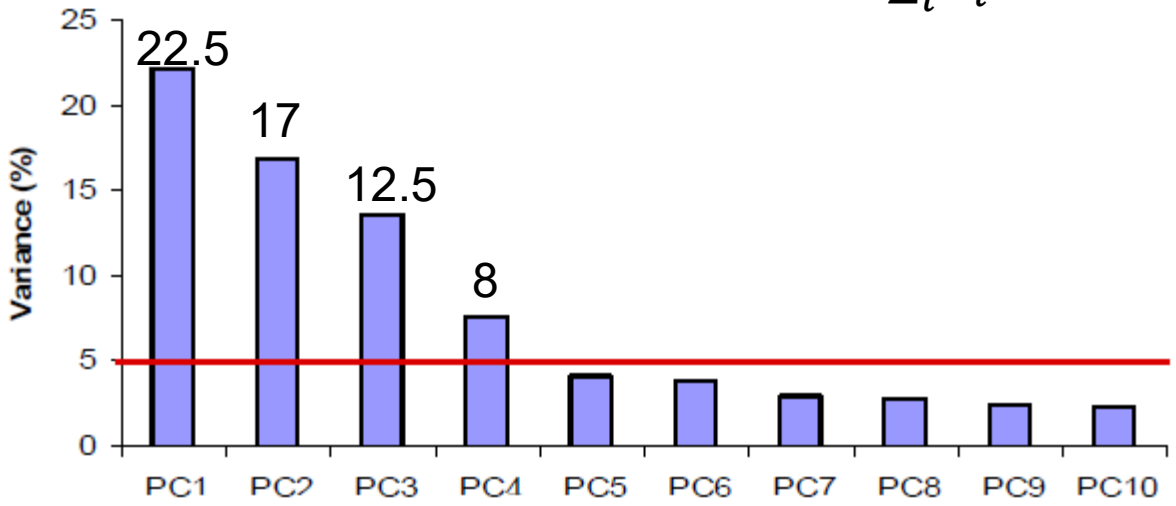
$$\left(\mathbf{v}_1^T(\mathbf{x}-\boldsymbol{\mu}), \mathbf{v}_2^T(\mathbf{x}-\boldsymbol{\mu}), ..., \mathbf{v}_k^T(\mathbf{x}-\boldsymbol{\mu})\right)$$

Retaining a fixed percentage of variance

- Often we select just enough eigen-vectors to retain a fixed percentage of the variance

Rationale: you might loose some info. but if the discarded eigen-values are small, not much is lost.

- e.g., 60%, the smallest k such that $\frac{\sum_{i=1}^{k} \lambda_i}{\sum_i \lambda_i} \geq 60\%$



- Can be tuned as a hyperparameter

Reconstruction after Projections

- Projection of $\mathbf{x}$ (with centering) into the k-dimensional space defined by $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k \in R^d$:

$$\left(\mathbf{v}_1^T(\mathbf{x}-\boldsymbol{\mu}), \mathbf{v}_2^T(\mathbf{x}-\boldsymbol{\mu}), \ldots, \mathbf{v}_k^T(\mathbf{x}-\boldsymbol{\mu})\right)$$

- The reconstructed $\mathbf{x}$

$$\hat{\mathbf{x}} = \left((\mathbf{x}-\boldsymbol{\mu})^{\mathbf{T}}\mathbf{v}_1\right)\mathbf{v}_1 + \left((\mathbf{x}-\boldsymbol{\mu})^{\mathbf{T}}\mathbf{v}_2\right)\mathbf{v}_2 + \cdots + \left((\mathbf{x}-\boldsymbol{\mu})^{\mathbf{T}}\mathbf{v}_{\mathrm{k}}\right)\mathbf{v}_{\mathrm{k}}$$

Fact: PCA projections lead to the least reconstruction error $\sum_i |(\mathbf{x}_i - \boldsymbol{\mu}) - \hat{\mathbf{x}}_i|^2$

One can show this goal leads to the same objective $\max \mathbf{v}^T \Sigma \mathbf{v}$

PCA projection example: MNIST

Original Image

Reconstructed image from its PCA projection to k dimensions

$k = 200$

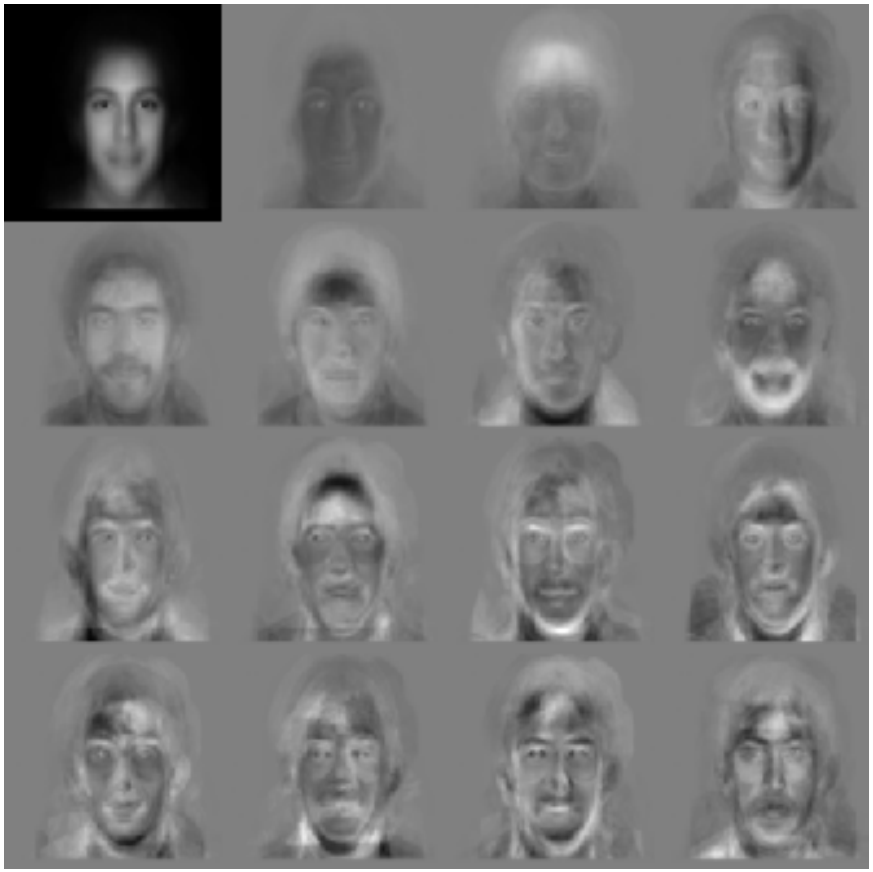
$k = 150$

$k = 100$

$k = 50$

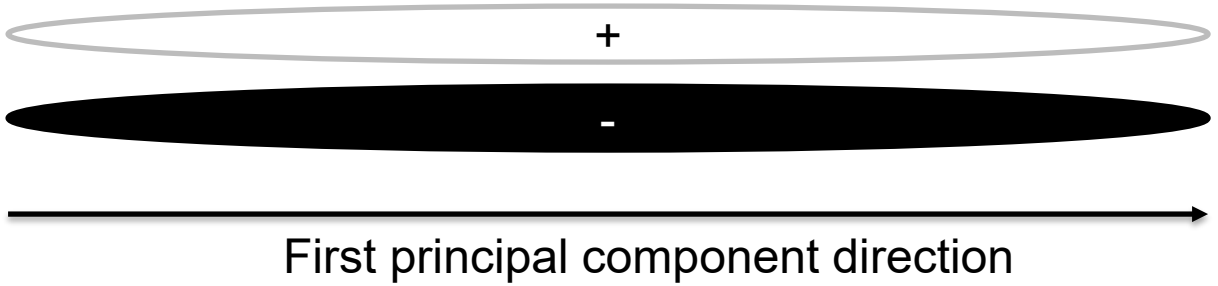
PCA for Face Images: Eigenfaces

- Database of 128 carefully-aligned faces.
- Here are the mean and the first 15 eigenvectors.
- Each eigenvector can be shown as an image
- These images are face-like, thus called eigenface



Summary of PCA: A Useful Preprocessing Step

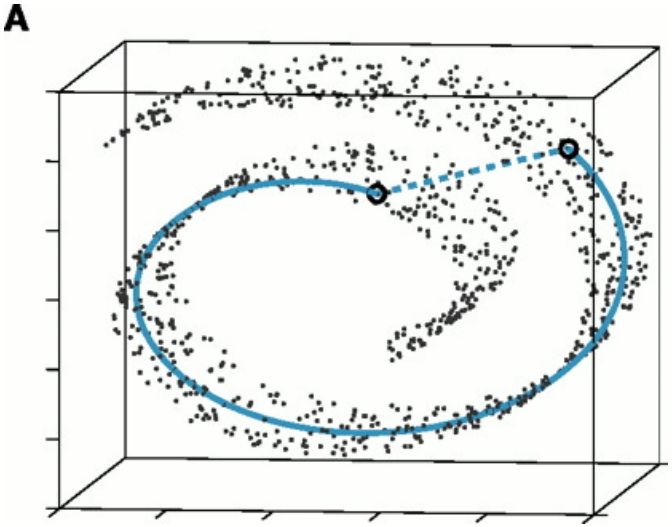
- Two interpretations:
  - Maximizing the retained data variance
  - Minimizing reconstruction error
- Helps to reduce the computational complexity
- Can help supervised learning:
  - Removes correlation between features, leads to simpler hypothesis space, less chance of over-fitting
- May lose important information when the small variance directions contain useful information:
  - E.g. for binary classification



Nonlinear Dimension Reduction

Nonlinear Methods

- Data often lies on or near a nonlinear low-dimensional curve
- We call such low dimension structure **manifolds**



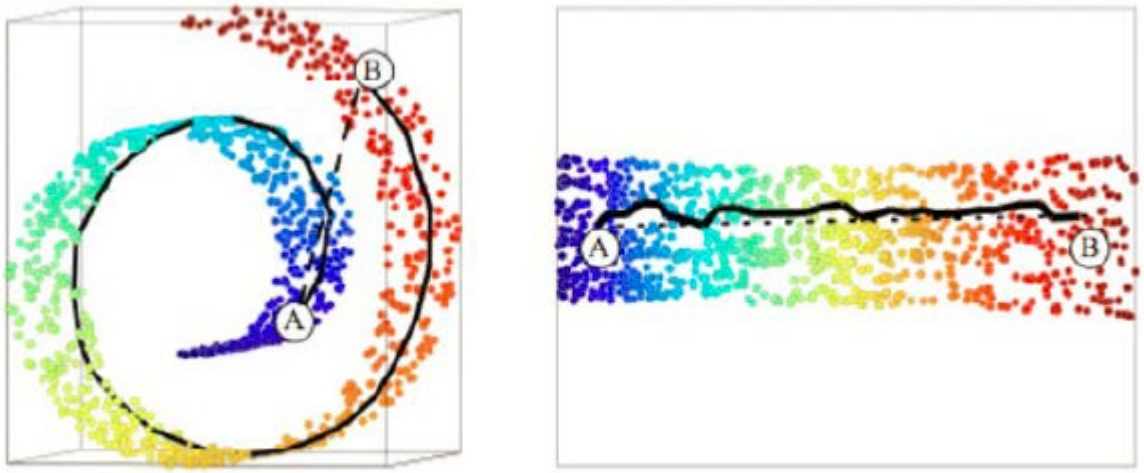
Swiss roll data

A 2-d manifold embedded in a 3-d space

An n-d manifold can be embedded in a much larger dimensional space, locally it resembles a n-d Euclidean space

ISOMAP: Isometric Feature Mapping
(Tenenbaum et al. 2000)

- A nonlinear method for dimensionality reduction
- Preserves the global, nonlinear geometry of the data by preserving the geodesic distances
- Geodesic: originally geodesic means the shortest route between two points on the surface of the manifold



Tenenbaum, Joshua B., Vin De Silva, and John C. Langford. "A global geometric framework for nonlinear dimensionality reduction." *science* 290.5500 (2000)

ISOMAP

- Two steps
  1. Approximate the geodesic distance between every pair of points in the data
     - The manifold is locally linear
     - Euclidean distance works well for points that are close enough
     - For the points that are far apart, their geodesic distance can be approximated by summing up local Euclidean distances
  2. Find a Euclidean mapping of the data that preserves the geodesic distance

Geodesic Distance

- Construct a graph by
  - Connecting i and j if
    - $d(i,j) \leq \varepsilon$ (ε-isomap) or
    - i is one of j's k nearest neighbors (k-isomap)
  - Set the edge weight equal $d(i,j)$ – Euclidean distance
- Compute the Geodesic distance between any two points as the ***shortest path distance***

Compute the Low-Dimensional Mapping

- We can use **Multi-Dimensional scaling** (MDS), a class of statistical techniques that

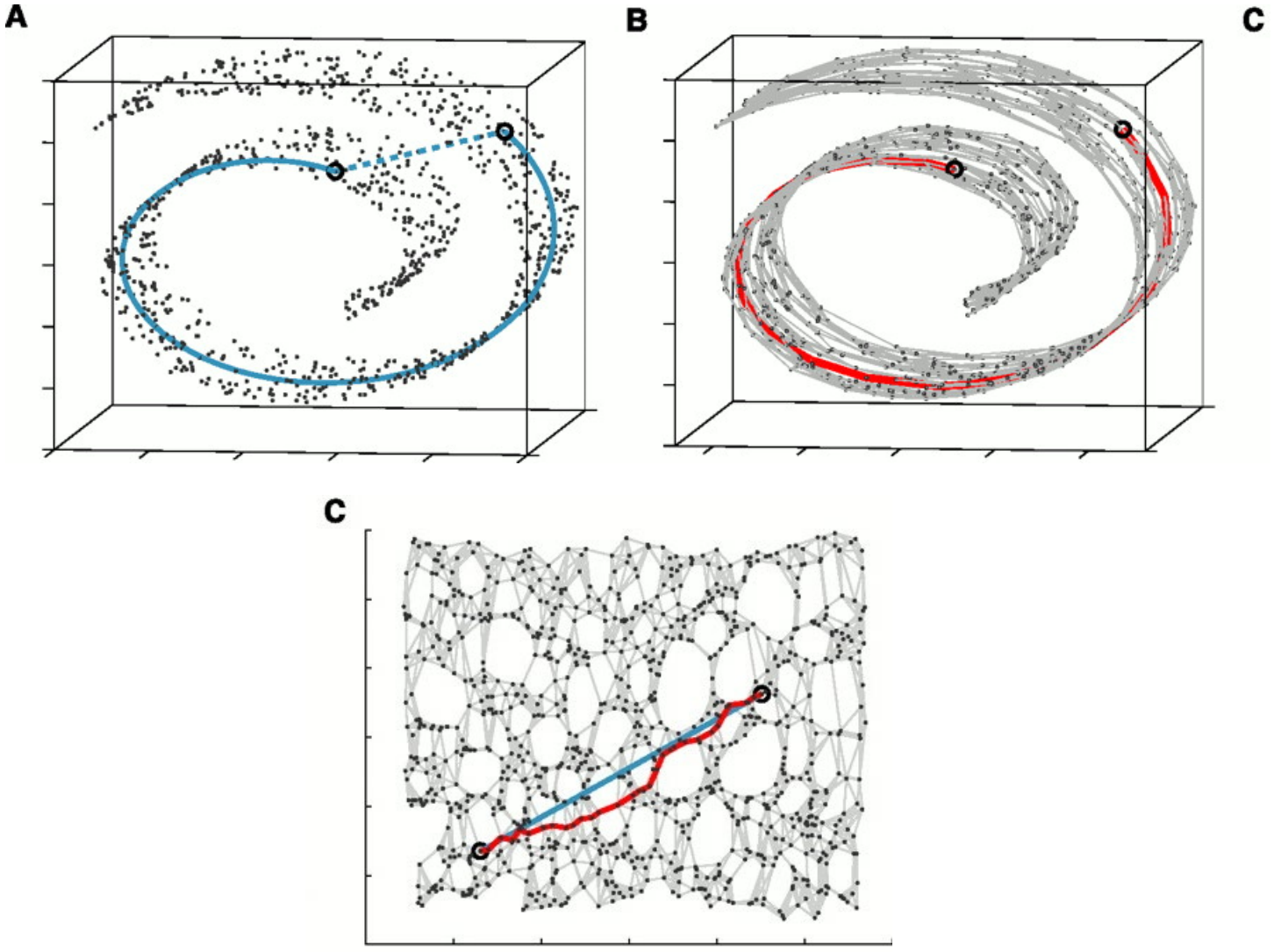
Given:

n x *n* matrix of dissimilarities between *n* objects

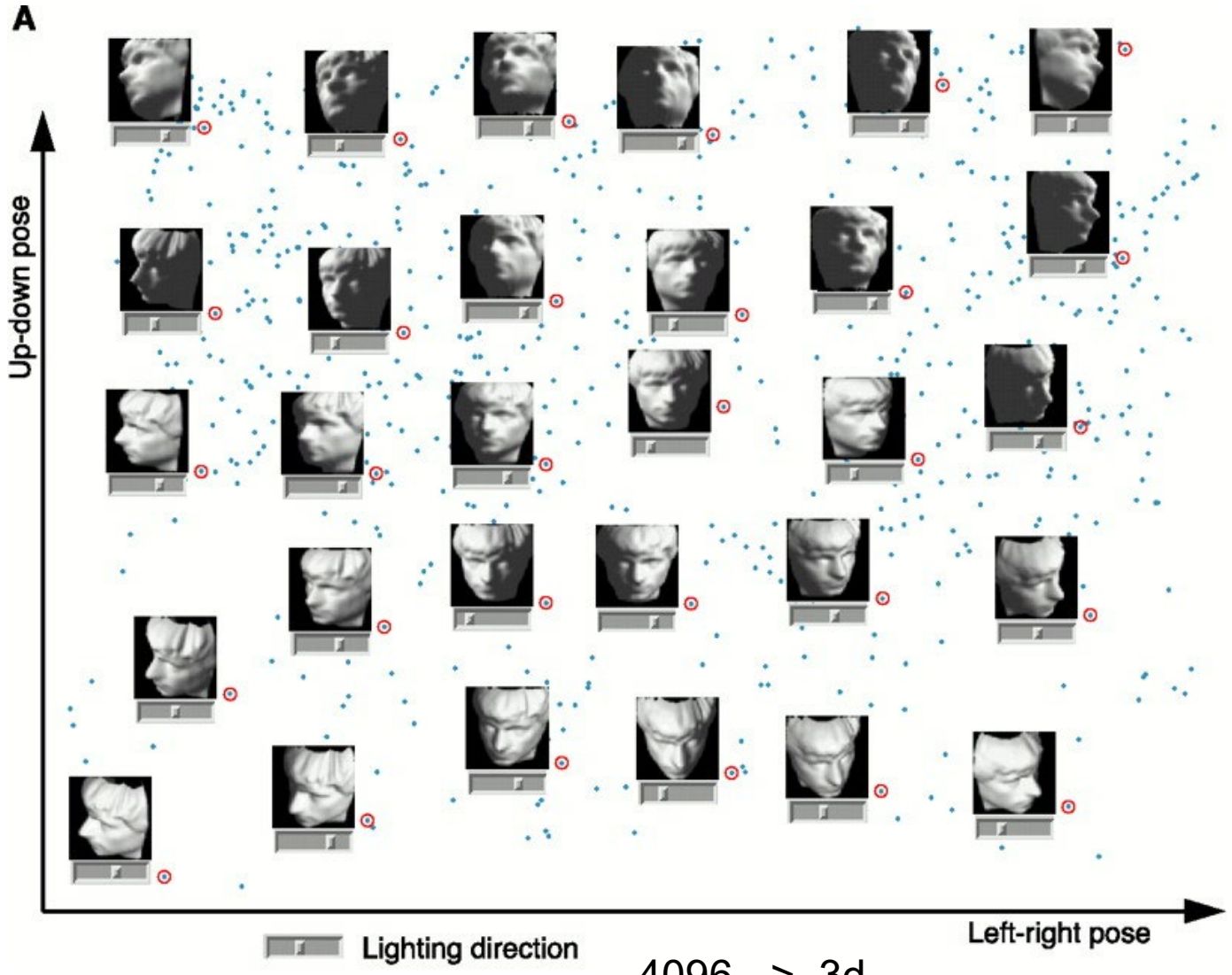
Outputs: a coordinate configuration of the data in a low-dimensional space R^d whose Euclidean distances closely match given dissimilarities.

Details of MDS is not discussed here.

ISOMAP on Swiss Roll Data

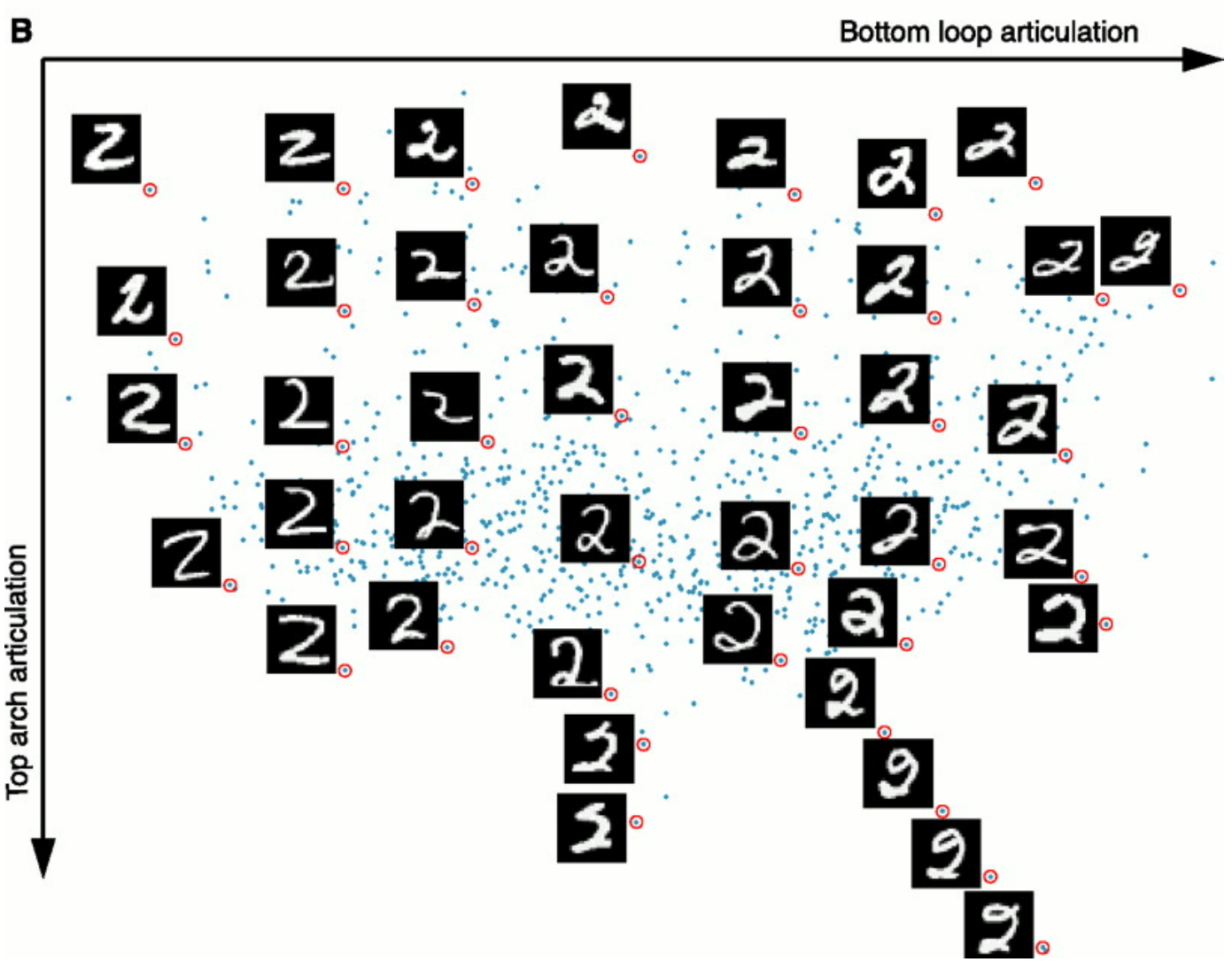


ISOMAP Examples



4096 -> 3d

ISOMAP Examples



Summary of ISOMAP

- Preserve global nonlinear structure by approximating geodesic distance
- Sensitive to the parameters used in the graph construction
  - k: for k-isomap
  - ϵ: for ϵ-isomap
- If data is overly sparse, ISOMAP may fail
  - the shortest path approximation to the geodesic distances can be poor