CSE/ISyE 6740 Computational Data Analysis

Study Notes: Dimensionality Reduction (Lecture 3-1)

Chenghao Wen
Based on Slides by Kai Wang
kwang692@gatech.edu

September 8, 2025

MATRIX AND VECTOR CONVENTION

- A data point (feature vector): $\mathbf{x}_i \in \mathbb{R}^d$, a column vector:

$$\mathbf{x}_i = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_d^{(i)} \end{bmatrix}$$

- Feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_d^{(n)} \end{bmatrix}$$

- Each row is a data point; n points in d dimensions.

WHAT IS DIMENSIONALITY REDUCTION?

- Goal: Reduce number of variables from d to $d' \ll d$.
- Mapping: $f : \mathbb{R}^d \to \mathbb{R}^{d'}$
- Methods: combine, transform, or select features.
- Can be linear (e.g., PCA) or nonlinear (e.g., Isomap).

WHY DIMENSIONALITY REDUCTION?

Applications:

- **Visualization**: Reduce to 2D/3D for plotting.
- **Feature engineering**: Combine correlated features.
- **Noise reduction / data cleaning**: Remove low-variance components.

- **Speed up learning**: Fewer features $\Rightarrow$ faster training.
- **Model simplification**: Avoid overfitting.

PRINCIPAL COMPONENT ANALYSIS (PCA)

Algorithm Steps

Given data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$:

1. Compute mean:
$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

2. Compute covariance matrix:
$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top$$

3. Find top d' eigenvectors $\mathbf{w}_1, \ldots, \mathbf{w}_{d'}$ of $\mathbf{C}$ (largest eigenvalues $\lambda_1 \geq \cdots \geq \lambda_{d'}$).
4. Project data:
$$z_i^{(k)} = \frac{\mathbf{w}_k^\top(\mathbf{x}_i - \boldsymbol{\mu})}{\sqrt{\lambda_k}}, \quad k = 1, \ldots, d'$$

   Result: $\mathbf{z}_i \in \mathbb{R}^{d'}$ is the reduced representation.

Optimization Criterion

Maximize variance along direction $\mathbf{w}$ ($\|\mathbf{w}\| = 1$):

$$\max_{\mathbf{w}:\|\mathbf{w}\|=1} \frac{1}{n}\sum_{i=1}^{n} \Big(\mathbf{w}^\top(\mathbf{x}_i - \boldsymbol{\mu})\Big)^2 = \max_{\mathbf{w}} \mathbf{w}^\top\mathbf{C}\mathbf{w}$$

Eigenvalue Problem

- The solution satisfies: $\mathbf{C}\mathbf{w} = \lambda\mathbf{w}$
- Variance along $\mathbf{w}$ is λ.
- Eigenvectors are orthonormal: $\mathbf{w}_i^\top\mathbf{w}_j = 0$,$\mathbf{w}_i^\top\mathbf{w}_i = 1$
- Principal directions = eigenvectors with largest eigenvalues.

Derivation via Lagrangian

$$\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^\top\mathbf{C}\mathbf{w} + \lambda(1 - \|\mathbf{w}\|^2)$$

Taking derivative:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = 2\mathbf{C}\mathbf{w} - 2\lambda\mathbf{w} = 0 \Rightarrow \mathbf{C}\mathbf{w} = \lambda\mathbf{w}$$

Multiple Principal Directions

- First PC: direction of max variance.
- Second PC: orthogonal, second-max variance.
- Continue for d' components.
- Result: orthogonal basis ordered by explained variance.

RECONSTRUCTION FROM PCA

Given reduced coordinates $\mathbf{z}_i$, reconstruct approximate original:

$$\hat{\mathbf{x}}_i = \boldsymbol{\mu} + \sum_{k=1}^{d'} z_k^{(i)} \sqrt{\lambda_k}\, \mathbf{w}_k$$

- Minimizes squared reconstruction error.
- Error = distance to principal subspace.

WHEN TO USE PCA?

Good for:

- Feature distribution
- Visualization (2D/3D plots)
- Feature engineering (especially $d \gg n$)
- Data compression
- Denoising

Drawbacks:

- Components may lack interpretability.
- High variance $\neq$ high predictive power.
- Unsupervised: ignores labels.

LIMITATIONS OF PCA

- Assumes linear relationships.
- Fails on nonlinear manifolds (e.g., Swiss roll).
- Uses Euclidean distance globally.

NONLINEAR DIMENSIONALITY REDUCTION

Motivation

- Example: face images with pose/lighting variations.

- Linear methods fail to "unroll" curved structures.

Geodesic vs Euclidean Distance

- **Euclidean**: straight-line ("as the crow flies").
- **Geodesic**: shortest path *along the manifold.*
- Locally, Euclidean $\approx$ geodesic.
- Globally, Euclidean can be misleading.

ISOMAP: ISOMETRIC FEATURE MAPPING

Algorithm Steps

Given $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$:

1. **Build neighborhood graph**:

$$A_{ij} = \begin{cases} 1 & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| \leq \varepsilon \text{ or } j \in \text{k-NN of } i \\ 0 & \text{otherwise} \end{cases}$$

   This adjacency matrix A defines the connections between data points within a certain threshold ϵ or among the k-nearest neighbors.

2. **Compute shortest path distances** D_{ij} (geodesic approx):
   - Floyd-Warshall: $O(n^3)$
   - Dijkstra $\times n$: $O(n(E + n \log n))$

3. **Formulate distance matrix**: Given the distance matrix D, find representation $z^i \in \mathbb{R}^{d'}$ such that:

$$D_{ij}^2 = \|z^i - z^j\|^2 = (z^i - z^j)^\top (z^i - z^j) = z^{i\top} z^i + z^{j\top} z^j - 2z^{i\top} z^j$$

   In matrix format, let $Z = (z^1, z^2, ..., z^n)^\top \in \mathbb{R}^{n \times d'}$:

$$(D)^2 = a1^\top + 1a^\top - 2ZZ^\top \in \mathbb{R}^{n \times n}.$$

   where $a = (z^{1\top} z^1, z^{2\top} z^2, ..., z^{n\top} z^n)^\top$

4. **Construct a special centering matrix** $H = I - \frac{1}{n} 11^\top$: Verify:

$$\left(I - \frac{1}{n} 11^\top\right) 1a^\top \left(I - \frac{1}{n} 11^\top\right) = 0$$

$$\left(I - \frac{1}{n} 11^\top\right) a1^\top \left(I - \frac{1}{n} 11^\top\right) = 0$$

   Then apply H to both sides of $(D)^2$:

$$C = -\frac{1}{2} H (D)^2 H = -\frac{1}{2} H (a1^\top + 1a^\top - 2ZZ^\top) H = HZZ^\top H$$

$$HZ = \left(I - \frac{1}{n}11^\top\right) Z = Z - \mu 1^\top = \tilde{Z}$$

Ultimately we get $C = \tilde{Z}\tilde{Z}^\top$

5. **Eigen-decompose** Perform eigenvalue decomposition on C:

$$Cw = \lambda w$$

Take the eigenvectors $w^1, w^2, ..., w^{d'}$ of C corresponding to the largest eigenvalues $\lambda_1, \lambda_2, ..., \lambda_{d'}$. Reduced representation:

$$\tilde{Z} = (w^1, w^2, ..., w^{d'}) \begin{pmatrix} \lambda_1^{1/2} & & \\ & \ddots & \\ & & \lambda_{d'}^{1/2} \end{pmatrix}$$

TAKEAWAYS

PCA Summary

- Linear, fast, interpretable.
- Maximizes variance.
- Solves eigenvalue problem of covariance matrix.
- Good for decorrelation, compression, visualization.

Isomap Summary

- Nonlinear, preserves geodesic distances.
- Uses graph-based shortest paths.
- Better for curved manifolds.
- Computationally heavier than PCA.

When to Use Which?

Use PCA	**Use Isomap**
Linear data	Nonlinear manifolds
Fast embedding needed	Topology preservation
Feature decorrelation	Visualization of complex data
$d \gg n$	Sufficient data for graph