TEXAS TECH UNIVERSITY SYSTEM

Group 1 Presentation

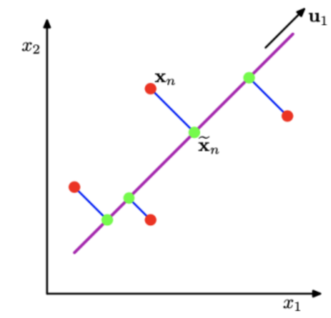
Dimensionality Reduction Techniques

Rishikesh
Joseph

Pattern Recognition
Spring 2024
Dr. Sari-Saraf

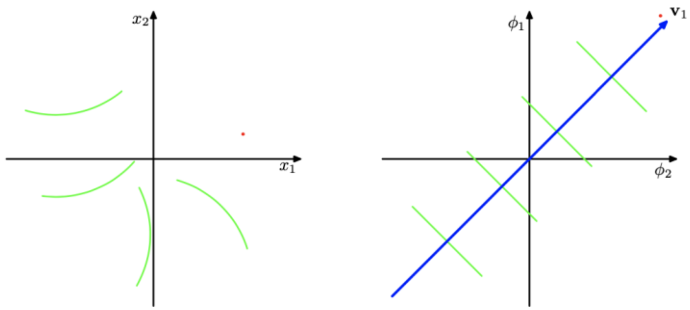
1. Overview

Dimensionality reduction is an important technique in ML when we are dealing with high-dimensional data. It aims to preserve as much relevant information possible While reducing the number of features and dimension



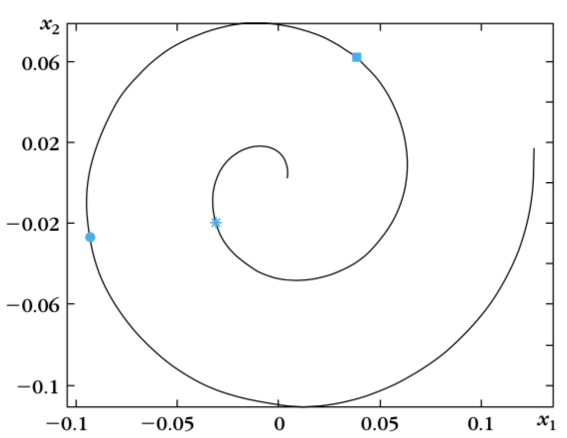
Reasons for Reducing Dimensionality:

- Curse of dimensionality
- Feature extraction
- Data visualization
- Computational efficiency



Methods:

- Principal Component Analysis (PCA)
- Kernel-PCA
- Multidimensional Scaling (MDS)
- Isometric Feature Mapping (ISOMAP)
- Etc.



2. Principal Component Analysis (PCA)

- PCA is an **orthogonal projection** of data onto a lower dimensional linear space known as the principal subspace, aiming to **maximize the variance** of the **projected data.**
  - **Eigenvalues** represent the amount of variance carried in each principal component
  - **Eigenvectors** define the directions of maximum variance
- Resulting components are **uncorrelated**, meaning that new features (principal components) provide independent information.
  - Covariance between any pair of components is zero
- Important to note that PCA might not be effective if the data exhibits **non-linear relationships.**
  - Data may lie on complex manifolds

3. How does PCA work?

1. Calculate Mean:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

2. Calculate Covariance Matrix

$$s = \begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) \\ cov(x_2, x_1) & cov(x_2, x_2) \end{bmatrix}$$

$$cov(x_1, x_1) = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ij} - \mu_j)^T (x_{ij} - \mu_j)$$

3. Find Eigenvalues

$$0 = \det(s - \lambda I)\text{ , } where\ I\ is\ the\ identity\ matrix$$

4. Find Eigenvectors

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} = (S - \lambda I)u$$

$$e_1 = \begin{bmatrix} \frac{u_1}{\| u \|} \\ \frac{u_2}{\| u \|} \end{bmatrix}$$

5. Compute the PCA component

$$e_1^T \begin{bmatrix} x_{1k} - \mu_1 \\ x_{2k} - \mu_2 \end{bmatrix}$$

$$Reduce\ l\ from\ 2\ to\ 1$$

Features	X1	X2	X3	X4
X_1	4	8	13	7
X_2	11	4	5	14

$$\mu_1 = 8, \qquad \mu_2 = 8.5$$

$$cov(x_1, x_2) = \frac{1}{3}\left((4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\right) = 14$$

$$\mathrm{S} = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}, \quad 0 = det[S - \lambda I], \quad 0 = det\begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix}$$

$$0 = \lambda^2 - 37\lambda + 201, \quad \lambda_1, \lambda_2 = 30.38, 6.62$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} (14-\lambda)u_1 - 11u_2 \\ -11u_1 + (23-\lambda)u_2 \end{bmatrix}$$

$$(14-\lambda)u_1 = 11u_2, \quad \frac{u_1}{11} = \frac{u_2}{14-\lambda} = t, \quad u_1 = 11t, \quad u_2 = (14-\lambda)t$$

$$u = \begin{bmatrix} 11 \\ 14-\lambda \end{bmatrix}, \lambda_1 = 30.38, \| u \| = 19.7348$$

$$e_1 = \begin{bmatrix} \frac{11}{19.7348} \\ \frac{14-30.38}{19.7348} \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

4. How does PCA work?

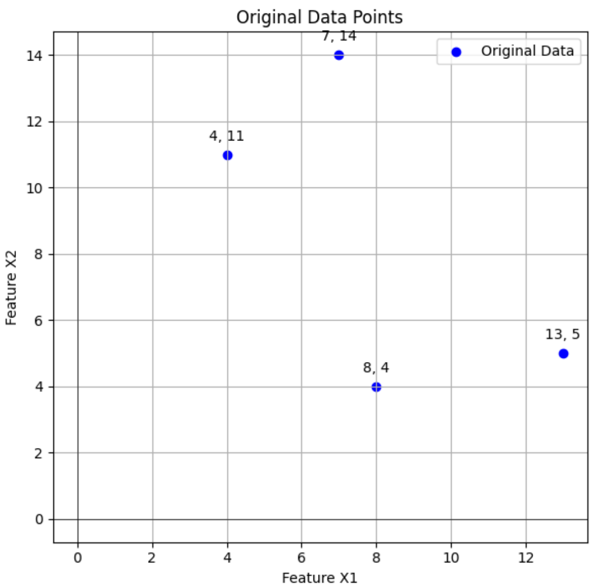
$$e_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix} \qquad e_1^T \begin{bmatrix} x_{1k} - \mu_1 \\ x_{2k} - \mu_2 \end{bmatrix}$$

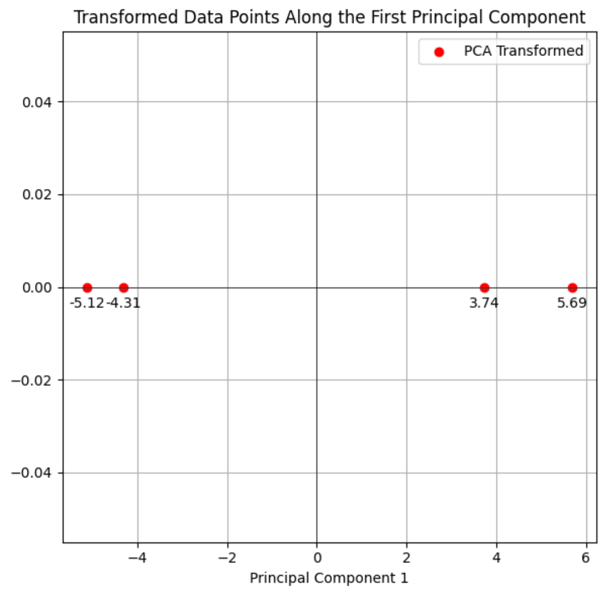
$$\begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} x_{11} - 8 \\ x_{21} - 8.5 \end{bmatrix}$$

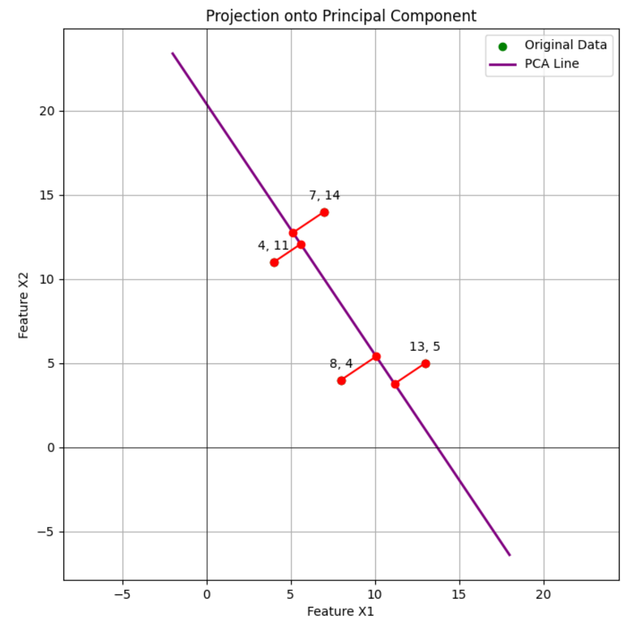
$$0.5574(4-8) - 0.8303(11-8.5) = -4.30535$$

$Reduce\ l\ from\ 2\ to\ 1$

Features	X1	X2	X3	X4
X_1	4	8	13	7
X_2	11	4	5	14
PC	-4.30535	3.7351	5.6928	-5.2238

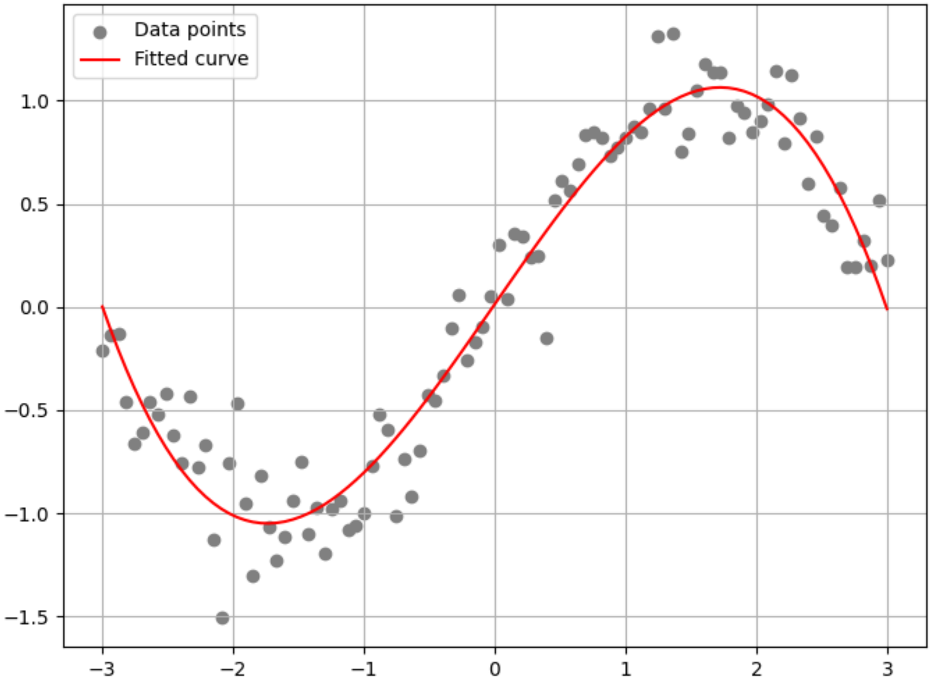






5. Kernel PCA: Extension of PCA!

- **Extension of Traditional PCA**:
  - Adapts PCA for data with nonlinear relationships.
- **Utilizes Kernel Methods**:
  - Projects data into a higher-dimensional feature space using a kernel function, allowing linear analysis techniques in this new space.
- **Enhanced Capability**:
  - Capable of uncovering complex patterns in data that are not apparent in the original space.
- **Nonlinear Dimensionality Reduction**:
  - Effectively reduces dimensions while preserving nonlinear structures in the data.



6. Kernel PCA

- Implicit Mapping to RKHS: Data x is mapped into a higher-dimensional space H without direct computation using $x \mapsto \phi(x)$. $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|2)$.

- Use feature transformations to "linearize" the data
  - Via some specific choice of basis functions $\boldsymbol{\phi}(\boldsymbol{X_n})$
  - Do PCA in this space

$$S = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{X}_n \boldsymbol{X}_n^T \qquad R = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\phi}(\boldsymbol{x}_n) \boldsymbol{\phi}(\boldsymbol{x}_n)^T$$

  - Gram Matrix K is central, computed as $K_{ij} = k(x_i, x_j)$ and used as a proxy for R.
    $K\alpha = n\lambda\alpha$
  - Normalize eigenvectors to ensure orthonormality in H.
    $\alpha^T K\alpha = n$

  - Then project the data points onto the kernel PCA components $y(k) = \sum_{i=1}^{N} a_k(i) K(x, x_i)$

7. Kernel-PCA Steps

- **Select Kernel Function**:
  - Choose a kernel based on data characteristics, commonly used kernels include polynomial, radial basis functions, and Gaussian.
- **Compute Kernel Matrix *K***:
  - Calculate the Gram matrix $K(i,j) = K(x_i,x_j)$ for all pairs i,j from 1 to n
  - Use kernel approximations for large datasets
- **Eigenvalue Decomposition**:
  - Compute the m dominant eigenvalues λ_k and eigenvectors $\boldsymbol{a}_k$ for $\mathbf{k} = 1,2,\ldots,m$ of the kernel matrix K
  - Perform the required normalization of the eigenvectors to ensure they form a proper basis in the feature space.
- **Projection**:
  - Compute the m projections of the data onto each one of the dominant eigenvectors to achieve dimensionality reduction.
- **Kernel Trick Explanation**:
  - Mapping data into a higher-dimensional space to make nonlinear features linearly separable, allow linear methods like PCA to be effective in this transformed space.

8. Kernel-PCA Steps

- $x_1 = (1, 2)$
- $x_2 = (2, 3)$
- $x_3 = (3, 5)$

$$K(x, y) = (1 + x^T y)^2$$

- $K(x_1, x_1) = (1 + [1, 2] \cdot [1, 2])^2 = (1 + 1 + 4)^2 = 36$
- $K(x_1, x_2) = (1 + [1, 2] \cdot [2, 3])^2 = (1 + 2 + 6)^2 = 81$
- $K(x_1, x_3) = (1 + [1, 2] \cdot [3, 5])^2 = (1 + 3 + 10)^2 = 196$
- $K(x_2, x_2) = (1 + [2, 3] \cdot [2, 3])^2 = (1 + 4 + 9)^2 = 196$
- $K(x_2, x_3) = (1 + [2, 3] \cdot [3, 5])^2 = (1 + 6 + 15)^2 = 484$
- $K(x_3, x_3) = (1 + [3, 5] \cdot [3, 5])^2 = (1 + 9 + 25)^2 = 1225$

kernel matrix K is:

$$K = \begin{bmatrix} 36 & 81 & 196 \\ 81 & 196 & 484 \\ 196 & 484 & 1225 \end{bmatrix}$$

To center the kernel matrix in the feature space:

$$\tilde{K} = K - 1_n K - K 1_n + 1_n K 1_n$$

where 1_n is an $n \times n$ matrix with all entries equal to $\frac{1}{n}$ (in this case, $\frac{1}{3}$).

$$\tilde{K} = \begin{bmatrix} 158.33 & 54.00 & -212.33 \\ 54.00 & 19.67 & -73.67 \\ -212.33 & -73.67 & 286.00 \end{bmatrix}$$

- **Eigenvalues:** $[462.72, 1.28, -1.52 \times 10^{-13}]$
- **Eigenvectors:**

$$\begin{bmatrix} -0.584 & -0.570 & -0.577 \\ -0.202 & 0.791 & -0.577 \\ 0.786 & -0.221 & -0.577 \end{bmatrix}$$

- **Projections:**

$$\begin{bmatrix} -270.32 & -0.73 \\ -93.43 & 1.02 \\ 363.74 & -0.28 \end{bmatrix}$$

9. Comparison Between Kernel PCA and PCA

- **Objective**:
  - Both aim to find principal components and reduce dimensionality.
- **Data Handling**:
  - PCA: Works directly on raw data, best for linearly separable data.
  - Kernel PCA: Transforms data into a higher-dimensional space, suited for nonlinear data.
- **Mathematical Approach**:
  - PCA: Involves eigendecomposition of the covariance matrix of the original data.
  - Kernel PCA: Involves eigendecomposition of the kernel matrix, representing data in an implicitly defined feature space.
- **Application**:
  - PCA: Effective for visualizing and understanding linear relationships.
  - Kernel PCA: Preferable for complex datasets where linear methods fail to reveal inherent structures.

Multidimensional Scaling (MDS)

10. How does MDS work?

- **Primary Goal**:
  - Project high-dimensional data into a lower-dimensional space while preserving the pairwise distances between data points.
- **Methodology**:
  - Involves computing a distance matrix representing the pairwise distances between all data points.
  - Perform eigendecomposition of this distance matrix to identify the principal components.
- **Relation to PCA**:
  - When distances are Euclidean, MDS yields results equivalent to PCA, making it suitable for linear reductions.
  - MDS provides a more flexible framework as it can work with any form of distance metric, enhancing its applicability to diverse datasets.

11. MDS Methodology

- **Gram Matrix Construction**:
  - Construct the matrix K, which encapsulates either the squared Euclidean distances or the inner products between data points.
- **Preserve Euclidean Distances**:
  - Optionally, MDS can be formulated specifically to preserve Euclidean distances among points, rather than just inner products.
  - Useful when maintaining the directionality of data.
- **Eigendecomposition**:
  - Perform eigendecomposition on the distance matrix to extract the principal components.
  - This involves calculating the eigenvalues and eigenvectors of K.

12. Comparison Between MDS and PCA

- **Objective Comparison**:
  - MDS: Aims to preserve the distance relationships between data points in a lower-dimensional space.
  - PCA: Focuses on maximizing variance and identifying directions of highest variance in the data.
- **Methodological Approach**:
  - MDS: Works with a distance matrix that represents the dissimilarities (distances) among data points.
  - PCA: Operates on a covariance matrix that reflects variances and correlations among features.
- **Matrix Comparisons**:
  - PCA: Utilizes the Correlation Matrix:

$$R_x = \frac{1}{n}\sum X_k X_k^T = \frac{1}{n} X_k X_k^T$$

  - MDS: Uses the Gram Matrix:

$$K = XX^T$$

  - Both matrices, R_x and K, share the same rank and eigenvalues, though their eigenvectors differ but are related.

Isometric Feature Mapping (ISOMAP)

13. Extension of MDS!

- **Overview of ISOMAP**:
  - An extension of Multidimensional Scaling (MDS) that aims to maintain the geodesic distances between points when projecting them into a lower-dimensional space.
- **Geodesic Distances**:
  - Euclidean distances fail to capture the intrinsic geometry when data resides on curved manifolds.
  - Unlike traditional MDS that considers Euclidean distances, ISOMAP uses geodesic distances measured along the data's manifold.
  - Example: For points on a circle, the geodesic distance is the arc length around the circle, not the straight-line chord distance.

14. ISOMAP Visualization

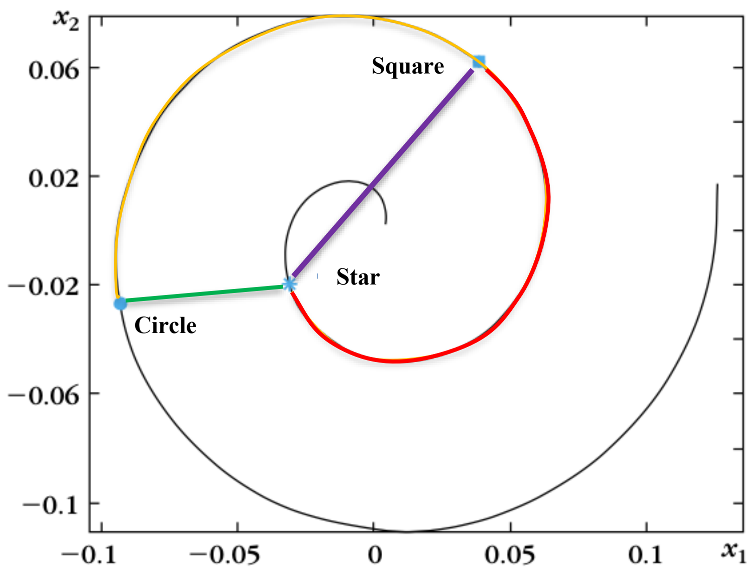


Figure 3: Spiral showcasing difference between Euclidean and Geodesic distances [2]

- Observing Euclidean distances, the points of closer distance is the circle and star shown by the green line compared to the purple line.
- However, if constrained to following the path of the spiral, the geodesic distance determines closeness.
- The star and the square are now closer in distance compared to that of the star and the circle shown by the red and orange lines, respectively.

15. ISOMAP Algorithm

- **Neighborhood Construction**:
  - Determine the nearest neighbors of each point, either by finding the K nearest neighbors or a radius-based approach.
- **Graph Construction**:
  - Construct a graph by linking each data point to its neighbors.
  - Label the edges with the Euclidean distances as weights between connected points.
- **Geodesic Distance Calculation**:
  - Compute shortest paths between all pairs of points in the neighborhood graph, typically using algorithms like Floyd-Warshall or Dijkstra's.
- **Dimensionality Reduction**:
  - Apply classical MDS to the matrix of geodesic distances to project the data into a lower-dimensional space while preserving geodesic distances.

Conclusion

16. Overview of Techniques

- Kernel PCA (K-PCA):
  - Enhances traditional PCA by using kernel methods to reveal non-linear patterns, making it ideal for complex datasets where linear relationships are insufficient.
- Multidimensional Scaling (MDS):
  - Preserves the pairwise distances between data points during dimensionality reduction, suitable for visualizing and exploring data structure in a lower-dimensional space.
- Isometric Feature Mapping (ISOMAP):
  - Extends MDS by approximating geodesic distances on a manifold, perfect for data embedded on curved surfaces, maintaining global geometries effectively.
- Together, these techniques simplify high-dimensional data analysis by retaining crucial structural information, aiding in the insightful exploration of data across various applications.

References

1. Bishop, C. M. (2009). *Pattern Recognition and Machine Learning*. Springer.
2. Theodoridis, S., & Koutroumbas, K. (2009). *Pattern Recognition*, *4th Edition*. Academic Press.

Questions?