Dimensional Reduction

Luu Minh Sao Khue

1 Introduction to Dimensional Reduction

Dimensional reduction is the process of reducing the number of features (dimensions) in a dataset while preserving its essential information. It is widely used in machine learning and data science to:

- Visualize high-dimensional data in 2D or 3D.
- Improve computational efficiency.
- Reduce noise and prevent overfitting.

1.1 Challenges: The Curse of Dimensionality

High-dimensional data suffers from the following challenges:

- Data points become sparse, making clustering and distance metrics ineffective.
- Increased computational cost for machine learning algorithms.
- Overfitting, as models may fit noise instead of meaningful patterns.

Dimensional reduction addresses these issues by finding meaningful lower-dimensional representations of the data.

2 Key Concepts

2.1 Feature Selection vs. Feature Extraction

Feature selection involves choosing a subset of the most important features (e.g., correlation with the target variable). Feature extraction involves creating new features by transforming the original data into a lower-dimensional space (e.g., Principal Component Analysis).

2.2 Linear vs. Nonlinear Dimensional Reduction

Linear methods assume linear relationships between features (e.g., PCA). Nonlinear methods capture complex, nonlinear patterns (e.g., t-SNE).

2.3 Variance Preservation and Projection

Dimensional reduction techniques aim to preserve variance (PCA focuses on directions of maximum variance) and project data from high-dimensional space to a lower-dimensional subspace.

3 Principal Component Analysis (PCA)

3.1 Intuition

PCA is a linear dimensional reduction technique that finds new axes (principal components) in the data space. These components maximize variance and are orthogonal (uncorrelated). By projecting the data onto the first few principal components, most of the information is retained while reducing dimensions.

3.2 Mathematical Steps

Given a dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$ with n samples and d features:

1. Standardize the data by centering it:

$$\mathbf{X}_{\text{centered}} = \mathbf{X} - \mu, \quad \mu = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$$

2. Compute the covariance matrix:

$$\mathbf{C} = \frac{1}{n-1} \mathbf{X}_{\text{centered}}^{\top} \mathbf{X}_{\text{centered}}$$

3. Perform eigen decomposition to find eigenvalues λ_i and eigenvectors $\mathbf{v}_i$:

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

4. Order eigenvectors by their eigenvalues (largest to smallest) and select the top k eigenvectors to form the projection matrix $\mathbf{W}$. 5. Project the data into the lower-dimensional space:

$$\mathbf{Z} = \mathbf{X}_{\text{centered}} \mathbf{W}$$

3.3 Example

Consider the following dataset in a 2D space:

$$\mathbf{X} = \begin{bmatrix} 2.5 & 2.4 \\ 0.5 & 0.7 \\ 2.2 & 2.9 \\ 1.9 & 2.2 \\ 3.1 & 3.0 \end{bmatrix}.$$

Step 1: Center the Data

Compute the mean for each feature:

$$\mu_1 = \frac{1}{5}(2.5 + 0.5 + 2.2 + 1.9 + 3.1) = 2.04, \quad \mu_2 = \frac{1}{5}(2.4 + 0.7 + 2.9 + 2.2 + 3.0) = 2.24.$$

Subtract the mean from each value to center the dataset:

$$\mathbf{X}_{\text{centered}} = \begin{bmatrix} 2.5 - 2.04 & 2.4 - 2.24 \\ 0.5 - 2.04 & 0.7 - 2.24 \\ 2.2 - 2.04 & 2.9 - 2.24 \\ 1.9 - 2.04 & 2.2 - 2.24 \\ 3.1 - 2.04 & 3.0 - 2.24 \end{bmatrix} = \begin{bmatrix} 0.46 & 0.16 \\ -1.54 & -1.54 \\ 0.16 & 0.66 \\ -0.14 & -0.04 \\ 1.06 & 0.76 \end{bmatrix}.$$

Step 2: Compute the Covariance Matrix

The covariance matrix is:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}_{\text{centered}}^{\top}\mathbf{X}_{\text{centered}}$$

Compute:

$$\mathbf{C} = \begin{bmatrix} 0.616 & 0.615 \\ 0.615 & 0.716 \end{bmatrix}.$$

Step 3: Perform Eigen Decomposition

The purpose of this step is to find the eigenvalues (λ) and eigenvectors ($\mathbf{v}$) of the covariance matrix $\mathbf{C}$, which represent the directions and magnitude of variance in the data.

Characteristic Equation Eigenvalues satisfy the characteristic equation:

$$\det(\mathbf{C} - \lambda\mathbf{I}) = 0,$$

where $\mathbf{I}$ is the identity matrix. Substitute the covariance matrix $\mathbf{C}$:

$$\mathbf{C} = \begin{bmatrix} 0.616 & 0.615 \\ 0.615 & 0.716 \end{bmatrix}, \quad \mathbf{C} - \lambda\mathbf{I} = \begin{bmatrix} 0.616 - \lambda & 0.615 \\ 0.615 & 0.716 - \lambda \end{bmatrix}.$$

Compute the determinant:

$$\det(\mathbf{C} - \lambda\mathbf{I}) = (0.616 - \lambda)(0.716 - \lambda) - (0.615)^2.$$

Expand:

$$(0.616 - \lambda)(0.716 - \lambda) = 0.616 \cdot 0.716 - (0.616 + 0.716)\lambda + \lambda^2,$$

$$\det(\mathbf{C} - \lambda\mathbf{I}) = \lambda^2 - 1.332\lambda + 0.044.$$

Solve for Eigenvalues Use the quadratic formula:

$$\lambda = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

where $a = 1$, $b = -1.332$, and $c = 0.044$. Substitute:

$$\lambda = \frac{1.332 \pm \sqrt{1.332^2 - 4 \cdot 1 \cdot 0.044}}{2},$$

$$\lambda = \frac{1.332 \pm \sqrt{1.774 - 0.176}}{2},$$

$$\lambda = \frac{1.332 \pm \sqrt{1.598}}{2}.$$

Simplify:

$$\lambda = \frac{1.332 \pm 1.264}{2}.$$

The eigenvalues are:

$$\lambda_1 = 1.284, \quad \lambda_2 = 0.048.$$

Find Eigenvectors For each eigenvalue, solve the equation:

$$(\mathbf{C} - \lambda \mathbf{I})\mathbf{v} = 0.$$

For $\lambda_1 = 1.284$:

$$\mathbf{C} - \lambda_1 \mathbf{I} = \begin{bmatrix} 0.616 - 1.284 & 0.615 \\ 0.615 & 0.716 - 1.284 \end{bmatrix} = \begin{bmatrix} -0.668 & 0.615 \\ 0.615 & -0.568 \end{bmatrix}.$$

Solve:

$$\begin{bmatrix} -0.668 & 0.615 \\ 0.615 & -0.568 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = 0.$$

The solution is:

$$\mathbf{v}_1 = \begin{bmatrix} 0.677 \\ 0.735 \end{bmatrix}.$$

For $\lambda_2 = 0.048$:

$$\mathbf{C} - \lambda_2 \mathbf{I} = \begin{bmatrix} 0.616 - 0.048 & 0.615 \\ 0.615 & 0.716 - 0.048 \end{bmatrix} = \begin{bmatrix} 0.568 & 0.615 \\ 0.615 & 0.668 \end{bmatrix}.$$

Solve:

$$\begin{bmatrix} 0.568 & 0.615 \\ 0.615 & 0.668 \end{bmatrix} \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = 0.$$

The solution is:

$$\mathbf{v}_2 = \begin{bmatrix} -0.735 \\ 0.677 \end{bmatrix}.$$

Final Eigenvalues and Eigenvectors The eigenvalues and their corresponding eigenvectors are:

$$\lambda_1 = 1.284, \quad \mathbf{v}_1 = \begin{bmatrix} 0.677 \\ 0.735 \end{bmatrix},$$

$$\lambda_2 = 0.048, \quad \mathbf{v}_2 = \begin{bmatrix} -0.735 \\ 0.677 \end{bmatrix}.$$

Step 4: Select the Principal Component

Since we are reducing to 1D, select the eigenvector corresponding to the largest eigenvalue λ_1:

$$\mathbf{v}_1 = \begin{bmatrix} 0.677 \\ 0.735 \end{bmatrix}.$$

Step 5: Project the Data

Project the centered data onto $\mathbf{v}_1$:

$$\mathbf{Z} = \mathbf{X}_{\text{centered}}\mathbf{v}_1.$$

Compute:

$$\mathbf{Z} = \begin{bmatrix} 0.485 \\ -2.165 \\ 0.548 \\ -0.122 \\ 1.981 \end{bmatrix}.$$

Final Result

The data in 2D is projected into 1D space:

$$\mathbf{Z} = \begin{bmatrix} 0.485 \\ -2.165 \\ 0.548 \\ -0.122 \\ 1.981 \end{bmatrix}.$$

4 t-Distributed Stochastic Neighbor Embedding (t-SNE)

4.1 Intuition

t-SNE is a nonlinear dimensional reduction technique used primarily for visualization. It preserves local neighborhood relationships while mapping high-dimensional data to 2D or 3D.

4.2 Mathematical Steps

1. Compute pairwise similarities in the high-dimensional space:

$$p_{ij} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq l}\exp(-\|\mathbf{x}_k - \mathbf{x}_l\|^2/2\sigma_k^2)}$$

2. Compute pairwise similarities in the low-dimensional space:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k\neq l}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

3. Minimize KL divergence between P and Q:

$$KL(P\|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

4.3 Example

Consider the following nonlinear dataset in 2D:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 2 & 4 \\ 3 & 9 \\ 4 & 16 \\ 5 & 25 \end{bmatrix}.$$

This dataset represents a quadratic relationship between the two features. The goal is to map this 2D dataset into a 1D space using t-SNE.

Step 1: Compute Pairwise Distances in High-Dimensional Space

Calculate the Euclidean distances between all pairs of points)

$$D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2}.$$

For this dataset:

$$\mathbf{D} = \begin{bmatrix} 0 & \sqrt{10} & \sqrt{80} & \sqrt{250} & \sqrt{520} \\ \sqrt{10} & 0 & \sqrt{50} & \sqrt{200} & \sqrt{450} \\ \sqrt{80} & \sqrt{50} & 0 & \sqrt{130} & \sqrt{320} \\ \sqrt{250} & \sqrt{200} & \sqrt{130} & 0 & \sqrt{170} \\ \sqrt{520} & \sqrt{450} & \sqrt{320} & \sqrt{170} & 0 \end{bmatrix}.$$

Step 2: Compute High-Dimensional Probabilities (P_{ij})

After calculating the pairwise Euclidean distances in Step 1, the next step is to convert these distances into high-dimensional probabilities P_{ij}. These probabilities represent the similarity between each pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the high-dimensional space.

The probability P_{ij} is defined as:

$$p_{ij} = \frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-d_{ik}^2/2\sigma_i^2)},$$

where:

- d_{ij}: Euclidean distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$.
- σ_i: Gaussian bandwidth parameter specific to $\mathbf{x}_i$, which determines the spread of the distribution around $\mathbf{x}_i$. For simplicity, we assume $\sigma_i = 1$ for all points.
- $\exp(-d_{ij}^2/2\sigma_i^2)$: The numerator, which represents the unnormalized similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$.

- The denominator ensures that the probabilities p_{ij} sum to 1 for each point i, i.e., $\sum_{j \neq i} p_{ij} = 1$.

1. Compute the Numerators for p_{ij}

Given the distances d_{ij}^2 from Step 1, calculate the numerator $\exp(-d_{ij}^2/2\sigma_i^2)$. Assume $\sigma_i = 1$. Using the dataset:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 2 & 4 \\ 3 & 9 \\ 4 & 16 \\ 5 & 25 \end{bmatrix},$$

the squared distances are:

$$\mathbf{D}^2 = \begin{bmatrix} 0 & 10 & 80 & 250 & 520 \\ 10 & 0 & 50 & 200 & 450 \\ 80 & 50 & 0 & 130 & 320 \\ 250 & 200 & 130 & 0 & 170 \\ 520 & 450 & 320 & 170 & 0 \end{bmatrix}.$$

Compute the numerator for each p_{ij}:

$$\exp(-d_{ij}^2/2\sigma_i^2) = \exp(-d_{ij}^2/2).$$

For example, for $i = 1$ (point $\mathbf{x}_1 = [1, 1]$):

$$\begin{aligned} &\exp(-d_{12}^2/2) = \exp(-10/2) = \exp(-5) \approx 0.0067, \\ &\exp(-d_{13}^2/2) = \exp(-80/2) = \exp(-40) \approx 4.25 \times 10^{-18}, \\ &\exp(-d_{14}^2/2) = \exp(-250/2) = \exp(-125) \approx 0, \\ &\exp(-d_{15}^2/2) = \exp(-520/2) = \exp(-260) \approx 0. \end{aligned}$$

2. Compute the Denominator for p_{ij}

For each i, sum the numerators for all $j \neq i$:

$$\sum_{j \neq i} \exp(-d_{ij}^2/2).$$

For $i = 1$:

$$\sum_{j \neq 1} \exp(-d_{1j}^2/2) = \exp(-5) + \exp(-40) + \exp(-125) + \exp(-260).$$

Using approximate values:

$$\sum_{j \neq 1} \approx 0.0067 + 4.25 \times 10^{-18} + 0 + 0 = 0.0067.$$

3. Compute p_{ij}:

Divide each numerator by the corresponding denominator to get p_{ij}:

$$p_{ij} = \frac{\exp(-d_{ij}^2/2)}{\sum_{k \neq i} \exp(-d_{ik}^2/2)}.$$

For $i = 1$ and $j = 2$:

$$p_{12} = \frac{\exp(-d_{12}^2/2)}{\sum_{j \neq 1} \exp(-d_{1j}^2/2)} = \frac{0.0067}{0.0067} = 1.$$

For $i = 1$ and $j = 3$:

$$p_{13} = \frac{\exp(-d_{13}^2/2)}{\sum_{j \neq 1} \exp(-d_{1j}^2/2)} = \frac{4.25 \times 10^{-18}}{0.0067} \approx 0.$$

Repeat for all pairs (i, j) to construct the matrix P.

4. Symmetrize P_{ij}:

To ensure symmetry, define:

$$P_{ij} = \frac{p_{ij} + p_{ji}}{2n},$$

where n is the total number of points.

For example, if $p_{12} = 1$ and $p_{21} = 0.9$, the symmetrized P_{12} becomes:

$$P_{12} = \frac{1 + 0.9}{2 \times 5} = 0.19.$$

Resulting Matrix: After computing and symmetrizing, the final high-dimensional probability matrix P might look like:

$$\mathbf{P} = \begin{bmatrix} 0 & 0.2000 & 0.0000 & 0.0000 & 0.0000 \\ 0.2000 & 0 & 0.1000 & 0.0000 & 0.0000 \\ 0.0000 & 0.1000 & 0 & 0.1000 & 0.0000 \\ 0.0000 & 0.0000 & 0.1000 & 0 & 0.1000 \\ 0.0000 & 0.0000 & 0.0000 & 0.1000 & 0 \end{bmatrix}.$$

Step 3: Initialize Low-Dimensional Representation

Randomly initialize the points in the low-dimensional (1D) space

$$\mathbf{Y} = \begin{bmatrix} 0.1 \\ 0.3 \\ 0.5 \\ 0.7 \\ 0.9 \end{bmatrix}.$$

Step 4: Compute Low-Dimensional Probabilities (Q_{ij})

Use the t-distribution to compute the pairwise probabilities in the low-dimensional space:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}.$$

For $i = 1, j = 2$:

$$\|\mathbf{y}_1 - \mathbf{y}_2\|^2 = (0.1 - 0.3)^2 = 0.04.$$

Compute:

$$q_{12} = \frac{1/(1 + 0.04)}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}.$$

Step 5: Minimize KL Divergence

Use gradient descent to minimize the KL divergence:

$$KL(P\|Q) = \sum_{i \neq j} P_{ij} \log \frac{P_{ij}}{Q_{ij}}.$$

Compute the gradient for $\mathbf{Y}_i$:

$$\frac{\partial KL}{\partial \mathbf{Y}_i} = 4 \sum_{j \neq i} (P_{ij} - Q_{ij})\, Q_{ij} (\mathbf{Y}_i - \mathbf{Y}_j).$$

Update $\mathbf{Y}_i$ using a learning rate $\eta = 0.1$:

$$\mathbf{Y}_i = \mathbf{Y}_i - \eta \cdot \frac{\partial KL}{\partial \mathbf{Y}_i}.$$

Step 6: Iterate Until Convergence

Repeat the process for all points $\mathbf{Y}_i$ and iterate until the KL divergence stabilizes.

Result: After several iterations, the resulting 1D embedding is:

$$\mathbf{Y} = \begin{bmatrix} 0.15 \\ 0.45 \\ 0.85 \\ 1.25 \\ 1.75 \end{bmatrix}.$$

The 1D embedding preserves the local relationships of the original nonlinear dataset.