



Dimensionality reduction

General idea

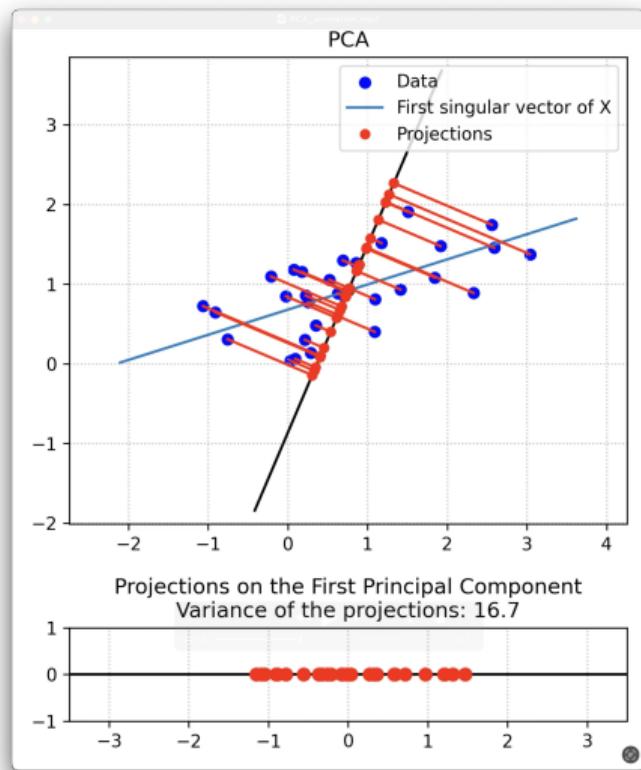
PCA

PCA optimization problem



The first component should be defined in order to maximize the projection variance. Suppose, we've already normalized the data, i.e. $\sum_i a_i = 0$, then sample variance will become the sum of all squared projections of data points to our vector $\mathbf{w}_{(1)}$, which implies the following optimization problem:

PCA optimization problem



The first component should be defined in order to maximize the projection variance. Suppose, we've already normalized the data, i.e. $\sum_i a_i = 0$, then sample variance will become the sum of all squared projections of data points to our vector $\mathbf{w}_{(1)}$, which implies the following optimization problem:

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \sum_i (\mathbf{a}_{(i)}^\top \cdot \mathbf{w})^2 \right\}$$

or

PCA optimization problem



The first component should be defined in order to maximize the projection variance. Suppose, we've already normalized the data, i.e. $\sum_i a_i = 0$, then sample variance will become the sum of all squared projections of data points to our vector $\mathbf{w}_{(1)}$, which implies the following optimization problem:

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \sum_i (\mathbf{a}_{(i)}^\top \cdot \mathbf{w})^2 \right\}$$

or

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \{\|\mathbf{Aw}\|^2\} = \arg \max_{\|\mathbf{w}\|=1} \{\mathbf{w}^\top \mathbf{A}^\top \mathbf{Aw}\}$$

PCA optimization problem



The first component should be defined in order to maximize the projection variance. Suppose, we've already normalized the data, i.e. $\sum_i a_i = 0$, then sample variance will become the sum of all squared projections of data points to our vector $\mathbf{w}_{(1)}$, which implies the following optimization problem:

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \sum_i (\mathbf{a}_{(i)}^\top \cdot \mathbf{w})^2 \right\}$$

or

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \{\|\mathbf{Aw}\|^2\} = \arg \max_{\|\mathbf{w}\|=1} \{\mathbf{w}^\top \mathbf{A}^\top \mathbf{Aw}\}$$

since we are looking for the unit vector, we can reformulate the problem:

$$\mathbf{w}_{(1)} = \arg \max \left\{ \frac{\mathbf{w}^\top \mathbf{A}^\top \mathbf{Aw}}{\mathbf{w}^\top \mathbf{w}} \right\}$$

PCA optimization problem



The first component should be defined in order to maximize the projection variance. Suppose, we've already normalized the data, i.e. $\sum_i a_i = 0$, then sample variance will become the sum of all squared projections of data points to our vector $\mathbf{w}_{(1)}$, which implies the following optimization problem:

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \sum_i (\mathbf{a}_{(i)}^\top \cdot \mathbf{w})^2 \right\}$$

or

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \{\|\mathbf{Aw}\|^2\} = \arg \max_{\|\mathbf{w}\|=1} \{\mathbf{w}^\top \mathbf{A}^\top \mathbf{Aw}\}$$

since we are looking for the unit vector, we can reformulate the problem:

$$\mathbf{w}_{(1)} = \arg \max \left\{ \frac{\mathbf{w}^\top \mathbf{A}^\top \mathbf{Aw}}{\mathbf{w}^\top \mathbf{w}} \right\}$$

It is known, that for the positive semidefinite matrix $\mathbf{A}^\top \mathbf{A}$ such vector is nothing else, but an eigenvector of $\mathbf{A}^\top \mathbf{A}$, which corresponds to the largest eigenvalue.

Algorithm derivation

So, we can conclude, that the following mapping:

$$\Pi_{n \times k} = A_{n \times d} \cdot W_{d \times k}$$

describes the projection of data onto the k principal components, where W contains first (by the size of eigenvalues) k eigenvectors of $A^\top A$.

Now we'll briefly derive how SVD decomposition could lead us to the PCA.

Firstly, we write down SVD decomposition of our matrix:

$$A = U \Sigma W^\top$$

and to its transpose:

$$\begin{aligned} A^\top &= (U \Sigma W^\top)^\top \\ &= (W^\top)^\top \Sigma^\top U^\top \\ &= W \Sigma^\top U^\top \\ &= W \Sigma U^\top \end{aligned}$$

Then, consider matrix AA^\top :

$$\begin{aligned} A^\top A &= (W \Sigma U^\top)(U \Sigma V^\top) \\ &= W \Sigma I \Sigma W^\top \\ &= W \Sigma \Sigma W^\top \\ &= W \Sigma^2 W^\top \end{aligned}$$

Which corresponds to the eigendecomposition of matrix $A^\top A$, where W stands for the matrix of eigenvectors of $A^\top A$, while Σ^2 contains eigenvalues of $A^\top A$.

At the end:

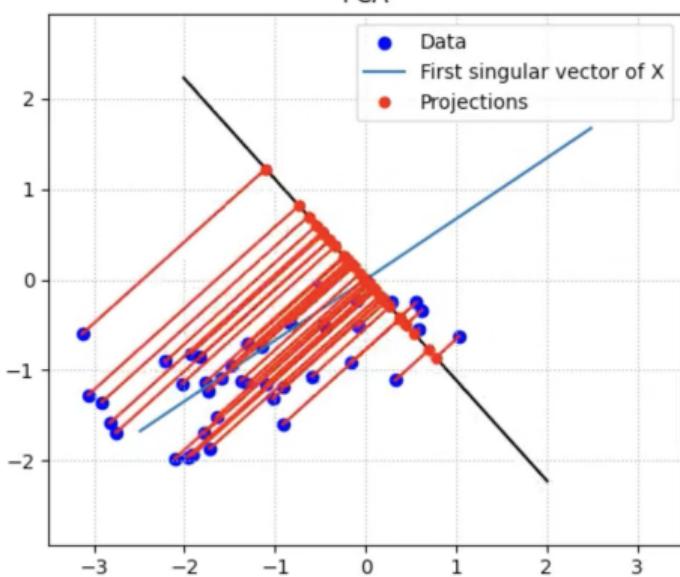
$$\begin{aligned} \Pi &= A \cdot W = \\ &= U \Sigma W^\top W = U \Sigma \end{aligned}$$

The latter formula provide us with easy way to compute PCA via SVD with any number of principal components:

$$\Pi_r = U_r \Sigma_r$$

Exercise 1

What could be wrong with this PCA?

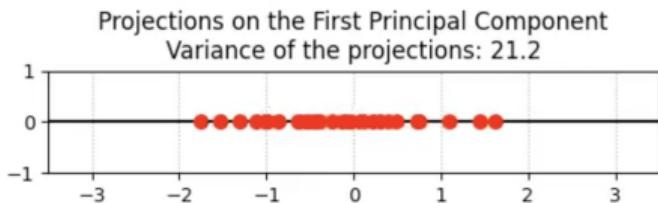


Projections on the First Principal Component
Variance of the projections: 13.2



Exercise 2

What could be wrong with this PCA?

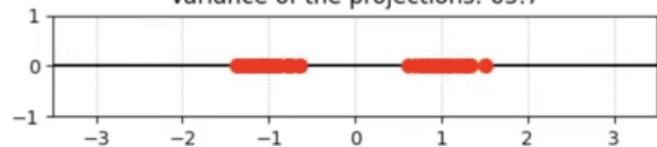


Exercise 3

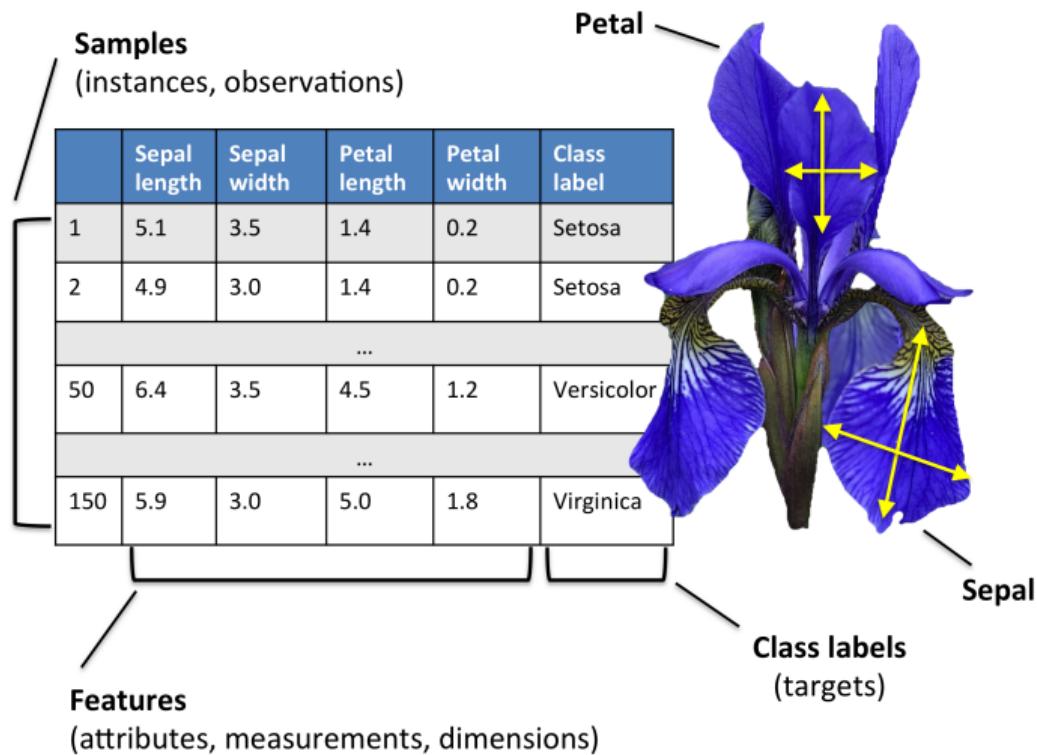
What could be wrong with this PCA?



Projections on the First Principal Component
Variance of the projections: 65.7



Iris dataset variance



Iris dataset variance



Wine dataset variance



PCA on MNIST

2D PCA of MNIST



Other methods

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique
particularly well-suited for visualizing high-dimensional
data in 2 or 3 dimensions.

- Key Concepts:

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- Key Concepts:

- Pairwise Similarities: Computes probabilities that pairs of high-dimensional objects are related.

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.
- **Student's t-Distribution:** Uses a heavy-tailed distribution in the low-dimensional space to effectively model distant points and mitigate the "crowding problem."

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Algorithm Steps:**

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.
- **Student's t-Distribution:** Uses a heavy-tailed distribution in the low-dimensional space to effectively model distant points and mitigate the "crowding problem."

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.
- **Student's t-Distribution:** Uses a heavy-tailed distribution in the low-dimensional space to effectively model distant points and mitigate the "crowding problem."

- **Algorithm Steps:**

1. **Compute High-Dimensional Probabilities:** Use Gaussian distributions to model pairwise similarities.

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.
- **Student's t-Distribution:** Uses a heavy-tailed distribution in the low-dimensional space to effectively model distant points and mitigate the "crowding problem."

- **Algorithm Steps:**

1. **Compute High-Dimensional Probabilities:** Use Gaussian distributions to model pairwise similarities.
2. **Initialize Low-Dimensional Embedding:** Start with random positions.

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.
- **Student's t-Distribution:** Uses a heavy-tailed distribution in the low-dimensional space to effectively model distant points and mitigate the "crowding problem."

- **Algorithm Steps:**

1. **Compute High-Dimensional Probabilities:** Use Gaussian distributions to model pairwise similarities.
2. **Initialize Low-Dimensional Embedding:** Start with random positions.
3. **Optimize Embedding:** Iteratively update positions to minimize divergence between distributions.

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.
- **Student's t-Distribution:** Uses a heavy-tailed distribution in the low-dimensional space to effectively model distant points and mitigate the "crowding problem."

- **Algorithm Steps:**

1. **Compute High-Dimensional Probabilities:** Use Gaussian distributions to model pairwise similarities.
2. **Initialize Low-Dimensional Embedding:** Start with random positions.
3. **Optimize Embedding:** Iteratively update positions to minimize divergence between distributions.

- **Considerations:**

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.
- **Student's t-Distribution:** Uses a heavy-tailed distribution in the low-dimensional space to effectively model distant points and mitigate the "crowding problem."

- **Algorithm Steps:**

1. **Compute High-Dimensional Probabilities:** Use Gaussian distributions to model pairwise similarities.
2. **Initialize Low-Dimensional Embedding:** Start with random positions.
3. **Optimize Embedding:** Iteratively update positions to minimize divergence between distributions.

- **Considerations:**

- **Perplexity Parameter (Perplexity):** Balances attention between local and global aspects of the data.

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.
- **Student's t-Distribution:** Uses a heavy-tailed distribution in the low-dimensional space to effectively model distant points and mitigate the "crowding problem."

- **Algorithm Steps:**

1. **Compute High-Dimensional Probabilities:** Use Gaussian distributions to model pairwise similarities.
2. **Initialize Low-Dimensional Embedding:** Start with random positions.
3. **Optimize Embedding:** Iteratively update positions to minimize divergence between distributions.

- **Considerations:**

- **Perplexity Parameter (Perplexity):** Balances attention between local and global aspects of the data.
- **Computational Complexity:** Can be slow for large datasets due to pairwise computations.

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE)

is a nonlinear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data in 2 or 3 dimensions.

- **Key Concepts:**

- **Pairwise Similarities:** Computes probabilities that pairs of high-dimensional objects are related.
- **High to Low Dimensional Mapping:** Seeks a low-dimensional embedding where the probability distributions of pairwise similarities are preserved.
- **Cost Function:** Minimizes the Kullback-Leibler divergence between the high-dimensional and low-dimensional probability distributions.
- **Student's t-Distribution:** Uses a heavy-tailed distribution in the low-dimensional space to effectively model distant points and mitigate the "crowding problem."

- **Algorithm Steps:**

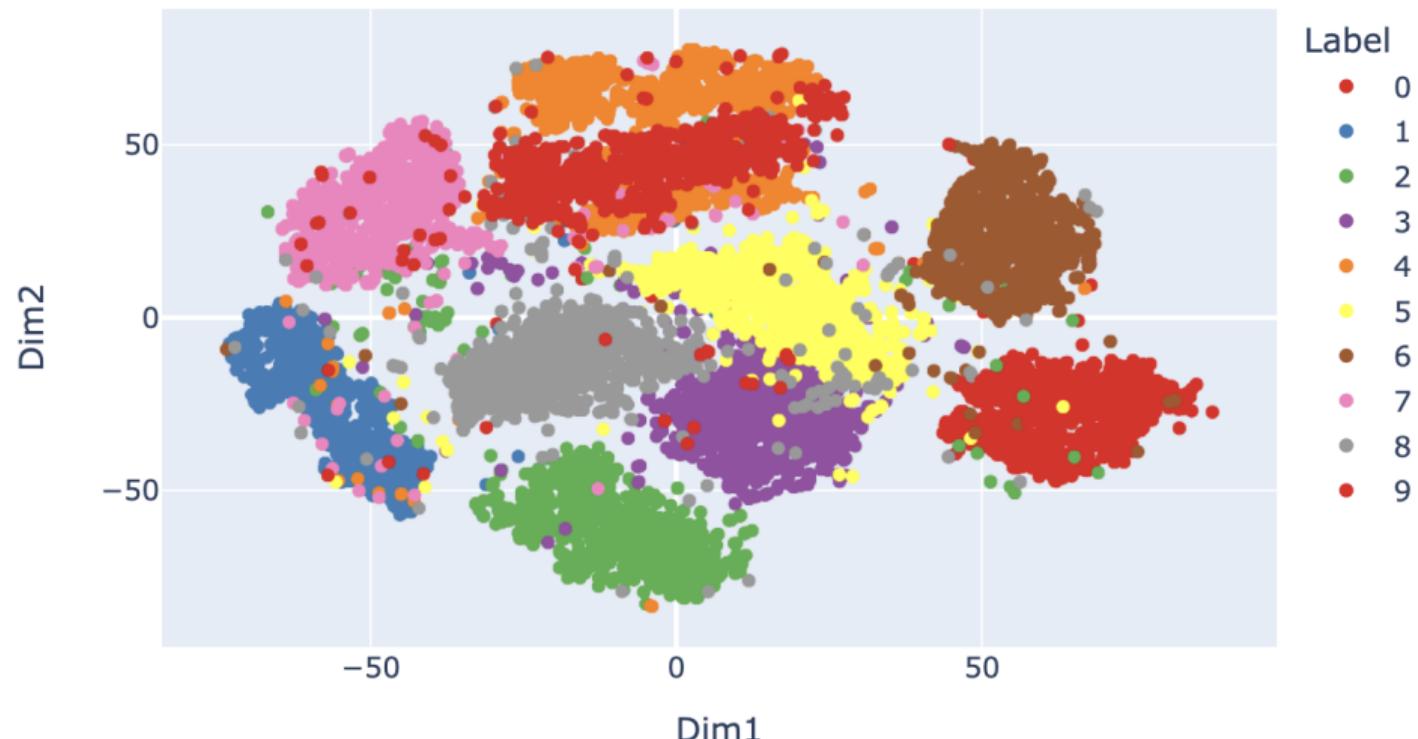
1. **Compute High-Dimensional Probabilities:** Use Gaussian distributions to model pairwise similarities.
2. **Initialize Low-Dimensional Embedding:** Start with random positions.
3. **Optimize Embedding:** Iteratively update positions to minimize divergence between distributions.

- **Considerations:**

- **Perplexity Parameter (Perplexity):** Balances attention between local and global aspects of the data.
- **Computational Complexity:** Can be slow for large datasets due to pairwise computations.
- **Random Initialization:** Different runs may yield different results; multiple runs can help validate findings.

t-SNE on MNIST

2D t-SNE of MNIST



UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- Key Concepts:

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- Key Concepts:

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

UMAP

Uniform Manifold Approximation and Projection (UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.

UMAP

Uniform Manifold Approximation and Projection (UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.
2. **Compute Fuzzy Simplicial Sets:** Model the probability distribution of data relationships.

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.
2. **Compute Fuzzy Simplicial Sets:** Model the probability distribution of data relationships.
3. **Optimize Low-Dimensional Embedding:** Apply stochastic gradient descent to minimize cross-entropy between high and low-dimensional graphs.

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.
2. **Compute Fuzzy Simplicial Sets:** Model the probability distribution of data relationships.
3. **Optimize Low-Dimensional Embedding:** Apply stochastic gradient descent to minimize cross-entropy between high and low-dimensional graphs.

- **Advantages:**

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.
2. **Compute Fuzzy Simplicial Sets:** Model the probability distribution of data relationships.
3. **Optimize Low-Dimensional Embedding:** Apply stochastic gradient descent to minimize cross-entropy between high and low-dimensional graphs.

- **Advantages:**

- **Speed:** Faster than t-SNE, suitable for large datasets.

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.
2. **Compute Fuzzy Simplicial Sets:** Model the probability distribution of data relationships.
3. **Optimize Low-Dimensional Embedding:** Apply stochastic gradient descent to minimize cross-entropy between high and low-dimensional graphs.

- **Advantages:**

- **Speed:** Faster than t-SNE, suitable for large datasets.
- **Preservation of Structure:** Maintains more global structure compared to t-SNE.

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.
2. **Compute Fuzzy Simplicial Sets:** Model the probability distribution of data relationships.
3. **Optimize Low-Dimensional Embedding:** Apply stochastic gradient descent to minimize cross-entropy between high and low-dimensional graphs.

- **Advantages:**

- **Speed:** Faster than t-SNE, suitable for large datasets.
- **Preservation of Structure:** Maintains more global structure compared to t-SNE.
- **Scalability:** Can handle millions of data points efficiently.

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.
2. **Compute Fuzzy Simplicial Sets:** Model the probability distribution of data relationships.
3. **Optimize Low-Dimensional Embedding:** Apply stochastic gradient descent to minimize cross-entropy between high and low-dimensional graphs.

- **Advantages:**

- **Speed:** Faster than t-SNE, suitable for large datasets.
- **Preservation of Structure:** Maintains more global structure compared to t-SNE.
- **Scalability:** Can handle millions of data points efficiently.

- **Parameters:**

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.
2. **Compute Fuzzy Simplicial Sets:** Model the probability distribution of data relationships.
3. **Optimize Low-Dimensional Embedding:** Apply stochastic gradient descent to minimize cross-entropy between high and low-dimensional graphs.

- **Advantages:**

- **Speed:** Faster than t-SNE, suitable for large datasets.
- **Preservation of Structure:** Maintains more global structure compared to t-SNE.
- **Scalability:** Can handle millions of data points efficiently.

- **Parameters:**

- **Number of Neighbors ($n_{neighbors}$):** Controls local versus global structure preservation.

UMAP

Uniform Manifold Approximation and Projection

(UMAP) is a nonlinear dimensionality reduction technique that preserves both local and global data structure.

- **Key Concepts:**

- **Manifold Learning:** Assumes data lies on a manifold in high-dimensional space.
- **Topological Data Analysis:** Utilizes concepts from topology to model the manifold structure.
- **Graph Construction:** Builds a weighted graph representing data relationships in high-dimensional space.
- **Optimization:** Seeks a low-dimensional embedding that has a similar topological structure to the high-dimensional graph.

- **Algorithm Steps:**

1. **Construct High-Dimensional Graph:** Use k-nearest neighbors to build the graph.
2. **Compute Fuzzy Simplicial Sets:** Model the probability distribution of data relationships.
3. **Optimize Low-Dimensional Embedding:** Apply stochastic gradient descent to minimize cross-entropy between high and low-dimensional graphs.

- **Advantages:**

- **Speed:** Faster than t-SNE, suitable for large datasets.
- **Preservation of Structure:** Maintains more global structure compared to t-SNE.
- **Scalability:** Can handle millions of data points efficiently.

- **Parameters:**

- **Number of Neighbors ($n_{neighbors}$):** Controls local versus global structure preservation.
- **Minimum Distance (min_dist):** Dictates how tightly points are packed in the low-dimensional space.

UMAP on MNIST

2D UMAP of MNIST

