DIMENSIONALITY REDUCTION USING T-SNE AND UMAP

What Is Dimensionality Reduction?

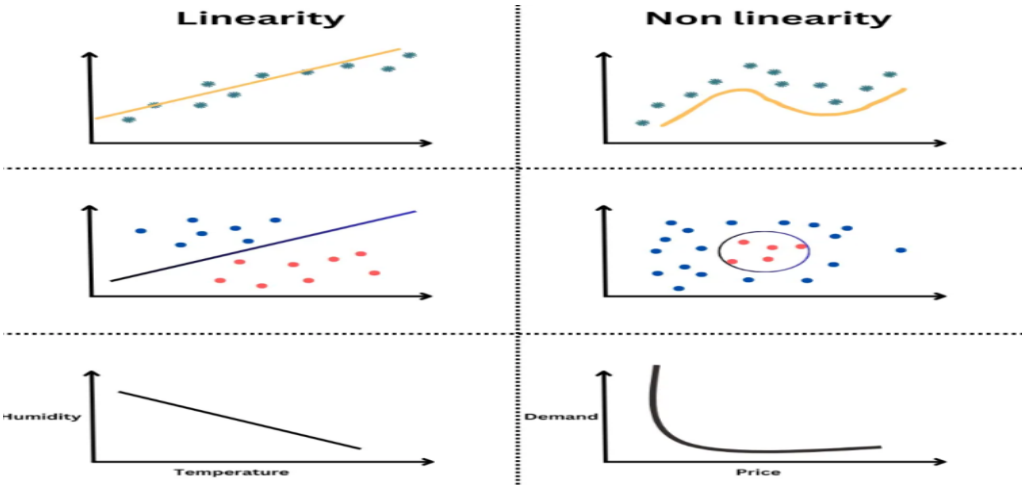
- Dimensionality reduction transforms data from a high-dimensional feature space into a lower-dimensional representation while retaining as much relevant structure as possible.
- The transformation can be **linear**, as in **Principal Component Analysis (PCA),** or **nonlinear**, as in t-SNE and UMAP, and typically aims to preserve variance, neighborhood relationships, or class separation.

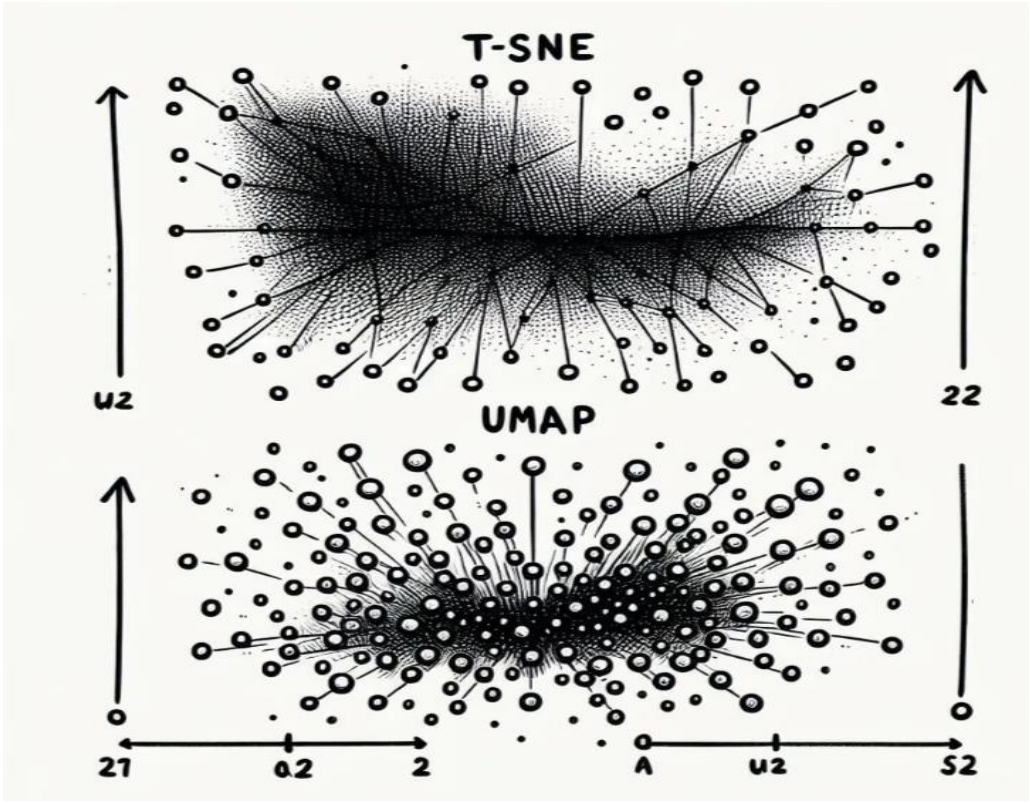
Why Dimensionality Reduction Matters

- ***Visualization***: Projecting data to 2D or 3D enables intuitive exploration and communication of clusters and trends.
- ***Noise reduction***: Low-variance components often correspond to noise or redundant information, which can be removed to stabilize models.
- ***Efficiency***: Fewer dimensions lead to faster training, smaller models, and lower storage costs, especially for linear models and k-NN.
- ***Regularization***: Compressing features can reduce overfitting by forcing models to focus on dominant patterns rather than spurious fluctuations.

What is Non-Linear Dimensionality Reduction?

- When datasets have high dimensionality — think of 100+ features — comprehending their relationships and patterns becomes challenging. While techniques like **PCA (Principal Component Analysis)** reduce dimensionality linearly, they may fail to capture non-linear relationships.
- Non-linear dimensionality reduction techniques like t-SNE and UMAP are designed to **preserve local and global structures** in data with non-linear dependencies, making them invaluable for data visualization and clustering.





What is t-SNE ?

- t-SNE (t-distributed Stochastic Neighbor Embedding) is an ***unsupervised non-linear dimensionality reduction technique*** for data exploration and visualizing high-dimensional data.

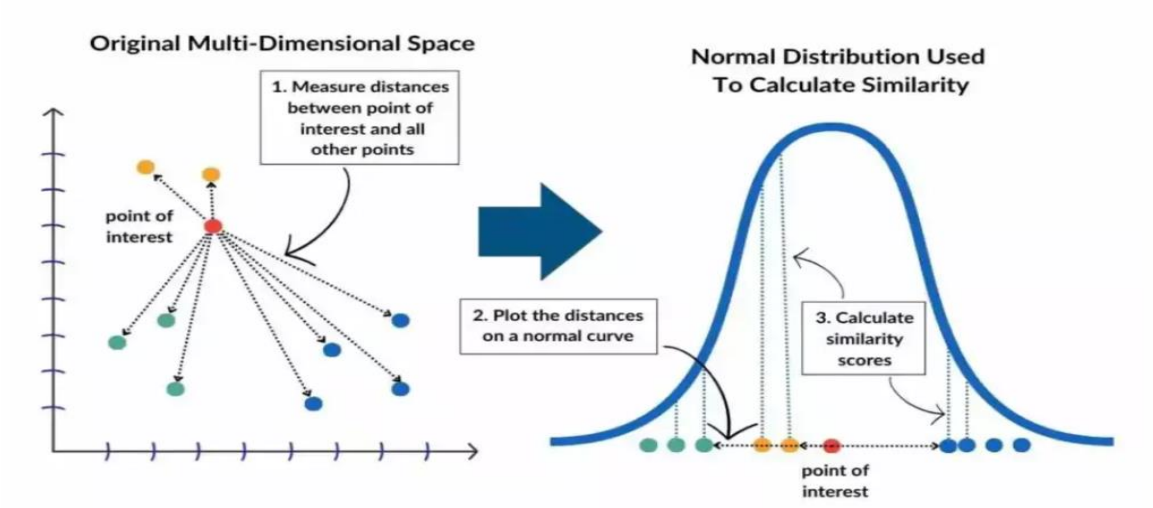
- **Non-linear dimensionality** reduction means that the algorithm allows us to separate data that **cannot be separated by a straight line**.

t-SNE **vs** **PCA**

- PCA (Principal Component Analysis) is a linear technique that works best with data that has a linear structure. It seeks to identify the underlying principal components in the data by projecting onto lower dimensions, minimizing variance, and preserving large pairwise distances.
- t-SNE is a nonlinear technique that focuses on preserving the pairwise similarities between data points in a lower-dimensional space. t-SNE is concerned with preserving small pairwise distances whereas, PCA focuses on maintaining large pairwise distances to maximize variance.

t-SNE: A Journey Through Neighborhoods

- Think of t-SNE as a detective that uncovers hidden communities in a city. It maps high-dimensional data points into a lower-dimensional space by focusing on **preserving local neighborhoods** — the points close to each other in the original space remain close in the reduced space.



How t-SNE Works

- **1.Compute Pairwise Similarities:**
- In the original high-dimensional space, t-SNE calculates the **probability of similarity between each pair of points** using a Gaussian distribution.

- In the lower-dimensional space, it uses a Student's t-distribution to model these similarities, ensuring a better spread of points.
- **2.Minimize Divergence:**
- The goal is to minimize the difference (Kullback-Leibler divergence) between the similarity distributions of the original and reduced spaces.
- **3.Iterative Optimization:**
- The positions of the points in the reduced space are updated iteratively to preserve the neighborhoods as closely as possible.

Limitations and Challenges of t-SNE

- **Computational cost**: t-SNE is computationally expensive, especially for large datasets. Its pairwise similarity calculations scale poorly with the size of the dataset, making it less suitable for datasets with millions of points.
- **Sensitivity to hyperparameters**: The performance of t-SNE is highly dependent on hyperparameters like perplexity and learning rate. Finding the optimal values often requires trial and error, which can be time-consuming.
- **Lack of interpretability**: The resulting embeddings in t-SNE are often non-deterministic (due to random initialization and gradient descent) and lack a direct interpretable relationship with the original high-dimensional data.
- **Not suitable for out-of-sample data**: t-SNE cannot generalize to new, unseen data without re-computing the embeddings from scratch, which is inefficient for dynamic datasets.

Applications of t-SNE

- **Clustering and classification:** to cluster similar data points together in lower dimensional space. It can also be used for classification and finding patterns in the data.
- **Anomaly detection:** to identify outliers and anomalies in the data.
- **Natural language processing:** to visualize word embeddings generated from a large corpus of text that makes it easier to identify similarities and relationships between words.
- **Computer security:** to visualize network traffic patterns and detect anomalies.
- **Cancer research:** to visualize molecular profiles of tumor samples and identify cancer subtypes.
- **Geological domain interpretation:** to visualize seismic attributes and to identify geological anomalies.
- **Biomedical signal processing:** to visualize electroencephalogram (EEG) and detect patterns of brain activity.

UMAP: Beyond t-SNE

Why UMAP?

- Stands for *Uniform Manifold Approximation and Projection*
- While t-SNE is excellent at capturing local structures, it has its limitations:
- Computationally intensive for large datasets.

- Struggles to preserve global relationships.
- Enter UMAP — a faster, more scalable alternative that balances both **local and global structures** in data.

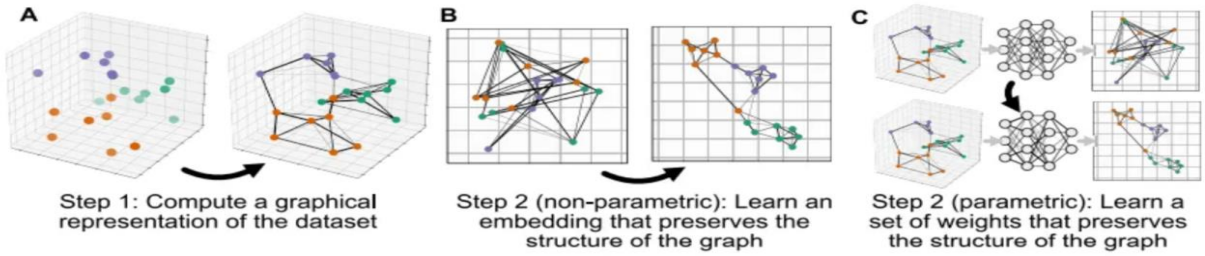


Figure 1: Overview of UMAP (A → B) and Parametric UMAP (A → C).

The Math Behind UMAP

- UMAP relies on **manifold learning**, assuming that the data lies on a low-dimensional manifold embedded in high-dimensional space. Here's how it works:
- **1.Graph Construction:**
- UMAP builds a weighted graph of the data's nearest neighbors in high-dimensional space.
- It uses a fuzzy membership function to quantify relationships between points.
- **2.Graph Optimization:**
- It **maps** the high-dimensional graph into lower dimensions while minimizing the loss of structure.
- This step preserves both local and larger-scale relationships.

Advantages of UMAP

- **Speed:** Faster than t-SNE, especially for large datasets.
- **Scalability:** Handles millions of data points efficiently.
- **Flexibility:** Works well with sparse and dense data.

Comparison Table for t-SNE and PCA

Characteristic	t-SNE	PCA
Type	Non-linear dimensionality reduction	Linear dimensionality reduction
Goal	Preserve local pairwise similarities	Preserve global variance
Best used for	Visualizing complex, high-dimensional data	Data with linear structure
Output	Low-dimensional representation	Principal components
Use cases	Clustering, anomaly detection, NLP	Noise reduction, feature extraction
Computational intensity	High	Low

Choosing the Right Technique

1.Use PCA

- When you want speed, scalability, and interpretability.
- For preprocessing data or simplifying features in a pipeline.
- If the dataset is believed to have linear relationships.

2.Use t-SNE

- When exploring **local clusters** in a complex dataset.
- For smaller datasets (up to tens of thousands of points).
- If your primary goal is visualization.

3.Use UMAP

- When working with **large datasets**.
- If you need to balance **local and global structures**.
- For tasks requiring fast visualization or embedding sparse data.