

UMAP

ML2: AI Concepts and Algorithms (SS2025)
Faculty of Computer Science and Applied Mathematics
University of Applied Sciences Technikum Wien

Lecturer: Rosana de Oliveira Gomes

Authors: B. Knapp, M. Blaickner, S. Rezagholi, R.O. Gomes



Clustering

k-means
Hierarchical clustering
DB-scan

Regression

KNN regression
Regression trees
Linear regression
Multiple regression
Ridge and Lasso regression
Neural networks

Classification

KNN classification
Classification trees
Ensembles & boosting
Random Forest
Logistic regression
Naive Bayes
Support vector machines
Neural networks

Supervised learning

Data handling

EDA
Data cleaning
Feature selection
Class balancing
etc

AI

Non-supervised learning

Dimensionality reduction

PCA / SVD
tSNE
UMAP
MDS

Reinforcement learning

Covered in a separate lecture.

Dimensionality Reduction Recap

Dimensionality curse: challenges and complexities that arise when working with high-dimensional data, where the number of features or variables is significantly large

High dimensional systems: e.g. genomics, environmental science, NLP, customer segmentation.

Dimensionality reduction transforms data from high-dimensional space into a low-dimensional space while **preserving as much of the dataset's original information as possible.**

How to convert a multi-dimensional dataset into a small dimension visualization?

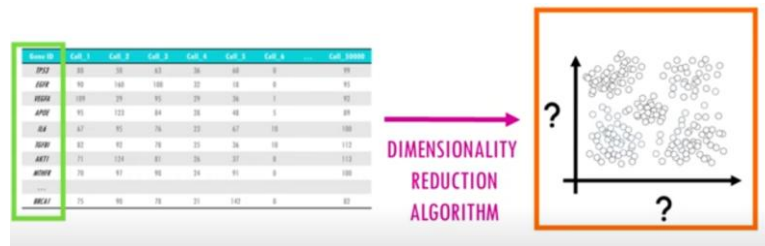



Image Source: 

Common algorithms

- Principle Component Analysis
- T-SNE
- UMAP

PCA Recap: Principal Component Analysis

PCA captures the essence of the data into few **principal components (usually 2-5 PCs)**.

Principal Components:

- Summarize patterns or variations in the dataset.
- Are ranked and independent from each other.
- PCs are constructed as a *linear combination* or mixture of the original variables.

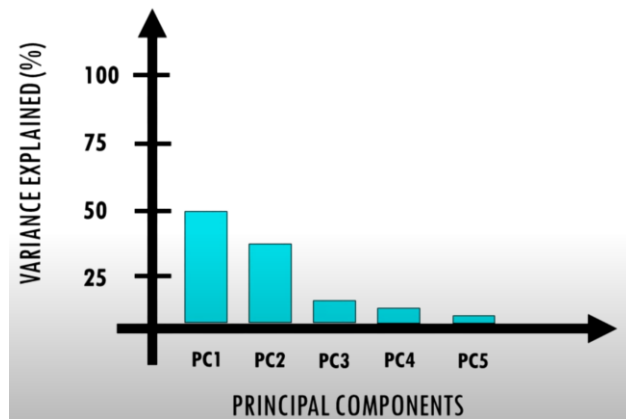


Image Source:

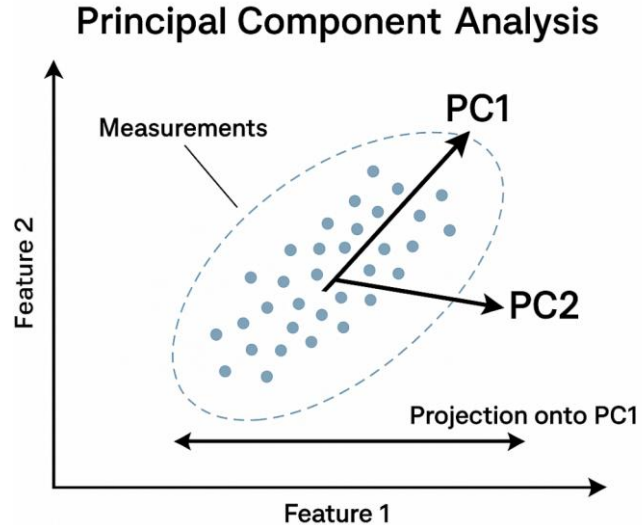
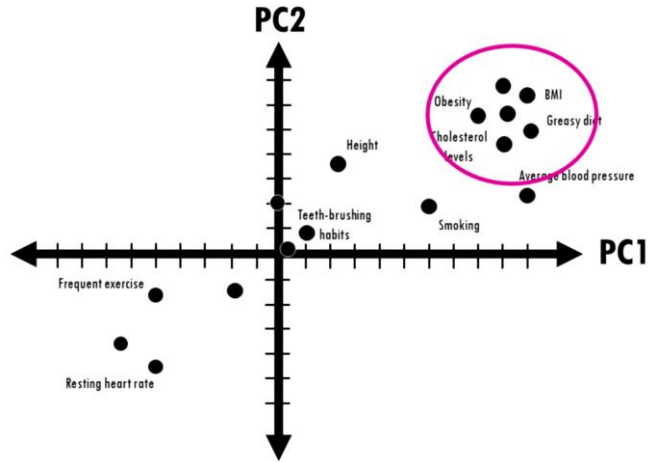


	Lifespan	PC1	PC2	PC3	PC4	PC5
Person 1	82	-1	3	-1	4	4
Person 2	73	2	4	2	5	5
Person 3	95	3	2	4	2	2
Person 4	92	4	4	5	-4	-4
Person 5	87	5	5	2	2	5
Person 6	65	2	5	-4	3	2
Person 7	93	-4	-6	5	5	-4
Person 8	80	-3	-6	-6	2	5
...						
Person 20	72	8	-3	-6	-3	-6

	PC1	PC2	PC3	PC4	PC5
Height	-1	3	-1	4	4
Average heart rate	9	7	5	-4	-4
BMI	10	6.5	2	2	5
Cholesterol levels	9	5	-4	3	2
Average cigarettes/day	7	2	5	5	-4
Greasy diet	10	5	-6	2	5
Frequent exercise	-5	-6	8	1	9
Eye colour	0.1	0.3	0.1	0.3	0.3
Teeth-brushing habits	0.2	0.2	0.2	0.2	0.2

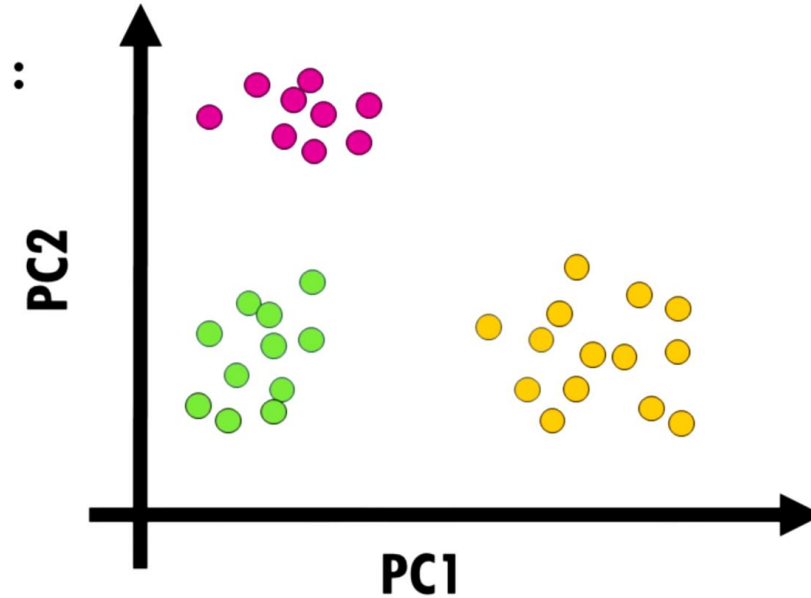
PCA Recap: Principal Component Analysis

The contribution of the variables to each principal component (PC) is indicated by their *loadings*.



PCA Recap: Principal Component Analysis

Observations with similar profiles are clustered together in a PCA plot.



t-SNE Recap: T-distributed Stochastic Neighbour Embedding

[Visualizing Data using t-SNE Laurens van der Maaten, Geoffrey Hinton: 9\(86\):2579–2605, 2008.](#)

Dimensionality reduction based on similarities across data points.
Allows for **non-linearity** across variables.

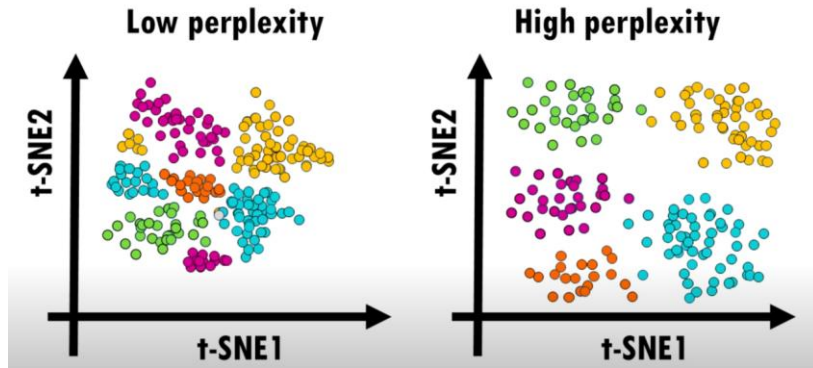

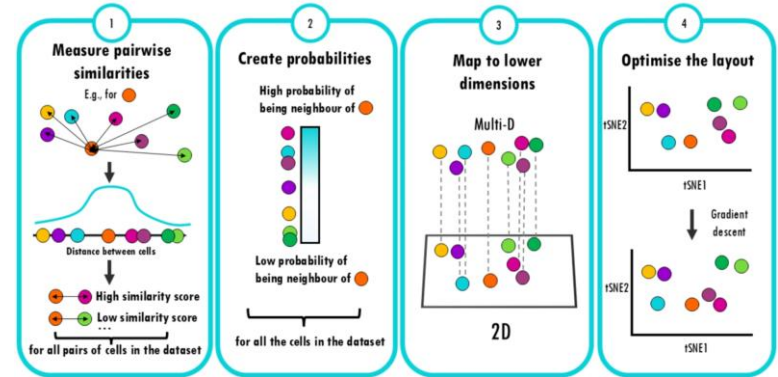


Image Source: 

Perplexity hyperparameter:
how many neighbour each data points is considering when building a similarity map

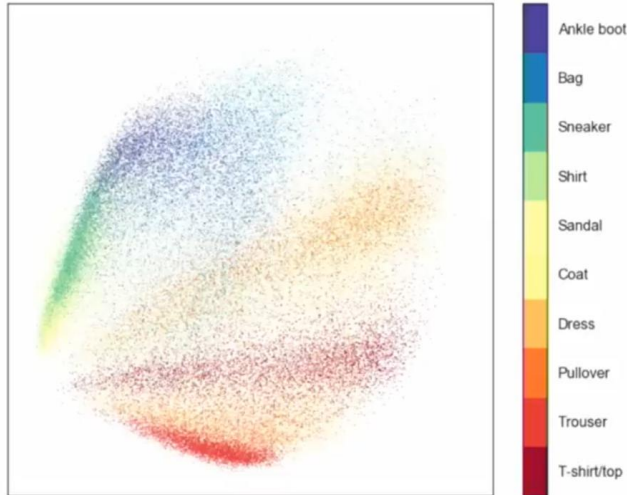


Algorithm:

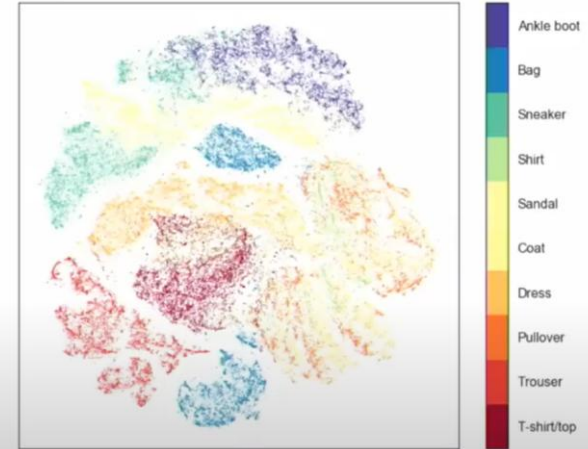
1. Measure similarity score for all data-pairs in the dataset.
2. Calculate likelihood that two data points are neighbours.
3. Map the data to lower dimensions.
4. Make the distribution of similarities in the lower dimension space match the original distribution as closely as possible (with gradient descent).

PCA vs t-SNE

PCA on Fashion MNIST



t-SNE on Fashion MNIST

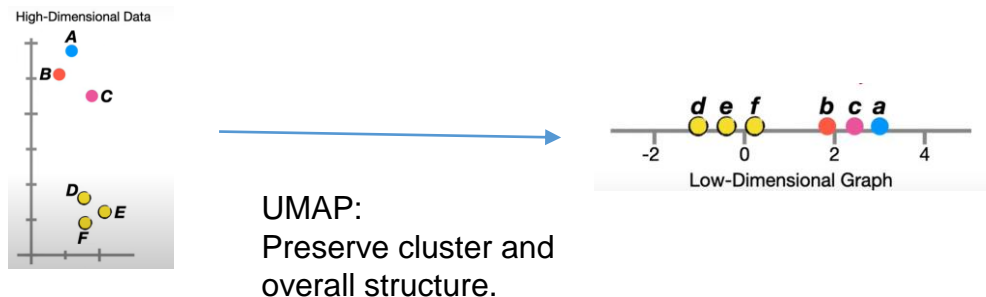


UMAP: Uniform Manifold Approximation and Projection

UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction

Leland McInnes, John Healy, James Melville (2018)

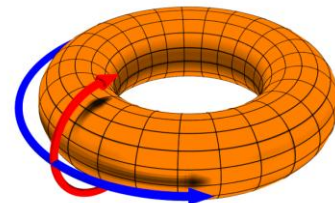
Motivation: Visualization of complex, high dimensional data in fewer dimensions applying topology concepts.



Assumptions

High-dimensional data lies on a lower dimensional manifold (curved surface).

Riemannian manifold: the data is uniformly distributed.

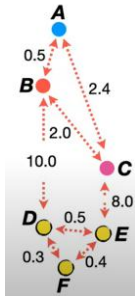


UMAP: Algorithm

Construct a High-dimensional graph:
capture the local relationship between points.

Step 1: Local Distances and Weights

Raw Distances



E.g.: Looking at **point A**.

- Calculates the distances between distances across all k-neighbors.
- Calculate similarity scores keeping the same distances in the lower dimensional set (depends on the number of neighbors).

Low-Dimensional Representation:

UMAP tries to find a **low-dimensional embedding** that preserves these local structures as much as possible.

Step 2: Low-Dimensional Distances

- The pairwise similarity between points in the low-dimensional space is dependent on hyperparameters that control the shape of the curve.

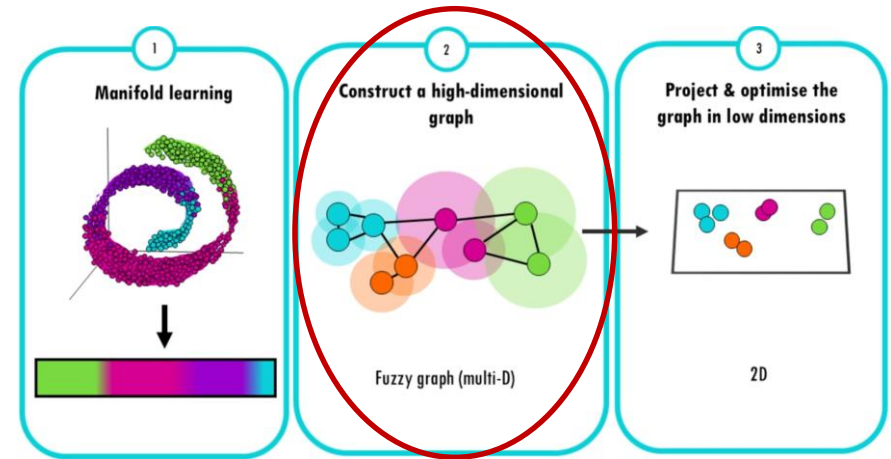
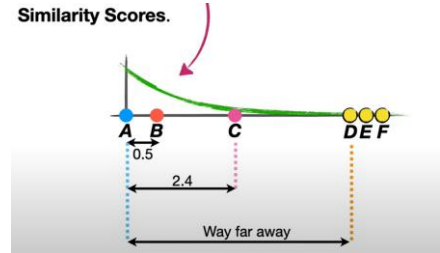


Image Source:



In relation to **point A**.

Similarity Scores.



UMAP: Algorithm

Steps 1 and 2:

- **similarity measures** used to capture the relationships between points in the **high-dimensional** and **low-dimensional** spaces.

Goal: UMAP tries to **match** these two graphs as closely as possible, meaning that if two points are close in the high-dimensional space (high similarity), they should also be close in the low-dimensional space (high similarity).

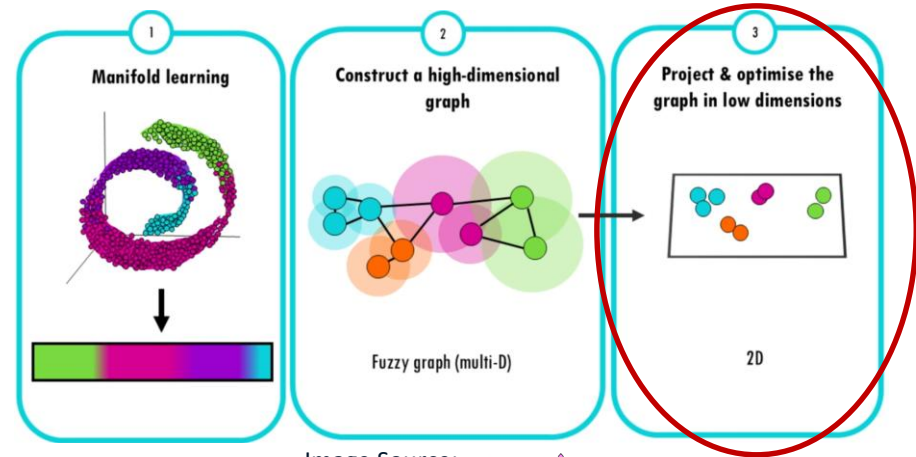


Image Source: [Bioinformatics](#)

Optimization:

Step 3: UMAP minimizes the cross-entropy loss between the graphs:

$$\text{Loss} = \sum_{i \neq j} f_{ij} \log \left(\frac{f_{ij}}{g_{ij}} \right) + (1 - f_{ij}) \log \left(\frac{1 - f_{ij}}{1 - g_{ij}} \right)$$

→ Similarity in high-dimension (points to f_{ij})

→ Similarity in low-dimension (points to g_{ij})

UMAP: High-dimensional Fuzzy Graph Construction

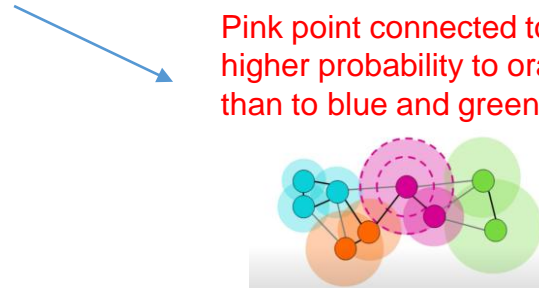
1. Position the data points in a multi-dimensional space.
2. Draw a radius from each point and connect points where the radii overlap.
3. Fuzzy Graph: decreasing probability of two data points being connected as the radius grows.



Radius can vary in size!



Pink point connected to higher probability to orange than to blue and green points.



UMAP: n_neighbours

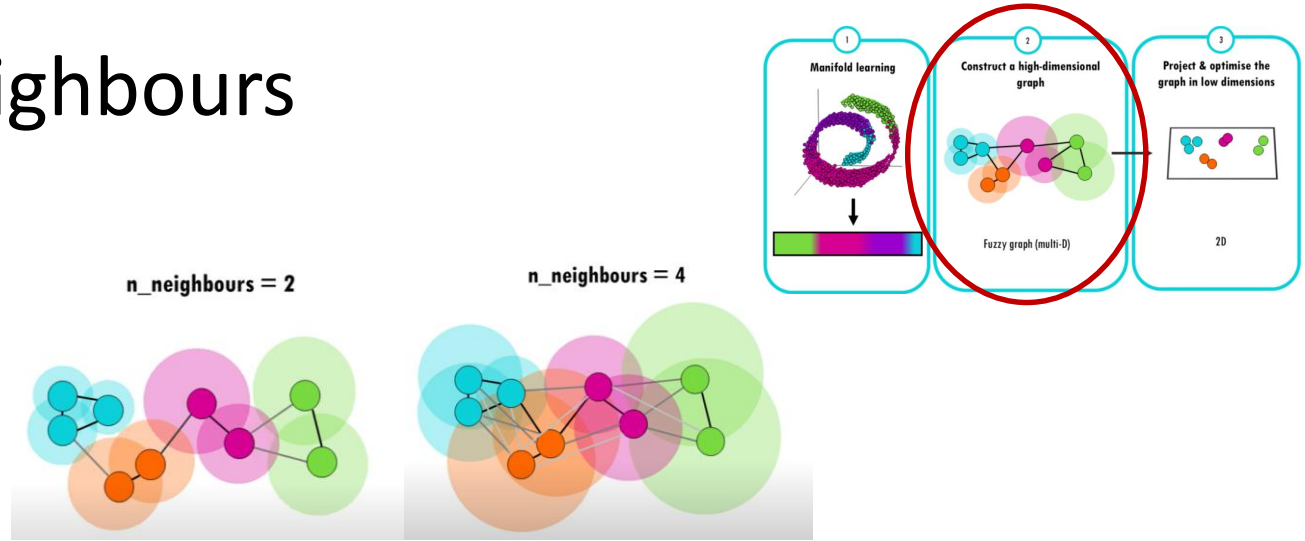


Image Source: [BioStatistics](#)

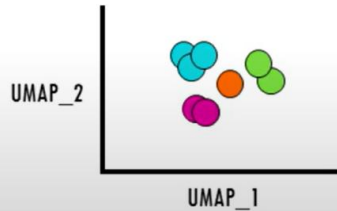
n_neighbours is proportional to the radii around the points.

- Size of radii is based on the distance of the point's nearest end neighbour.
- Points connected to the same number of neighboring points.
- The probability of the connections (in shades of gray) decreases as points farther away get connected.

UMAP: n_neighbours

Low n_neighbours

Focus on local relationships



High n_neighbours

Focus on global relationships

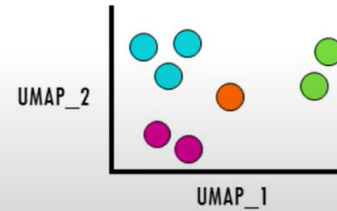
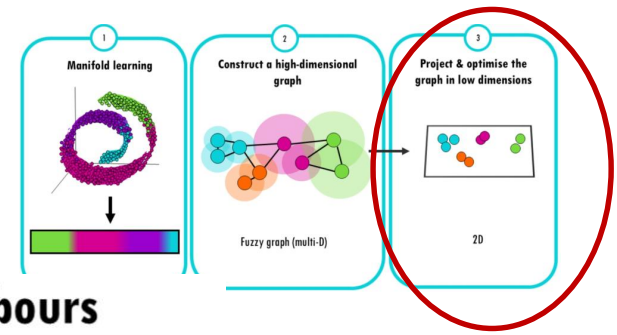


Image Source: 

n_neighbours controls how much the local and global structures are preserved.

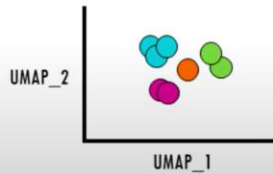
Includes the point itself.

UMAP: min_dist



Low n_neighbours

Focus on local relationships



High n_neighbours

Focus on global relationships

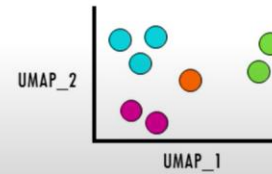


Image Source: 

min_dist is the minimum distance between points in the low-dimensional space.

Related to visualization of relationships which were already computed.

UMAP: hyperparameters

Image adapted from McInnes et al. (2018): UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction

Check out: [Understanding UMAP](#) to play around with parameters

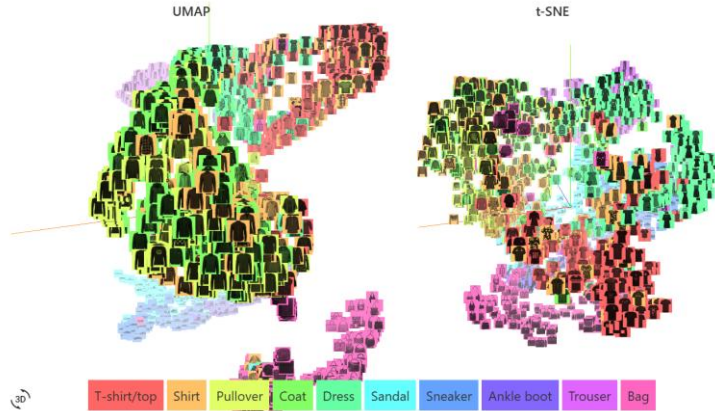
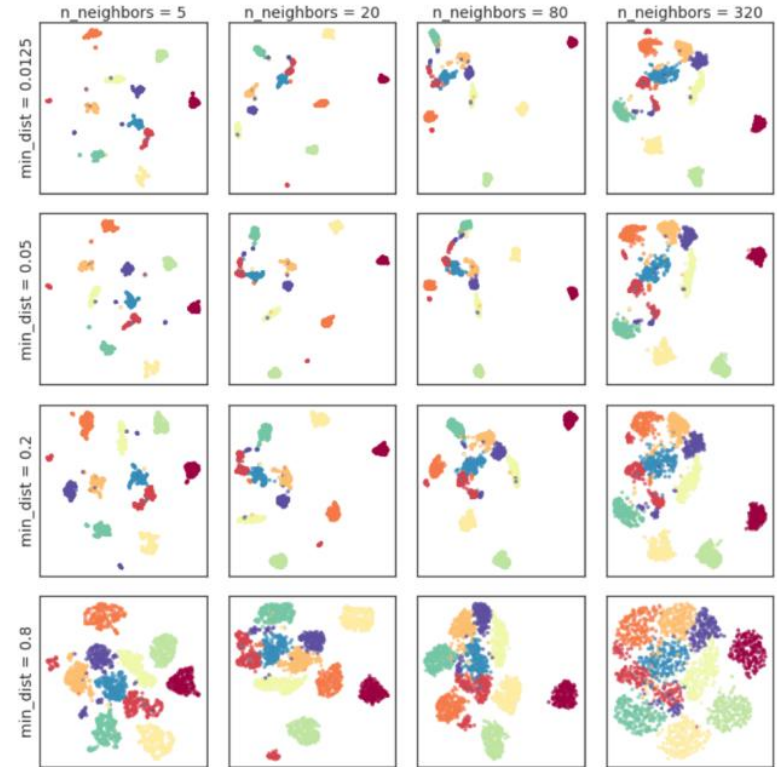


Figure 2: Dimensionality reduction applied to the Fashion MNIST dataset. 28x28 images of clothing items in 10 categories are encoded as 784-dimensional vectors and then projected to 3 using UMAP and t-SNE.



Variation of UMAP hyperparameters n and min-dist result in different embeddings. The data is the PenDigits dataset, where each point is an 8x8 grayscale image of a hand-written digit.

UMAP: Properties

- As a nonlinear dimensionality reduction: lacks the strong interpretability of PCA.
- Preservation of Local and Global structures: UMAP tries to keep clusters of data together, while respecting the overall organization of the data.
- Similarly to t-SNE, UMAP assumes that retaining local geometry is more important than retaining large-scale geometry (global).
- It is known that UMAP is almost entirely driven by the k-neighborhood graph!
- UMAP is not robust for small dataset.

	t-SNE	UMAP
COIL20	20 seconds	7 seconds
MNIST	22 minutes	98 seconds
Fashion MNIST	15 minutes	78 seconds
GoogleNews	4.5 hours	14 minutes

UMAP is much faster
than t-SNE!

Summary: Dimensionality Reduction Methods

	PCA	t-SNE	UMAP
Type	Linear	Non-linear	Non-linear
Focus	Maximize variance (global structure)	Preserving local structure	Preserving local and global relationships
Preserves	Spread of data (variance)	Similarity across neighbors	Shape and clusters (local and global)
Speed	Very Fast	Slow (especially for	Faster than t-SNE
Dimensionality	any	2-3	any
Stochasticity	Deterministic	Stochastic	Stochastic
Interpretability	Easy to interpret	Hard to interpret	More interpretable than t-SNE, but less than PCA
Outputs	Orthogonal axes	Non-linear clusters	Dense, non-linear manifold

When to use what?

PCA works well for linear relationships (preserving variance).

E.g.: Stock price analysis where the relationships between financial metrics are often linear.

t-SNE for clear, detailed cluster separation (local structures).

E.g.: Identifying subtypes in medical diagnostics (e.g., cancer cell types) where clear separation can improve treatment outcomes.

UMAP for large, complex, nonlinear data (typically tens of thousands to millions of samples)

E.g.: Analyzing high-dimensional gene expression or natural language embeddings, where patterns are inherently nonlinear.

Consider the **trade-off** between speed, interpretability, and structure preservation.

Multiple dimensionality reduction methods can be used in combination.

Takeaway

PCA (Principal Component Analysis): Captures **linear** relationships among variables while maximizing **variance**. It is fast and interpretable, but limited to linear patterns.

t-SNE (t-Distributed Stochastic Neighbor Embedding): Focuses on **local** structures and is great for **cluster visualization**. It captures nonlinear relationships, but is computationally intensive and less interpretable.

UMAP (Uniform Manifold Approximation and Projection): Balances **local** and **global** structures. It is very fast, flexible, and effective for large, nonlinear data. UMAP maintains more global structure than t-SNE.

Next up: MDS

Assignment: UMAP

1 Setup

Use the dataset `sklearn.datasets.load_digits` and the Python package `umap-learn`.

2 UMAP

Use UMAP, for different values of the hyperparameter `n_neighbors`, on the digits dataset to project it to 2d.

Using the original labels, compute the silhouette index to choose a value of `n_neighbors`.

3 Interpretation

Verify, using a color-coded 2d plot of the dimensionality-reduced data, whether the clusters make sense. Compute the distances (in the dimensionality-reduced representation) between classes according to

$$\text{dist}(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{x \in C_1} \sum_{y \in C_2} d(x, y).$$

Do these distances agree with your visual impression? Does UMAP retain some of the (large-scale) geometry of ‘digit space’?

References

UMAP Paper (see Moodle)

[UMAP Uniform Manifold Approximation and Projection for Dimension Reduction | SciPy 2018 |](#)

[BioTuring Webinar: A Practical Guide to UMAP by its author John Healy](#)

```
pip install umap-learn  
conda install -c conda-forge umap-learn
```



<https://umap-learn.readthedocs.io> (Also includes many examples!)