Week 6 Topic 4:

Parameter Tuning in UMAP

(**U**niform **M**anifold **A**pproximation and **P**rojection)

Saygin Ileri, Bjørnar Ørjansen Kaarevik, Tortein, Nordgård-Hansen,
Andreas Raja Goklas Sitorus

2.10.2024

what is UMAP? why use it?

- a dimension **reduction** technique in **ML.** (recent development, 2018)
- can be used for **visualization** for **complex** (high-dimensional) datasets. **Gain** more **insight!**
- similar theory and usage to **t-SNE**.
- competitive with t-SNE for visualization **quality** and **preservation** of global **structure**, much **faster** algorithm.

why use UMAP?

- captures both **local** and **global nonlinear** relationships
(best for data with intricate patterns)

- better preserves the **local** structure
(important for **clustering** tasks)

- **adaptive** parameterization
adjust the trade-off (**local/global** structure)

- better handles with **noise**

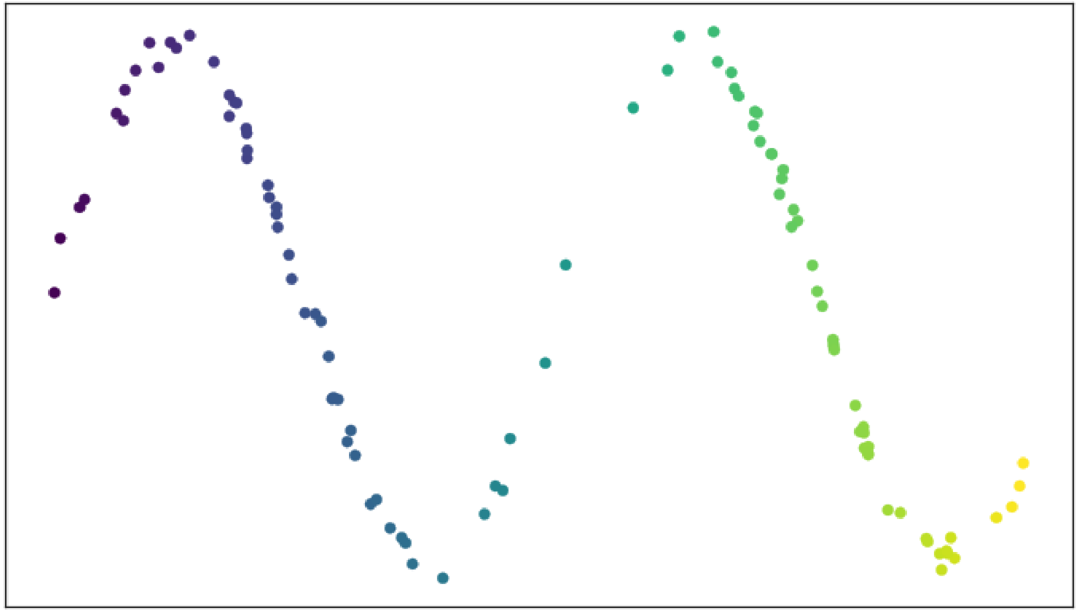
- **faster**: 784 ->3 dimension in 4 mins (t-SNE: 27mins)

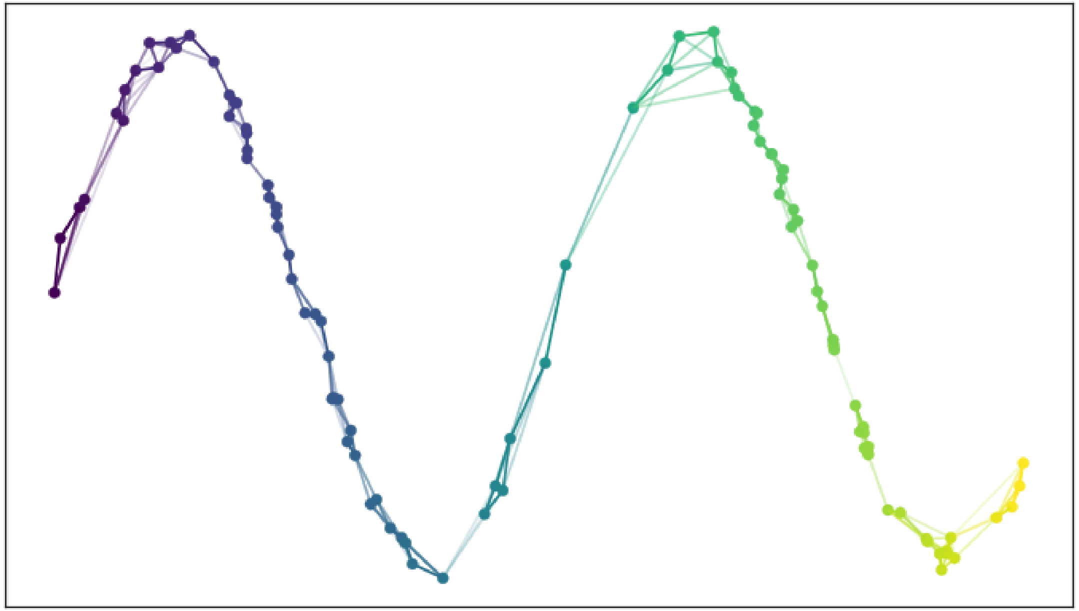
The Essence of UMAP

"**Preserve** the **topological data** embedded in the fuzzy simplicial set defined by the pseudo-metrics **inherent to the dataset**"

UMAP is trying to learn the manifold where the data comes from

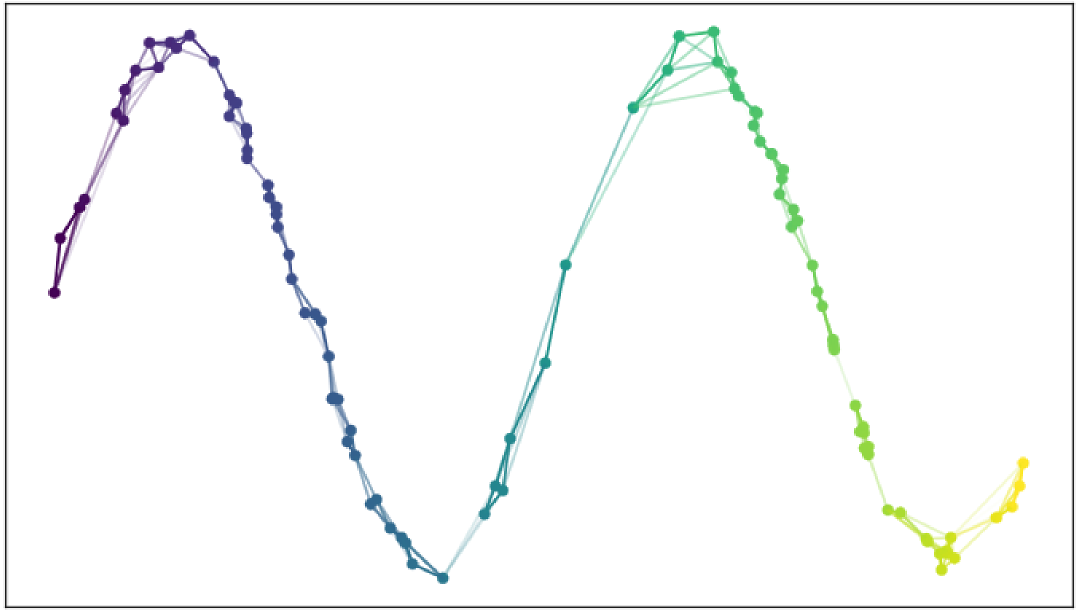
Assumption: **Uniform** distribution on the **manifold** with respect to the "appropriate" **metric**





Locally, the "appropriate" metric can be swapped with the metric of the ambient space

- Each datapoint gets its own *pseudo*-metric
- Each metric defines a topology (notion of closeness)
- These topologies are all incompatible
- UMAP combines the incompatible topologies into a single topological structure



Actually, we work with a weighted graph

The UMAP weighted graph

- Each vertex has k neighbours
- Each edge is a fuzzy connection
- The weight is the fuzzy membership strength
- This means "how likely is the connection to be real"

General graph algorithms

1. **Construction:**
   Construct a weighted graph
2. **Visualization:**
   Compute layout of the graph

Hyperparameters during construction and visualization can be tuned

Construction

- Number of neighbours
    - *Local/global tradeoff*
- Metric (distance function)

Visualization

- Target dimension
- Minimum distance
    - *Local/global tradeoff*
- (Number of training epochs)

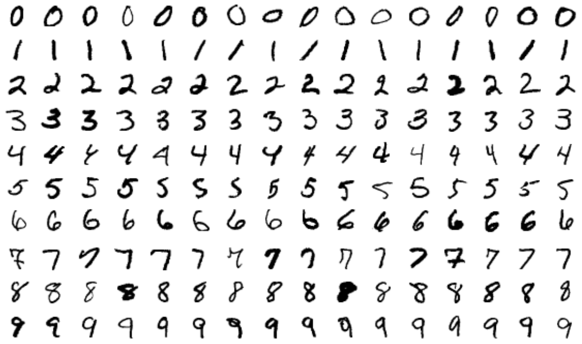
Implementation and Results

Objective:

- Reduce the dimension of the dataset and visualize it into 2D and 3D graph
- Visualize the effect of the hyperparameter change to the dimension reduction results

Dataset used for the code implementation:

- MNIST dataset
    - 28 x 28 pixels images, flattened into a vector of length **784 features**
    - Labels: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9



UMAP for Supervised Dimension Reduction and Metric Learning

- Python library
- Source: https://umap-learn.readthedocs.io/en/latest/supervised.html#umap-on-fashion-mnist

Dataset used for the code implementation:

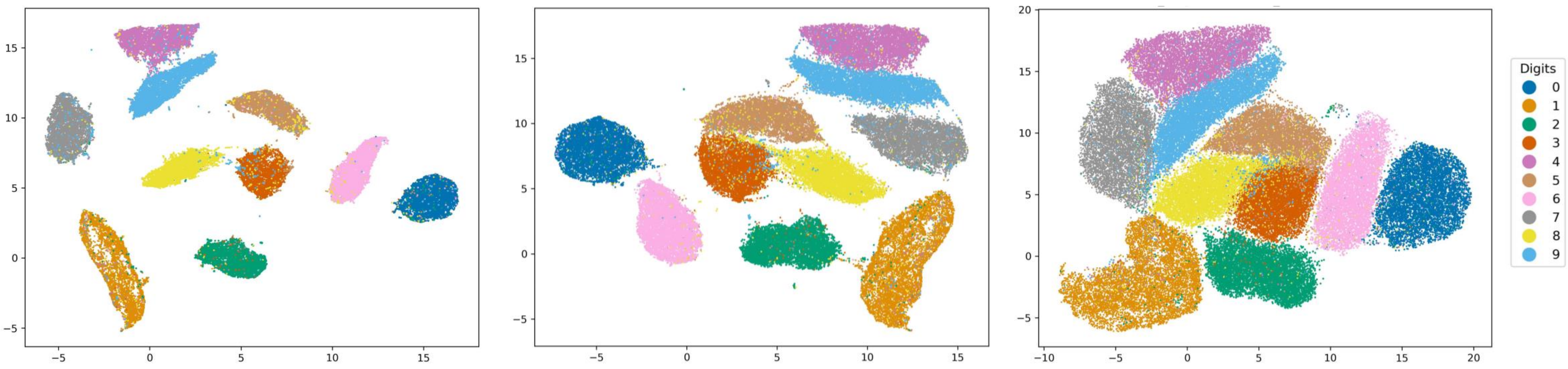
- **`n_neighbors`**: Number of nearest neighbors to consider, balancing local vs. global structure.
- **`min_dist`**: Minimum distance between points in reduced space, controlling point spread.
- **`n_components`**: Number of output dimensions (e.g., 2D or 3D).
- **`metric`**: Distance metric for computing point similarities (e.g., `"euclidean"`).

UNIFORM MANIFOLD
UMAP
APPROXIMATION & PROJECTION

```
umap.UMAP(n_neighbors=n_neighbors, min_dist=min_dist, n_components=dim, metric=metric).fit_transform(dataset.data)
```

Effect on changing the `min_dist` – 2D Graph

Same for all plots `n_neighbors = 5, metric = 'Euclidean', n_components = 2`

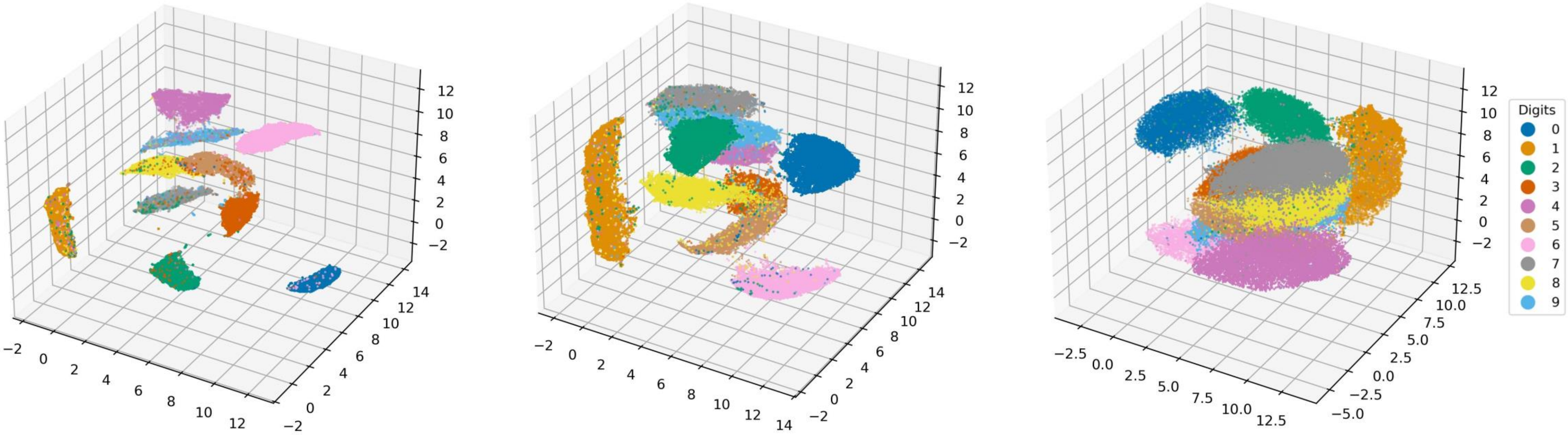


(a) (b) (c)

Hyperparameter for Figure: (a) `min_dist = 5,` (b) `min_dist = 0.25,` (c) `min_dist = 0.6`

Effect on changing the `min_dist` – 3D Graph

Same for all plots `n_neighbors = 5, metric = 'Euclidean', n_components = 3`

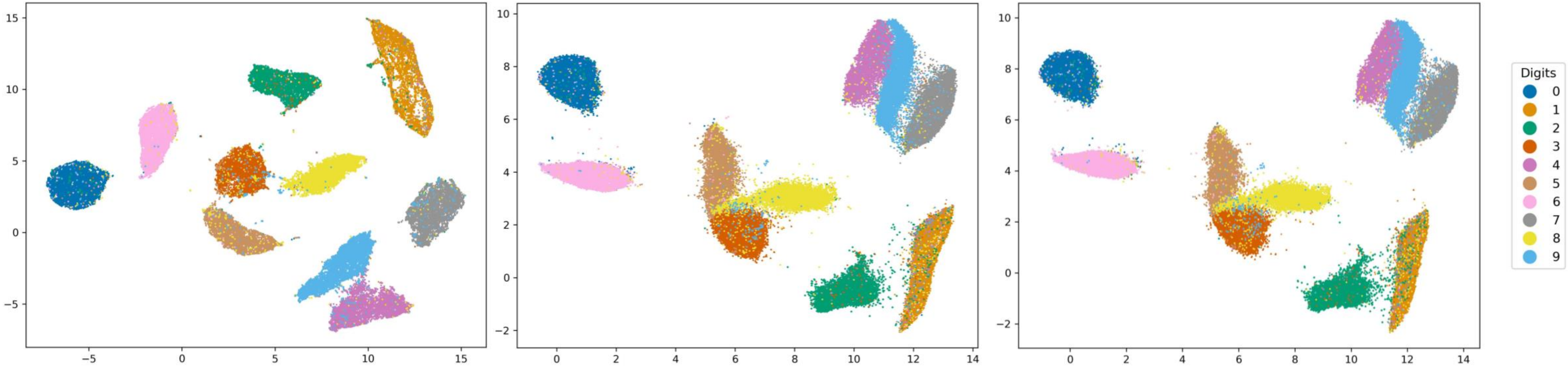


(a) (b) (c)

Hyperparameter for Figure: (a) `min_dist = 5,` (b) `min_dist = 0.25,` (c) `min_dist = 0.6`

Effect on changing the `n_neighbors` – 2D Graph

Same for all plots `min_dist = 0.01, metric = 'Euclidean', n_components = 2`

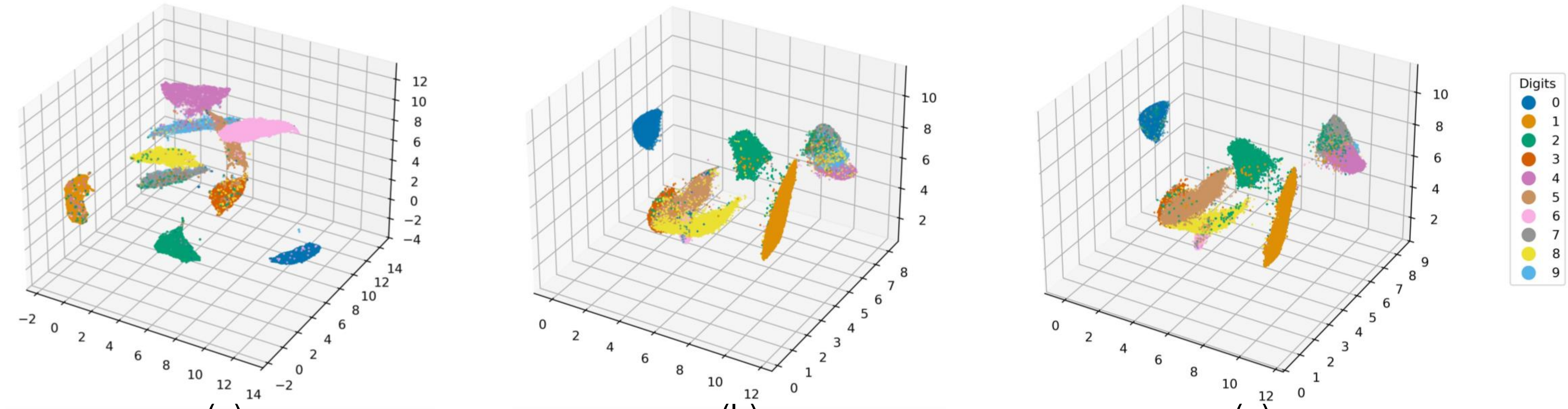


(a) (b) (c)

Hyperparameter for Figure: (a) `n_neighbors = 5,` (b) `n_neighbors = 500,` (c) `n_neighbors = 1000`

Effect on changing the `min_dist` – 3D Graph

Same for all plots `min_dist = 0.25, metric = 'Euclidean', n_components = 3`



(a) (b) (c)

Hyperparameter for Figure: (a) `n_neighbors = 5,` (b) `n_neighbors = 50,` (c) `n_neighbors = 100`

Conclusion on hyperparameters tuning

- **`n_neighbors`**:
  Smaller values (e.g., 5-15) focus on local structure; larger values (e.g., 50+) emphasize global patterns. Start with 15 for balanced results.
- **`min_dist`**:
  Use lower values (e.g., 0.001-0.1) for tight clustering and detail retention. Higher values (e.g., 0.3-0.8) create more spread-out, general patterns.
- **`n_components`**:
  Set based on needs—2D or 3D for visualization, higher for feature preservation.
- **`metric`**:
  Choose based on data type; `"euclidean"` works well for continuous data, while `"cosine"` or `"correlation"` are better for text or sparse data. Experiment with different metrics to see what fits best.

Conclusion on UMAP

- **UMAP effectively balances local and global data structures**, making it a powerful tool for dimensionality reduction and visualization, especially for complex datasets.
- **Flexible with customizable parameters**, UMAP can adapt to various data types and structures, offering both detailed clustering and broader pattern discovery based on the chosen hyperparameters.

references

- https://arxiv.org/abs/1802.03426
- https://medium.com/@aeonaten/understanding-umap-uniform-manifold-approximation-and-projection-cede51c477d9
- https://pair-code.github.io/understanding-umap/
- https://lvdmaaten.github.io/tsne/
- https://umap-learn.readthedocs.io/en/latest/parameters.html
- https://umap-learn.readthedocs.io/en/latest/how_umap_works.html