UMAP + Graph Optimization

Sofia Escalante Escobar

Universidad Nacional de Colombia
Signal Processing and Recognition Group - SPRG

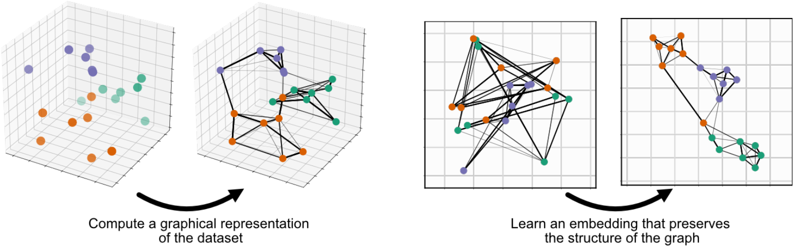
October 8, 2024

Outline

UMAP

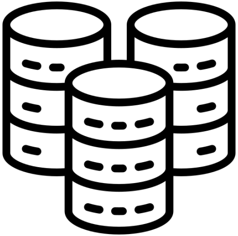
UMAP is a nonparametric, graph-based dimensionality reduction algorithm that uses Riemannian geometry and algebraic topology to embed structured data in low dimensions [Sainburg et al., 2021] [Yi et al., 2024]

$$\mathcal{L}(Y) = \sum_{(i,j)\in E} \left[w_{ij} \log\left(\frac{f(y_i, y_j)}{w_{ij}}\right) + (1 - w_{ij}) \log\left(\frac{1 - f(y_i, y_j)}{1 - w_{ij}}\right)\right] \tag{1}$$



Outline

Motivation I



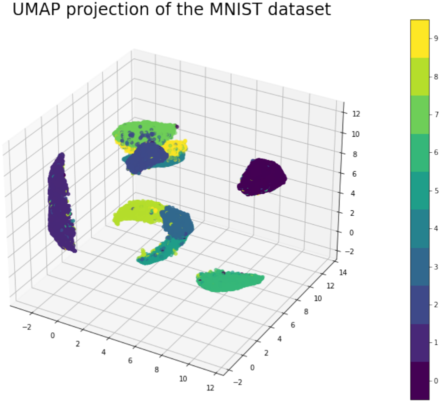
In modern data science and machine learning, datasets often contain hundreds or even thousands of features,this high dimensionality can lead to challenges in:

- Computation
- Storage
- Model interpretability

[Ghojogh et al., 2023] [Geron, 2019]

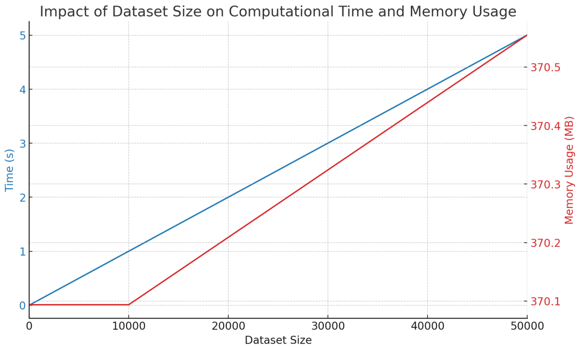
Motivation II

As datasets grow, dimensionality reduction becomes crucial, UMAP effectively preserves local and global structures, but large datasets demand more scalable and efficient graph-based methods. [Aggarwal, 2020] [Yi et al., 2024]



Outline

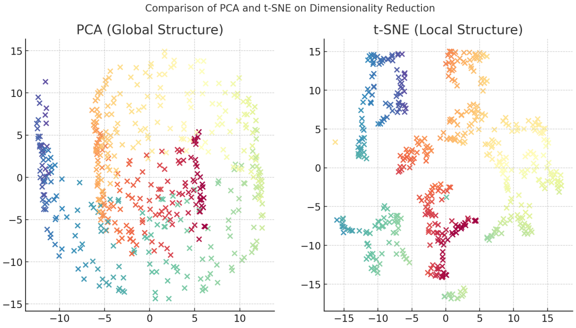
Challenges in Dimensionality Reduction



- **High-dimensional data**: As the dimensionality of data increases, many traditional techniques struggle with scalability.
- **Graph construction**: Techniques like UMAP rely on graph-based methods, where constructing and optimizing the graph for large datasets is computationally expensive.

[Ghojogh et al., 2023] [Allaoui et al., 2020]

Flexibility in Dimensionality Reduction



- **Static graph structures**: Many dimensionality reduction techniques create a fixed graph, making it challenging to incorporate new information without full re-computation.
- **Dynamic updates**: Ensuring that dimensionality reduction techniques are adaptable to new data without losing important structural information is a significant challenge.

[Wang et al., 2021]

Outline

Graphs

A graph G is defined as:

$$G = (V, E)$$

where V = vertices (nodes), $E \subseteq V \times V$ = set of edges.
The adjacency matrix A of a graph G with vertices $v_1, v_2, \ldots, v_n$ is a $|V| \times |V|$ matrix where:

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between } v_i \text{ and } v_j \\ 0 & \text{otherwise} \end{cases}$$

Alternatively, the graph can be represented by an edge list:

$$E = \{(v_i, v_j) | \text{edge exists between } v_i \text{ and } v_j\}$$

[Hakim, 2023]

Parametric UMAP

Parametric UMAP replaces the non-parametric embedding with a neural network, which takes high-dimensional input data and produces a low-dimensional embedding. [Sainburg et al., 2021]

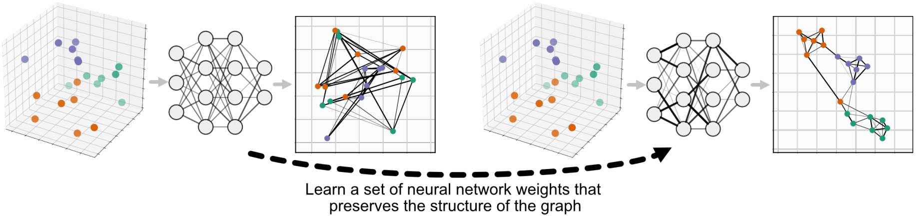
$$f(x) \to z$$

$$x \in \mathbb{R}^{N\times p}(\text{input}) \to z \in \mathbb{R}^{N\times d} \quad d \ll p \quad (\text{output})$$

$$\text{Model:} \quad z = (f_L \circ f_{L-1} \circ \ldots \circ f_L)\, x$$

$$z_L = f_L(z_{L-1}) = \phi(z_{L-1} \odot W_L + b_L)$$

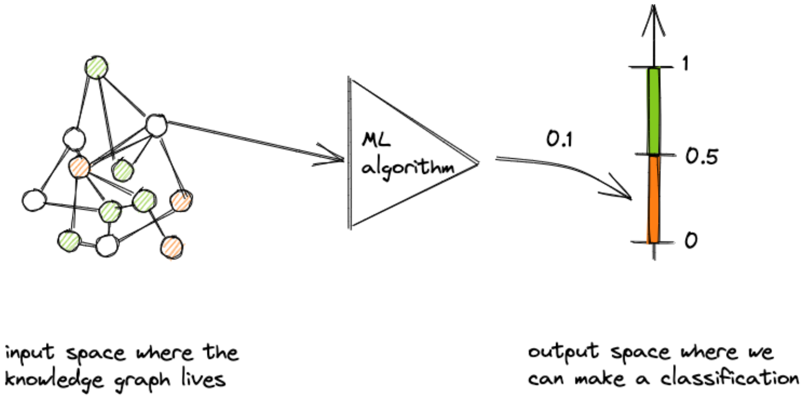
$$\phi(\cdot) = \text{función de activación}$$



Outline

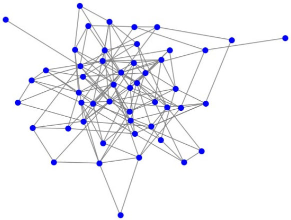
Alternative Graph Constructions

An **alternative graph construction** is proposed to enhance **computational times and cost** while preserving the **data structures**.



Igraph I

Erdős-Rényi Random Graph



In igraph, the Erdős-Rényi random graph model a graph is generated by connecting each pair of N vertices with a probability p:

$$P(G) = p^{|E|}(1-p)^{\frac{|V|(|V|-1)}{2}-|E|}$$

[Hakim, 2023] [McInnes et al., 2018]

NetworkX for UMAP Graph Construction

NetworkX provides flexible graph operations ideal for UMAP. It supports dynamic graph updates with easy node and edge manipulation:

$$G = (V, E)$$

where V are vertices and E are edges. NetworkX efficiently computes shortest paths, clustering, and more using adjacency matrices A:

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

This enables scalable graph-based dimensionality reduction.

Outline

Results

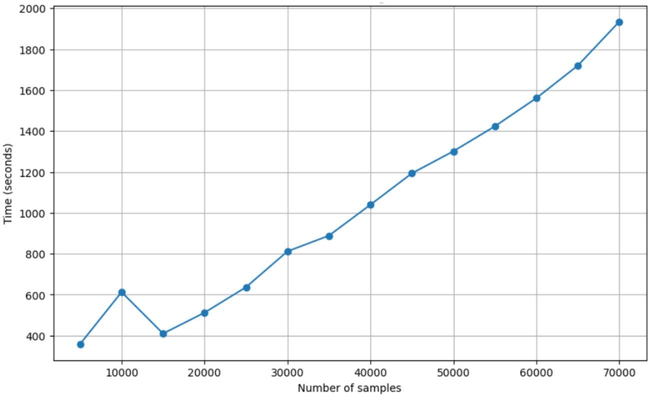
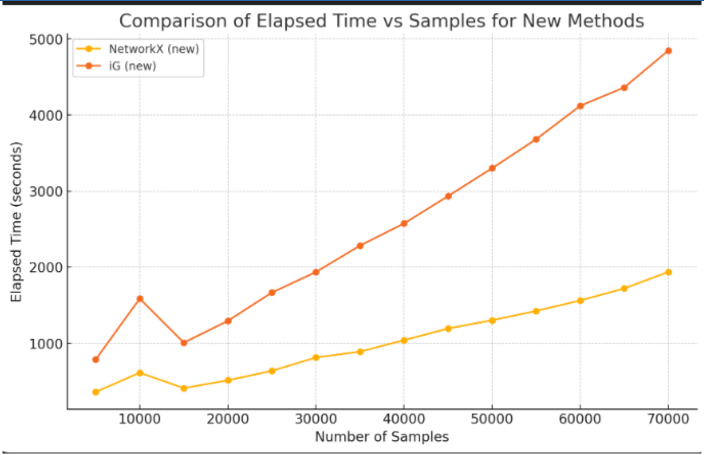
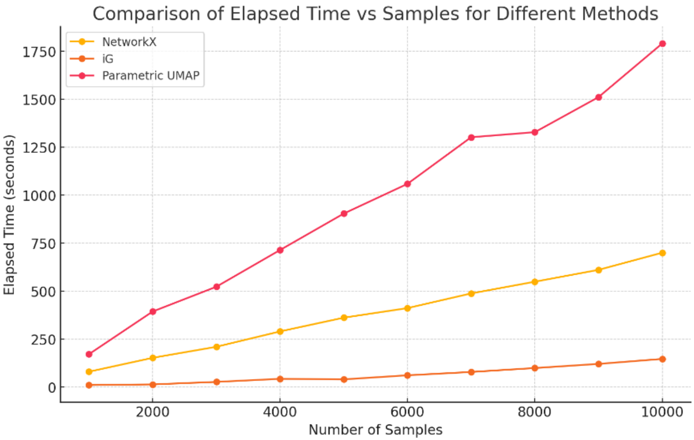


Figure: Parametric UMAP using NetworkX Graph Construction


Comparison of Elapsed Time vs Samples for New Methods
NetworkX (new)
iG (new)
Elapsed Time (seconds)
5000
4000
3000
2000
1000
10000
20000
30000
40000
50000
60000
70000
Number of Samples


Comparison of Elapsed Time vs Samples for Different Methods
NetworkX
iG
Parametric UMAP
Elapsed Time (seconds)
0
250
500
750
1000
1250
1500
1750
2000
4000
6000
8000
10000
Number of Samples

Scalability Metrics Used

Key metrics for evaluating scalability:

Time Complexity: Measures how runtime grows with data size, aiming for efficient scaling ($O(n \log n)$).

Speedup: Assesses performance gain with added resources:

$$\text{Speedup} = \frac{\text{Single-core time}}{\text{Multi-core time}}$$

Scalability Testing: Gradually increases data size to identify bottlenecks and test large-scale performance.

Future Work: Improving UMAP Scalability

To enhance UMAP's scalability for large datasets:

- **FAISS**: Efficient k-nearest neighbor search, significantly speeding up graph construction for large-scale data.
- **HNSW**: Builds scalable, multi-layered graphs with efficient dynamic updates, ideal for streaming data.

These methods will be tested to reduce computational load and memory usage while improving scalability.

Outline

References I

Aggarwal, C. C. (2020).
Machine Learning for Data Science: Foundations, Techniques, and Applications.
Springer.

Aggarwal, C. C. and Reddy, C. K. (2021).
Data Clustering: Algorithms and Applications.
CRC Press.

Allaoui, M., Kherfi, M. L., and Cheriet, A. (2020).
Considerably improving clustering algorithms using umap dimensionality reduction technique: A comparative study.
In El Moataz, A., Mammass, D., Mansouri, A., and Nouboud, F., editors, *Image and Signal Processing*, pages 317–325, Cham. Springer International Publishing.

Geron, A. (2019).
Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.
O'Reilly Media.

Ghojogh, B., Crowley, M., Karray, F., and Ghodsi, A. (2023).
Uniform Manifold Approximation and Projection (UMAP), pages 479–497.
Springer International Publishing, Cham.

Hakim, M. A. (2023).
What is the best python graph tool? graph-tool vs networkx.
Accessed: 2024-09-05.

References II

McInnes, L., Healy, J., and Melville, J. (2018).
Umap: Uniform manifold approximation and projection for dimension reduction.
arXiv preprint arXiv:1802.03426.

Sainburg, T., McInnes, L., and Gentner, T. Q. (2021).
Parametric umap: learning embeddings with deep neural networks for representation and semi-supervised learning.

Wang, Y., Huang, H., Rudin, C., and Shaposhnik, Y. (2021).
Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization.
Journal of Machine Learning Research, 22(201):1–73.

Yi, W., Bu, S., Lee, H.-H., and Chan, C.-H. (2024).
Comparative analysis of manifold learning-based dimension reduction methods: A mathematical perspective.
Mathematics, 12(15).