UMAP in context of other dim. reduction methods

Anh Vu Le

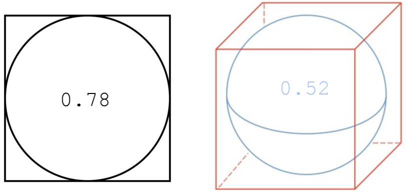
Outline

1. Introduction
2. PCA
3. Methods related to UMAP
    a. ISOMAP
    b. t-SNE
4. Taxonomy of dimensionality reduction methods
5. UMAP

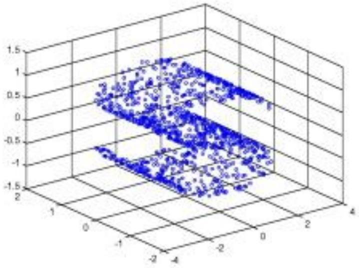
What is dimensionality reduction about?

*“ ...transformation of **high-dimensional data** into a **meaningful representation of reduced dimensionality”***

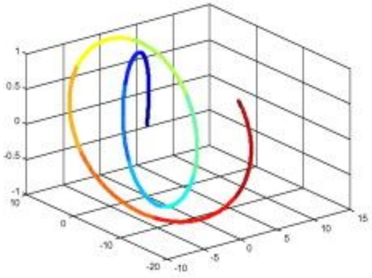
- **intrinsic** dimensions of data (~ **manifold**)
  - smaller dimension, own geometry
- **mitigates the curse of dimensionality**



- **classification, visualization,** and **compression**



(d)



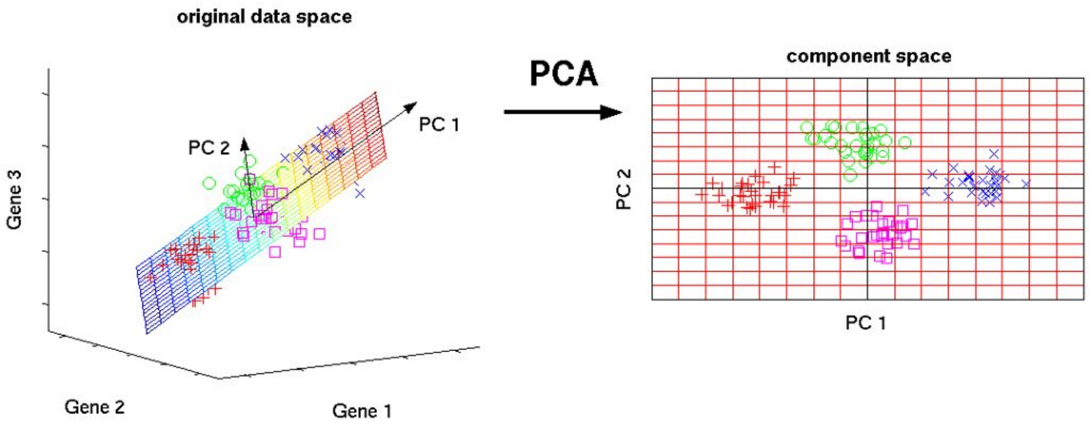
(g)

PCA - the queen of dimensionality reduction

*“PCA transforms (possibly correlated) data **linearly into new properties** that are **not correlated**”*

Properties of transformation:

- **Preserves variance**
  - Axis are directions of greatest variance
- properties are **uncorrelated**



The elegance of PCA

"PCA transforms *possibly correlated* data **linearly into new properties** that are **not correlated**"

$$cov(\mathbf{X}) = \mathbf{C_X} = \mathbf{X}^T\mathbf{X} \text{ is a covariance matrix}$$

We want to find the projection $\mathbf{P}$ to uncorralated space, s.t.:
$\mathbf{XP} = \mathbf{T}$
where the uncorrelateness is expressed in the fact, that $\mathbf{C_T}$ is **diagonal**

The elegance of PCA

PCA transforms *possibly correlated* **data linearly into new properties** that are **not correlated** with each other

$$cov(\mathbf{X}) = \mathbf{C_X} = \mathbf{X}^T\mathbf{X} \text{ is a covariance matrix}$$

We want to find the projection $\mathbf{P}$ to uncorralated space, s.t.:

$\mathbf{XP} = \mathbf{T}$

where the uncorrelateness is expressed in the fact, that $\mathbf{C_T}$ is **diagonal**

The covariance matrix of $\mathbf{T}$ can be expressed as:

$$\boxed{\mathbf{C_T}} = \mathbf{T}^T\mathbf{T} = (\mathbf{XP})^T\mathbf{XP} = \mathbf{P}^T(\mathbf{X}^T\mathbf{X})\mathbf{P} = \boxed{\mathbf{P}^T\mathbf{C_X}\mathbf{P}}$$

We know, that $\mathbf{C_X}$ is a **symetric** matrix.

The fine property of symetrix matrices, that their eigenvectors are **orthogonal** and:

$\boxed{\mathbf{C_X} = \mathbf{EDE}^T}$ where D is diagonal matrix

Now the magic: what if $\mathbf{P}$ is said to be $\mathbf{E}$?

The elegance of PCA

$$\mathbf{C_T} = \mathbf{P}^T \mathbf{C_X} \mathbf{P}$$
$$\mathbf{C_X} = \mathbf{E}\mathbf{D}\mathbf{E}^T$$

If $\mathbf{P}$ is said to be $\mathbf{E}$, then

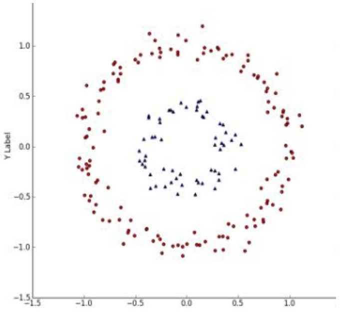
$$\mathbf{C_T} = \mathbf{P}^T \mathbf{C_X} \mathbf{P} = \mathbf{P}^T(\mathbf{E}\mathbf{D}\mathbf{E}^T)\mathbf{P} = \mathbf{P}^T(\mathbf{P}\mathbf{D}\mathbf{P}^T)\mathbf{P} = \mathbf{D}$$

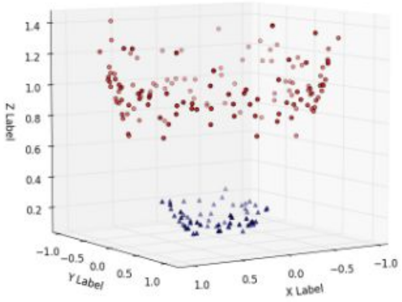
Choosing the **projection matrix** to be **eigenvectors** of the **covariance matrix of *X*** gives us the projection we want:

- uncorrelated properties (diagonal C_T)
- no loss of information (P is orthogonal)

Kernel PCA

- Some data linearly inseparable





$$cov(\mathbf{X}) \rightarrow K(\mathbf{X})$$
$$k_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

Transformation matrix is again the eigenmatrix of K

- Vapnik–Chervonenkis theory - projection into a **higher dimensional space** may provide us with **better classification power**.

- **Kernel trick**, a method to project original data into higher dimension **without sacrificing too much computational time**

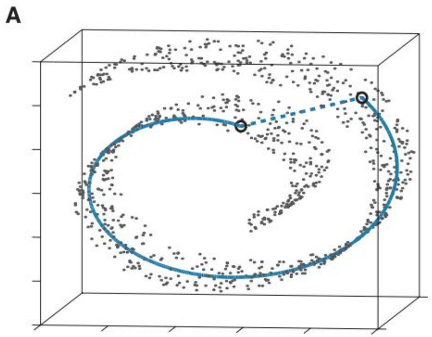
Focusing on local patches of manifold

ISOMAP

- Close points in original space don’t need to be close on manifold

Steps:

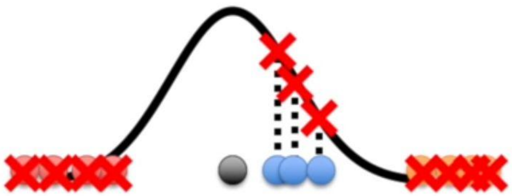
1. Construct a **neighbourhood** graph (B)
2. Compute **shortest paths** between points in the graph (B)
3. Embed points with knowledge shortest paths in **lower dimension**



Focusing on local patches of manifold (**t-SNE**)

<u>**Key koncept:**</u> t-SNE maps **distances** to **probabilities**

Each point *in original space forms a* ***<u>Gaussian</u>*** *around itself*

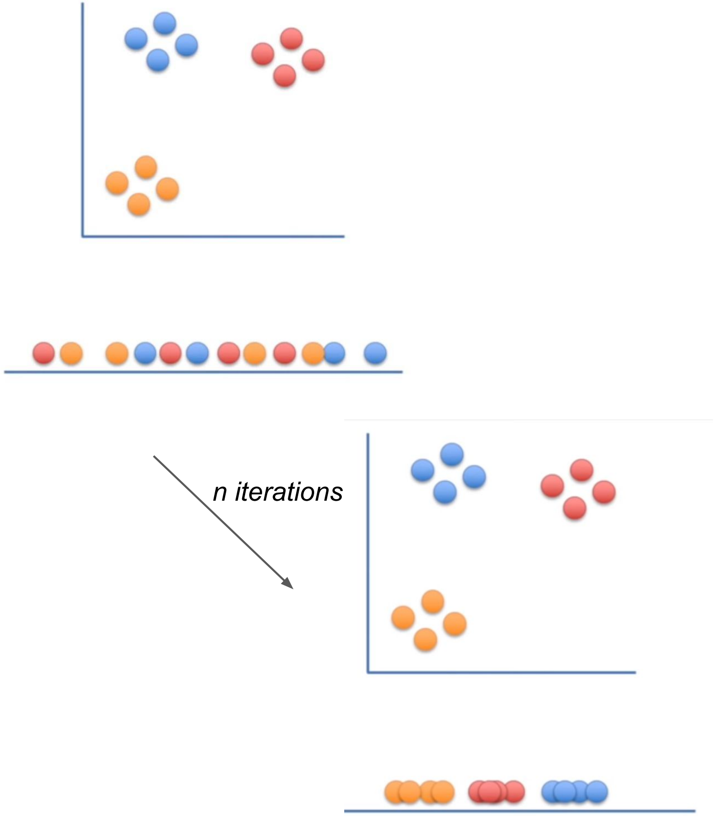


On **lower dimension,** place points **randomly** at first

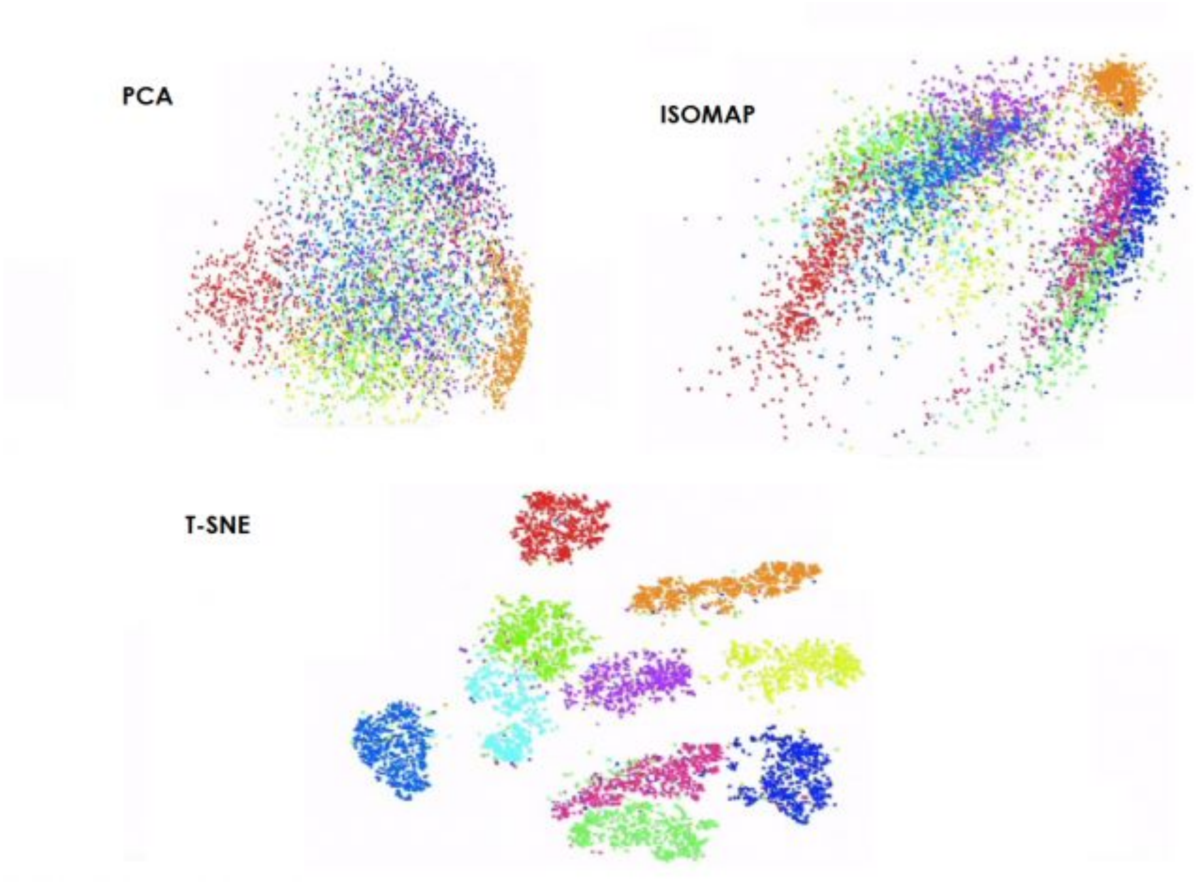
Each point *in lower dim. forms a* ***<u>t-distribution</u>*** *around itself*

Minimize entropy: $$C = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

p_{ij}...probability distance in original space
q_{ij}...probability distance in reduced space

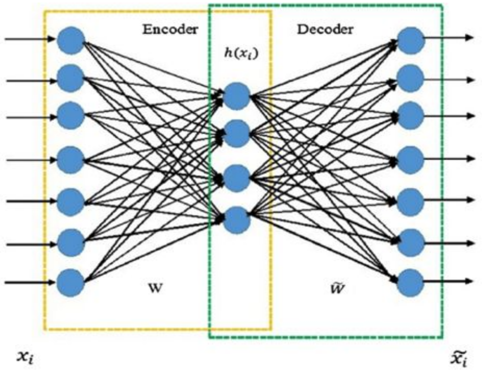


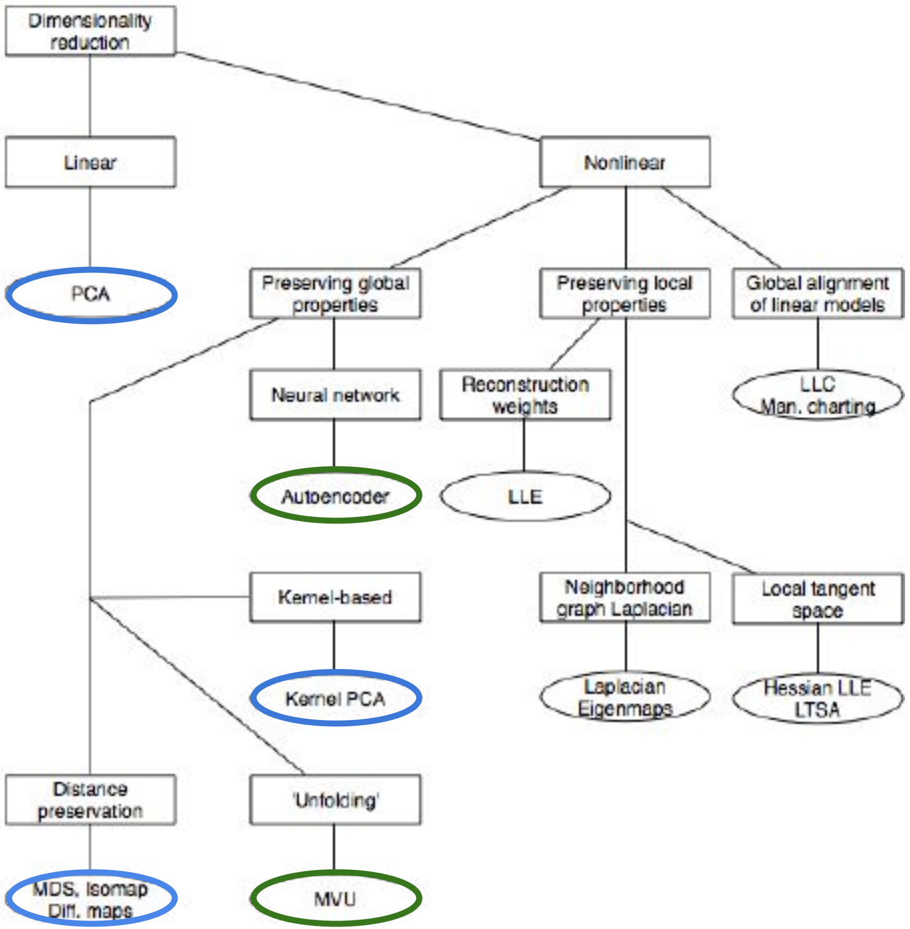
t-SNE in practice



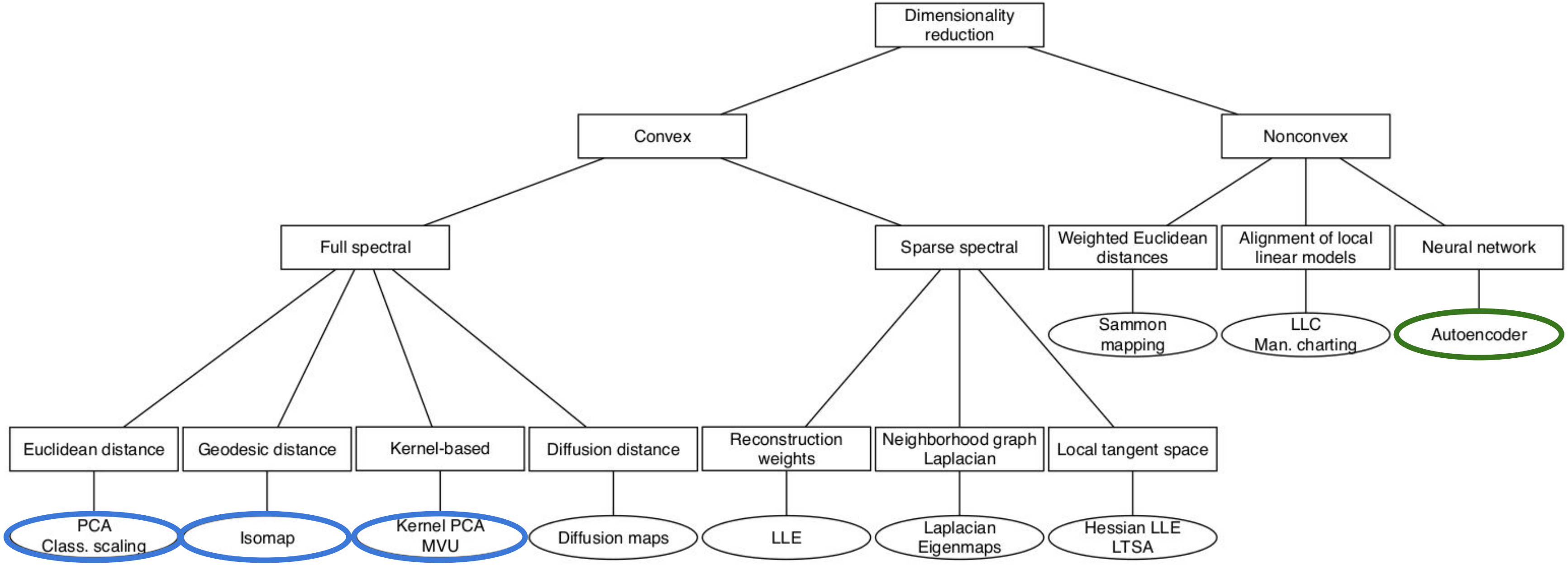
Taxonomy of techniques

- **MDS** (Multidimensional scaling)
    - “Fabricating” coordinate system based on similarity/distances
    - 3rd step of ISOMAP
- **MVU** (Maximum Variance Unfolding)
    - Learning kernel function for kernel PCA
- **Autoencoder**

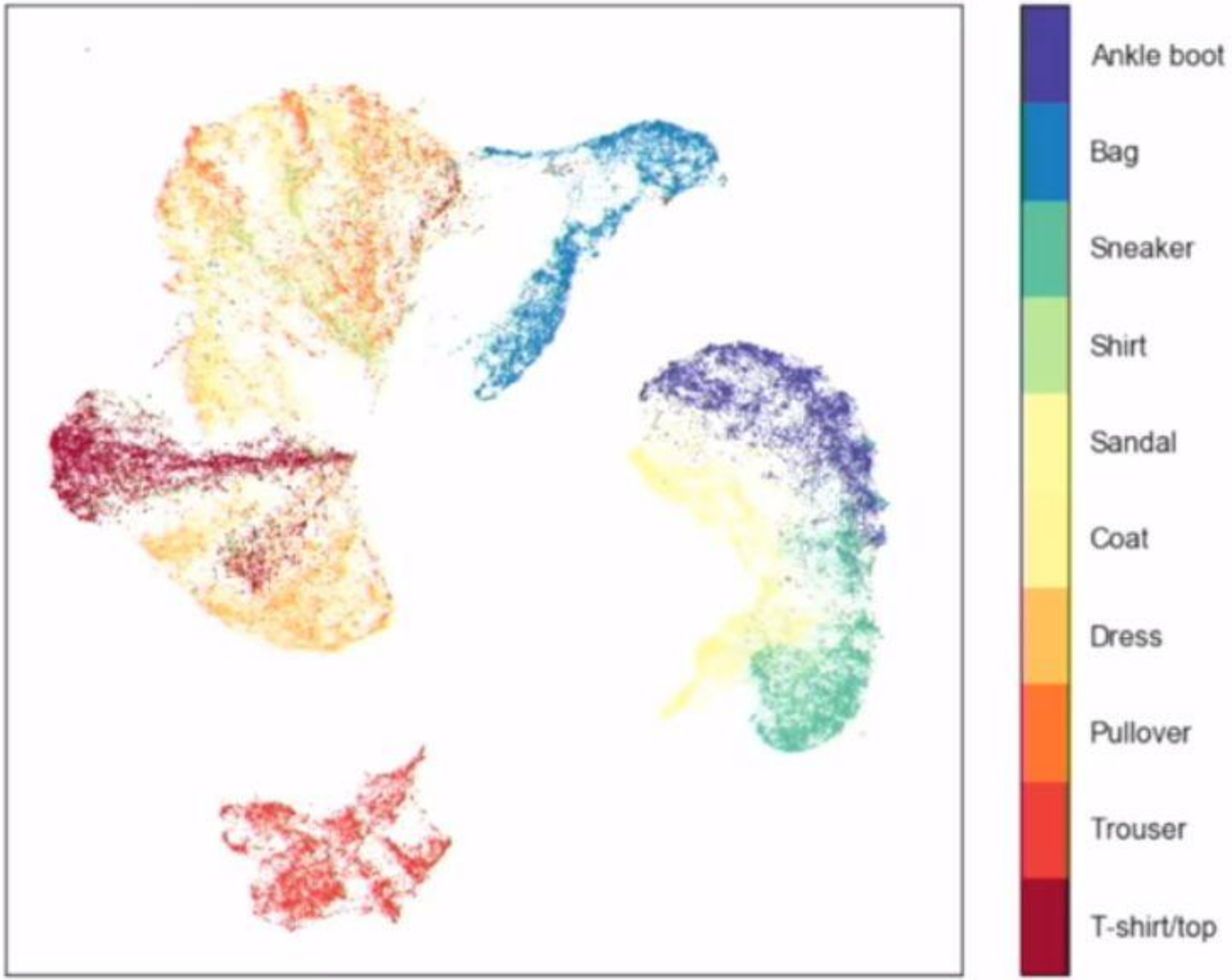




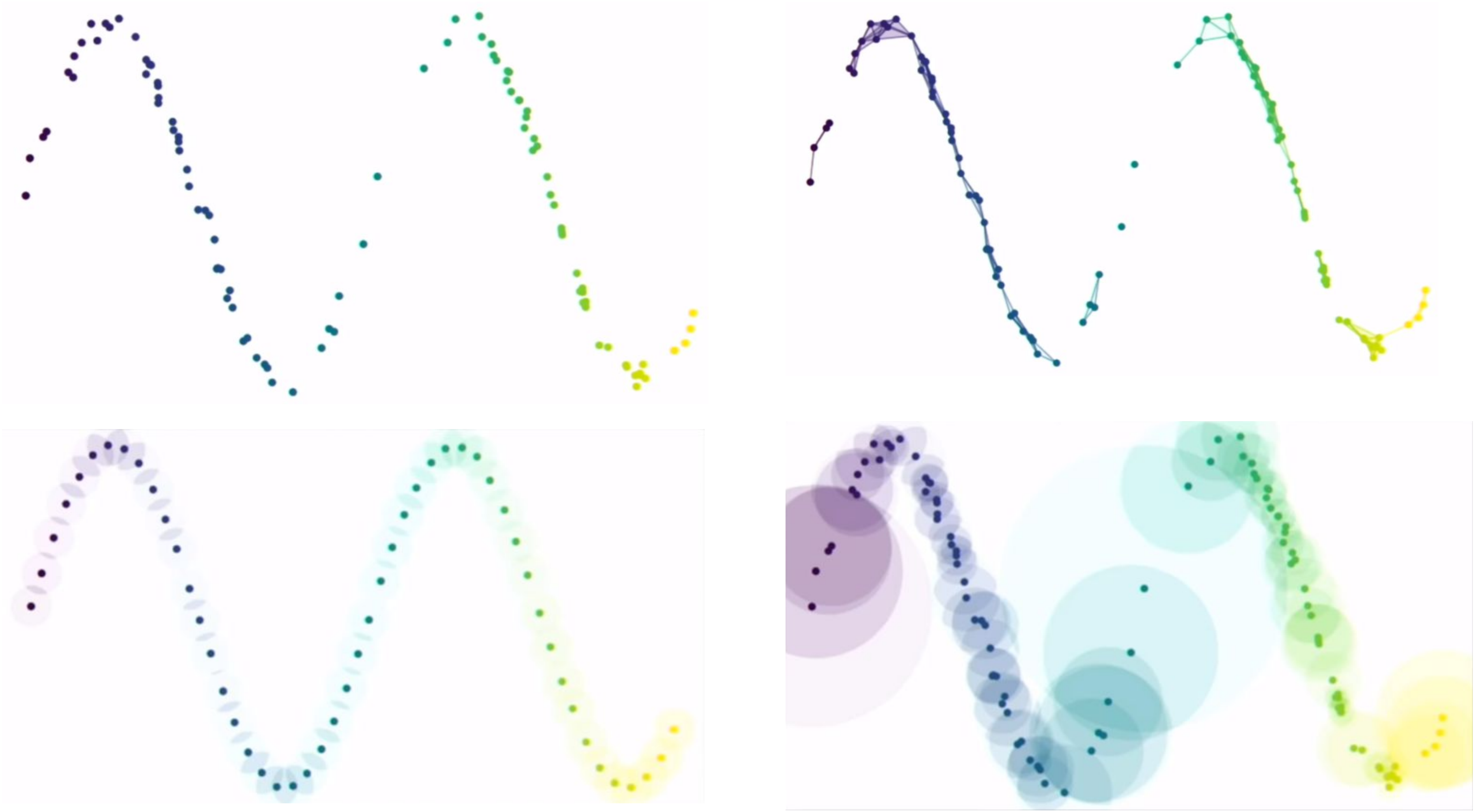
Taxonomy of techniques



UMAP


Ankle boot
Bag
Sneaker
Shirt
Sandal
Coat
Dress
Pullover
Trouser
T-shirt/top

	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

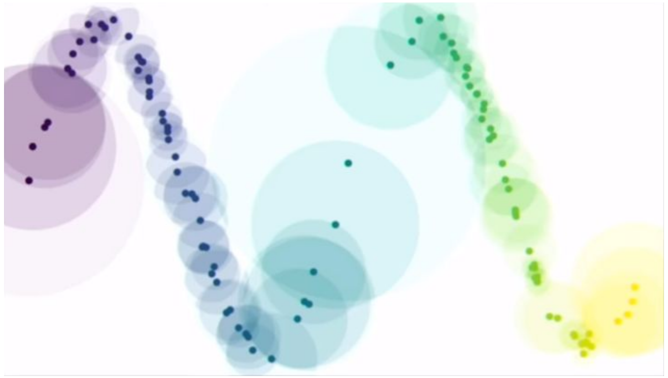


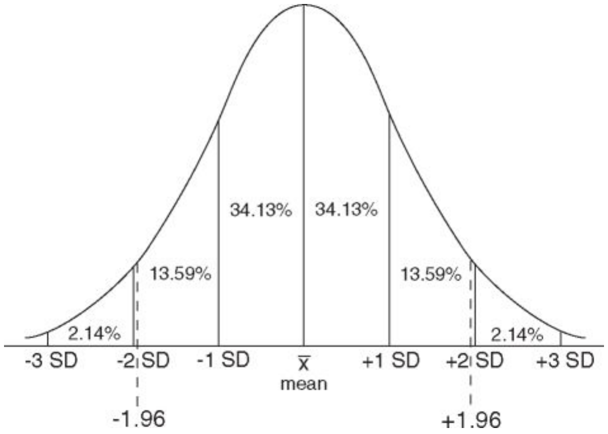
Algorithm 2 Constructing a local fuzzy simplicial set

function LocalFuzzySimplicialSet(X, x, n)
  knn, knn-dists ← ApproxNearestNeighbors(X, x, n)
  ρ ← knn-dists[1] ▷ Distance to nearest neighbor
  σ ← SmoothKNNDist(knn-dists, n, ρ) ▷ Smooth approximator to knn-distance
  fs-set$_0$ ← X
  fs-set$_1$ ← $\{([x, y], 0) \mid y \in X\}$
  for all $y \in$ knn **do**
    $d_{x,y} \leftarrow \max\{0, \text{dist}(x, y) - \rho\}/\sigma$
    fs-set$_1$ ← fs-set$_1$ $\cup$ $([x, y], \exp(-d_{x,y}))$
  return fs-set

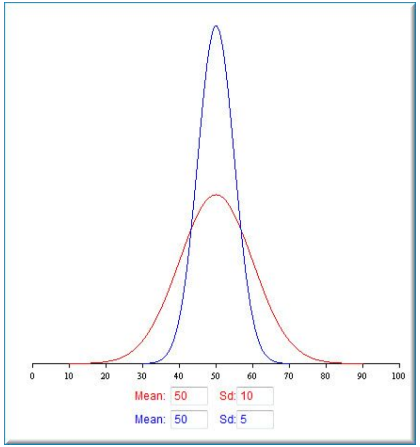
Algorithm 3 Compute the normalizing factor for distances σ

function SmoothKNNDist(knn-dists, n, ρ)
  Binary search for σ such that $\sum_{i=1}^{n} \exp(-(\text{knn-dists}_i - \rho)/\sigma) = \log_2(n)$
  return σ



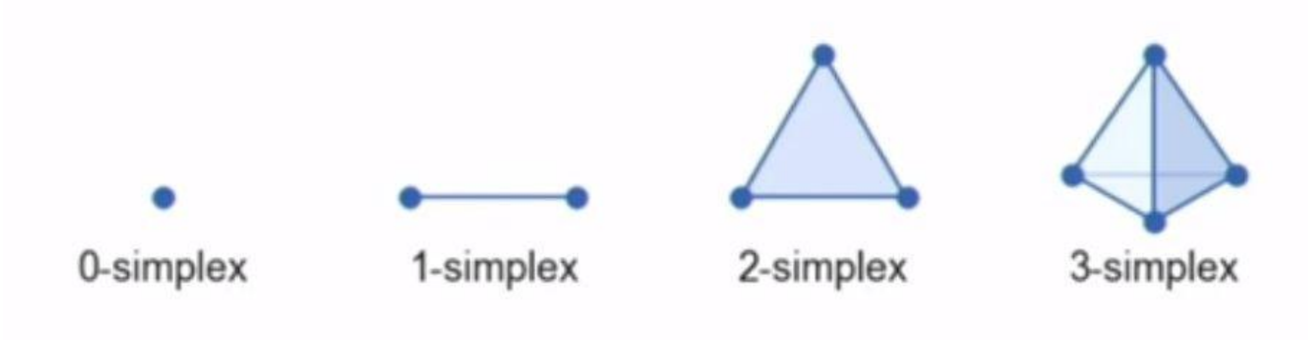


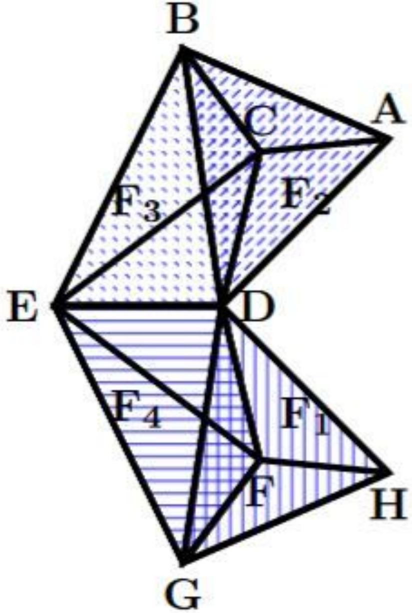
https://ebrary.net/74165/environment/sampling_theory



http://onlinestatbook.com/2/summarizing_distributions/spread_sim.html

“...converting the [set of] ***metric spaces*** *into* ***fuzzy simplicial sets”***





Fuzzy set:

- Carrier set *A*
- Membership function $\mu : A \to [0, 1]$

Example:

$\mu(x)$ for $x \in A$ to be the **membership strength** of x to the set A.

UMAP - cost function

Optimizes fuzzy set cross entropy:

$$C_{UMAP} = \sum_{i \neq j} \boxed{v_{ij} \log\left(\frac{v_{ij}}{w_{ij}}\right)} + \boxed{(1 - v_{ij}) \log\left(\frac{1 - v_{ij}}{1 - w_{ij}}\right)}$$

v_{ij}...membership strenght of j to fuzzy set of i in orig. space
w_{ij}...membership strenght of j to fuzzy set of i in red. space

$$C = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

p_{ij}...probability distance in original space
q_{ij}...probability distance in reduced space

Algorithm 5 Optimizing the embedding

function OptimizeEmbedding(top-rep, Y, min-dist, n-epochs)
  $\alpha \leftarrow 1.0$
  Fit Φ from Ψ defined by min-dist
  for $e \leftarrow 1, \ldots,$ n-epochs **do**
    for all $([a, b], p) \in$ top-rep$_1$ **do**
      if Random( ) $\leq p$ **then** ▷ Sample simplex with probability p
        $\boxed{y_a \leftarrow y_a + \alpha \cdot \nabla(\log(\Phi))(y_a, y_b)}$
        for $i \leftarrow 1, \ldots,$ n-neg-samples **do**
          $c \leftarrow$ random sample from Y
          $\boxed{y_a \leftarrow y_a + \alpha \cdot \nabla(\log(1 - \Phi))(y_a, y_c)}$
    $\alpha \leftarrow 1.0 - e/$n-epochs
  return Y

Sources

McInnes, Leland, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." *arXiv preprint arXiv:1802.03426* (2018).

He, Jinrong, et al. "Intrinsic dimensionality estimation based on manifold assumption." *Journal of Visual Communication and Image Representation* 25.5 (2014): 740-747.

van der Maaten, Laurens & Postma, Eric & Herik, H.. (2007). Dimensionality Reduction: A Comparative Review. Journal of Machine Learning Research - JMLR. 10.

https://blog.datawow.io/the-curse-of-dimensionality-c99409eb58e9

https://towardsdatascience.com/kernel-pca-vs-pca-vs-ica-in-tensorflow-sklearn-60e17eb15a64

https://towardsdatascience.com/understanding-pca-autoencoders-algorithms-everyone-can-understand-28ee89b570e2

https://medium.com/bluekiri/understanding-principal-component-analysis-once-and-for-all-9f75e7b33635

http://blog.thegrandlocus.com/img/Gauss_vs_Cauchy.png