Recitation 7a

Dimensionality Reduction

Principal Component Analysis

X contains n samples (rows), each representing a datapoint with f features (columns)

U contains n orthonormal columns

V^T contains f orthonormal rows

$$X = U\Sigma V^T$$

$$X = \begin{bmatrix} - & x_1 & - \\ & \vdots & \\ - & x_n & - \end{bmatrix} \qquad U = \begin{bmatrix} | & & | \\ u_1 & \dots & u_n \\ | & & | \end{bmatrix} \qquad V^T = \begin{bmatrix} - & v_1 & - \\ & \vdots & \\ - & v_f & - \end{bmatrix}$$

V can be viewed as a latent basis

$$X = \begin{bmatrix} - & x_1 & - \\ & \vdots & \\ - & x_n & - \end{bmatrix} \qquad U = \begin{bmatrix} | & & | \\ u_1 & \dots & u_n \\ | & & | \end{bmatrix} \qquad V^T = \begin{bmatrix} - & v_1 & - \\ & \vdots & \\ - & v_f & - \end{bmatrix}$$

$$X = U\Sigma V^T$$

$$x_i = \sum_{j=1}^{f} (u_j[i]\sigma_j) v_j^T$$

Supposing n≥f, although a similar analysis can be done assuming the opposite

Variance along any axis is a linear combination of singular values due to orthogonality of U

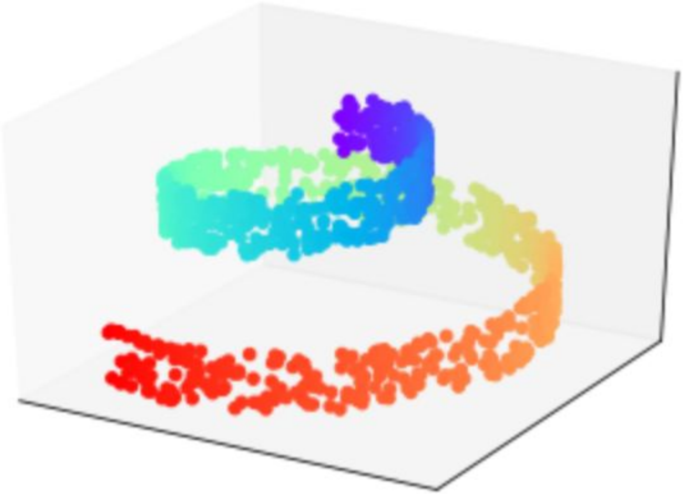
$$
\begin{aligned}
\mathrm{Var}(x_i w) &= \mathrm{Var}\Big(\sum_{j=1}^{f} u_j[i]\sigma_j v_j^T w\Big) \\
&= \sum_{j=1}^{f} (\sigma_j v_j^T w)^2 \mathrm{Var}(u_j[i]) \\
&= \sum_{j=1}^{f} \sigma_j^2 \cos^2(\theta_j)
\end{aligned}
$$

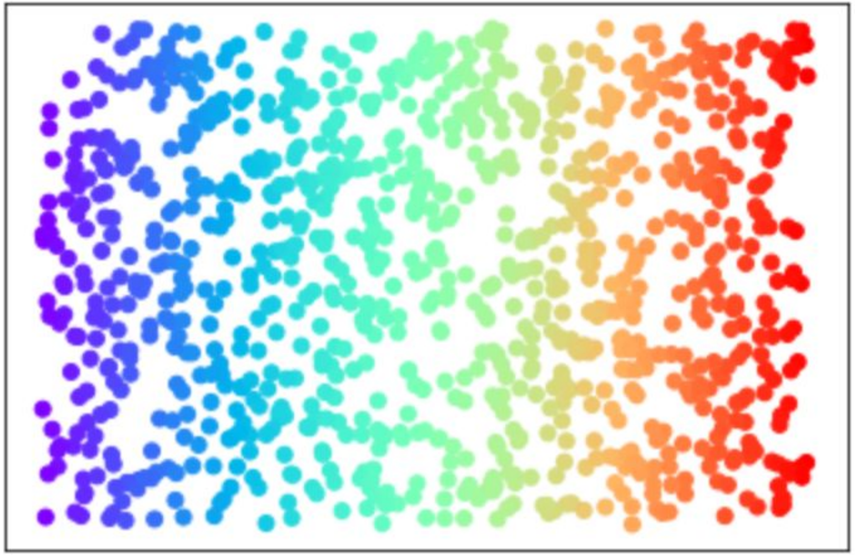
Finding the SVD

$$X = U\Sigma V^T$$
$$XX^T = U\Sigma^2 U^T$$
$$X^T X = V\Sigma^2 V^T$$

Diagnoalize XX^T and X^TX!

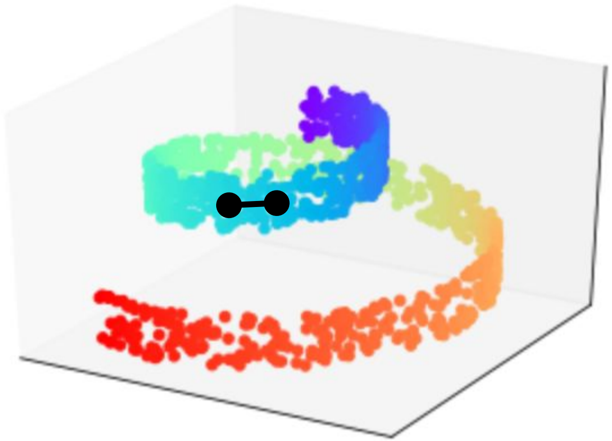
Non-linear dimensionality reduction algorithms



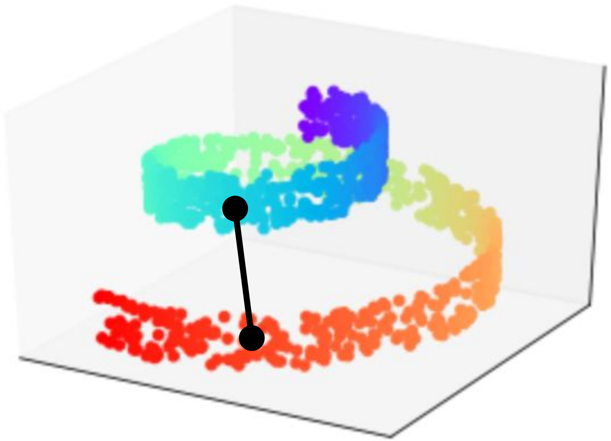


Manifolds

- Idea: the embedded space resembles euclidean space *locally*
  - Intuition: Distances between points are meaningful up until a certain extent



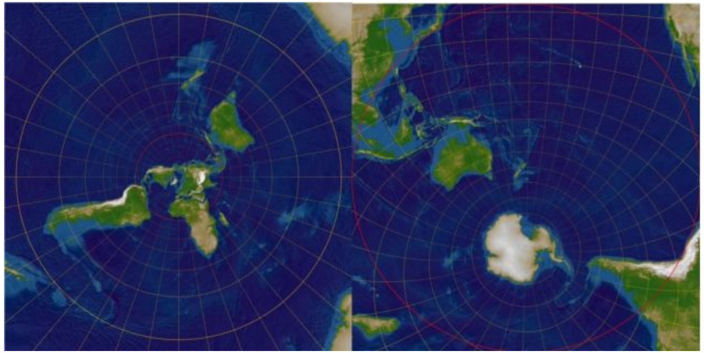
Small distances in ambient space are representative of distances on manifold

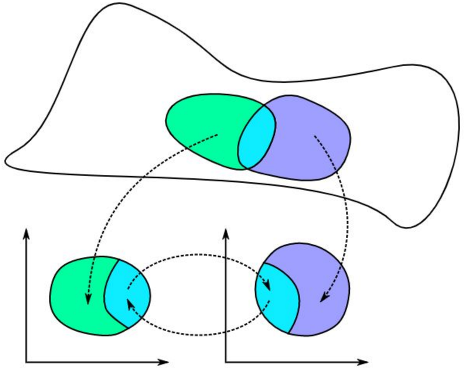


Large distance in ambient space may not be representative of distance on manifold

Charts and Atlases

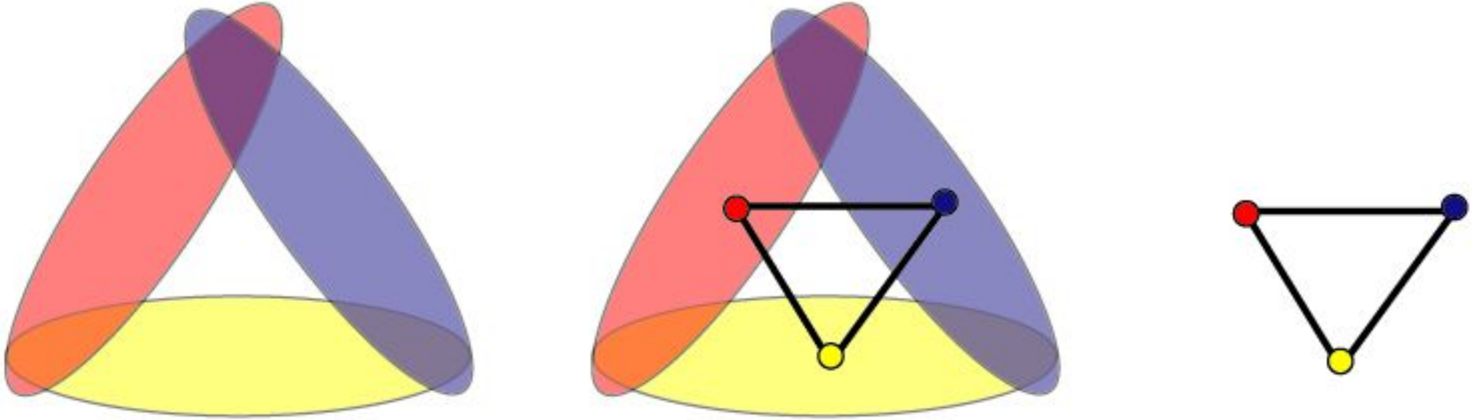
- Globally, a manifold’s topology may not even be (one-to-one, continuously) mappable to euclidean space
- An approach often taken is to take small pieces that are mappable, and then glue together (identify) those pieces to get a representation of the manifold
    - The resulting construct is known as an atlas





https://upload.wikimedia.org/wikipedia/commons/f/f0/Polar_stereographic_projections.jpg
https://upload.wikimedia.org/wikipedia/commons/0/06/Two_coordinate_charts_on_a_manifold.svg

Nerves



https://en.wikipedia.org/wiki/Nerve_of_a_covering#/media/File:Constructing_nerve.png

General strategy

- Step 1: Construct a set of locally faithful representations of the data
    - “Chart”

- Step 2: Combine the representations into a globally coherent representation
    - “Atlas”

- Step 3: Try to construct a lower dimensional dataset that is faithful to the global representation
    - “Nerves”

t-SNE: t-distributed stochastic neighbor embedding

- Step 1: For each point, it generates a probability of jumping to a different point from there

- Step 2: Symmetrize the distances

- Step 3: Find a lower dimensional embedding that generates a similar distribution

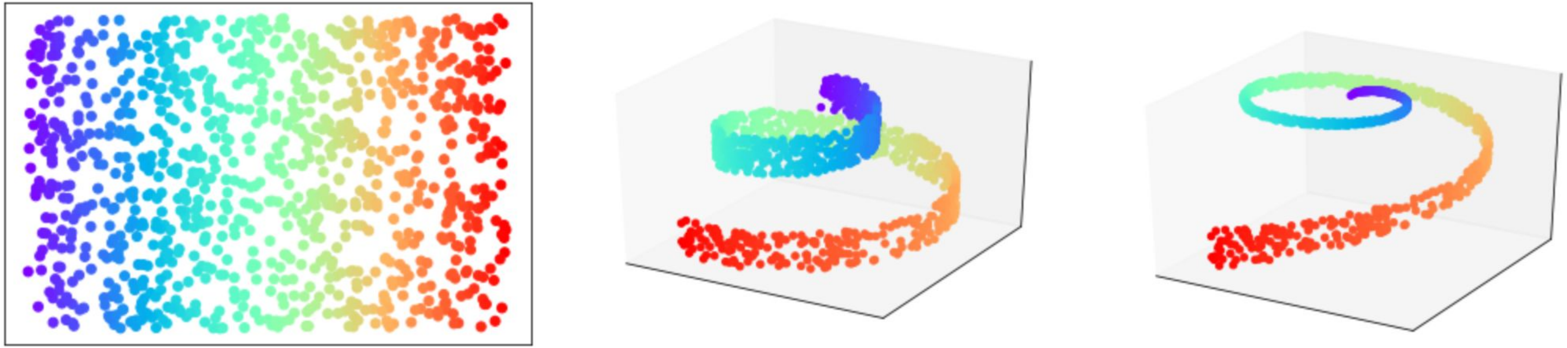
Local structure

- We determine the probability of going from point a to point b to be proportional to the exponential of the negative square euclidean distance
  - “On a random walk, you can get from one point to another if you’re close”
  - The probabilities decay exponentially, so large distances have negligible effect on the overall distribution

$$P_{i,j} = p(x_j | x_i) = \frac{\exp(-||x_i - x_j||^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2 / 2\sigma_i^2)}$$

Controlling heterogeneity

- Even if we assume that data is distributed uniformly on the embedded manifold, its embedding into the ambient space may induce non-uniformities



Solution: Modify sigma!

- We adjust sigma until the conditional distribution hits a fixed entropy/perplexity (hyperparameter)
    - perplexity = 2^{entropy}
- We find the correct sigma by binary search
    - Try the halfway point between the current upper bound and lower bound
    - Adjust the bounds
    - If bounds don’t exist, explode/decay exponentially

$$P_{i,j} = p(x_j|x_i) = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k\neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)}$$

Why do we want to fix perplexity?

- Suppose you have a bag with n items, and you draw an item
- Suppose all items are equally likely to be drawn
    - Entropy is log(n)
    - The number of items is 2^entropy = n
- Suppose one item is drawn with overwhelming probability
    - Entropy is close to 0
    - One can argue that the number of items is 2^entropy = 1, since once item is drawn with overwhelming probability
- Therefore, 2^entropy is like “the number of items you can draw from”
    - Therefore, fixing perplexity is like fixing the number of neighbours you have!

$$H(P_i) = -\sum_j p_{x_j|x_i} \log_2 p_{x_j|x_i}$$

Symmetrizing the distances

$$P_{ij}^{symmetric} = p(x_i, x_j) = \frac{p_{x_j|x_i} + p_{x_i|x_j}}{2N}$$

Finding a good lower dimensional embedding

$$Q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

$$C = KL(P||Q) = \sum_i \sum_j P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

Optimize with gradient descent

Why use a t-distribution instead of a Gaussian?

- Points can crowd together in high dimensional space without collapsing
- To avoid collapse in a lower dimensional space, we allow more measure in the tails of the distribution
  - Famously, the distribution in question has tails so heavy the expected values are undefined

Parametric t-SNE

Instead of calculating the embedding directly, train a function f() that embeds into the lower dimensional space

$$Q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

$$Q_{ij} = \frac{(1 + ||f(x_i) - f(x_j)||^2)^{-1}}{\sum_{k \neq l}(1 + ||f(x_k) - f(x_l)||^2)^{-1}}$$

Parametric t-SNE

We have an output, a target, and a cost function, so we can train this just like any other ML model we’ve trained so far.

We can even perform updates in batch if we expect f() to generalize

$$Q_{ij} = \frac{(1 + ||f(x_i) - f(x_j)||^2)^{-1}}{\sum_{k \neq l}(1 + ||f(x_k) - f(x_l)||^2)^{-1}}$$

$$C = KL(P||Q) = \sum_i \sum_j P_{ij} \log \frac{P_{ij}}{Q_{ij}}$$

UMAP: Uniform Manifold Approximation and Projection

- An alternative to t-SNE with strong theoretical foundations and fast runtime
- Theoretical assumptions:
    - Data is uniformly distributed on the underlying manifold
    - The manifold is connected
    - The reduced representation should reproduce the connective structure of the underlying manifold

UMAP - local structure

- For each point, find the k nearest neighbours and compute a weight for each of them
- The weight is local to the point and represents the probability that the neighbours are connected “from the point’s perspective”
    - The edge the weight is on is directed from the origin to the neighbour

$$\rho_i = \min\{d(x_i, x_{i_j}) \mid 1 \leq j \leq k, d(x_i, x_{i_j}) > 0\}$$

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right) = \log_2(k)$$

$$w((x_i, x_{i_j})) = \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right)$$

https://arxiv.org/pdf/1802.03426.pdf

UMAP - global structure

- Two different local structures may disagree on whether an edge exists
- To get the global probability that an undirected edge exists, we calculate the probability that either directed edge exists

$$B = A + A^\top - A \circ A^\top$$

https://arxiv.org/pdf/1802.03426.pdf

UMAP - reduced representation

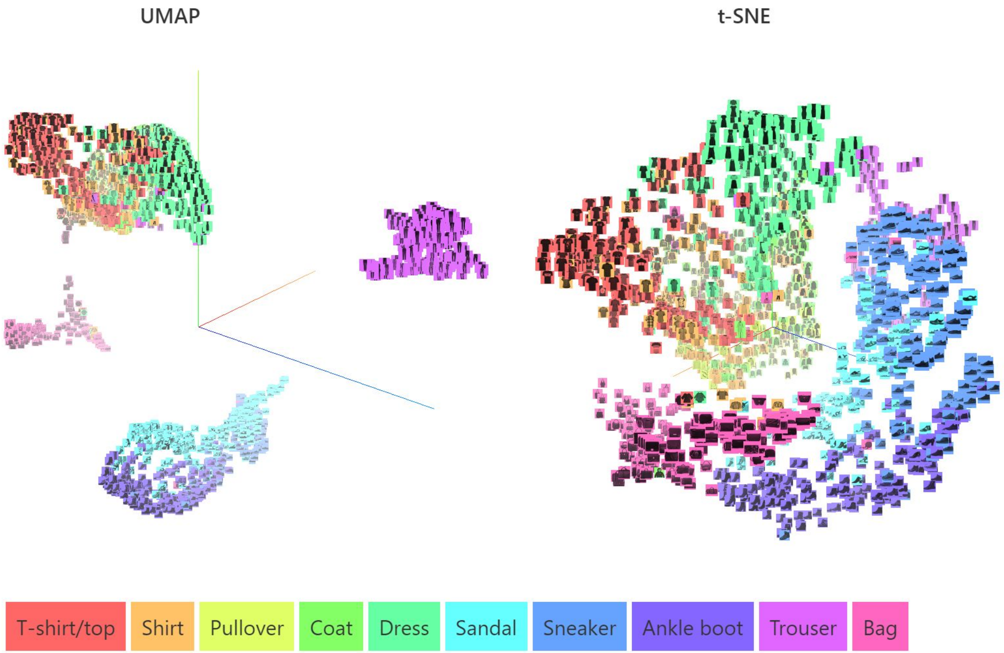
- Each edge then exerts a pulling force between the vertices it’s between
  - O(kn) updates
- Each pair of vertices has a repulsive force between them
  - O(n^2) updates, but can use sampling to reduce the runtime

$$\frac{-2ab\|\mathbf{y_i}-\mathbf{y_j}\|_2^{2(b-1)}}{1+\|\mathbf{y_i}-\mathbf{y_j}\|_2^2}w((x_i,x_j))\left(\mathbf{y_i}-\mathbf{y_j}\right)$$

$$\frac{2b}{\left(\epsilon+\|\mathbf{y_i}-\mathbf{y_j}\|_2^2\right)\left(1+a\|\mathbf{y_i}-\mathbf{y_j}\|_2^{2b}\right)}\left(1-w((x_i,x_j))\right)\left(\mathbf{y_i}-\mathbf{y_j}\right)$$

https://arxiv.org/pdf/1802.03426.pdf

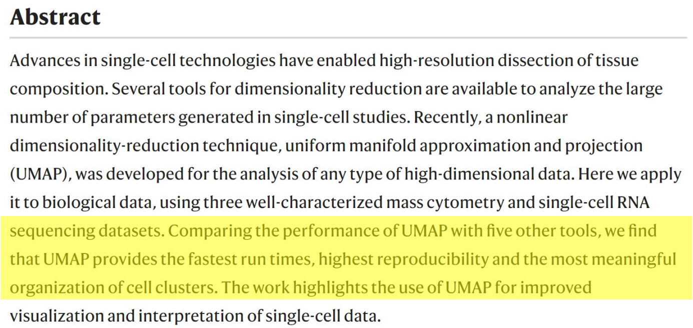
t-SNE vs UMAP



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.

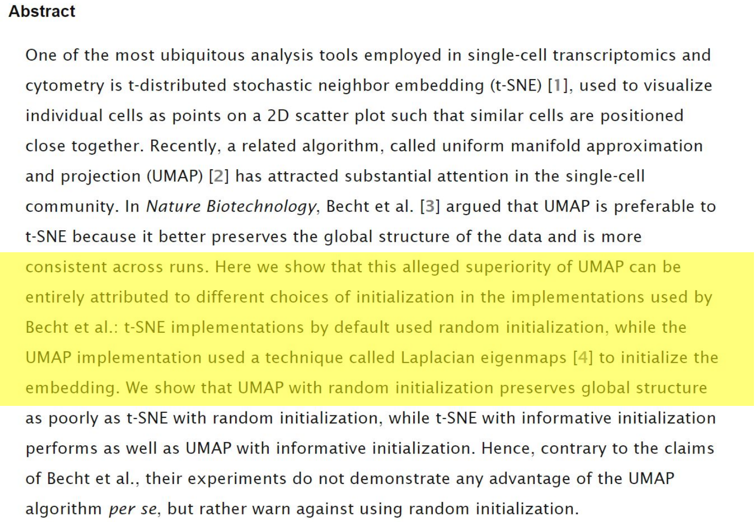
https://pair-code.github.io/understanding-umap/

t-SNE vs UMAP



Abstract

Advances in single-cell technologies have enabled high-resolution dissection of tissue composition. Several tools for dimensionality reduction are available to analyze the large number of parameters generated in single-cell studies. Recently, a nonlinear dimensionality-reduction technique, uniform manifold approximation and projection (UMAP), was developed for the analysis of any type of high-dimensional data. Here we apply it to biological data, using three well-characterized mass cytometry and single-cell RNA sequencing datasets. Comparing the performance of UMAP with five other tools, we find that UMAP provides the fastest run times, highest reproducibility and the most meaningful organization of cell clusters. The work highlights the use of UMAP for improved visualization and interpretation of single-cell data.



Abstract

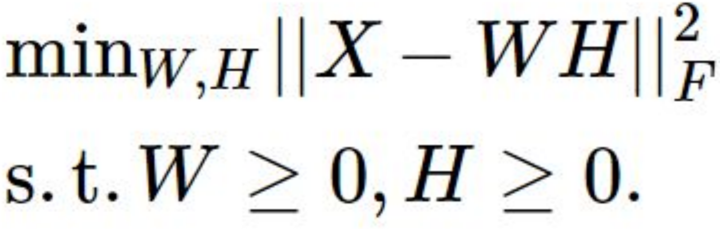
One of the most ubiquitous analysis tools employed in single-cell transcriptomics and cytometry is t-distributed stochastic neighbor embedding (t-SNE) [1], used to visualize individual cells as points on a 2D scatter plot such that similar cells are positioned close together. Recently, a related algorithm, called uniform manifold approximation and projection (UMAP) [2] has attracted substantial attention in the single-cell community. In *Nature Biotechnology*, Becht et al. [3] argued that UMAP is preferable to t-SNE because it better preserves the global structure of the data and is more consistent across runs. Here we show that this alleged superiority of UMAP can be entirely attributed to different choices of initialization in the implementations used by Becht et al.: t-SNE implementations by default used random initialization, while the UMAP implementation used a technique called Laplacian eigenmaps [4] to initialize the embedding. We show that UMAP with random initialization preserves global structure as poorly as t-SNE with random initialization, while t-SNE with informative initialization performs as well as UMAP with informative initialization. Hence, contrary to the claims of Becht et al., their experiments do not demonstrate any advantage of the UMAP algorithm *per se*, but rather warn against using random initialization.

https://www.nature.com/articles/nbt.4314
https://www.biorxiv.org/content/10.1101/2019.12.19.877522v1

NMF: Nonnegative Matrix Factorization

We want to find an approximation of X as the product of *entrywise positive* matrices

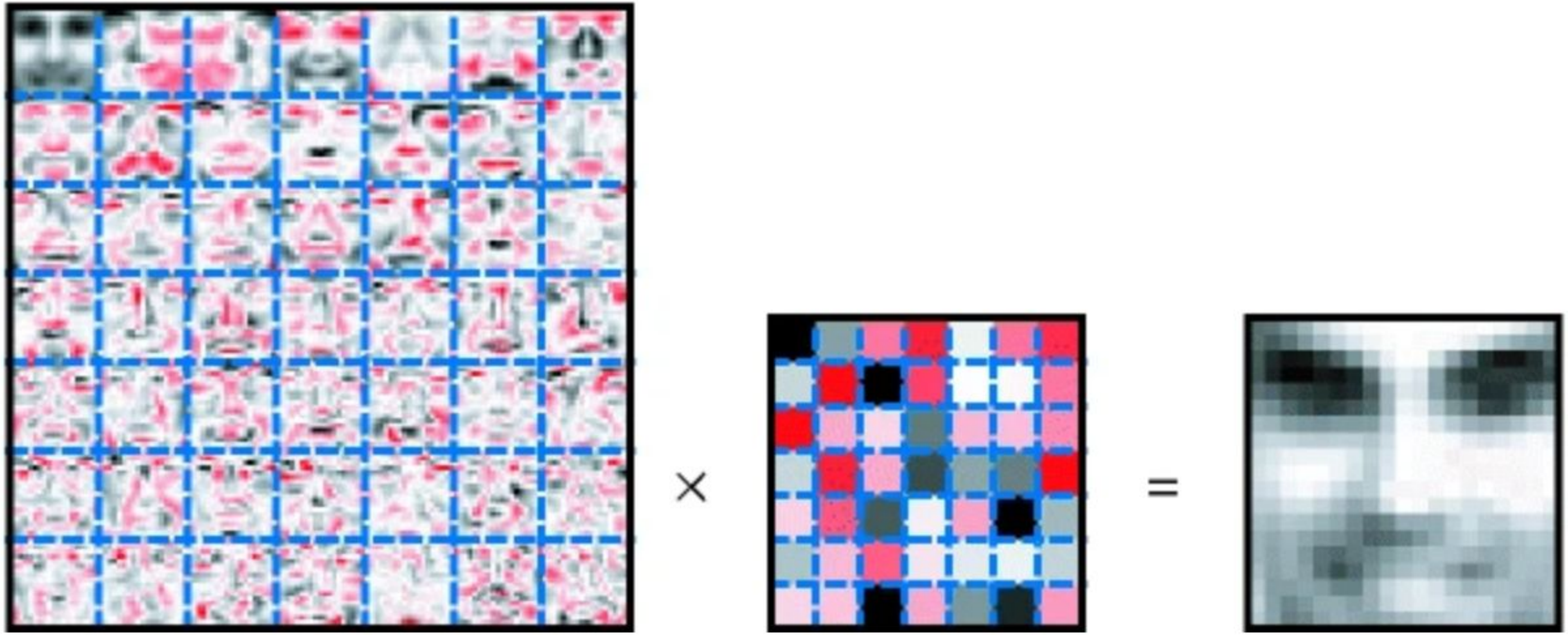
Each latent basis will have an additive contribution to the sample



$$\min_{W,H} ||X - WH||_F^2$$
$$\text{s. t.}\, W \geq 0, H \geq 0.$$

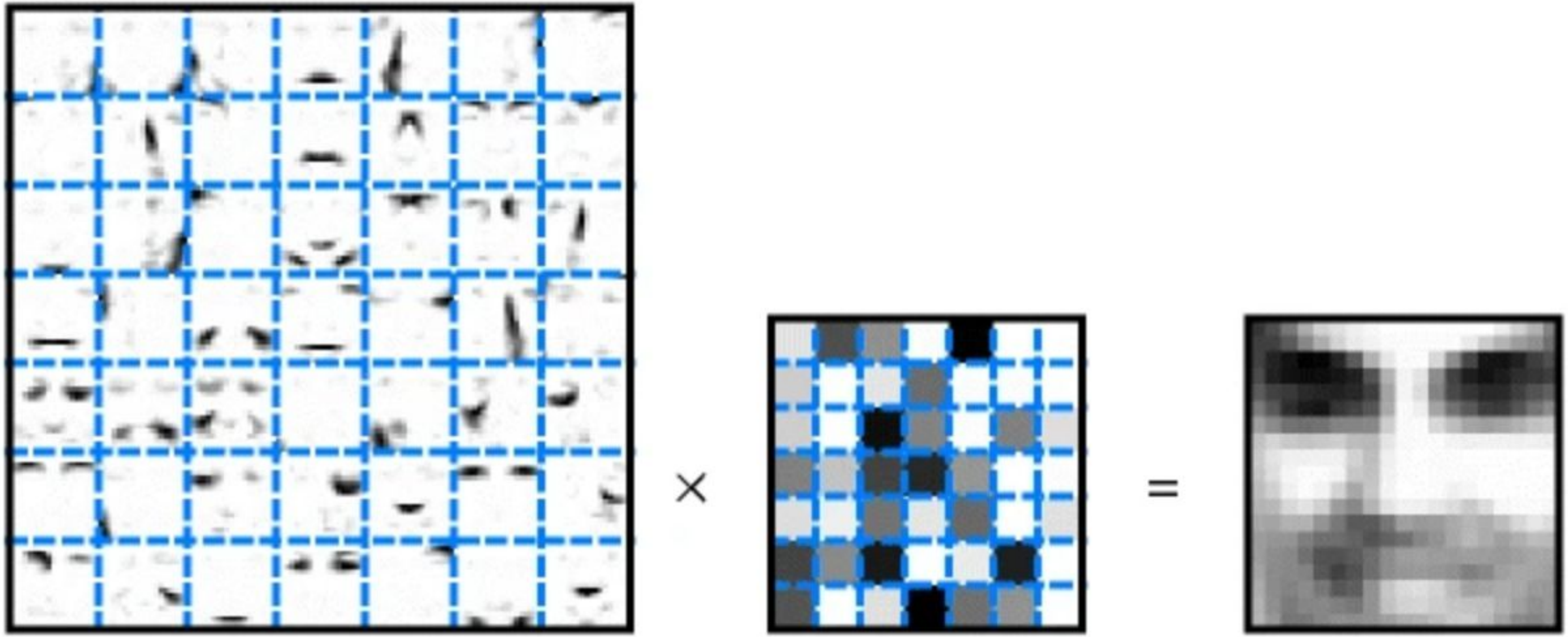
https://academic.oup.com/bioinformatics/article/32/1/1/1743821

PCA



https://www.nature.com/articles/44565

NMF



https://www.nature.com/articles/44565