Manifold Methods for Dimension Reduction

CRISTIAN BRAVO ROMAN

OFFICE 280

CBRAVORO@UWO.CA

This Lecture

What's a manifold

Manifold Methods

- t-SNE
- UMAP

You should read…

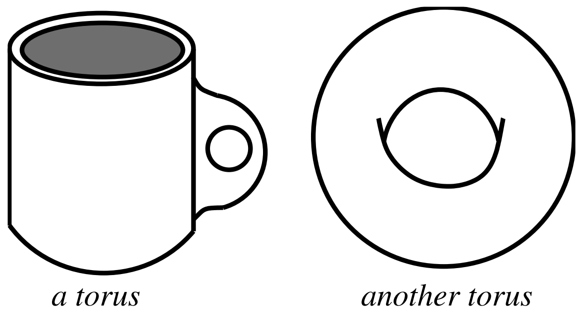
- Chapter 14 of the text book.
- The original t-SNE paper https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf and the paper explaining its pitfalls: https://distill.pub/2016/misread-tsne/
- The intro, experiments and appendix C of the UMAP paper: https://arxiv.org/pdf/1802.03426.pdf and its more coloquial explanation https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668

Manifolds

A **manifold** is a topological space that is **locally Euclidian**

- This means that, around every point, we can put an open unit ball in $\mathbb{R}^n$.
- Any object that, at small scales, looks nearly flat is a manifold.
- Can it be “charted”? Then it’s a manifold!

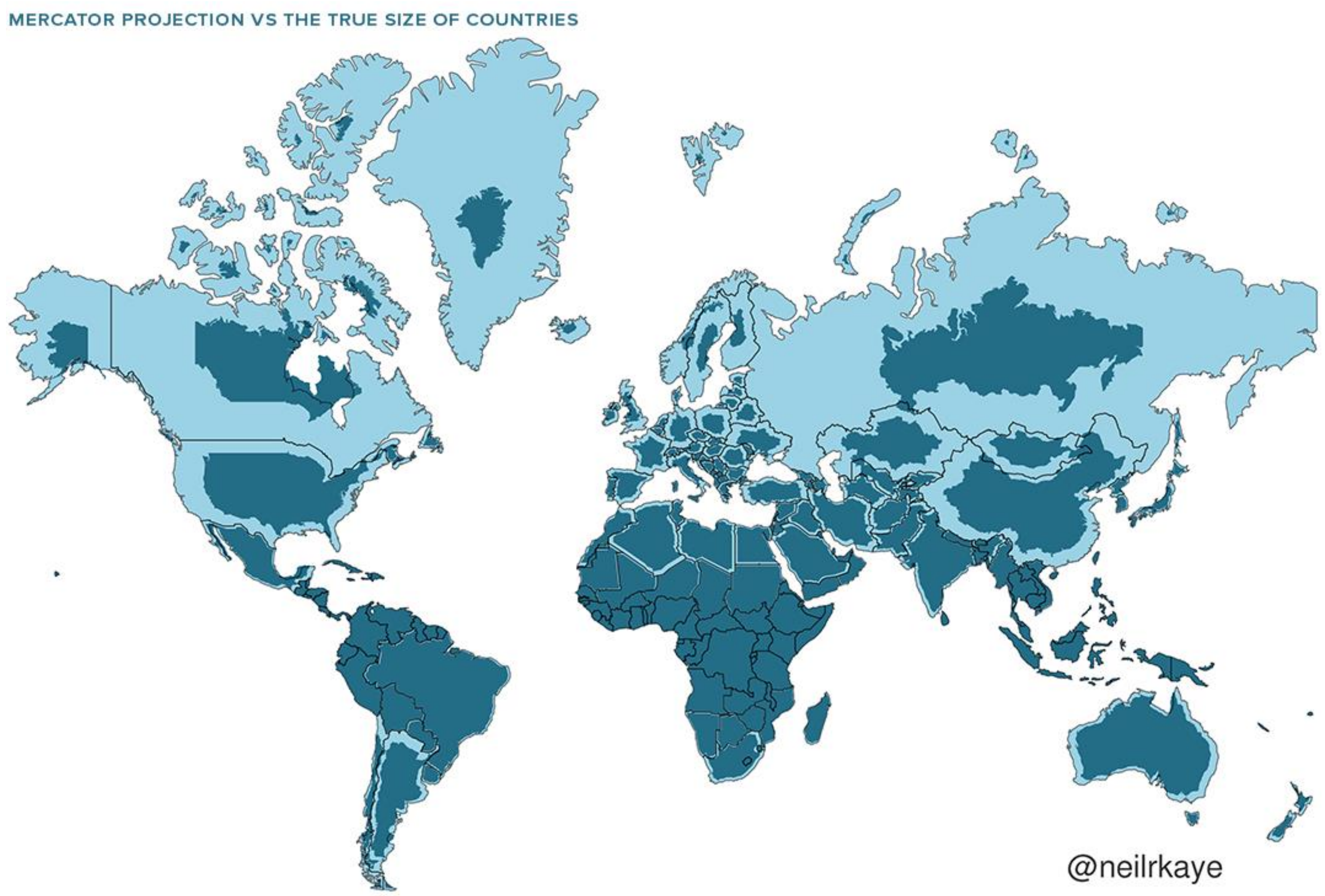
They key is that the distance needs not be consistent in different parts of the space.

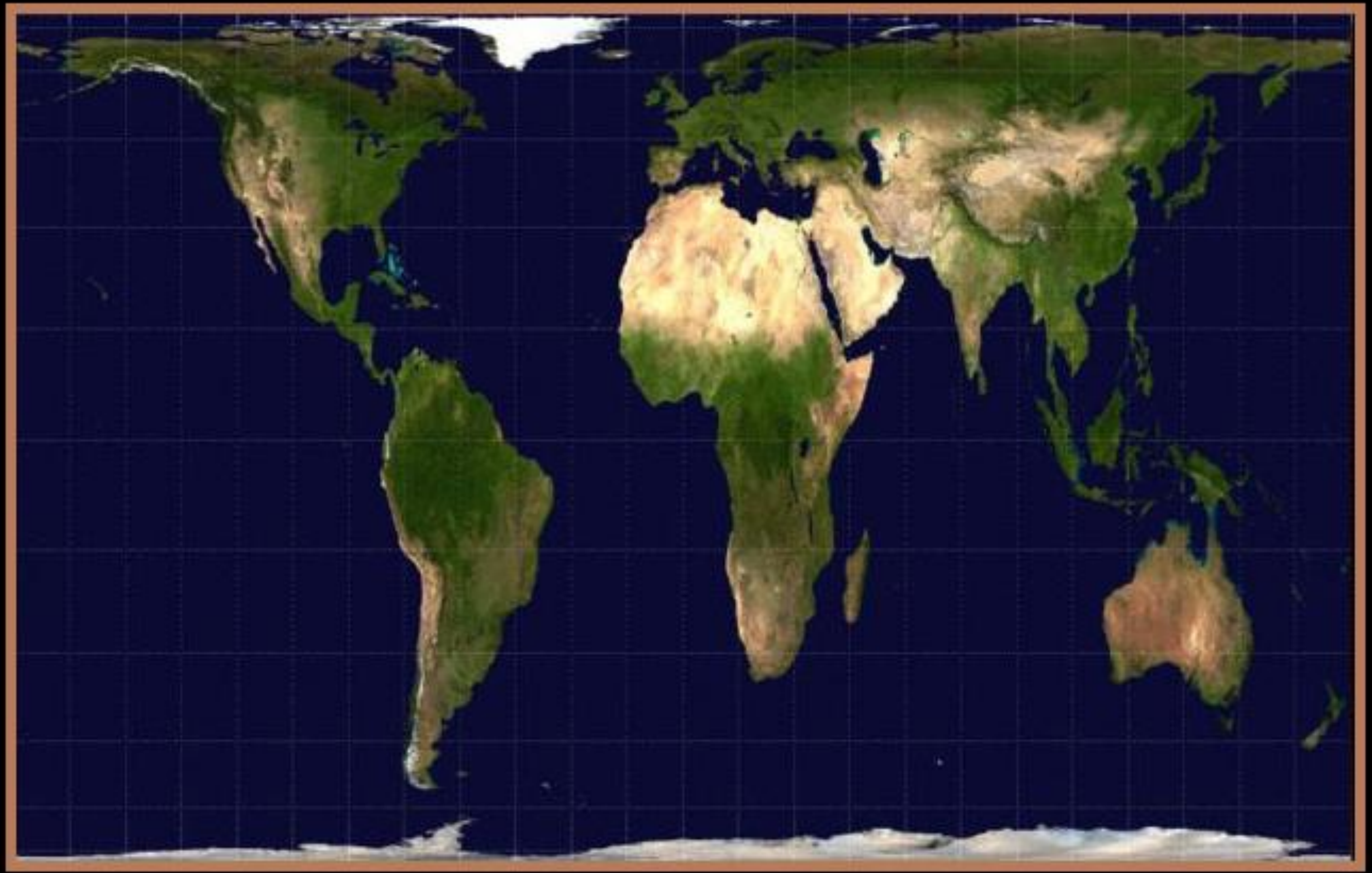


source: Wolfram

Think of a Mercator Projection.

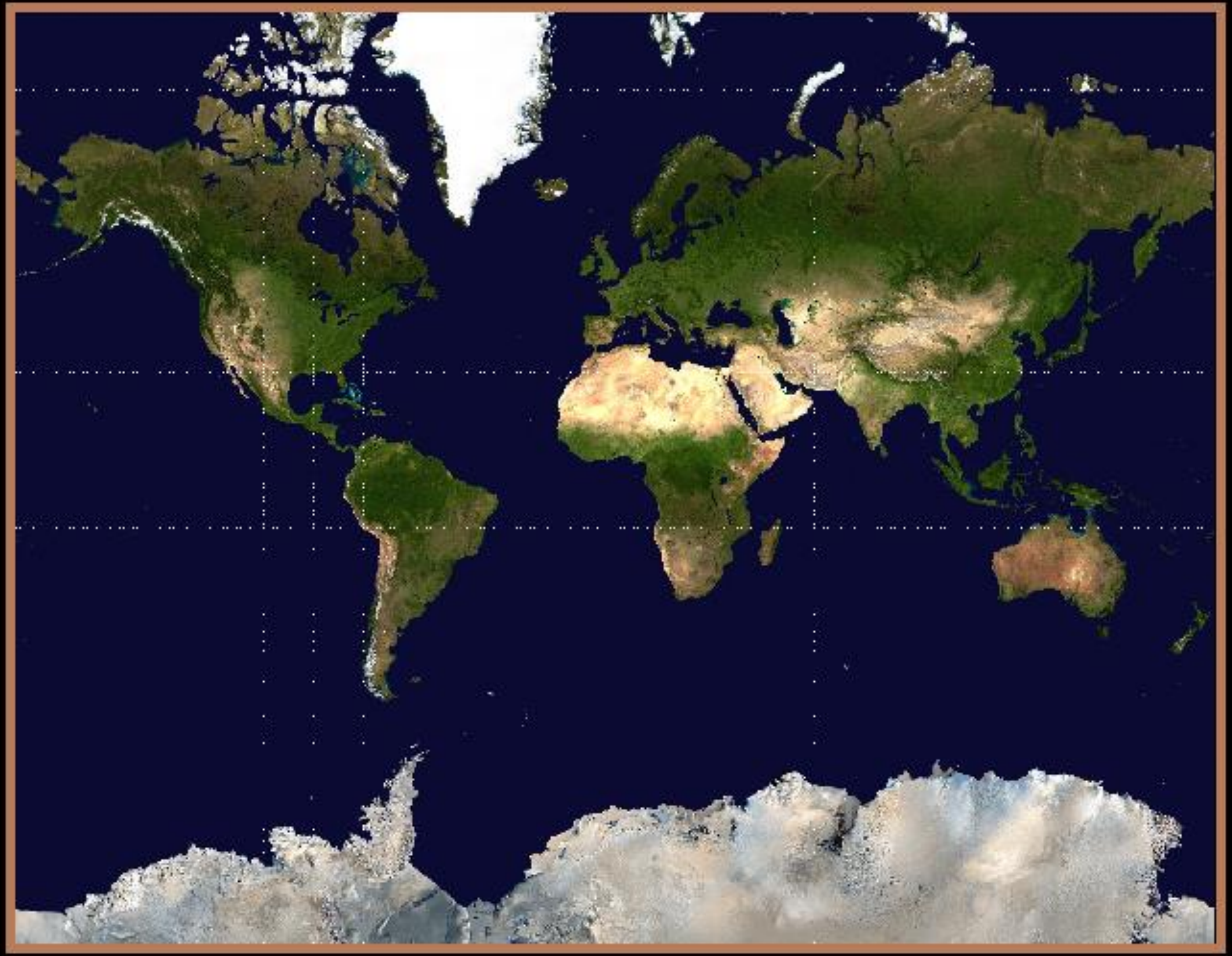
- It’s locally Euclidian (measuring distance on a piece of the map is consistent.
- But it is globally distorted!


MERCATOR PROJECTION VS THE TRUE SIZE OF COUNTRIES
@neilrkaye



Peters Projection

The true representaion of land area
(the "size" of continents and countries)



Mercator Projection

Incorrect/false repesentation of land area

Why is this useful?

When doing **non-linear dimensionality reduction** this is what we are doing!

- Compare this to PCA, that creates a rotation of the space, but leaves the distances alone.

A non-linear method will **modify the distances locally** to accomplish compression.

- Some distances will be preserved.
- Some will be distorted.

The idea is to preserve the **topology** of the space when reducing dimensions.

- Topology "studies properties of spaces that are invariant under any continuous deformation. It is sometimes called 'rubber-sheet geometry'"

Methods

Many sophisticated mathematical concepts are part of this area. Most of this concepts come from Manifold Theory in mathematics.

There are many methods, but we will focus on two:

- t-SNE: t-distributed stochastic neighbor embedding.
- UMAP: Uniform Manifold Approximation Projection.

I will be skimming most of the math, but the papers a great source of information if you want to get into detail!

t-Distributed Stochastic Neighbor Embedding

t-distributed stochastic neighbor embedding

Method proposed by van der Maaten and Hinton in 2008. The method is based **on defining a probability distribution around each point**. For this, we will define the following:

- **Perplexity**: How well a probability distribution predicts a sample.
- $p_{j|i}$: Probability that

What is the probability that element i picks j as its neighbour?

- It should depend on the distance between the points.
- It should depend on the shape of the distribution we chose.

Picking a Gaussian with standard deviation (*bandwith)* σ_i (unique for each point and a function of the perplexity) we can use:

$$p_{j|i} = \frac{\exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma_i^2}\right)}{\sum_{k\neq i} \exp\left(-\frac{\left\|x_i - x_k\right\|^2}{2\sigma_i^2}\right)}$$

Creating a Projection

Now that we have a probability we can think on creating distances. The probability $p_{j|i}$ is not symmetric. To make symmetric we simply calculate

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

We will now build a projection $y \subset \mathbb{R}^d$ with $d < V$ (usually just 2 or 3) so that its probabilities q_{ij} look as similar as possible to p_{ij}. We cannot use the same Gaussian given the tails can be lost. But we **can** use a t-distribution.

Defining

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

(a t distribution with one degree of freedom) then we have two distributions that **we want to make as similar as possible!**

A Loss for Manifold Projection

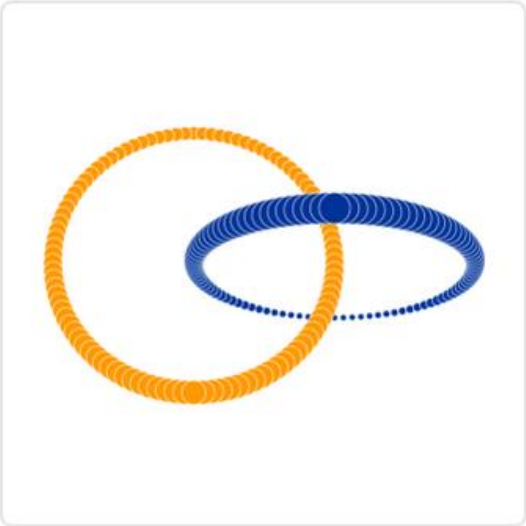
How can we measure if two distributions are similar?

We can measure the difference between two distributions using the **Kullback-Leibler Divergence**.

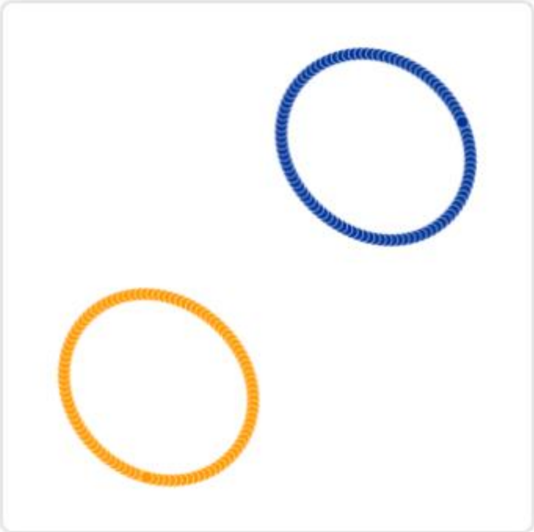
This divergence is such that:

$$D_{p||q} = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$$

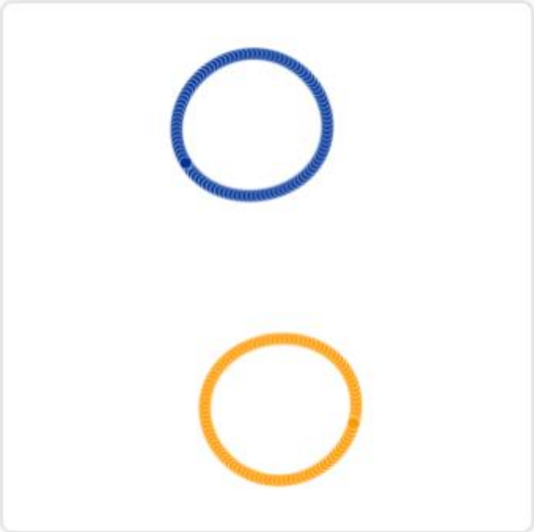
This is our loss function. t-SNE will minimize this divergence to obtain the best projection.



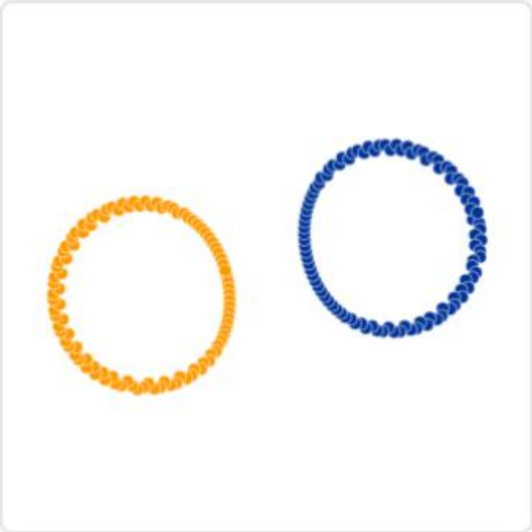
Original



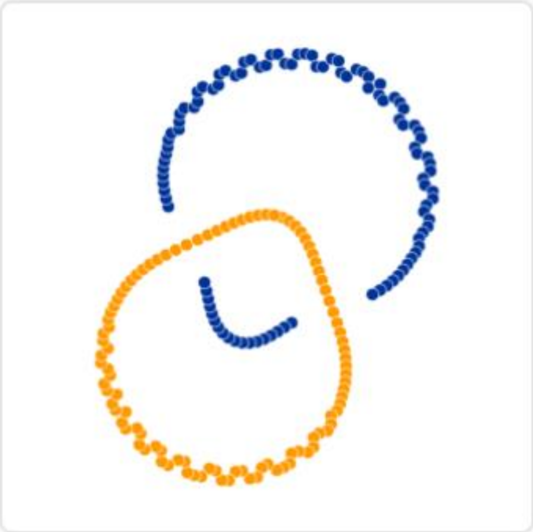
Perplexity: 2
Step: 5,000



Perplexity: 5
Step: 5,000

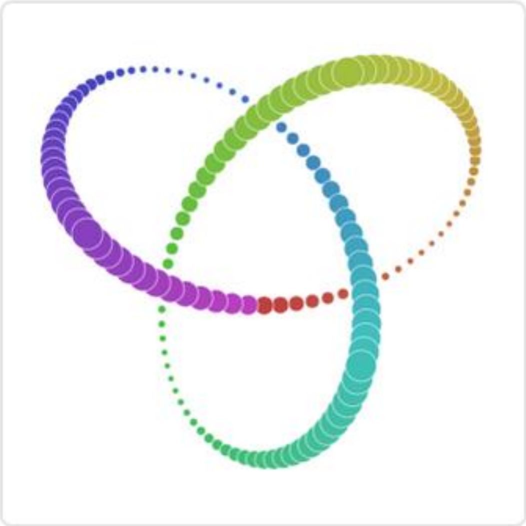


Perplexity: 30
Step: 5,000

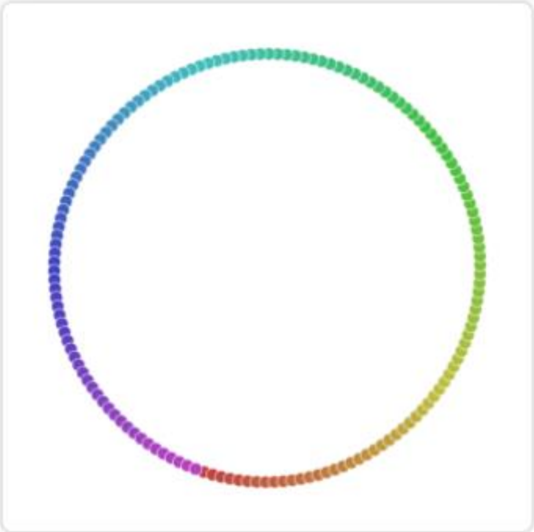


Perplexity: 50
Step: 5,000

Perplexity: 100
Step: 5,000

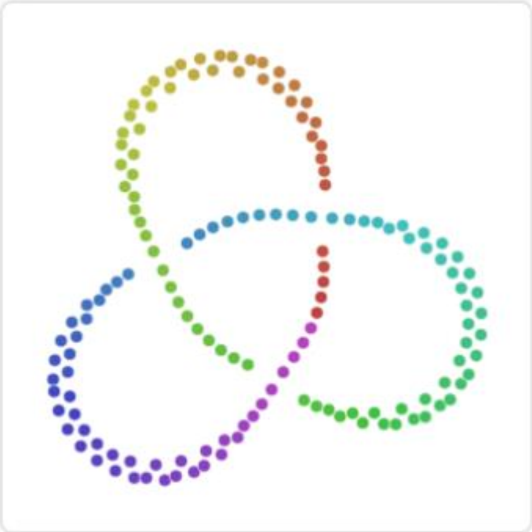


Original

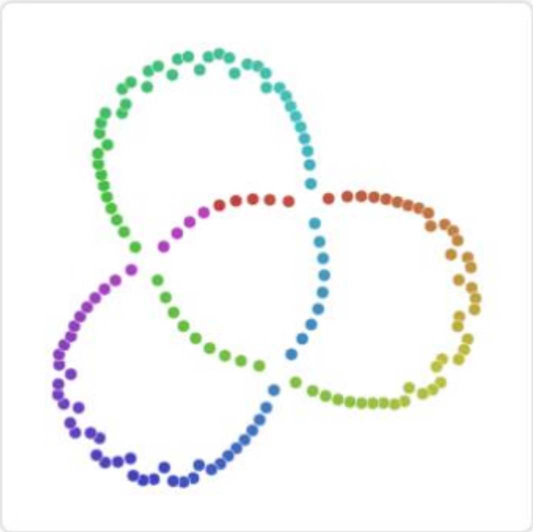


Perplexity: 2
Step: 5,000

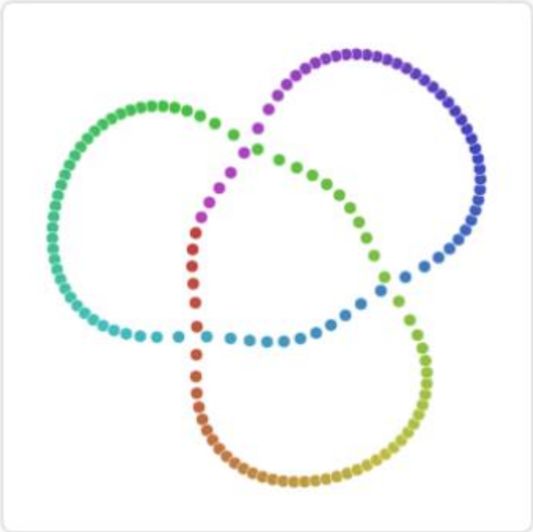
Perplexity: 5
Step: 5,000



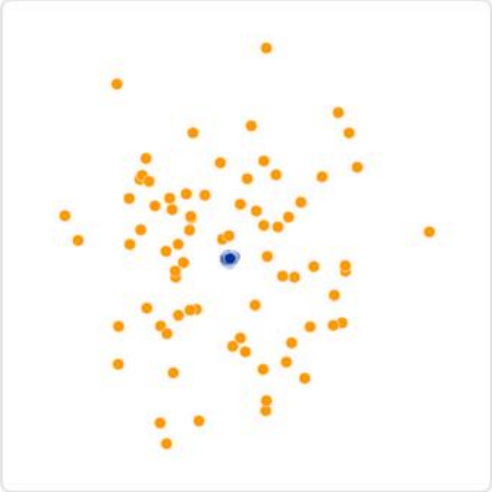
Perplexity: 30
Step: 5,000



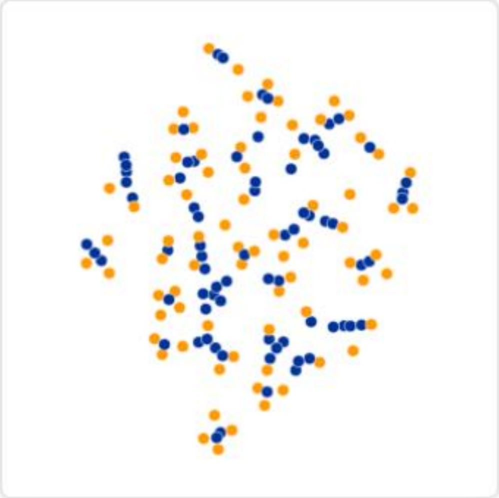
Perplexity: 50
Step: 5,000



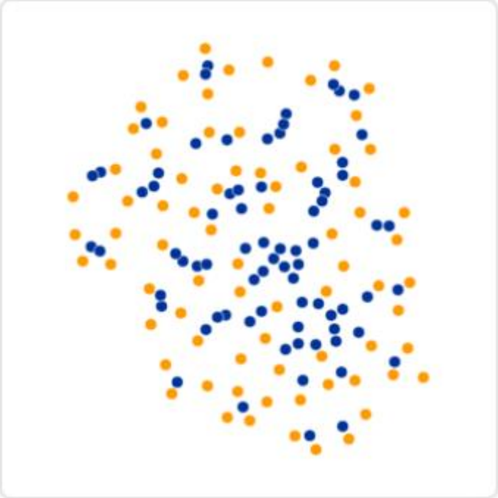
Perplexity: 100
Step: 5,000



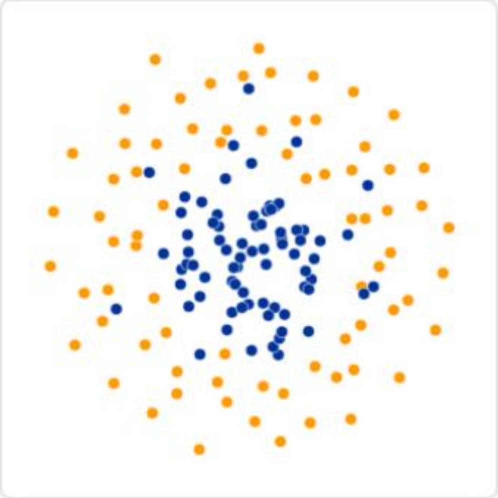
Original



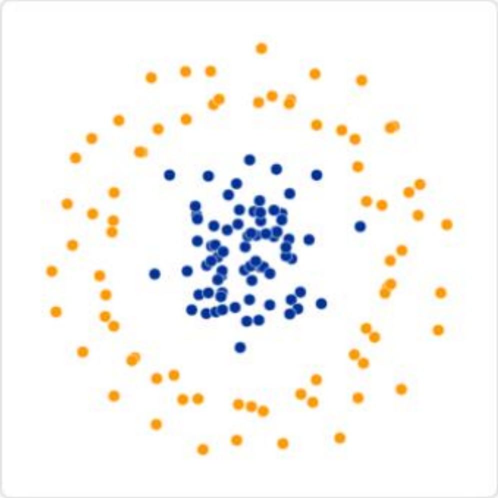
Perplexity: 2
Step: 5,000



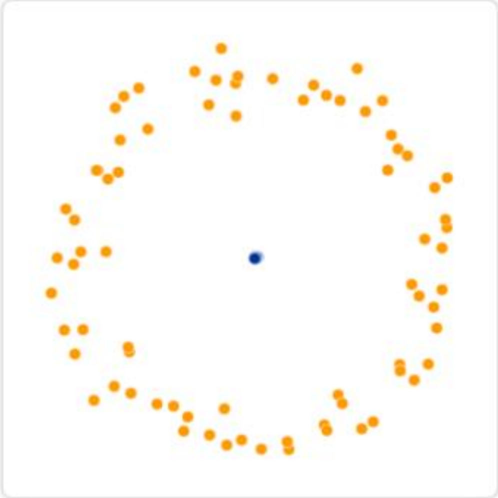
Perplexity: 5
Step: 5,000



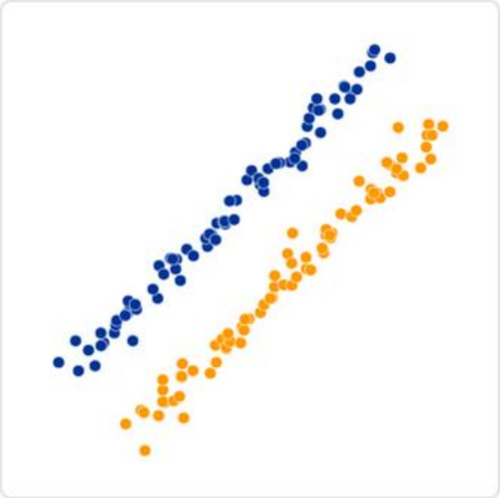
Perplexity: 30
Step: 5,000



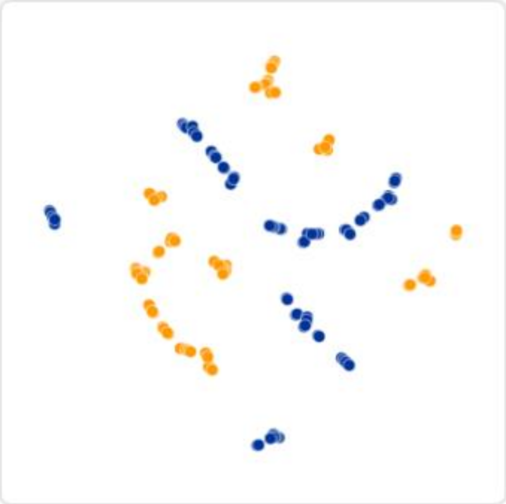
Perplexity: 50
Step: 5,000



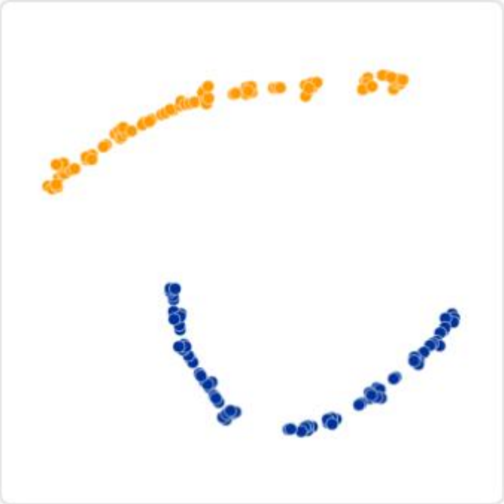
Perplexity: 100
Step: 5,000



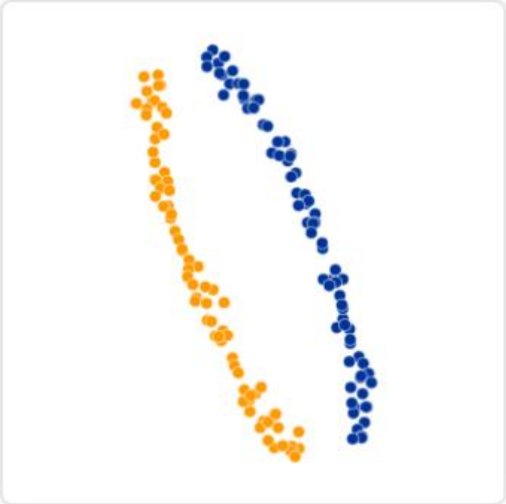
Original



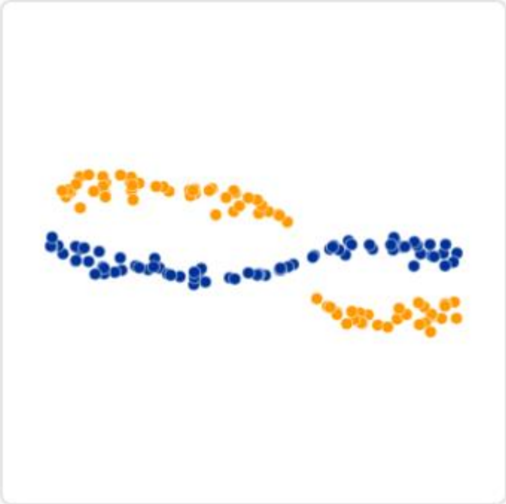
Perplexity: 2
Step: 5,000



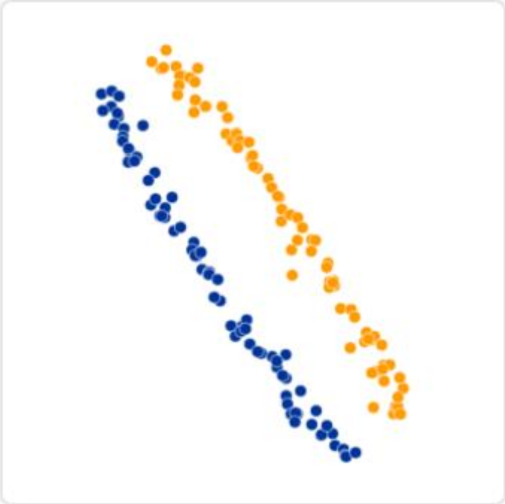
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000



Perplexity: 100
Step: 5,000

t-SNE in Practice

The method requires several parameters, but the most important ones are:
- The perplexity: Complex, depends on the problem. Try several in a wide range and choose.
- The number of epochs: enough to get convergence!

The method is supposed to work for more than 3 dimensions, but it is exponentially more expensive to do so!
- In practice this means t-SNE can only be used for plotting high dimensional spaces.
- We may lose a lot of information.

Read https://distill.pub/2016/misread-tsne/

These shortcomings lead to new methods. The most important (and modern one) is **Uniform Manifold Approximation Projection** (UMAP).

Uniform Manifold Approximation Projection

The Problems with t-SNE

1. It does not scale well. Calculating

$$\sum_{k}\sum_{k\neq l}\left(1+\|y_k-y_l\|^2\right)^{-1}$$

In the q_{ij} equation is very costly.

2. It cannot work on sparse high-dimensional data directly! Normally it first compresses the data with PCA.

3. It is very expensive in memory as it works with large dense matrices.

4. **It only preserves local structure**. And you have to be very careful with the perplexity parameter.

Enter UMAP

UMAP tries to handle these problems by dealing with more expensive parts of the t-SNE equations, while also adding a few intelligent tricks. Its principles are the same though!

In UMAP case:

- The probability $p_{i|j}$ is now modelled as an exponential. It is also allowed to use any distance $d(x_i, x_j)$ not just Euclidian

$$p_{i|j} = \exp\left(-\frac{d(x_i, x_j) - \rho_i}{\sigma_i}\right)$$

Where ρ_i is the **minimum distance parameter**, or the closest distance we will allow a point to look for neighbours (closer points are ignored, or "clumped" together. Note that the probabilities are **not normalized** thus making UMAP significantly more efficient than t-SNE.

- UMAP also **does not use perplexity** but uses the number of nearest neighbours to determine the distributions.

$$k = 2^{\Sigma p_{ij}}$$

UMAP Assumptions (cont'd)

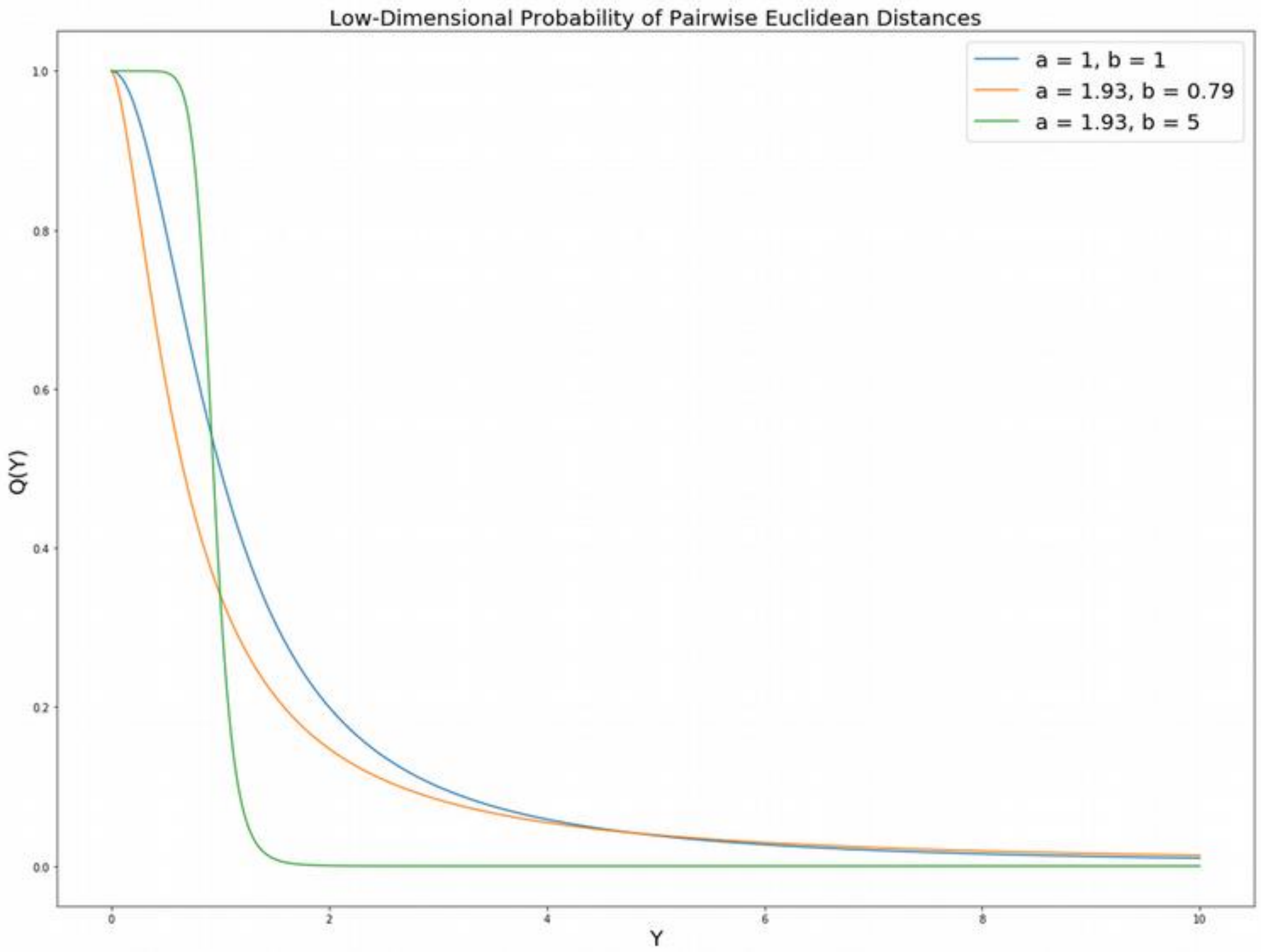
The ρ_i parameter also causes that some points are together, so to make the probabilities symmetric we need to correct the probabilities.

$$p_{ij} = p_{i|j} + p_{j|i} - p_{i|j} \cdot p_{j|i}$$

Given these corrections, it is also necessary to adjust the t-distribution accordingly. To calculate the distribution q_{ij} UMAP uses the following:

$$q_{ij} = \left(1 + a\left(y_i - y_j\right)^{2b}\right)^{-1}$$

This is close to a t-distribution, but has the two parameters a and b, and it is also not normalized. In practice, this is found



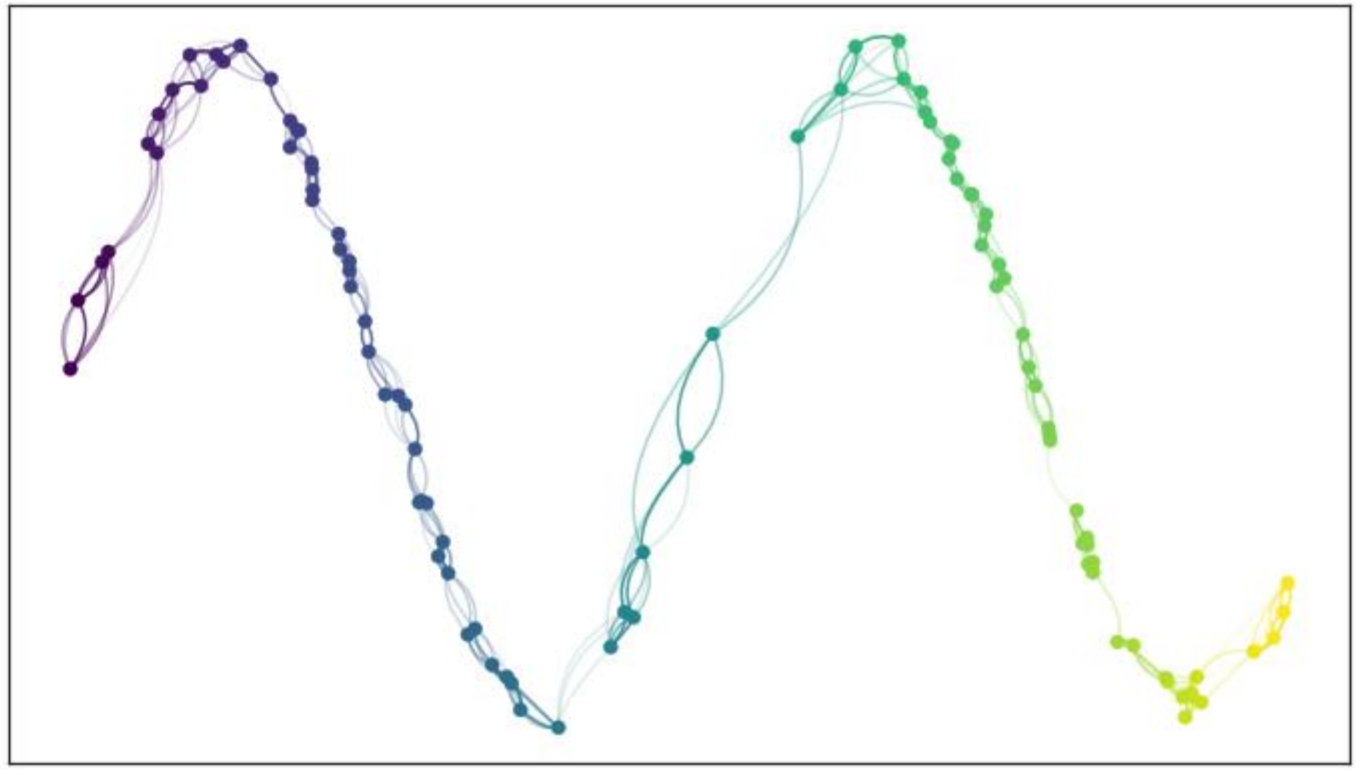
Source: Oskolov (2019, TDS)

UMAP’s Loss Function

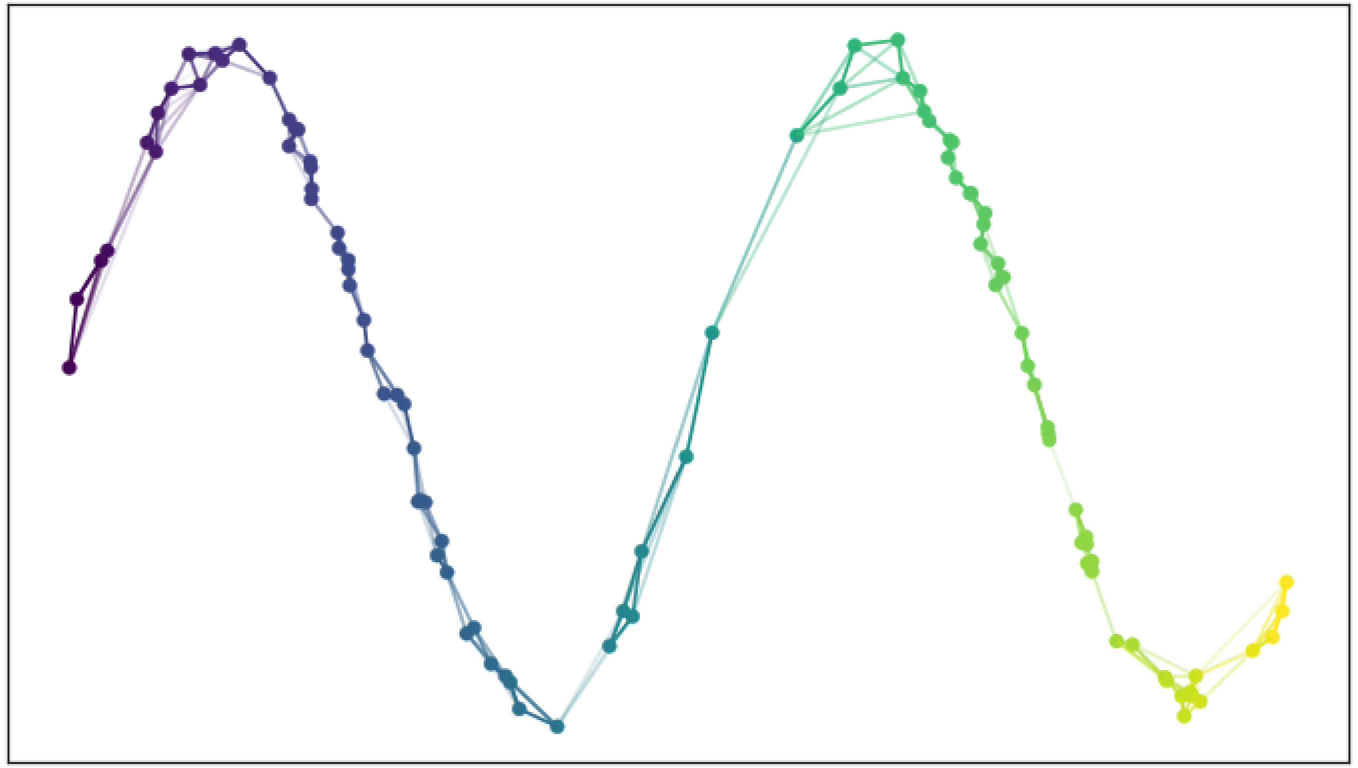
Finally, UMAP uses a different loss function. **Binary Cross-Entropy.** This follows likelihood methods, trees, etc.

$$CE(X,Y) = \sum_i \sum_j \left[p_{ij}(X) \log\left(\frac{p_{ij}(X)}{q_{ij}(Y)}\right) + (1 - p_{ij}(X)) \log\left(\frac{1 - p_{ij}(X)}{1 - q_{ij}(Y)}\right)\right]$$

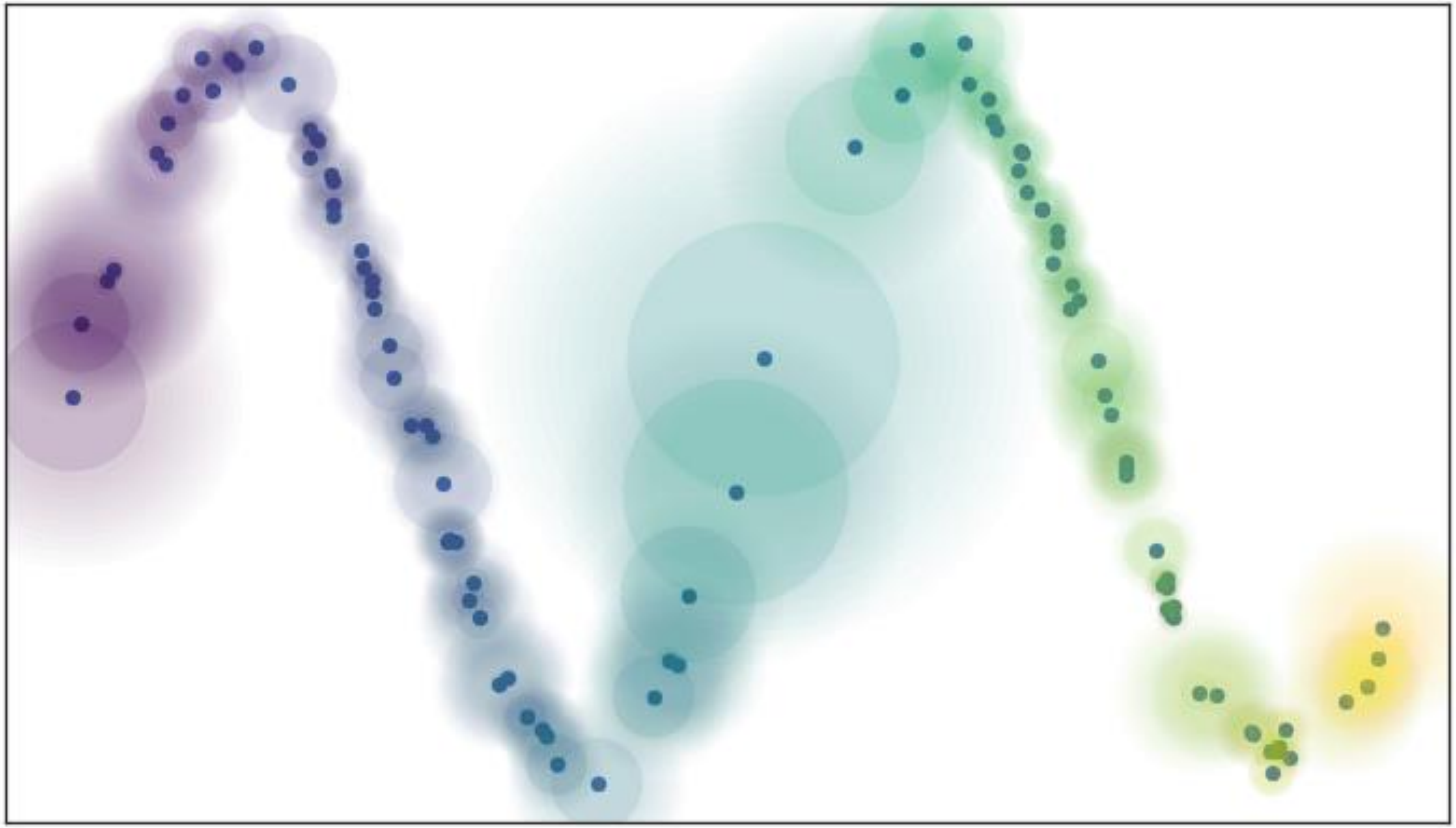
Also, and finally, UMAP initializes the mapping using spectral clustering instead of



Nearest neighbours without grouping.



Nearest neighbours with grouping.



Fitting a fuzzy distribution around each point

UMAP in Practice

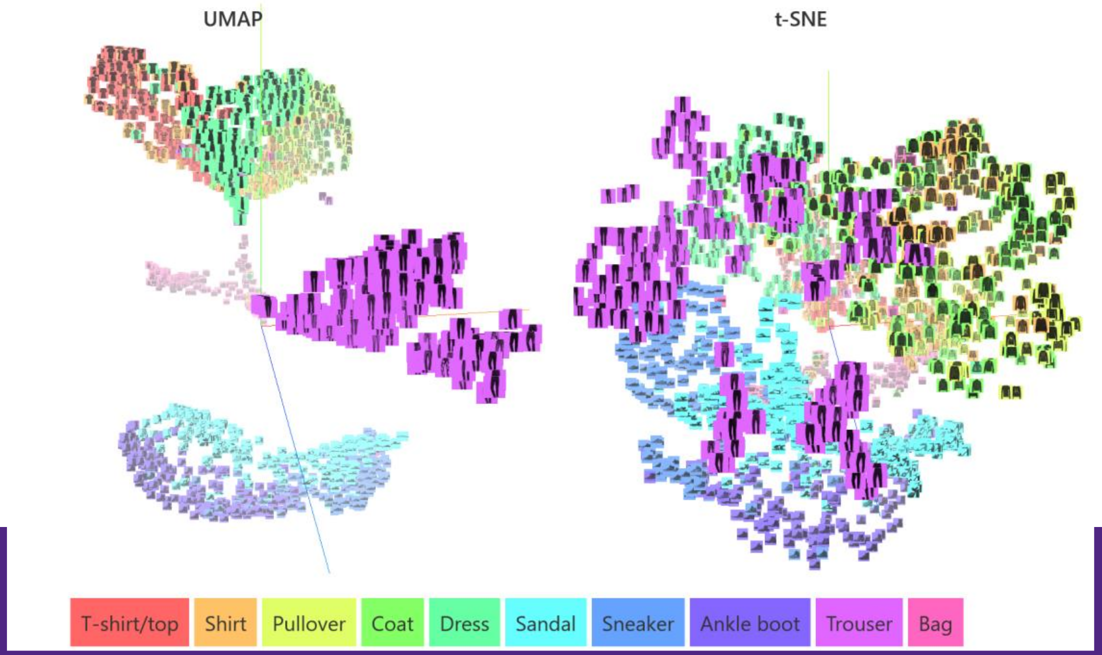
As with t-SNE, there are two very important parameters

- The number of nearest neighbours.
- The minimum distance.

And also make sure to use the correct distance for your problem!

Determining the optimal values is up to you. Check the coursework to study its effects!

- Experiment in this site: https://pair-code.github.io/understanding-umap/



Takeaways

Manifold methods work by compressing the space in a non-linear manner.

As such, they can be arbitrary and require careful selection of parameters.

Two methods: t-SNE and UMAP.

UMAP is better grounded in theory and more efficient, but less accepted than t-SNE.

t-SNE is only good for plotting in two or three dimensions, use UMAP for more.