Indian Institute of Technology, Kharagpur

UNDERSTANDING UMAP

UNIFORM MANIFOLD APPROXIMATION AND PROJECTION

UMAP is a new dimensionality reduction technique that offers a number of advantages over t-SNE, most notably **increased speed** and **better preservation of the data's global structure**.
UMAP, at its core, works very similarly to t-SNE - both use graph layout algorithms to arrange data in low-dimensional space.
In the simplest sense, UMAP

1. Constructs a high dimensional graph representation of the data
2. Optimises a low-dimensional graph to be as structurally similar as possible.

2

STEP 1: CALCULATION OF SIMILARITY SCORES

The first thing that UMAP does is calculate the distance between each pair of high dimension points. After this, the similarity scores are calculated by the formula:

$$p_{i|j} = e^{-\frac{d(x_i, x_j) - \rho_i}{\sigma_i}}$$

$d(x_i, x_j)$=Distance (need not be Euclidean distance) between points i and j
ρ_i=Distance of point i from its Nearest Neighbour
The value of $\boldsymbol{\sigma}_i$ is chosen such that
$\Sigma p_{i|j}$ (for j=1 to n)=$\log_2(n)$, where n=no. of nearest neighbours.
UMAP scales the curve so that regardless of how close or far the neighbouring points are, the sum of similarity scores will be equal to $\log_2$(number of nearest neighbours).

Formula used in t-SNE:

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

A significant difference in the formula for UMAP and t-SNE is that the expression in t-SNE formula has a summation in the denominator which is used for normalisation. This does not happen in UMAP so the computational time of UMAP is less than t-SNE.

STEP 2: SYMMETRIZATION OF SIMILARITY SCORES

3

Since, there should be a unique similarity score for a pair of points, symmetrization is applied to $p_{i|j}$ and $p_{j|i}$ by using the formula given below.

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

These symmetric similarity scores are used for deciding the order in which points are chosen to be moved to form a cluster. Pair of points having higher symmetric similarity score are given preference.

STEP 3: LOW DIMENSIONAL(LD) SIMILARITY SCORES

For calculating the low dimensional similarity scores of a pair of points, the formula given below is used :

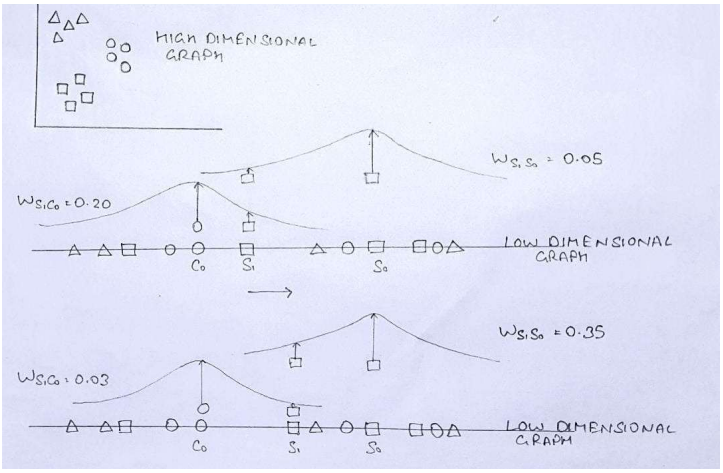
$$q_{ij}=1/(1+a|y_i-y_j|^{2b})$$

The UMAP defaults use min_dist = 0.1, spread = 1, which results in a=1.577 and b=0.8951. If you use min_dist = 0.001, spread = 1 then you get the result for a=1.929 and b=0.7915.

HOW ARE LOW-DIMENSIONAL (LD) SIMILARITY SCORES USED TO FORM CLUSTERS?

LD (low dimensional) similarity scores are used to know whether or not a point is in a proper position in the LD graph.
If a pair of points are **neighbours in the HD graph**, then they **should have high LD similarity scores**. So, after the LD graph is initialised UMAP brings neighbour points close to each other and this is judged by their similarity scores, that is, if their LD similarity score is increasing the points are getting closer. Similarly for pairs of **points that are not neighbours**, UMAP tries to **reduce their LD similarity scores** and this is how clusters are formed.

4



In this example graph, dimension is reduced from 2 to 1. Here, three clusters are there in the higher dimension and the lower dimension is assumed to be initialised. If the square S_1 is taken for example to be moved in the low dimension, then it should be moved towards square S_1 and away from circle C_0. This is what happens when we move the square S_1 towards right and this can be mathematically confirmed as the low dimensional similarity scores of the two squares increase and that of the circle and square decrease. UMAP repeats this step for all the points and a number of points to form the clusters in the lower dimension.

5

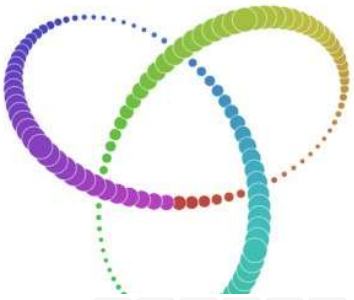
TWO IMPORTANT PARAMETERS OF UMAP

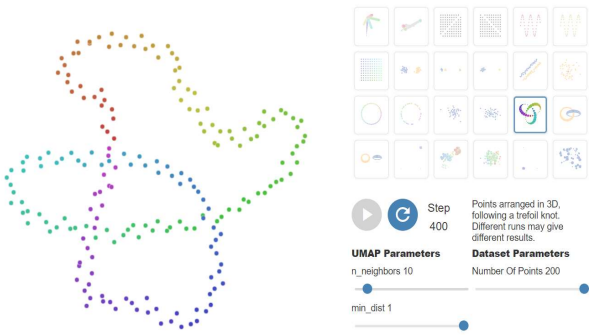
The two most commonly used parameters: n_neighbors and min_dist, which are effectively used to control the balance between local and global structure in the final projection.

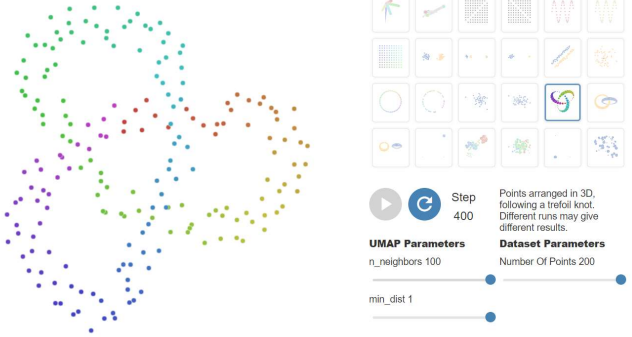
NUMBER OF NEAREST NEIGHBOURS

The number of approximate nearest neighbours is used to construct the initial high-dimensional graph. Low values will push UMAP to focus more on local structure by constraining the number of neighbouring points considered when analysing the data in high dimensions, while high values will push UMAP towards representing the big-picture structure while losing fine detail.

On the right side, a 3D plot of a trefoil knot is given and this knot is reduced to a 2 dimensional plot, keeping the minimum distance constant and changing the number of nearest neighbours. The effect of lower and higher number of nearest neighbours can be seen clearly in the two graphs given below.





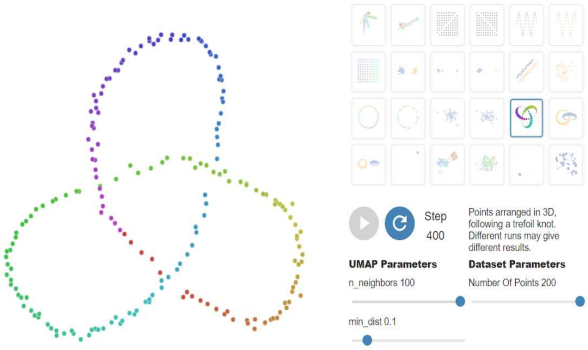


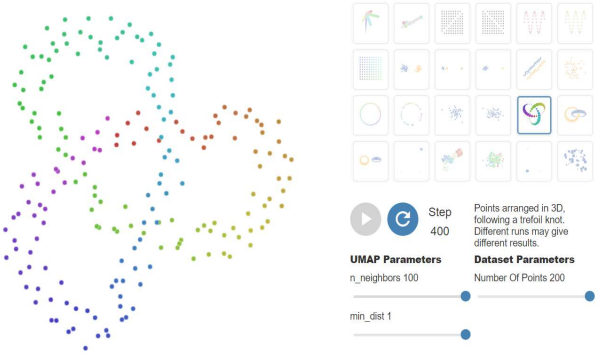
6

MINIMUM DISTANCE (between points in low-dimensional space)

This parameter controls how tightly UMAP clumps points together, with low values leading to more tightly packed embeddings. Larger values of min_dist will make UMAP pack points together more loosely, focusing instead on the preservation of the broad topological structure.

tSNE algorithm is applied to the same 3D image but now in the plots obtained, the n_ neighbour is set to be 100 in both but the min dist is 0.1 in one of them and 1 in the other.
So when min_dist is small, the points are quite close to each other and the plot is compact whereas when the min_dist is higher the plot looks clearer and better.





7

COMPARISON OF TWO COST FUNCTIONS AND THEIR EFFECT ON LOCAL AND GLOBAL STRUCTURE

(Explanation for why t-SNE can't preserve global structure whereas UMAP can)

COST FUNCTION OF t-SNE

The cost function of t-SNE is KL divergence function.

Now, here when X is small or when X is large the first part of the cost function is zero in both the cases.

1) $$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$
$$= \sum_i \sum_j p_{ij} \log p_{ij} - p_{ij} \log q_{ij}.$$

2) $$P(X) \approx e^{-X^2} \qquad Q(Y) \approx \frac{1}{1+Y^2}$$

3) $$KL(X,Y) \approx -P(X) \log Q(Y) = e^{-X^2} \log(1+Y^2)$$

Therefore, expression 3 is obtained. When X is small, the exponential part will be approximately 1, so $\log(1+Y^2)$ is obtained. Now to make the cost function close to zero, Y should be as small as possible. So when X tends to 0, Y also tends to 0 and this is how local structure is preserved in tSNE.

Analysis of the global structure:

Now if x tends to infinity, then again the first part of the KL function will be zero as discussed earlier and only the expression 3 is left. Now if X is large then the exponential part will be approximately 0. So irrespective of the value of Y the cost function or the KL function will be near about 0. Hence, when X is large, Y can have any value between 0 to infinity but still the cost function will be approximately 0.

This is the reason tSNE can't preserve the global structure as it doesn't ensure that Y tends to infinity when X tends to infinity.

8

COST FUNCTION OF UMAP:

Analysis of the cost function of UMAP:
It is a binary cross entropy cost function.
Here also a similar approach as of tSNE will be taken. If x tends to 0, then the expression reduces to (5) and here again Y should be approximately 0 to make the cost function 0. So this way UMAP also preserves local structure. Now when x tends to infinity our expression reduces to (6), now here Y should also tend to infinity in order to reduce the cost function and make it approximately 0. Hence, UMAP ensures that when X tends to infinity Y also tends to infinity.

So in this way UMAP preserves the global structure unlike tSNE which can't preserve the global structure.

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

5) $$X \to 0 : CE(X,Y) \approx \log(1 + Y^2)$$

6) $$X \to \infty : CE(X,Y) \approx \log\left(\frac{1 + Y^2}{Y^2}\right)$$

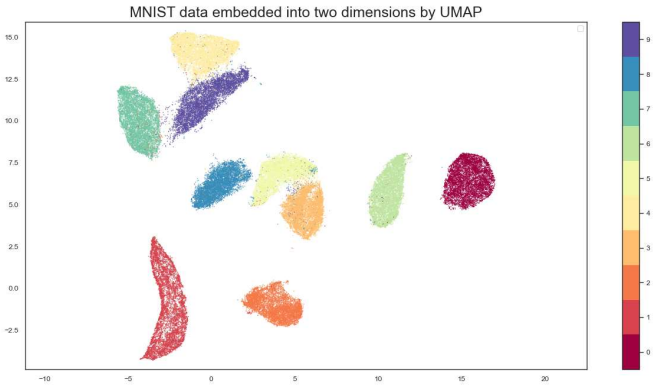
9

CODE FOR UMAP

```
import umap.umap_ as umap
from sklearn.datasets import fetch_openml
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
sns.set(context="paper", style="white")
mnist = fetch_openml("mnist_784",
version=1)

reducer = umap.UMAP(random_state=42)
embedding = reducer.fit_transform(mnist.data)

fig, ax = plt.subplots(figsize=(12, 10))
color = mnist.target.astype(int)
plt.scatter(embedding[:, 0], embedding[:, 1], c=color, cmap="Spectral", s=0.1)
plt.gca().set_aspect('equal', 'datalim')
plt.colorbar(boundaries=np.arange(11)-0.5).set_ticks(np.arange(10))
plt.title("MNIST data embedded into two dimensions by UMAP", fontsize=18)
plt.legend()
plt.show()
```



10

REFERENCES

https://www.google.com/url?sa=t&rct=j&q&esrc=s&source=web&cd&cad=rja&uact=8&ved=2ahUKEwjVxpPp1P34AhXi8DgGHWnWBCsQFnoECAQQAQ&url=https%3A%2F%2Fpair-code.github.io%2Funderstanding-umap%2F&usg=AOvVaw2ixAa8WAmamc7rCflqDg4P&hl=en_GB

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

https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668

TEAM MEMBERS:

- ADITYA KRISHNA DAS
- YASASWANI RONGALI
- YASH KUMAR