ICCS481: Report

Kriangsak Thuiprakhon
kriangsak.thi@student.mahidol.edu

1 Selection of Techniques

The previous sections gave a brief introduction to UMAP and t-SNE. Here, we will offer a, hopefully, a guide for which methods to use given a certain types of data points with some constrains. For instance, if one is to visualize a given high dimensional data points such that the dimensions is reduced down to, say, 3, with global structure preserved, which of the two one shall pick. In this section, a comparison of the two methods and suggestions for selection of methods will be presented.

1.1 Comparison between t-SNE and UMAP

t-SNE has been around since 2008 and had been used very commonly for the Single Cell Genomics and Data Science. Yet, in 2019, UMAP paper was released to beat what t-SNE has to offers. UMAP, since then, is now predominantly used in Data Science. Here are a few reasons why t-SNP is to put, almost, total stop after UMAP has been born.

- t-SNE is unable to scale efficiently when the sample sizes is too large. Though, FItSNe was given rise to accelerate the performance of the original t-SNE, it is too heavy on memory.
- t-SNE **does not** preserve globacl structure. In other words, distances between clusters is not guaranteed meaningful.
- t-SNE only embeds data into, mostly, 2 or 3 dimensions. This is great only for data visualizations, as it is hard for t-SNE to work for a general dimensionality reduction. With that being said, is is not practical to use t-SNE as a means for data pre-processing for further data analysis.
- t-SNE does not leverage features as opposed to PCA.
- t-SNE is takes up to much memory for its computations. This is obvious when it uses a large perplexity hyperparameter. This problem still persists on any version of t-SNE.

The *key* differences between the two methods are as follows:

1. as discussed in the previous section, UMAP uses ***exponential probability distribution*** in high dimension with flexibility that any distance can be plugged in: not necessarily Euclidan distances as required in t-SNE. Moreover, the probabilities are not normalized. the conditional probability is defined as follows:
$$p_{i|j} = e^{\frac{-d(x_i,x_j)-\rho_i}{\sigma_i}}$$
The key to to local connectivity preservation is the parameter ρ since it represents the distance from each i-th data point to its first k-NN. This important parameter bridges section two and three in the original paper: connecting the fancy topological data analysis to the algorithmic implementation.

2. ***Absence of normalization***
Due to its functional form of the high or low dimensional probabilities are already scaled. This absence results in dramatic reduce in time complexity. For instance, as conditional probability construction in t-SNE, having to normalize the probability means that there is a need for approximation for the summation or integration which are usually computationally expensive.

3. As opposed to using perplexity, UMAP replaces it with the ***number of nearest neighbors*** where the number **k** is defined in a similar manner as is for t-SNE perplexity but *without* the log function.

4. ***Different symmetrizations*** t-SNE applies Gaussian probability to the high-dimensional space in order to satisfy symmetry constraint. However, UMAP symmetrize the probability as follows:

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

   This step is mandatory since it can happen that the the weight of the graph between A and B node does not equal that of B and A nodes. Benefits to this specific polymerization has bot been observed.

5. While t-SNE uses of t-student distribution for low-dimensional distance modeling, UMAP adopts the concept of the family of curves with, again, no normalization applied: i.e.,

$$\Phi(\mathbf{x}, \mathbf{y}) = \left(1 + a\left(\|\mathbf{x} - \mathbf{y}\|_2^2\right)^b\right)^{-1}$$

   the hyperparameters **a** and **b** for such equation are obtained by non-linear least-square fitting. The parameter **b** has a string influence on the family of curves. In addition, $min_distant$ plays a vital role in obtaining tightly packed clusters in UMAP dimensionality reduction. This is because, by plotting using a few parameters different from that being used in UMAP, it implies that anything below UMAP min_dist parameter, all data points will be equally tightly connected. This also means that the assignment to data points in low dimensional coordinates is nearly the same.

6. UMAP uses ***binary cross-entropy (CE)*** as its cost function while KL-divergence is used in t-SNE the second term that is sought to minimized in UMAP is what preserves ***global structure*** of the given data points.

7. Different low-dimensional coordinate construction.UMAP uses ***Graph Laplacian*** to construct low-dimensional coordinates based on the theory that this is equivalent to KL-divergence minimization suggested by Linderman and Steinerberger. This construction makes UMPA more stable from run to run as randomization is not used, unlike t-SNE's random normal initialization.

8. UMAP uses ***Stochastic Gradient Descent (SGD)*** as opposed to a normal GD used in t-SNE. This, as discussed in our lectures, boosts up the computation with less memory required.

PRESERVATION OF GLOBAL AND LOCAL STRUCTURES

- T-SNE PRESERVES ONLY LOCAL STRUCTURE

  In t-SNE, thre are a few different approaches to understanding its locality. we will be presenting you the analysis looking at KL-divergence. When plotted, KL is a weird looking function, it is no where near symmetry. The original paper sought to minimize Y, the Euclidean distance in low dimension, at small X, distance in higher dimension. This is because at a large high-dimensional distance KL goes to zero. This means that when X is small, KL will behave almost entirely according to Y. However, at a large X, the distance projected onto a lower dimension is not guaranteed to be be large (i.e., Y can be either large or small). This implies that it there might exist points closely mapped in a lower dimension while they are distant in higher dimensions. Though t-SNE does guarantee that points close to each other will resemble in a lower dimensions.

- HOW DOES UMAP PRESERVE GLOBAL STRUCTURE ?

  In a super brief explanation, UMAP uses different a cost function from t-SNE. KL-divergence is used in t-SNE, UMAP uses Cross-Entropy (CE). This allows UMAP to capture both local and global structure of the input data. This happens as a result of CE as a cost function. That is, when we take the limit approaching large and small values, distances in both higher and lower dimensions have to be of the same fashion magnitude. That is to say, to minimize the cost function, points close together in higher dimensions will be projected onto a lower dimension with preserved locality. The same applies to points distant in higher dimensions, points far apart in higher dimensions will stay distant in lower dimensions.

- WHAT MAKES UMAP FASTER THAN t-SNE UMAP can be used when it comes to data sets with a large number of data points, for general dimension embedding and when the a data set contains large ambient dimensions. Both UMAP and t-SNE proceed in the same direction. That is, they initialize a high dimensional representation of the data and apply binary searches for the desired variances . The second step is to optimize low dimensional representation via GD. Both do not run run multi-threading but sequential manner. However, there are a few remarks that explain the speedy computation in the younger dimensionality reduction algorithm, UMAP.
  - In the first step, constructing high dimensional graphs, computations in UMAP drop out log function for how it defines parameter k. In addition, the absence of normalization, as stated earlier, is also a key. These are the two important factors dramatically fasten up the computations required in UMAP.
  - The optimization step in UMAP is also faster with more efficient techniques are used. First, UMPA uses SGD as opposed to GD. There is also an absence of normalization in lower dimensions. Also, since there is no normalization, NN-descent can be done for a general dimension embedding. Lastly, and most importantly, UMAP introduces the **local connectivity ρ parameter** this is used to solve sparsity in a data set. This vital parameter ρ captures isolated points in the data set: this is known as locally broken manifold. With this parameter, UMAP glues together the data set with dense and sparse data points allowing it to be highly flexible for dimensionality reduction on a very high dimensional data set, without the need of reprocessing steps (i.e., PCA).

All in all, UMAP was born to conquer t-SNE in many aspects. It started since how a very abstract branch of mathematics is applied in the setting and used to glue together the points in order for them to belong to the same CATEGORY. The maths behind UMAP takes way more time to be understood compared to t-SNE. With also a different fashion of computations, UMAP becomes a much faster algorithm for dimensionlity reduction. It should be spontaneous to choose UMAP as a prime method for dimensional reductions, to say the least.