Review article of t-SNE *vs* UMAP

Suchanun Piriyasatit, Suchanuch Piriyasatit, and Kriangsak Thuiprakhon

Mahidol University International College, Nakhon Pathom, Thailand

1 Introduction

The current harvesting of data goes on to the level that it is practically impossible to be interpreted without an aid of visualizations techniques. This report will give a brief introduction to the two visualization techniques: t-Stochastic Neighbor Embedding and Uniform Manifold Approximation and Projection for Dimension Reduction. Particularly, the two techniques try map a set of data points in high dimensions into one with lower dimensions (i.e., 1, 2 or 3 dimensions). In fact, we will (1) give a summary of t-SNE and UMAP, (2) present the selection among the two techniques with regards to the nature of the given data set, (3) present the implementation of a parallelized algorithm of UMAP, (4) analyze the actual running time of UMAP, as oppose to the presented empirical time complexity in the original paper.

2 t-Stochastic Neighbor Embedding (t-SNE)

Goal: Convert high-dimensional datapoints $X = \{x_1, x_2, \ldots, x_n\}$ into two or three-dimensional $Y = \{y_1, y_2, \ldots, y_n\}$ that correctly model similar and dissimilar points.

2.1 SNE

Procedures:
1. Convert high-dimensional Euclidean distances between datapoints in X into conditional probabilities which represents similarities, $p_{j|i}$

Def: (In high dimension) $p_{j|i}$ is the probability that x_i will pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i.

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

σ_i = the variance of the Gaussian centered at x_i
We set $p_{i|i} = 0$ (because we are only interested in pair-wise data)

2. Given any low-dimensional counterparts Y, we can compute similar conditional probabilities, $q_{j|i}$

Def: (In low dimension) $q_{j|i}$ is similar to $p_{j|i}$ but we set the Guassian's variance $\sigma_i = \frac{1}{\sqrt{2}}$.

$$q_{j|i} = \frac{\exp\left(-\left\|y_i - y_j\right\|^2\right)}{\sum_{k \neq i} \exp\left(-\left\|y_i - y_k\right\|^2\right)}$$

We set $q_{i|i} = 0$ (because we are only interested in pair-wise data)

observation: If Y is a good mapping of X, $p_{j|i}$ should be equal to $q_{j|i}$. So we want to minimize the mismatch between $p_{j|i}$ and $q_{j|i}$.

3. Use KL-divergence to measure the information loss when we use $q_{j|i}$ to approximate $p_{j|i}$. KL-divergence is simply the expected value of the log-ratio of the two distribution P and Q:

$$Cost = \sum_i KL\left(P_i \| Q_i\right) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

What is KL-divergence?
...... **observation of cost function:**
KL-divergence is not symmetric so ....
How to choose Guassian's variance σ_i for $p_{i|j}$?
idea:
1. We *pick* a perplexity. (see below)
2. Perform a binary search for σ_i that produces a P_i with the same perplexity. (perplexity increases monotonically with the variance σ_i)

def: the entropy of P_i (measured in bits) is a measurement of randomness of P_i

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

(high entropy $\rightarrow$ messy data)

def: Perplexity of P_i tells us the effective number of neighbors,

$$Perp(P_i) = 2^{H(P_i)}$$

4. Use gradient descent method to minimize the cost function in 3. The gradient of the cost function is:

$$\frac{\delta C}{\delta y_i} = 2 \sum_j \left(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j}\right)\left(y_i - y_j\right)$$

The gradient update is

$$\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) \left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)} \right)$$

η = learning rate
$\alpha(t)$ = momentum at iteration t (need this term to avoid poor local minima and to speed up the optimization.)

2.2 Symmetric SNE

Instead of minimizing the Kullback-Leibler divergences between the condi- tional probabilities $p_{j|i}$ and $q_{j|i}$, we can minimize a single Kullback-Leibler divergence between a joint probability distribution, P, (in the high-dimensional space) and a joint probability distribution, Q (in the low-dimensional space)

What is the difference?
- in a joint probability distribution of P, $p_{ji} = p_{ij}$.
- in a joint probability distribution of Q, $q_{ji} = q_{ij}$.
(unlike in the conditional probability where $p_{i|j}$ is not neccessary equal to $p_{j|i}$)

Why?
- will give a simpler form of gradient.
- produce a map Y in low-dimensional space that is just as good, sometimes a little better.

Our cost function of KL-divergence then is

$$Cost = KL(P \| Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

2.3 Problems with SNE

- Need extra computation time to get good result.
- Hard to optimize. (a lot of exponentials in gradient)
- **Crowding problem** = ...

3 UMAP

Theoretical basis: ...
UMAP constructs a weighted k-neighbor graph to represent a high-dimensional data and compute a structurally similar low-dimensional graph.

3.1 High-Dimensional Graph Construction

A weighted k-neighbor graph is computed. Given the following input, dataset $X = \{x_1, \ldots, x_N\}$, metric or dissimilarity measure $d : X \times X \to R_{>0}$, and an integer k. For each x_i, compute the set $\{x_{i_1}, \ldots, x_{i_k}\}$ of its k nearest neighbors using the metric d.

We then construct a weighted directed graph from pairs of nearest neighbors, $\bar{G} = (V, E, w)$, where V is the set of X and E is the set of pairs of nearest neighbors, $\left\{\left(x_i, x_{i_j}\right) | 1 \leq j \leq k, 1 \leq i \leq N\right\}$ and the weight function for each x_i describing the similarity as

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

$$w((x_i, x_k)) = 0, \forall x_k \notin \text{ k-nearest neighbor of } x_i$$

where

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

and σ_i to be the value such that the sum of the weights sum up to $\log_2(k)$, i.e.,

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

3.2 Low-Dimensional Graph Construction

UMAP initializes low-dimensional data Y using symmetric normalized Laplacian method ....
It uses stochastic gradient descent to minimize a cross entropy loss function.

Definition: Fuzzy Set Cross Entropy Function

$$C(X, Y) = \sum_i \sum_j \left[w((x_i, x_j)) \log\left(\frac{w((x_i, x_j))}{\Phi(y_i, y_j)}\right) + (1 - w((x_i, x_j))) \log\left(\frac{1 - w((x_i, x_j))}{1 - \Phi(y_i, y_j)}\right)\right]$$

where Φ is an approximation of similarity weight between two points defined as follows

Definition: $\Phi : R^d \times R^d \to [0,1]$ as an approximation of similarity weight between two points in R^d, as

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

where a and b are chosen by non-linear least squares fitting such that $\Phi(\mathbf{x},\mathbf{y}) = \Psi(\mathbf{x},\mathbf{y})$, where

$$\Psi(\mathbf{x},\mathbf{y}) \begin{cases} 1 & \text{if } \|\mathbf{x}-\mathbf{y}\|_2 \leq \text{ min-dist} \\ \exp\left(-\left(\|\mathbf{x}-\mathbf{y}\|_2 - \text{ min-dist }\right)\right) & \text{otherwise} \end{cases}$$

4 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.

5 Conclusion