t-SNE & UMAP

Liam McDevitt
lm15ue@brocku.ca

Department of Computer Science
Brock University
COSC 5P77 Presentation

July 7, 2022

Overview

Introduction

- t-SNE (2008) [1] and UMAP (2018 - 2020) [2] are competing machine learning algorithms for understanding and visualizing high dimensional datasets onto, most often, scatter plots.
- In many problem domains, the ability to visualize high-dimensional data is a great asset.
- As an example, single-cell transcriptomics or single-cell RNA sequencing (scRNA-seq) has been a quite popular problem to visualize since now from millions of cells we can capture data expressing an individual gene [3].

Introduction (cont.)

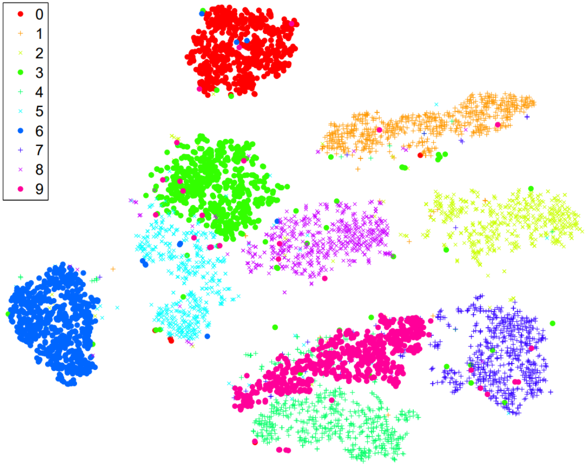

Figure 1: From van der Maaten & Hinton (2008): A t-SNE visualization of 6000 handwritten digets from the MNIST data set [1].

Dimensionality Reduction

- Core tool behind being able to visualize and understand high dimensional data [1], [2].
- A technique where we attempt on reducing the number of dimensions in the problem to 2 or 3 dimensions [1], [2].
- When reducing the number of dimensions in the problem we want to preserve the structure of the data to ensure its integrity (i.e. keeping distant points distant and keeping close neighbours close [1], [2].

t-SNE

- t-distributed stochastic neighbour embedding (t-SNE) was developed by van der Maaten and Hinton in 2008 [1].
- Recent improvements have been made by using upgraded versions of t-SNE such as Fit-SNE (2019) [4] for a faster implementation and Principal Component Analysis (PCA) [5], [6] for initialization as well as many other little beneficial tweaks to ensure it is still competitive with new similar techniques popping up in recent years.

t-SNE (cont.)

- t-SNE is an upgrade on the Stochastic Neighbour Embedding technique developed by Hinton and Roweis in 2002 [7].
- One problem the upgrade helps to solve is the Crowding Problem. The Crowding Problem comes from trying to map high-dimensional data into a lower dimensional map. Since the amount of space for the visualization is so small it is hard to represent distant and close points creating a tendency for the points to cluster in the center [1].
- The reason for the "t" in the name is because a heavy-tailed t-distribution with one degree of freedom (Cauchy distribution) is used over a Gaussian in the low-dimensional space when calculating distances between points [1].

t-SNE (cont.)

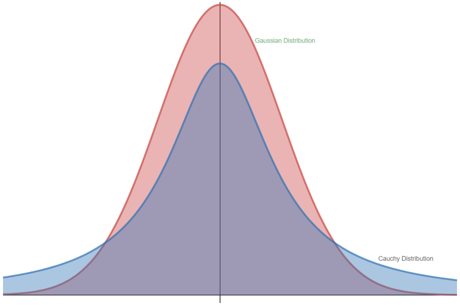

Figure 2: Gaussian Distribution vs. Cauchy Distribution.

- Using this distribution helps model distances from the high-dimensional to low-dimensional map helping to more accurately represent distances between data points [1].
- In addition, t-SNE changes up the cost function from SNE to a symmetrized version to help with optimization and the crowding problem in SNE [1].

t-SNE (cont.)

- Perplexity is t-SNE's most important parameter [1].
- The Gaussian kernel width is determined by this parameter [1].
- This width dictates the number of nearest neighbours a specific datapoint is attracted to [1].

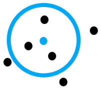

Figure 3: Perplexity radius example.

- Usually, the perplexity value is set to 30 or 50 [3]. The original paper says that is can typically be between 5 and 50 but changes in perplexity drastically effect the visualization [1].

t-SNE (cont.)

- The t-SNE algorithm contains 3 main steps [1]:

  1. Calculate the joint probability P in the high-dimensional space signifying how likely datapoint x_i would select x_j as a neighbour and vice a versa.

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{1}$$

  2. Calculate the joint probability Q in the low-dimensional space signifying how likely datapoint y_i would select y_j as a neighbour and vice a versa.

$$q_{ij} = \frac{exp((-||y_i - y_j||)^2)}{\sum_{k \neq l} exp(-||y_k - y_l||)^2} \tag{2}$$

  3. Use gradient descent to minimize a single Kullback-Leibler divergence between a joint probability distribution P and Q.

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{3}$$

UMAP

- Uniform Manifold Approximation and Projection (UMAP) is a new (updated again in 2020) dimensionality reduction visualization technique (similar to t-SNE) by McInnes et al. originally in 2018 [2].
- UMAP creates two graphs, one for the high-dimensional data, one for the low-dimensional data, and then optimizes the similarity (cross-entropy) between the two so the low-dimensional graph is as close to matching the high-dimensional one as possible [2].
- Each of the high-dimensional and low-dimensional topological structures (graphs) are put together by UMAP using local manifold approximations which combines the approximations' local fuzzy simplicial sets [2].
- In essence, the fuzzy simplicial complex is a weighted graph where each edge has a weight corresponding to how likely the points are connected to one another [2].

UMAP (cont.)

- In determining how connected our graph representations are UMAP develops an increasing radius from each point connecting different points together when their radii interact [2].
- To combat against too large, where everything will be connected together, and too small, where there will be many detached clusters, UMAP decides on its radius size for each point based on how far away it is from each of its closest neighbours [2].
- From these connections between points the graph becomes "fuzzy" by decreasing the likelihood points are connected the farther they are way from each other. i.e., as the radius expands it gets "fuzzier" decreasing the chances that point will form a cluster with others further out [2].

UMAP (cont.)

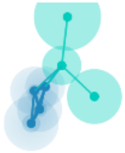

Figure 4: Example of connecting points in UMAP by Coenen & Pearce. This snapshot is from Understanding UMAP.

- UMAP wants to preserve local and global structure as much as possible and this is further helped by making sure, as the radius of each point expands, each point is at least connected to its closest neighbour [2].

UMAP (cont.)

- There are four hyper-parameters that dictate the UMAP algorithm's learning process [2]:
  1. n, the number of neighbours when initializing the high-dimensional graph.
  2. min-dist, the distance given between points in the low-dimensional embedding space.
  3. n-epochs, the number of epochs to use for training when optimizing the low-dimensional graph.
  4. d, the dimension we're embedding to. i.e. 100-dimensions to two-dimensions for visualization purposes.

Website Examples

- t-SNE visualization examples created by Wattenberg et al. in 2016 [8] on Olah & Carter's platform Misreading t-SNE.
- UMAP visualization examples powered by umap-js on Coenen & Pearce's platform Understanding UMAP. The mammoth visualization idea was from Max Noichl and the Smithsonian Institute supplied the 3D mammoth model.

Learning Materials

- t-SNE
  1. https://distill.pub/2016/misread-tsne/
  2. https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding
  3. https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1
- UMAP
  1. https://pair-code.github.io/understanding-umap/
  2. https://pair-code.github.io/understanding-umap/supplement.html
  3. https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

Questions

Questions?

Bibliography

[1] L. van der Maaten and G. Hinton, “Viualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, Nov. 2008.

[2] L. McInnes, J. Healy, and J. Melville, “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction,” en, *arXiv:1802.03426 [cs, stat]*, Sep. 2020, arXiv: 1802.03426. [Online]. Available: http://arxiv.org/abs/1802.03426 (visited on 02/22/2021).

[3] D. Kobak and P. Berens, “The art of using t-SNE for single-cell transcriptomics,” en, *Nature Communications*, vol. 10, no. 1, p. 5416, Dec. 2019, ISSN: 2041-1723. DOI: 10.1038/s41467-019-13056-x. [Online]. Available: http://www.nature.com/articles/s41467-019-13056-x (visited on 02/15/2021).

Bibliography (cont.)

[4] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, and Y. Kluger, "Fast Interpolation-based t-SNE for Improved Visualization of Single-Cell RNA-Seq Data," *Nature methods*, vol. 16, no. 3, pp. 243–245, Mar. 2019, ISSN: 1548-7091. DOI: 10.1038/s41592-018-0308-4. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6402590/ (visited on 02/22/2021).

[5] J. Lee, D. Peluffo, and M. Verleysen, "Multiscale stochastic neighbor embedding: Towards parameter-free dimensionality reduction,", Apr. 2015.

[6] C. de Bodt, D. Mulders, M. Verleysen, and J. A. Lee, "Perplexity-free t-SNE and twice Student tt-SNE," en, *Computational Intelligence*, p. 6, 2018.

[7] G. Hinton and S. Roweis, "Stochastic Neighbor Embedding," en, p. 8, 2002.

[8] M. Wattenberg, F. Viégas, and I. Johnson, "How to use t-sne effectively," *Distill*, 2016. DOI: 10.23915/distill.00002. [Online]. Available: http://distill.pub/2016/misread-tsne.