CSE 519: Progress Report
Evaluating Dimension Reduction Methods

1. Introduction

Dimensionality reduction has a key role to play in Data Science, mainly when it comes to data visualisation and data preprocessing. The core idea of dimensionality reduction is to find ‘latent’ features in the data, by producing low dimensional representations of high dimensional data. The greater the number of features, the more difficult it becomes to visualize the data. Almost all scientific studies nowadays face an explosion in the number of variables and dealing with such high dimensional data becomes a problem. Dimensionality reduction makes it possible to visualize such data on a plot, removing redundant features and noise. Moreover, dimensionality reduction also aids in avoiding overfitting in data. Also, all machine learning algorithms are employed on massive data samples where it would be beneficial to exclude the redundant features, as less dimensions mean less storage space and computing.

The objective of this project is to devise a way to qualitatively and quantitatively rate a dimensionality reduction model on how ‘good’ it is. This would include analysing the tradeoffs of different dimensionality reduction algorithms and reach a point where we can tell which algorithm is better than the other backed with analytical evidence.

2. Analysis of Popular Dimensionality Reduction Algorithms

A few popular dimensionality reduction algorithms are PCA, ISOMAP, MDS, t-SNE, UMAP and Autoencoder. In the context of our project, we have analysed in detail the algorithms Principle Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP).

There are basically two approaches in which dimensionality reduction is performed, by applying matrix factorisation (PCA uses this) or by the neighbour graph approach (t-SNE and UMAP use this).

Principal Component Analysis (PCA) is a linear dimensionality reduction technique that can be viewed as an algorithm trying to find a way to reconstruct the data as a linear combination of a small number of prototypes.

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear dimensionality reduction technique that can be seen as a graph based algorithm where a low dimensional representation is built, that gives a graph that matches as closely as possible to the graph built in high dimensional case. It uses the KL divergence loss function to project the high dimensional probability to low dimensional probability and then minimizes this loss using the gradient descent optimisation.

Uniform Manifold Approximation and Projection (UMAP) is a t-SNE like non-linear dimensionality reduction technique with a more firm mathematical foundation. It uses the binary cross-entropy loss function instead of the KL divergence function as in t-SNE. The additional second term of this loss function enables UMAP to capture the global structure of the data which t-SNE cannot. Also, unlike t-SNE, UMAP does not apply normalisation to the probability distribution of the low or high dimensional points. This absence of normalisation, drastically reduces the computation time.

High Level Analysis of PCA, UMAP and t-SNE

1. The benchmarking done for UMAP consists of a detailed comparative study of the performance of different dimensionality reduction methods and how their performance is affected by increasing dataset size. PCA, UMAP and multicoreTSNE were applied on the full MNIST digit dataset in batches and the time they took to complete were noted and represented graphically. As we can see from Fig.2.1. The multicore implementation of TSNE increases exponentially. Even though UMAP is visibly slower than PCA its performance is much better than TSNE and the difference just keeps increasing as the dataset size gets bigger.

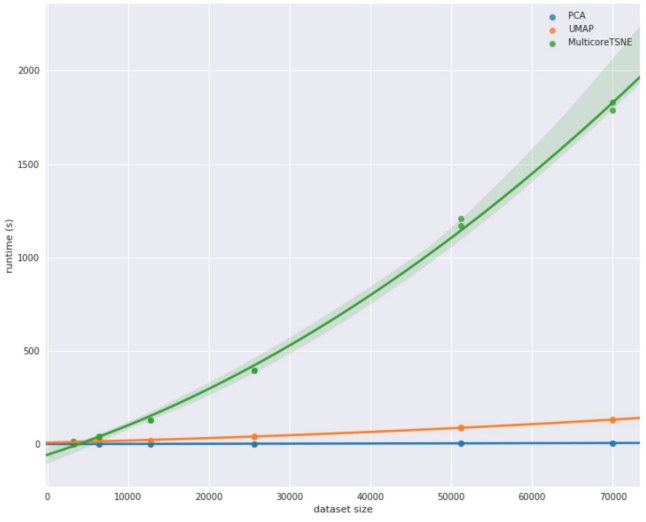


Fig 2.1: Performance Scaling w.r.t. Dataset size

2. In a practical scenario, t-SNE would only be able to embed 2-3 dimensions, which makes it possible to be used for only visualisation and not as a general dimensionality reduction algorithm.

3. Unlike PCA, t-SNE and UMAP find an embedding of a dataset into a low-dimensional space by minimizing a non-convex loss function. Therefore, it is not possible to embed test points in an existing map for these techniques, as the axes are not interpretable as in PCA. Thus, the location of test points on the mapping space would not be a valid metric to compare these techniques.

4. Comparing performance scaling by dataset size, PCA is the fastest followed by UMAP and t-SNE respectively. Also, comparing t-SNE and UMAP, larger the dataset, the scale up is more speeding. This means that for larger datasets the scaling performance of UMAP is only going to asymptotically grow, even though it is significantly less than PCA. t-SNE consumes too much memory and time for its computations.

3. Datasets

MNIST:

A dataset of 28x28 pixel grayscale images of handwritten digits consisting of 10 digit classes (0 through 9) and 70000 total images. This is treated as 70000 different 784 dimensional vectors.

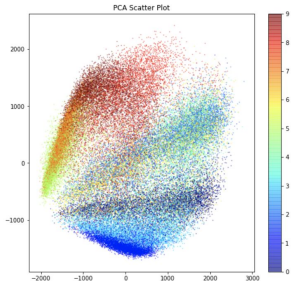
F-MNIST or Fashion MNIST:

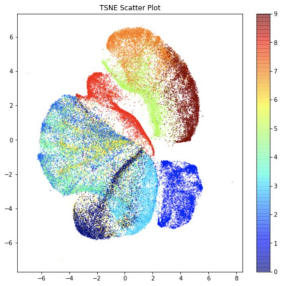
A data set of 28x28 pixel grayscale images of fashion items (clothing, footwear and bags) consisting of 10 classes and 70000 total images. As with MNIST, this is treated as 70000 different 784 dimensional vectors.

Breast Cancer Wisconsin (Diagnostic) Data Set:

A data set of features computed from a digitized image of a fine needle aspirate (FNA) of breast mass. They describe the characteristics of the cell nuclei present in the image. Ten real-valued features are computed for each cell nucleus. The labels are ‘Benign’ and ‘Malignant’.

4. Results and Validations





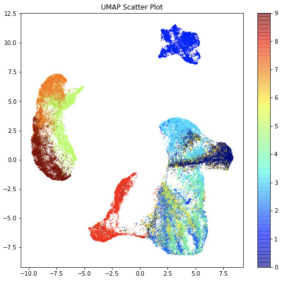
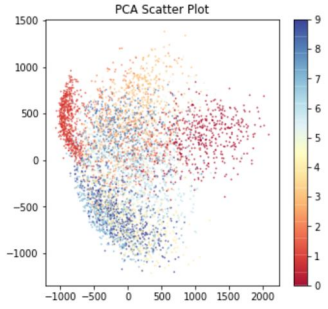
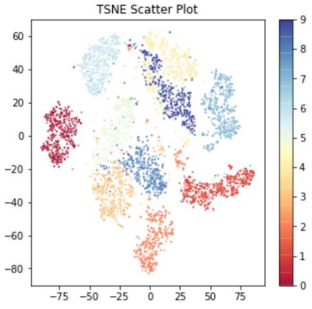


Fig.4.1: PCA on F-MNIST **Fig.4.2: t-SNE on F-MNIST** **Fig.4.3: UMAP on F-MNIST**

PCA, t-SNE and UMAP were applied on the **F-MNIST** dataset to reduce it from a feature space of 784 to 2. PCA (Fig.4.1) was applied on the F-MNIST dataset and as it can be seen that the clusters are not very clear. PCA selects the principal components based on the Eigenvalues with the maximum variance hence the resulting clusters are directed towards the maximal variance direction. t-SNE (Fig.4.2) resulted in clearer clusters but t-SNE takes a really long time to run - it took around 2000 times more time to be applied to the dataset as compared to PCA. UMAP (Fig.4.3) resulted in even clearer clusters since it preserves both global and local structures. Time taken to apply UMAP on the dataset was longer than PCA, around 40 times more but compared to t-SNE it was fast.





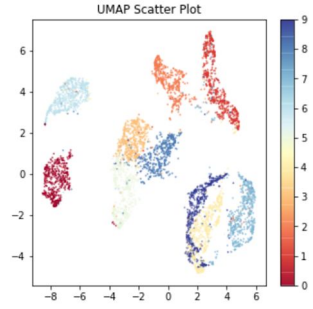


Fig.4.4: PCA on MNIST digit **Fig.4.5: PCA + t-SNE on MNIST** **Fig.4.6: UMAP on MNIST digit**

PCA(Fig.4.4) was applied on a subset of **MNIST digits data set** with 5000 records. The key observation here is that the global structure of the data is preserved even though it has been projected down to 2 dimensions from 784. The clusters however, are not clear. t-SNE(Fig.4.5) focuses on local structure and managed to understand the structure of the digits far better than PCA, thus retaining a lot more information but it has a drawback since it is very slow. We used PCA to reduce the number of features from 784 down to 20. We applied t-SNE on that reduced dataset to speed it up by a significant percent. However it still fared slower that UMAP and PCA. In future work we aim on analyzing the effects of stacking up and combining different dimensionality reduction methods to observe their outcomes. UMAP(Fig.4.6), picked out the digits very clearly and at the same time also captured similar global structure like PCA.

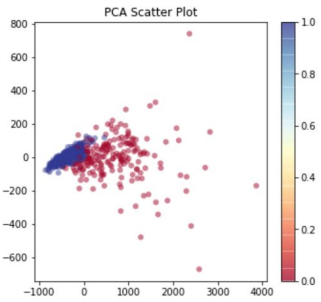


Fig.4.7: PCA

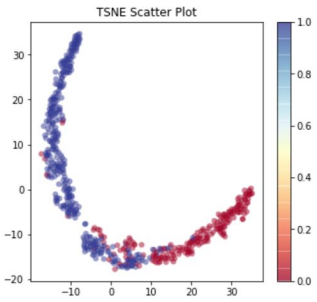


Fig.4.8: t-SNE

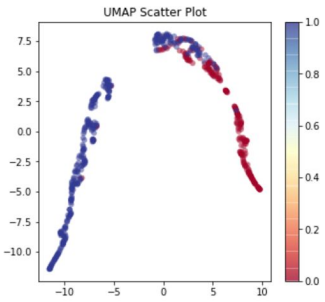


Fig.4.9: UMAP

PCA, TSNE and UMAP were applied on the **Breast Cancer Dataset** once using normalization and once without. Fig.4.7, Fig.4.8 and Fig.4.9 show the data points on a 2-D scale after PCA, t-SNE and UMAP were applied respectively on the full dataset of 569 data points without applying normalization. The time taken to apply the 3 methods were similar in trend with the others with PCA being the fastest followed by UMAP and t-SNE being the slowest. It is hard to see the clusters in PCA and t-SNE but in UMAP the data points seem to be divided somewhere in the middle.

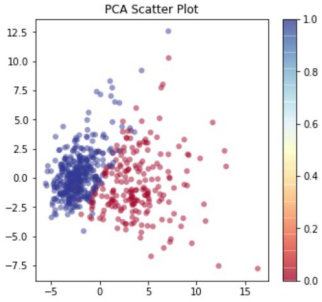


Fig.4.10: PCA

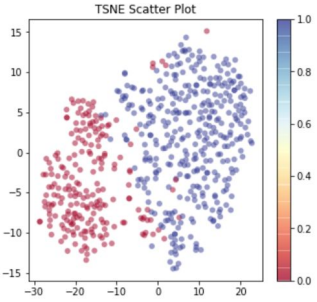


Fig.4.11: t-SNE

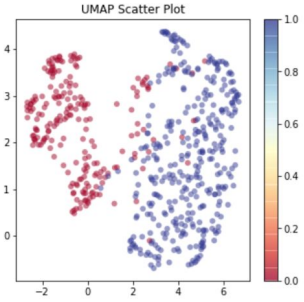


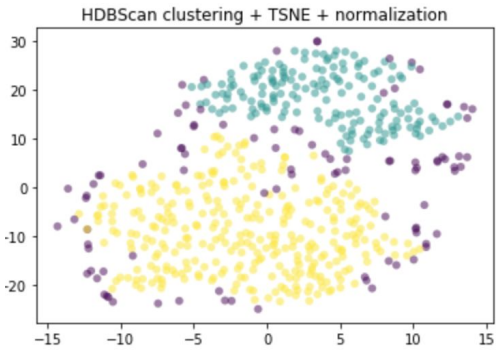
Fig.4.12: UMAP

Fig.4.10, Fig.4.11 and Fig.4.12 show the data points on a 2-D scale after PCA, t-SNE and UMAP were applied respectively on the full normalized dataset of 569 data points. It was observed that the clusters look more circular and spread out but to quantify the difference we introduced new metric using normalized mutual information. The scatter plot for PCA did not change much but the clusters became much clearer for t-SNE and UMAP.

Table.4.2: Time Taken in seconds

Method Used	Normalized dataset	Non-Normalized dataset
TSNE	7.172541379928589	6.113628387451172
UMAP	1.502000093460083	1.4437689781188965
PCA	0.005753517150870	0.0114958286285400

Table.4.2 shows the time taken for applying PCA, t-SNE and UMAP on the normalized and non-normaziled dataset. PCA is the fastest followed by UMAP. t-SNE is by far the slowest.



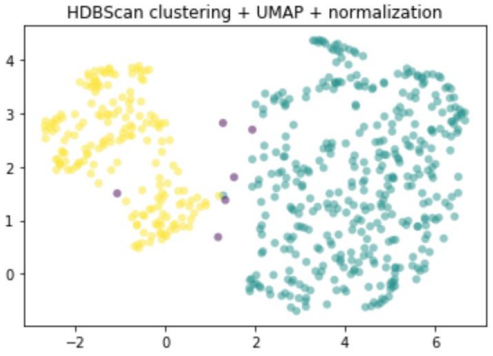
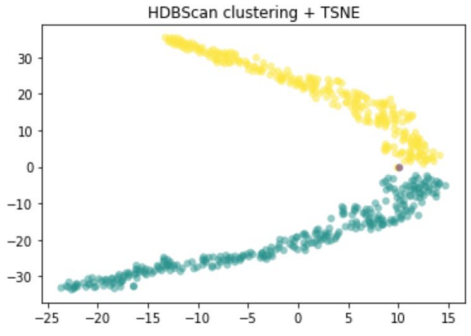


Fig.4.13: HDBScan on t-SNE (normalized) **Fig.4.14: HDBScan on UMAP (normalized)**

We applied HDBScan on the t-SNE features (Fig.4.15) and the UMAP features (Fig.4.16) obtained from the unprocessed dataset to get the normalized mutual information score as given in **Table.4.1**. We applied t-SNE and UMAP on the normalized dataset as well and applied HBDScan on that data which was visualized in Fig.4.13 and Fig. 4.14 and recorded in Table.4.1.



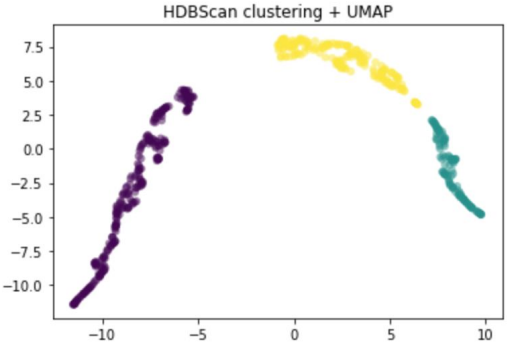


Fig.4.15: HDBScan on t-SNE **Fig.4.16: HDBScan on UMAP**

As it can be observed from the table below the normalized mutual information score for UMAP on the normalized dataset is the highest i.e. it did the best when HDBScan was used to clusted the reduced dataset. Fig.4.14 also supports this finding because it shows that HDBScan clustered the data into 2 clusters more effectively as compared to the other datasets. It is an expected outcome because Fig.4.12 showed that UMAP on the normalized dataset did the best visually to divide the data into clusters.

Table.4.1 Normalized Mutual Information Score

Method Used	Normalized dataset	Non-normalized dataset
TSNE	0.5948855432190262	0.41099173198419464
UMAP	0.7035849931709405	0.45472454401019796

5. Metrics

a. Quantitative metrics

i. Time Taken

Time taken is a simple quantitative metric which basically depends on factors like computational complexity of algorithms, size of dataset. etc. A more complex algorithm will take more time. Complex does not mean better. In fact, taking into account the huge volumes of high dimensional data that we have to deal with to solve any machine learning problem, a faster algorithm giving approximately the same results will always be chosen over the slower algorithm.

tSNE is more computationally intensive than UMAP hence tSNE since it requires to perform global normalization unlike UMAP. tSNE, as compared to UMAP, has a higher space complexity as well ,which increases exponentially with an increase in the number of input features.

Moreover, when we get new data samples we have to apply the tSNE algorithm on the entire new dataset. In case of UMAP, it acts like a function which takes a sample point and outputs the compressed sample. Hence, we can just generate the new embeddings and append it to the already processed dataset.

We recorded the time taken by UMAP and tSNE on the normalized Breast Cancer Dataset and it verified that UMAP is faster than tSNE (Fig.5.1). Hence in terms of the time metric UMAP performs vastly better as compared to tSNE.

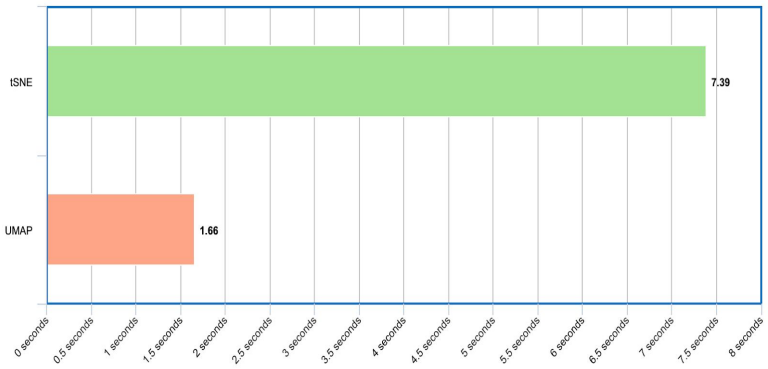


Fig 5.1. Recorded time metric for tSNE and UMAP

ii. Normalized Mutual Information Score

Mutual Information or MI is calculated between two clusterings. It is a measure of similarity which compares the two labels of the same data. MI does not depend on the absolute values of the labels and is a symmetric measure of similarity. It is generated using the following formula -

$$MI(U,\ V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} log \frac{N|U_i \cap V_j|}{|U_i||V_j|}$$

where, $|U_i|$ is the number of the samples in cluster U_i and $|V_i|$ is the number of the samples in cluster V_i.

Normalized Mutual Information or NMI is obtained by normalizing the MI to scale the MI values between a range of 0 and 1. 0 signifies that there is no mutual information and 1 signifies that the points are in perfect correlation with each other. Till now it is impossible in practice to get a score of 1 because dimension reduction is a kind of compression and as we know any sort of compression must have a loss. Hence we want a score as close to 1 as possible.

We calculated the NMI score for tSNE and UMAP applied on the normalized Breast Cancer Dataset as shown in Fig.5.2. As we can see UMAP has a higher NMI score than tSNE, hence, UMAP can be said to be the superior algorithm.

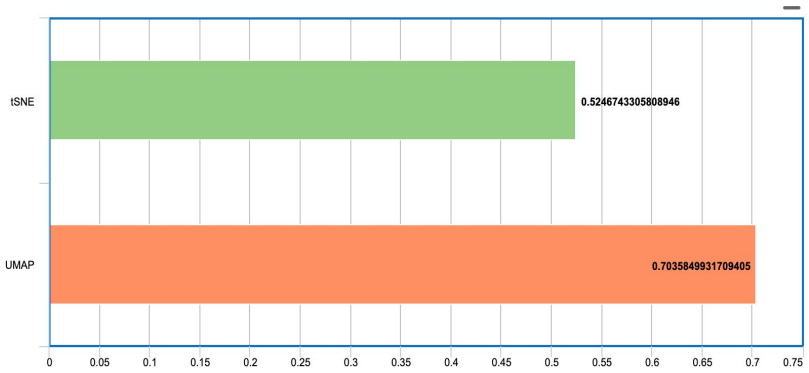


Fig.5.2.NMI score for UMAP and

iii. Stability of Sub-Sample Embeddings

Since both tSNE and UMAP are stochastic dimensionality reduction techniques, we can use some metric to determine the stability of UMAP and tSNE embeddings under sub sampling and compare them. Normalised Procrustes distance is one such metric to measure the distance between two such distributions. The Procrustes distance between two datasets X = {x1,...,xN} and Y = {y1,...,yN}, such that xi corresponds to yi, can be given as:

$$d_p(X, Y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i')^2}$$

Where Y′ ={y1′,...,yN′} is the optimal translation, uniform scaling, and rotation of Y for minimizing the squared error.
Also, we normalize these datasets before computing the distance as any distance metric in embedding space is potentially sensitive to the scale of the embedding. Now, stability can be examined by considering the normalized Procrustes distance between embedding of a sub-sample and the original dataset. Logically, on increasing the size of the sub-sample, the average distance per point between the embeddings should decrease.
A comparison between the stability of UMAP and tSNE based on the above metric deemed UMAP as the more stable algorithm as its sub-sample embeddings were more similar to that of the full dataset. This is verified by Fig._ (McInnes L et al. paper referred), which shows the average Procrustes distance per point for tSNE, UMAP and LargeVis (which is another method for dimensionality reduction).

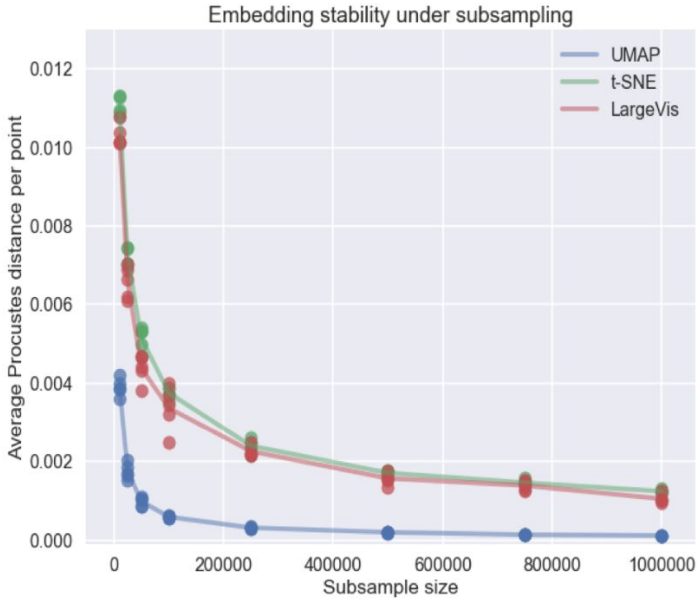


Fig 5.3. Recorded time metric for tSNE and UMAP

b. Qualitative metrics

i. Dataset size performance

Fig.5.4 shows the performance of various dimensionality reduction methods on the full Google News dataset, for different dataset sizes(McInnes L et al. paper referred). In this case, both the normal and the optimized 8 core tSNE implementations underperformed in comparison to UMAP.

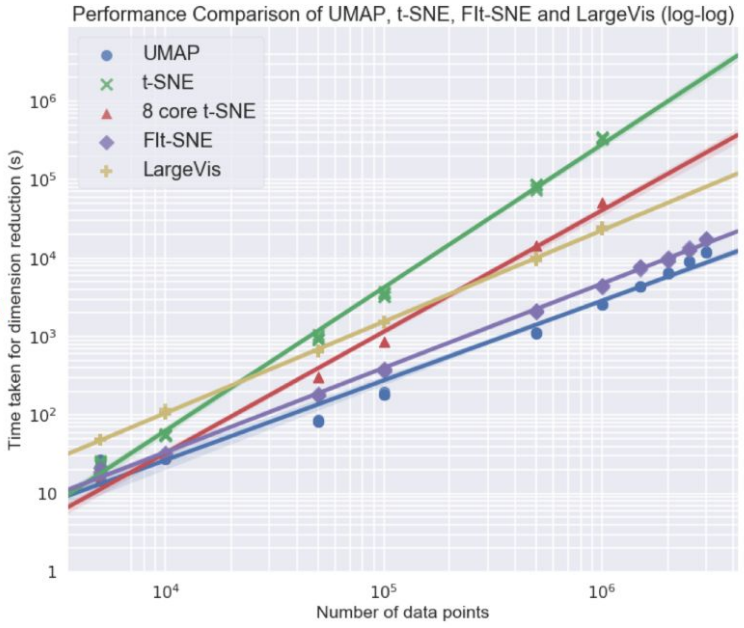


Fig 5.4. Dataset size metric for tSNE and UMAP

ii. Scaling Performance with varying ambient dimensions

For tSNE, as discussed previously, it is advisable to reduce the dimensions with PCA first before applying tSNE for high dimensional data. UMAP, on the other hand, is very efficient in terms of high dimensional data because it uses a combination of the local connectivity constraint and approximate nearest neighbour search. Fig.5.5 shows the performance of various dimensionality reduction methods with respect to the varying ambient dimensions of the data. We can see an abrupt increase in run time for tSNE and other algorithms after crossing the dimension value of one thousand, while UMAP relatively is more stable with a gradual and minimal increase in running time.

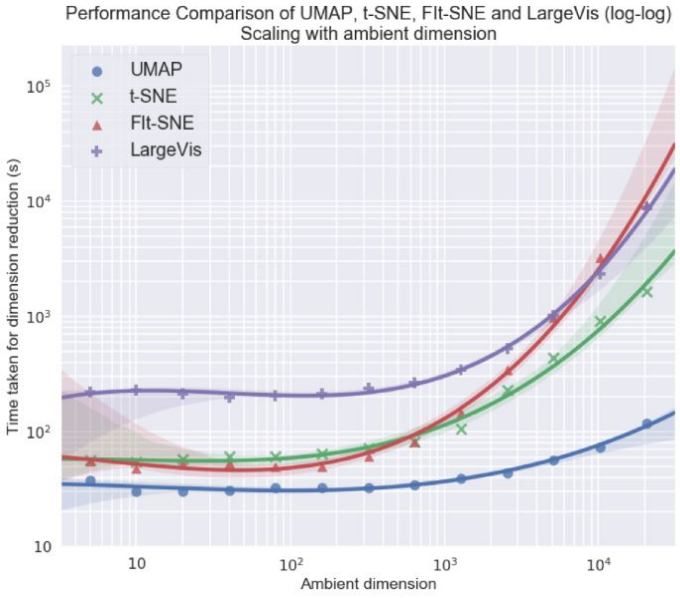


Fig 5.5. Scaling Performance Metric Visualization

6. Future Work

The following approach can also be employed in future for further analysis of the dimensionality reduction methods: Different reduction algorithms applied to a number of different data sets with varying properties like dataset size, kind of data, etc. sometimes have distinct characteristics for a few special cases. For example, UMAP works exceptionally well for medical imaging datasets. So if a dataset concerning with medical imaging comes up, UMAP should score better than the other algorithms. These attributes of the dataset could be collected alongside the results i.e. the scores obtained on applying the different dimensionality reduction methods so that it can be used to train a model to predict the score of a dimensionality reduction algorithm given a certain kind of dataset.

7. Conclusion

We have presented a detailed analysis of the popular dimensionality reduction algorithms, Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP). We discussed about the various qualitative and quantitative metrics that affect the performance of these algorithms.

In terms of quantitative metrics, we have taken into account the time complexity, normalized mutual score and the stability of sub-sample embeddings into consideration in order to assess the performance of the algorithms. In case of time taken, since tSNE is much more computationally intensive than UMAP, the latter is preferred as it produces the same (rather better) results than tSNE in less time. In case of normal mutual information, UMAP(0.70 for normalized data) has a slightly higher value than tSNE(0.59), signifying that the reduced embeddings are more strongly correlated to the initial input data in case of UMAP which is more desirable. While comparing the stability of UMAP and tSNE based on Normalised Procrustes distance, UMAP performed better as its sub-sample embeddings were more similar to that of the full dataset.

Moving on to qualitative metrics, we looked into the performance of both tSNE and UMAP for in terms of varying dataset size and varying number of features in the dataset (i.e. performance on high dimension data). For both metrics, UMAP algorithm performed much more efficiently.

From the above observations, it can be clearly concluded that UMAP is the most efficient algorithm here as it fares well in all the metrics that we have taken into consideration.

8. References

- https://arxiv.org/pdf/1802.03426.pdf
- https://www.youtube.com/watch?v=NEaUSP4YerM
- https://stats.stackexchange.com/questions/402668/intuitive-explanation-of-how-umap-works-compared-to-t-sne
- https://github.com/lmcinnes/umap/blob/master/doc/benchmarking.rst
- https://www.youtube.com/watch?v=YPJQydzTLwQ&feature=youtu.be
- Xiao, Han, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." *arXiv preprint arXiv:1708.07747* (2017).
- https://arxiv.org/abs/1802.03426
- https://www.biorxiv.org/content/biorxiv/early/2019/02/15/549659.full.pdf
- http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.44.3452

- https://deeplearn.org/arxiv/56827/umap:-uniform-manifold-approximation-and-projection-for-dimension-reduction
- https://dl.acm.org/citation.cfm?id=775143
- https://arxiv.org/pdf/1403.2877.pdf
- https://github.com/lmcinnes/umap/blob/master/doc/benchmarking.rst
- https://www.analyticsvidhya.com/blog/2018/06/google-open-sources-approach-to-visualize-large-and-high-dimensional-datasets-using-tsne-a-must-know-for-data-scientists/
- https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/
- https://towardsdatascience.com/introduction-to-image-segmentation-with-k-means-clustering-83fd0a9e2fc3
- https://www.biorxiv.org/content/biorxiv/early/2018/04/10/298430.full.pdf
- https://www.sciencedirect.com/science/article/pii/S0957417410008638
- https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/modeling-statistics/regression/supporting-topics/basics/a-comparison-of-the-pearson-and-spearman-correlation-methods/
- http://theprofessionalspoint.blogspot.com/2019/03/advantages-and-disadvantages-of-t-sne.html
- https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668
- https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html
- https://www.researchgate.net/publication/334223637_Spectral_Overlap_and_a_Comparison_of_Parameter-Free_Dimensionality_Reduction_Quality_Metrics/fulltext/5d1d709b299bf1547c957913/Spectral-Overlap-and-a-Comparison-of-Parameter-Free-Dimensionality-Reduction-Quality-Metrics.pdf
- https://lvdmaaten.github.io/tsne/
- https://arxiv.org/pdf/1802.03426.pdf
- https://www.researchgate.net/publication/334223637_Spectral_Overlap_and_a_Comparison_of_Parameter-Free_Dimensionality_Reduction_Quality_Metrics/fulltext/5d1d709b299bf1547c957913/Spectral-Overlap-and-a-Comparison-of-Parameter-Free-Dimensionality-Reduction-Quality-Metrics.pdf
- https://distill.pub/2016/misread-tsne/
- https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume23/roy05a-html/node3.html
- https://lvdmaaten.github.io/tsne/
- https://www.meta-chart.com/bar#/display