ClusterView: Dimensionality Reduction for Preserving Global and Local Data Structure

No Author Given

No Institute Given

Abstract. The use of dimensionality reduction for analyzing and visualizing high-dimensional data has proven to be highly useful. Using methods such as PCA, t-SNE, UMAP, etc., we can project data to lower dimensions (2 or 3) making it possible for us to visualize it. These methods aim to preserve the structure of the original data, however, some information is always lost. Nonlinear techniques such as UMAP and t-SNE are excellent at preserving the local structure around each data point but do not account as much for the global structure. On the other hand, techniques such as PCA preserve global pair-wise distances but are not so effective in separating the clusters. Therefore, we design a novel algorithm, ClusterView, to effectively visualize clusters while preserving relative global pair-wise distances (and minimize stress metric).
To keep the global structure intact, we first perform graph clustering on the near-neighbor graph of the dataset and compute the cluster centers. The cluster centers are projected to lower dimensions using a force-directed layout. Next, we perform dimensionality reduction on the original data using UMAP. To preserve the relative orientations/placements of the clusters, we translate the cluster centers of the UMAP layout to the positions computed by the force-directed layout. We find that for most datasets, this translation reduces the value of stress considerably while preserving other local metrics such as trustworthiness and continuity. This approach gives good results for different values of clusters as well as different numbers of features and records.

1 Introduction

In the field of Data Analytics and Visualization, dimensionality reduction is a very important tool that facilitates the understanding of complex data. It enables us to effectively see data in a lower dimensional plane while keeping intact the properties of the higher dimension because of which it is used in various fields such as speech recognition [1], signal processing [2, 3], bioinformatics [4], etc. Since it is such a useful tool, there are a variety of different works in this field. Some of these works are linear such as the most widely known PCA whereas others are nonlinear such as t-SNE [5] and UMAP [6].

While nonlinear techniques are highly successful in capturing the local structures of the data, they have not achieved the same in the corresponding global layouts. Often, they show poor results for global metrics such as stress. A recent study [7] disproved the claim that UMAP was better at preserving global structure than TSNE and concluded that it does not outperform TSNE. Therefore, to improve upon these existing limitations, we present a method for improving the global properties of UMAP layouts.

In this method, ClusterView, graph clustering is applied to the graph that UMAP internally creates and then representative points (cluster centers) are obtained for each cluster. Keeping the clusters the same in the lower dimension, they are translated according to the force-directed layout of the higher dimensional cluster centers. Since the new embedding tries to keep the relationship between cluster centers of higher dimensions intact in the lower dimensions, it improves the global structure.

The results obtained from our method were quantitatively compared with UMAP, t-SNE and UMATO (Uniform Manifold Approximation with Two-phase Optimization) [8] using the global metric of stress and local metrics of trustworthiness and continuity. We observed that the value of stress is lowered by a lot as compared to UMAP and t-SNE, and is comparable to UMATO.

2 Method

We evaluated different approaches for the values of stress, trustworthiness and continuity.

Stress [9] is a global metric that measures the preservation of the point-pairwise distances from higher dimension to lower dimension.

For a set of points $x_1, ..., x_n$, in a K-dimensional space $\mathbb{R}^K$ and their respective low dimensional embeddings $\tilde{x}_1, ..., \tilde{x}_n$ in $\mathbb{R}^k$, we define its Stress, M_σ, as:

$$M_\sigma = \frac{\sum_{i,j}(||d_{i,j}|| - ||\tilde{d}_{i,j}||)^2}{\sum_{i,j}||d_{i,j}||^2}$$

where $d_{i,j} = ||x_i - x_j||$ and $\tilde{d}_{i,j} = ||\tilde{x}_i - \tilde{x}_j||$.

The local metrics that were used were **continuity** and **trustworthiness**. Continuity [9], M_C, measures the proportion of points that are close in the projection which were also close in the higher dimension whereas trustworthiness [9], M_T, measures the proportion of points that were close in the higher dimension and are also close in the obtained projection. Both of these metrics range from [0,1] with 1 being the best value that they can have. For the evaluation, we chose the number of nearest neighbours to be considered for local metrics as $K = 7$, in line with other works such as [9], [10] and [11].

We also consider a composite metric, that takes into consideration both the local and the global structure, similar to what [9] use. We aggregate the previously considered metrics and obtain another metric, μ:

$$\mu = \frac{1}{3}(M_C + M_T + (1 - M_\sigma))$$

Ideally, the value of μ should be in the range of [0,1]. However, because the stress value, M_σ, can be greater than 1 for datasets where the distance between pair of points in the projection is more than the distance between the same points in the higher dimension, it causes the value of the composite metric μ to be negative.

2.1 Our Approach: ClusterView

The algorithm we used consists of three stages. The first stage involves clustering in the higher dimension. The second stage is calculating the cluster centers in the higher dimension and obtaining their mappings in two dimensions using force-directed layout. In the third stage, we calculate the cluster centers on UMAP embeddings and then find the difference in coordinates between the cluster centers obtained in second stage and those obtained from UMAP embeddings. After that, we translate the UMAP clusters by the difference so that the cluster centers from the second stage match the cluster centers of the mappings obtained now.

Graph Clustering Firstly, we cluster the points using graph clustering on the graph that UMAP makes. The graph uses the nearest neighbors of a point to determine the point's connectivity. The algorithm used for graph clustering is METIS [12], which utilizes a multilevel approach to create partitions in graphs.

Force-Directed Layout After obtaining the clusters from UMAP's graph, the center of each of them is calculated in the higher dimension. Then, we applied a force-directed layout to the centers. The force-directed layout considers the magnitude of the edges between each node corresponding to a cluster center as an inverse function of the distance between them. It gives the mappings of the centers in two dimensions.

Translation of Clusters Now, UMAP is applied to the data, and embeddings are obtained in two dimensions. Again, cluster centers are calculated considering the clusters to be the same as computed for the higher dimensions. Next, these coordinates are compared to the ones that were given by the force-directed layout of the previous step. The difference in their coordinates is calculated. Using the calculated difference, all the points of every cluster are translated. This process ensures that the coordinates of the centers of all the clusters after translation are the same as those obtained in the previous step from force-directed layout.

3 Dataset and Experiments

The data we are using consists of seven datasets which are the Penguins, Iris, MNIST [13], Wine, FMNIST [14], Diabetes and Wheat Seed datasets. We normalized the datasets using StandardScaler. These datasets provide a lot of variety to us in terms of their classes, number of features, and number of records.

```
Data: $X = \{x_1, x_2 ... x_n\} \quad x_i \in \mathbb{R}^k$
Result: $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2 ... \tilde{x}_n\} \quad \tilde{x}_i \in \mathbb{R}^2$
$G \leftarrow UMAP\_KNN\_GRAPH(X)$;
$C \leftarrow GRAPH\_CLUSTERING(G)$ ;
/* $C = \{c_1, c_2, ...c_p\}$ where $p$ = number of clusters */
$M \leftarrow UMAP(X)$ ;
/* $M = \{m_1, m_2, ...m_n\}$ where $m_i \in \mathbb{R}^2$*/
foreach cluster $c_i \in C$ do
    $r_i \leftarrow CENTRE(c_i)$ ;
    $u_i \leftarrow CENTRE(UMAP(c_i))$ ;
end
/* $R = \{r_1, r_2, ...r_p\}$ where $r_i$ is the representative of $c_i$, $r_i \in \mathbb{R}^k$ */;
/* $U = \{u_1, u_2, ...u_p\}$ where $u_i$ is the representative of $c_i$, $u_i \in \mathbb{R}^2$ */
$FD \leftarrow FORCE\_DIRECTED\_LAYOUT(R)$ ;
/* $FD = \{fd_1, fd_2, ...fd_p\}$ where $fd_i \in \mathbb{R}^2$*/
foreach cluster $c_i \in C$ do
    $diff_i \leftarrow fd_i - u_i$ ;
    foreach $\tilde{x}_j \in c_i$ do
        $\tilde{x}_j \leftarrow m_j + diff_i$ ;
    end
end
```

Algorithm 1: ClusterView

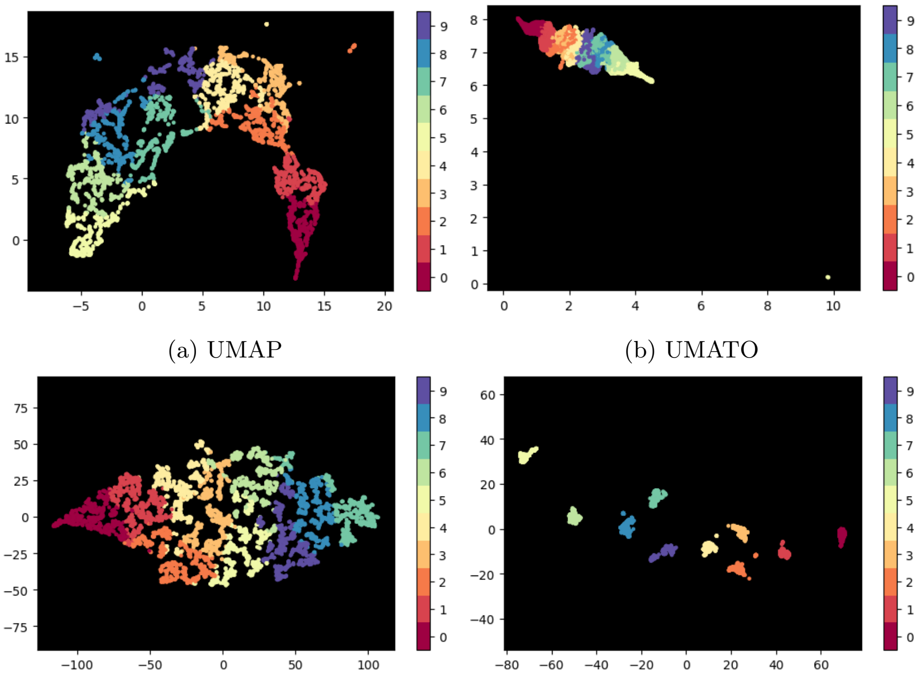
While MNIST and FMNIST contain a large number of records, for other datasets they are less than 10000. There is also a wide range in the number of features considered, with Penguins and Iris having 4 each and MNIST and FMNIST having 784 each.

3.1 Experiments

Our experimentation consisted of running all four of the dimensionalty reduction techniques, i.e. UMAP, t-SNE, UMATO and ClusterView, on the seven datasets and comparing their stress M_σ, continuity M_C and trustworthiness M_T. Finally, to compare which method preserves both local and global structure, we used a composite metric μ. The outcome of our experimentation yielded a set of numerical results, which provided us with insights into the efficacy of our methodology.

Experiment 1: UMAP and t-SNE As a part of our first experiment, we used UMAP and t-SNE. We ran both of them on all of the datasets. For MNIST and FMNIST, we used 10000 datapoints.

Experiment 2: UMATO The second experiment consisted of us running UMATO on all of the datasets and then computing the three metrics. The number of hub points used were kept the default 300 in all cases, except for Iris and Wheat Seed where they were lowered down to 50.



(a) UMAP

(b) UMATO

(c) TSNE

(d) ClusterView

Fig. 1: Wine Dataset

Experiment 3: ClusterView In our third experiment, we used Metis to cluster the graph that we obtained from the result of the first experiment. The embeddings for each dataset that were obtained in the first experiment from UMAP were used for translation.

4 Results and Discussions

We used the stress metric to compare the performance of UMAP, t-SNE, UMATO and ClusterView.

Upon comparing the performance of the four methods in Table 2, we observe that the value of stress is most often the least for our method. We also note that for the datasets where the value of stress is not the least for our method, it is always comparable to the other two whereas the same cannot be said for the others.

For the wine dataset, the value of stress went down by 10× of the original UMAP by using our method. For UMATO, the value is increasing.

From Table 2, we also see that the value of continuity and trustworthiness obtained from our method is comparable to the other methods. While it does not

Dataset	Classes	Features	Records
Penguins	3	4	333
Iris	3	4	150
MNIST	10	784	70000
Wine	10	11	4898
FMNIST	10	784	60000
Diabetes	2	8	767
Wheat Seed	3	7	209

Table 1: Datasets and their Traits

perform better than t-SNE, in some cases it is better than UMAP, and almost always gives better values than UMATO.

The composite metric, μ, is for most cases the best for ClusterView. This implies that our method is good at improving global structure while keeping the local relationships in the data mostly undisturbed. This is in accordance to the design of ClusterView: keeping local structure preserved because it uses UMAP as its base, and improving the global layout because of the representative cluster centres used.

4.1 Discussion

We observed that ClusterView performed very well on most of the datasets in improving the global metrics of the projection obtained from dimensionality reduction after UMAP. Not only that, but it also provided comparable values of local metrics. Because of the improvement in stress and comparable values to local metrics, it also had good values for the composite metric μ.

However, it did not perform well for MNIST as compared to the other methods. This is because the clusters were large in the higher dimension, and we did not compare the scale of force directed layout with the original data. Since the clusters remained larger than the distance between their centers in the lower dimension, they started overlapping and resulted in increase in stress.

A method that may further improve the layout in cases where clusters are larger than the distance between their centers is introducing 'shrinking' to them. We can shrink or expand the clusters based on their scales in higher dimension, which might prove beneficial and help modify the results accordingly.

5 Conclusion

In this paper, we contributed a novel method for dimensionality reduction, ClusterView, which works upon improving the global properties of UMAP layouts, while keeping the local comparable. This method consists of partitioning a graph that is already created by the UMAP algorithm and using representative points, i.e. cluster centers, from the clusters to keep intact the overall global layout of

Dataset	Metric	UMAP	UMATO	TSNE	ClusterView
Penguins	M_σ	6.8858	**0.1701**	115.2405	0.3329
	M_T	0.9851	0.9189	0.9910	0.9265
	M_C	0.9870	0.9101	0.9896	0.9450
	μ	-1.3046	**0.8863**	-37.4200	0.8462
Iris	M_σ	8.9352	0.5175	35.4993	**0.1788**
	M_T	0.9844	0.9008	0.9928	0.9494
	M_C	0.9888	0.8644	0.9931	0.9693
	μ	-1.9873	0.7492	-10.8378	**0.9133**
MNIST	M_σ	0.9941	0.9970	**0.9456**	0.9948
	M_T	0.9665	0.9105	0.9915	0.9545
	M_C	0.9798	0.9730	0.9815	0.9204
	μ	0.6507	0.6288	**0.6758**	0.6267
Wine	M_σ	0.6978	0.9561	0.1158	**0.0755**
	M_T	0.9979	0.9677	0.9995	0.9979
	M_C	0.9969	0.9939	0.9971	0.9869
	μ	0.7657	0.66858	0.9602	**0.9698**
FMNIST	M_σ	0.6598	0.9365	1.0249	**0.2919**
	M_T	0.9712	0.9400	0.9867	0.9723
	M_C	0.9881	0.9853	0.9875	0.9641
	μ	0.7665	0.6629	0.6498	**0.8815**
Diabetes	M_σ	0.8952	0.9846	0.6302	**0.4801**
	M_T	0.9884	0.9562	0.9940	0.9882
	M_C	0.9901	0.9720	0.9916	0.9901
	μ	0.6944	0.6479	0.7851	**0.8327**
Wheat Seed	M_σ	0.2271	0.7103	6.2629	**0.0515**
	M_T	0.9941	0.8883	0.9973	0.9755
	M_C	0.9905	0.8705	0.9932	0.9797
	μ	0.9192	0.6828	-1.0908	**0.9679**

Table 2: Comparison between the results of the methods where M_σ: Stress, M_T: Trustworthiness, M_C: Continuity

data. On comparison with other methods like UMAP, UMATO and TSNE, we found that ClusterView performs better in the preservation of global structure while keeping the local structure comparable. We also discussed ways in which its performance can be further improved by considering the scale of the clusters and shrinking/expanding them accordingly in the lower dimension.

References

1. P. Fewzee, F. Karray, Dimensionality reduction for emotional speech recognition, in: 2012 International Confernece on Social Computing, 2012, pp. 532–537.
2. S. Kay, Dimensionality reduction for signal detection, IEEE Signal Processing Letters 29 (2022) 145–148.
3. M. T. Guimarães, A. G. Medeiros, J. S. Almeida, M. Falcão y Martin, R. Damaševičius, R. Maskeliūnas, C. L. Cavalcante Mattos, P. P. Rebouças Filho,

An optimized approach to huntington's disease detecting via audio signals processing with dimensionality reduction, in: 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–8.

4. G. Armstrong, G. Rahman, C. Martino, D. McDonald, A. Gonzalez, G. Mishne, R. Knight, Applications and comparison of dimensionality reduction methods for microbiome data, Frontiers in Bioinformatics 2 (2022).
5. L. van der Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579–2605.
6. L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction (2020). arXiv:1802.03426.
7. D. Kobak, G. Linderman, Umap does not preserve global structure any better than t-sne when using the same initialization (12 2019).
8. H. Jeon, H.-K. Ko, S. Lee, J. Jo, J. Seo, Uniform manifold approximation with two-phase optimization, 2022 IEEE Visualization and Visual Analytics (VIS), IEEE, 2022.
9. M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, A. C. Telea, Toward a quantitative survey of dimension reduction techniques, IEEE Transactions on Visualization and Computer Graphics 27 (3) (2021) 2153–2173.
10. L. van der Maaten, E. Postma, H. Herik, Dimensionality reduction: A comparative review, Journal of Machine Learning Research - JMLR 10 (01 2007).
11. Y. Koren, L. Carmel, Robust linear dimensionality reduction, Visualization and Computer Graphics, IEEE Transactions on 10 (2004) 459 – 470.
12. G. Karypis, V. Kumar, Metis—a software package for partitioning unstructured graphs, partitioning meshes and computing fill-reducing ordering of sparse matrices (01 1997).
13. Y. LeCun, C. Cortes, MNIST handwritten digit database (2010) [cited 2016-01-14 14:24:11].
URL http://yann.lecun.com/exdb/mnist/
14. H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, CoRR abs/1708.07747 (2017). arXiv:1708.07747.