A Tool for Visualizing and Analyzing High-Dimensional Clustering Performance

Justin Lin*and Julia Fukuyama†

Abstract

Technological advances have spurred an increase in data complexity and dimensionality. We are now in an era in which data sets containing thousands of features are commonplace. To digest and analyze such high-dimensional data, dimension reduction techniques have been developed and advanced along with computational power. Of these techniques, nonlinear methods are most commonly employed because of their ability to construct visually interpretable embeddings. Unlike linear methods, these methods unevenly stretch and shrink space to create a visual impression that reflects the structure of the high-dimensional data. Unfortunately, capturing high-dimensional structures in a significantly lower number of dimensions requires drastic manipulation of space. As such, nonlinear dimension reduction methods are known to occasionally create false structures, especially in noisy settings. In efforts to deal with this phenomenon, we developed an interactive tool that enables analysts to better understand and diagnose their dimension reduction results. It uses various analytical plots to provide a multi-faceted perspective on captured structures in the data to determine if they're faithful or consquences of the dimension reduction process. The tool is available in an R package named *DRtool*.

1 Introduction

The potency of nonlinear dimension reduction methods lies in their flexibility, allowing them to model complex data structures. That same flexibility, however, makes them difficult to use and interpret. Each method requires a slew of hyperparameters that need to be calibrated, and even when adequately calibrated, these methods require a trained eye to interpret. For example, the two most popular nonlinear dimension reduction methods, t-SNE and UMAP, are known to generate unintuitive results ([5], [20]). The results often cluster, even when no clusters exist in the data, and cluster sizes/placements can be unreliable. We've developed an interactive tool that analysts may use to conduct a post-hoc analysis of their high-dimensional clustering. The tool uses the minimum spanning tree (MST) to model the global structure of clusters and provide an additional perspective on inter-cluster relationships. This allows analysts to extract more information from their dimension reduction results by making it easier to differentiate the signal and the noise.

2 Methods

2.1 The Minimum Spanning Tree

Graphs have been applied to many multivariate statistical problems. The authors of [18] introduced the minimal ascending path spanning tree as a way to test for multimodality. The Friedman-Rafsky test [11], along with its modern variations [2, 3, 4], use the MST to construct a multivariate two-sample test. Single-linkage clustering [10] and runt pruning [15] are both intimately related with the MST. In the context of dimension reduction, IsoMap [16] makes use of neighborhood graphs, [12] introduces the maximum information spanning tree, and [17] uses the MST. These methods, which fall under the category of manifold learning, use graphs to model high-dimensional data assumed to be drawn uniformly from a high-dimensional manifold. An accurate low-dimensional embedding can then be constructed from these graphs. It's apparent that graphs are useful for modeling high-dimensional data, especially when

*Department of Mathematics, Indiana University
†Department of Statistics, Indiana University

it comes to dimension reduction and cluster analysis. Our tool uses the MST to analyze the reliability of visualizations produced by nonlinear dimension reduction methods.

We've opted for the MST for a couple of key properties. Firstly, the MST and shortest paths along it are quick to compute. Secondly, the MST contains a unique path between any two vertices, providing a well-defined metric on the data. Lastly, it provides a good summary of the data's structure. It contains as a subgraph the nearest-neighbor graph, and any edge deletion in the MST partitions the vertices into two sets for which the deleted edge is the shortest distance between them [11].

2.1.1 MST Stability

The MST is meant to provide a robust estimation of the data's global structure, and more specifically, inter-cluster relationships. As such, it should be stable in the presence of noise and unaffected by local transformations of the data. To demonstrate MST stability, we study the effect of random noise on the inter-cluster relationships explained by the MST.

To derive the inter-cluster relationships from the MST, we simplified the medoid subtree using the following procedure:

Algorithm 1 Simplified Medoid Subtree

Require: MST $T = (V, E)$ with cluster medoids $m_1, \ldots, m_k \in V$

$T' = (V', E') \Leftarrow$ minimal subtree of T containing all m_i

repeat

  Let $v \in V'$ with $deg(v) = 2$ and neighbors $a, b \in V'$. Let $d(v, a)$ and $d(v, b)$ be the weights of the edges connected v and to a and b.

  Replace v and its two incident edges with an edge connecting a and b with weight $d(v, a)$ and $d(v, b)$.

until T' contains no longer contains non-medoid vertices with degree two

output T'

The simplification process essentially replaces paths of non-medoid vertices with one edge of equal length. We refer to this tree as the simplified medoid subtree. It encode global inter-cluster relationships within the data.

2.1.2 Robinson-Foulds Metric

To compare simplified medoid subtrees, we used the Robinson-Foulds metric [19]. The R-F metric was originally introduced to quantify the dissimilarity of phylogenetic trees, but the algorithm generalizes to arbitrary weighted trees. It looks at partitions of each tree created by removing individual edges, then counts the number of partitions present in one tree but not in the other. We modified the algorithm (Algorithm 2) to specifically measure the dissimilarity in medoid vertices and applied a normalization so that the distances range from zero to one.

Algorithm 2 Robinson-Foulds Distance

Require: Trees $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ with medoids $m_1, \ldots, m_k \in V_1$ and $n_1, \ldots, n_k \in V_2$

$P_1 \Leftarrow \{\}$

for $e \in E_1$ **do**

  $G \Leftarrow (V_1, E_1 \setminus \{e\})$ with connected components G_1 and G_2

  $M_1 \Leftarrow \{m_1, \ldots, m_k\} \cap V(G_1)$

  $M_2 \Leftarrow \{m_1, \ldots, m_k\} \cap V(G_2)$

  $P_1 \Leftarrow \textsc{Add}(P_1, \{M_1, M_2\})$

$P_2 \Leftarrow \{\}$

for $e \in E_2$ **do**

  $G \Leftarrow (V_2, E_2 \setminus \{e\})$ with connected components G_1 and G_2

  $M_1 \Leftarrow \{n_1, \ldots, n_k\} \cap V(G_1)$

  $M_2 \Leftarrow \{n_1, \ldots, n_k\} \cap V(G_2)$

  $P_2 \Leftarrow \textsc{Add}(P_2, \{M_1, M_2\})$

output $\frac{|P_1 \Delta P_2|}{2|P_1 \cap P_2|}$

2.2 The PCA – rCCA Method

To understand the relationship between two specific clusters, the tool requires the user to specify a path connecting said clusters. Once a path is specified, a local projection method is applied to visualize the path and nearby points of interest in two dimensions.

Let $P \in \mathbb{R}^{k \times p}$ be the matrix of high-dimensional path points with endpoints $p_1, p_k \in \mathbb{R}^p$. Let $X \in \mathbb{R}^{n \times p}$ be the matrix of points of interest. By default, the points of interest include the path points and all points belonging to the same cluster as p_1 or p_k. Alternatively, the user may select their own points of interest.

The idea is to use regularized canonical-correlation analysis (rCCA) to determine the two-dimensional linear projection that best unwinds the path. Given two data matrices with an equal number of rows, rCCA iteratively calculates linear combinations of the matrix features, known as canonical variate pairs, that maximize covariance. The canonical variate pairs are chosen to be orthogonal to all previous pairs, so they give rise to a projection subspace. To unwind P, we use CCA to compare P against the data matrix modeling a d-dimensional polynomial

$$P_d = \begin{bmatrix} 1 & 1^2 & \cdots & 1^d \\ 2 & 2^2 & \cdots & 2^d \\ \vdots & \vdots & & \vdots \\ n & n^2 & \cdots & n^d \end{bmatrix}$$

and use the first two canonical variate pairs to construct a two-dimensional projection of P that maximizes its covariance with P_d. This process generates a two-dimensional subspace on which we can project all of X. We use rCCA as opposed to ordinary CCA due to the high-dimensionality of X. Often times $p >> n$, leading to singular covariance matrices. The regularization constant is chosen using cross-validation. There is no regularization constant needed for P_d.

One issue with this method is the projected path often travels along the outskirts of the plot. This is due to the near-orthogonality of high-dimensional data [7] caused by the large number of noise dimensions. Because the non-path points are often nearly orthogonal with the projection subspace, they are overly shrunk in the projection. The path points are less affected because the projection subspace is selected to retain the path's shape. While this phenomenon doesn't discredit the entire plot, it leads to misrepresentation of the path's location relative to the rest of the points.

To alleviate this issue, we propose a PCA – CCA method. Prior to the CCA step, we first apply PCA on the entirety of X to reduce the levels of noise. This prevents the confusion of excess noise for independence. When CCA is applied post-PCA, the projected path's relative position to the rest of the points is more credible.

The tool allows the user to specify two different hyperparameters – the dimension of the initial PCA projection and the degree of the CCA comparison polynomial. The percentage of the variance retained during the PCA process is provided. It is recommended the user scans through a range of numbers of dimensions. As the number of dimensions increases, the path will begin to circumvent the rest of the points but more inter-cluster separation might appear. A smaller number of dimensions is useful for understanding the path's trajectory with respect to the points, while a larger number might be better for separating clusters.

To calibrate the degree of the CCA comparison polynomial, the user should scan over different values. In most cases, the path's shape will stabilize beyond a certain degree. It is recommended to set the degree to this degree of stabilization.

2.3 The Tool

After providing the original data matrix, the two-dimensional embedding data matrix, and a clustering, the user is met with a dashboard containing the low-dimensional plot colored according to the clustering and a sidebar with various settings. From there, the user must specify the path and points of interest in one of two ways – by simply inputting the endpoints or defining their own custom clusters. The former method will use the path points and points belonging to the same cluster as either endpoint as the points of interest. The latter method will choose the (high-dimensional) medoids of the selected clusters as endpoints and use the path points in addition to the custom clusters as the points of interest.

Once a path is chosen, its low-dimensional projection will overlay the low-dimensional embedding. Furthermore, the user will be presented with two analytical plots. The first plot will show the results of the PCA – CCA method. The sidebar includes hyperparameter settings as well as a bandwidth

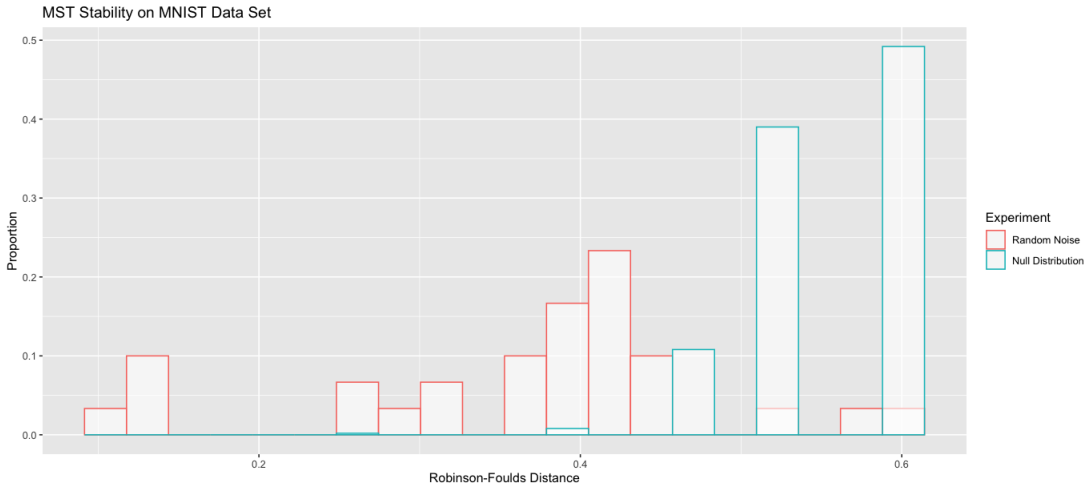

Figure 1: MST Stability on MNIST Data Set

adjustment. The bandwidth adjustment allows for a density overlay that may help users distinguish clusters. The second plot is a bar plot of path weights. The plot represents the weights of the original path in high dimension, so it provides insight on how the path was transformed throughout the dimension reduction process.

Other settings include a viewing of the entire medoid subtree and path highlighting. The medoid subtree is helpful for understanding the global positioning of clusters, and the path-highlighting function allows the user to traverse along the path while highlighting the corresponding segment in all three plots.

3 Results

3.1 MST Stability Experiment

1,500 samples were randomly chosen from the MNIST data set of handwritten digits [6]. Each 784×784-pixel image was flattened into a vector of length 784^2, so the data contain 1,500 samples in 784^2 dimensions. A PCA pre-processing step was employed to reduce the number of dimensions to 300. The simplified medoid subtree T was then calculated.

Random Gaussian noise was then added to the data and the new simplified medoid subtree T' was calculated. The R-F distance $RF(T, T')$ was recorded. This process was repeated 30 times.

To better interpret the R-F distances, we designed a null distribution of distances as a reference for comparison. These distances should represent R-F distances between trees that do not portray similar global structures and inter-cluster relationships. To generate the null distribution from the data, we randomly permuted the class labels and computed the R-F distances between the resulting simplified medoid subtrees and the original simplified medoid subtree. By randomly re-labelling the clusters, we are simulating examples with distinct global structures. Figure 1 shows the R-F distances produced by adding noise and permuting the class labels. The simplified medoid subtree trees generated by adding noise were significantly closer to the original simplified medoid subtree than those generated by randomly permuting the class labels in terms of R-F distance.

3.2 Using the Tool

To demonstrate how to use the tool, we explore the MNIST data in detail. Again, the 784×784-pixel images were flattened and 1,500 samples were randomly sampled. A PCA pre-processing step was applied prior to applying UMAP [13] to construct a two-dimensional embedding. To demonstrate the tool's utility, we study a k-means clustering instead of the true class labels (Figure 2). The reader may follow along using the `run_example()` function in our *DRtools* package.

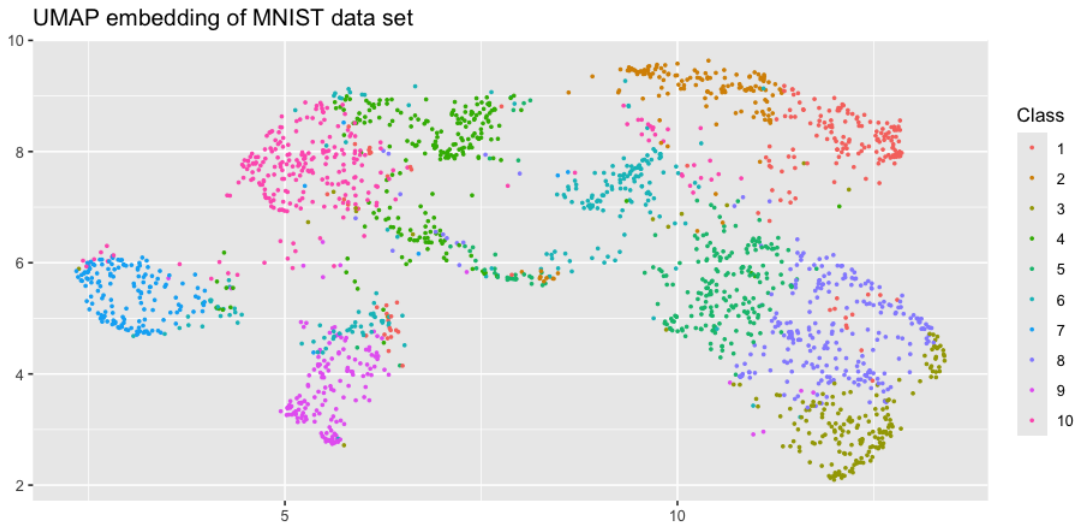

Figure 2: MNIST Embedding with k-Means Coloring

3.2.1 Classes 1 and 2

According to the UMAP embedding, classes 1 and 2 may have been incorrectly separated by the k-means clustering.

References

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science, vol. 1973*, 2001.

[2] Bhaswar B. Bhattacharya. A general asymptotic framework for distribution-free graph-based two-sample tests. *Journal of the Royal Statistical Society Series B: Statistical Methodology 81:3, 575-602*, 2019.

[3] Hao Chen and Jerome H. Friedman. A new graph-based two-sample test for multivariate and object data. *Journal of the American Statistical Association 112:517, 397-409*, 2017.

[4] Hao Chen, Xu Chen, and Yi Su. A weighted edge-count two-sample test for multivariate and object data. *Journal of the American Statistical Association 113:523, 1146-1155*, 2018.

[5] Andy Coenen and Adam Pearce for Google PAIR. Understanding UMAP. *https://pair-code.github.io/understanding-umap/.*

[6] Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine 29:6*, 2012.

[7] Persi Diaconis and David Freedman. Asymptotics of graphical projection pursuit. *The Annals of Statistics 12:3, 783-815*, 1984.

[8] Vito Di Gesù and Valery Starovoitov. Distance-based functions for image comparison. *Pattern Recognition Letters 20:2, 207-214*, 1999.

[9] González et al. CCA: An R package to extend canonical correlation analysis. *Journal of Statistical Software 23:13*, 2008.

[10] J. C. Gower and G. J. S. Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 18:1, 54-64*, 1969.

[11] Jerome H. Friedman and Lawrence C. Rafsky. Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *Annals of Statistics 7:4, 697-717*, 1979.

[12] Bracken M. King and Bruce Tidor. MIST: Maximum information spanning trees for dimension reduction of biological data sets. *Bioinformatics 25:9, 1165-1172*, 2009.

[13] Leland McInnes et al. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software 3:29*. 2018.

[14] Abdul Wahab Qurashi, Violeta Holmes, and Anju P. Johnson. Document processing: Methods for semantic text similarity analysis. *IEEE*, 2020.

[15] Werner Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *Journal of Classification 20, 25-47*, 2003.

[16] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science 290:2319*, 2000.

[17] Daniel Probst and Jean-Louis Reymond. Visualization of very large high-dimensional data sets as minimum spanning trees. *Journal of Cheminformatics 12:12*, 2020.

[18] Gregory Paul M. Rozál and J.A. Hartigan. The MAP test for multimodality. *Journal of Classification 11, 5-36*, 1994.

[19] D. F. Robinson and L. R. Foulds. Comparison of Phylogenetic Trees. *Mathematical Biosciences 53, 131-147*, 1981.

[20] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to Use t-SNE Effectively. *Distill*, 2016.

[21] Shengqiao Li. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics 4(1):66-70*, 2011.