Evaluating Dimensionality Reduction Methods

(Dated: December 7, 2019)

Abstract-Dimensionality reduction is a series of techniques in machine learning and statistics to reduce the number of features to consider by obtaining a set of principal variables. One of major advantages of performing dimensionality reduction is the ease of visualization into 2 or 3 dimensions. It also facilitates in compression, improving the accuracy and running time. In this work, we have evaluated different dimensionality reduction techniques (both linear and non-linear) on a diverse set of datasets including tabular, images, and word embedding, and evaluated the performances both quantitatively and qualitatively.

Keywords- Dimentionality Reduction , PCA, t-SNE, UMAP, Autoencoder ,GLOVE, Ciphar, word2vec,Tabular , Kernal PCA,Isomap

I. INTRODUCTION

Dimensionality reduction plays a crucial role in machine learning and data science in terms of preprocessing and visualization of data.Dimensionality reduction techniques effectively convert the dataset X with dimensionality D into a new dataset Y with dimensionality d [1], by preserving the relevant structure(i.e., geometry of the data). Dimensionality reduction is beneficial not only for the visualization but also for compression, improving the accuracy and performance of the model.

We have done a background study on dimensionality reduction techniques such as PCA,t-SNE, UMAP, Isomap, and AutoEncoder. We have then implemented them on different datasets such as image,tabular,word-embedding.
Firstly, we have done the qualitative analysis by visualizing the datasets in two dimensions after performing the dimensionality reduction techniques mentioned above.We then projected the results in 2 dimensions for the analysis and display purpose. We also combined it with the color-coding of the different class labels in the plot to get a clear picture of how well the new dimensions capture the real structure in the data. In an ideal scenario, there should be cleanly separated clusters in the new 2-dimensional feature space pertaining to different class labels.
Secondly,we proceeded with the quantitative analysis measured the performance in terms of accuracy,run time and also used other few metrics like mantel test ,Davies-Bouldin Index etc explained in the below sections .Done the comparison and analysis of all the dimentionality reduction techniques .We also tried to analyze if the model is able to retain the important information after reducing the dimensions.

II. DIMENSIONALITY REDUCTION METHODS

A. PCA (Principal Component analysis)

The main linear technique for dimensionality reduction, principal component analysis [2], performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. This is one technique that we have implemented in almost all datasets.

B. t-SNE (t-Distributed Stochastic Neighbor Embedding)

t-SNE [3] is a technique for dimensionality reduction that is particularly well suited for the visualization of highdimensional datasets. The technique can be implemented via Barnes-Hut approximations, allowing it to be applied on large real-world datasets. The t-SNE maps the multi-dimensional data to a lower dimensional space and attempts to find patterns in the data by identifying observed clusters based on similarity of data points with multiple features.

C. UMAP (Uniform Manifold Approximation and Projection)

Uniform Manifold Approximation and Projection (UMAP)[4] is a dimension reduction technique that can be used for visualisation similarly to t-SNE, but also for general non-linear dimension reduction. The algorithm is founded on three assumptions about the data 1. The data is uniformly distributed on Riemannian manifold; 2. The Riemannian metric is locally constant (or can be approximated as such); 3. The manifold is locally connected.

D. IsoMap

Isomap stands for isometric mapping. Isomap is a nonlinear dimensionality reduction method based on the spectral theory which tries to preserve the geodesic distances in the lower dimension. Isomap starts by creating a neighborhood network. After that, it uses graph distance to the approximate geodesic distance between all pairs of points. And then, through eigenvalue decomposition of the geodesic distance matrix, it finds the low dimensional embedding of the dataset.

E. Autoencoders

An autoencoder can be defined as a neural network whose primary purpose is to learn the underlying manifold or the feature space in the dataset. An autoencoder tries to reconstruct

the inputs at the outputs. Unlike other non-linear dimension reduction methods, the autoencoders do not strive to preserve to a single property like distance, topology.

III. DATASETS

A. Image Datasets

The MNIST dataset[3] is a large database of handwritten digits that is commonly used for training various image processing systems. The Digit MNIST dataset contains 28 x 28 gray-scale images of handwritten digits from 0 to 9, belonging to 10 classes, where 60,000 images for training and 10, 000 images for testing. The Fashion MNIST dataset [2] is similar dataset to Digit MNIST, which contains Zalando's article images with 60,000 images for training and 10,000 images for testing, each 28 x 28 grayscale image, associated with one of the 10 classes. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes and we take the grayscale equivalents. We use a subset of the datasets during dimensionality reduction for the analysis and plotting.

B. Word Embeddings

Word2vec [3]is a group of related shallow two-layer neural network models that are used to produce word embeddings trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large text corpora. It produces a vector space with every single word in the corpus being allotted a corresponding vector in the space. The words sharing common contexts in the corpus are located near to one another in the space. Here, we use pre-trained word embeddings, which includes word vectors for a vocabulary of 3 million words and phrases that they trained on roughly 100 billion words from a Google News dataset. The vector length is 300 features. The GloVe [4] is an is used to get vector representations for words. It is trained on aggregated global word-word co-occurrence statistics from a large text corpus, and the ultimate representations illustrate interesting linear substructures of the word vector space. We use pre-trained word embeddings, which includes word vectors for a vocabulary of 1.2 million words and phrases that they trained on about 2 billion words. The vector length is 50 features.

C. Tabular Datasets

The Boston Housing dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. It was obtained from the StatLib archive, and has been used extensively throughout the literature to benchmark algorithms. However, these comparisons were primarily done outside of Delve and are thus somewhat suspect. The dataset is small in size with only 506 cases. The 3d S-curve dataset have three dimensions, and well defined manifolds. Real world examples usually have more dimensions and often are much noisier, the manifolds may not be well sampled and exhibit holes and large pieces may be missing. It is commonly used to showcase a dimensionality reduction technique. The wine dataset contains the results of a chemical analysis of wines grown in a specific area of Italy. Three types of wine are represented in the 178 samples, with the results of 13 chemical analyses recorded for each sample. The Type variable has been transformed into a categoric variable. The data contains no missing values and consits of only numeric data, with a three class target variable (Type) for classification.

IV. EXPERIMENTAL RESULTS

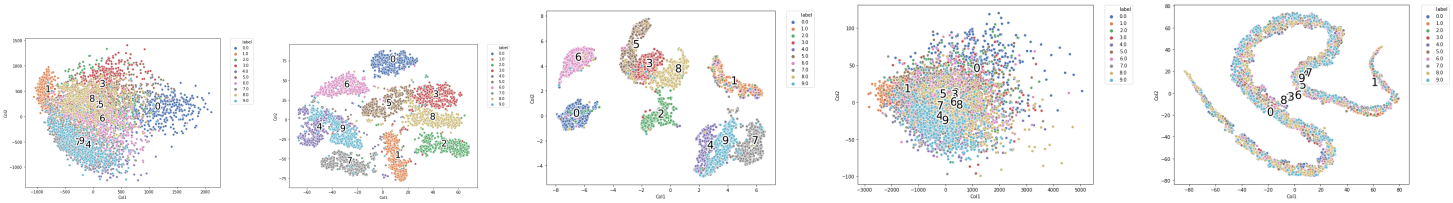
A. Dimensionality Reduction on Image Datasets

1. Qualitative Analysis

After performing dimensionality reduction on the dataset using PCA, t-SNE, UMAP, autoencoders, we are able to visualize the datasets in two dimensions. We also combine it with color coding the different class labels in the plot to get the clear picture of how well the new dimensions capture the real structure in the data. In an ideal scenario, there should be cleanly separated clusters in the new 2-dimensional feature space pertaining to different class labels.

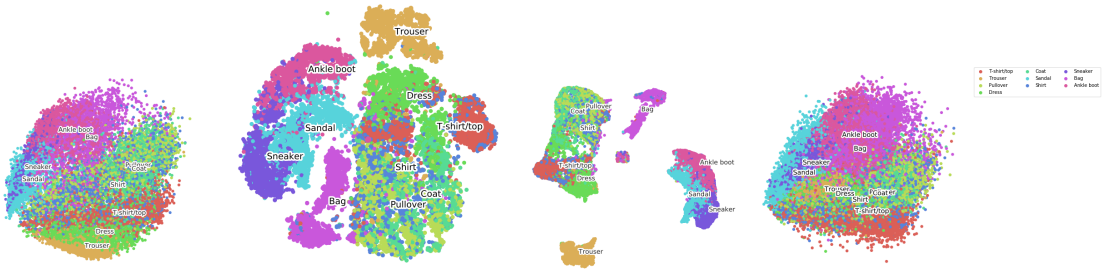
From the plots, our general observation is that that UMAP and t-SNE are successfully able to pull together clusters corresponding to similar class labels. On the other hand, the performance of PCA is very poor for both the datasets. It is clearly not capturing well the structure in the data in each class labels and hence the reduced dimensions are not good representatives of the data. We also observe some structural similarity between the images data even after reducing the dimensions to just 2. This is strongly felt in the case of cluster of digits 4 and 9 frequently appearing together and even overlapping in some cases (MNIST). Similarly in Fashion-MNIST, similar articles are clustered together for instance clusters of Sneaker, Sandal, and Ankle boot are grouped together while clusters of clothing articles like T-shirt/top, Shirt, and Coat are also grouped together. We can infer from this and the plots that t-SNE and UMAP are the strongest dimensionality reduction algorithms among all.

CIFAR is a bit more difficult dataset as compared to MNIST and Fashion MNIST. We performed the same set of dimensionality reduction techniques. After performing the dimensionality reduction, we visualized the datasets. From the given plots, we notice that PCA and t-SNE don't perform well. There are not visibly separate clusters and there is a lot of overlap between the data points belonging to different classes. UMAP was able to do slightly better than the two methods, but still not good enough. There was very slightly noticeable grouping of similar data points. For example, automobile, truck and ship were located closer to each other while dog and cat were located together. UMAP has performed a little better than the PCA and t-SNE in this regard.



a)PCAplot *b)t − SNEplot* *c)UMAPplot* *d)AutoEncoder* *Tsne − autoencodere)*

FIG. 1. Dimentionality results for digit MNIST



a)PCAplot *b)t − SNEplot* *c)UMAPplot* *d)Autoencoder*

FIG. 2. Dimentionality results for fashion MNIST

2. *Quantitative Analysis*

Apart from analyzing the quality of clustering produced using the different dimensionality reduction techniques, we have tried to quantitatively measure their performance and effectiveness. Our quantitative analysis is focused on following metrics: Davies-Bouldin Index The Davies-Bouldin index captures the intuition that dimensional reduction is good if the clusters produced using the technique are (1) well-spaced from each other and (2) themselves very dense. Lower the value of the DB index, better is the dimensional reduction.
We computed the DB indices after reducing the dimensions of the MNIST, FashionMNIST, and CIFAR datasets using PCA, t-SNE and UMAP. Then we compared the values by plotting them together. Our general observation is that UMAP produces better results than t-SNE and PCA, and t-SNE produces better results than PCA. This is evident from the lower corresponding DB indices. We also observed that none of these DR methods work well on the slightly harder dataset, CIFAR, as the DB indices are much higher. In that case, t-SNE performs the worse. Accuracy To measure how well the dimensionality reduction techniques are able to separate classes, we have trained multiple models based on random forest classifiers with the exact same parameters (nestimators=100) and tabulated the prediction accuracy on the hold out from the dataset using k-folds crossvalidation. Our tests reveal a striking observation in the case of t-SNE the accuracy (93.58%) actually increases while predicting on the MNIST dataset as compared to the accuracy on the original unreduced dataset (92.819%). This explains that the t-SNE dimensionality reduction is actually helping the classifier to avoid overfitting and this in turn leads to a higher accuracy as compared to the original dataset. The accuracy increases despite the loss of so much information contained in so many features in the original dataset. Running Time

TABLE I. Quantitative result for CIFAR data

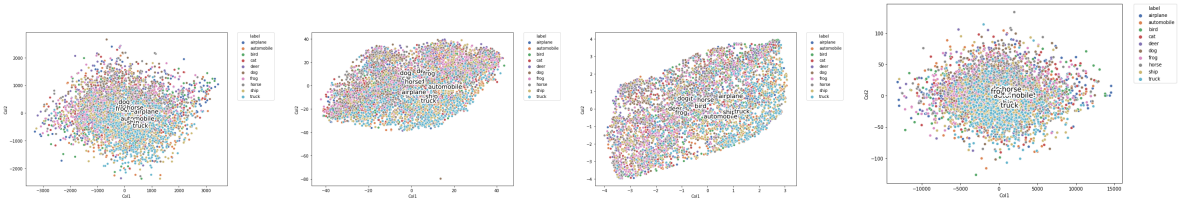
Technique	**Running Time(sec)**	**Accuracy(%)**
PCA	0.4375	14.86
t-SNE	109.984375	22.28
UMAP	17.71875	14.28
Autoencoder	48.9043	12.74

TABLE II. Quantitative result for Digit MNIST

Technique	**Running Time(sec)**	**Accuracy(%)**
PCA	0.2968	42.823
t-SNE	83.0312	30.281
UMAP	13.5937	90.501
Autoencoder	49.9805	22.18

TABLE III. Quantitative result for Fashion MNIST

Technique	**Running Time(sec)**	**Accuracy(%)**
PCA	1.05	50.25
t-SNE	830.18	81.79
UMAP	38.80	74.97
Autoencoder	46.1340	49.78



a) PCA − CIFAR *b) t − SNE − CIFAR* *c) UMAP − CIFAR* *d) CIFAR*

FIG. 3. Dimentionality Reduction on CIFAR dataset

B. Word Embeddings

Word embeddings have become the fundamental building blocks for specific natural language processing and information retrieval applications. These embeddings are learned from unlabeled text corpora, thus capture several linguistic regularities, such as analogy relationships. They are used in numerous downstream applications as well as for constructing representations for sentences, paragraphs, and documents. Reducing the size of word embeddings can improve their efficiency in memory-constrained devices, availing several real-world applications. We apply the dimensionality reduction algorithms like PCA, SVD, t-SNE, and UMAP on the famous pre-trained word embeddings like Glove and Word2Vec and analyze the performance. We apply the algorithms on a subset of the vocabulary for the analysis and visualization.

Pre-trained Word Embeddings:

1. **word2vec**

   Word2vec [3]is a group of related shallow two-layer neural network models that are used to produce word embeddings trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large text corpora. It produces a vector space, typically of several hundred dimensions, with every single word in the corpus being allotted a corresponding vector in the space. The words sharing common contexts in the corpus are located near to one another in the space. Here, we use pre-trained word embeddings, which includes word vectors for a vocabulary of 3 million words and phrases that they trained on roughly 100 billion words from a Google News dataset. The vector length is 300 features.

2. **GloVe**

   The GloVe [4] is an unsupervised learning algorithm for obtaining vector representations for words. The algorithm is trained on aggregated global word-word co-occurrence statistics from a large text corpus, and the ultimate representations illustrate interesting linear substructures of the word vector space. Here, we use pre-trained word embeddings, which includes word vectors for a vocabulary of 1.2 million words and phrases that they trained on about 2 billion words. The vector length is 50 features.

Results & Analysis:

After performing dimensionality reduction on the pre-trained word embeddings, we can visualize the words in two dimensions. We can evaluate the performance of the various dimensionality reduction algorithms by analyzing the plots. In this scenario, we can analyze the features like the similarities in the meanings of the words, connections between the syntactic usage, common words in the phrases, and how similar words or numbers get clustered together. If the dimensionality reduction algorithm is good enough, the nearby words in higher dimensions would be nearby even in the lower dimensions. After applying various dimensionality reduction algorithms to word embeddings, we get the plots shown in figure-3 and figure-4. For similarity analysis, we zoom into these plots and examine the points in the neighborhoods.

The results after applying t-SNE are in figures 7 and 11. In both the pre-trained embeddings, there are the clusters of syntactically and semantically similar words. As we can see, all the numbers in digit forms and in the word forms are placed close to each other in a lower dimension in both the embeddings. Moreover, it is also able to cluster the numbers representing years and months together appropriately. Furthermore, politics related words like 'leaders', 'vote', 'elections', 'opposition', etc. are falling under the same zone. It also finds relationship between relationship-related terms like 'parents', 'child', 'mother', 'wife', 'friends', etc. together. Thus, we can say that the t-SNE retains the intricacies of the words extremely well. We can see the results after applying the UMAP algorithm in figures 8 and 12. The words related to nations are positioned closer in both pre-trained embeddings. The words like 'play', 'player', 'games', 'fans', etc. are placed nearby in the reduced dimensions. The terms like 'growth', 'economy', 'interest', 'sales', 'prices', 'trading', etc. are also coming together. These interesting results show that the UMAP is able to preserve the complex relationship among the words.

We can analyse the time complexities of the algorithms from table IV and table V. As we can see, t-SNE takes the longest period to execute. PCA and Truncated SVD are taking very short period to execute. The UMAP takes a little longer than PCA and SVD, but it is far shorter than the runtime of t-SNE algorithm. From the above analysis, we can see that the t-SNE and UMAP are preserving complex relationships among the words. There is a trade-off between the runtime and the performance. Though PCA and SVD are much faster, they are not able to preserve the intricacies of the semantics of the words. The UMAP algorithm gives the best results in terms of runtime and performance. It can preserve non-linearities, semantics, and similarities among the words. At

a) PCA *b) SVD* *c) t − SNE* *d) UMAP*

FIG. 4. Dimensionality results for GLOVE

a) PCA *b) SVD* *c) t − SNE* *d) UMAP*

FIG. 5. Dimensionality results for word2vec

the same time, it takes a significantly short period to reduce the dimensions compared to the t-SNE.

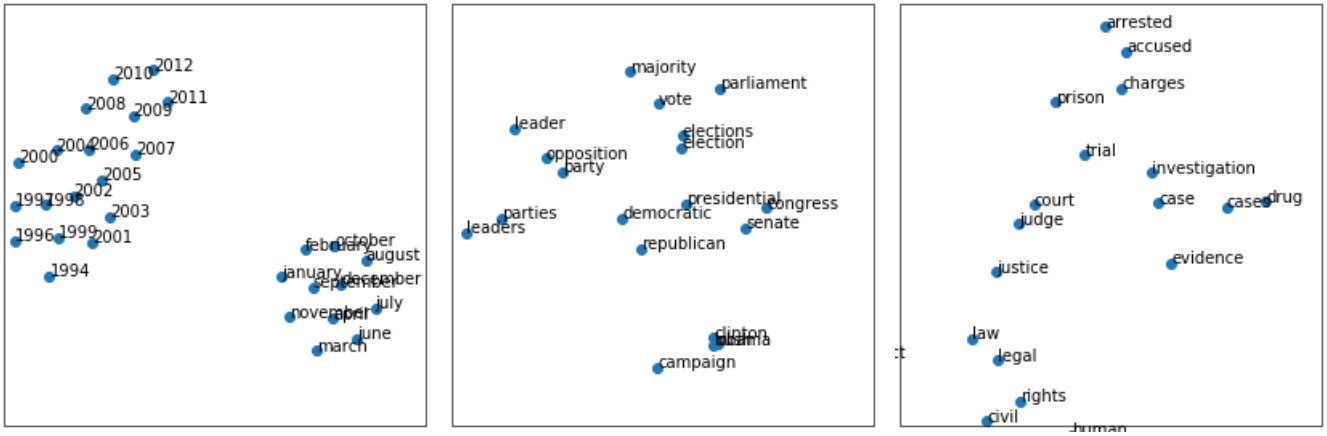
C. Tabular Datasets

Tabular or Structured datasets are widely used in variety of usecases. In this section we apply the dimensionality reduction algorithms on several tabular datasets and compute various the metrics to estimate the quality of the algorithm quantitatively. There are a number of test data sets that are often used to showcase a dimensionality reduction technique. Common ones being the 3d S-curve and the Swiss roll, among others. Moreover we use the popular datasets like Boston Housing and Wine quality dataset for the evaluation purpose.

We can see the lower-dimensional embeddings of the 3D S curve data after applying algorithms like PCA, UMAP, TSNE, ISOMAP and Kernel PCA. As we can see PCA and Kernel PCA try to preserve the S shape of the curve. The results of TSNE and UMAP are quite similar. This is because PCA focuses on the global structure, but UMAP and TSNE try to fit the local structure as well. ISOMAP does not seem to retain the useful information in the lower dimensions. Thus, we can say that t-SNE and UMAP is very good a maintaining close and medium distances for the given data set, whereas PCA is only better at maintaining the very large distances. The large distances are dominated by the overall bent shape of the S in 3D space, while the close distances are not affected by this bending. The lower dimensional embeddings for boston housing and wine quality datasets also give similar insights.

We have also studied the effect of neighborhood size on the quality measure of the dimensionality reduction algorithms for above dataset. The plots show how Rnx varies with the K neighborhood size on the log scale. We can note that the PCA and Isomap algorithms generally increase the score when we increase the neighborhood size. If the value for k is too low, the inner structure of the manifold will still be recovered, but it will be imperfect, therefore the score is lower than optimal. If k is too large, the error of the embedding is much larger due to short circuiting and we observe a very steep drop in the Qlocal score. The short circuiting can be observed in Figure 3e with the edges that cross the gap between the tips and the center of the S-shape. It is suprising that the kernel pca performance decreases with the neighborhood size. This can be dependent upon the type of kernel. UMAP and TSNE increase the quality score with the neighborhood size till some limit. After that the score starts declining. Thus, we can conclude that the neighborhood size is very crucial parameter for measuring the quality of the results of any dimensionality reduction algorithm.

We have shown the metrics for each algorithm and dataset in the tables. From local and global Q-scores for all the 3 datasets we can observe that tSNE and UMAP are better at local relationships but PCA and Isomap work well for global relationships. Kernel PCA sometimes works well and in some cases it fails. We can note that the PCA is performing best at cophenetic correlation and distance correlation. Also, it can be noted that UMAP sometimes work better than TSNE algorithm. Thus, we can say that for the tabular datasets where there are comparatively less number of features and relationship between them is not complex, PCA works well maintaining the global relationships. At the same time, UMAP and



a) Month – Years *b) Politics* *c) Law*

FIG. 6. t-SNE on GLOVE data

a) Stocks – Economics *b) Numbers* *c) Nations*

FIG. 7. UMAP on GLOVE data

TSNE also give very good results while maintaining the local complexities. We also saw that the neighborhood size is very important factor defining the quality score of the algorithm.

TABLE IV. Quantitative result(Running Time(sec)) for GLOVE and word2vector

Technique	GLOVE	Word2vector
t-SNE	7.38	7.92
UMAP	2.74	2.905
PCA	0.0146	0.0160
Trucated SVD	0.0206	0.0367

V. METRICS

There is no straightforward way to directly measure the quality of any output or to contrast two methods by an objective measure like for instance modeling efficiency or classification error. This is because every method optimizes a different loss function, and it would be unreasonable to compare t-SNE and PCA by means of either recovered variance or KL-Divergence. One good measure would be the reconstruction error, i.e., reconstructing the original data from a limited number of dimensions, but not many methods provide forward and inverse mappings. However, there is a set of independent estimators on the quality of a low-dimensional embedding.

A. Cophenetic correlation

An old measure originally developed to compare clustering methods in the field of phylogenetics is a cophenetic correlation . It is a measure of how faithfully a dimensionality reduction method preserves the pairwise distances between the original unmodeled data points. This method consists simply of the correlation between the upper or lower triangles of the distance matrices (in dendrograms they are called cophenetic matrices, hence the name) in a high and low dimensional space. Additionally the distance measure and correlation method can be varied. Some studies use a measure called

"residual variance"which is defined as

$$1 - r2(D, \hat{D}),$$

where r is the Pearson correlation and D, $\hat{D}$ are the distances matrices consisting of elements dij and $\hat{d}ij$ respectively

B. Reconstruction error

The fairest and most common way to evaluate the quality of a dimensionality reduction when the method provides an inverse mapping is the reconstruction error. It is useful when we want to do some sensitivity analysis. The reconstruction error can be exactly computed from the eigenvalues of the covariance matrix. The root mean squared error is defined as: RMSE = Formula with $\hat{x}'i = f1(yi)$, $f-1$ being the function that maps an embedded value back to feature space.

C. Co-ranking matrix-based measures

The co-ranking matrix is a way to capture the changes in ordinal distance. As before, let dij = d(xi, xj) be the distances between xi and xj , i.e., in high dimensional space and d^ij = d(yi, yj) the distances in low dimensional space, then we can define the rank of yj with respect to yi
r^ij $= |k : (\hat{d}ik < dij) or (\hat{d}ik = dij and 1 \leq k < j \leq)|$ and, analogously, the rank in high-dimensional space as:
r^ij $= |k : (dik < dij) or (dik = dij and 1 \leq k < j \leq)|$ where the notation $|A|$ denotes the number of elements in a set A. This means that we simply replace the distances in a distance matrix column-wise by their ranks. Therefore rij is an integer which indicates that xi is the rij -th closest neighbor of xj in the set X. The co-ranking matrix Q then has elements

$$qkl = |(i, j) : \hat{r}ij = k and rij = l|$$

which is the 2d-histogram of the ranks. That is, qij is an integer which counts how many points of distance rank j became rank i. In a perfect DR, this matrix will only have non-zero entries in the diagonal; if most of the non-zero entries are in the lower triangle, then the DR collapsed far away points onto each other; if most of the non-zero entries are in the upper triangle, then the DR teared close points apart. A good embedding should scatter the values around the diagonal of the matrix. If the values are predominantly in the lower triangle, then the embedding collapses the original structure causing far away points to be much closer; if the values are predominantly in the upper triangle the points from the original structure are torn apart. Nevertheless this method requires visual inspection of the matrix. For an automated assessment of quality, a scalar value that assigns a quality to an embedding is needed.

A number of metrics can be computed from the co-ranking matrix. For example:

$$Q_{NX}(k) = \frac{1}{kn} \sum_{i=1}^{k} \sum_{j=1}^{k} qij$$

which is the number of points that belong to the k-th nearest neighbors in both high- and low-dimensional space, normalized to give a maximum of 1. This quantity can be adjusted for random embeddings, giving the Local Continuity Meta:

$$LCMC(k) = Q_{NX}(k) - \frac{k}{n-1}$$

The above measures still depend on k, but LCMC has a well defined maximum at kmax. Two measures without parameters are then defined:

$$Q_{local} = \frac{1}{k_{max}} \sum_{k=1}^{k_{max}} Q_{NX}(k)$$

$$Q_{global} = \frac{1}{n - k_{max}} \sum_{k=k_{max}}^{n-1} Q_{NX}(k)$$

These measure the preservation of local and global distances respectively. The original authors advised using Qlocal over Qglobal, but this depends on the application.

LCMC(k) can be normalized to a maximum of 1, yielding the following measure for a quality embedding :

$$R_{NX}(k) = \frac{(n-1)Q_{NX}(k) - k}{n - 1 - k}$$

where a value of 0 corresponds to a random embedding and a value of 1 to a perfect embedding into the k-ary neighborhood. To transform $R_{NX}(k)$ into a parameterless measure, the area under the curve can be used:

$$AUC_{\ln k}(R_{NX}(k)) = \frac{(\sum_{k=1}^{n-2} R_{NX}(k))}{(\sum_{k=1}^{n-2} \frac{1}{k})}$$

This measure is normalized to one and takes k at a log-scale. Therefore it prefers methods that preserve local distances. In R, the co-ranking matrix can be calculated using the the coRanking::coranking function. The dimRed package contains the functions Q_{local}, Q_{global}, Q_{NX}, LCMC, and R_{NX} to calculate the above quality measures in addition to AUC-lnK-R-NX. Calculating the co-ranking matrix is a relatively expensive operation because it requires sorting every row of the distance matrix twice. It therefore scales with $O(n^2 logn)$. There is also a plotting function plot-R-NX, which plots the R_{NX} values with log-scaled K and adds the $AUC_{\ln K}$ to the legend

D. Davies-Bouldin Index:

The formula for the Dunn Index is as follows:

$$DB = \frac{1}{n} \sum_{1}^{n} max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

where i, j and k are each indices for clusters, d measures the inter-cluster distance and d' measures the intra-cluster difference. The Davies-Bouldin index [5]captures the intuition

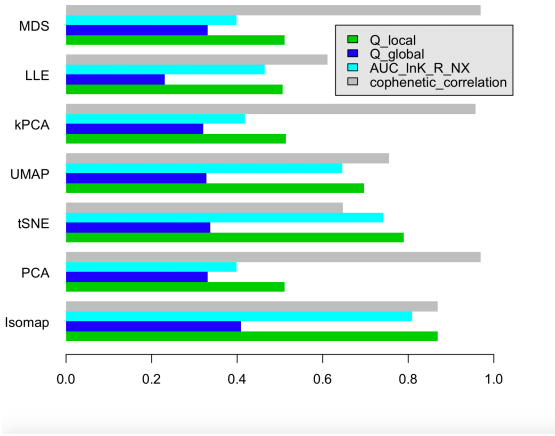


FIG. 8. C tabular dataset

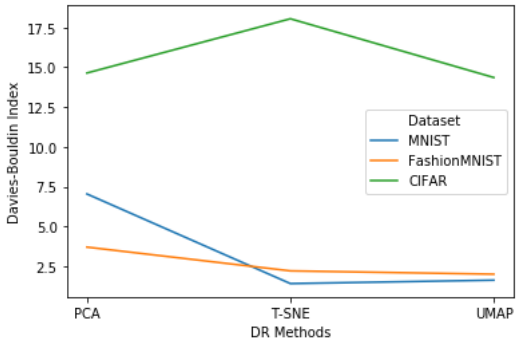


FIG. 9. Davies Bouldin Index

that dimensional reduction is good if the clusters produced using the technique are (1) well-spaced from each other and (2) themselves very dense. This is because the measure's 'max' statement repeatedly selects the values where the average point is farthest away from its centroid, and where the centroids are closest together. Lower the value of the DB index, better is the dimensional reduction.

We computed the DB indices after reducing the dimensions of the MNIST, FashionMNIST, and CIFAR datasets using PCA, t-SNE and UMAP. Then we compared the values by plotting them together. Our general observation is that UMAP produces better results than t-SNE and PCA, and t-SNE produces better results than PCA. This is evident from the lower corresponding DB indices. We also observed that none of these DR methods work well on the slightly harder dataset, CIFAR, as the DB indices are much higher. In that case, t-SNE performs the worse.

E. Mantel test

The Mantel test [6] is a significance test of the correlation between two distance matrices. Using this test, we try to compare the pairwise distances between examples in the original dataset and the distances between examples in the reduced dataset.

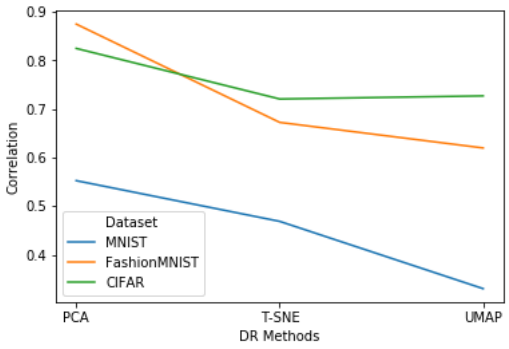


FIG. 10. mantel test

We took into account the pairwise Euclidean distance between the examples. From our observations, we learned that PCA was superior to the other DR methods in terms of retaining the correlation in the distances between examples even after dimensionality reduction. However, it still didn't mean that PCA produced better dimensionality results as this was opposite of what we observed from the visualizations.

VI. CONCLUSION

We applied the dimensionality reduction algorithms on various types of datasets including tabular datasets, image datasets and word embeddings. We noted that some algorithms work better than the others in certain settings. The PCA outperforms TSNE and UMAP when there are less number of features and relationship among them is not quite complex. But, UMAP and TSNE also give satisfying results while keeping the local relationships. We got such results with the tabular datasets. Various qualitative and quantitative metrics helped analyze overall quality and performance of the algorithms. We noted that if the use case is to keep the performance of a predictive model as high as possible even after reducing the dimensions, it is best to use t-SNE. If the use case is to reduce the features to give a good representation of their actual classes and form a good grouping of similar classes during visualization, it is best to use UMAP. Also, if it is a time-sensitive application, t-SNE is not recommended. UMAP can be a good alternative in that case. Also, for the case of word embeddings, UMAP turns out to be giving the best results, balancing the time and performance at a time. We therefore conclude that depending on the dataset and the use case, our choice of the dimensional reduction technique can be different in different cases.

TABLE V. Quantitative result for 3D s curve Dataset

Techniques	Q local	Q global	mean-R-NX	AUC-lnK-R-NX	total correlation	cophenetic correlation	distance correlation
AutoEncoder	0.6077826	0.3276235	0.6992404	0.5530285	0.6621097	0.9283138	0.979616
Isomap	0.8671646	0.4101891	0.6803224	0.8120624	0.636807	0.8622763	0.9550563
kPCA	0.5085764	0.3209259	0.6955559	0.4115659	0.6831085	0.9542763	0.9832591
PCA	0.4996866	0.3331783	0.7471252	0.3824272	0.6674151	0.9672047	0.9848122
tSNE	0.7881571	0.2351932	0.2885991	0.672681	0.2700766	0.3066052	0.6375044
UMAP	0.6839011	0.3429575	0.5464716	0.6622713	0.3266789	0.7754763	0.9202285

TABLE VI. Quantitative result for Boston Housing Dataset

Techniques	Q local	Q global	mean-R-NX	AUC-lnK-R-NX	total correlation	cophenetic correlation	distance correlation
AutoEncoder	0.07312524	0.06416582	0.22515085	0.0670564	2.59E-01	0.54106207	0.5790622
Isomap	0.56956538	0.32030632	0.6	7677224	0.3.54E-	0.	0.9631138
kPCA	0.29050992	0.01892104	0.02678786	0.1016096	2.05E-02	0.02420813	0.2143724
PCA	0.53827723	0.35575945	0.85163396	0.5526995	4.26E-01	0.99594821	0.9977224
tSNE	0.73131679	0.30948369	0.52821373	0.6928574	3.18E-01	0.67532707	0.8315338
UMAP	0.62534241	0.33033946	0.68185068	0.6034233	4.30E-01	0.89202675	0.9613029

TABLE VII. Quantitative result for Wine Dataset

Techniques	Q local	Q global	mean-R-NX	AUC-lnK-R-NX	total correlation	cophenetic correlation	distance correlation
AutoEncoder	0.001715924	-0.045735881	-0.12081039	-0.03629985	N/A	N/A	0
Isomap	0.897041437	0.480129659	0.993996372	0.95236159	0.29581766	0.99999688	0.9999981
kPCA	0.325810957	0.009091157	0.004859261	0.11218131	0.01978011	-0.04809024	0.1517153
PCA	0.926749521	0.473391721	0.99627997	0.96369176	0.29707065	0.99999904	0.9999993
tSNE	0.784871319	0.401764589	0.854774023	0.78226538	0.38102555	0.89397694	0.9792273
UMAP	0.67794586	0.353466214	0.659410345	0.66667627	0.30482866	0.70021788	0.9033309

[1] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, no. 66-71, p. 13, 2009.

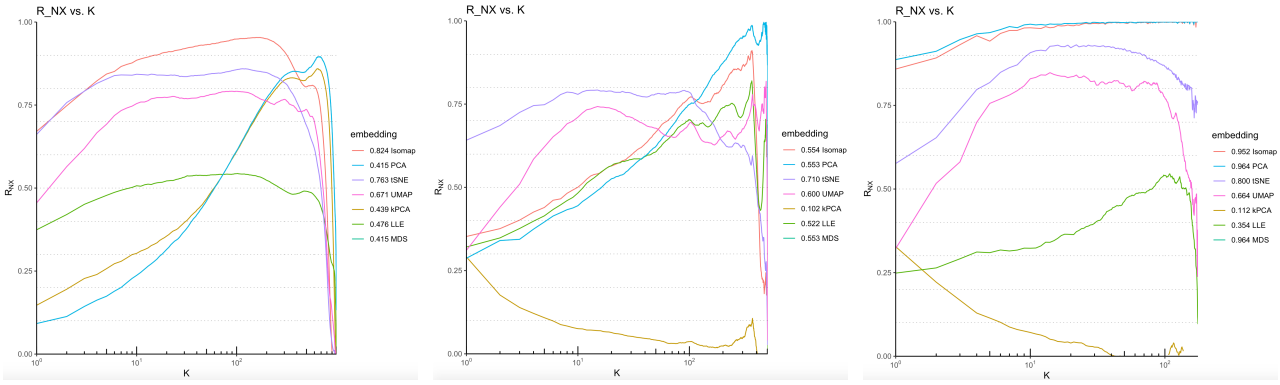
[2] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, " *arXiv preprint arXiv:1708.07747*, 2017.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[4] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

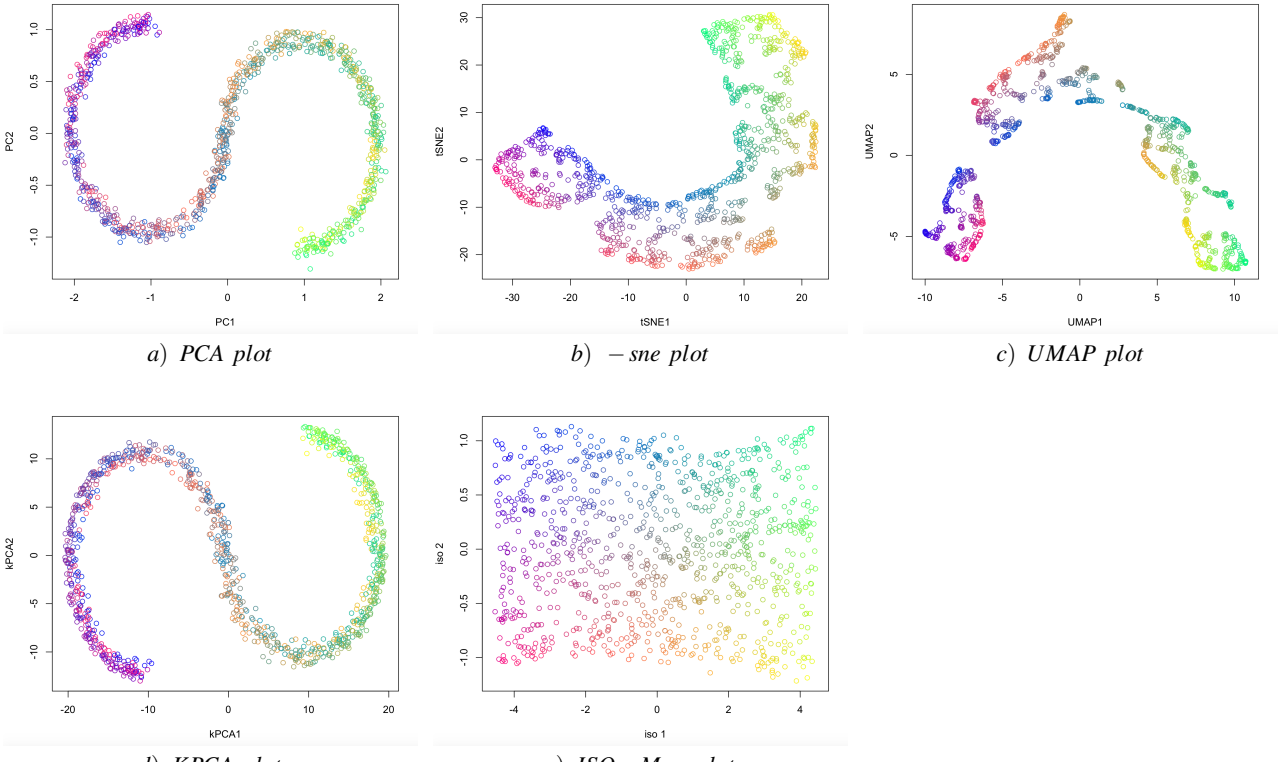
[5] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[6] N. Mantel, "The detection of disease clustering and a generalized regression approach," *Cancer research*, vol. 27, no. 2 Part 1, pp. 209–220, 1967.



a) *3DsCurve* *b*) *BostonHousing* *c*) *Wine*

FIG. 11. Effect of neighborhood size on quality measure



a) *PCA plot* *b*) $-$*sne plot* *c*) *UMAP plot*

d) *KPCA plot* *e*) *ISO* $-$ *Map plot*

FIG. 12. Dimentionality Result for 3D S curve dataset