Measuring Variance Propagation in t-SNE

Emile Timothy Anand, Sai Advaith Maddipatla, Katelyn Chu
Computing and Mathematical Sciences
California Institute of Technology, Pasadena, California
{eanand,amaddipa,kchu}@caltech.edu

Abstract

T-distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction technique used to visualize high-dimensional data. This report seeks to quantify how noise (as measured by the variance of a dataset) is propagated by t-SNE. The experiments outlined in this report conclusively demonstrate that t-SNE is not a variance-invariant embedding, even when there is a ground truth knowledge of the uncertainty of the dataset.

1 Introduction

T-SNE (t-distributed stochastic neighbor embedding) is a dimensionality reduction embedding technique that is used to visualize high-dimensional data in a two dimensional space, such that t-SNE: $\chi = (x_1, ..., x_N) \rightarrow Y = (y_1, ..., y_N)$ for $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}^2$, where d is the dimensionality of the dataset. For each pair of points $x_i, x_j \in \chi$ such that for $i \neq j$, t-SNE computes a conditional probability distribution $p_{j|i}$ based on their similarity in the dataset (2):

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/(2\sigma_i^2))}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/(2\sigma_i^2))} \Rightarrow p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N} \tag{1}$$

Each point $x_i \in \chi$ then has a natural characterization with the Gaussian probability density function, where the mean is centered at x_i. Using this probability distribution, t-SNE seeks to learn $y_1, ..., y_N \in \mathbb{R}^2$. Specifically, t-SNE computes a conditional probability distribution $q_{j|i}$, which measures the similarity for each pair of points y_i, y_j such that $i \neq j$:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|y_k - y_l\|^2)^{-1}} \tag{2}$$

The t-SNE algorithm learns $y_1, ..., y_N \in \mathbb{R}^2$ by using a gradient-descent algorithm (or the Barnes Hut approximation) to minimize the Kullback-Leibler (KL) divergence (equivalent to the relative entropy) of the distributions of p_{ij} and q_{ij}, defined by:

$$KL(P\|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3}$$

When the t-SNE algorithm is used to embed a high-dimensional dataset onto $\mathbb{R}^2$, points that more similar in χ get mapped closer to each other in Y and tend to form clusters, whereas points that are more dissimilar are mapped further away from each other, as is observed in Figure 1.

A significant amount of progress has been made in quantifying the uncertainty of classification models (3). For instance, several deep learning algorithms commonly use Bayesian techniques to effectively model the uncertainty of their predictions, using Bayesian Neural Networks, Monte Carlo Dropouts, and Markov Chain Monte Carlo (MCMC) methods. Similarly, progress has been made to classify the uncertainties of variational autoencoders by using MCMC methods. However, in the

context of dimensionality reduction, while a significant amount of research progress has been made towards embedding large datasets using dimensionality reduction tools such as t-SNE and UMAP (Uniform Manifold Approximation and Projection), there has been no prior work on quantifying the propagated variance of dimensionality reduction embeddings such as t-SNE and UMAP.

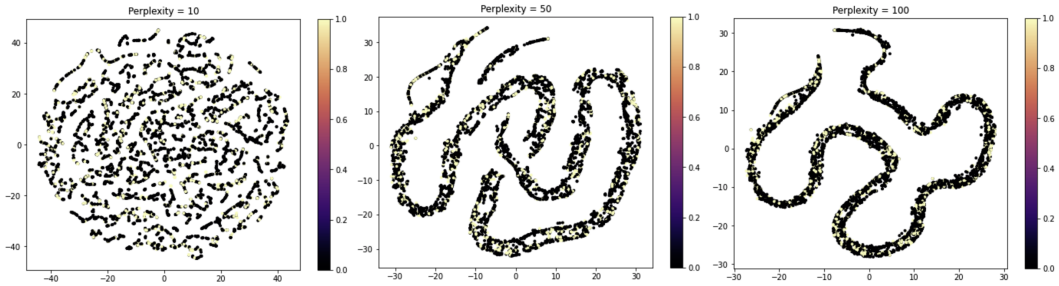

Figure 1: The t-SNE outputs of 600 objects from the the ogbg-molhiv dataset with a perplexity (dimensionality reduction parameter) of 10, 50, and 100, respectively. Low perplexity values form local complex structures, whereas higher perplexity values form simpler global structures with more significant clustering.

Typical real-world scientific data has several sources of uncertainty and noise. For instance, in *The Specious Art of Single-Cell Genomics*, when single-cell data is embedded by the t-SNE algorithm, it distorts the properties of the cells, which indicates that t-SNE should only be used as a tool to visualize high-dimensional data, rather than for other downstream tasks (7). Indeed, it is essential that the uncertainty of a dataset reflects in the transformed dataset, especially for transformations that reduce its dimensionality such as t-SNE. Hence, it is valuable to measure the uncertainty of a dataset after it has undergone the process of dimensionality reduction. Ultimately, understanding how these uncertainties propagate in the embedding will be of critical value to real-world problems, since knowledge of the propagated uncertainties can provide reliable quantified confidence estimates to hypotheses made by looking at visual embeddings produced by t-SNE.

2 Contributions

This project tackles the fundamental problem of quantifying the propagated variance between datasets and embeddings produced by the t-SNE algorithm. We hope to use the knowledge of this to apply a confidence gradient in the form of a density envelope around each cluster in the embedding produced by the t-SNE algorithm to visualize the uncertainty of the original dataset. This report studies the cases for when there is 1) a known pointwise error estimate in the dataset, and for when 2) the quantitative uncertainty for the points in the dataset are unknown. For each of these cases, this report seeks to determine whether t-SNE propagates the variance of a dataset, and, if so, by how much? Ultimately, our vision for this project is to be able to identify potentially corrupt and ambiguous outputs of t-SNE embeddings, and to quantify how corrupt these outputs might be.

3 Methodology

This section details our approach to quantifying the uncertainty propagated by t-SNE when there is a known pointwise error estimate for a synthetic dataset, and for where the quantitative uncertainties of the points in the dataset are unknown (i.e. any natural dataset like MNIST).

3.1 Quantifying Uncertainty Propagation when the Pointwise Error is Unknown

To quantify the uncertainty propagation between a dataset and its embedding using the t-SNE algorithm, an appropriate dataset must be chosen. Without loss of generality, any algorithm or pipeline that can quantify the uncertainty propagation in one dataset can be used to quantify the uncertainty propagation for any another dataset. Hence, for the purpose of this experiment, the MNIST dataset will be used (8). MNIST is a valuable utility, since it is a big dataset corresponding to noisy real-world data (thus, containing variance) that has a finite number of classes.

3.1.1 Approach I: Boostrapping

Now that we have a dataset in hand, our focus is to to determine a way to find the distribution of the variance of the classes of the original MNIST dataset and the embedding of MNIST produced by t-SNE. To do this, we shall use the bootstrapping (9) method to approximate the statistics of the MNIST dataset. Let the first 10 classes of MNIST be denoted by $\{D_0, D_1, ..., D_9\}$, where for $i \in \{0, ..., 9\}$, D_i is the set of images that are labeled with i in the dataset. Next, for $i \in \{0, ..., 9\}$, consider all the sub-datasets $D_i \subset \chi$. Let $N_i = |D_i|$. Then, $\forall i \in \{0, .., 9\}$, we randomly sample n images from D_i and estimated their variance, which we averaged over K iterations.

To estimate the variance of a single sample of n images, we took the mean-value of the images (pixel by pixel) and determined the variance of the Laplacian of the mean image. The Laplacian of an image can be used to effectively extract the variance of an image (10) by detecting the high-variance features of the image, such as the edges. Thus, by using the Laplacian of the n images, we extracted an estimate of the uncertainty of those n images, and averaged all these bootstrapped variance estimates to get an estimate of the variance of D_i. To bootstrap, we consider n samples from D_i. Let the j'th bootstrap from the K iterations be S_j, such that $S_j = (X_1, X_2, ..., X_n) \sim D_i$. We then obtain the mean image from S_j ($\overline{S_j}$) and estimate $\hat{\sigma}_j^2$ (the bootstrapped estimate of variance obtained) by using the laplacian of $\overline{S_j}$. By averaging $\sigma_j^2, \forall j \in \{1, ..., K\}$, this yields an accurate estimate of the variance of D_i. In order for the estimate to be accurate, we need to verify that K is large. Thus, let us label this variance as $\tilde{\sigma_i^2}$ We will repeat this for every classes i in the dataset.

Next, we projected the entire MNIST dataset onto 2 dimensions using t-SNE to produce Figure 2a. We then used the same method as earlier to quantify the variance of the embedding produced by t-SNE. However, in this case, instead of using the Laplacian to compute the variance, we centered a 2-dimensional centroid on the mean point for each cluster and calculated the set of Euclidean distances of each point from the center of the centroid. Using this method, we calculated $\hat{\sigma_i^2}$, the variance of the 2-dimensional embeddings produced by t-SNE. Then, using the estimates of the variances of the 2-dimensional embeddings from t-SNE, we computed the absolute value of the difference in variances to envelope the uncertainties of the clustering. This difference in variance was denoted by $\delta_i = |\hat{\sigma_i^2} - \tilde{\sigma_i^2}|$ for each class i from 0 to 9.

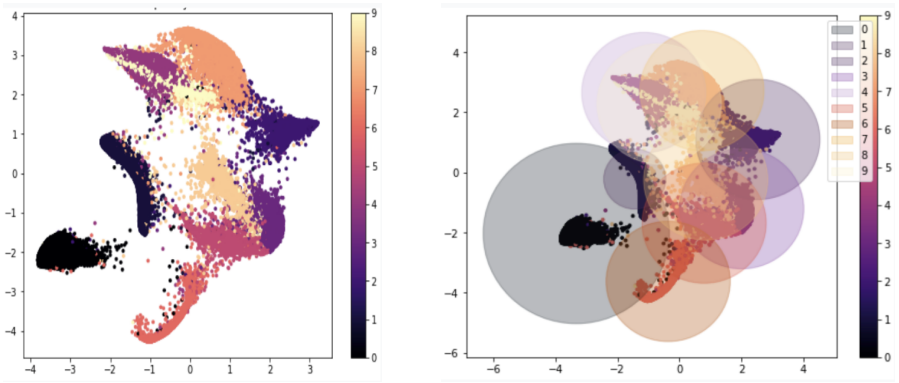

Figure 2A: A 2D t-SNE embedding of MNIST.
Figure 2B: Variance envelopes (centroids) for each class in MNIST.

3.1.2 Approach II: Quantifying the Correlation Between t-SNE and PCA

In this section, we detail another approach to estimate the variance propagation of t-SNE when we do not have pointwise error estimates for the dataset. This method is necessitated since there are shortcomings in the previous experiment. As demonstrated previously, even though we can use the Laplacian to quantify the variance of the mean image of the sample, it is not a reliable metric of variance for other datapoints (the mean is not a reliable quantification) to represent the dataset). Furthermore, the Laplacian does not converge in probability to the actual variance of the sample. This means that as the sample size $n \to \infty$, the averaged sample variance must converge to the true

variance of D_i. This is not guaranteed to happen, however, since the Laplacian and the mean are not fully reliable statistical measures. It is a well-known fact that PCA (Principal Component Analysis) is a variance-invariant dimensionality reduction method: the greater the number of dimensions the MNIST dataset is projected down to, the greater the variance of the original dataset that is preserved. So, one can use PCA to project the MNIST dataset down to a single dimension to robustly estimate a significant amount of the variance in the dataset.

We first projected MNIST onto one-dimension using PCA, and embedded the MNIST down to 1D using t-SNE (instead using t-SNE to minimize the KL divergence for $y \in \mathbb{R}$). We then performed the bootstrapping method on the naïve variances to find the variance of each class. We then repeated the experiment with the dataset projected down to 2D. However, in this case, we computed the covariance matrix $\Sigma = \frac{1}{d-1}\sum_{i=1}^{d}(x_i - \bar{x})(x_i - \bar{x})^T$, where d is the number of dimensions we are projecting down to, and compared the covariance matrices using their spectral density. Finally, we repeated this experiment by projecting the dataset down to 3D using t-SNE (instead using t-SNE to minimize the KL-divergence for $y \in \mathbb{R}^3$). In our results, we present statistics of bootstrapped variances and find their correlations to quantify the extent of variance propagation by the t-SNE algorithm.

3.2 Known Pointwise Error Estimates

To study the case for when there is a known pointwise error in a dataset, we constructed a 20 dimensional dataset with 20 classes and 6000 points. To construct the 300 points in each class, 20 random points $\{z_i\}_{i=1}^{2}0$ were sampled from $(1, 10)$. For each z_i, 300 values of Gaussian noise, $\{e_{z_i}\}_{j=1}^{300}$ were sampled randomly from $\mathcal{N}(0.2, \sigma_{z_i}^2)$, for $\sigma_{z_i} \in_U (0.0, 2.0)$. was added to the point. Thus, for all $i \in \{1, ..., 20\}$ and $j \in \{1, ..., 300\}$, we set $\{x_{z_i}\}_j = \Phi(r_{z_i} + \{e_{z_i}\}_j)$ and $\{y_{z_i}\}_j = i$, where $\Phi : \mathbb{R} \to \mathbb{R}^{20}$. Specifically, each element in the tuple produced by Φ was the output of various trigonometric functions and polynomials that are detailed in (1). An embedding of this dataset produced by the t-SNE algorithm is depicted below in Figure 3.

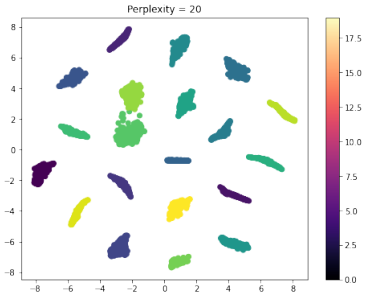

Figure 3: The t-SNE representation of a synthetic dataset with 6000 points and 20 classes, where each class is contained in its cluster. Every point in a cluster i has some added Gaussian noise sampled randomly from $\mathcal{N}(0.2, \sigma_i^2)$, where $\sigma_i \in (0.0, 2.0)$.

3.2.1 Measuring Variance Preservation through Spectral Density Correlation

Let χ denote the synthetic dataset, and let χ_i denote the data for each class $i \in \{1, ..., 20\}$. To measure the correlation of the variance in the dataset with the variance of the embedding produced by t-SNE (this is necessary before describing any possible method to quantify the overall propagated variance), every single subdataset $\{\chi_1, ..., \chi_{20}\}$ was projected to two-dimensions using PCA. Then, the covariance matrix, Σ_i, of the two-bases representation of each subdataset was computed. A singular-value decomposition (SVD) of Σ_i was performed such that $\Sigma_i = M_i \Lambda_i M_i^T$. Here, Λ_i is a diagonal matrix of the form $diag(\sigma_1, \sigma_2)$, where $\{\sigma_1, \sigma_2\}$ are the spectral densities of the first and second coordinates of the dataset, as produced by the PCA algorithm for the subdataset χ_i. From this, the spectral density $\|\Sigma\|_{\text{spec}_i} = \sigma_1^2 + \sigma_2^2$ was calculated for each subdataset χ_i.

Next, the embedding Y=TSNE(χ) was computed. Then, the covariance matrix, ζ_i, was computed for each class Y_i where $i \in \{1, ..., 20\}$. As before, the SVD of ζ_i was performed such that $\zeta_i = M_i' \Lambda_i' M_i'^T$, where Λ_i' is a diagonal matrix of the form $diag(\sigma_1', \sigma_2')$, where $\{\sigma_1, \sigma_2\}$ are the

spectral densities of the first and second coordinates of the i'th cluster in the embedding produced by the t-SNE algorithm of χ. From this, the spectral density $\{\|\zeta\|_{\text{spec}}\}_i = \sigma_1'^2 + \sigma_2'^2$ was calculated for each cluster Y_i in the embedding produced by the t-SNE of χ.

Finally, for each class i, a scatter plot of $\{\|\Sigma\|_{\text{spec}_i}\}_{j=1}^{20}$ versus $\{\|\zeta\|_{\text{spec}_i}\}_{j=1}^{20}$ was plotted, and the correlation of the scatter-plot graph was measured.

3.2.2 Measuring Correlations in Distance

Next, we measured how accurately distances (as measured by the Euclidean distance) between points in χ correlate to distances between points in t-SNE. For each cluster $\chi_i \subset \chi$ for $i \in \{1, ..., 20\}$, we found the mean, $\overline{\chi_i}$ of all the points that are contained within it, such that the j'th element of $\overline{\chi_i}$ is the mean of the j'th elements in the tuple of points in χ_i. Next, for each point $\alpha \in \chi_i$, we computed the Euclidean distances $\{d_j\}_{j=1}^{300}$. Next, for each cluster $Y_i \subset Y$ for $i \in \{1, ..., 20\}$, we found the mean $\overline{Y_i}$ for all the points that are contained within it, such that the j'th element of $\overline{Y_i}$ is the mean of the j'th elements in the tuple of points in Y_i. Next, for each point $\beta \in Y_i$, we computed the Euclidean distances $\{d_j'\}_{j=1}^{300}$. We then plotted the 20 distance correlation graphs and measured the correlations of the corresponding points.

4 Results

This section describes the results obtained by performing the experiments detailed above.

4.1 Quantifying Uncertainty Propagation when the Pointwise Error is Unknown

4.1.1 Approach I: Bootstrapping

We first found the differences between the variance estimates of the 2D embedding produced by t-SNE and the original dataset, where the variances were estimated using bootstrapping. We used these differences as a metric of the uncertainty in each clustering to form an envelope that was centered at the mean point (expected value) of the cluster. When performing the boostrapping algorithm, we used $n = 1500$ samples from the dataset, and estimated the variance over $K = 5000$ iterations, while ensuring that each point in the dataset had an equal likelihood of being sampled. Finally, we used the scikit-learn implementation of t-SNE, where we set constant parameters of 250 iterations and a perplexity of 20. These values were parameterized to ensure a fast run-time (t-SNE runs in $O(n!)$ time) and a qualitative output. The output of this experiment is depicted below in Figure 4.

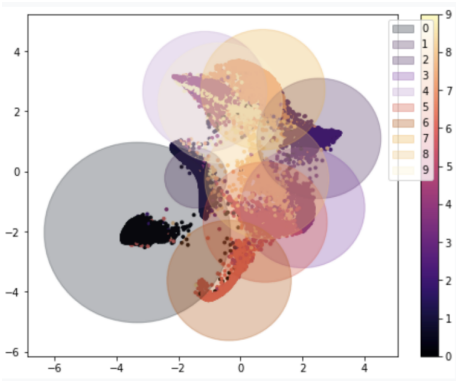

Figure 4: The 2D embedding of MNIST using t-SNE with a variance envelope computed using the bootstrapping techniques.

Figure 4 demonstrates that there is some degree of variance propagation in the t-SNE algorithm, since otherwise each variance envelope would have a radius of 0. However, it is also evident by the non-uniformity of the radii of the variance envelopes that the variance of the original MNIST dataset is not uniformly propagated in the t-SNE algorithm, since otherwise all the radii would be

equal. Regardless, this is not sufficient evidence of our hypothesis that t-SNE does not accurately propagate the variance of the dataset, since the bootstrapping technique is not a reliable statistical tool (as is explained in the methodology). Therefore, we now analyze the results produced by the second approach.

4.1.2 Approach II: Quantifying the Correlation Between t-SNE and PCA

As described in the methodology, we collected the bootstrapped sample variances using a similar bootstrapping setup as described in the previous section. For the 1-dimensional case, we produced histograms after bootstrapping with parameters $n = 1500$ samples over $K = 5000$ iterations. We observe in Figures 5 and 6 that the variances in Figure 5 (characterized by the frequency peak) is noticeably higher than the variances in Figure 6. This is an expected result, since higher dimensional spaces tend to display a higher variance. However, we sought to determine if these variances have any correlation, since it would imply that t-SNE propagates variances uniformly.

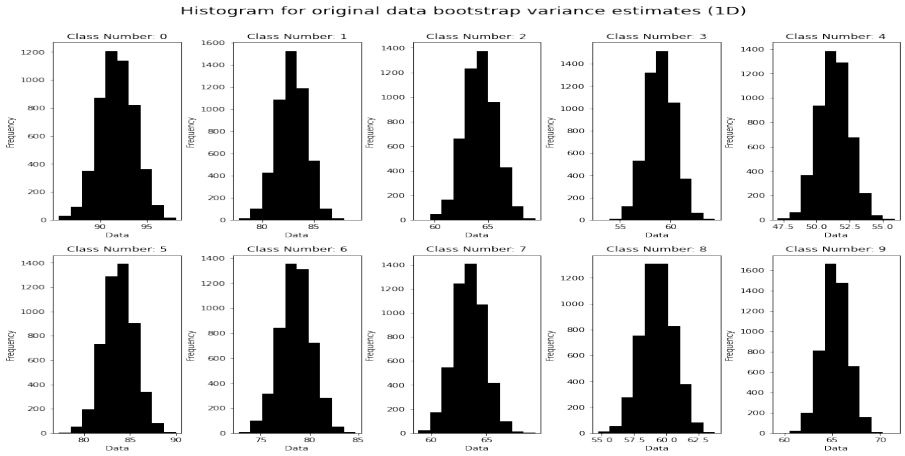

Figure 5: 1D Original Data Bootstrap Variance

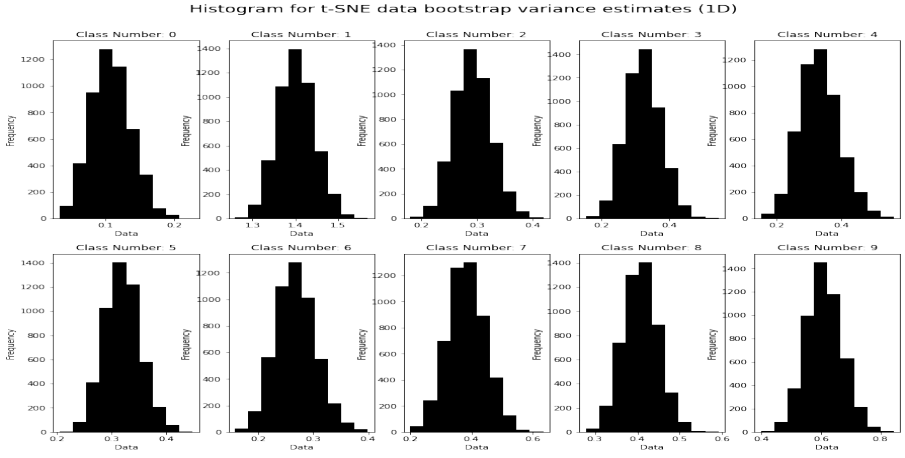

Figure 6: 1D T-SNE Data Bootstrap Variance

We used the Kolmogorov–Smirnov test (11) and the Pearson Correlation Coefficient (12) to determine the correlation of the variances. When using the Kolmogorov–Smirnov test, we observed that the p-value for each class in MNIST was approximately zero, which is indicative of the variances not originating from the same distributions. To a reasonable degree, this proves the fact that the variances are uncorrelated. Furthermore, when using the Pearson Correlation Coefficient, we observed that the pairwise correlations between the variances was, again, approximately zero, which implies negligible correlation. Hence, for the $1D$ case, the data, backed up by these statistical tests, provide evidence that the t-SNE algorithm does not propagated variance in a correlated manner.

To examine the $2D$ case, we found the covariance matrix of the data, and plotted its spectral density (the L^2-norm of the singular values of the covariance matrix) on a histogram in Figures 7 and 8. Similarly, the p-value of the Kolmogorov–Smirnov test is approximately zero, which indicates the variances having no correlation. When comparing the variances using the Pearson correlation coefficient, we observed that the correlation between the variances was approximately zero, which, again, implies negligible correlation. Hence, for the $2D$ case, the data, backed up by these statistical tests, provide evidence that the t-SNE algorithm does not propagated variance in a correlated manner.

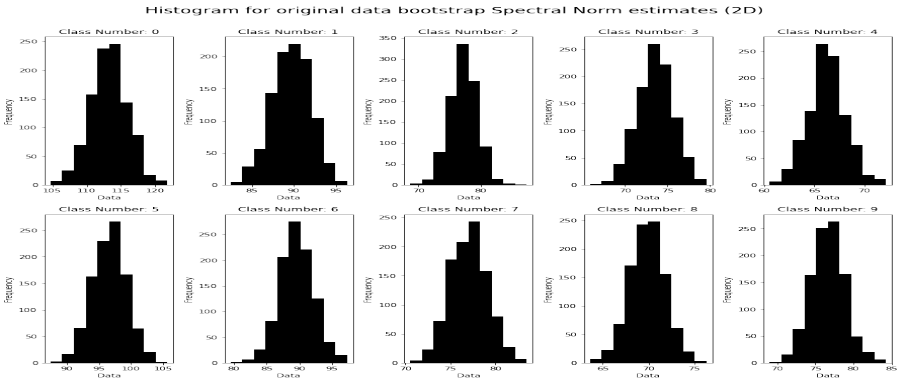

Figure 7: 2D Original Data Bootstrap spectral density

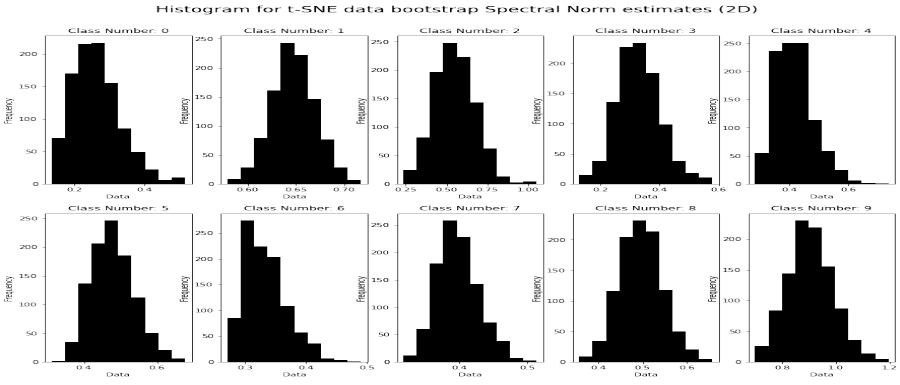

Figure 8: 2D t-SNE Data Bootstrap Spectral Density

Finally, for the $3D$ case, we computed the covariance matrix for each bootstrapped sample, and plotted the spectral density of the resultant covariant matrices on a histogram in Figures 9 and 10. We notice, using the Kolmogorov–Smirnov test and the Pearson correlation coefficients, that these results are similar to the $2D$ and $1D$ case, which again provides evidence that the t-SNE algorithm does not propagate variance in a correlated manner.

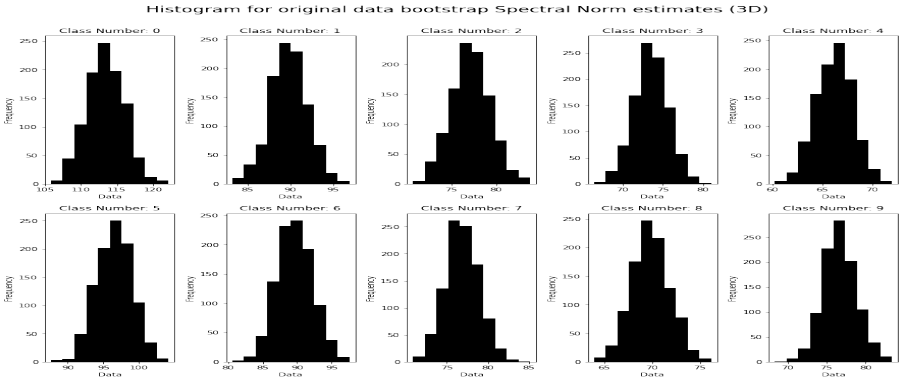

Figure 9: 3D Original Data Bootstrap Spectral Density

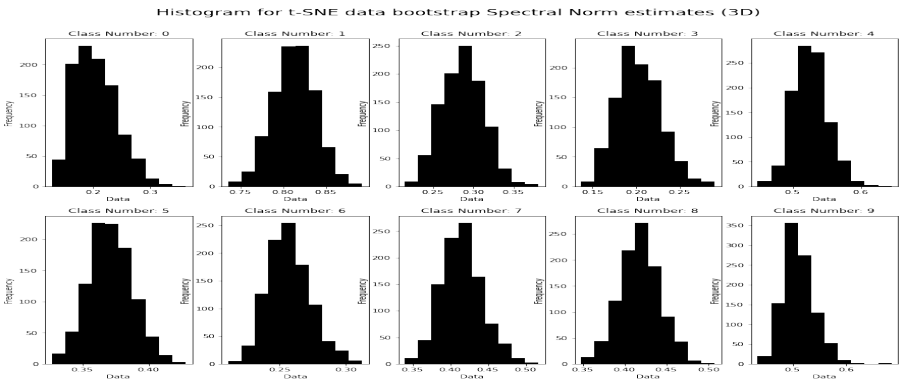

Figure 10: 3D t-SNE Data Bootstrap Spectral Density

Hence, using MNIST we have shown that for the most part, variance does not propagate through t-SNE embedding from the original dataset to the lower dimensional embedding of t-SNE.

4.2 Known Pointwise Error Estimates

4.2.1 Measuring Variance Preservation through Spectral Density Correlation

The spectral density for each cluster in the PCA of χ was plotted against the spectral density for each cluster in TSNE(χ), as described in the methodology. The scatter plot that was obtained is displayed below in Figure 11. It is observed that the scatter plot had a Pearson Correlation Coefficient of -0.222, which implies a negligible negative correlation between the spectral density.

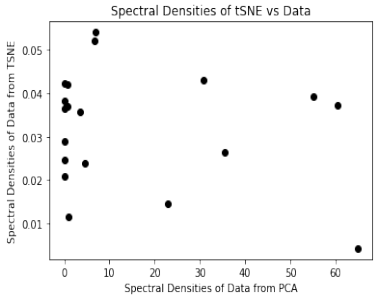

Figure 11: A scatter plot of the spectral densities of the synthetic dataset, as viewed by PCA and by the t-SNE algorithm.

The spectral density of a dataset is a value associated with the dataset's distribution, and is an effective proxy for the variance of a dataset. The negligible correlation between the spectral density of TSNE(χ) and PCA(χ) indicates that their corresponding variances do not correlate. This implies that the variance of a dataset is not uniformly propagated by the t-SNE algorithm, which corroborates findings from the previous methods that t-SNE is not a variance-invariant transformation.

4.2.2 Measuring Correlations in Distance

The previous experiments demonstrate, without a doubt, that t-SNE is not a variance-invariant transformation since it does not uniformly propagate the variance of a dataset during the embedding process. Yet, it is still valuable to test whether t-SNE uniformly propagates distance: do the Euclidean distances between points in a dataset correlate to the Euclidean distances between points in the embedding? The mathematical formulation of t-SNE forces points in the original dataset that are more similar (as measured by L^2-norm) to be mapped closer together in the t-SNE, and vice versa. Therefore, one would expect there to be a strong correlation in the distances between points in the original dataset and the distances between points in the embedding.

Thus, as described in the methodology, we computed the Euclidean distances for each point in a cluster with the (pointwise) mean of the cluster. For each point in a cluster, we plotted the Euclidean distance of the point in the original dataset versus the Euclidean distance of the point in the embedding. Figure 12 depicts the scatter-plots obtained for the first 8 classes. It is, as expected, clear that there is a strong positive correlation in these distances, which affirms that t-SNE uniformly propagates distance, upto effects of stochasticity.

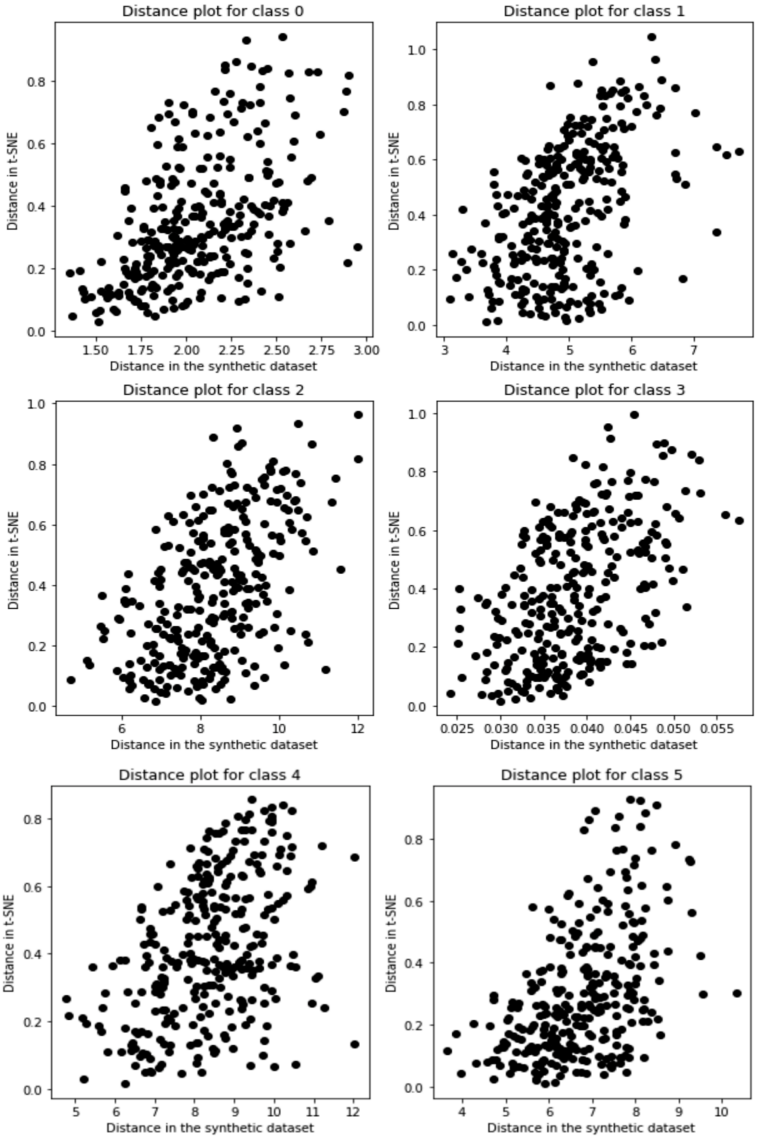

Figure 12: Scatter-plots of the Euclidean distance between every point in a cluster and its mean point for the original dataset versus the embedding produced by the t-SNE algorithm.

We then recorded the Pearson correlation coefficients of the distances from Figure 12 in Table 1.

Class	Pearson Correlation Coefficient
0	0.5259
1	0.4714
2	0.5012
3	0.4883
4	0.4238
5	0.4772

Table 1: The Pearson Correlation Coefficients of the distance plots of the first 6 classes in the dataset.

5 Conclusion and Future Work

Through this work, we effectively motivated the problem of quantifying uncertainty propagation in embeddings produced by the t-SNE algorithm. We notice that, regardless of whether the uncertainty of the dataset is known, t-SNE cannot accurately propagate uncertainties, but that it still maintains some invariant features of the dataset, such as its Euclidean distance. The implication of this result is that the lack of accurate variance propagation in t-SNE implies that its embeddings should always be viewed critically when used as a basis for forming hypotheses. However, since the mathematical formulation of t-SNE does not guarantee any preservation of variances, this result should not be viewed as a surprise. In the future, we hope to analytically prove that dimensionality reduction methods such as t-SNE and UMAP do not accurately propagate variance. We conclude with an open problem: Is there a dimensionality-reduction visualization algorithm that uniformly propagates the variance of a dataset?

Acknowledgments

We are grateful for the opportunity to have pursued this topic through the CS 159 class (*Advanced Topics in Machine Learning*) at the California Institute of Technology. We thank Professor Yisong Yue, and the TA's Christopher Yeh and Jennifer Sun for their support in this project, and for the attendees of the CS 159 poster session who provided invaluable feedback.

References

[1] Github Repository: Containing source-code and data: github.com/emiletimothy24/CS159/tree/main/data. Details of Φ, the function used to create the synthetic dataset, are in the README.md file in CS159/data

[2] van der Maaten, L. and G. Hinton (2008). Visualizing data using t-SNE. Journal of Machine Learning Research 9 (Nov), 2579–2605.

[3] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, & Saeid Nahavandi (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion, 76, 243–297.

[4] Geng, Y., Han, Z., Zhang, C., & Hu, Q.. (2022). Uncertainty-Aware Multi-View Representation Learning.

[5] Oh, S., Murphy, K., Pan, J., Roth, J., Schroff, F., & Gallagher, A.. (2018). Modeling Uncertainty with Hedged Instance Embedding.

[6] Jochen Gortler, Thilo Spinner, Dirk Streeb, Daniel Weiskopf, & Oliver Deussen (2020). Uncertainty-Aware Principal Component Analysis. IEEE Transactions on Visualization and Computer Graphics, 26(1), 822–831.

[7] Chari, T., Banerjee, J. & Pachter, L. The specious art of single-cell genomics. Preprint at bioRxiv, https://www.biorxiv.org/content/10.1101/2021.08.25.457696v3 (2021).

[8] LeCun, Y. & Cortes, C. (2010). MNIST handwritten digit database.

[9] B. Efron. "Bootstrap Methods: Another Look at the Jackknife." Ann. Statist. 7 (1) 1 - 26, January, 1979. https://doi.org/10.1214/aos/1176344552

[10] J. L. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez and J. Fernandez-Valdivia, "Diatom autofocusing in brightfield microscopy: a comparative study," Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, 2000, pp. 314-317 vol.3, doi: 10.1109/ICPR.2000.903548.

[11] Kolmogorov, A. (1933). Sulla determinazione empirica di una lgge di distribuzione. Inst. Ital. Attuari, Giorn., 4, 83-91.

[12] Freedman, D., Pisani, R., & Purves, R. (2007). Statistics (international student edition). Pisani, R. Purves, 4th Edn. WW Norton & Company, New York.