Mini Project 4: Reproducibility in ML

Alvin Chen
Department of Computer Science
McGill University
alvin.chen@mail.mcgill.ca

Ziqi Li
Department of Computer Science
McGill University
ziqi.li@mail.mcgill.ca

Emily Tam
Department of Computer Science
McGill University
emily.yw.tam@mail.mcgill.ca

Reproducibility Summary

Scope of Reproducibility

The paper "Visualizing Data using t-SNE" by Laurens van der Maaten and Geoffrey Hinton claims that a new technique called t-SNE produces significantly better maps compared to other non-parametric visualization techniques, namely Sammon mapping, isomap, and locally linear embedding (LLE) [1]. t-SNE allows for the visualization of high-diemsnional data by giving datapoints a location in a two- or three-dimensional map, which reveals both the local and global structure of data. The objective of this reproduction is to compare the visualization performance of the aforementioned techniques and extend the results of the paper by Maaten and Hinton.

Methodology

In this project, we compare the visualization performance of t-SNE, Sammon mapping, isomap and LLE on the MNIST data set as in the original paper. We used the scikit-learn package for implementing these four visualization techniques and we used the code by Shivani Chander for plotting the 2D maps [2].

Results

Our results demonstrate that t-SNE does outperform other non-parametric visualization techniques, as found in the original paper. In addition, we extended the results of the original project by exploring different parameter choices and by using the 2D representation of data to train KNN and decision tree models.

1 Abstract

The main claim of the paper "Visualizing Data using t-SNE" by Laurens van der Maaten and and Geoffrey Hinton is that t-SNE, a new technique for visualizing high-dimensional data, produces significantly better maps compared to other non-parametric visualization techniques. In this project, we reproduced the visualizations of the MNIST dataset from the paper by Maaten and Hinton using t-SNE, Sammon mapping, isomap, and LLE. We found that it was possible to reproduce the results from the original paper. We then extended the results of the original paper by exploring different parameter choices and using the 2D representation of data to train KNN and decision tree models.

2 Introduction

In this project, we reproduced the results of the paper "Visualizing Data using t-SNE" by Laurens van der Maaten and and Geoffrey Hinton [1]. The original paper was the first to use t-SNE, a technique to visualize high-dimensional data that performed significantly better than existing techniques in 2008. t-SNE is a variation of stochastic neighbour embedding (SNE) that is easier to optimize and produces better visualization, as it does not tend to crowd points together in the center. This technique gives each datapoint a location in a 2D (or 3D) map that reveals the structure of the data at the local and global scale. It is most useful for mapping high-dimensional data that lies on different, but related low-dimensional manifolds. For example, t-SNE is often used on datasets that contain images of objects that come from different classes that are seen from multiple viewpoints. The goal of the original paper was to reveal the advantages of t-SNE over other non-parametric visualization techniques, specifically Sammon mapping, isomap, and local linear embedding (LLE). As such, our goal in this project is to demonstrate that t-SNE does outperform these

33rd Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

other visualization techniques and extend the results of the original project by exploring different parameter choices and using the 2D representation of data to train KNN and decision tree models.

3 Scope of reproducibility

The original paper illustrated the performance of t-SNE on different datasets and compared t-SNE with other non-parametric visualization techniques. We decided to focus on reproducing the results of the section that compared t-SNE with other techniques the MNIST dataset. The main claim of the original paper is:

- Claim 1: t-SNE outperforms other non-parametric techniques, such as Sammon mapping, isomap, and LLE, in visualizing real-world datasets, as it retains both the local and global structure of data in one map.

4 Methodology

In this project, we used the scikit-learn package to implement the four visualization techniques mentioned in the original paper and we used code by Shivani Chander for plotting the 2D visualization maps [2]. We also used code by Schwartz *et al.* to display the distance matrix of the data points, and the similarity matrix with both constant and variable σ [3].

4.1 Model descriptions

As in the original paper, we compared the following visualization techniques: t-SNE, Sammon mapping, isomap and LLE. We used the models implemented by the scikit-learn package, each of which had many different hyperparameters. The original paper only mentioned the value of the cost function hyperparameters for these models. Therefore, we have only explicitly changed the cost function hyperparameters and kept the default values for all other hyperparameters.

4.2 Datasets

As in the original paper, we have used the MNIST dataset, which is a handwritten digits dataset that is widely used for different machine learning and pattern recognition models. It contains a total of 70,000 instances, from which 60,000 are for training and the remainder are for testing. The samples for each digit in MNIST are nearly evenly distributed and each image in this dataset is represented by 28×28 pixels such that each pixel has an integer from 0 to 255 representing the color intensity. For data pre-processing, we randomly selected 6000 data from the training dataset for computational efficiency. We vectorized the dataset to 2D: number of data $\times$ image pixels. Then, we performed dimensionality reduction on the dataset to reduce the number of dimensions to 30 as was done in the original paper.

We also used the UCI ML hand-written digits dataset, which can be downloaded from https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits. Similar to the MNIST dataset, this dataset contains images of handwritten digits. There are 10 classes, where each class is a different digit. There are 1797 instances and 64 attributes, as each image of an integer is 8×8.

4.3 Hyperparameters

The original paper only mentioned the value of the cost function hyperparameters for these models. Therefore, we have only explicitly changed the cost function hyperparameters while keeping the default value from scikit-learn for all other hyperparameters. The values for the cost function hyper-parameters in the original paper are as follows: t-SNE used $perp = 40$, Sammon mapping had no cost function parameters, isomap used $k = 12$, and LLE used $k = 12$.

4.4 Experimental setup and code

First, we acquired the MNIST dataset using the Tensorflow package and randomly selected 6000 data from the training dataset. Then, we vectorized the dataset to 2D and performed dimensionality reduction using PCA on the dataset. After this pre-processing of the dataset, we used the scikit-learn implementation for t-SNE, Sammon mapping, isomap, and LLE, and created scatter plots for each visualization technique. We evaluated the experiments by quantitative analysis, describing the difference in clustering structure of the points on each graph. Our code can be found in the code.zip file.

We extended the results of the original paper by adding the following experiments: (1) comparing t-SNE to UMAP, (2) tuning the hyperparameters of the t-SNE model to generate a better map, (3) trying t-SNE on another dataset, (4) displaying the distance matrix of the data points and the similarity matrix with both a constant and variable σ, and (5) using the 2D representation of data to train a decision tree and KNN model. We used the UMAP implementation from

the umap-learn module for the comparison of t-SNE and UMAP. For the hyperparameter tuning, we manually searched over the ranges suggested by scikit-learn for perplexity (between 0 and 100), learning rate (between 10 and 1000), and number of iterations (between 250 and 1250). We also tried both random initialization and PCA initialization for t-SNE. We then played around with perplexity using the UCI ML hand-written digits dataset instead of MNIST. We displayed the distance matrix of the data points and similarity matrix with both constant and variable σ using the code from Schwartz *et al.* [3]. Finally, we used the scikit-learn implementation for decision trees and plotted the decision boundaries from t-SNE.

4.5 Computational requirements

For this reproduction project, we are using the colab server for running all experiments. The t-SNE, isomap and LLE visualization techniques only took 1-5 minutes to run. However, the Sammon mapping took around 20 minutes to run. In total, it takes around 1 hour to run the entire notebook because we performed extensive parameter tuning and extended the work in the original paper.

5 Results

Note: Refer to Appendix for higher resolution figures.

5.1 Results reproducing original paper

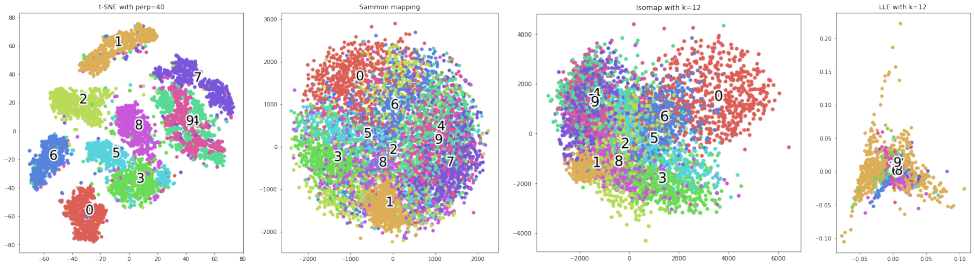


Figure 1: Visualizations of 6,000 handwritten digits from the MNIST dataset using t-SNE, Sammon mapping, isomap, and LLE from the ipynb file in code.zip.

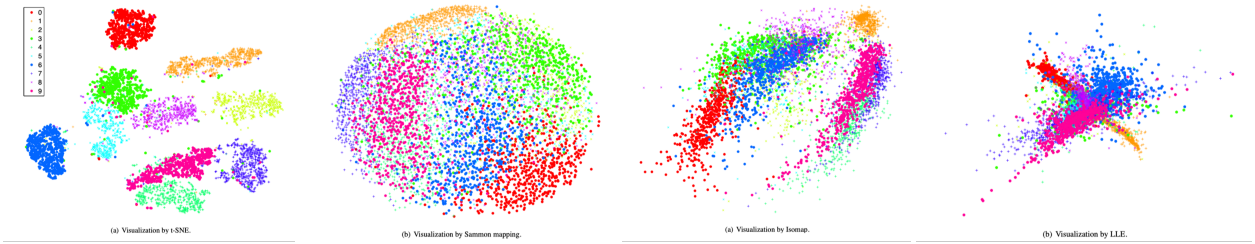


Figure 2: Visualizations of 6,000 handwritten digits from the MNIST dataset using t-SNE, Sammon mapping, isomap, and LLE from the original paper [1].

Our results supported the main claim of the original paper: t-SNE outperformed other non-parametric techniques, specifically Sammon mapping, isomap, and LLE, in visualizing the MNIST dataset, as it retained both the local and global structure of the data in one map. In Figure 1, we see that separation of the data using t-SNE is almost perfect. Only two digits remain unseparated: digit 4 and digit 9. In the original paper, digits 4 and 9 had greater separation but they were still very close to each other in the scatter plot. In Section 5.2, we will discuss how we tuned the model to obtain better separation between the digit classes. However, this naive implementation still demonstrated the significant advantage of t-SNE over Sammon mapping, isomap, and LLE. As in the original paper, we found that some points were clustered with the wrong class, but the majority of points within the same class are clustered together. With Sammon mapping, we see that the data points form a ball shape. Data points are not well-separated by class, with only digits 0 and 1 being distinguishable within the ball. This is similar to the original paper, which found that only three classes,

digits 0, 1, and 7, were somewhat separated. Isomap and LLE both failed to separate the classes. In the original paper, Maaten and Hinton also saw large overlaps between the digit classes.

The position of classes in our maps differs from the figures provided in the original paper. This is simply because we randomly chose data from the MNIST dataset. We also used a different random state for the t-SNE algorithm (it was not provided in the paper) and likely used different parameter values for any parameter that was not specified in the paper (only perplexity for the t-SNE model and k-value for the isomap and LLE models were provided).

5.2 Results beyond original paper

5.2.1 Additional Result 1

In our research on the topic of high-dimension data visualization, we also encountered a technique called Uniform Manifold Approximation and Projection (UMAP) [5]. It is touted as a technique just as powerful, if not more powerful, than t-SNE for high-dimension data visualization, with proponents claiming it maintains global positioning better than t-SNE [5]. We wanted to compare t-SNE to UMAP, since UMAP was developed over a decade after the t-SNE paper was published. We followed the example on the UMAP official documentation on the PCA-reduced MNIST dataset. In Figure 3, we see that UMAP produces a map with greater separation of clusters than t-SNE. The t-SNE map in

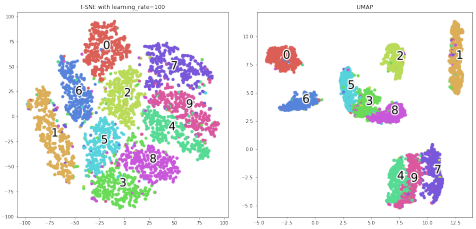


Figure 3: Visualizations of 6,000 handwritten digits from the MNIST dataset using t-SNE and UMAP.

Figure 3 was obtained after tuning the hyperparameters of the t-SNE model (refer to Additional Result 2). UMAP also clusters data points within the same class closer together compared to t-SNE. However, UMAP does not achieve the same degree of separation between digits 3, 5, and 8, or digits 4, 7, and 9 when compared to t-SNE.

5.2.2 Additional Result 2

Since the original paper did not mention whether they tuned the hyperparameters of the t-SNE model, we decided to investigate whether the hyperparameter values that they chose produced the best visualization or if a better map could be generated.

We started by exploring the effect of perplexity value on the resulting map. Perplexity is related to the number of nearest neighbours that is used in manifold learning algorithms [1]. Different values for perplexity can lead to significantly different results so we started by tuning this parameter. Typical values are between 5 and 50, but we decided to manually search over the range suggested by scikit-learn, which was between 0 and 100 [1]. We found that the best map was generated using $perp = 10$. When $perp < 5$, no separation was achieved between the data points. When $perp > 10$, we found that less separation was achieved between clusters of digits 4 and 9. Once $perp > 40$, we saw that more data points were clustered with the wrong class.

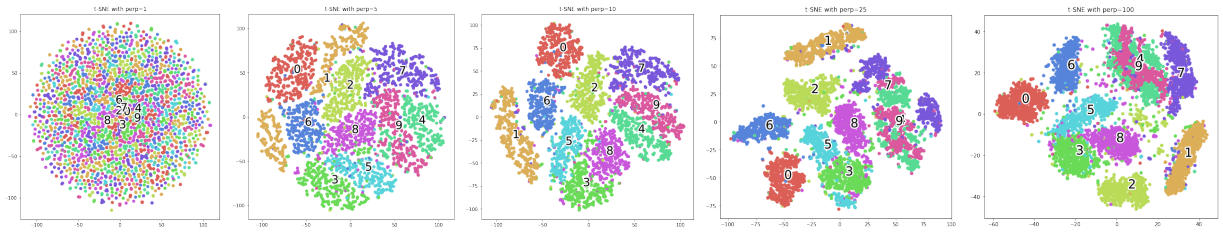


Figure 4: Visualizations of 6,000 handwritten digits from the MNIST dataset using t-SNE with $perp = 1, 5, 10, 25, 100$.

Next, we explored the effect of learning rate on visualization using t-SNE. We tried $learning\_rate = 10, 100, 250, 500, 750, 1000$ and $perp = 10$, as scikit-learn suggested searching over the range $[10, 1000]$. The best

map was produced with $learning\_rate = 100$ and $perp = 10$. However, there was not a significant difference between the map produced with different learning rates (refer to the ipynb file in code.zip for the maps).

Two other parameters that we investigated were the initialization of embedding and the maximum number of iterations for the optimization. There are two options for the initialization: random and PCA. We found that PCA failed to separate the data points by class (refer to Figure 6 in the Appendix). For the maximum number of iterations, we tried $250, 500, 750, 1000, 1250$. The scikit-learn implementation requires that a minimum of 250 iterations are performed and sets the default value of $n\_iter = 1000$. We found that $n\_iter = 250$ failed to separate the classes well but anything above $n\_iter = 500$ did not significantly effect the map (refer to the ipynb file in code.zip for the maps).

5.2.3 Additional Result 3

We decided to try t-SNE on another dataset, namely the UCI ML hand-written digits dataset. This dataset contains less instances and has lower dimensions so it did not take as long to compute the maps. From Additional Result 2, we saw that the hyperparameter with the biggest effect was perplexity. As such, we chose to tune perplexity on this dataset, and left the rest of the parameters with their default values. Visualization of the digits from the UCI dataset using t-SNE with $perp = 1, 5, 10, 25, 100$ showed greater separation of clusters compared to the MNIST dataset (refer to Figure 7 of the Appendix).

5.2.4 Additional Result 4

Based on the code from Schwartz *et al.*, we created a plot of the distance matrix for the data points, as well as the similarity matrix with both a constant and variable σ. From this, we can observe the 10 groups in the data, which correspond to the 10 different numbers in the dataset (refer to Figure 8 of the Appendix).

5.2.5 Additional Result 5

One possible extension of using t-SNE embedding could be directly using the 2D (or 3D) representation of the data to train and predict. The low dimensionality of the t-SNE results could easily be used in an intuitive model such as KNN and/or decision trees. Both of these models take advantage of the geometry of embedded data and could produce solid results. To test out this idea, we decided to use the t-SNE embedded MNIST data to train a KNN model and decision tree model. We used a 5-fold cross validation on the t-SNE embedded MNIST training set for both models. The KNN model used 5 neighbors, and the decision tree had a max depth of 10. The accuracy value obtained from the KNN model with 5-fold cross validation was 97.4% and the accuracy value obtained from the decision tree model with 5-fold cross validation was 97.0%.

Fold	1	2	3	4	5
KNN	0.976	0.974	0.974	0.971	0.974
DT	0.970	0.970	0.966	0.966	0.970

Table 2: Accuracies obtained using t-SNE embedded MNIST data to train a decision tree and KNN model.

Fold	1	2	3	4	5
KNN	0.976	0.974	0.974	0.974	0.973
DT	0.843	0.843	0.836	0.845	0.852

Table 2: Control accuracies obtained using just PCA-reduced MNIST data to train decision tree and KNN model.

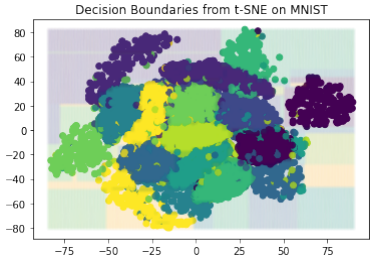


Figure 5: Boundaries generated by decision tree classifier on t-SNE embedded MNIST.

6 Discussion

Overall, our results supported the main claim of the paper. We were able to faithfully replicate the maps produced by t-SNE, Sammon mapping, isomap, and LLE, although the visualization by isomap differed slightly compared to the original paper. This is likely due to the use of different parameter values; the original paper only specified that the k-value was 12 and failed to mention the values of other parameters. The paper claimed that t-SNE outperformed Sammon mapping, isomap, and LLE, in visualizing the MNIST dataset, as it generates maps with distinct clusters based on class, and we were able to confirm this result. Figure 1 and Figure 2 show the results of our implementation using scikit-learn compared to the implementation by Maaten and Hinton and shows how close our results were to the original paper. Our approach had better runtime than mentioned in the paper due to advancements in research on t-SNE. The sklearn approach uses the Barnes-Hut approach rather than the "standard" method proposed in the paper to achieve a runtime of approximately $O(n\log n)$ compared to the standard $O(n^2)$ [1]. The original paper did not mention how the hyperparameters of the model were chosen and whether these hyperparameters were tuned to obtain the best map. Thus, we decided to perform additional experiments that tuned the following hyperparameters: perplexity, learning rate, and number of iterations. We also explored the effect of using a random initialization of embedding compared to a PCA initialization of embedding. We found that the best map was obtained with $perp = 10$ and $learning\_rate = 100$. As long as the number of iterations was greater than or equal to 500, there was no significant effect on the map. For the initialization of embedding, we found that PCA initialization did not work as well as random initialization.

In our extension of t-SNE, we found that the method uses Euclidean distance as a measure, and realized that the 2D embedded result could be used to classify. We therefore trained a KNN model and decision tree on the t-SNE embedded data and achieved an average accuracy of 0.97 for both models with a 5-fold cross validation. We found that the decision tree had a far better accuracy using the t-SNE embedded data compared to a tree trained on just PCA reduced data. Interestingly, the t-SNE trained KNN model performed roughly the same as the control KNN. As both of these models do not scale very well, running the entire MNIST set to train yielded worse results with roughly 0.90 accuracy for both. The accuracy of 0.97 is comparable to the accuracies achieved with a 1 and 2 layer MLP from MP3, and KNN and DT take less time to train and predict.

While researching high-dimension visualization techniques, we discovered UMAP, which can also be used for non-linear dimension reduction. We found that UMAP separated the classes better than t-SNE on the MNIST dataset. While the technique itself is relatively similar to t-SNE, we found UMAP to be significantly faster than t-SNE and in addition to yielding better results. When ran on the entire PCA-reduced MNIST set, t-SNE took approximately 15 minutes whereas UMAP took less than 2 minutes. The UMAP embedded data was used to train KNN and DT with the same hyperparameters and found approximately the same accuracy in a 5-fold cross validation, suggesting that in this aspect, UMAP has not significantly outperformed t-SNE. UMAP is a relatively new method, and future investigations could experiment with the faithfulness of UMAP versus other dimension reduction techniques.

6.1 What was easy

By using the scikit-learn package, we were able to implement models for t-SNE, Sammon mapping, isomap and LLE quite easily. The scikit-learn package also allowed us to perform a PCA dimensionality reduction on the dataset as was done in the original paper. Therefore, we did not have to implement the models by ourselves during the reproduction process. In order to understand the sklearn implementation and attempt ablation studies, we dug into the source code of the class. The code was relatively easy to understand thanks to its extensive commenting and our prior experience with gradient descent.

6.2 What was difficult

The math behind the model itself required a quick tutorial on what KL-divergence is, but that was the only challenge thanks to our robust prior experience with probability, gradient descent, and Gaussian and student-t distributions.

The most difficult portion of this was attempting an ablation study. While we believe ablation studies apply more to neural networks, we attempted to modify the sklearn package code by writing a class that inherits from sklearn's TSNE. Clearly any modification to its components, such as its probability method and kl-divergence calculator, results in a total failure of the system. The only component that could have been modified a little was the gradient descent, however the sklearn package for the gradient descent was already optimized very well with momentum, and any changes to the method resulted in far slower convergence.

As one may notice, our isomap looks very different from the author's isomap result. We found the paper's hyperparameters and techniques to be slightly vague for some of the other dimension reduction techniques, but we were still able to achieve similar results.

Statement of Contributions

Code: Ziqi, Alvin, Emily

Report: Alvin, Ziqi, Emily

References

1. van der Maaten, Laurens. (2013). Barnes-hut-sne. 1301.3342.
2. Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
3. Shivani, C. (2017). Visualisation of High Dimensional Data using tSNE – An Overview. https://github.com/shivanichander/tSNE?fbclid=IwAR1OLRKdyJuHYWCVcVOalviy0LYEGB2ifF5fXNXCg4D48AjmcEmZtCeCxqM
4. Schwartz, Z., Rossant, C., Chasen, F., & Odewahn, A. (2015). An illustrated introduction to the t-SNE algorithm. https://github.com/oreillymedia/t-SNE-tutorial
5. UCI Machine Learning Repository. Optical Recognition of Handwritten Digits Data Set. https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits
6. McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

Appendix

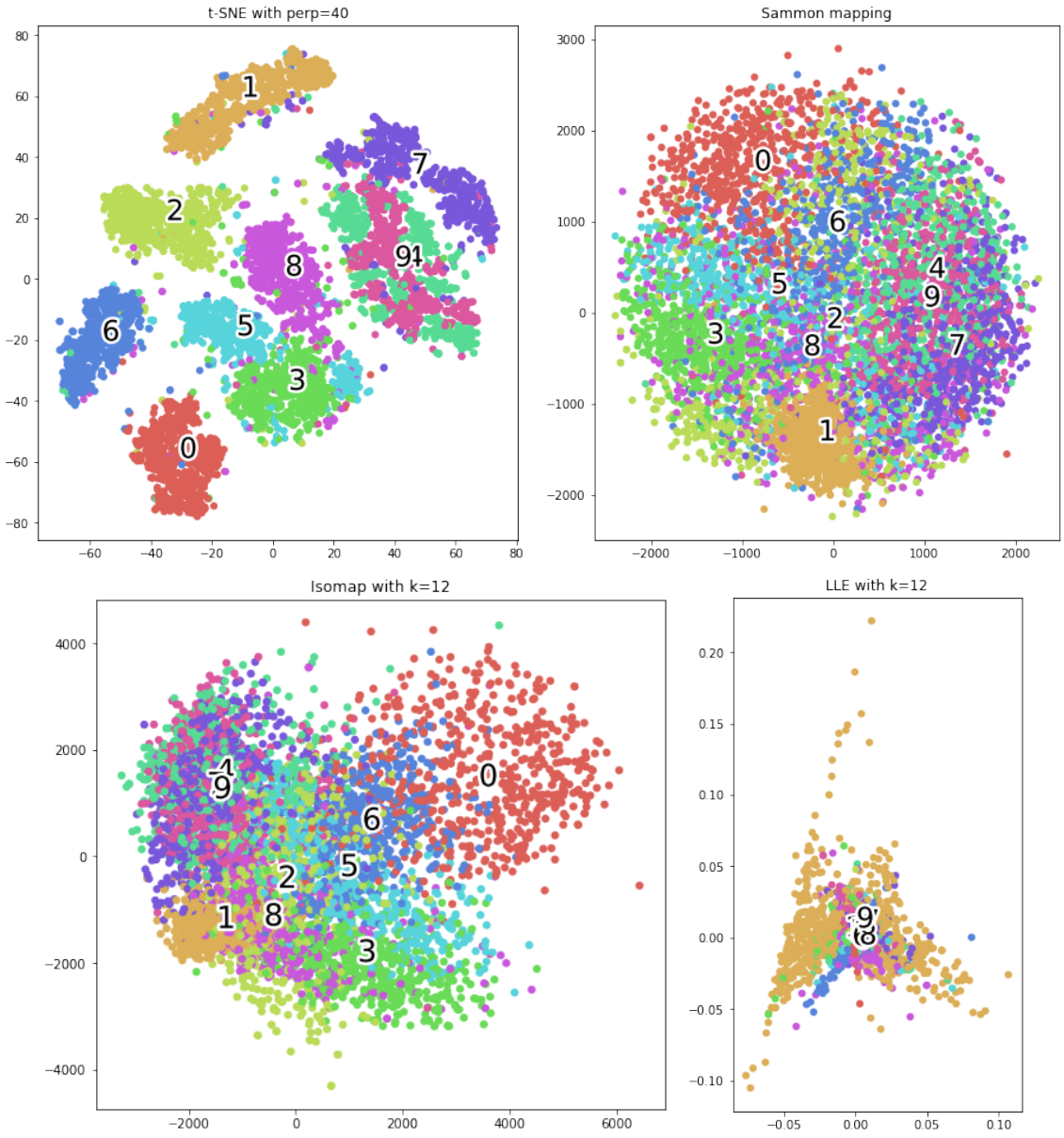
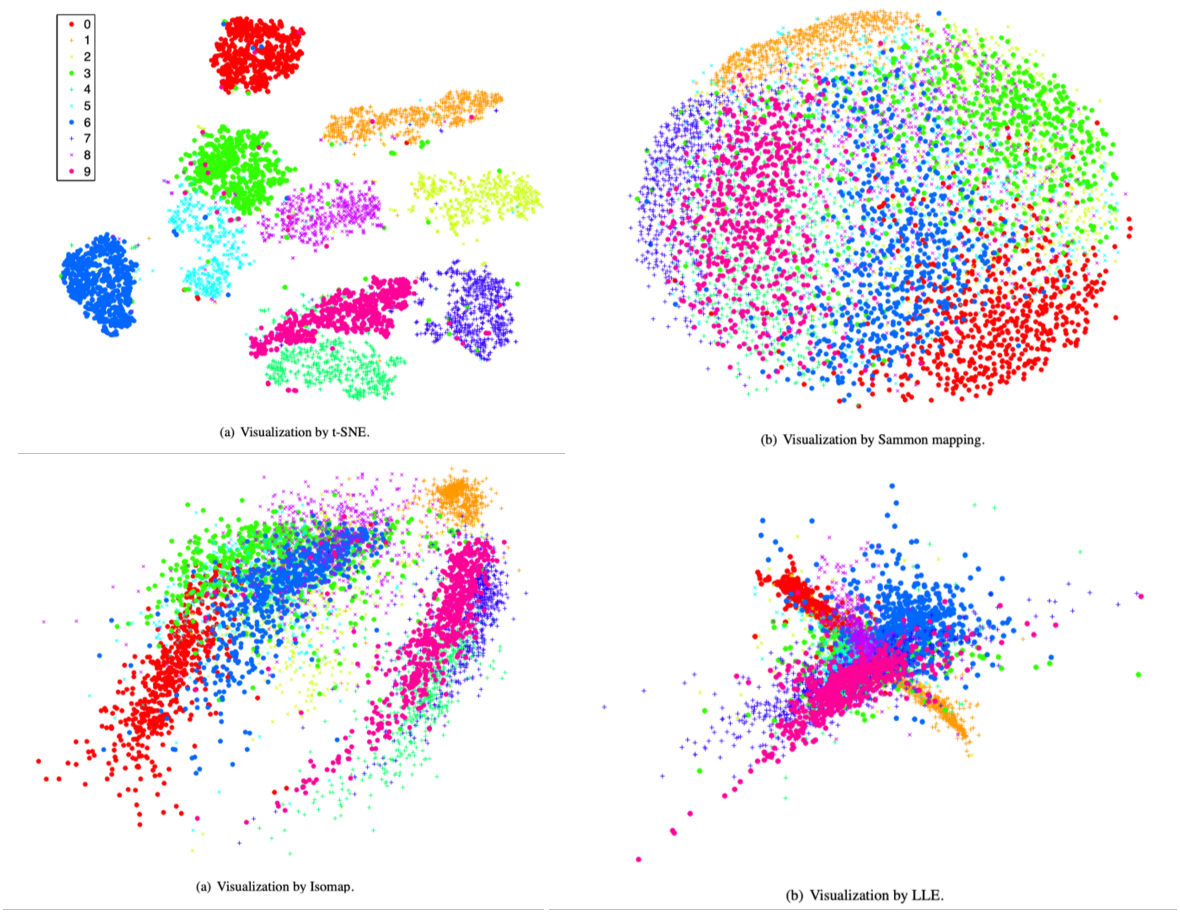


Figure 1: Visualizations of 6,000 handwritten digits from the MNIST dataset using t-SNE, Sammon mapping, isomap, and LLE from the ipynb file in code.zip.



(a) Visualization by t-SNE.

(b) Visualization by Sammon mapping.

(a) Visualization by Isomap.

(b) Visualization by LLE.

Figure 2: Visualizations of 6,000 handwritten digits from the MNIST dataset using t-SNE, Sammon mapping, isomap, and LLE from the original paper [1].

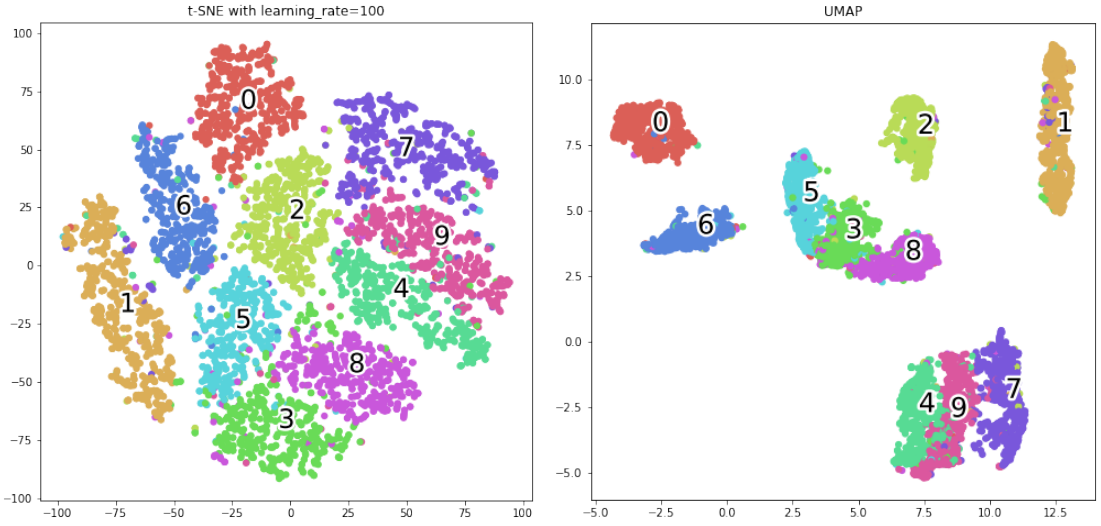


Figure 3: Visualizations of 6,000 handwritten digits from the MNIST dataset using t-SNE and UMAP.

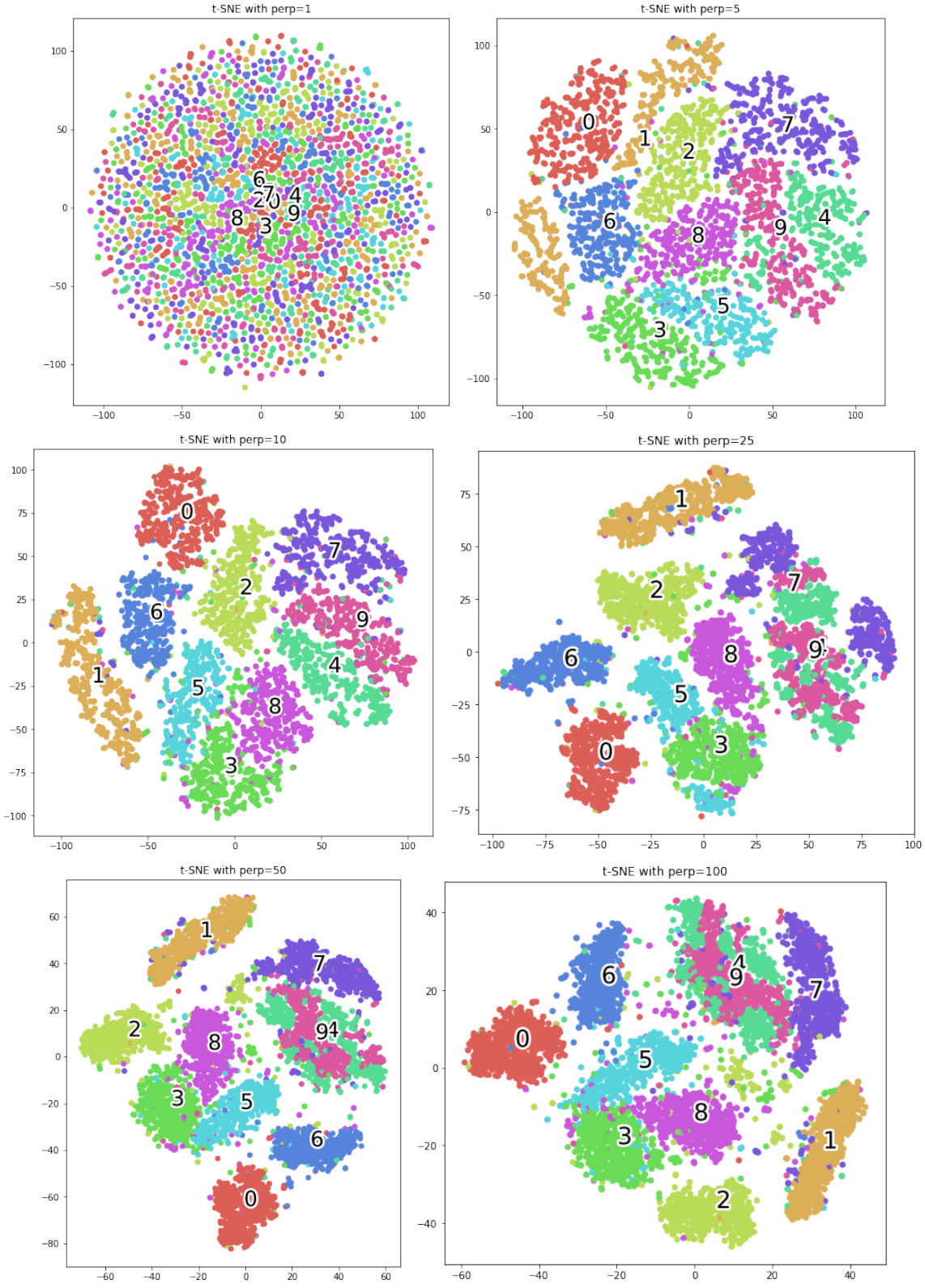


Figure 4: Visualizations of 6,000 handwritten digits from the MNIST dataset using t-SNE with $perp = 1, 5, 10, 25, 50, 100$.

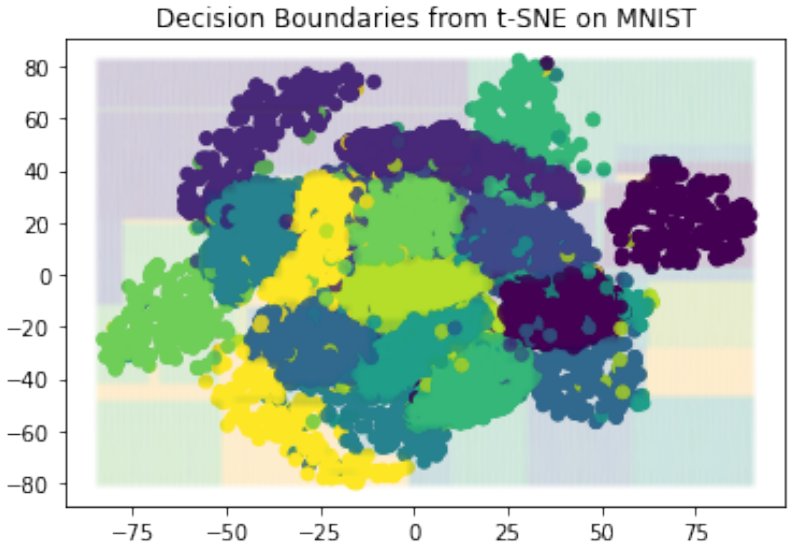


Figure 5: Decision Boundaries from t-SNE on MNIST.

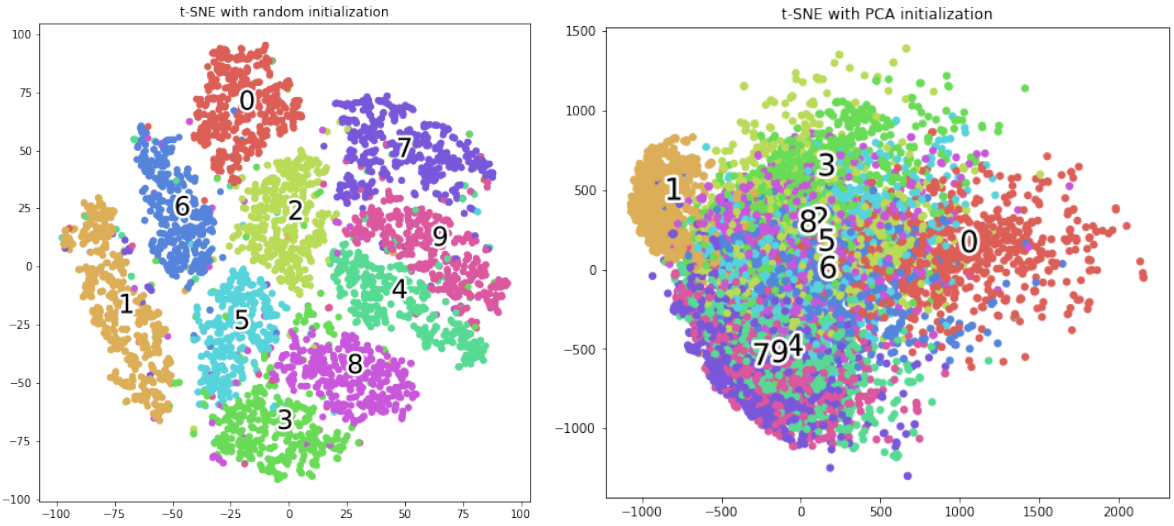


Figure 6: Visualizations of 6,000 handwritten digits from the MNIST dataset using t-SNE with random initialization and PCA initialization.

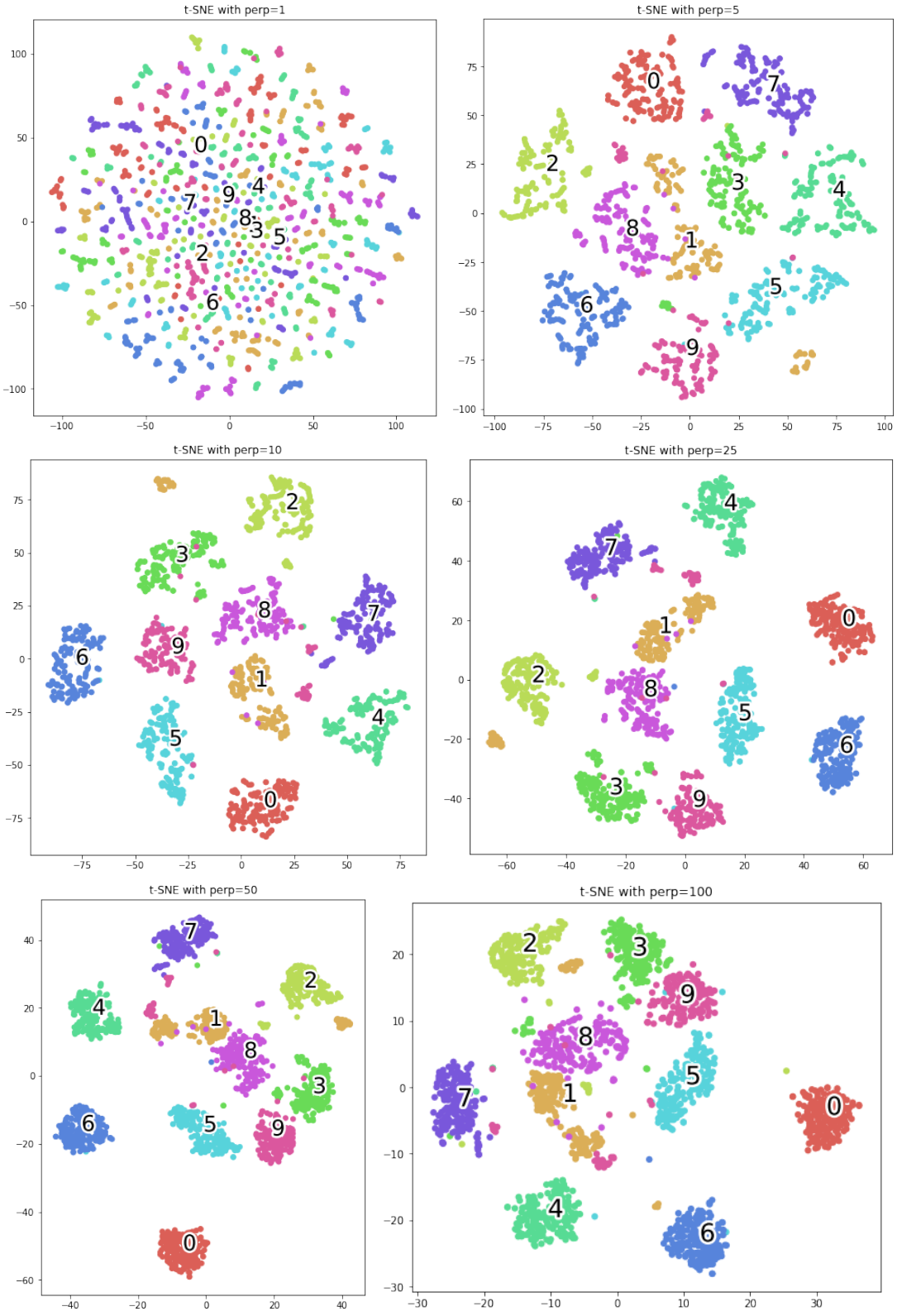


Figure 7: Visualizations of hand-written digits from the UCI dataset using t-SNE with $perp = 1, 5, 10, 25, 50, 100$.

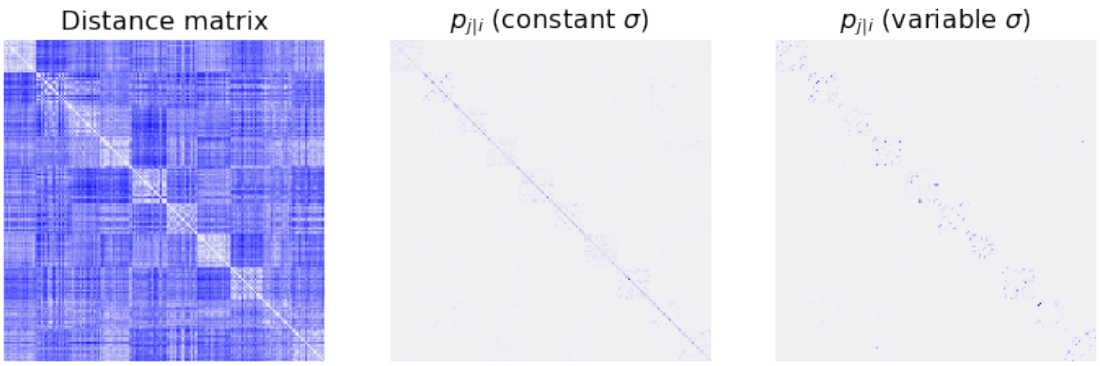


Figure 8: The distance matrix of the datapoints from the UCI dataset, as well as the similarity matrix with both a constant and variable σ.