CSIC5011 Final Project: Visualization Comparison on Fashion MNIST

DONG Hanze {hdongaj}, Li Donghao {dlibf}, WU Jiamin {jwubz}
Department of Mathematics, HKUST

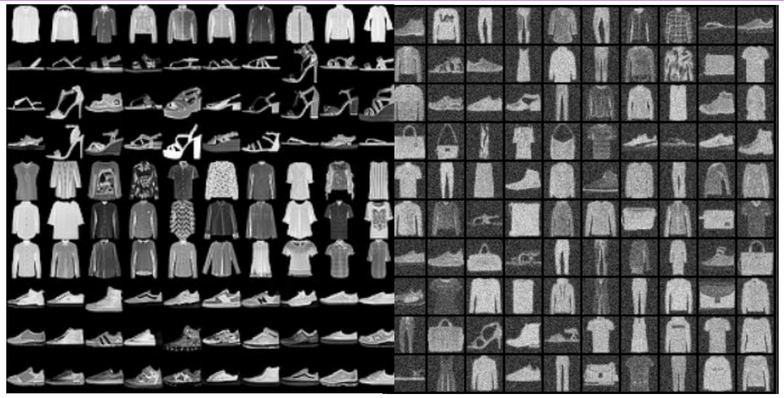
1. Introduction and Problem Define

High-dimensional datasets and representations are widely used in current machine learning tasks, and are proved practical in many downstream tasks. However, the visualization of high-dimensional vectors is extremely hard, which makes it not easy to understand the behavior of these distributions. On the other hand, high-dimensional vectors in real data often lie on a low-dimensional manifold, which provides us chances to visualize them and to get a low-dimensional illustration. In the project, we want to compare different dimension reduction methods including what we learned in class and recent state-of-the-art methods. We want to illustrate which is best for visualization image datasets in practice.

Problem define: Comparison among different dimension reduction techniques and visualize them on the Fashion MNIST dataset. We compare the results both on the visualization performance and the running time. Also, we have shown whether the algorithm needs careful parameter tunning and could handle noisy data.

2. Dataset

Fashion MNIST: The dataset we used is the fashion MNIST data consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. It contains figures about clothing and is thus harder for dimension reduction than the hand written MNIST dataset. We compare different methods on the dataset with (right) or without Gaussian noise with the standard variance as 0.1(left). In both cases, we can classify them by eyes. To speed up the process, we use 5000 images from the test set to conduct the experiments.



3. Methods

PCA: is a linear dimensionality reduction technique, which is based on the eigenvector of data distribution. PCA keeps the eigenvectors of top k eigenvalues.

MDS: is short for Multidimensional scaling. Similar to PCA, also use the eigen-decomposition and calculate the pair distances.

ISOMAP: is short for isometric mapping, which can be viewed as an extension of multi-dimensional scaling (MDS) or kernel PCA. It seeks a low-dimensional embedding, which maintains geodesic distance between points.

LLE: is short for Locally-Linear Embedding. It is a graph based algorithm first find K nearest neighbors, which can be changed in the training process. Then it estimate local properties and find a global embedding that preserves the local properties with the loss functior $E(W) = \sum_i |\mathbf{X}_i - \sum_j \mathbf{W}_{ij}\mathbf{X}_j|^2$

Hessian LLE: Different from the basic LLE method, it computes the embedding based on the Hessian. It cost much more than LLE so only be used on heavy work. (Crashed on our dataset)

Modified LLE: Compared with LLE, MLLE uses multiple weights, which are the local orthogonal projection of the original weights, in each neighborhood.

LTSA: is short for Local tangent space alignment. After finding the K nearest neighbors, it computes the tangent space at every point because when a manifold is correctly unfolded, all of the tangent hyperplanes to the manifold will become aligned. (Crashed on our dataset)

T-SNE: is short for t-distributed stochastic neighbor embedding. It regard the similarity between data as the joint probability and the objective minimize the Kullback-Leibler divergence between low-dimensional distribution and original one. However, the non-convexity of this problem makes the solution not unique and depend on the initialization.

UMAP: is short for Uniform Manifold Approximation and Projection. It requires the manifold to be locally connected. It assume the Riemannian metric is locally constant and the data on it is uniformly distributed.

DensMap: A density-preserving visualization tools based on t-SNE and UMAP. It computes estimates of the local density and uses those estimates as a regularizer in the optimization of the low dimensional representation.

VAE: is short for variational autoencoder, which is one special case for amortized variational inference. The parametrization is based on deep learning models, which has been proven useful in generation and latent factor inference. The dimension reduction here refers the latent factor code of data distribution, whose dimension is much lower than original one.

4. Visualization

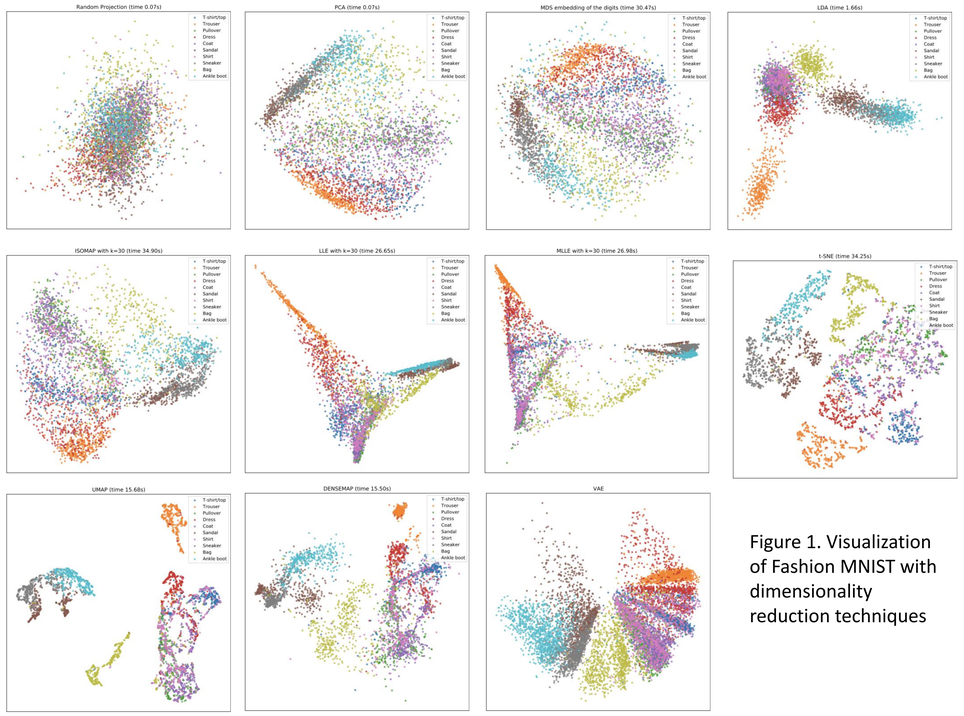


Figure 1. Visualization of Fashion MNIST with dimensionality reduction techniques

5. Analysis and Conclusion

	RP	PCA	LDA	MDS	ISOmap	LLE	MLLE	t-sne	UMAP	DENSMAP	VAE
Run time	0.06s	0.05s	~1.5s	~35s	~50s	~25s	~30s	~35s	14s	17s	~2h
parameter-tuning	No	No	No	No	k	k	k	ppl	k	k	Yes
linear svm (LS)	27.67	47.62	61.42	42.72	54.49	53.95	53.27	67.00	66.85	64	64.1
rbf svm (RS)	29.95	52.69	63.45	52.12	56.32	58.55	63.175	70.42	70.42	70.3	67.99
LS (noisy data)	24.74	47.77	59.92	36.75	52.96	57.17	59.8	66.90	64.79	62.45	
RS (noisy data)	26.7	53.02	62.00	48.9.0	56.65	61.17	64.67	70.87	68.52	69.37	
remark			supervised								large data

A summary of different dimension reduction methods. We show the running time of each algorithm and whether sensitive to parameter tuning. The visualization quality is shown by classification accuracy, and cross-validation accuracy means better visualization quality.

Analysis:
In general, we have compared different dimensionality reduction methods on the Fashion MNIST dataset to visualize the data distribution pattern on 2-d axis.
Manifold-based algorithms, such as T-SNE, UMAP provides the best visualization. Hessian LLE and LTSA crashed on the dataset.
For the graph-based methods, we need to tune k large enough to make sure the result is acceptable. When k=5, the performances are poor.
From the visualization, we can observe that there are clear 4 classes: coat and dress, trousers, bags, and shoes. Although there are 10 classes in total, it is hard for even person's eyes to classify the sub-classes in each of the 4 classes, such as different kinds of shoes. However, in terms of the speed, linear method such as LDA, PCA is superior to other methods. When labels are available, LDA is a practical and fast algorithm.

Conclusion:
If you have a new dataset, our suggestion is to try PCA first as a baseline since it is fast and does not need parameter-tuning. If a label is given, you can try LDA, which is fast and could help you check whether a linear project is good enough. Then just try t-SNE or UMAP and try to tune the parameters like perplexity for t-SNE or number of neighbors for UMAP. If your data is about images, you can try to use VAE since it is suitable for image data.

6. References

[1] van der Maaten, L.J.P.; Hinton, G.E. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9:2579-2605, 2008.
[2] van der Maaten, L.J.P. t-Distributed Stochastic Neighbor Embedding https://lvdmaaten.github.io/tsne/
[3] L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research 15(Oct):3221-3245, 2014.
[4] Xiao, Han, et al. "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms." ArXiv:1708.07747 [Cs, Stat], Sept. 2017. arXiv.org.
[5] Zhang, Zhenyue; Hongyuan Zha (2004). "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment". SIAM Journal on Scientific Computing. 26 (1): 313−338. CiteSeerX 10.1.1.211.9957
[6] S. T. Roweis and L. K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science Vol 290, 22 December 2000, 2323–2326.
[7] McInnes, L, Healy, J, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*, ArXiv e-prints 1802.03426, 2018
[8] Narayan, A., Berger, B., & Cho, H. (2020). Density-preserving data visualization unveils dynamic patterns of single-cell transcriptomic variability. bioRxiv.

7. Contribution

DONG Hanze: Presentation + Analysis + part of poster
LI Donghao: Experiment + Analysis + part of poster
WU Jiamin: Visualization + Analysis + part of poster