Rules of Thumb to Apply Dimension Reduction Techniques

(With hand-written digits dataset as an example)

Juncheng Li (SID: 20377124)

Introduction

If we want to reduce the dimensionality of the dataset, the problems are:

1. What does my dataset in the high dimension space look like? Does it cluster, according to the kind of its own? Is it folded, like a Swiss roll?
2. What outcome should I expect? Should it maintains the clustering effect? Or should it preserves the geodesic distance within the manifold?
3. What methods are available to achieve the desired outcome?

As dimensionality reduction methods increase, good choices of methods are more valuable. However, they largely depend on our priori knowledge about the methods and the dataset.

The main purpose of this poster is to provide some rules of thumb when deciding, what techniques to be applied. There are a great number of factors involved in making the decision, such as, purpose, size of the dataset, computation resources, cost, knowledge about the data and understanding of the method. By taking them into consideration, we are more likely to pick out the 'method' by a minimal amount of trails. As a result, under various circumstances, we can always pick out the method that is most effective at its cost.

Hand-written Digits Data

This dataset consists of 7291 handwritten numbers from 0 to 9 offered by Prof. Tibshirani. Each digit is stored by row and is reshaped into a 16 by 16 gray matrix, with the first tuple represent the correct type of the digit. Proportion of Each type of digit is balanced.

A noteworthy fact is that contributor commented the dataset is notoriously difficult keeping the misclassification rate low.

Methodology

1. Legendary methods for dimensional reduction
   - Principal Component Analysis
   - Multi-dimensional Scaling
2. Manifold Learning Methods
   - Local Linear Embedding
   - Stochastic Neighborhood Embedding

In the given dataset, the goal of dimension reduction technique is to tell different digits apart. Consequently, similar digits should cluster. But, how to tell whether two digits are similar? In other words, can we find a metric that achieve our goal?

One way is to calculate Euclidean distance between digits since similar digits tend to obtain lower values. However, distance between digits of different kinds may not be large enough to tell apart, making it difficult to find a threshold.

Another way is to choose the representative of the digits of each kind, and we can do it by nearest neighbor clustering, implying that Local Linear Embedding may be helpful.

But apart from clustering, we might want to preserve the cluster when the data is projected to lower dimension, including repulsing the different cluster apart, Stochastic Neighborhood Embedding may come into play.

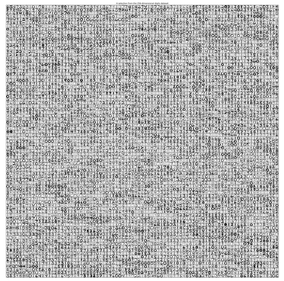
Therefore, it might be better to shift our focus to manifold learning methods rather than resort to methods that reduce dimension based on the Euclidean distance such as PCA and MDS.

How?

1. Scaling;
2. Extract principal components;
3. Plot;

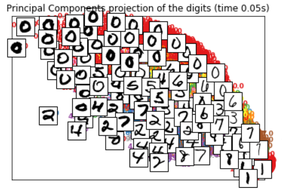
Results

Digits



I reshaped each row of the dataset into a 16*16 matrix and plot the greyscale value, so that we can have a glimpse over the distribution of digits and their shape.

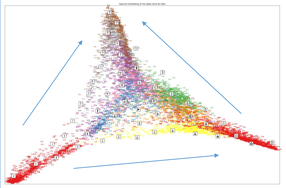
PCA



Euclidean Distances between non-zero pixel dots are inadequate to tell digits apart.

Spectrum Embedding

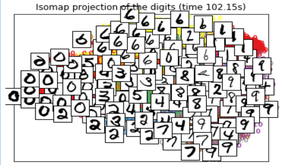
MDS fail to extract principal components due to the singularity error incurred during the computation procedure. Alternatively, we might consider apply spectrum embedding.



1. Bend upward, changing from 1 to 7;
2. Bend downward such that 1 and 0 can be told apart;
3. Straighten upward, from 0 to 7

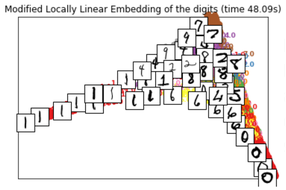
Results (Continued)

ISOMAP



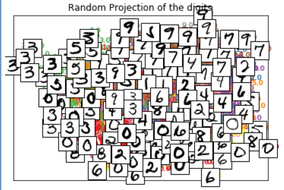
The Geodesic distance between digits may be a good indicator to tell them apart.

Modified LLE



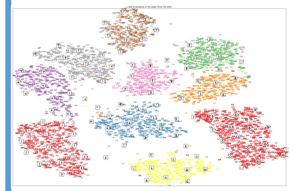
Interestingly, Patterns of MLLE and Spectrum embedding are similar.

Random Projection



Outcome of diffusion distance could not reproduce meaningful results.

T-SNE



Different groups of digits separate quite well because each digit clusters.

Conclusions

1. To classify, we might want to apply t-SNE.
2. Local Representation may be a good starting point when we want to preserve the clustering effect.
3. If we have little knowledge about the pattern of different categories of the data, we might begin with methods that are computationally cheaper than others, say, PCA,.
4. Visualization techniques provide a way to compare the performance of different learning methods, and may give more information than the key indices.

Takeaways

1. Ensure the dataset has been cleaned up.
2. Check the conditional number of the dependent variable matrix(i.e., the ratio of the largest singular value to the smallest one. When condition number is large (> 1,000), methods that put rigorous requirement on the stability of the inverse of the matrix should be avoided.
3. Proper scaling may be helpful in reducing the conditional number.

Future Work

1. Apply different dataset on the compressed dataset, and tune the parameters to improve the performance.
2. Seek explanations to the outperformance of Stochastic Linear Embedding.
3. Specify one of the factors that affect the method selection to further explore .

Reference

1. T. Hastie, Zip code digits datasets of "The Elements of Statistical Learning", https://web.stanford.edu/~hastie/ElemStatLearn/datasets/zip.digits/ , 2009;
2. Y. Yao, A Mathematical Introduction to Data Science, Chapter 6 – Chapter7, 2017;
3. J. Vanderplas, Comparison of Manifold Learning methods, https://nbviewer.jupyter.org/url/math.stanford.edu/~yuany/course/data/plot_compare_methods.ipynb, replicated at 13, October, 2017;