Mini-Project Report

Xiaotong Li[1], Yuchen Yuan[2]

[1]Department of Electrical Eningeering, [2]Department of Computer Engineering
University of California, Santa Cruz
1156 High St, Santa Cruz, CA 95064
Student ID:1634362, 1633572
E-mail: {xli239, yyuan31}@ucsc.edu.

Introduction

Mapping high-dimensional data into low-dimensional space through a certain method is a core problem of machine learning and data mining[1]. The common idea is to express some structures or features of high-dimensional space into low-dimensional space, for example, we will set data points that have similar characteristics together, while the unrelated data points set far apart . The normal dimensionality reduction methods are linear and non-linear. In this mini-project, we mainly try to use three different non-linear methods, t-SNE, LargeVis and TriMap. A non-linear dimensionality reduction method can effectively extract the global features in high-dimensional space. t-SNE was proposed by Maaten and Hinton[3]. LargeVis was proposed by using the algorithm of knn graph and the stochastic gradient descent for the training process [1]. TriMap is a dimensionality reduction method that uses triple embedding[2]. We try to use the 60,000 data points dataset Fashion-MNIST which has 10 different classes to experiment and measure the three dimensionality reduction methods using mean precision-recall and trustworthiness-continuity performance.

Dimensionality Reduction

In this section, we briefly introduce three dimensionality reduction methods and apply these three methods to the Fashion-MNIST dataset with 10 categories and 60,000 data points.

t-SNE

In low-dimensional space, we can define the distribution of data points:

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

In high-dimensional space, we can define it this way

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{2}$$

Then we use KL divergence to measure the similarity of this two distribution, and the gradient can be computed as

$$\frac{\partial C}{\partial y_i} = 4\Sigma_j(p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1} \tag{3}$$

This so-called t-SNE algorithm, which is added two improment based on SNE algorithm: Firstly, change original SNE into a symmetric SNE; Secondly, in a low-dimensional space using the t-distribution instead of the original Gaussian distribution, high Dimensional space unchanged.

LargeVis

For LargeVis algorithm, the first step is to use a random projection tree to get a space partition, on the basis of which we find the k nearest-neighbor of each point and get a rough kNN map, which is not required to be completely accurate. The second step is to use the neighbor search algorithm to find potential neighbors and calculate the distance between the neighbors

and the current point, then search the neighbors 'neighbors and the current point. Taking the k nodes with the shortest distance as k nearest neighbor, finally it will get an accurate kNN map. LargeVis also uses stochastic gradient descent for training, using negative and edge sampling optimizations. This technique is very efficient on sparse graphs because the two nodes connected by the edges of the different threads are rarely repeated.

TriMap

TriMap mainly use a tuple(i, j, k) to represent the relationship with point i and j. Different with t-SNE, TriMap use Gaussian similarity function in the high-dimensional space[2]

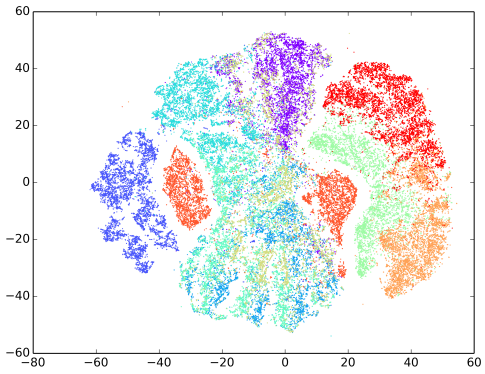
$$p_{ij} = \exp(-\frac{||x_i - x_j||^2}{\sigma_{ij}^2}) \tag{4}$$

in which $\sigma_{ij}^2 = \sigma_i \sigma_j$. For low dimension, it is represented as

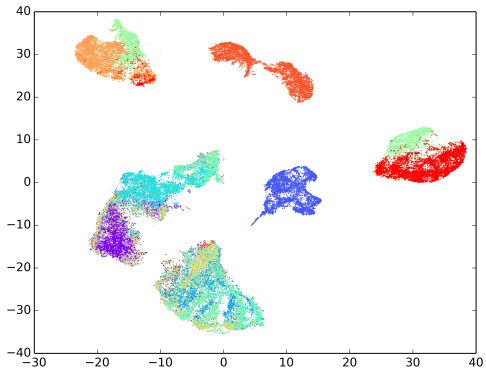
$$q_{ij} = (1 + \frac{||y_i - y_j||^2}{a})^{-\frac{1+\alpha}{2}} \tag{5}$$

Then it use two different ways, which is Nearest-neighbors triplets and random triplets to form the embedding.

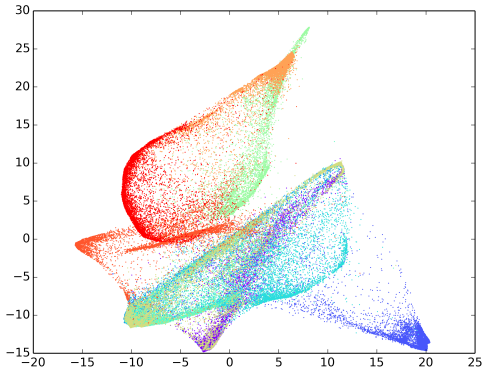
Figure.1 The three graphs below are the results after using three different dimension reduction methods on the Fashion-MNIST dataset[7]. Each example is a 28×28 grayscale image, associated with a label from 10 classes. The classification of figure (a) looks better and different types of data points were significantly spaced. Data points of figure (b) and (c) in some region will be concentred together. .



(a) t-SNE



(b) LargeVis



(c) TriMap

Figure 1: Dimensionality Reduction

Experiment and Quality Evaluate

In this section, we introduce and apply two tools to measure the effect of dimensionality reduction: mean precision-recall and trustworthiness-continuity. By analyzing these two types of curves, we can evaluate the effect of dimensionality reduction

Mean Precision-Recall

The paper gives the traditional definition of precision[4]:

$$\mathrm{Precision}(i) = \frac{N_{\mathrm{TP},i}}{k_i} \tag{6}$$

Which means in low-dimension space, the output point i has k_i neighbors. Generally, Number of TP (True Positive) means number of points that both in high-dimension space and in low-dimension space. However, the definition of recall is

$$\mathrm{Recall}(i) = \frac{N_{\mathrm{TP},i}}{r_i} \tag{7}$$

Here r_i means in high-dimension space, the number of neighbors of input point i. As we know, the value of N_{tp} would change according to the r_i and k_i. To get the precision-recall curve, we fixed r_i, and change the value of k_i to see how the precision and recall value fluctuate. Our test dataset is MNIST2500, which contains 2500 points and each point has 784 dimension data [**?**]. Using KNN tool from scikit-learn to get k nearest neighbors of input points and output points depends on given r and k, We divide our code into 3 parts:

(1) For each point i, we calculate the input neighbor with $r = 75$, then calculate the output neighbor with r from 1 to 75.

(2) For each value of r, we compare the same points in low-dimension neighbor and high-dimension, got the value of TP then store.

(3) After got 75 precision and recall values, we calculate the mean value of them and plot the curve.

Trustworthiness-Continuity

It's kindly same to previous paper, firstly, it gives the definition of two terms. Trustworthiness[5][6],

$$T(k) = 1 - A(k) \sum_{i=1}^{N} \sum_{x_j \in U_k(x_i)} (r(x_i, x_j) - k) \tag{8}$$

let N be the number of data at high-dimension space and $r(x_i, x_j)$ be the rank of the data sample x_j in the ordering according to distance from x_i in the original data space. Denote by $U_k(x_i)$ the set of those data samples that are in neighborhood of sample x_i in the low-dimension data space but not in the high-dimension data space. $A(k) = 2/(N \times k(2N - 3k - 1))$ scales the values between zero and one. Similarly, the definition of Continuity is[5][6]

$$C(k) = 1 - A(k) \sum_{i=1}^{N} \sum_{x_j \in V_k(x_i)} (\hat{r}(x_i, x_j) - k) \tag{9}$$

$r(x_i, x_j)$ be the rank of the data sample x_j in the ordering according to distance from x_i in the low-dimension space. $V_k(x_i)$ be the set of those data samples that are in the neighborhood of the data sample x_i in the high-dimension space but not in the low-dimension data space. For the same test dataset MNIST2500 [**?**]. We still use kNN tool to find nearest neighbor. The difference between previous one is we generate a rank-matrix in both high-dimension space and low-dimension space both. We divide our code into 5 parts:

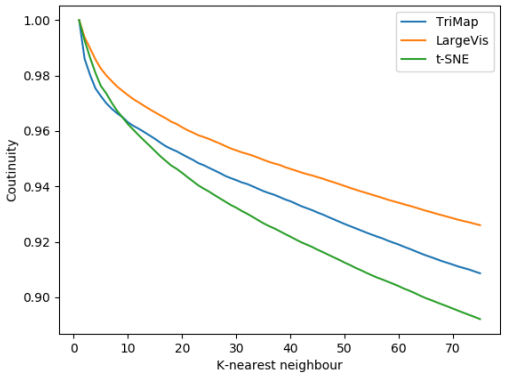
(1) According to the high-dimension dataset size, we create the high-rank matrix of (2500×2500) to store the rank between each pair of points. Same to low-dimension dataset, and low-rank matrix.

(2) For each point i, we calculate the input neighbor and output neighbor with $r = k$ from 1 to 75. Because the Trustworthiness and Continuity needs the same k value

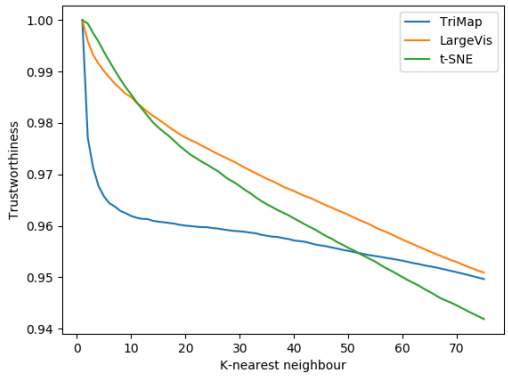
(3) For each value of k we compare the points in low-dimension neighbor and high-dimension neighbor, got the U_k and V_k.

(4) For every point in U_k , we got the rank value from high-dimension rank matrix, same to every point in V_k. Then store these rank value.

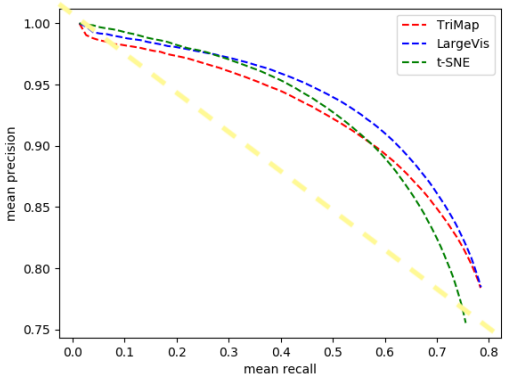
(5) After executing $k = 75$, sum all the rank value and put them in the formula, got the final value of Trustworthiness and Continuity.



(a) Continuity



(b) Trustworthiness



(c) Mean Precision-Recall

Figure 2: Quality Measurement

Figure.2

1. From the figure (a) we can see that the three methods are relatively smooth curve, when

changing the k value, the recall value increases, indicating the predicted point contains more and more correct points, at the same time it contains more other errors, so the precision value reduces. largeVis is closer to (1, 1) points than the other two methods, indicating that the quality is better.

2. For Trustworthiness, which is figure (b) we find k neareast-neighbors in both high and low dimensions. The TriMap indicated by the blue line decreases very rapidly when the number of neighbors increases in low dimension. When the number of nearest-neighbours is also increasing, However, LargeVis and t-SNE did not change very much due to the growth of k, and finally three methods have similar values.

3. In Continuity figure (c), these three methonds changes similar with the increase of k nearest-neighbour, but TriMap and LargeVis are little better compared with the performance.

Conclusion

In this project, we learned and understood three different non-linear dimensionality reduction methods, t-SNE, LargeVis, and TriMap. LargeVis has great advantages when it comes to running large datasets, while TriMap and t-SNEs take a long time to run on computer. At the same time, we measure and analyze the quality of these three methods by implementing mean Precision-Recall and Trustworthiness-Continuity. In trustworthiness, TriMap decreases significantly with he number of k nearest-neighbors increase, and the performance of the other two methods is slightly stable. In the mean-Precision-Recall and Continuity, these three methods performance are similar.

References

[1] Tang, Jian, et al. Visualizing large-scale and high-dimensional data. *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016.

[2] Amid, Ehsan, and Manfred K. Warmuth. *Transformation invariant and outlier revealing dimensionality reduction using triplet embedding.* (2018).

[3] Maaten, Laurens van der, and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research* 9.Nov (2008): 2579-2605.

[4] Venna, Jarkko, et al. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research* 11.Feb (2010): 451-490.

[5] Kaski, Samuel, et al. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC bioinformatics* 4.1 (2003): 48.

[6] Venna, Jarkko, and Samuel Kaski. Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. *Proceedings of WSOM.* Vol. 5. 2005.

[7] Xiao, Han, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint* arXiv:1708.07747 (2017).

[8] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits, 1998.