A distance function for ordinal regression dimensional reduction (CSIC 5011 Mini project)

ZHONG Guangzheng
Hong Kong Baptist University
22482032@life.hkbu.edu.hk

Abstract

Ordinal regression (OR) aims to solve multi-class classification problems with ordinal classes. However, most of dimensional reduction methods are put forward for unsupervised learning problem, and ignore the class information. In this project, we proposed a distance function for ordinal regression dimensional reduction problem. The class information in datasets is considered and the resulting representation in low dimensional space is clear and easily to separate. Moreover, we also consider the ordinal information in classes. We introduce the distance function in 3 dimensional reduction methods and perform experiments on the Hand-written Digits dataset. The code can be found in : https://github.com/zhong-gz/CSIC-5011-Miniproject.

I. Introduction

Ordinal regression (OR) is proposed to solve multi-class classification problem where the classes are ordinal [1]. In this project, we consider the Hand-written Digits dataset. In this dataset, the classes is number $[0, 1, \cdots, 9]$. In this example, the number 9 is larger than the other numbers. Therefore, after mapping data into low dimensional space, number 9 should be far from number 0 and close to number 8. In OR dataset, the classes contain a natural order. OR is different from the traditional multi-class classification. In multi-class classification, the classes are unordered. In OR, the classes are ordinal. OR has been widely used in various applications, such as social science, face recognition, medicine research, credit rating and so on.

Dimensional reduction are proposed for high dimensional data. One of the problems in high dimensional datasets is that, not all features is important for particular task. Although some of the methods with high computational complexity shows good performance in prediction or classification problems [2], there is still meaningful to reduce the dimension of data, which may obtain models with better explain-ability.

There are two main applications of dimensional reduction. The first is feature extraction [3]. Due to the increasing dimension of data, an effective technique is required for all data mining and machine learning tasks. To improve the performance of machine learning methods and reduce the training time, dimensional reduction is considered as a pre-processing. The second is visualization [4]. Good visualizations help to get better output. It also explains the distribution of data, and the working principle of machine learning methods.

Most of existing dimensional reduction methods is proposed for unsupervised learning, such as PCA [5], MDS [6], isomap [7] and SNE [8]. However, these methods ignore the class information in dataset, as a result, the resulting data in low dimensional space could be mixed and difficult to separate. A typical supervised dimensional reduction methods is LDA [9], [10], which maps k classes data into $(k-1)$-dimensional space. In lower dimensional space, the intra-class variance is minimized and between-class variance is maximized. However, the dimension of resulting space is $(k-1)$, which may still contain insignificant information. Moreover, it could be difficult to visualize data if the classes are more than 4.

In this project, we introduce a new distance function to 3 unsupervised dimensional reduction methods, including MDS, isomap and t-SNE [11]. In our distance function, we consider not only the difference of classes, but also the distance of classes. As a result, our resulting low dimensional data shows a clear separation and order among classes.

This project report is organized as follows. In Section II, we proposed our distance function. We introduce our distance function into 3 unsupervised dimensional reduction methods, and conduct experiments in section III. The conclusion and future work are presented in section IV.

II. Distance Function

In this section, we proposed our distance function for ordinal regression dimensional reduction. Different from the traditional distance function, our distance function consider the ordinal class information in datasets. Based on this idea, we proposed the distance function $\hat{d}_{ij}$ as follows.

$$\hat{d}_{ij}^2 = (|c_i - c_j| + 1) \|x_i - x_j\|^2 \tag{1}$$

where the c_i is the class of the i^{th} data, x_i is the i^{th} data and $\|x_i - x_j\|^2$ is the euclidean distance between the i^{th} data and j^{th} data.

The value of $(|c_i - c_j| + 1)$ will be 1 if i^{th} data and j^{th} data belongs to the same class. And the value of $(|c_i - c_j| + 1)$ increases when the distance between classes increases. For example, the distance of data in the same class would be $\|x_i - x_j\|^2$. And the distance of data in class 0 and 9 will be $10 \cdot \|x_i - x_j\|^2$. Therefore, data with short class distance is close to each other. For data with long class distance, the data will be far from each other.

Our distance function (1) can be applied into MDS, isomap and SNE-based methods directly. The distance function in MDS is $\|x_i - x_j\|^2$, which can be replaced with our function. The closest distance in isomap is also defined by euclidean distance. Moreover, the distribution (probability) in SNE-based methods is constructed by euclidean distance. Therefore, our distance function can be applied to replace the euclidean distance in above methods and other methods which apply euclidean distance.

III. Experiment Results

In this section, we introduce our distance function to MDS, isomap and t-SNE methods. Experiments on Hand-Written Digits dataset is conducted.

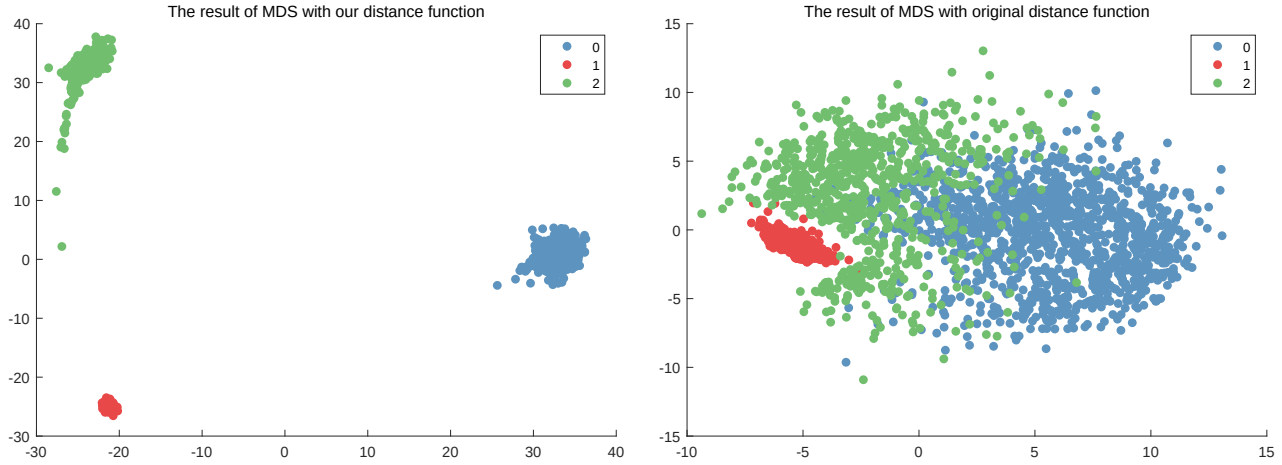
A. MDS

We first compare the output low dimensional data in Nonmetric MDS method. The original optimization problem of Nonmetric MDS is shown as follows.

$$\begin{aligned} &\min_{Y_i \in \mathbb{R}^k} \sum_{i,j} \left(\|Y_i - Y_j\|^2 - \tilde{d}_{ij}^2 \right)^2, \\ &\text{s.t.} \sum_{i}^{N} Y_i = 0. \end{aligned} \tag{2}$$

where the distance function $\tilde{d}_{ij}^2$ is replaced with our distance function $\hat{d}_{ij}^2$ in the following experiments. We compare our distance function with euclidean distance.

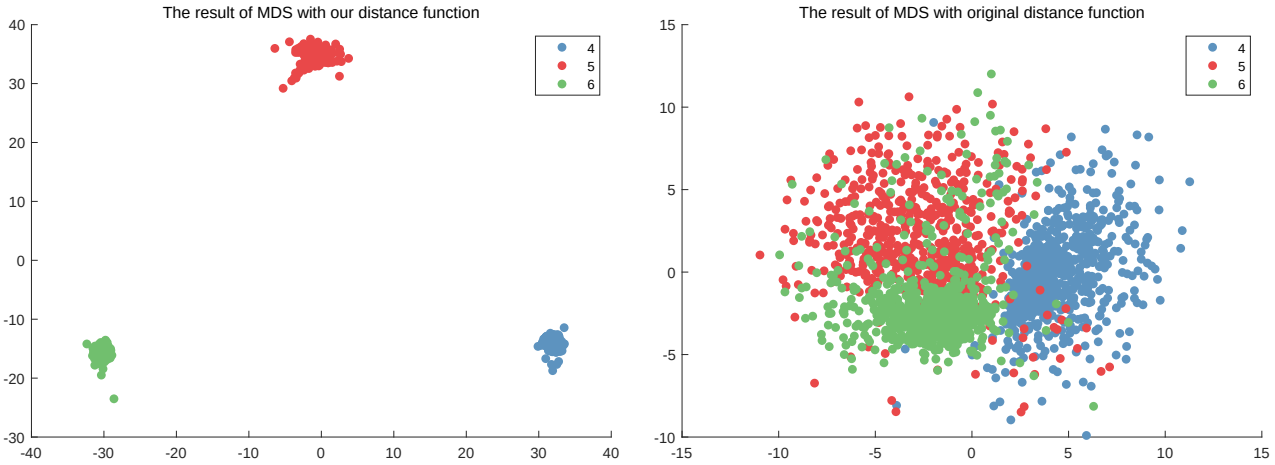
We consider 3, 6, 10 classes of data in dataset and perform MDS methods with our distance function. The 2-dimension data are plotted in figure 1, 2, 3 and 4.



(a) MDS result with our distance function.

(b) MDS result with euclidean distance function.

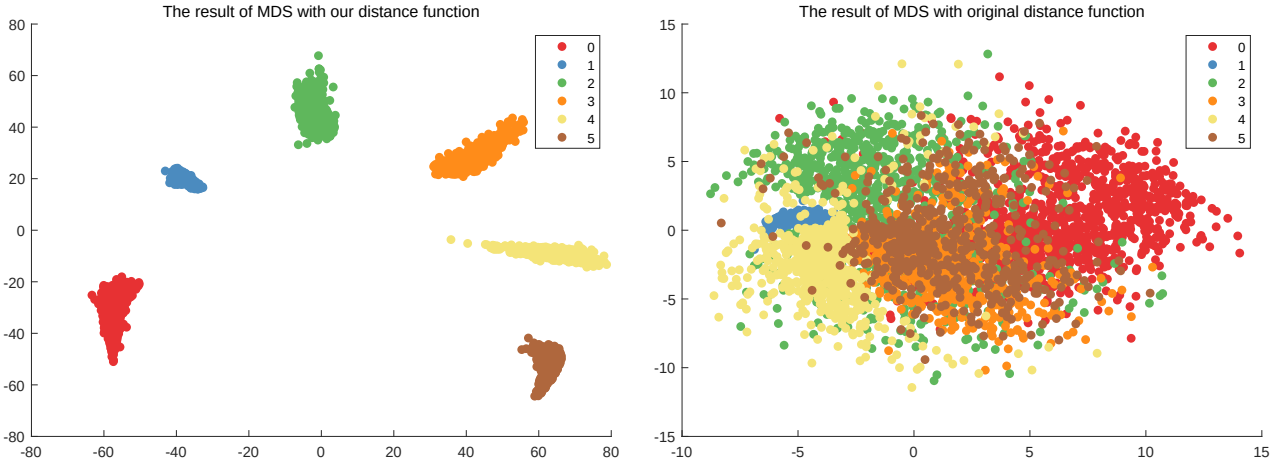
Fig. 1: MDS results with 3 classes data (numbers 0, 1, 2) in Hand-written Digits dataset.



(a) MDS result with our distance function.

(b) MDS result with euclidean distance function.

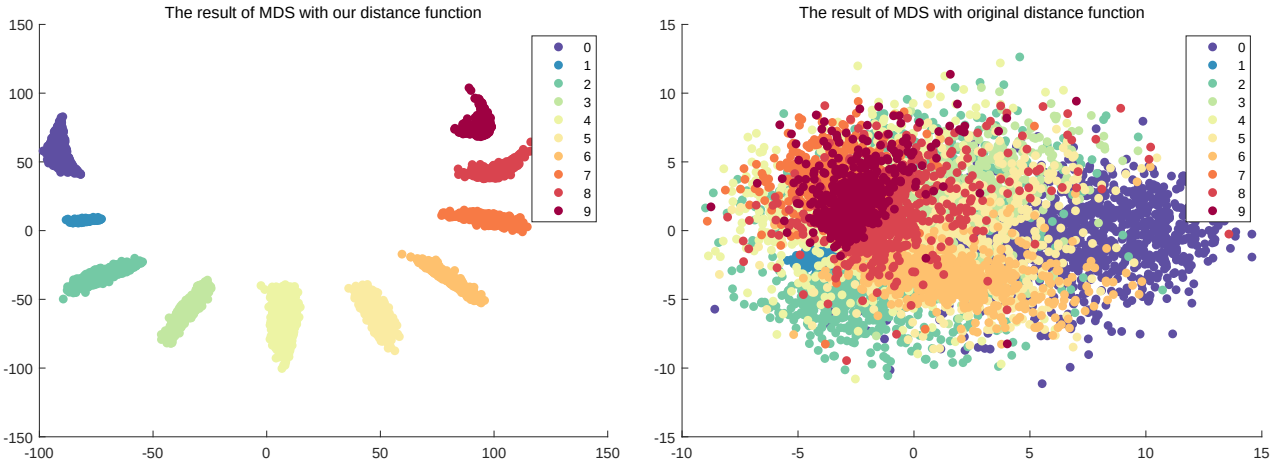
Fig. 2: MDS results with 3 classes data (numbers 4, 5, 6) in Hand-written Digits dataset.



(a) MDS result with our distance function.

(b) MDS result with euclidean distance function.

Fig. 3: MDS results with 6 classes data in Hand-written Digits dataset.



(a) MDS result with our distance function.

(b) MDS result with euclidean distance function.

Fig. 4: MDS results with all classes data in Hand-written Digits dataset.

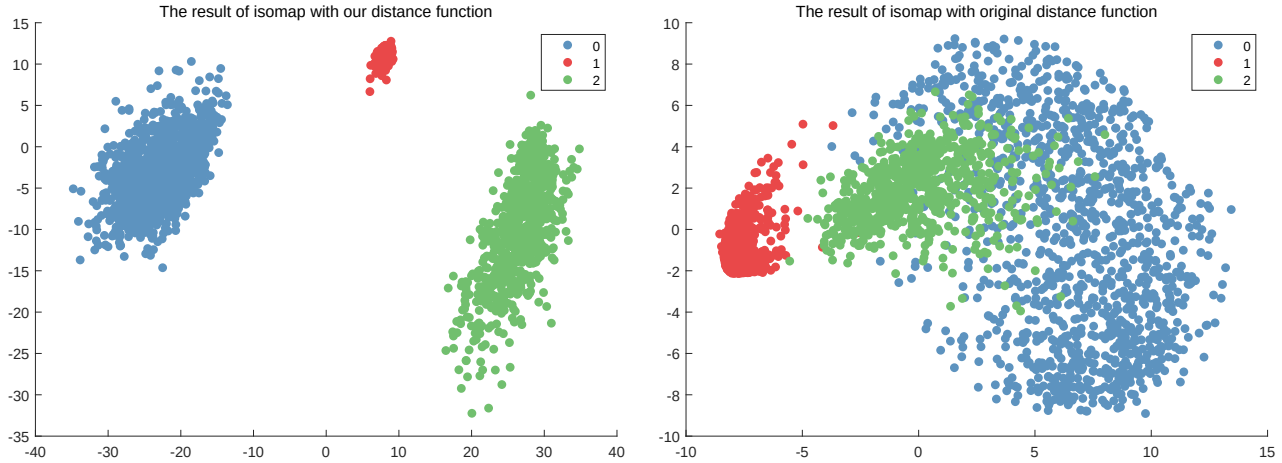
In fig. 1 and 2, we can see that 3 digital numbers are far from each other with our distance function. However, the order of classes is not clear. For example, the gap between numbers 0 and 2 should be larger than that of numbers 0 and 1 or numbers 1 and 2.

In fig. 3 and 4, different classes can be easily separated by applying our distance function. Moreover, the order of classes can also be persevered. On the contract, the 2-dimension embedding data obtained by MDS methods with euclidean distance is mixed together, which is difficult to identify the classes.

In these figures, data from different classes distributes in a circle shape. The main reason maybe the constraints $\sum_i^N Y_i = 0$. This constraints ensure the center of data be 0. As a result, all data will circle around the 0 point when they are enforced to be far from data in other classes.

B. Isomap

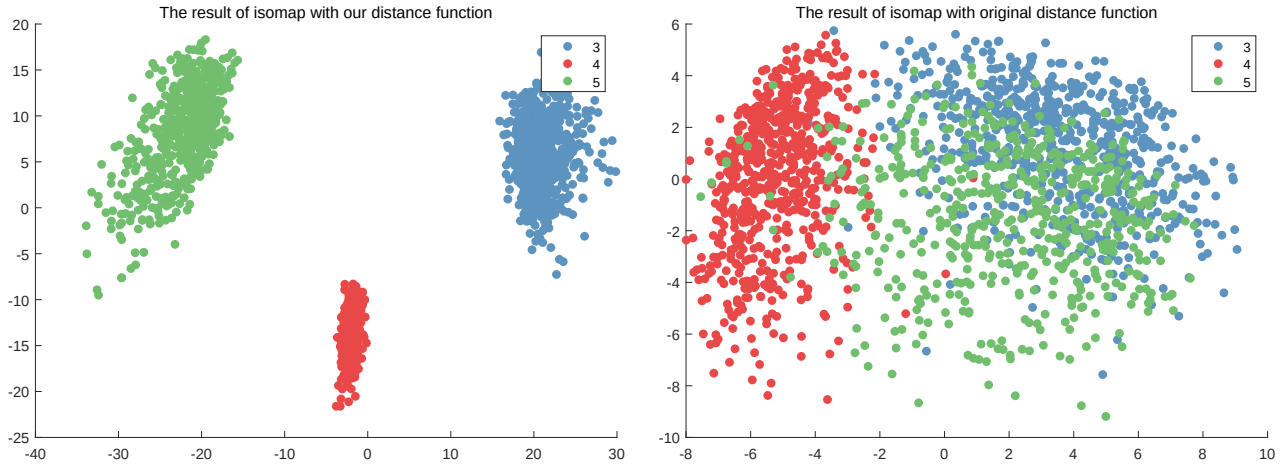
We then introduce our distance function into isomap method. Our distance function is applied for constructing the n-by-n neighborhood graph in the first step of isomap method. In the next step, the shortest path distance between nodes in different classes is relatively large and that of nodes in the same class is small. The result of two distance function in isomap methods is shown in fig. 5, 6, 7 and 8.



(a) isomap result with our distance function.

(b) isomap result with euclidean distance function.

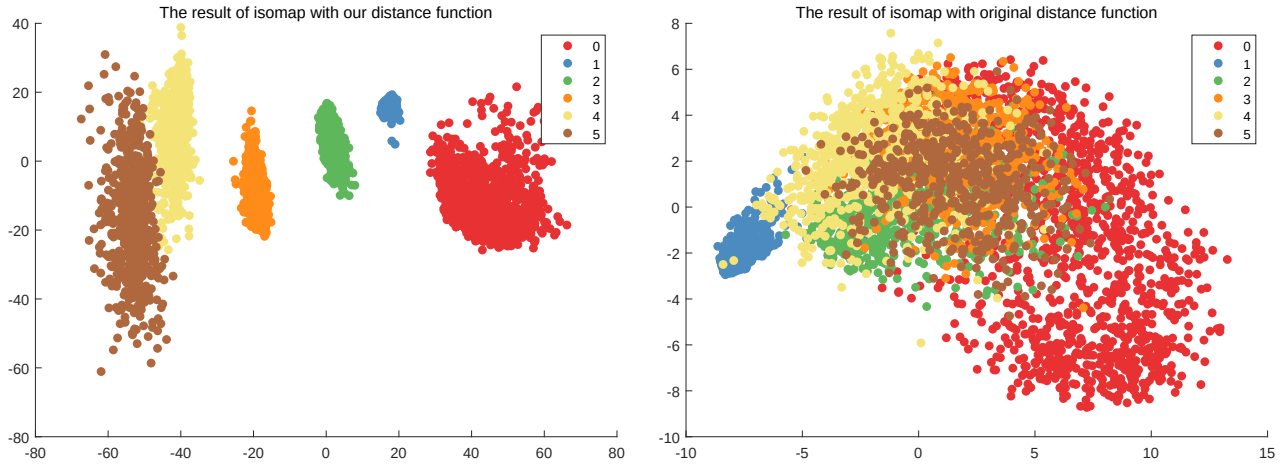
Fig. 5: isomap results with 3 classes data (numbers 0, 1, 2) in Hand-written Digits dataset.



(a) isomap result with our distance function.

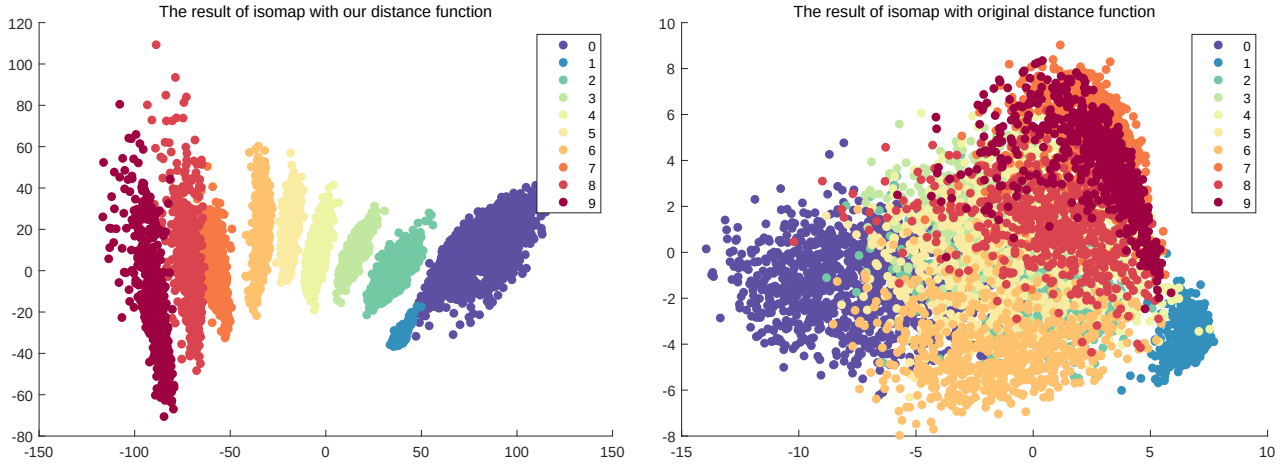
(b) isomap result with euclidean distance function.

Fig. 6: isomap results with 3 classes data (numbers 3, 4, 5) in Hand-written Digits dataset.



(a) isomap result with our distance function. (b) isomap result with euclidean distance function.

Fig. 7: isomap results with 6 classes data in Hand-written Digits dataset.



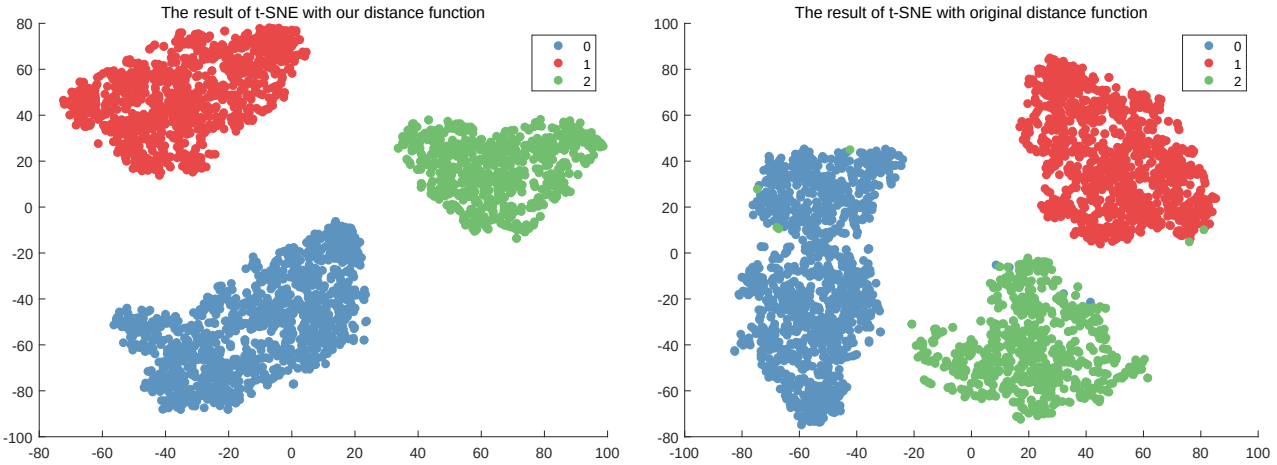
(a) isomap result with our distance function. (b) isomap result with euclidean distance function.

Fig. 8: isomap results with all classes data in Hand-written Digits dataset.

In fig. 5, 6, 7 and 8, the 2-dimension data with our distance function is separable. Moreover, the ordinal information of classes is also preserved. It is worth to notice that, in fig. 5 and 6, the ordinal information of 3 classes is revealed, which shows better visualization than MDS with our distance function. On the contract, the results obtained with euclidean distance is still mixed together.

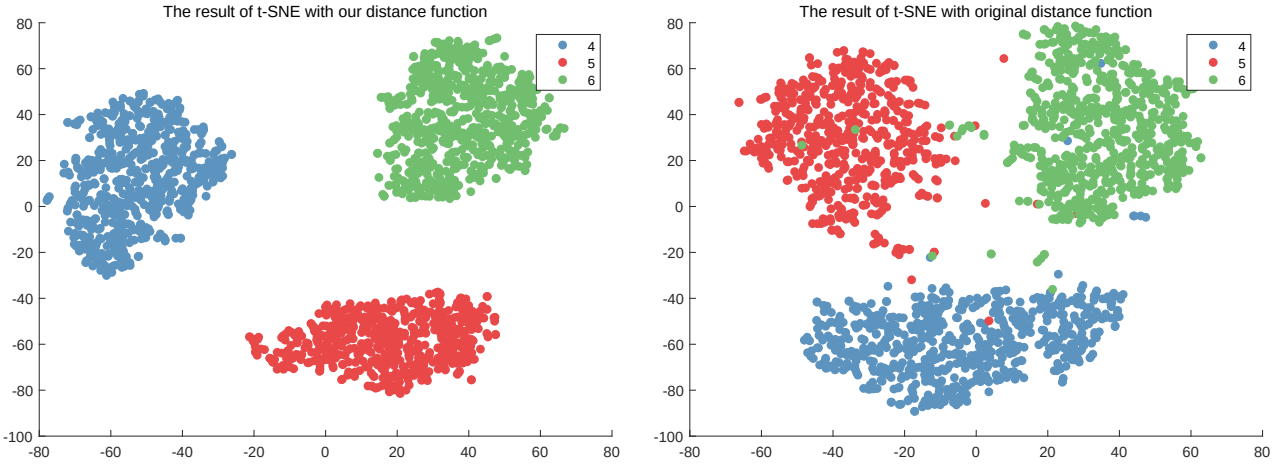
C. t-SNE

Lastly, we compare our distance function with euclidean distance in t-SNE method. In SNE-based method, the pairwise distances are applied to construct the probability that a map point will pick another map point as its neighbor. In t-SNE, a heavy tailed student t-distribution is employed for building the probability. In our experiment, we replace the pairwise distances with our distance function. The perplexity of t-SNE is set to be 30. The results are shown in fig. 9, 10, 11 and 12.



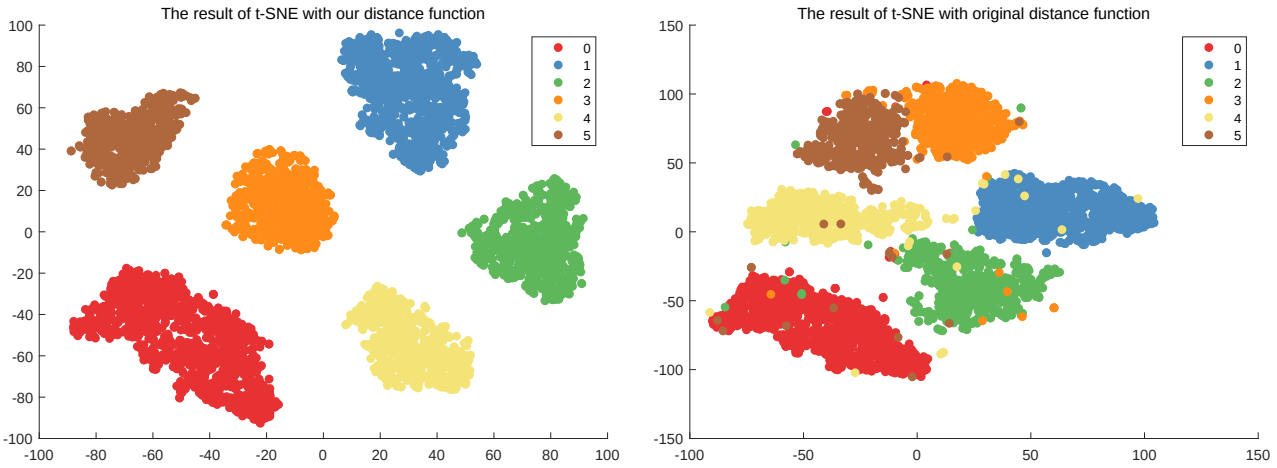
(a) t-SNE result with our distance function. (b) t-SNE result with euclidean distance function.

Fig. 9: t-SNE results with 3 classes data (numbers 0, 1, 2) in Hand-written Digits dataset.



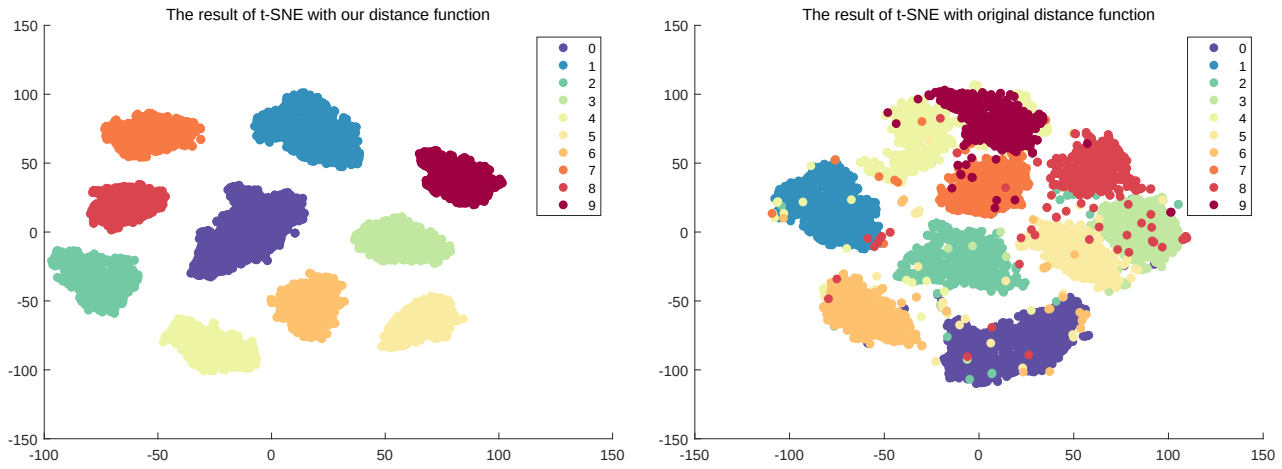
(a) t-SNE result with our distance function. (b) t-SNE result with euclidean distance function.

Fig. 10: t-SNE results with 3 classes data (numbers 4, 5, 6) in Hand-written Digits dataset.



(a) t-SNE result with our distance function. (b) t-SNE result with euclidean distance function.

Fig. 11: t-SNE results with 6 classes data in Hand-written Digits dataset.



(a) t-SNE result with our distance function.

(b) t-SNE result with euclidean distance function.

Fig. 12: t-SNE results with all classes data in Hand-written Digits dataset.

The improvement of our distance function is not satisfactory enough. The gap among classes is clear, where there is no instance located in other classes. However, it is reasonable as our distance function contains the class information. It can be seen that, t-SNE with euclidean distance function gather data in same classes with a relatively high accuracy.

Moreover, the ordinal information is ignored. The main reason could be that, the heavily tail t-distribution only preserves the information for data in close distance, and focus on local distribution. As a result, the data that far from each other does not show great influence on t-SNE method.

IV. Conclusion

In this mini project, we propose a distance function which take class information into account. Moreover, the order of classes is also considered. We apply our distance function in 3 dimensional reduction methods, which includes MDS, isomap and t-SNE. Hand-Written Digits dataset is employed in our experiments. The result on MDS and isomap is acceptable, and the ordinal information is well-preserved. However, the performance on t-SNE is not satisfactory, which may due to the property of t-distribution.

The distance function we proposed can be only conducted on existing data. In the future, a mapping function for ordinal regression dimensional reduction could be considered. With a mapping function, the new data (or testing data) can be mapped into low dimension space and be classified by appropriate ordinal regression methods.

References

[1] P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 42, no. 2, pp. 109–127, 1980.

[2] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[3] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 2, pp. 56–70, 2020.

[4] S. Kaski and J. Peltonen, "Dimensionality reduction for data visualization [applications corner]," *IEEE signal processing magazine*, vol. 28, no. 2, pp. 100–104, 2011.

[5] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[6] A. Mead, "Review of the development of multidimensional scaling methods," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 41, no. 1, pp. 27–39, 1992.

[7] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[8] G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," *Advances in neural information processing systems*, vol. 15, 2002.

[9] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[10] C. R. Rao, "The utilization of multiple measurements in problems of biological classification," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 10, no. 2, pp. 159–203, 1948.

[11] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.