Simulation for approximate nearest neighbors using Swissroll dataset

1 Notation

The data points are denoted as $x_i, i = 1, \ldots, N$ in high-dimensional space $\mathbb{R}^D$ and $y_i, i = 1, \ldots, N$ in low-dimensional space $\mathbb{R}^d$. While δ_{ij} and d_{ij} denote the distance from x_i to x_j and the distance from y_i to y_j respectively.

Based on the distances, the ranks of the points x in high-dimensional space can be calculated as $\rho_{ij} = |\{k : \delta_{ik} < \delta_{ij} \text{ or } (\delta_{ik} = \delta_{ij} \text{ and } k < j)\}|$, where $|\cdot|$ denotes the set cardinality. And the ranks of y are denoted as $r_{ij} = |\{k : dik < dij \text{ or } (dik = dij \text{ and } k < j)\}|$.

Then the K-ary neighborhoods of x_i and y_i can be denoted as $U_K(i) = j : 1 \leq \rho_{ij} \leq K$ and $V_K(i) = j : 1 \leq r_{ij} \leq K$, respectively, where K is the size of the neighborhood.

The co-ranking matrix (Lee, Verleysen, et al. 2008) can then be defined as

$$\mathbf{Q} = [Q_{kl}]_{1 \leqslant k,l \leqslant N-1}, \text{ with } Q_{kl} = |\{(i,j) : \rho_{ij} = k \text{ and } r_{ij} = l\}|.$$

2 Swissroll dataset

To demonstrate, we now apply manifold learning algorithms to the widely-used Swissroll dataset. The key idea to build the Swissroll mapping is to randomly generate N data points in two-dimensional space, called meta data, and then map them to a three-dimensional space with specific smooth functions and some error terms, denoted as X. The manfold learning algorithms are then applied to the 3-D data X to get a 2-D embedding Y. In this way, the algorithms can be applied to obtain the embedding Y, and compare it with the meta data.

Suppose a manifold is given by

$$X = \mathcal{M}(\theta) + \varepsilon,$$

where $\mathcal{M}(\theta)$ is the parameterization of the manifold and $\varepsilon \sim \mathcal{N}(\mu, \sigma^2)$ is the error term.

The mapping function for Swissroll data is given by

$$\begin{cases} X_1 = \theta_1 \times \cos\theta_1 + \varepsilon_1, \\ X_2 = \theta_2 + \varepsilon_2, \\ X_3 = \theta_1 \times \sin\theta_1 + \varepsilon_3. \end{cases}$$

Here we consider a Swissroll dataset consists of `N=2000` points, and the error term $\varepsilon \sim \mathcal{N}(0,\ 0.05^2)$. The 3-D plot of Swissroll data and the corresponding meta data are shown in Figure 1.

```
N <- 2000; sigma <- 0.05
sr <- dimRed::loadDataSet("Swiss Roll", n = N, sigma = sigma)
cols <- dimRed:::colorize(sr@meta[,seq_len(min(3, ncol(sr@meta)))], drop=FALSE])
meta <- sr@meta
par(mfrow = c(1,2))
plot(sr, type = "3vars")
plot(sr@meta, col = cols, xlab = "")
```

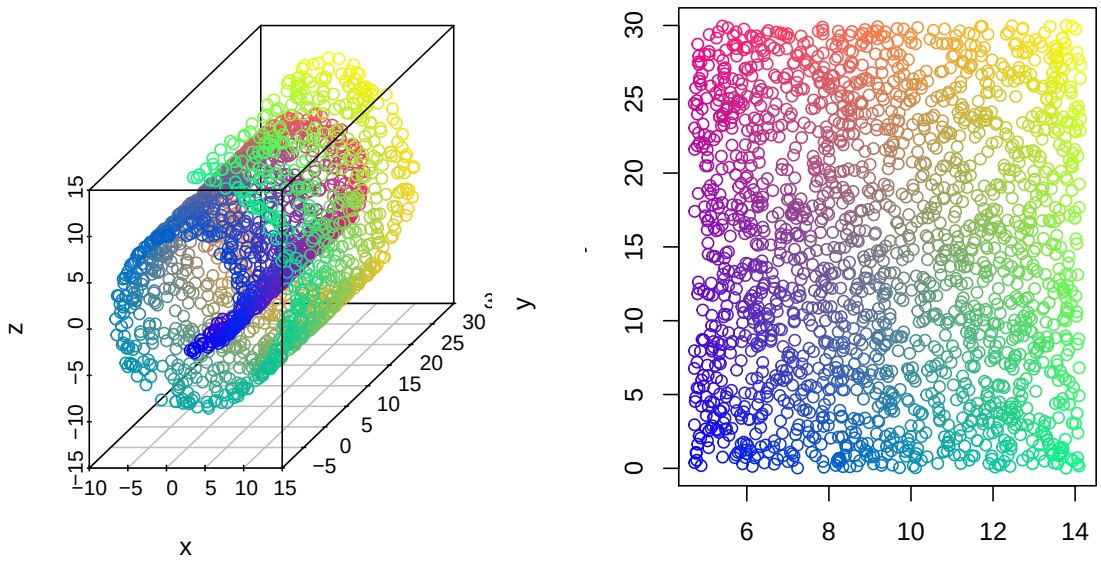


Figure 1: *The 3-D plot and the corresponding meta data plot of a Swissroll data with N=2000.*

3 Isomap embedding

Isomap, short for isometric feature mapping, was one of the first algorithms introduced for manifold learning. It can be viewed as an extension to MDS, a classical method for embedding dissimilarity information into Euclidean space. Isomap consists of three main steps:

1). Construct the K-nearest neighborhood graph for the high-dimensional dataset;

2). Estimate the geodesic distances (distances along a manifold) between points in the input using shortest-path distances (Dijkstra's or Floyd's, Dijkstra (1959); FloydRobert (1962)) on the neighborhood graph;

3). Use MDS to find points in low-dimensional Euclidean space whose interpoint distances match the distances found in Step 2.

When applying Isomap to the Swissroll data, we initialize the number of nearest neighbors (NN) as `K=50`, and the number of embedded dimensions as `d=2`. The 2-D embedding plot is show in Figure 2 (b).

```
K <- 50; d = 2
sr_isomap <- embed(sr, "Isomap", knn = K, ndim = d, get_geod = FALSE,
                   .mute = c("message", "output"))
# X: 3d sr data; Y: 2d embedded data; meta: 2d meta data
X <- as.matrix(sr_isomap@org.data)
Y <- sr_isomap@data@data
pars <- sr_isomap@pars
```

4 Isomap with approximate nearest neighbors (ANN)

The first step of Isomap is to assign neighbors to each data point, which requires calculating pairwise distances. The complexity of pairwise distances is $O(N^2)$ for N observations, which is not efficient when the data size N is large.

One solution for large data size is finding approximate nearest neighbors using kd-trees. By constructing kd-trees, there is no need to compute all pairwise distances, and it scales the computation to $O(N \log(N))$.

When using kd-trees, the dedault value for the error bound parameter is $\epsilon = 0$, which implies exact nearest neighbour search. By tuning this parameter, we could attain the embedding using ANN. We start with $\epsilon = 1$ and get the embedding plot in Figure 2 (c).

```
eps <- 1
sr_isomap_ann <- embed(sr, "annIsomap", knn = K, eps = eps, ndim = d, get_geod = FALSE)
X1 <- as.matrix(sr_isomap_ann@org.data)
Y1 <- sr_isomap_ann@data@data
pars1 <- sr_isomap_ann@pars
```

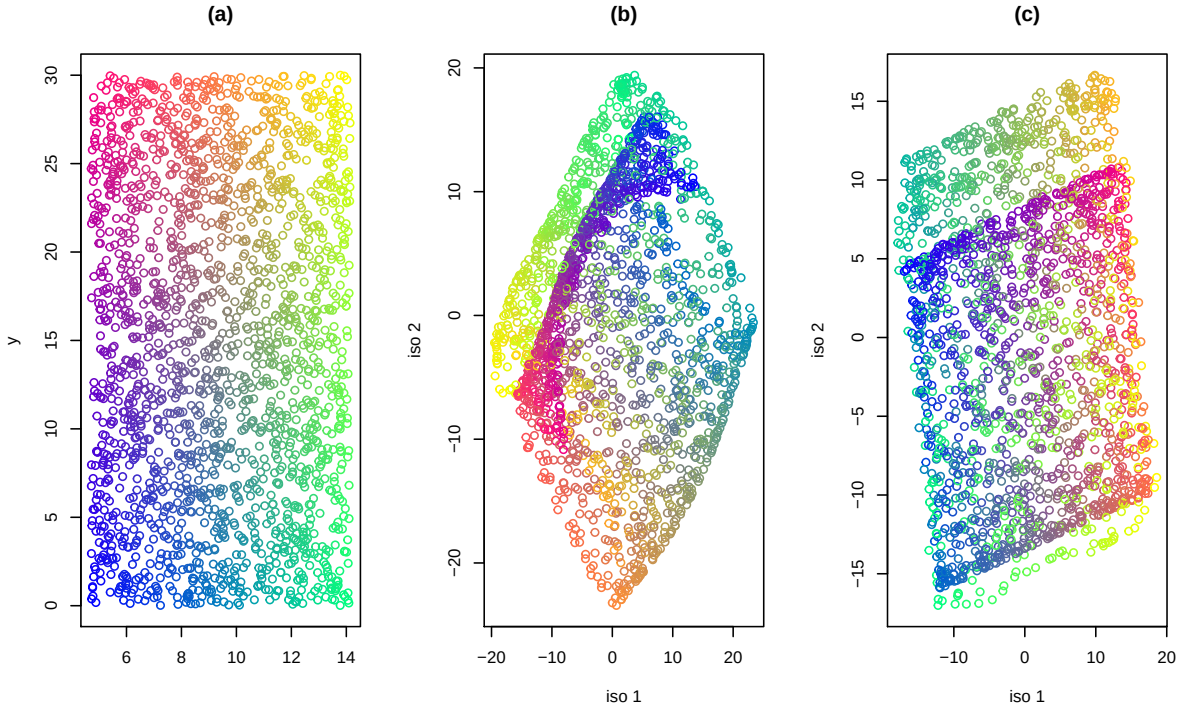


Figure 2: *The Swissroll meta data and the Isomap 2-D embedding using exact NN and ANN with K=50 and eps=1. (a) Swissroll meta data with N=2000 points; (b) Isomap embedding of Swissroll data using exact NN; (c) Isomap embedding of Swissroll data using ANN.*

5 Quality measures

With various manifold learning algorithms, it is worth considering the question of quality assessment to compare and choose from these methods. We can also apply these measures to explore different variation of each method, including using exact NN and ANN. Lee & Verleysen (2008) and Gracia et al. (2014) provided some good reviews on the rank-based criteria of dimensionality reduction quality.

Manifold learning aims at a mapping $\mathcal{M}$ from high-dimensional datasets to low-dimensional representations such that $\mathrm{y} = \mathcal{M}(\mathrm{x})$. If we could obtain the inverse $\mathcal{M}^{-1}$ in closed form, then the reconstruction error can be used as a quality criterion among and within different methods. The reconstruction error can be written as an expectation

$$E_{\text{rec}} = \mathrm{E}\left[\left(x - \mathcal{M}^{-1}(\mathcal{M}(x))\right)^2\right].$$

This approach is only applicable to few algorithms, such as PCA and auto-encoders, bacause most ML methods are nonparametric and the closed form of $\mathcal{M}$ and $\mathcal{M}^{-1}$ is not available.

Since most methods aim at optimizing a given objective function, it is straightforward to calculate the value of objective function after convergence. However, due to different settings of the function, this comparison is limited within the same algorithm.

Then, we try to assess the intrinsic goal of preserving the data set structure, which can be relaxed in the constrains as in the objective function. For a broader applicability, several quality assessment criteria have been proposed, including three rank-based criteria: the trustworthiness and continuity (T&C) measures (Venna & Kaski 2006), the local continuity meta-criterion (LCMC) (Chen & Buja 2009), and the mean relative rank errors (MRREs) (Lee & Verleysen 2007), and Procrustes measure (Goldberg & Ritov 2009). All these criteria are built upon the idea that a faithful embedding preserves the local neighborhood structure of each point (2009). Therefore, they determine the quality of the embedding by analyzing the degree in preserving the K-ary neighborhood structures, for varying value of K.

5.1 Trustworthiness & Continuity (T&C)

Venna & Kaski (2006) defined two quality measures for manifold embeddings to distinguish two type of errors where distant points become neighbors, or neighbors are embedded faraway from each other.

- the trustworthiness of the embedding, where trustworthiness errors are defined as distant input points that entered the same output neighborhood.

$$M_T(K) = 1 - \frac{2}{G_K} \sum_{i=1}^{N} \sum_{j \in V_K(i) \notin U_K(i)} (\rho ij - K),$$

where the normalizing factor

$$G_K = \begin{cases} NK(2N - 3K - 1) & \text{if } K < N/2 \\ N(N-K)(N-K-1) & \text{if } K \geqslant N/2 \end{cases}$$

- the continuity of the embedding, where continuity errors are defined as data points in the same input neighborhood but in different output neighborhood.

$$M_C(K) = 1 - \frac{2}{G_K} \sum_{i=1}^{N} \sum_{j \in U_K(i) \notin V_K(i)} (rij - K).$$

5.2 Mean Relative Rank Errors (MRREs)

Lee & Verleysen (2008) developed the mean relative rank errors which is based on similar principle to that of the T&C, while the two elements are defined as

$$W_n(K) = 1 - \frac{1}{H_K} \sum_{i=1}^{N} \sum_{j \in U_K(i)} \frac{|\rho ij - rij|}{\rho ij},$$
$$W_v(K) = 1 - \frac{1}{H_k} \sum_{i=1}^{n} \sum_{j \in V_{k(i)}} \frac{|\rho ij - rij|}{rij},$$

where H_K is the normalizing factor defined as

$$H_K = n \sum_{i=1}^{K} \frac{|N - 2i + 1|}{i}.$$

The MRREs criterion Q_M evaluates (using an error value) the first K rows and columns of the co-ranking matrix Q.

5.3 Local Continuity Meta-Criterion (LCMC)

Chen & Buja (2009) suggested the local continuity criterion to compute the average size of the overlap of K-nearest neighborhoods in the low-dimensional embedding and in the high-dimensional space. The LCMC is defined as

$$Q_{LC}(K) = 1 - \frac{1}{NK} \sum_{i=1}^{N} \left| U_K(i) \bigcap V_K(i) \right| - \frac{K^2}{N-1}.$$

If the overlap between two K neighboring sets is calculated, then the $Q_{LC}(K)$ gives a general measurement for the local faithfulness of the computed embeddings. The interval of $Q_{LC}(K)$ is

$[0, 1]$, and values next to 1 mean a high neighborhood overlap between the two dimensional spaces, and next to 0 values the opposite.

From an intuitive point of view, T&C and MRREs try to detect what goes wrong in a given embedding, whereas the LCMC accounts for things that work well.

5.4 Co-ranking Matrix ($Q_{NX}(K)$)

Many different concepts and quality criteria for DR can be summarized using the co-ranking framework, presented by Lee & Verleysen (2008). Several of the aforementioned methods based on distance ranking in local neighborhoods (T&C, MRREs, LCMC), are easily unified into an overall framework.

The co-ranking matrix Q is defined with its element being

$$Q_{kl} = \left|\left\{(i,j)|\rho_{ij} = k \text{ and } r_{ij} = l\right\}\right|.$$

Errors of a DR mapping correspond to off-diagonal entries of this co-ranking matrix. A point j that gets a lower rank with respect to a point i in the low-dimensional space than in the high-dimensional space, i.e. $\rho_{ij} > r_{ij}$, is called an *intrusion*. Analogously, if δ_j has a higher rank in the low-dimensional space it is called an *extrusion*. As shown in Figure 4 from Lee & Verleysen (2008), intrusions and extrusions correspond to off-diagonal entries in the upper or lower triangle, respectively.

We can also associate the above quality measures with co-ranking matrix using the idea of intrusions and extrusions as Figure 3.

Since the preservation of local relationships is important, rank errors for large ranks are not as critical as rank errors of close points. Therefore, Lee and Verleyssen distinguished two types of intrusions/extrusions, those within a K-neighborhood, which are benevolent, and those moving across this boundary, which are malign with respect to quality. A *K-intrusion* (resp. *K-extrusion*) is an intrusion for which $r_{ij} < K$ (resp. $\rho_{ij} < K$). Subsequently, mild K-intrusions are events for which $r_{ij} < \rho_{ij} \leq K$, while hard K-intrusions are defined by $r_{ij} \leq K < \rho_{ij}$. Mild K-extrusions and hard K-extrusions are defined accordingly.

Define $Q_{NX}(K)$ as the criterion that summarizes co-ranking matrix Q in a simple way: it counts the number of points that remain inside the K-neighborhood while projecting, i.e., all points

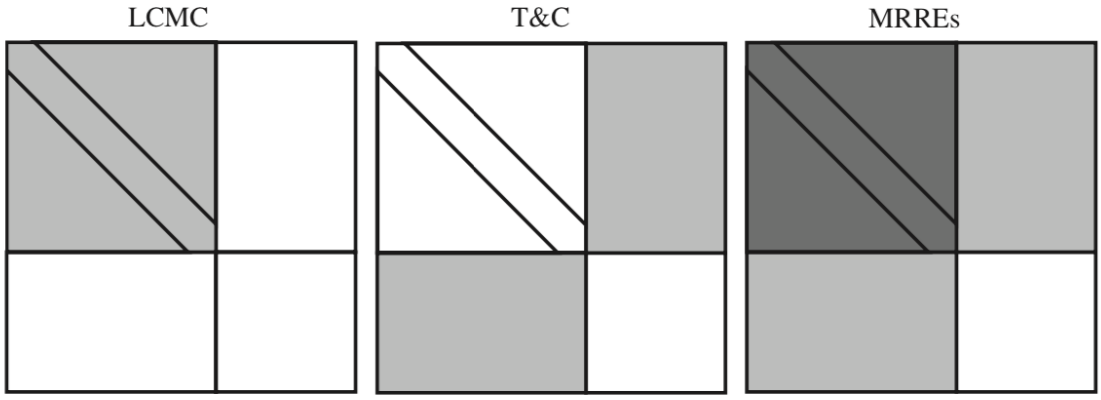


Figure 3: *For all pairs of quality criteria, a schematic illustration of the co-ranking matrix is shown: the blocks that are taken into account are shaded. The LCMC quantifies the true positives, the TC focus on the false positives and false negatives, and the MRREs encompass the positives (true and false) and negatives (true and false). For the MRREs, the block UL is covered twice.*

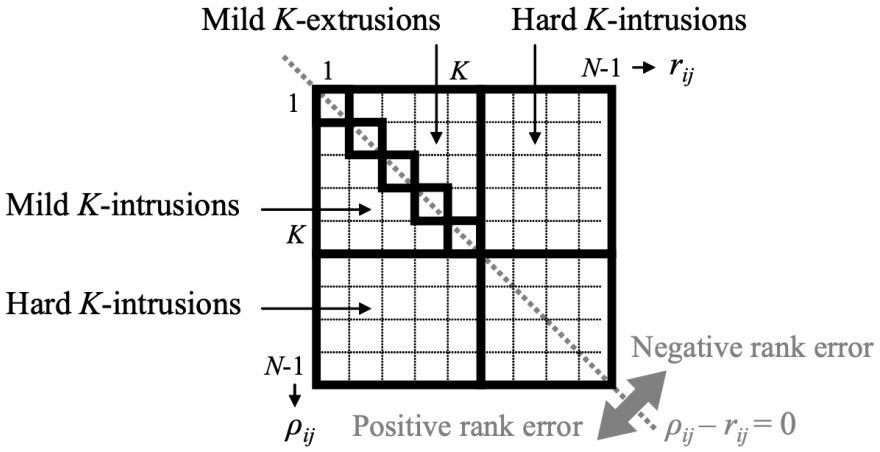


Figure 1: Block division of the co-ranking matrix, showing the different types of intrusions and extrusions, and their relationship with the rank error.

Figure 4: *Intrusions and extrusions of co-Ranking matrix.*

which keep their rank, and all mild in- and extrusions. It is defined as

$$Q_{NX}(K) = \frac{1}{KN} \sum_{k=1}^{K} \sum_{l=1}^{K} Q_{kl}.$$

The range is $Q_{NX}(K) \in [0, 1]$, where 1 means a perfect embedding.

5.5 Procrustes measure ($R(X, Y)$)

Despite the K-ary neighborhood based measures, Goldberg & Ritov (2009) also presented the Procrustes measure, a novel measure based on Procrustes rotation that enables quantitative

comparison of the output of manifold learning algorithms. The function based on the Procrustes analysis works as a measure of local embedding quality, and compares each neighborhood in the highdimensional space and its corresponding low-dimensional embedding.

First, define the Procrustes statistic $G(X, Y)$ as

$$\begin{aligned} G(X,Y) &= \inf_{\{A,b:A'A=I,b\in\mathbb{R}^D\}} \sum_{i=1}^{k} \|x_i - Ay_i - b\|^2 \\ &= \inf_{\{A,b:A'A=I,b\in\mathbb{R}^D\}} \operatorname{tr}\left(\left(X - YA' - 1b'\right)'\left(X - YA' - 1b'\right)\right) \\ &= \inf_{\{A:A'A=I\}} \operatorname{tr}\left(\left(X - YA'\right)' H \left(X - YA'\right)\right) \\ &= \inf_{\{A:A'A=I\}} \left\|H\left(X - YA'\right)\right\|_F^2, \end{aligned}$$

where $\|\|_F$ is the Frobenius norm.

They define how well an embedding preserves the local neighborhoods using the Procrustes statistic $G(X_i, Y_i)$ of each neighborhood-embedding pair (X_i, Y_i). Therefore, a global embedding that preserves the local structure can be found by minimizing the sum of the Procrustes statistics of all neighborhood-embedding pairs.

Let X_i be the neighborhood of $x_i (i = 1, \ldots, N)$ in the high dimension and Y_i be its embedding. Define

$$R(X,Y) = \frac{1}{N}\sum_{i=1}^{N} G\left(X_i, Y_i\right).$$

The function R measures the average quality of the neighborhood embeddings. Embedding Y is considered better than embedding $\tilde{Y}$ in the local-neighborhood-preserving sense if $R(X, Y) < R(X, \tilde{Y})$. This means that on the average, Y preserves the structure of the local neighborhoods better than $\tilde{Y}$.

6 Quality measures for Isomap embeddings

With these quality measures, we can further calculate the Isomap embedding quality of both exact NN and ANN as shown in Table **??**.

```
quality_isomap <- dr_quality(X, Y, pars)
quality_isomap_ann <- dr_quality(X1, Y1, pars1)
qualities <- rbind(quality_isomap$quality, quality_isomap_ann$quality)[, 1:6]
```

```
compare <- microbenchmark::microbenchmark(
  embed(sr, "Isomap", knn = K, ndim = d, get_geod = FALSE),
  embed(sr, "annIsomap", knn = K, eps = eps, ndim = d, get_geod = FALSE),
  times = 1,
  unit = "s"
)
# average running time
# summary(compare)[,"mean"]
```

```
qualities <- cbind(qualities, Time_sec = summary(compare)[, "mean"])
row.names(qualities) <- c("NN", "ANN")
knitr::kable(qualities, digits = 3) %>%
  kable_styling(latex_options = "striped", full_width = TRUE)
```

	K	M_T	M_C	W_n	W_nu	lcmc	Time_sec
NN	21	0.983	0.873	0.010	0.005	0.476	6.782
ANN	21	0.983	0.882	0.011	0.006	0.425	8.534

7 Improvements

The difference of the quality measures between exacct NN and ANN is not obvious. To improve the results, there are several tuning parameters.

7.1 Optimal parameters: K, eps

By varying the values of `K` and `eps`, the quality measures can guide in finding the optimal parameters. The optimal value for K can be obtained from the quality output, which is NA. Then we focus on tuning `eps`.

```
K <- qualities$K[1]
eps_seq <- seq(0, 2, 0.1)
quality_eps <- matrix(NA, length(eps_seq), ncol(qualities) + 1)
colnames(quality_eps) <- c(colnames(qualities), "eps")

for(i in 1:length(eps_seq)){
```

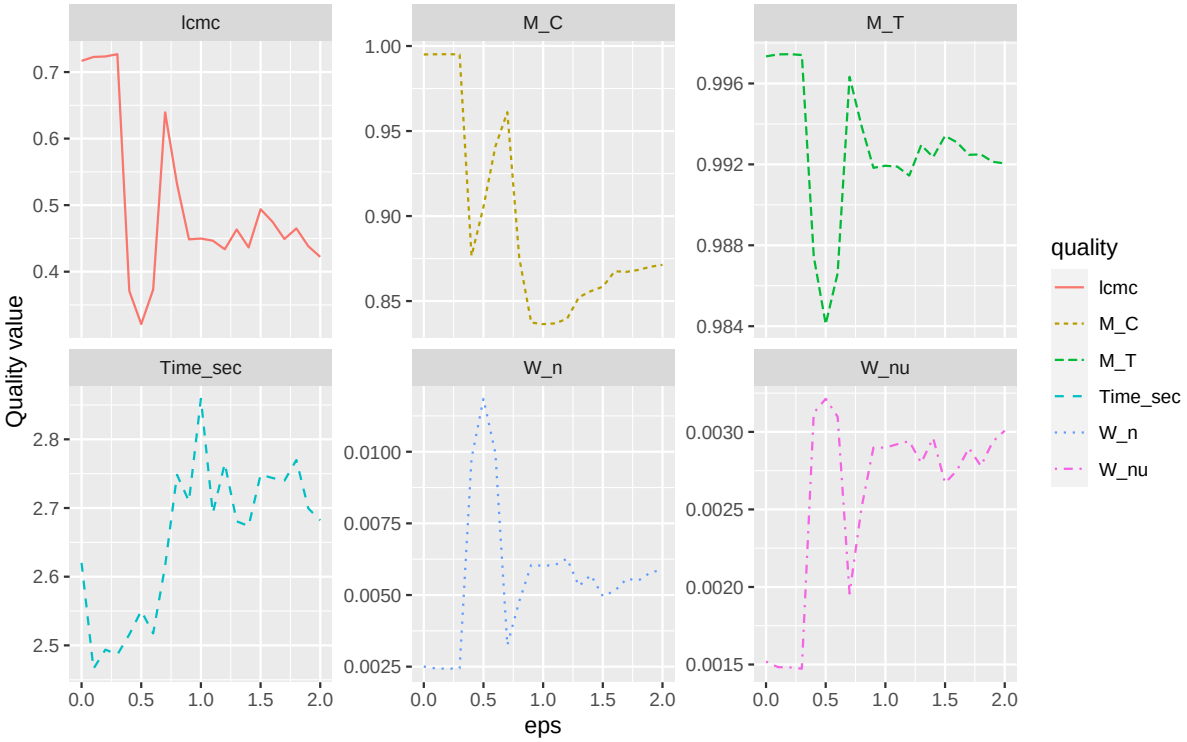
```
  eps <- eps_seq[i]
  # cat("eps =", eps, "\n")
  quality_eps[i, "Time_sec"] <- summary(microbenchmark::microbenchmark(
    sr_isomap_eps <- dimRed::embed(sr, "annIsomap", knn = K, eps = eps, ndim = d, get_geo
    times = 3,
    unit = "s"
  ))[,"mean"]

  Xi <- as.matrix(sr_isomap_eps@org.data); Yi <- sr_isomap_eps@data@data
  parsi <- sr_isomap_eps@pars

  quality_eps[i, 1:6] <- as.matrix(dr_quality(Xi, Yi, parsi)$quality[, 1:6])
  quality_eps[i, "eps"] <- eps
}
```

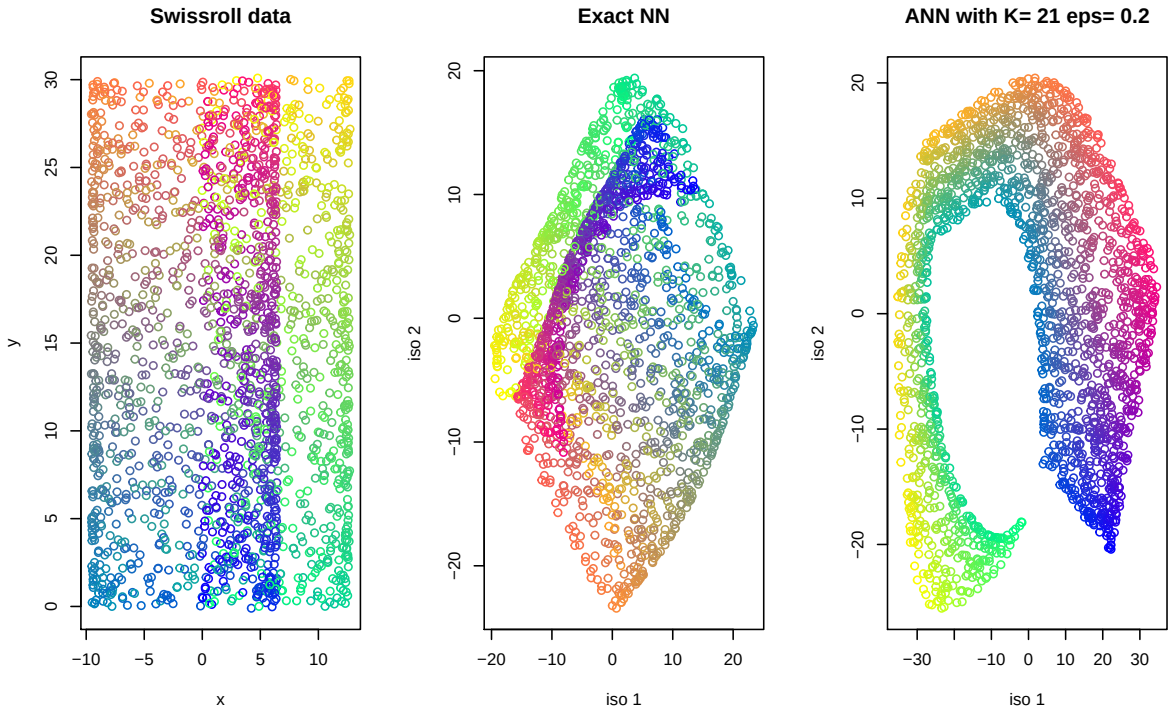
Finally, we can plot all the quality measures and computation time to find optimal eps.

```
df <- tidyr::gather(as_tibble(quality_eps), key = "quality", value = "value", -eps, -K)
p <- ggplot(df, aes(x = eps, y = value)) +
  geom_line(aes(color = quality, linetype = quality), size = 0.5) +
  # theme_bw() +
  labs(y = "Quality value") +
  facet_wrap(. ~ quality, ncol = 3, scales = "free_y")
p
```




```
# Plotting using optimal parameters
eps_optim <- quality_eps[which.max(as.tibble(quality_eps[-1,])$lcmc), "eps"]
sr_isomap_ann_optim <- embed(sr, "annIsomap", knn = K, eps = eps_optim, ndim = 2, get_geo


par(mfrow = c(1,3))
plot(sr, type = "2vars", main = "Swissroll data")
plot(sr_isomap, type = "2vars", main = "Exact NN")
plot(sr_isomap_ann_optim, type = "2vars", main = paste("ANN with K=", K, "eps=", eps_opti
```



7.2 Larger data size: N

A larger data size might show an obvious improvement with ANN.

7.3 More algorithms

7.4 More manifolds

8 Questions

1) With each setting (different algorithms or different NN searching method), we can get

- the embedding plot vs the meta data plot,
- different quality measures.

How do we compare the embedding plots?

2) Our expectation is to get "faster and not too much worse" ANN. "Faster" can be achived by increasing the data size and compare the computation time in seconds ("Time_sec" column in the quality table).

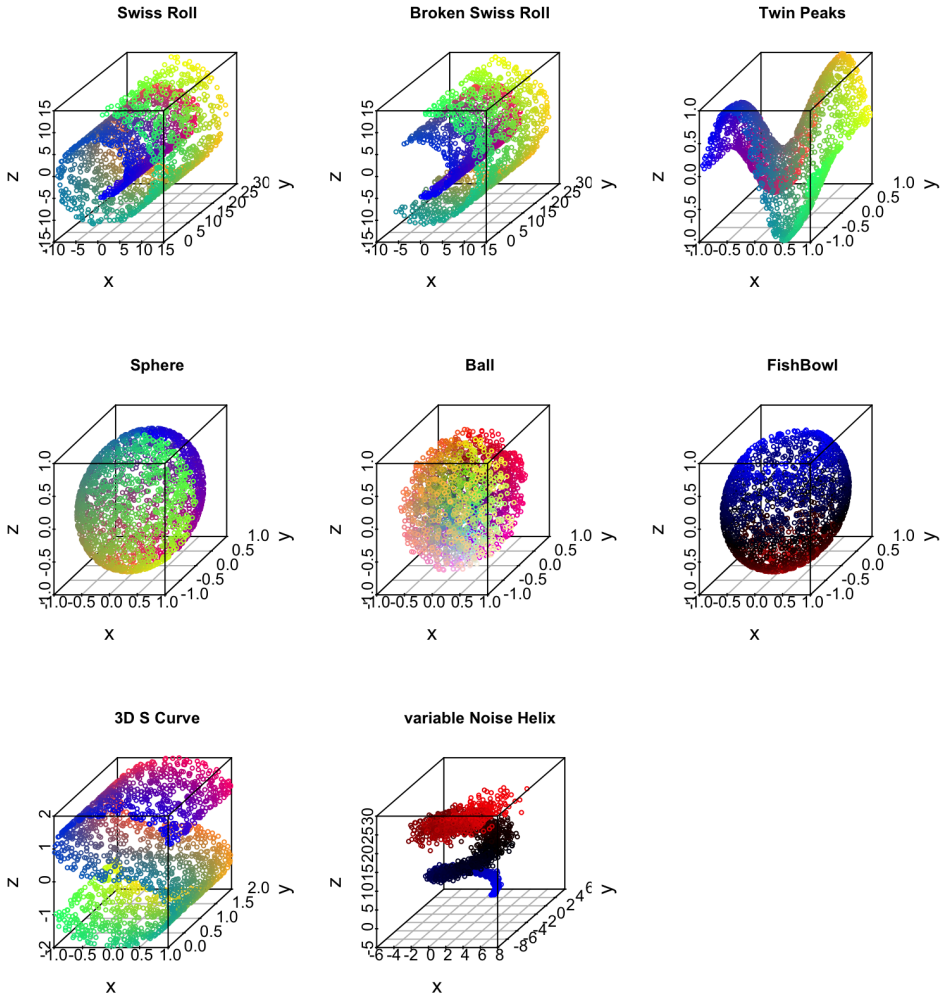
How do we define "not too much worse" with the plots and quality table?

3) To find optimal parameters, how to define “optimal” here?

Further, how do we reach a trade-off between computation time and quality measures?

4) Since Isomap is not doing a good job here, what else algorithms would be the top options?

5) Similar to 4), what else manifolds despite Swissroll?



References

Chen, L & A Buja (Mar. 2009). Local Multidimensional Scaling for Nonlinear Dimension Reduction, Graph Drawing, and Proximity Analysis. *J. Am. Stat. Assoc.* **104**(485), 209–219.

Dijkstra, EW (Dec. 1959). A note on two problems in connexion with graphs. *Numer. Math.* **1**(1), 269–271.

FloydRobert, W (June 1962). Algorithm 97: Shortest path. en. *Commun. ACM*.

Goldberg, Y & Y Ritov (Oct. 2009). Local procrustes for manifold embedding: a measure of embedding quality and embedding algorithms. *Mach. Learn.* **77**(1), 1–25.

Gracia, A, S González, V Robles & E Menasalvas (June 2014). A methodology to compare Dimensionality Reduction algorithms in terms of loss of quality. *Inf. Sci.* **270**, 1–27.

Lee, J & M Verleysen (2008). Quality assessment of nonlinear dimensionality reduction based on K-ary neighborhoods. In: *New Challenges for Feature Selection in Data Mining and Knowledge Discovery*. jmlr.org, pp.21–35.

Lee, JA & M Verleysen (Oct. 2007). *Nonlinear Dimensionality Reduction*. en. Springer Science & Business Media.

Lee, JA, M Verleysen, et al. (2008). Rank-based quality assessment of nonlinear dimensionality reduction. In: *ESANN*. elen.ucl.ac.be, pp.49–54.

Venna, J & S Kaski (July 2006). Local multidimensional scaling. en. *Neural Netw.* **19**(6-7), 889–899.