Lab x005 - ISOMAP, LLE, t-SNE

Jairo Cugliari

M2 MALIA - UL2

Exercice 1 : Isomap sur des variétés développables

1. Créer le jeu de données `swissroll` (cf. code ci-dessous).
2. Obtenir une configuration par MDS classique à partir de la matrice de distances.
3. Examiner les valeurs singulières de la décomposition spectrale. Combien de dimensions sont nécessaires pour représenter ces données ?
4. Obtenir la matrice de distances sur la configuration projetée sur le plan principal (les deux premières dimensions).
5. Obtenir une configuration par Isomap à l'aide du package `vegan`.
6. Examiner les valeurs singulières de la décomposition spectrale. Combien de dimensions sont nécessaires pour représenter ces données selon Isomap ?

Code pour générer les données swissroll (in R)

```
n <- 1000 # Random position on the parametric domain.
u <- matrix(runif(2 * n), ncol = 2)

v <- 3 * pi / 2 * (0.1 + 2 * u[, 1])

x <- -cos(v) * v
y <- 20 * u[, 2]
z <- sin(v) * v

swissroll <- cbind(x, y, z)
plot3d(swissroll[order(v), ], col = rainbow(n), size = 10)
```

Exercice 2 : LLE

Le jeu de données est défini comme suit :

$$x = e^{-0.2\cdot\left(\frac{-t}{10}\right)}\cdot\cos\left(\frac{-t}{10}\right), \quad y = e^{-0.2\cdot\left(\frac{-t}{10}\right)}\cdot\sin\left(\frac{-t}{10}\right), \quad t = 1,2,\ldots,300$$

Les données correspondantes peuvent être représentées par:

$$\text{dat} = \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{300} & y_{300} \end{pmatrix}$$

1. Obtenez le jeu de données simulées et représentez-les dans un graphique.
2. Appliquez une ACP sur les données. Quels sont les vecteurs propres (variance expliquée) des composantes principales obtenues ? Tracez la projection des points sur la première composante principale.
3. Utilisez la méthode LLE pour obtenir une représentation en 1 dimension du jeu de données. Tracez la représentation en 1D obtenue avec LLE. Comment se compare-t-elle à la projection sur la première composante principale obtenue précédemment ?

Exercice 3 : LLE

Vous allez implémenter LLE. La fonction principale est la suivante :

```
lle <- function(x, q, k = q + 1, alpha = 0.01) {
  stopifnot(q > 0, q < ncol(x), k > q, alpha > 0)
  kNNs = find.kNNs(x, k)
  w = reconstruction.weights(x, kNNs, alpha)
  coords = coords.from.weights(w, q)
  return(coords)
```

```
}
```

L'algorithme prend les paramètres suivants :

- `x` : un jeu de données où chaque ligne représente un point dans un espace de haute dimension.
- `q` : la dimension de l'espace de sortie (plus petit que l'espace d'origine).
- `k` : le nombre de voisins à prendre en compte pour chaque point (par défaut, $k = q+1$).
- `alpha` : un paramètre régularisateur pour éviter les problèmes d'inversion de matrice.

1. Ecrire la fonction `find.kNNs` qui calcule les distances entre chaque point du jeu de données et sélectionne les k plus proches voisins pour chaque point :

```
find.kNNs <- function(x, k, ...) {...}
smallest.by.rows <- function(m, k) {...}
```

2. La fonction `reconstruction.weights` calcule les poids qui permettent de reconstruire chaque point à partir de ses k plus proches voisins. Ces poids sont ensuite utilisés pour la réduction de dimension.

```
reconstruction.weights <- function(x, neighbors, alpha) {...}
local.weights <- function(focal, neighbors, alpha) {...}
```

3. La fonction `coords.from.weights` utilise les poids de reconstruction pour déterminer les nouvelles coordonnées dans l'espace de dimension réduite :

```
coords.from.weights <- function(w, q, tol = 1e-07) {...}
```

4. Testez la fonction complète `lle` avec un jeu de données bidimensionnel simulé.

Exercice 4 : t-SNE

1. Visitez le site `https://experiments.withgoogle.com/t-sne-map` pour expérimenter avec la visualisation de t-SNE.
2. Utiliser t-SNE sur un échantillon de 1000 images de MNIST
3. Utiliser différentes valeurs de perplexité (essayer de valeurs entre 5 - 50).
4. Comparer la projection obtenue avec celles de méthodes comme Isomap ou LLE.
5. Lire `https://distill.pub/2016/misread-tsne/` (très important pour l'interpretation de résultats). Choisir un ou deux exemples et les reproduire.