UNIVERSITÉ PARIS DESCARTES

Exploration visuelle des données

Nicoleta ROGOVSCHI

nicoleta.rogovschi@parisdescartes.fr

M2-INFO

Isometric feature mapping (Isomap)

Plan du cours

- Introduction et définitions
- Algorithme
- Exemple
- Conclusions

Réduction des dimensions par extraction de caractéristiques

Deux grandes familles de méthodes :

- **Méthodes linéaires**
  - Analyse en Composantes Principales (ACP)
  - Analyse Discriminante Linéaire (ADL)
  - Multi-Dimensional Scaling (MDS)
  - …
- **Méthodes non-linéaires**
  - → Isometric feature mapping (Isomap)
  - Locally Linear Embedding (LLE)
  - Kernel PCA
  - Segmentation spectrale (spectral clustering)
  - Methodes supervisées (S-Isomap)
  - …

Rappel MDS

On rencontre deux types de technique de MDS :

- MDS métrique (MDS classique)
  - On suppose que D est la matrice des distances aux carrée.

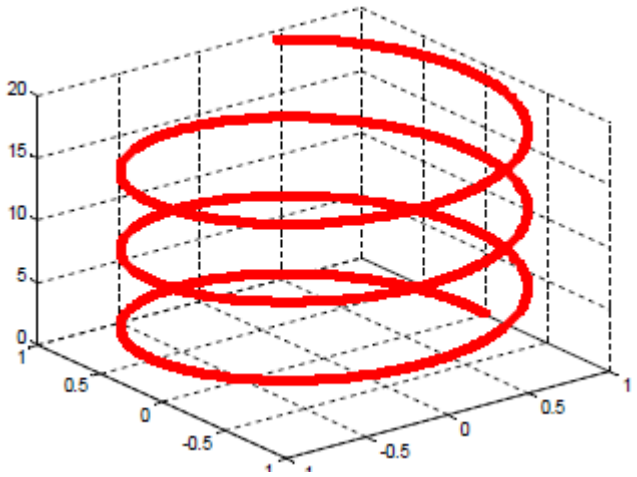
- MDS non-métrique
  - Traite des mesures de dissimilarités plus générales.

Introduction

- Des techniques comme : ADL, ACP et leurs variantes réalise une transformation globales des données
  - Ces techniques supposent que le maximum d'information dans les données est contenu dans un sous-espace linéaire
  - Quelle approche on va utiliser quand les données sont imbriquées dans un espace non-linèaire ?

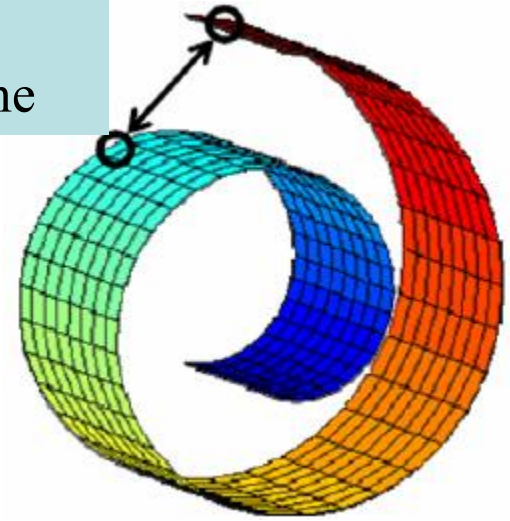
Introduction

- L'ACP ne peut pas découvrir la structure d'un jeu de données sous forme de spirale

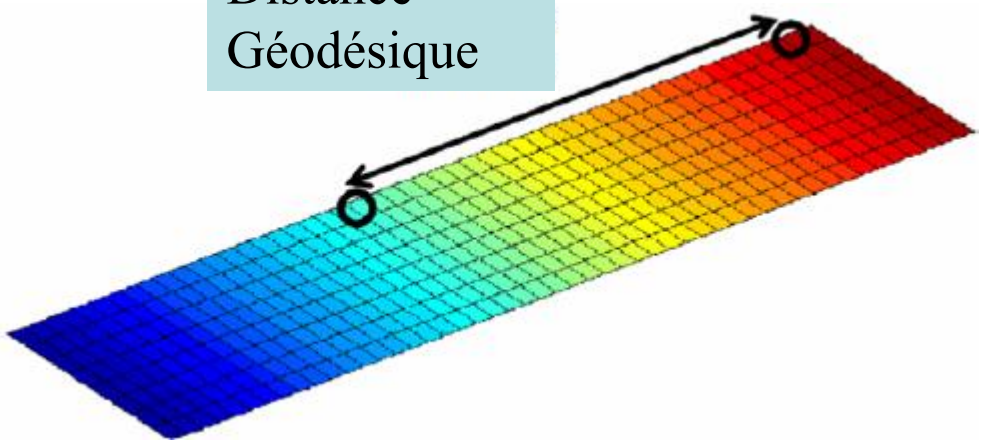


Distance Euclidienne vs. Distance Géodésique

Distance Euclidienne



Distance Géodésique



Introduction

- Le but de ISOMAP est de trouver une variété non-linéaire contenant les données
- On utilise le fait que pour des points proches, la distance euclidienne est une bonne approximation de la distance géodésique sur la variété
- On construit un graphe reliant chaque point à ses *k* plus proches voisins

Introduction

- Les longueurs des géodésiques sont alors estimées en cherchant la longueur du plus court chemin entre deux points dans le graphe
- Par la suite on applique MDS aux distances obtenues afin de déterminer un positionnement des points dans un espace de dimension réduite

ISOMAP

- ISOMAP [Tenebaum et al. 2000]
  - Pour des points voisins, la distance euclidienne fournit une bonne approximation à la distance géodésique
  - Pour des points éloignés, la distance géodésique peut être approximée avec une séquence de pas entres les groupes des points voisins

ISOMAP

ISOMAP est composé de 3 étapes:

1. Construire le graphe de voisinage G
2. Pour chaque paire de points du G, calculer le plus court chemin (la distance géodésique)
3. Utiliser le MDS classique sur les distances géodésiques

Distance euclidienne -> Distance géodésique

Algorithme ISOMAP

- Etape 1
  - Construire le graphe de voisinage, basé sur les distances $d_X(i,j)$ dans l'espace de départ X.
  - On peut le faire de deux manières différentes:
    - Connecter chaque points à tous les points selon un rayon fixé ε
    - Connecter chaque points à tous ses k plus proches voisins
  - On obtient un graphe pondéré de voisinage G, ou

$d_X(i,j)$ est le poids de chaque arrêt entre les points voisins.

Algorithme ISOMAP

- Etape 2
  - Calculer les distances géodésique $d_M(i,j)$ entre toutes les paires de points de la variété M en calculant les plus courts chemins $d_G(i,j)$ dans le graphe G.
  - On peut le faire en utilisant l'algorithme de Dijkstra ou l'algorithme de Floyd.

Algorithme ISOMAP

- Etape 3
  - Appliquer MDS classique sur la matrice du graphe des distances D.
  - Les vecteurs des coordonnées y_i sont déterminés de manière à minimiser la fonction de cout suivante:

$$E = \left\| \tau(D_G - \tau(D_Y)) \right\|_{L^2}$$

    - Où D_Y représente la matrice des distances euclidiennes

$\{d_y(i,j)=\| y_i - y_j \|\}$ et l'opérateur τ est déterminé de la manière suivante:

$$\tau = -HSH/2$$

    - Où S est la matrice des distances au carré $\{S_{ij}=D^2_{ij}\}$ et H est la matrice de centrage définit de la manière suivante :

$$H = I - \frac{1}{N} ee^T; \qquad e = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \end{bmatrix}^T$$

  - Le minimum global de E est obtenu en attribuant aux coordonnées y_i les *d* vecteurs propres les plus en haut de la matrice $\tau(D_G)$.

Complexité de ISOMAP

- Pour des jeux de données de grandes tailles ISOMAP peut être assez lent :

- Etape 1: Complexité de k-plus proches voisins $O(n^2 D)$

- Etape 2 : Complexité de l'algorithme de Djikstra
  $O(n^2 \log n + n^2 k)$

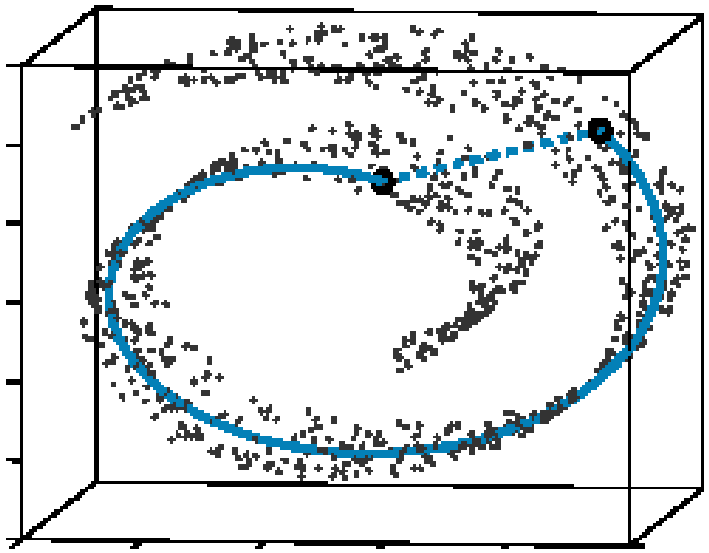
- Etape 3 : Complexité de MDS $O(n^2 d)$

Le jeu de données «Swiss roll»

• Le jeu de données «Swiss roll» contient 20000 points.

• On représente dans cette figure un échantillon de 1000 points.

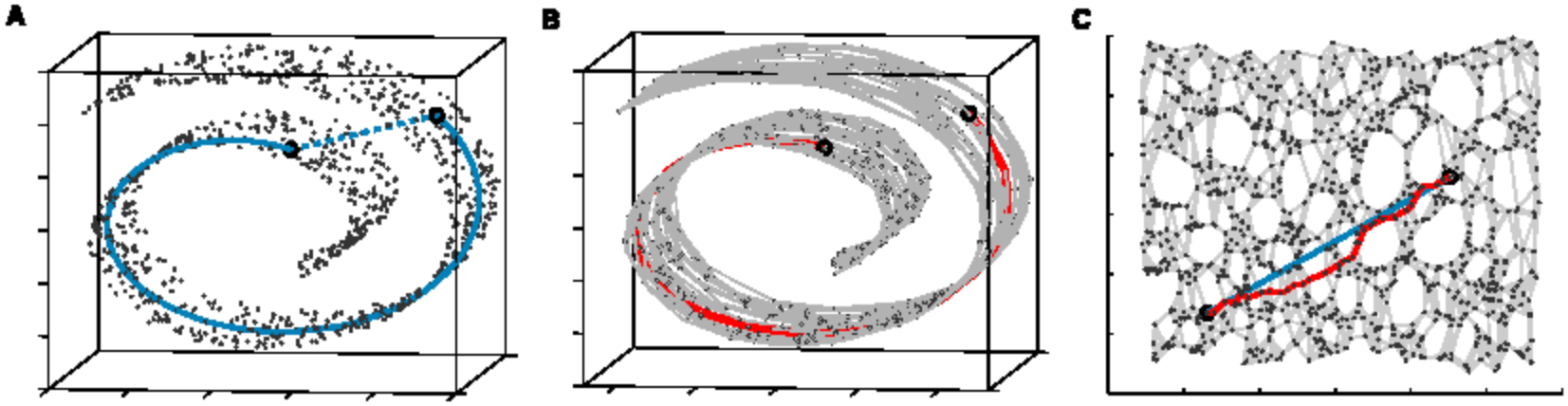
• Par la suite on va représenter sur cette exemple le déroulement de l'algorithme de ISOMAP.



Construction du graphe de voisinage G

K- plus proches voisins (K=7)

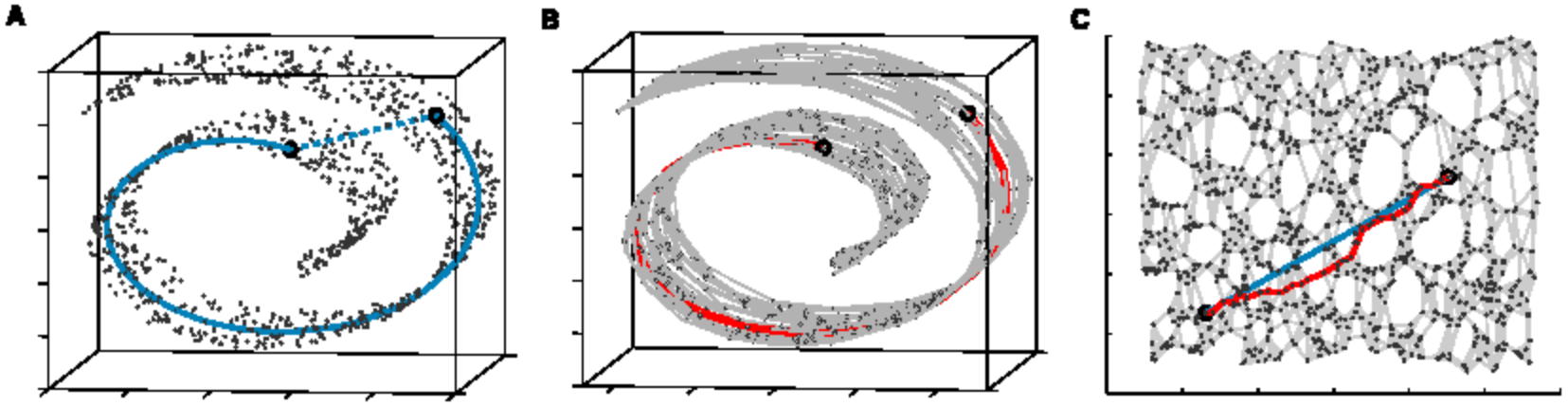
D_G est une matrice de distance Euclidienne 1000 x 1000 de deux points voisins (figure A)



[Tenebaum et al. 2000]

Calcul des plus courts chemins dans G

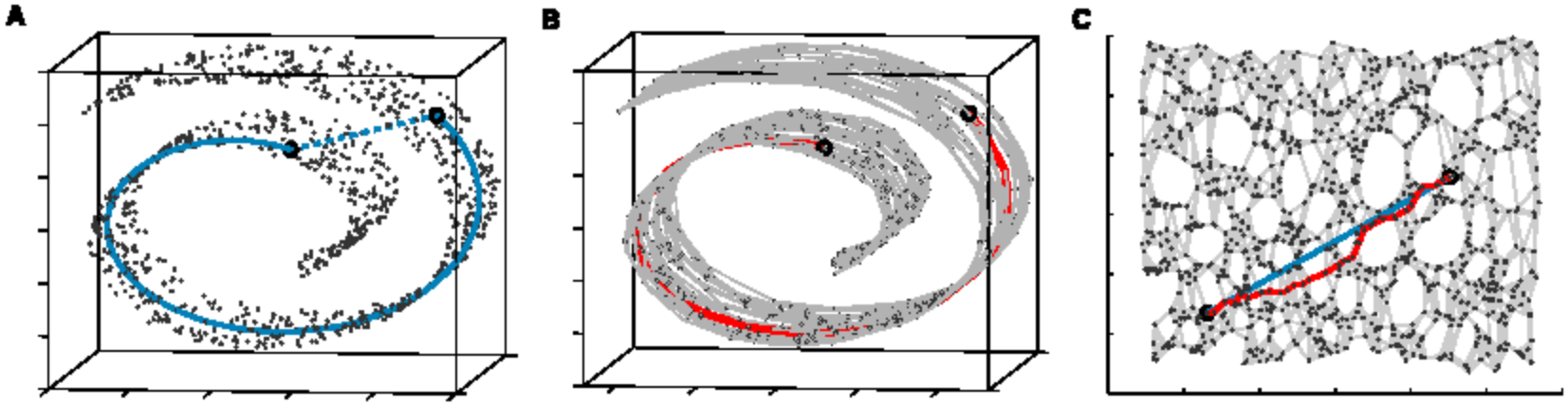
Maintenant D_G est une matrice de distances géodésiques de deux points arbitraires le long de la varieté M (figure B)



[Tenebaum et al. 2000]

Utilisation de MDS pour representer le graphe en R^d

Trouver un espace euclidien Y à d-dimensions qui préserve les distances par paires (Figure C)

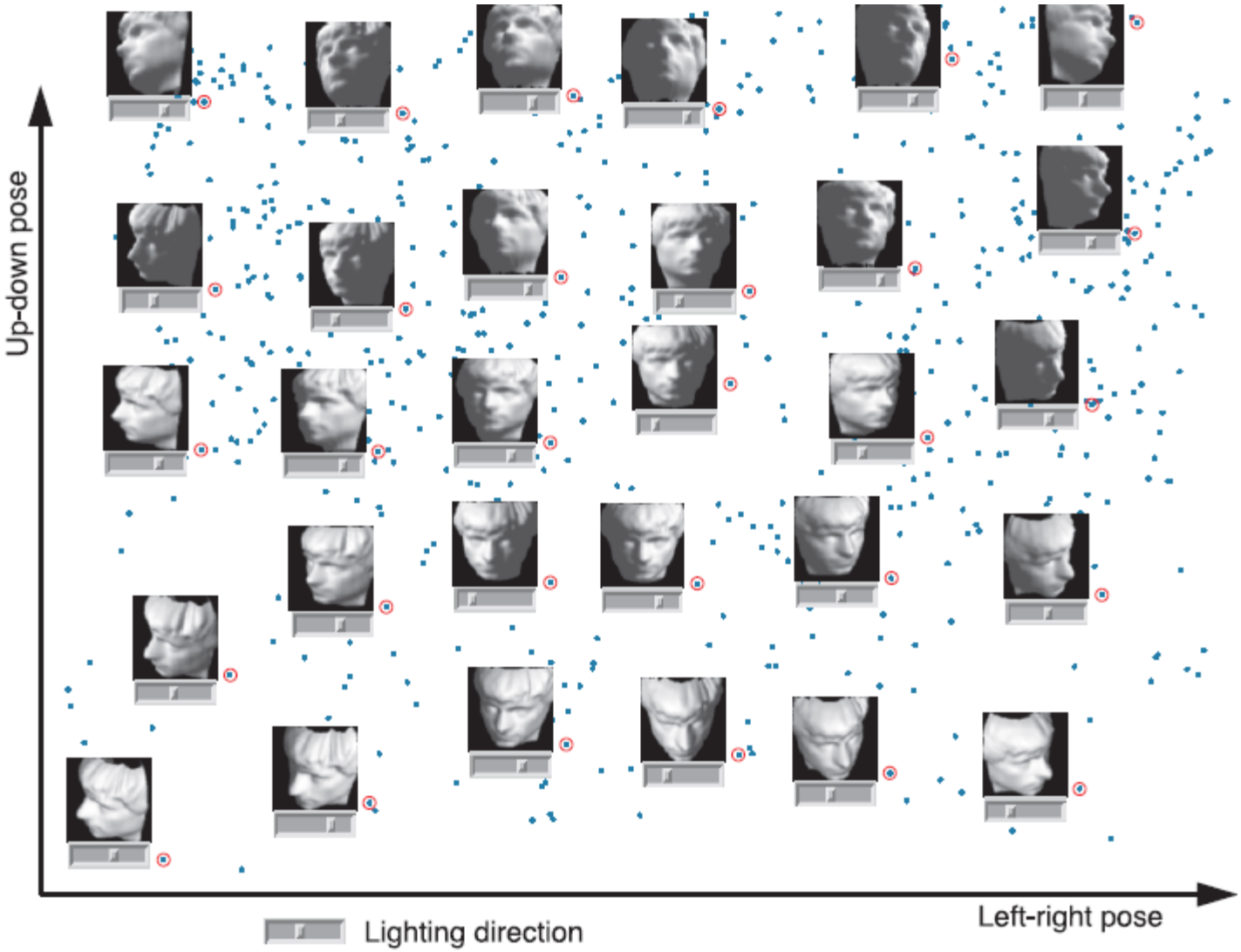


[Tenebaum et al. 2000]

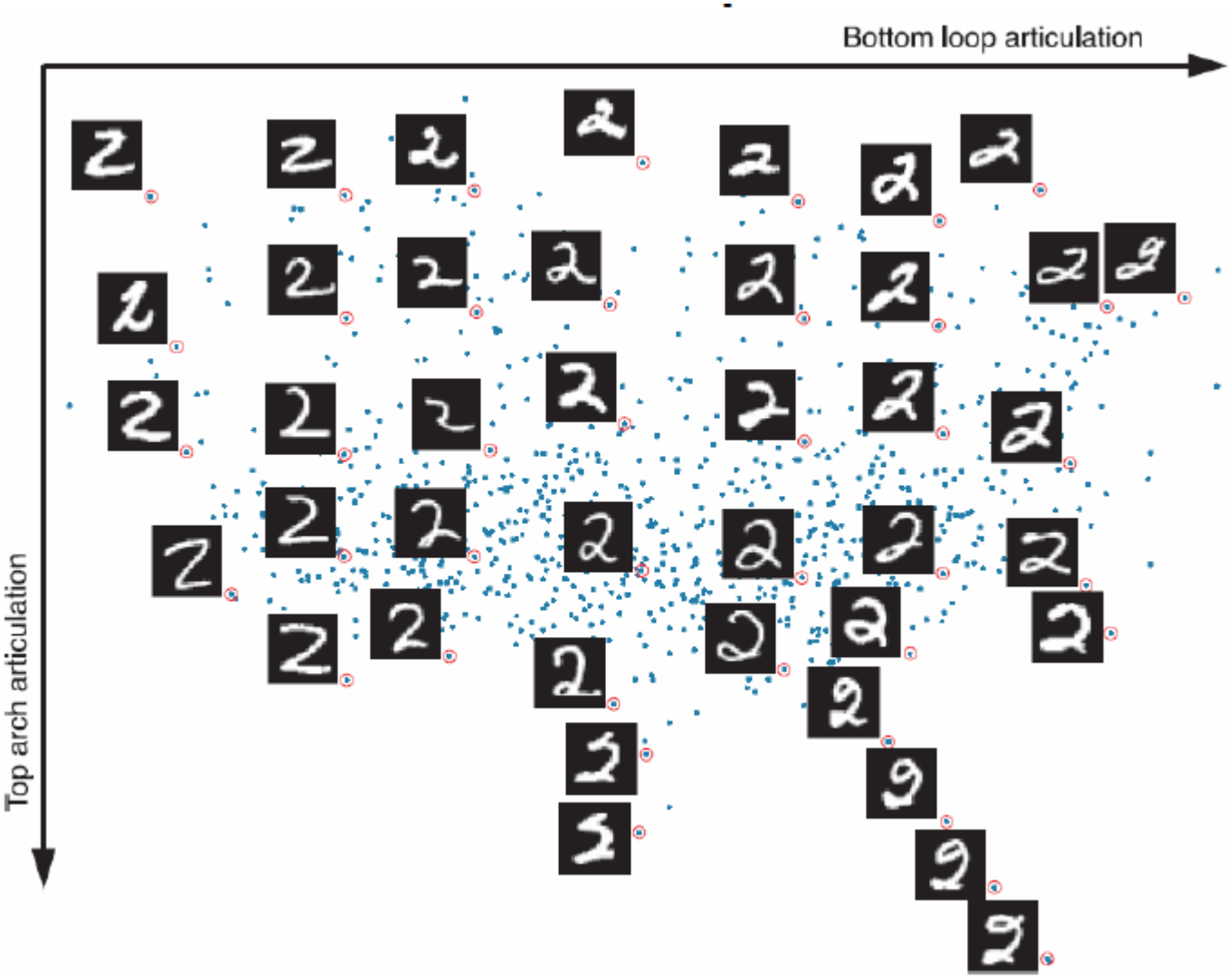
Exemple sur les images

Pour chaque image on a 64x64 = 4096 pixels

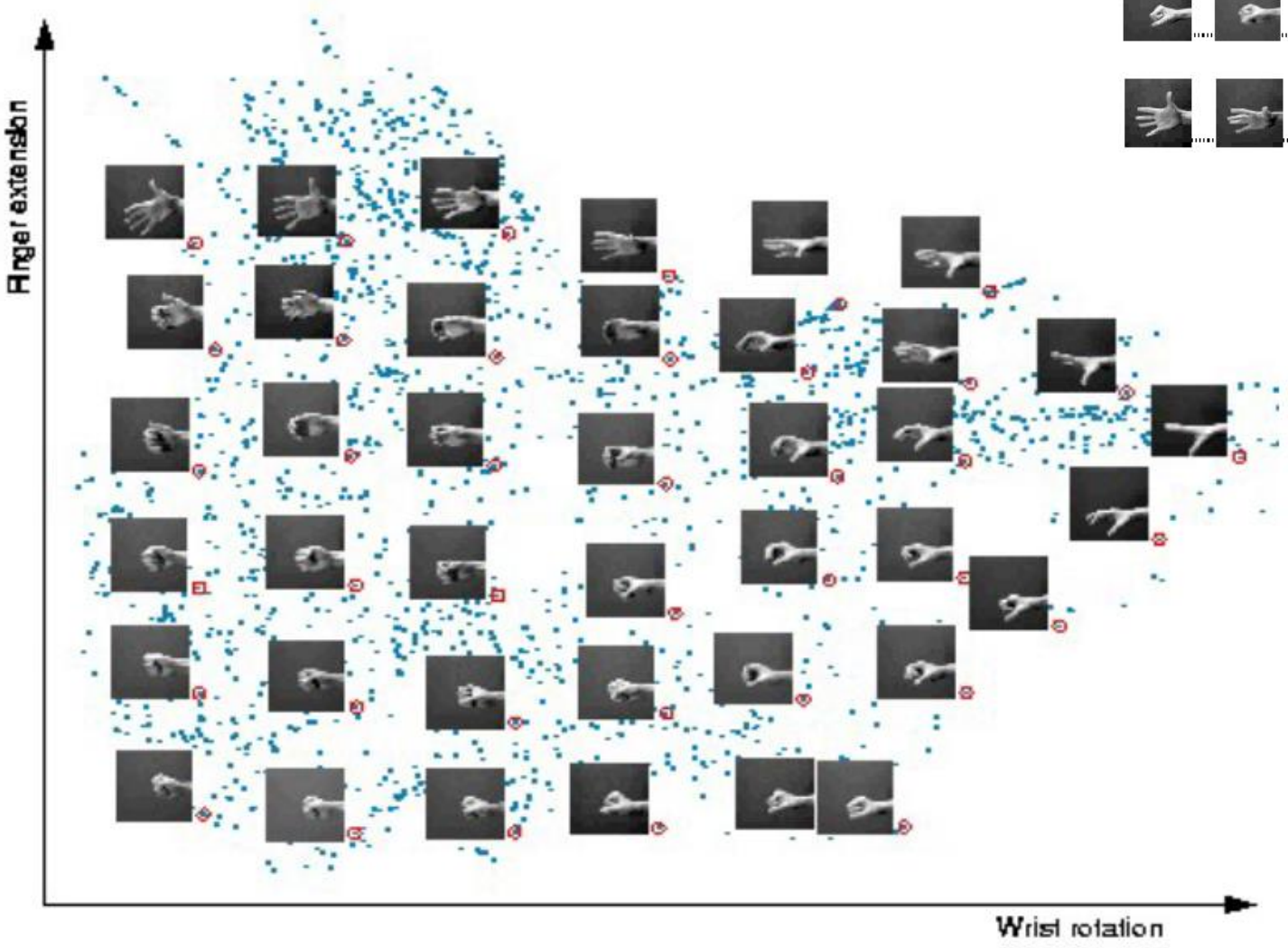
Résultats



Résultats



Résultats



Conclusions

- Avantages
  - Non-linéaire
  - Non-itérative
  - Préserve les propriétés globale des données
- Désavantages
  - Sensible aux bruits
  - Paramètres à fixer : k ou ε
  - Assez lent pour des grands jeux de données
  - k doit être élevé pour éviter les " raccourcis linéaires" près des régions de forte courbure de la surface

Locally Linear Embedding (LLE)

Réduction des dimensions par extraction de caractéristiques

Deux grandes familles de méthodes :

- **Méthodes linéaires**
  - Analyse en Composantes Principales (ACP)
  - Analyse Discriminante Linéaire (ADL)
  - Multi-Dimensional Scaling (MDS)
  - …
- **Méthodes non-linéaires**
  - Isometric feature mapping (Isomap)
  - → Locally Linear Embedding (LLE)
  - Kernel PCA
  - Segmentation spectrale (spectral clustering)
  - Methodes supervisées (S-Isomap)
  - …

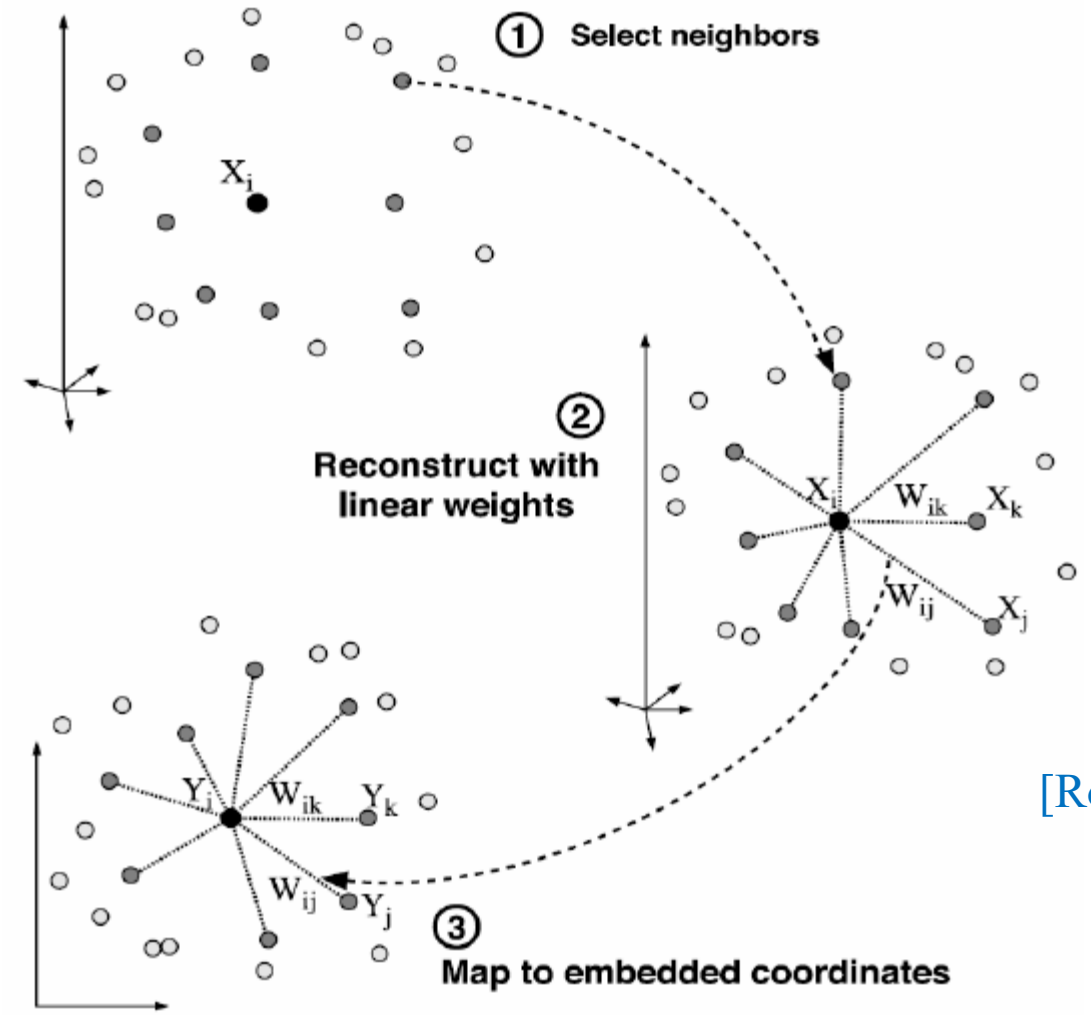
Locally Linear Embedding (LLE)

- LLE («plongement localement linéaire») aborde le même problème que ISOMAP par une voie différente.

- LLE préserve les propriétés locales des données en représentant chaque point par une combinaison linéaire de ses plus proches voisins.

- LLE construit une projection vers un espace linéaire de faible dimension préservant le voisinage.

Algorithme LLE



[Roweis and Saul 2000]

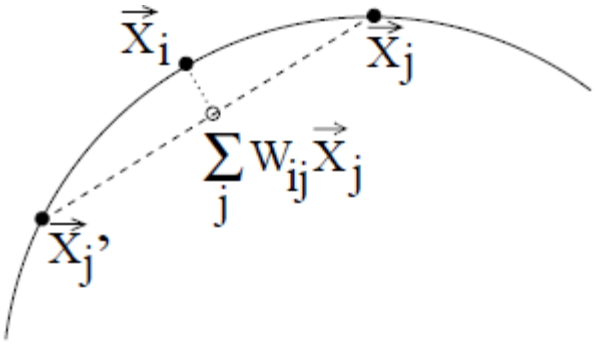
Algorithme LLE

LLE utilise 3 étapes:

- Calcule les k plus proches voisins

- Calcule les poids nécessaires pour reconstruire chaque point utilisant une combinaison linéaire des ses voisins

- Projette les résultats selon les nouvelles coordonnées trouvées.



Algorithme LLE

- La géométrie locale est modelée par des poids linéaires qui reconstruisent chaque point par une combinaison linéaire de ses voisins
- Les erreurs de reconstruction sont mesurées selon cette fonction de coût :

$$\varepsilon(W) = \sum_{i=1}^{N} \left| X_i - \sum_j W_{ij} X_j \right|^2$$

  - Où les poids W_{ij} le mesurent la contribution du j-ième exemple à la construction du i-ième exemple
- Les poids sont minimisés selon deux contraintes :
  1) Chaque point est reconstruit seulement par ces voisins
  2) $\sum_j W_{ij} = 1$

Algorithme LLE

- On cherche les coordonnées Y_i de d-dimension qui minimisent la fonction de coût suivante:

$$\phi(Y) = \sum_{i=1}^{N} \left| Y_i - \sum_j W_{ij} Y_j \right|^2$$

Estimation des paramètres

- On considère un échantillon x avec k plus proches voisins η_j et les poids reconstruits w_j (dont la somme est égale à 1). On peut trouver ces poids en 3 étapes :
  - Etape 1 : On calcule la matrice de corrélation de voisinage C_{jk} et son inverse C^{-1}

$$C_{jk} = \eta_j^T \eta_k$$

  - Etape 2 : On calcule le multiplicateur Langragien λ qui renforce la contrainte $\sum_j w_j = 1$

$$\lambda = \frac{1 - \sum_{jk} C^{-1}{}_{jk} (x^T \eta_k)}{\sum_{jk} C^{-1}{}_{jk}}$$

  - Etape 3 : Calculer les poids reconstruit de la manière suivante:

$$w_j = \sum_k C^{-1}{}_{jk} (x^T \eta_k + \lambda)$$

Estimation des paramètres

- On trouve les vecteurs Y_i en minimisant la fonction de coût suivante:

$$\phi(Y) = \sum_{i=1}^{N} \left| Y_i - \sum_j W_{ij} Y_j \right|^2$$

- Pour optimiser cette fonction on introduit 2 contraintes:

$$\sum_j Y_j = 0 \qquad \frac{1}{N} \sum_i Y_i Y_i^T = I$$

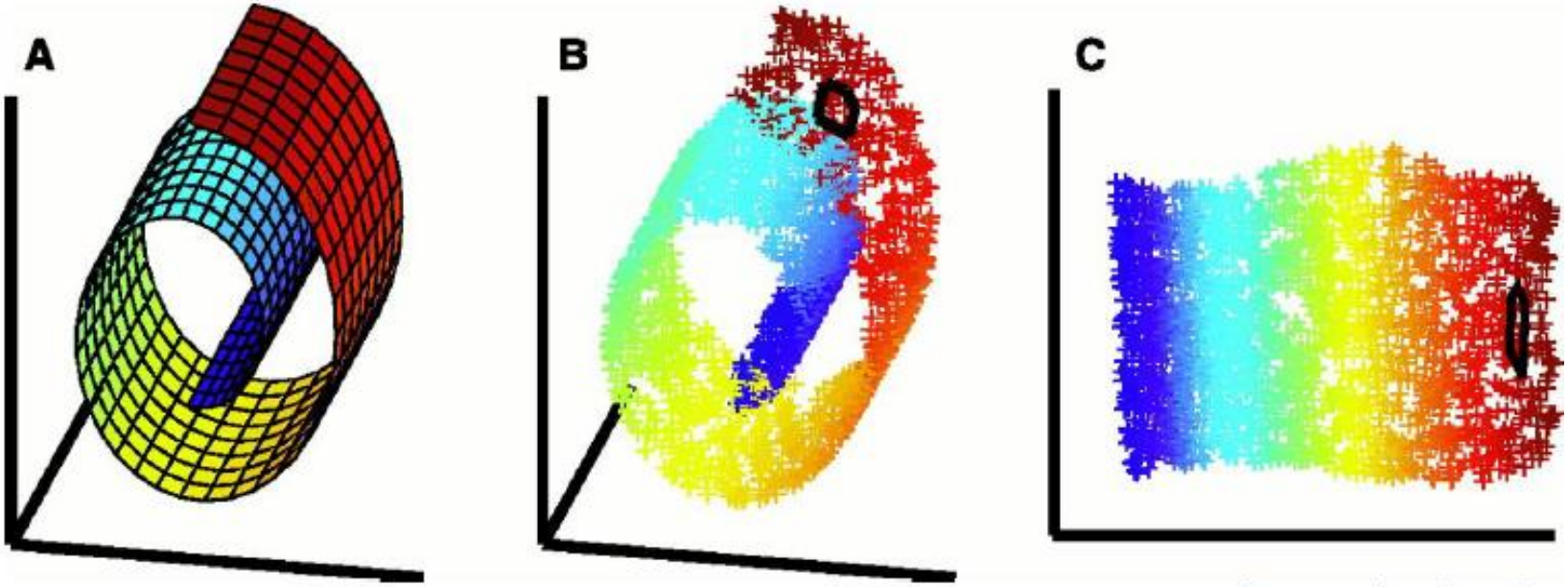
- Ce qui nous permet d'exprimer la fonction de coût de la manière suivante:

$$\phi(Y) = \sum_{ij} M_{ij} (Y_i^T Y_j)$$

- Où $M_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}$
- δ_{ij} est égal à 1 si i=j et est égal à 0 sinon.

- On retrouve la meilleure représentation en calculant les *d+1* vecteurs propres d'en bas de la matrice *M*

Exemples sur LLE



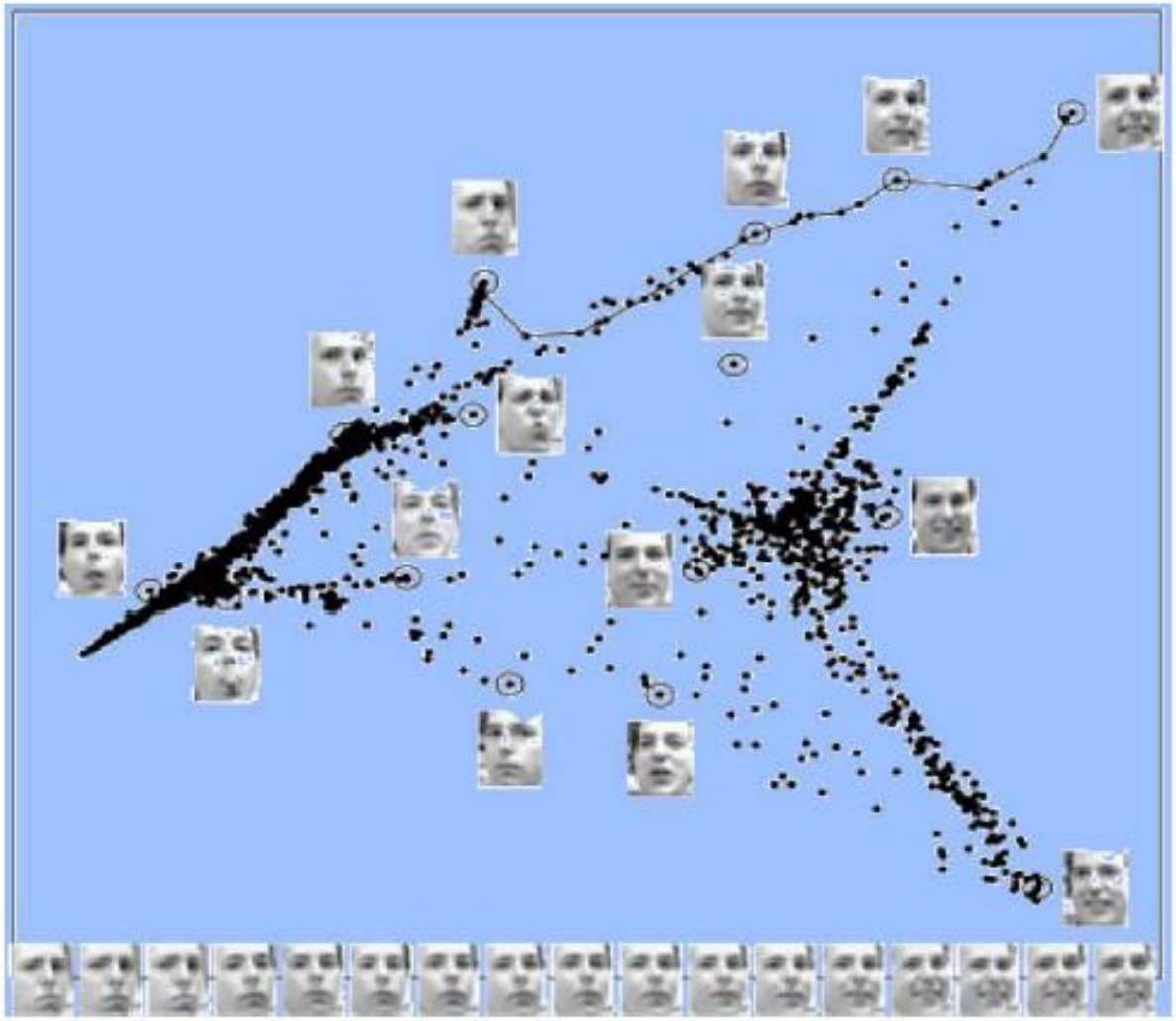
[Roweis and Saul, 2000]

Résultats

• Les points initiaux représentent des images de visages.

• Dans l'espace de 2 dimensions, ces images sont regroupées selon la position, l'éclairage et l'expression.

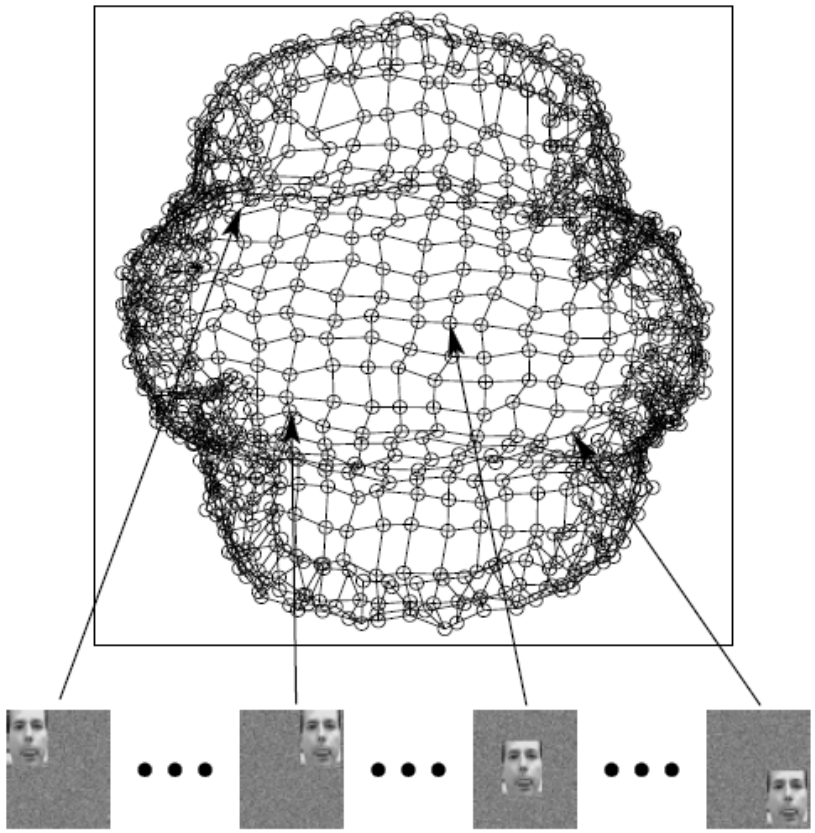
• Les images placées en bas de la figure correspondent aux points successifs rencontrées sur la ligne en haut à droite, balayant un continuum d'expression du visage.

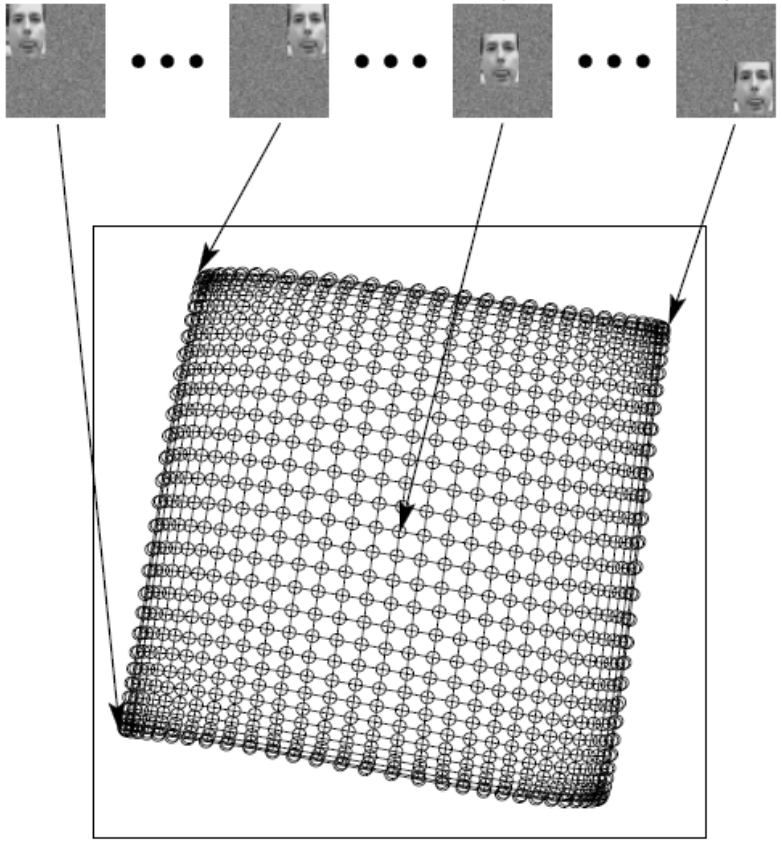


[Roweis and Saul 2000]

Résultats

- ACP vs LLE



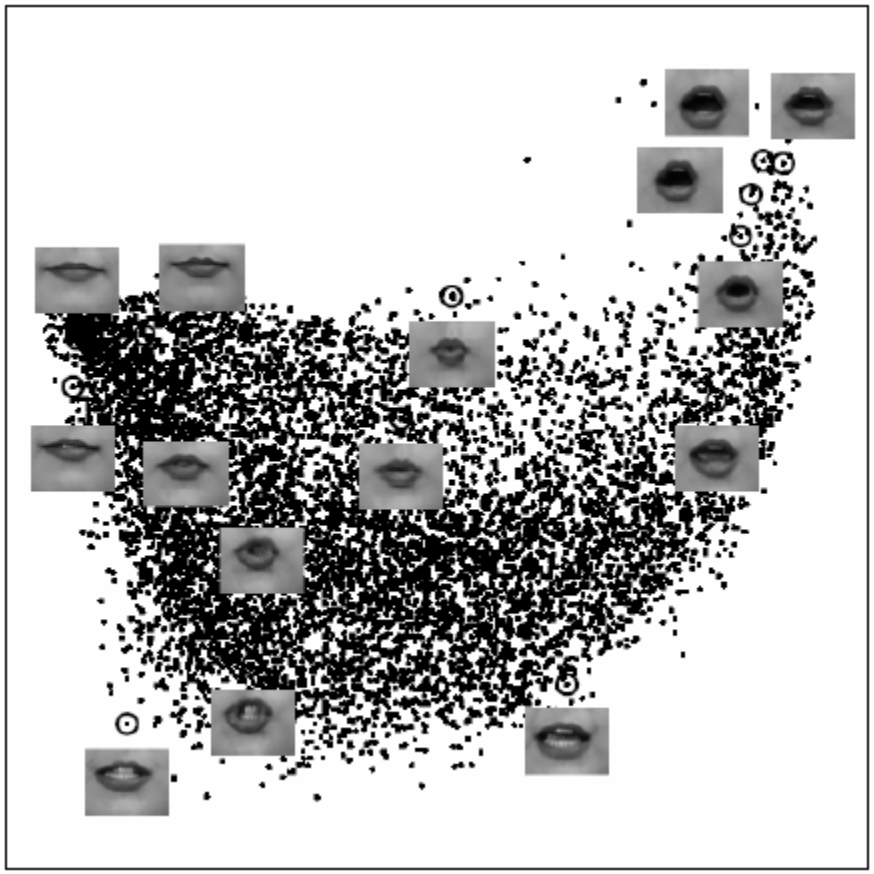


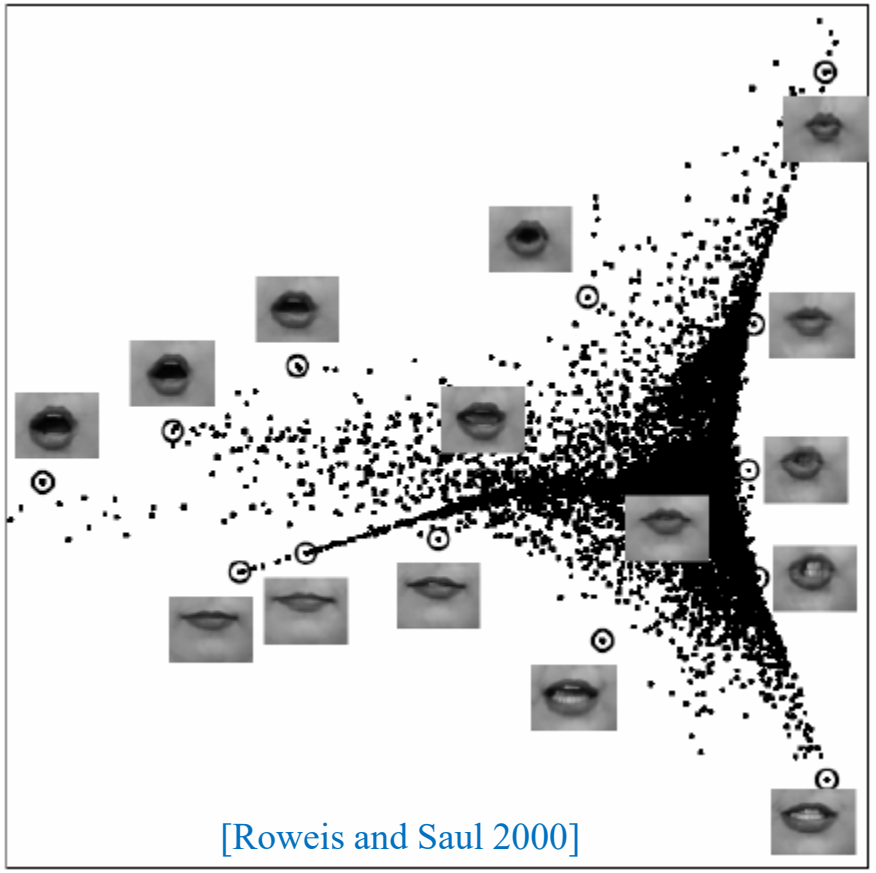
[Roweis and Saul 2000]

Contrairement à l'ACP, LLE préserve une topologie locale !

Résultats

- ACP vs LLE





[Roweis and Saul 2000]

Conclusions

- LLE est une technique non-linéaire qui préserve les propriétés locales des données en représentant chaque point par une combinaison linéaire de ses plus proches voisins dans un nouvel espace de dimensions réduite

- LLE utilise 3 étapes:
  - Calcule les k plus proches voisins
  - Calcule les poids nécessaires pour reconstruire chaque point utilisant une combinaison linéaire des ses voisins
  - Projette les résultats selon les nouvelles coordonnées trouvées.

Conclusions

- Sensible aux bruits
- Paramètres à fixer : k
- Assez lent pour des grands jeux de données

Références

- J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319-2323, 2000.
- Sam Roweis & Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323-2326, 2000.