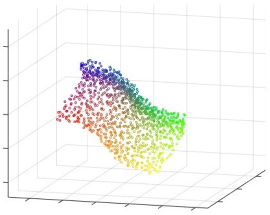
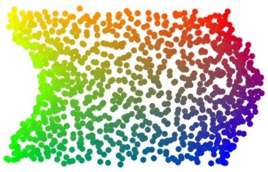
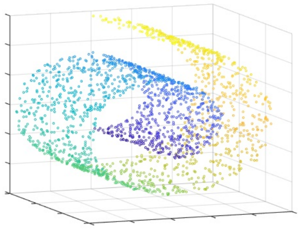
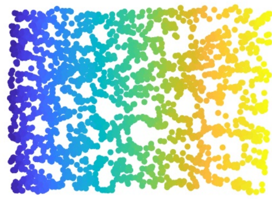
TP d'Introduction à l'Apprentissage Automatique
Exo 4: Réduction de dimension linéaire et non-linéaire

Télécom Physique Strasbourg - antoine.deleforge@inria.fr

Téléchargez les fichiers du TP (Matlab) en suivant ce lien : `members.loria.fr/ADeleforge/files/TP_ML_Exo4_TPS.zip`.

Partie I : Analyse en Composantes Principales sur Tapis Volant

1) Dans un nouveau script Matlab, générez un jeu `flyingcarpet(0,0)` à l'aide de la fonction fournie

```
[data, colors] = dataset_flyingcarpet(0,0);
```

puis visualisez-le à l'aide de :

```
figure(1); movegui('northwest');
scatter3(data(:,1),data(:,2),data(:,3),5,colors);
axis equal; axis([-4,4,-4,4,-4,4]);
```

Lancez le script plusieurs fois pour comprendre à quoi ressemblent ces données. Faites varier individuellement chacun des deux paramètres de la fonction (`noise` et `curviness`) entre 0 et 10 tout en maintenant l'autre à zéro pour visualiser à quoi ils correspondent.

Note : La coloration des points données dans `colors` est arbitraire et n'est là que pour faciliter la visualisation des points voisins. Elle ne sera pas utilisée par les algorithmes qui suivent.

2) On note $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N\times 3}$ le jeu de données `data`. Soustrayez sa moyenne $\boldsymbol{\mu} \in \mathbb{R}^3$ pour obtenir le jeu centré $\mathbf{X}_0$ et calculez sa matrice de covariance empirique $\mathbf{C} = \frac{1}{N-1}\mathbf{X}_0^\top\mathbf{X}_0 \in \mathbb{R}^{3\times 3}$. Calculez les 3 valeurs propres $\lambda_1 \geq \lambda_2 \geq \lambda_3$ de $\mathbf{C}$ en ordre décroissant grâce à la fonction Matlab `lambda=eigs(C,3)`. Pour `noise=curviness=0`, que pouvez-vous dire de la plus petite valeur propre λ_3 ? Comment interprétez-vous le ratio λ_1/λ_2 ? Faites varier `noise` et `curviness`. Quelle est leur influence sur le ratio λ_2/λ_3 ?

3) Effectuez une Analyse en Composantes Principales (ACP) sur un jeu de données `flyingcarpet(0,0)`, et visualisez ses deux composantes principales. Pour cela, calculez les 3 vecteurs propres $\mathbf{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3] \in \mathbb{R}^{3\times 3}$ associés à λ_1, λ_2 et λ_3 en utilisant la fonction `[V,~] = eigs(C,3)`, puis multipliez à droite le jeu de données centré $\mathbf{X}_0$ par la matrice $\mathbf{V}_2 = [\boldsymbol{v}_1, \boldsymbol{v}_2] \in \mathbb{R}^{3\times 2}$. Affichez le jeu de données 2D `data_red` résultant avec la coloration du jeu d'origine en utilisant :

```
figure(2); movegui('northeast');
scatter(data_red(:,1),data_red(:,2),50,colors,'filled');
```

Qu'observez-vous ? Faites ensuite varier indépendemment les paramètres `noise` et `curviness` pour comprendre leur influence sur la représentation 2D obtenue. Quelle limite de l'ACP le paramètre `curviness` met-il en avant ?

4) Reprojetez maintenant la représentation 2D vers l'espace 3D d'origine, en la multipliant à droite par $\mathbf{V}_2^\top$ et en rajoutant $\boldsymbol{\mu}$. Visualisez le résultat à l'aide de :

```
figure(3); movegui('southwest');
scatter3(data_rec(:,1),data_rec(:,2),data_rec(:,3),5,colors);
axis equal; axis([-4,4,-4,4,-4,4]);
```

Comparez le jeu ainsi reconstruit à l'original. Quelle est l'influence des paramètres `noise` et `curviness` sur la qualité de reconstruction ? Que se passe-t-il lorsqu'ils valent tous deux 0 ? Que se passe-t-il si vous utilisez les colonnes 2 et 3 de la matrice $\mathbf{V}$ à la place ? Pourquoi ?

Partie II : Déroulage Non-linéaire du Rouleau Suisse

Pour la suite du TP, nous aurons besoin de la Dimensionality Reduction Toolbox (`drtoolbox`) de Laurens van der Maaten [1]. Ajoutez la toolbox à votre path Matlab en utilisant `addpath(genpath('drtoolbox'));`

5) Dans un nouveau script, générez et visualisez un "Swiss Roll" dataset de $N = 2000$ points, grâce à :

```
[data, ~, t] = generate_data('swiss', N, noise);
colors = t(:,1);
figure(1); movegui('northwest');
scatter3(data(:,1),data(:,2),data(:,3),5,colors);
```

Faites varier le paramètre de bruit entre 0 et 10 pour voir son influence. Visualisez les deux composantes principales du jeu de données à l'aide d'une ACP puis reconstruisez-le, comme en **3)** et **4)**. Qu'observez vous et pourquoi ? Que dire des 3 valeurs propres de la matrice de covariance ?

6) Remplacez maintenant l'ACP par l'algorithme Local Tangent Space Alignment (LTSA) de la toolbox pour obtenir une représentation en $d = 2$ dimensions du rouleau suisse (avec `noise=0.1`) en utilisant `data_red = ltsa(data,d,k)`. Le paramètre k correspond au nombre de "plus proches voisins" utilisés par l'algorithme LTSA pour calculer des ACP locales, qui sont ensuite recollées. Essayez avec $k = 5, 10, 20, 40$. Quel est son impact sur la stabilité de l'algorithme et le temps de calcul ? Quel choix de k permet de mieux "dérouler" le rouleau suisse ? Comment expliquez vous le comportement pour des k trop grands ou trop petits ?

7) Répétez la question **6)** mais en utilisant le niveau de bruit `noise = 0.5`. Essayez de trouver "à la main" une bonne valeur de k.

8) Remplacez maintenant l'algorithme LTSA par Isomap [2] à l'aide de `data_isomap = isomap(data, d,k)`, en utilisant $k = 8$. Que dire de la qualité de la représentation ? Du temps de calcul ?

Partie III : Back to MNIST

9) Dans un nouveau script, charger 2000 échantillons du jeu de données MNIST en utilisant :

```
MNIST = load('mnist_test.csv');
labels = MNIST(1:2000,1);
data = MNIST(1:2000,2:end);
```

Calculer les 3 composantes principales du jeu de données à l'aide d'une ACP puis visualisez les en colorant les points en fonction des chiffres à l'aide de

```
fig=figure; clf(fig);
scatter3(data_red(:,1),data_red(:,2),data_red(:,3),5,labels);
colormap jet;
```

Qu'observez vous ? Même question avec Isomap [3] ($d = 3, k = 8$).

10) Exécutez le script `TP_DIMRED_MNIST_clustering.m` fourni. Combien de temps prennent les 10 itérations de k-means++ et les 5 itérations de GMM EM sur MNIST ?

11) Tracez les 784 valeurs propres ordonnées $\boldsymbol{\lambda}$ de la matrice de covariance de MNIST à l'aide de `plot`. En déduire un nombre d'axe principaux d suffisant pour conserver 90% de la variance du jeu de données.

12) Effectuez une analyse en composante principale sur MNIST pour réduire sa dimension à d, et faites maintenant tourner k-means++ et GMM EM sur le jeu réduit. Pensez à re-projeter les centroïdes obtenus dans l'espace d'origine avant de les visualiser, en utilisant le code de la question **4)**. Quel est l'effet sur les temps de calcul ? Observez-vous un effet notable sur la qualité des centroïdes obtenus ? Qu'en est-il en réduisant la dimension à 10 ou moins ? Profitez de cette accélération pour faire tourner GMM EM pendant 500 itérations supplémentaires. Parvenez-vous à reconnaître plus de chiffres ?

13) Visualisez les axes principaux de MNIST sous forme d'images.

1. Plus d'info ici : `https://lvdmaaten.github.io/drtoolbox/`
2. Si vous éprouvez des difficultés à exécuter isomap, veuillez charger les données avec `load('saved_isomap1.mat');`
3. Idem avec `load('saved_isomap2.mat');`