

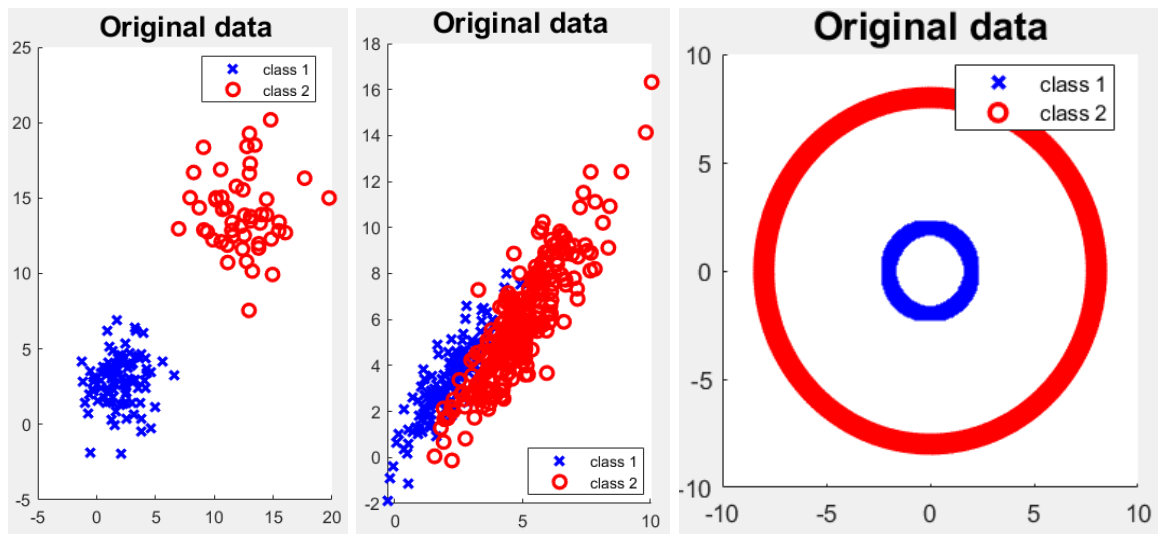
TP Unsupervised Learning Compte-rendu

Antoine Moulin

1 Data

2 Toy examples

La fonction `Toy_main.m` applique trois méthodes différentes (PCA, K-PCA & K-means) à trois situations différentes : deux gaussiennes séparées, deux gaussiennes qui se recoupent et enfin le cas où les données sont distribuées selon deux cercles concentriques :



Dans le cas des deux gaussiennes séparées, les trois méthodes sont efficaces, il n'y a pas vraiment de difficulté à classer les points. Comme il s'agit de gaussiennes, la méthode PCA est tout à fait pertinente. On remarque d'ailleurs que la première composante principale suffit souvent à déterminer la classe d'un point. L'utilisation du kernel-trick n'est ici pas nécessaire. Concernant le clustering, il donne de bons résultats également.

Dans le cas de gaussiennes qui se recoupent, la K-PCA n'est pas vraiment efficace contrairement à la PCA qui semble donner de meilleurs résultats.

Enfin, concernant les distributions en cercles, la PCA ne fonctionne pas du tout. Cela vient du fait que les deux classes ne sont pas linéairement séparables. C'est pour cela qu'on utilise le kernel-trick, qui permet de plonger les données dans un espace plus grand, dans lequel les données sont séparables. C'est cette méthode qui donne les meilleurs résultats. Concernant le K-means, comme deux points de classes différentes peuvent être plus proches que deux points d'une même classe, il ne peut pas fonctionner non plus puisque qu'il fait en sorte de minimiser les distances intra-clusters et de maximiser les distances inter-clusters.

3 Face recognition

3.1 Goal

3.2 PCA

1. D'un point de vue mathématique, la formule de la matrice de covariance donnée dans le cours ne fonctionnerait pas si les données n'étaient pas centrées. Mais ça ne répond pas vraiment à

la question. Il est nécessaire de centrer les données avant une PCA car il s'agit d'un procédé qui projette dans les directions qui maximisent la variance (formules dans le cours). De ce fait, si une feature a des valeurs bien plus élevées que les autres, elle prédominera et sera donc considérée comme une composante principale. Pour juger chaque composante de la même façon, on centre donc les données (et parfois il est préférable de les réduire également).

2. Calculons $x_p x_q^T$. U est une matrice orthogonale, donc $UU^T = I$, puis $x_p = y_p U^T, x_q = y_q U^T$. Ainsi, on a

$$x_p x_q^T = y_p U^T (y_q U^T)^T = y_p U^T U y_q^T = y_p y_q^T$$

i.e.

$$\boxed{x_p x_q^T = y_p y_q^T}$$

3. La matrice de covariance

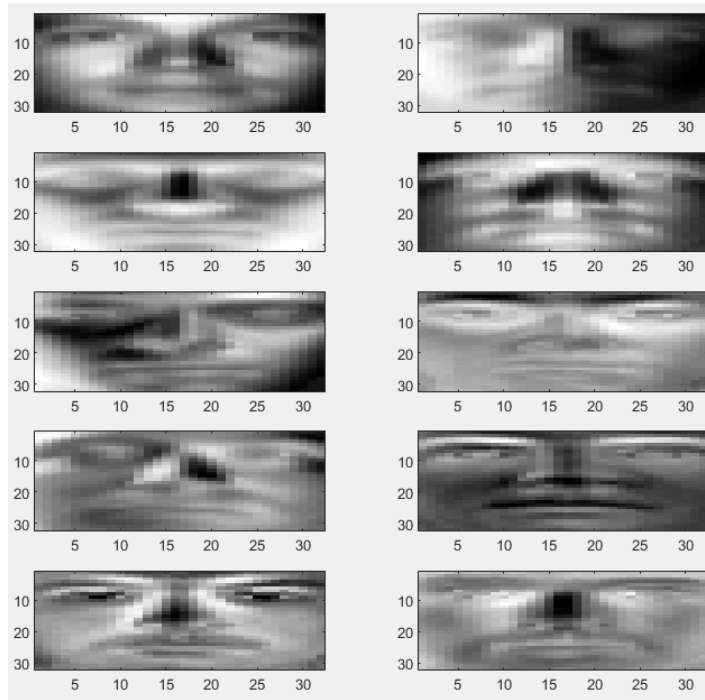
4. La matrice de covariance de Y est donnée par

$$\Sigma_Y = \frac{1}{N-1} (XU)^T XU = \frac{1}{N-1} U^T X^T XU = U^T \left(\frac{1}{N-1} X^T X \right) U = U^T C U$$

Or $C = U D U^T$, donc $U^T C U = D$ car U est une matrice orthogonale. Ainsi, on a bien :

$$\boxed{\Sigma_Y = D}$$

5. On exécute la section PCA de la fonction `Face_main.m` et on trouve :



Les visages propres ont la forme d'un véritable visage, avec des yeux, un nez, une bouche etc. En ce sens, les visages semblent réels. Il y a juste les niveaux de gris qui ne sont pas réalistes (exemple : le visage 8 dont les sourcils sont blancs avec le reste du visage très sombre) mais certaines images pourraient faire penser à des images de vrais visages qui ont été filtrés.

6. De façon générale, on sait que le produit vectoriel entre deux vecteurs \mathbf{u} et \mathbf{v} est donné par

$$\langle \mathbf{u}, \mathbf{v} \rangle = \|\mathbf{u}\| \cdot \|\mathbf{v}\| \cdot \cos((\mathbf{u}, \mathbf{v}))$$

Ainsi, comme les deux vecteurs sont non nuls, on peut diviser par les normes et calculer l'angle :

$$\boxed{\cos((\mathbf{u}, \mathbf{v})) = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}}$$

3.3 K-PCA

1. On ne peut pas afficher les vecteurs $(\alpha_i)_i$ comme avec la PCA car les vecteurs $(\alpha_i)_i$ n'appartiennent pas au même espace que les données de départ (ils appartiennent à un espace de dimension plus grande).
2. En utilisant le noyau gaussien (et $\sigma = 13$), on obtient les performances suivantes :

```
Percentage of correct answer in KPCA is 78.2101
Elapsed time is 4.539010 seconds.
```

Avec le noyau polynomial, en prenant l'exposant égal à 4 (semble être un bon exposant par rapport aux différentes valeurs testées), on obtient cette fois :

```
Percentage of correct answer in KPCA is 74.5136
Elapsed time is 4.696663 seconds.
```

Ainsi, ce nouveau noyau semble un peu moins performant mais les résultats restent similaires.

3.4 Performance

1. Voici les résultats obtenus avec une PCA :

```
Percentage of correct answer in PCA is 77.4319
Elapsed time is 0.712788 seconds.
```

On constate donc que le PCA a une bien meilleure complexité que le K-PCA. Sous certains paramètres, on peut obtenir de meilleurs résultats avec K-PCA, mais cette méthode est plus longue. Dans certains cas assez complexes (non linéairement séparables), il peut être préférable d'utiliser K-PCA comme avec l'exemple du cours. Autrement, la PCA semble être une bonne solution également.

2. Bien que la méthode avec les intensités donne d'aussi bons résultats, les méthodes précédentes ont l'avantage de supprimer la redondance des données.

3.5 ICA and NNMF

4 Skin lesion segmentation

4.1 Goal

1. A priori, il faut choisir deux classes : une pour la lésion et une pour le fond (la peau). Néanmoins, avec la présence d'autres éléments comme des poils, il peut être préférable de choisir plus de classes (par exemple 3) afin d'avoir une meilleure segmentation (car avec trois classes on aurait potentiellement une classe pour la peau, une pour la lésion et une pour les poils).
2. On utilise la troisième image. Avec trois classes on obtient, pour les différents labels, les similarités suivantes :

```
Similarity for the label 1 : 0.059566
Similarity for the label 2 : 0.68501
Similarity for the label 3 : 0.57812
```

3. Pour choisir la classe automatiquement, on pourrait prendre celle qui maximise la similarité.
4. En utilisant le fait que

$$(x_i - \mu_k)^2 = \frac{1}{N_k} \sum_{j>i} z_{jk} (x_i - x_j)^2$$

On a :

$$\sum_{i=1}^N z_{ik} (x_i - \mu_k)^2 = \frac{1}{N_k} \sum_i z_{ik} z_{jk} \sum_{j>i} (x_i - x_j)^2$$

puis

$$\sum_{i=1}^N z_{ik}(x_i - \mu_k)^2 = \frac{1}{2N_k} \sum_{i=1}^N \sum_{j \neq i} z_{ik} z_{jk} (x_i - x_j)^2$$

5. Une idée serait de calculer la moyenne puis de prendre l'observation la plus proche de la moyenne. Une autre méthode, peut-être un peu plus complexe utiliserait le fait suivant. Soit $A = \{a_1, \dots, a_n\}$ des réels. On définit $\Lambda_A : x \mapsto \sum_{i=1}^n \log(1 + |x - a_i|)$. On peut montrer que le minimum est nécessairement atteint en une des observations $(a_i)_i$. En utilisant cette fonction au lieu de la norme euclidienne, on pourrait faire en sorte que la moyenne soit une des observations.