
Département Traitement du Signal et des Images

# SI 221 : ACP et k-moyennes

Laurence Likforman, Chloé Clavel, Giovanna Varni, Hamid Jalazai

24 Octobre 2017

### Site pédagogique du module SI221

Site pédagogique du module SI221, Onglets "Ressources en ligne", puis "TP ACP et k-moyennes".

# 1 Description des données

## 1.1 Données LANDSAT sur Tarascon

Ce sont les données dans BaseImages.zip du TP "Réseaux de neurones" et elles sont accessibles à partir du site pédagogique (en fichiers isolées ou en .zip). La taille du pixel est de 30m:

| | |
|---|---|
| 0.45-0.52 $\mu$ | `landsattarasconC1.ima` |
| 0.52-0.60 $\mu$ | `landsattarasconC2.ima` |
| 0.63-0.69 $\mu$ | `landsattarasconC3.ima` |
| 0.75-0.90 $\mu$ (Proche InfraRouge : PIR) | `landsattarasconC4.ima` |
| 2.08 2-35 $\mu$ (Moyen InfraRouge : MIR) | `landsattarasconC5.ima` |
| 10.40-12.50 $\mu$ (infra rouge thermique) | `landsattarasconC6.ima` |
| 1.55-1.75 $\mu$ (Moyen InfraRouge : MIR) | `landsattarasconC7.ima` |
| 0.52-0.90 $\mu$ (panchromatique) | `landsattarasconC8.ima` |

C'est sur ce jeu de données que vous commencerez vos travaux.

## 1.2 Données SPOT sur Tarascon

Dans le dossier Spot.zip, vous trouverez un jeu "SPOT XS" sur la Camargue : 3 canaux (XS1 : 0.50-0.59$\mu$, XS2 : 0.61-0.68$\mu$ et XS3 : 0.78-0.89$\mu$). La taille du pixel est de 20m. La zone est un peu réduite par rapport au jeu Landsat

Ces fichiers ont pour nom `cam1, cam2 et cam3`.

C'est sur ce jeu de données que sera validé le TP.

## 1.3 Données LANDSAT sur Kedougou

Un autre jeu de données LANDSAT (seulement 6 canaux) est disponible sur le site dans le répertoire *Landsat_Kedougou*.

6 canaux Landsat sont disponibles. Ils couvrent les longueurs d'ondes suivantes : 0.45-0.52$\mu$, 0.52-0.60$\mu$, 0.63-0.69$\mu$, 0.76-0.90$\mu$, 2.08-2.35$\mu$, 1.55-1.75$\mu$.

A la différence des données acquises sur Tarascon, il y a un problème d'acquisition de données inhérent au capteur (vous trouverez deux fichiers pour le canal 6: sans bruit et avec bruit).

# 2 Quelques ordres Matlab

## 2.1 Lecture sous Matlab des données SPOT et Landsat

Chaque canal est archivé séparément au standard de TSI (image 8 bits, extension `.ima`, associé à un fichier ASCII, extension `.dim` contenant le nombre de colonnes et le nombre de lignes).
Pour lire ce type de données sous Matlab, on dispose d'une procédure (`.m`) pour lire les images `ima2mat` dont la syntaxe, appliquée à la donnée `cam1` est la suivante :

```
im1 = ima2mat('cam1');
```

L'image est alors accessible en tant que tableau Matlab `im1`. Les dimensions de cette matrice peuvent se récupérer ainsi :

```
nlig = size(im1,1)
ncol = size(im1,2)
```

## 2.2 Visualisation d'une image

- Visualiser une image peut s'effectuer avec l'odre `image`

  ```
  image(im1);
  ```

  Mais ATTENTION : `im1` est une "image Matlab", c'est à dire entre 0 et 255.

- une image en niveau de gris doit utiliser la colormap `gray` :

  ```
  colormap(gray)
  ```

  Cette colormap par défaut est sur 100 niveaux de gris.
  Une image en 256 niveaux de gris nécessite une colormap sur 256 niveaux :

  ```
  colormap(gray(256))
  ```

  Si l'on veut se contenter de $nn$ niveaux :

  ```
  colormap(gray(nn))
  ```

- Une image de labels (indice de classe) peut se visualiser avec la colormap `flag` (4 couleurs)

  ```
  colormap(flag)
  ```

  ou `prism` (8 couleurs).

  ```
  colormap(prism)
  ```

**Attention**: pour Matlab, une image est un entier compris entre 0 et 255. Si ce n'est pas le cas (image quelconque en flottant), il faudra recentrer les données (voir 2.4).

### 2.3 Statistiques et histogramme d'une image

- La moyenne d'une image `im1` (matrice 2-D) est donnée par `mean(im1(:))`
- la valeur minimale est donnée par : `min(im1(:))`
- la valeur maximale par `max(im1(:))`.

L'histogramme d'une image (`im1`) peut se calculer directement avec `hist` :

```
hist(im1(:))
```

ou, sur 256 niveaux par pas de 1 entre 0 et 255 :

```
hist(im1(:),0:1:255)
```

Cela donne des résultats différents : méfiez vous de `hist` utilisé sans argument complémentaire et consultez le "help" en ligne.

### 2.4 Recadrage d'une image

Si on veut afficher une matrice quelconque $M$, il faut donc la transformer en "image matlab" $I$. Par exemple, on peut définir I comme suit :

```
I  = 255 * (M - min(M(:))) / (max(M(:)) - min(M(:))) ;
```

### 2.5 Création d'une fonction Matlab

Une fonction Matlab a la structure suivante:

```
function [out1, out2, ...] = myfonc(in1, in2, ...)

[out1, out2, ...] = %calcules

end
```

Elle va enregistrée dans un fichier `.m` ayant le meme nom de la fonction, dans ce cas le fichier s'appelera `yfonc.m`.

## 3 Analyse en composantes principales

### 3.1 Rappels sur l'ACP

Soient $N$ individus pour lesquels on dispose de $p$ caractéristiques (ce qui veut dire que son vecteur d'état est de dimension $p$). On notera $x_k^i$ la $k$ième caractéristique de l'individu $i$.

On note $M$ la matrice dans laquelle chaque ligne est constituée par un individu et chaque colonne représente une variable.
L'ACP est constituée par les étapes suivantes :

1. centrage et réduction des données ; soient $m_k$ et $\sigma_k$, les moyenne et écart-type de la $k$ième variable, on notera $x_k'^i = \frac{x_k^i - m_k}{\sigma_k}$ la donnée centrée réduite ;
2. calcul de la matrice de covariance des données centrées réduites ;
3. calcul des valeurs propres $\lambda_j$ et vecteurs propres $u_j$ de la matrice de covariance ;
4. vérification de l'ordre des vecteurs propres selon les valeurs propres croissantes;
5. calcul des composantes principales $x_q''^i$ exprimées dans la base des vecteurs propres :

$$x_q''^i = x'^i u_q$$

en notant $x'^i$ le vecteur ligne constitué par les observations de l'individu $i$.

On obtient donc de nouvelles variables constituées par des combinaisons linéaires des anciennes. Les composantes principales contiennent une quantité d'information proportionnelle à la valeur propre correspondante. On définit ainsi le pourcentage d'inertie par $\frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j}$.

## 3.2 Rappels Matlab

Vous aurez probablement besoin des fonctions suivantes :

- Pour transformer une matrice `A` de taille $P \times Q$ en vecteur `B` de taille $PQ \times 1$ : `B = A(:);` Cette commande place les colonnes de la matrice A les unes à la suite des autres dans un grand vecteur colonne.
- Inversement, pour récupérer la matrice `A` à partir du vecteur `B` (il faut bien sûr connaître la taille de `A` auparavant !) : `A = reshape(B,P,Q);`
- L'écart type se calcule avec la fonction `std` et la matrice de covariance avec la fonction `cov`.
- La fonction `eig` permet d'obtenir les valeurs propres et vecteurs propres d'une matrice (entrer `help eig` à l'invite du prompt Matlab pour en connaître l'usage). Attention également à l'ordre des valeurs propres...
- La fonction `brighten(b)` permet d'"éclaircir" ($0 < b \leq 1$) ou d'"assombrir" ($-1 \leq b < 0$) la table des couleurs.

### 3.3 Application aux composantes de LANDSAT sur Tarascon

Effectuez l'analyse en composantes principales sur les données LANDSAT. Pour cette analyse on considère chaque pixel de l'image comme un individu à classer : cet individu est décrit par son vecteur d'état (les composantes de ce vecteur sont tout simplement les valeurs des différents canaux). Affichez les images résultats.

Calculez le pourcentage d'inertie associé à chaque image et interprétez qualitativement les résultats : **ce sont ces valeurs qui valideront votre TP**.

Combien d'images faut-il garder pour conserver au moins 95% de l'information ?

## 4 Classification automatique : algorithme des K-moyennes

Le but de ce TP est d'écrire sous Matlab l'algorithme des k-moyenness.

Bien se rappeler les différentes étapes de cet algorithme :

1. initialisation des centres des classes,
2. affectation des pixels à une classe
3. recalcul des centres de gravités
4. test de fin : repartir sur l'étape 2 si test non vérifié

### 4.1 Code utile Matlab

Pour gagner du temps de code, il vous est proposé une procédure Matlab (`.m`) appelée `kmeans2` et disponiblesur le site pédagogique. Cette procédure effectue avec un code Matlab optimisé *l'étape de classification* : à chaque pixel est attribué une classe.

**ATTENTION : ce n'est absolument pas l'algorithme des k-moyennes, simplement une étape du processus de l'algorithme des k-moyennes (l'étape 2)**.

Il y a deux arguments à passer :

- Un tableau pour lequel chaque ligne correspond à un point physique au sol, le nombre de colonnes correspondant au nombre de canaux des données. Il y a autant de lignes que de pixels sur l'image.
- Un tableau de prototypes (les prototypes sont ici les points qui permettent d'initialiser les centres des classes) : chaque ligne correspond à un prototype, décrit par ses valeurs dans chaque canaux. Il y a autant de lignes que de classes dans l'image.

La procédure renvoie une image dont la valeur en chaque pixel est la classe correspondante. Si le tableau de prototype comporte 5 prototypes, cette valeur sera comprise entre 1 et 5.

Exemple sur les données SPOT de Camargue :

```
proto1 = [ 1 1 1];
proto2 = [255 255 255];

tabproto = [proto1; proto2;];


classe = kmeans2( tabimage, tabproto];

colormap(flag);
image(reshape( classe,512,512));
```

Si vous souhaitez avoir un tableau qui soit la fonction indicatrice de la classe $n$ (*i.e.* avec une valeur de 1 si le pixel appartient à la classe $n$, 0 autrement), il suffit d'utiliser directement un ordre Matlab qui teste une égalité sur un tableau.
Exemple pour la classe 3 :

```
indiclasse3 = classe==3 ;
```

Ceci est très utile pour calculer le nombre de pixel de chaque classe.

## 4.2 Landsat Tarascon

Dans cette partie, on souhaite réaliser un classificateur automatique pour réaliser une segmentation simple de ces images : par exemple, pour séparer terres et rivières (ou fleuve).

- Tracer l'histogramme canal par canal est souvent instructif (voir paragraphe 2.3).
- La recherche de prototypes nécessite un peu de patience : il faut rechercher sur l'image les coordonnées de points dont on a pu (ou su) voir l'originalité (par exemple, l'eau se définit par une valeur faible dans le canal 3), ce qui demande un peu de tatonnement. Pour connaitre la valeur du pixel (122,234) du tableau 2-D `A` il suffit de taper en entrée Matlab `A(122,234)`.

### Classification rivière-terre

Commencez par valider votre code en prenant un seul canal. Choissisez une valeur dans l'eau et une valeur sur la Terre. Testez votre algorithme de k-moyenne sur les cas suivants :

- classification en utilisant le canal 4 (fichier `landsattarasconC4.ima`)
- classification en utilisant le canal 1 (fichier `landsattarasconC1.ima`)

Que pensez vous des résultats de la classification sur le canal 1 ?

# 5 BONUS : ACP et k-moyennes sur de nouvelles données

## 5.1 Application aux composantes XS de SPOT

Effectuez l'analyse en composantes principales sur les données SPOT XS. Affichez les images résultats.

Calculez le pourcentage d'inertie associé à chaque image et interprétez qualitativement les résultats : **ce sont ces valeurs qui valideront votre TP**.

Combien d'images faut-il garder pour conserver au moins 90% de l'information ?

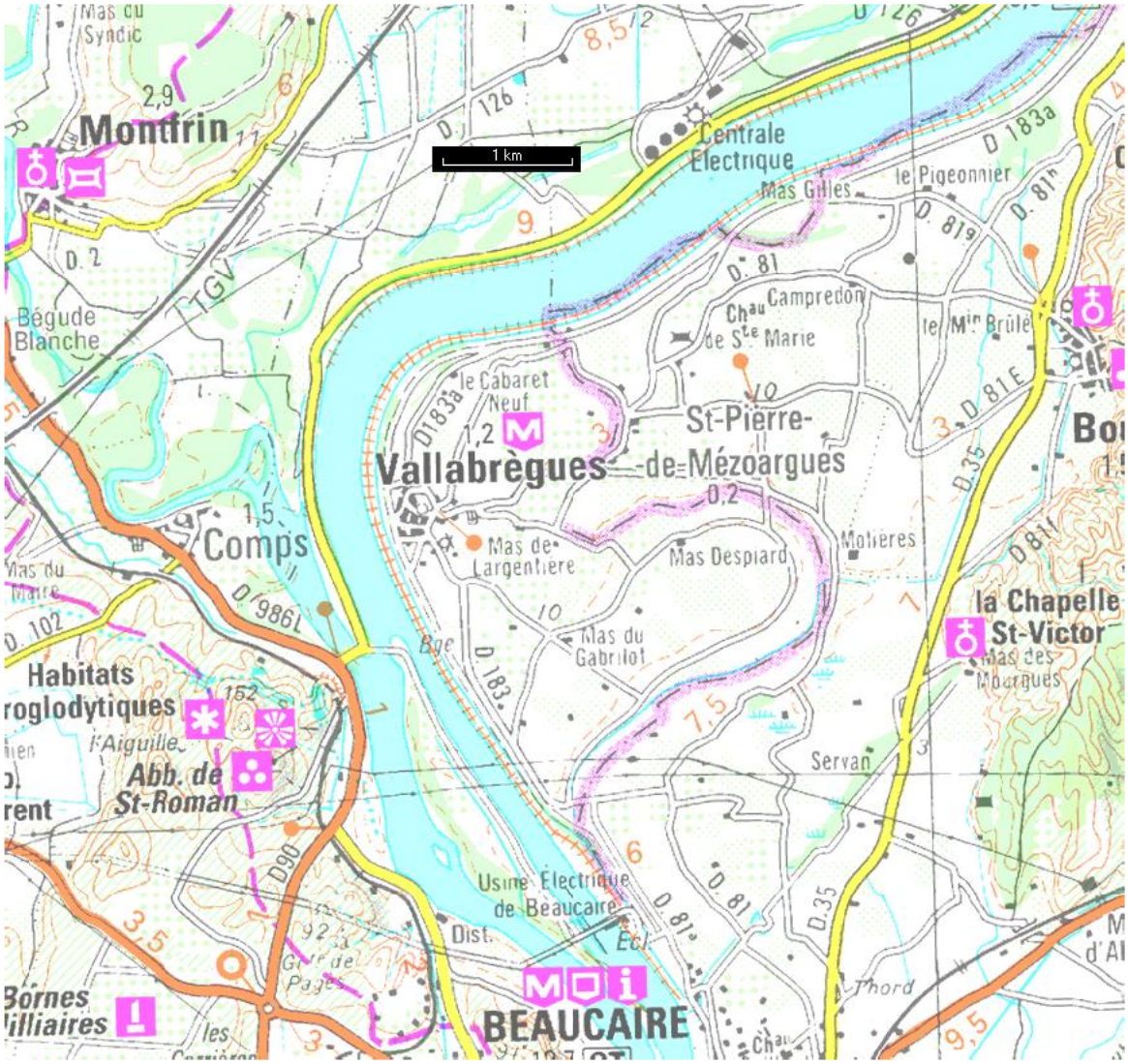
## 5.2 Application aux composantes de LANDSAT sur Kedougou

Effectuer une ACP sur ce jeu de données. Etudier soigneusement les images correspondant aux valeurs propres de faible valeur. Donnez votre avis sur les effets liés au capteur.

## 5.3 k-moyennes sur SPOT et LANDSAT-Kedougou

A partir du code des k-moyennes précédent, et en prenant un vecteur d'état à 3 ou 6 dimensions, proposez une classification automatique en 5 classes. Donnez alors la proportion de pixels dans l'image correspondant aux 5 classes que vous proposez.

En particulier, dans quel cas votre système est-il "bon" pour classifier correctement les rivières ?


Mas du Syndic
Montfrin
1 km
Centrale Électrique
Mas Gilles
le Pigeonnier
TGV
Bégude Blanche
Ch. Campredon de Ste Marie
le Min Brûlé
le Cabaret Neuf
Vallabrègues
St-Pierre-de-Mézoargues
Comps
Mas de Largentière
Mas Despiard
Mas du Gabriot
la Chapelle St-Victor
Mas des Mourgues
Habitats troglodytiques
l'Aiguille
Abb. de St-Roman
Usine Électrique de Beaucaire
Dist.
Bornes Milliaires
BEAUCAIRE