
Bases du traitement des images



Classification d'image



Dufrenois Franck
LISIC

Université du littoral

Classification d'images

Introduction

- **Objectif:** partitionner l'image en groupe de pixels ayant des caractéristiques communes:
 - Couleurs
 - Textures
 - Formes
- **Un cadre mathématique basé sur:** mab
Lois de probabilité - la distance - la densité
- **Avons-nous des informations au préalable?**
 - Classification supervisée
 - Classification non supervisée

Classification Bayesienne

Ingrédients?

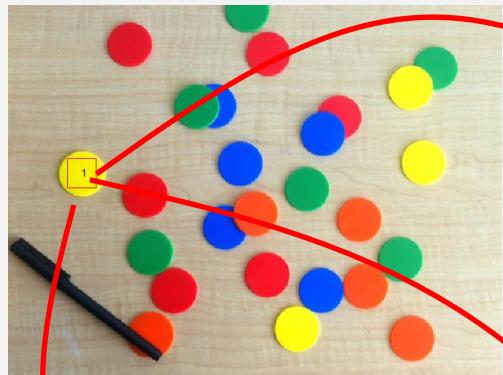


- L'image: I
- Un pixel: $x \in \{0,255\}^3$
- 7 Classes: F, S, R,O,V,J,B

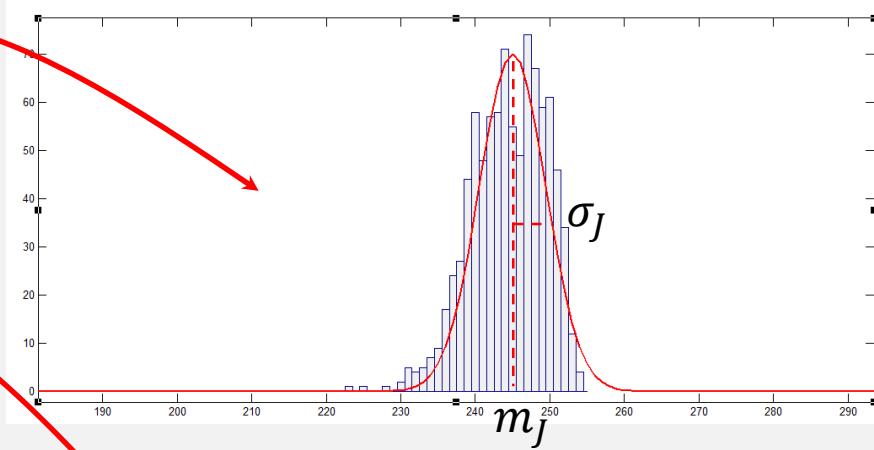
Question: Quel est la probabilité que le pixel x appartienne à la classe 'F', 'S', 'R',....?

Classification Bayesienne

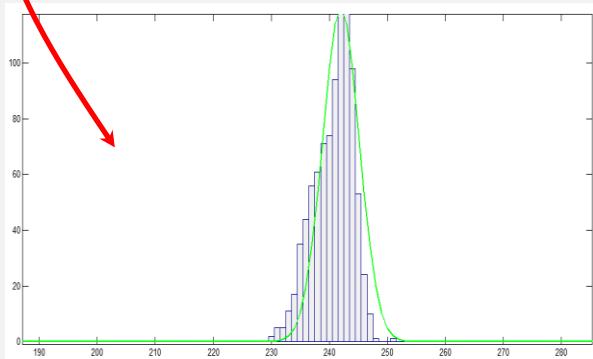
Connaissance à priori?



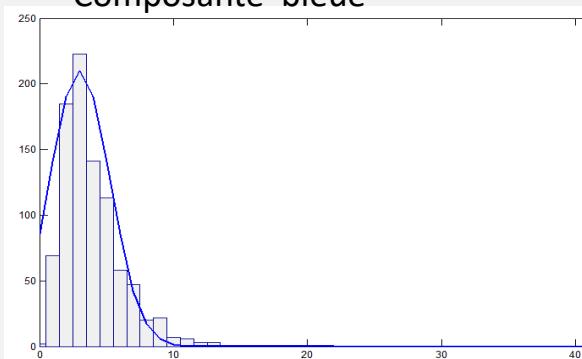
Composante rouge



Composante verte



Composante bleue



Classification Bayesienne

Connaissance à priori?

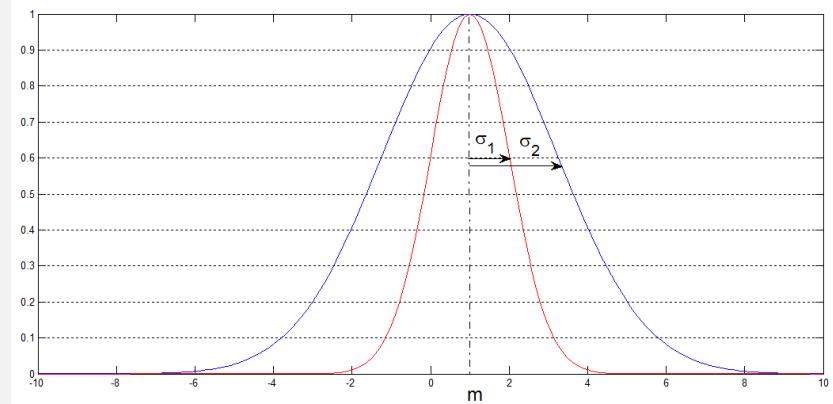
Loi de probabilité à priori: On peut définir une probabilité à priori que x appartienne à la classe $C=\{F,S,R,O,J,V,B\}=\{C_i\}_{i=1..7}$

$$P(x/C_i)$$

En pratique on suppose que la loi de probabilité qui régit x est une loi normale ou gaussienne: 2 paramètres (m :moyenne, σ : ecart type)

A une dimension:

$$P(x/C_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-m_i)^2}{2\sigma^2}}$$



Classification Bayesienne

Connaissance à priori?

Loi Normale à plusieurs dimensions

- C'est le cas pour une image couleur, on a 3 composantes (R,G,B) : la dimension est 3
- Généralisation de la loi gaussienne
- $x = (r, g, b)$,
- La classe Ci est définie par:
 - Sa moyenne: $\mathbf{m}_i = (\mathbf{m}_i^r, \mathbf{m}_i^g, \mathbf{m}_i^b)$
 - Sa matrice de variance-covariance: $\mathbf{E}_i = \begin{bmatrix} (\sigma_i^r)^2 & \sigma_i^r \sigma_i^g & \sigma_i^r \sigma_i^b \\ \sigma_i^g \sigma_i^r & (\sigma_i^g)^2 & \sigma_i^g \sigma_i^b \\ \sigma_i^b \sigma_i^r & \sigma_i^b \sigma_i^g & (\sigma_i^b)^2 \end{bmatrix}$

Classification Bayesienne

Connaissance à priori?

Loi Normale à plusieurs dimensions:

$$P(x/c_i) = \frac{1}{(2\pi)^{3/2} |E_i|^{1/2}} \exp[-(x - m_i) E_i^{-1/2} (x - m_i)^T]$$

$|E_i|$: déterminant de la matrice de variance-covariance

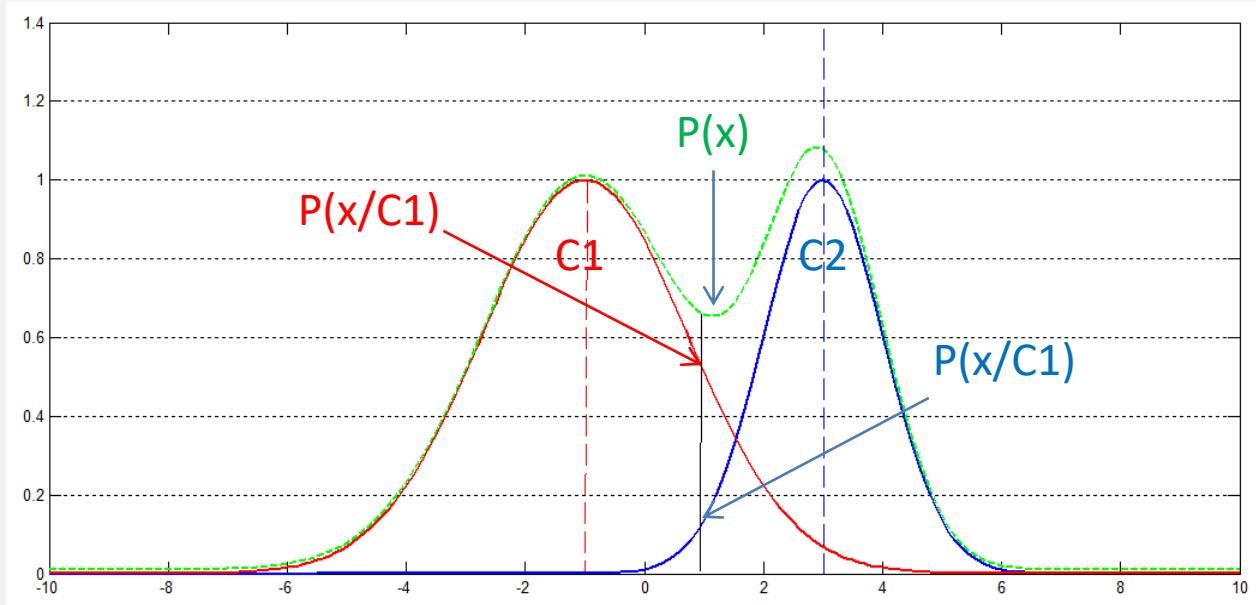
Classification Bayesienne

Règle de Bayes

- **Règle de Bayes:** $P\left(\frac{C_i}{x}\right)P(x) = P\left(\frac{x}{C_i}\right)P(Ci)$
 - $P(Ci)$: Probabilité de la classe Ci
 - $P\left(\frac{x}{Ci}\right)$: Probabilité à priori: sachant les caractéristiques de la classe Ci quelle est la probabilité que x appartient à Ci
 - $P\left(\frac{Ci}{x}\right)$: Probabilité à posteriori: sachant x, quelle est la probabilité qu'il appartient à Ci → **c'est ce qu'on recherche**
 - $P(x)$: probabilité de x

Classification Bayesienne

Règle de décision



Règle de décision:

$$\frac{P(\frac{C_1}{x})}{P(\frac{C_2}{x})} > \frac{P(C_2)}{P(C_1)}$$
$$\text{for } x \in C_1$$
$$\frac{P(\frac{C_1}{x})}{P(\frac{C_2}{x})} < \frac{P(C_2)}{P(C_1)}$$
$$\text{for } x \in C_2$$

Classification Bayesienne

Règle de décision

- **Règle de décision pour K classes**

$$\frac{P(\frac{C_i}{x})}{P(\frac{C_j}{x})} > \frac{P(C_j)}{P(C_i)} \quad \forall j = \{1 \dots k\} \setminus \{i\}$$

- **Règle de Bayes:** $P\left(\frac{C_i}{x}\right) P(x) = P\left(\frac{x}{C_i}\right) P(C_i)$

- **Simplification:**

On suppose que les classes sont équiprobables $P(C_1)=P(C_2)=\dots=P(C_k)$, et en s'appuyant sur le théoreme de Bayes, on obtient:

$$P\left(\frac{x}{C_i}\right) > P\left(\frac{x}{C_j}\right) \quad \forall j = \{1 \dots k\} \setminus \{i\}$$

Classification Bayesienne

Règle de décision

$$\frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-m_i)^2}{2\sigma^2}} \stackrel{x \in C_i}{>} \stackrel{x \in C_j}{<} \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma^2}}$$

On prend le logarithme:

$$-\log(\sqrt{2\pi}\sigma_i) - \frac{(x-m_i)^2}{2\sigma^2} \stackrel{x \in C_i}{>} \stackrel{x \in C_j}{<} -\log(\sqrt{2\pi}\sigma_j) - \frac{(x-m_j)^2}{2\sigma^2}$$

On pose $D_i(x) = \log(\sqrt{2\pi}\sigma_i) + \frac{(x-m_i)^2}{2\sigma^2}$

Classification Bayesienne

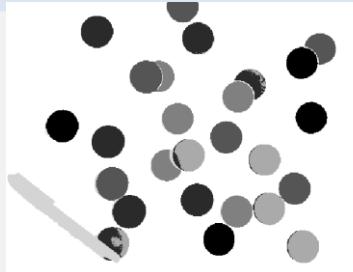
Règle de décision

- **Règle de décision:** on décide que x appartient à la classe l si

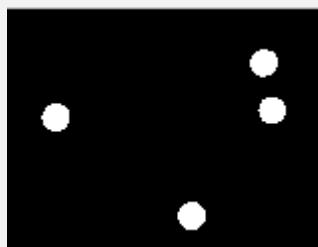
$$l = \operatorname{argmin}_{i=1 \dots k} (D_i(x))$$

Classification Bayesienne

Résultats



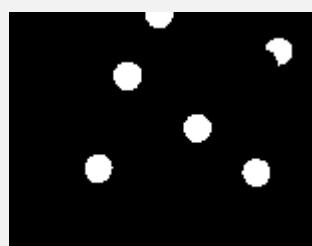
Etiquetage des pixels: 0,1,...,6



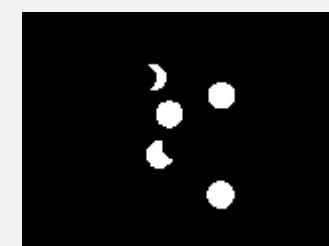
$R==0$



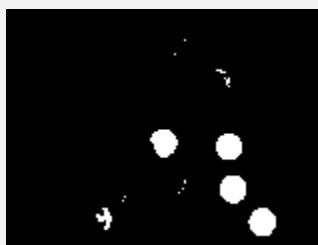
$R==1$



$R==2$



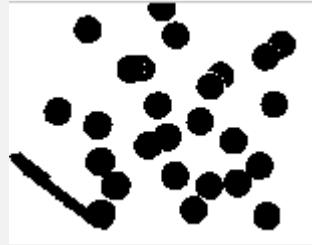
$R==3$



$R==4$



$R==5$

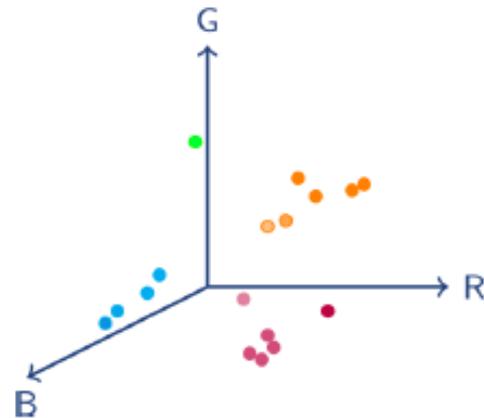


$R==6$

Clustering: k-moyenne

Introduction

Un pixel est maintenant représenté par un vecteur à valeurs dans $\mathbb{R}^B \Rightarrow$ travailler directement dans l'espace \mathbb{R}^B !



Le principe des méthodes de classification (ou plus exactement de coalescence) (*clustering*) est de regrouper les pixels en groupes homogènes.

Les k-moyennes

Illustration par l'image

Algorithme assez simple pour détecter plusieurs classes (≥ 2) dans une image
Considérons par exemple une image couleur contenant 3 classes , chaque plan contient n pixels (x_1, \dots, x_n), chaque pixel x_i contient d = 3 valeurs (RGB).



K-moyennes

Projections dans le plan RGB

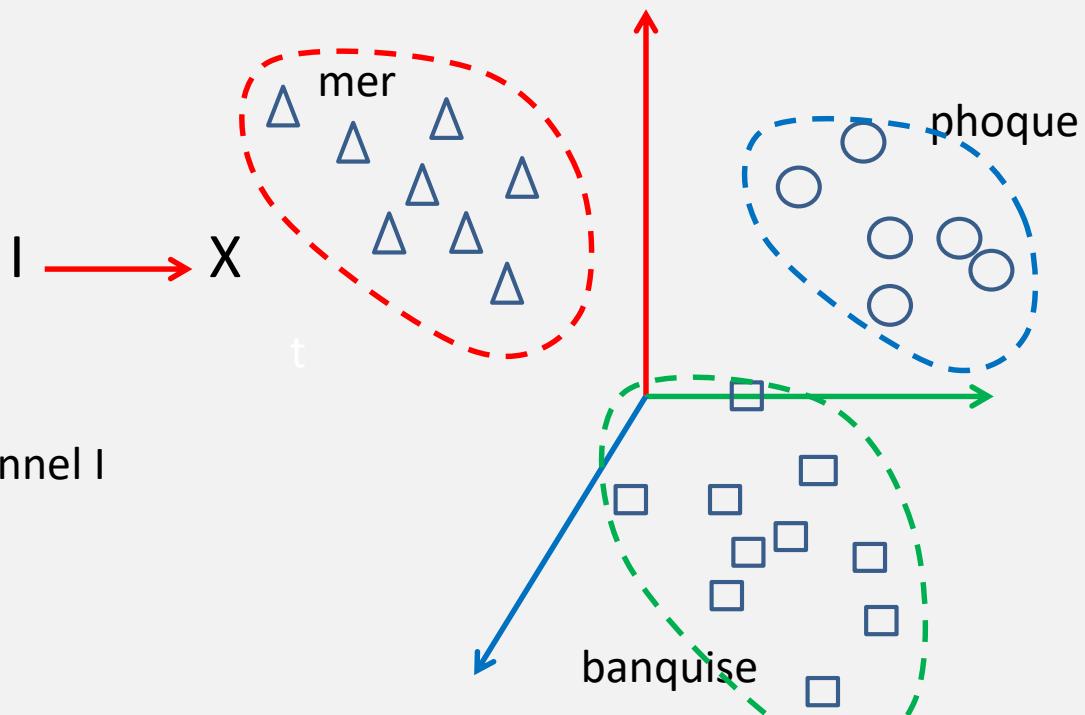


Image: tableau tridimensionnel I

$R=I(:,:,1);$

$G=I(:,:,2);$

$B=I(:,:,3);$

$X=[R(:),G(:),B(:)];$

K-moyennes

principes

Initialisation:

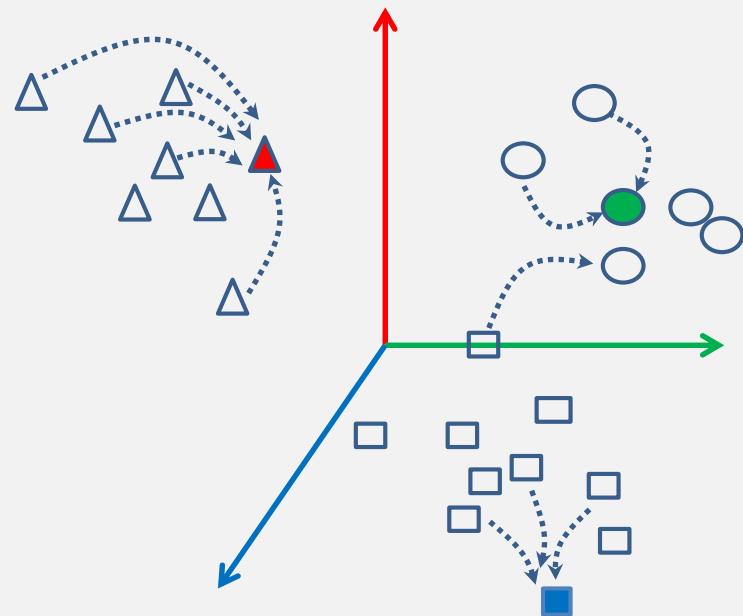
On définit le nombre de classes à détecter: **K**

On tire aléatoirement K données dans le tableau:
exemple des données en couleur (K=3).

Ces points deviennent le centre des classes.

Répétitions:

Tous les autres points sont affectés au centre le plus proche

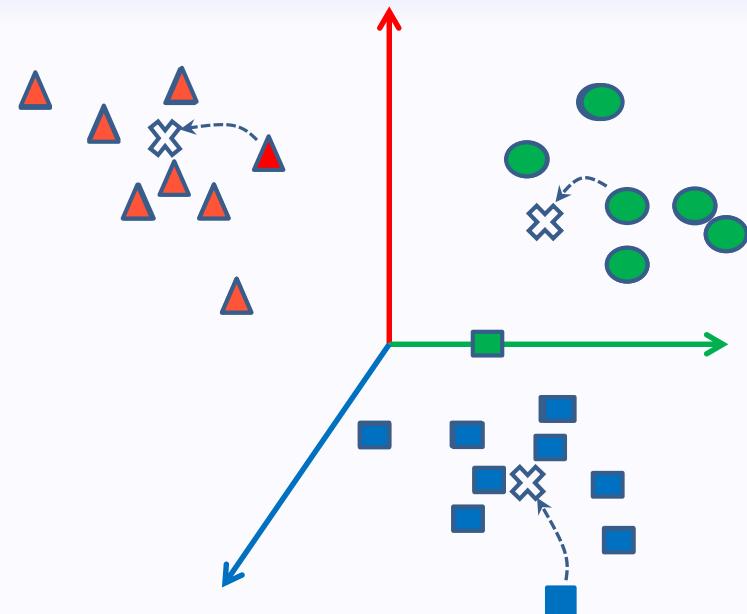


K-moyennes

principes

Répétitions:

- Tous les autres points sont affectés au centre le plus proche
- On rajuste les centres des classes en calculant le centre de chaque regroupement.
- On répète l'opération jusqu'à stabilisation des centres.



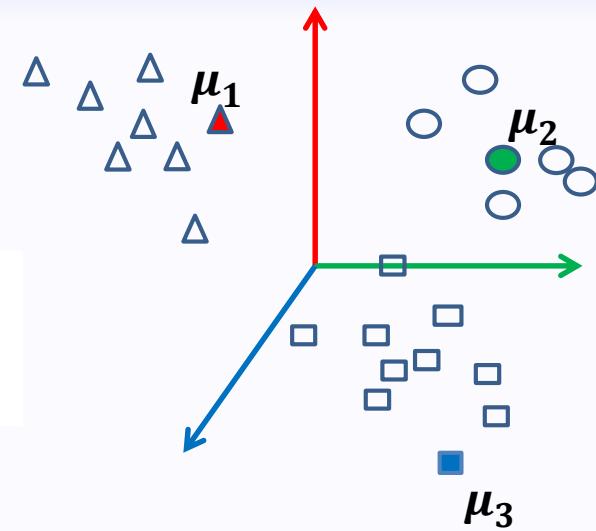
K-moyennes

Formellement

Soit un ensemble de données (x_1, \dots, x_n),
K-means minimise le critère d'erreur
(distorsion) suivant:

$$\mathcal{J}(\mu_1, \dots, \mu_K, z) = \sum_{k=1}^K \sum_{i=1}^n z_{ik} \|x_i - \mu_k\|^2$$

$$z_{ik} = \begin{cases} 1 & \text{si } i \in \text{classe } k \\ 0 & \text{sinon} \end{cases}$$



$$\|x_i - \mu_k\| = d(x_i, \mu_k) = \sqrt{\sum_{j=1}^d (x_{ij} - \mu_{kj})^2}$$

K-moyennes

Les étapes clés

Initialisation : On initialise les centres des classes $(\mu_1^{(0)}, \dots, \mu_K^{(0)})$ (à votre choix) pour donner le pas de départ de l'algorithme (par exemple on choisissant aléatoirement des centres "virtuels", ou K données parmi les données à traiter). Il s'agit donc de démarrer à l'itération $t = 0$ avec des valeurs initiales pour les paramètres du modèle $(\mu_1^{(0)}, \dots, \mu_K^{(0)})$.

Etape d'affectation (classification) : Chaque donnée est assignée à la classe du centre dont elle est la plus proche : $\forall i = 1, \dots, n$

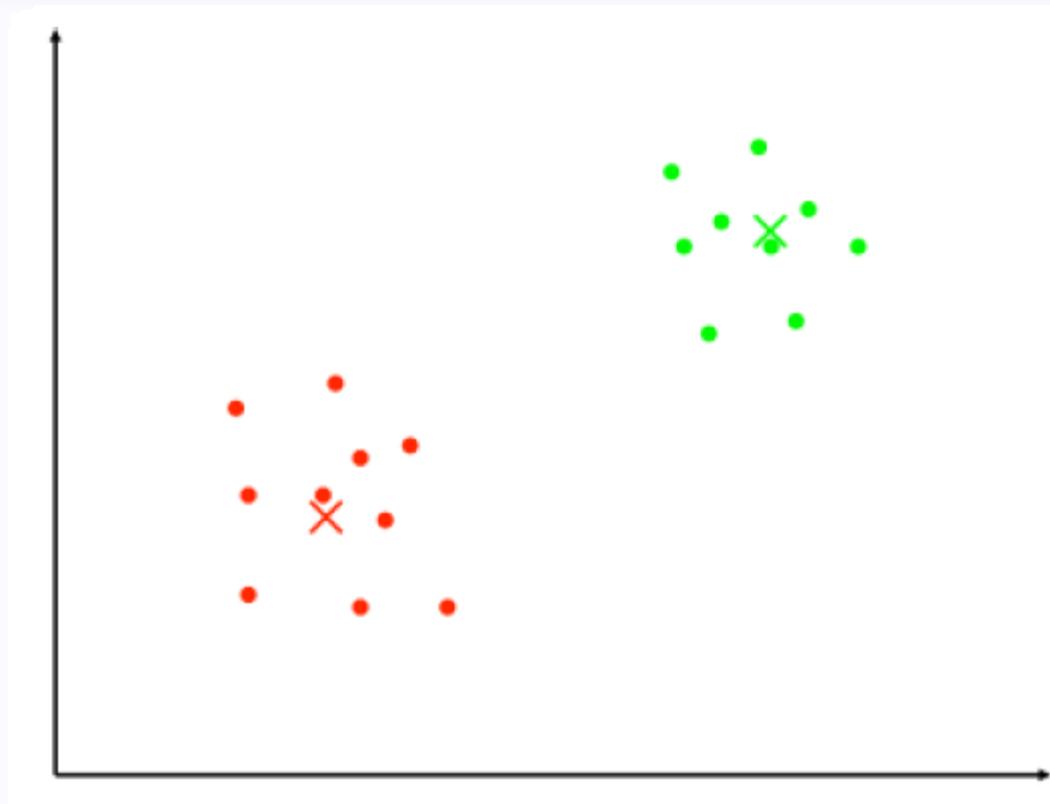
$$z_{ik}^{(t)} = \begin{cases} 1 & \text{si } k = \arg \min_{z \in \{1, \dots, K\}} \| \mathbf{x}_i - \mu_z \|^2 \\ 0 & \text{sinon.} \end{cases} \quad (3)$$

Etape de recalage des centres : le centre μ de chaque classe k est recalculé comme étant la moyenne arithmétique de toutes les données appartenant à cette classe (suite à l'étape d'affectation précédente) : $\forall k = 1, \dots, K$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^n z_{ik}^{(t)} \mathbf{x}_i}{\sum_{i=1}^n z_{ik}^{(t)}}, \quad (4)$$

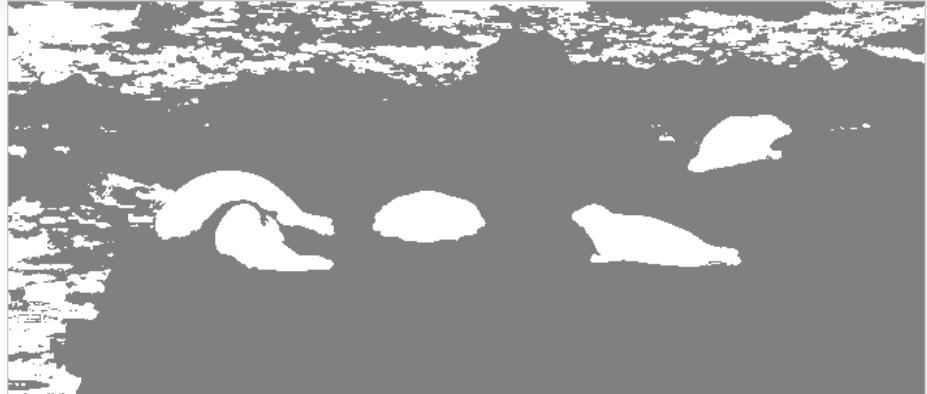
K-moyennes

Illustration en 2D

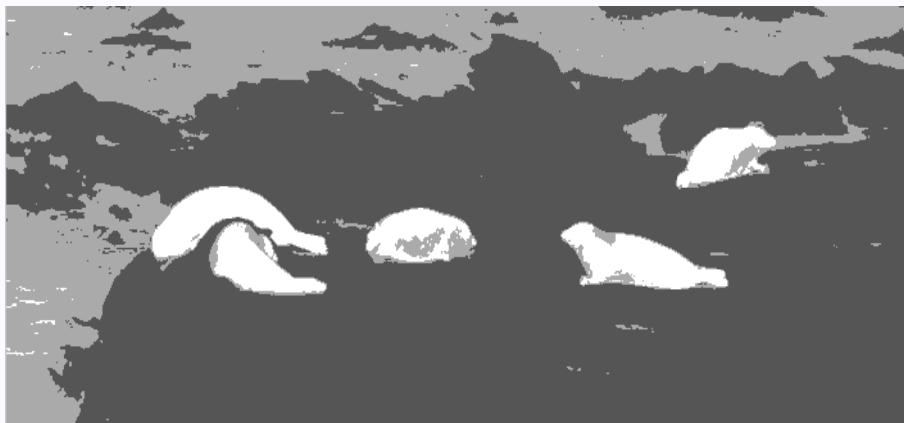


K-moyennes

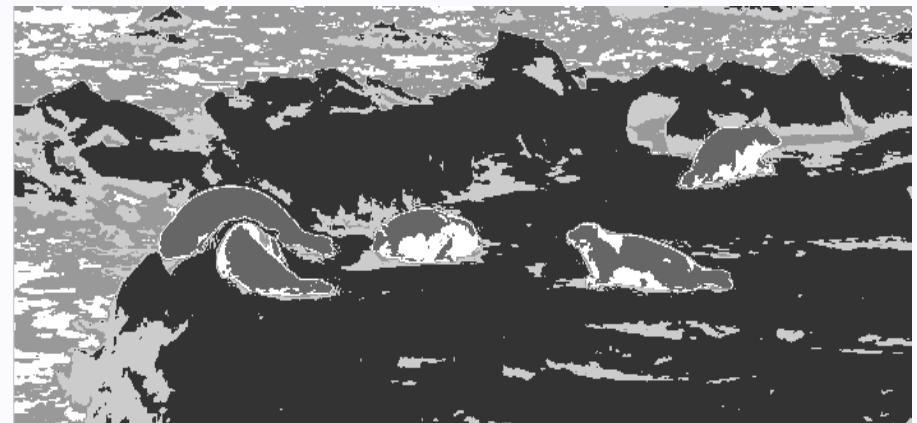
Application :segmentation d'images



K=2



K=3

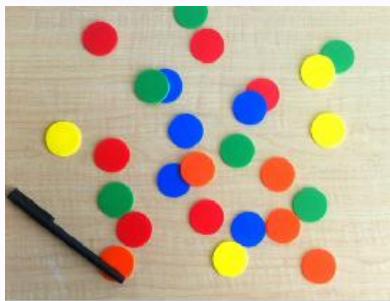


K=5

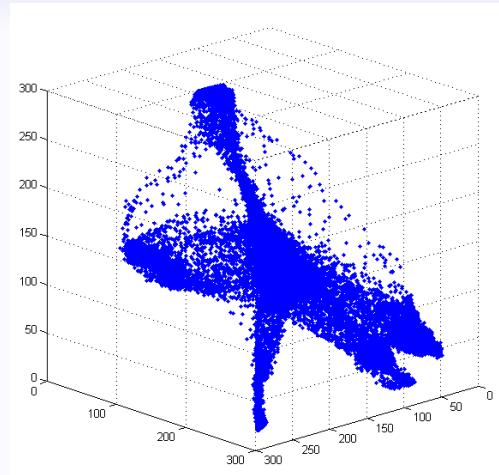
K-moyennes

Application: Les pieces

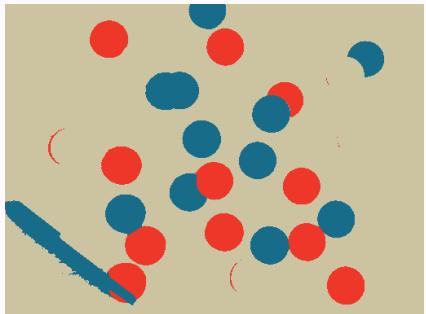
I



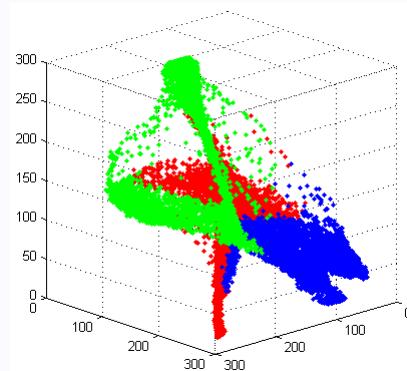
X



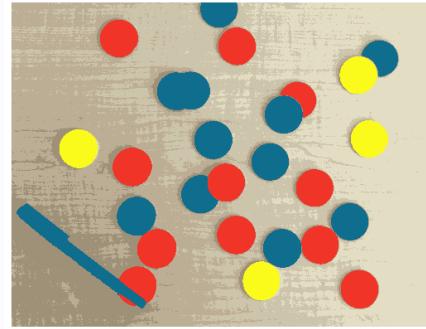
K=3



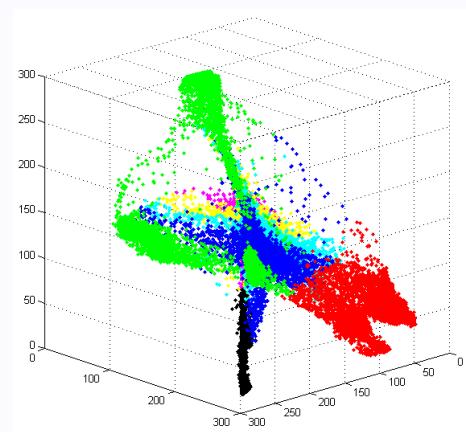
Etiquetage
des pixels



K=7



Etiquetage des
points RGB



K-moyennes

Avantages-Inconvénients

Avantages

- Méthode simple
- Implémentation facile
- Méthode généralement rapide
- Classes de variance conditionnelle minimale
- Fonctionne correctement lorsque les clusters sont sphériques



K-moyennes

Avantages-Inconvénients

Inconvénients

- Nécessite de connaître le nombre de classes
- Sensible aux minima locaux, donc à l'initialisation
- Peut être lent en grande dimension
- Échoue pour des structures non sphériques



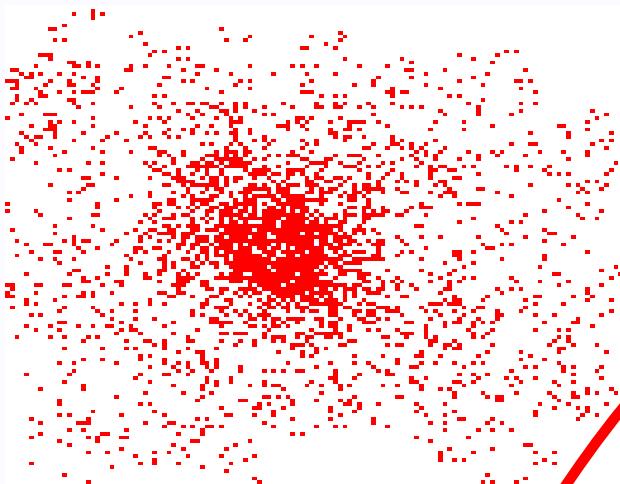
- Sensible aux valeurs aberrantes



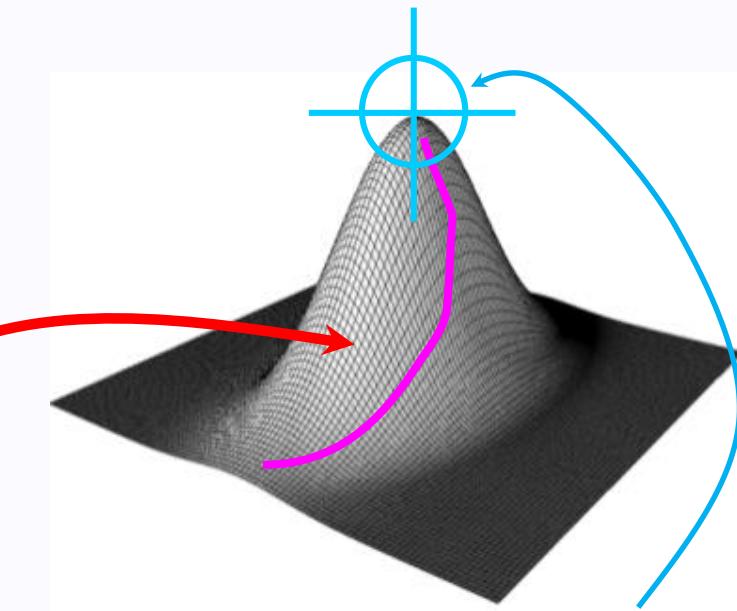
Mean Shift

Principe

- Le décalage moyen est une procédure permettant de localiser les maxima - les modes - d'une fonction de densité à partir de données discrètes échantillonnées à partir de cette fonction.



Points échantillonnés représentent une fonction de densité



Objectif: déterminer le maxima de cette fonction par une méthode du gradient

Mean Shift

Mathématiquement parlant....

- Il s'agit d'une méthode itérative, et nous commençons par une estimation initiale x . On considère une fonction noyau $K(x_i - x)$. Cette fonction détermine les poids du points x_i par rapport au centre x et permet d'évaluer la moyenne des points x_i ayant une influence (les points les plus proches).
- Plusieurs noyaux possibles: noyau uniforme – noyau gaussien - ...

- Déplacement moyen:

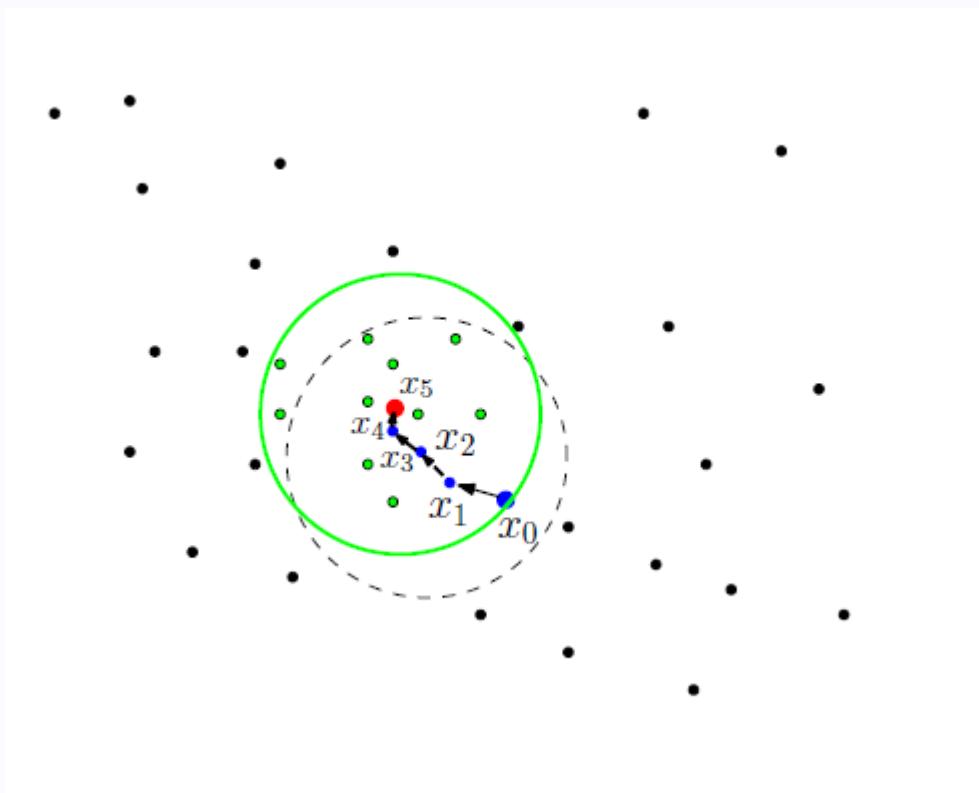
$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

ou $N(x)$ est un voisinage de x tel que $K(x_i - x) \neq 0$.

- **le décalage $m(x)-x$ est appelé le vecteur mean shift.**
- **Procédure itérative: $x_0=x$, $x_1=m(x_0)$, $x_2=m(x_1)$,..., jusqu'à stabilisation**

Mean Shift

Illustration du mean shift



Initialisation

x_0 : donnée initiale

R : rayon de la fenêtre

Iter 1

x_1 : centre des points dans le cercle
Déplacement du cercle de centre x_1

Iter 2

x_2 : centre des points dans le cercle
Déplacement du cercle de centre x_2

⋮

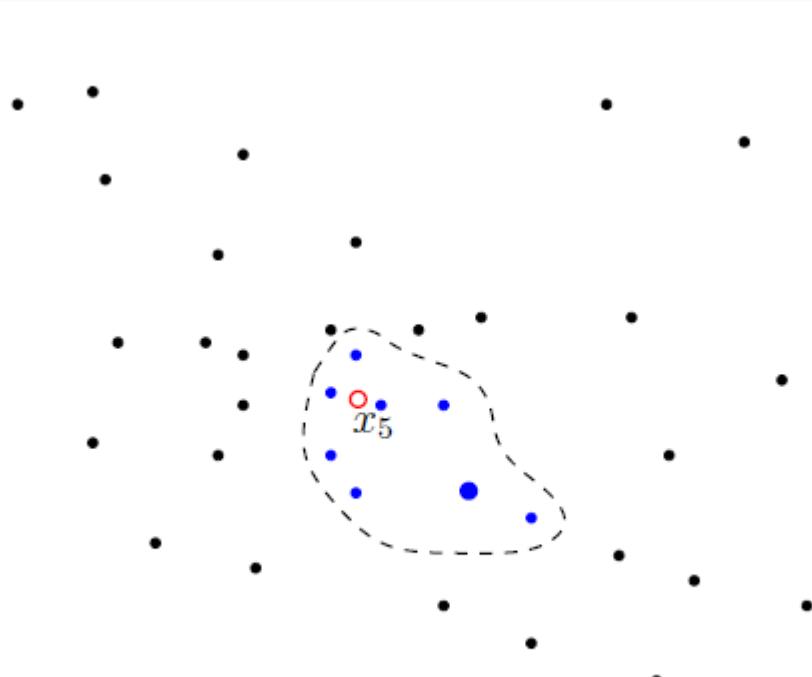
Iter 5

x_5 : point stationnaire

Convergence : $d(x_i, x_{i+1}) < \varepsilon$

Mean Shift

Création d'un cluster (groupe)



Étape 1

Un cluster associé au centre x_5 est créé :
Il est initialisé avec le premier point x_0

$$C_5 = \{x_0\}$$

Étape 2

Définition d'un rayon d'influence R_i :
Les points contenus dans le cercle
appartiennent au cluster:

$$C_5 = \{x_0, p_0\}$$

Étape 3

Mise à jour du cluster par déplacement du
cercle sur x_1 :

$$C_5 = \{x_0, p_0, p_1, p_2, p_3\}$$

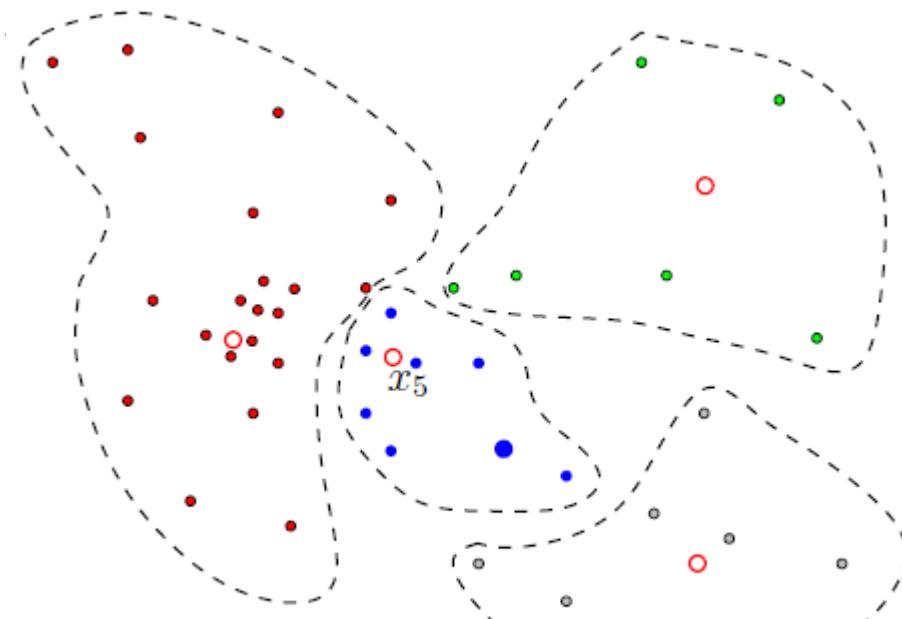
Étape n

Cluster détecté:

$$C_5 = \{x_0, p_0, p_1, p_2, p_3, p_4, p_5, p_6\}$$

Mean Shift

Création de plusieurs clusters



Étape 1

Répétition sur les autres points:
création d'autres clusters

$$C_5 = \{x_0, p_0, p_1, p_2, p_3, p_4, p_5, p_6\}$$

$$C_{10} = \{p_7, p_8, \dots\}$$

$$C_{\dots} = \{\dots\}$$

Étape 2

Fusion de clusters proches:
Définition d'un rayon de fusion: R_m

x_p, x_q points stationnaires du Mean-Shift

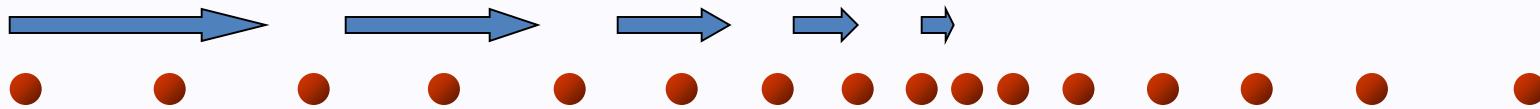
$$d(x_p, x_q) \leq R_m$$



fusionner C_p et C_q

Mean Shift

propriétés



- Vitesse de convergence automatique – La longueur du vecteur de déplacement dépend de l'amplitude du gradient.
- A proximité des maxima, les déplacements sont de plus en plus petits
- Pour le noyau uniforme, la convergence est atteinte en un nombre fini de pas.
- L'utilisation du noyau normal fournit une trajectoire lissée, mais sa convergence est plus faible que celle obtenue à partir du noyau uniforme.

Mean Shift

Avantages - Inconvénients

Avantages :

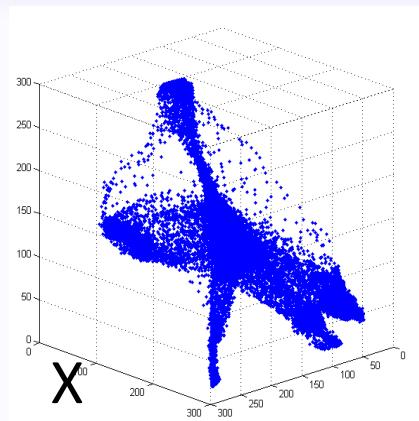
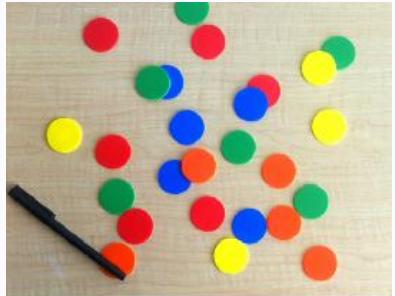
- Un outil indépendant de l'application
- Aucune hypothèse à priori sur la forme des classes
- L'espace des caractéristiques peut être arbitraire.
- Un SEUL paramètre à fixer : R

Désavantages :

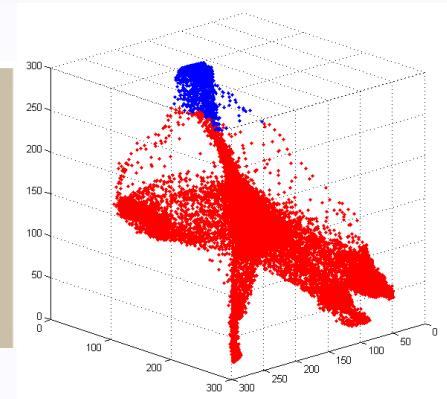
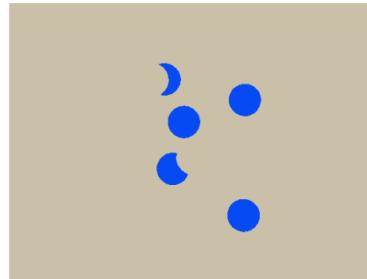
- La taille de la fenêtre n'est pas trivial
- Une taille inappropriée peut entraîner l'agglomération ou la fusion de plusieurs modes distincts. Elle peut générer également des modes "superficiels". → Utiliser une fenêtre adaptative

Mean Shift

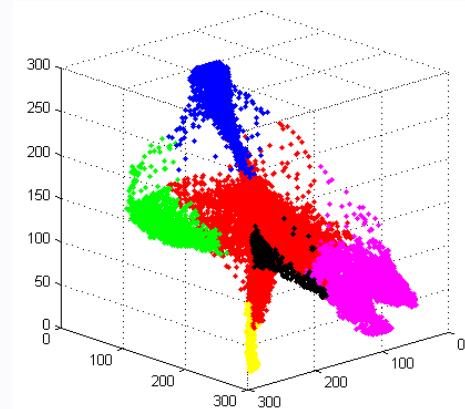
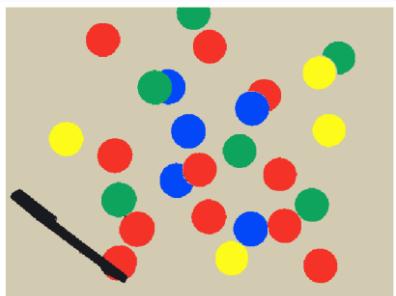
Exemple



R=0.5, K=2



R=0.2, K=6



R=0.1, K=58



Mean Shift

Avantages - Inconvénients

Avantages :

- Un outil indépendant de l'application
- Aucune hypothèse à priori sur la forme des classes
- L'espace des caractéristiques peut être arbitraire.
- Un SEUL paramètre à fixer : R

Désavantages :

- La taille de la fenêtre n'est pas trivial
- Une taille inappropriée peut entraîner l'agglomération ou la fusion de plusieurs modes distincts. Elle peut générer également des modes "superficiels". → Utiliser une fenêtre adaptative