Clustering

Introduction à l'apprentissage machine – GIF-4101 / GIF-7005

Professeur : Christian Gagné

Semaine 13

UNIVERSITÉ LAVAL

13.1 Quantification de vecteurs

Clustering

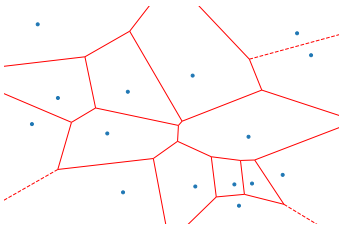
- Apprentissage supervisé
  - Étiquettes de classe disponibles
  - Méthodes paramétriques : observations suivent une certaine densité de probabilité $p(\mathbf{x}|C_i)$
- Un groupe de données par classe
  - Selon une loi normale, moyenne et covariance partagées par toutes les données
  - En pratique, les données d'une classe peuvent tenir dans plusieurs groupes
    - Écriture cursive : différentes façons de faire des 1 et des 7
    - Détecter des intrusions dans un système informatique
- Clustering
  - Identifier des groupes « naturels » dans les données

Quantification de vecteur

- Quantification de vecteurs
  - Discrétiser un espace $\mathbb{R}^D$, en le séparant en K régions
- Quantification possible en utilisant K vecteurs de référence $\mathbf{m}_i$
  - Assignation d'une donnée $\mathbf{x}^t$ au vecteur de référence le plus proche

$$b_i^t = \begin{cases} 1 & i = \text{argmin}_j \|\mathbf{x}^t - \mathbf{m}_j\| \\ 0 & \text{autrement} \end{cases}$$

- Partitionnement de l'espace selon un diagramme de Voronoï

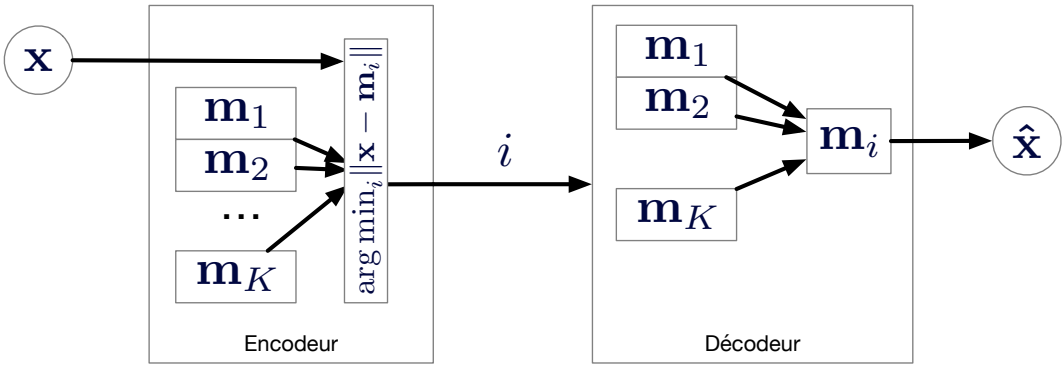


Compression et reconstruction

- Compression complète de l'espace $\mathbb{R}^D$ en K vecteurs de référence $\mathbf{m}_i$
  - Chaque point dans l'espace d'origine est associé à un des vecteurs de référence (valeurs discrètes)
- Exemple de la *colormap*
  - Couleur d'un pixel dans une image : 24 bits
  - Transmettre image de 640×400 pixels : plus de 6M bits
  - Compression avec une *colormap* de 256 couleurs différentes
    - La *colormap* tient sur 6144 bits
    - Pixels réfèrent à la *colormap* : 8 bits par pixel
    - Image encodée sur 2M bits, soit gain 3 : 1
  - Perte d'information si plus de 256 couleurs différentes dans l'image
    - Choix de couleurs minimisant un certain critère
- Erreur de reconstruction

$$E(\{\mathbf{m}_i\}_{i=1}^K|\mathcal{X}) = \sum_t \sum_i b_i^t \|\mathbf{x}^t - \mathbf{m}_i\|^2$$

Compression par clustering



13.2 K-means

K-means

- Calcul de l'optimum de l'erreur de reconstruction $E(\{\mathbf{m}_i\}_{i=1}^K|\mathcal{X})$ selon les $\mathbf{m}_i$ est impossible analytiquement
  - Position optimale des centres $\mathbf{m}_i$ dépend des étiquettes b_i^t
  - Choix optimal des étiquettes b_i^t dépend de la position des centres $\mathbf{m}_i$!
- Résolution itérative, par approximations successives des b_i^t et $\mathbf{m}_i$
  - Estimer $b_i^t(j+1)$ selon les $\mathbf{m}_i(j)$
  - Estimer $\mathbf{m}_i(j+1)$ selon les $b_i^t(j+1)$
  - Répéter jusqu'à convergence ou épuisement des ressources

Estimation des centres

- Estimation des centres $\mathbf{m}_i$ selon les étiquettes b_i^t
  - $\mathbf{m}_i$ avec dérivée partielle de $E(\{\mathbf{m}_i\}_{i=1}^K|\mathcal{X})$ selon $\mathbf{m}_j$

$$\begin{aligned}\frac{\partial E(\{\mathbf{m}_i\}_{i=1}^K|\mathcal{X})}{\partial \mathbf{m}_j} &= \frac{\partial \sum_t \sum_i b_i^t (\mathbf{x}^t - \mathbf{m}_i)^\top (\mathbf{x}^t - \mathbf{m}_i)}{\partial \mathbf{m}_j} = 0 \\ &= -2\sum_t b_j^t (\mathbf{x}^t - \mathbf{m}_j) = 0 \\ \mathbf{m}_j &= \frac{\sum_t b_j^t \mathbf{x}^t}{\sum_t b_j^t}, j = 1, \ldots, K\end{aligned}$$

Algorithme des K-means

1. Initialiser les centres $\mathbf{m}_i$ aléatoirement
2. Tant que le critère d'arrêt n'est pas atteint, répéter :
   2.1 Estimer les étiquettes des données b_i^t selon les positions des centres $\mathbf{m}_i$

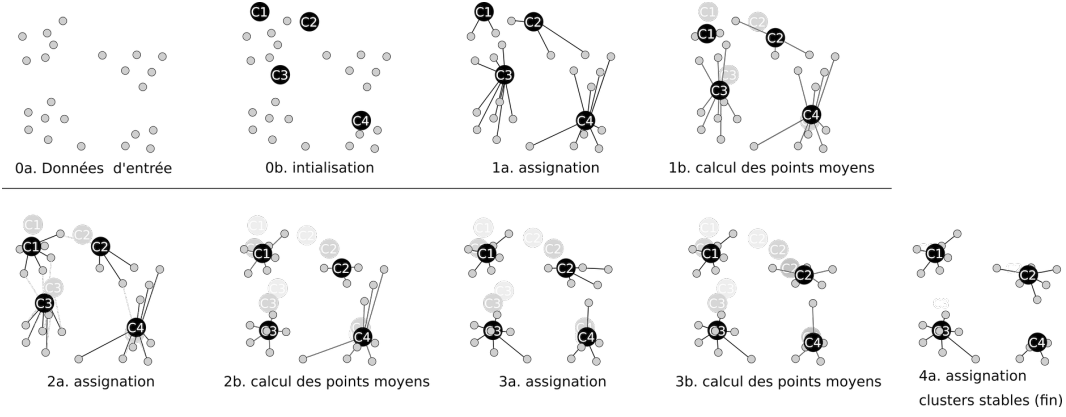
$$b_i^t = \begin{cases} 1 & i = \text{argmin}_j \|\mathbf{x}^t - \mathbf{m}_j\| \\ 0 & \text{autrement} \end{cases}, \; i = 1, \ldots, K, \; t = 1, \ldots, N$$

   2.2 Optimiser la position $\mathbf{m}_i$ des centres avec les nouvelles étiquettes b_i^t

$$\mathbf{m}_i = \frac{\sum_t b_i^t \mathbf{x}^t}{\sum_t b_i^t}, \; i = 1, \ldots, K$$

3. Retourner les valeurs des centres $\mathbf{m}_i$

Illustration de K-means



Initialisation et critères d'arrêt

- Approches possibles pour initialisation des centres $\mathbf{m}_i$
  - Sélectionner aléatoirement K instances de $\mathcal{X}$
  - Calculer le vecteur moyen de toutes les données et initialiser K centres autour de cette moyenne, avec légères variations aléatoires pour chacun
  - Basée sur la composante principale
    1. Calculer la composante principale
    2. Projeter les données sur la droite correspondante
    3. Partitionner les données sur la droite en K groupes de taille égale
    4. Calculer la moyenne de chacun de ces groupes dans l'espace d'origine et les utiliser comme centres de départ
- Critères d'arrêt
  - Nombre maximum d'itérations
  - Variation de la position des centres inférieure à un seuil

Propriétés des K-means

- Aucune garanti de convergence vers l'optimum global
  - Issue dépend du choix des positions initiales des centres
- Convergence relativement rapide
- Nombre de centres à utiliser fixé à l'avance
  - Nécessite une connaissance du nombre de groupes formant les données
  - Si nombre de groupes inconnu, détermination de K empirique
  - Algorithme *leader cluster* : ajout incrémental de centres lorsque distance d'une donnée à son centre dépasse un seuil
  - Variation : ajouter un centre lorsque le nombre de données associées à un centre dépasse un seuil

Illustration de K-means : 2 groupes

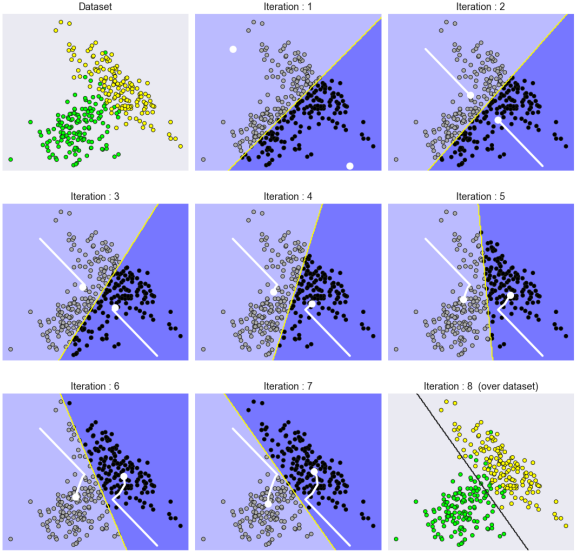
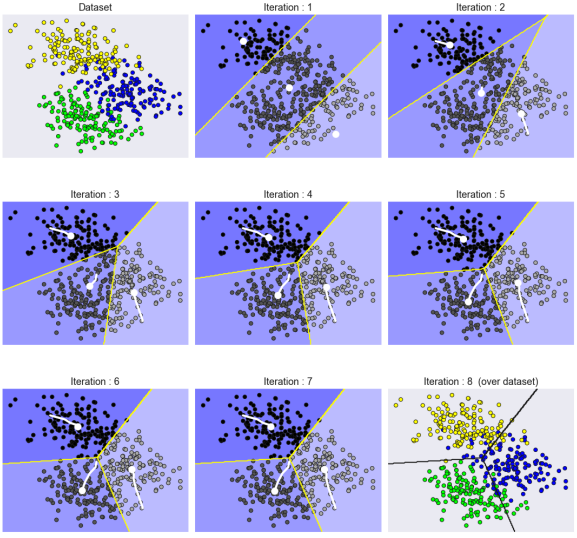
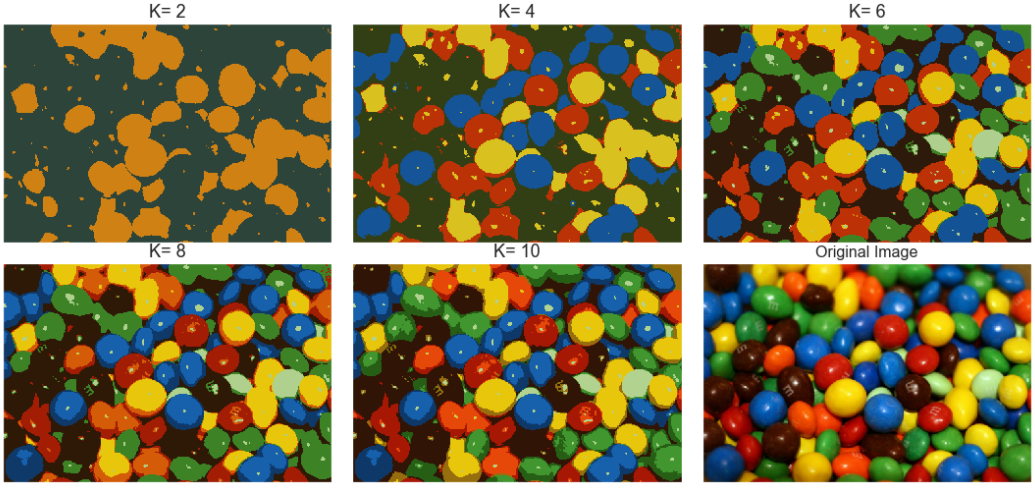


Illustration de K-means : 3 groupes



Application : compression de la colormap



13.3 Densité-mélange

Densité-mélange

- Densité-mélange : combinaison de lois de densité associées à plusieurs groupes

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x}|\mathcal{G}_i)P(\mathcal{G}_i)$$

- Lien direct avec le cas supervisé
  - Formulation similaire, mais les groupes sont connus et identifiés dans le cas supervisé
  - Peut être utilisé avec des méthodes paramétriques, lorsqu'il y a beaucoup de groupes dans chaque classe
- Densité-mélange de composantes suivant une loi normale multivariée
  - Densité de composantes : $(\mathbf{x}|\mathcal{G}_i) \sim \mathcal{N}_D(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
  - Paramétrisation : $\Phi = \{P(\mathcal{G}_i), \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{K}$
- Utilise des échantillons non étiquetés, $\mathcal{X} = \{\mathbf{x}^t\}_{t=1}^{N}$

Probabilités de la densité-mélange

- Densité-mélange

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x}|\mathcal{G}_i) P(\mathcal{G}_i)$$

- Proportion du groupe $\mathcal{G}_i$ dans le mélange, $P(\mathcal{G}_i)$

$$\sum_{i} P(\mathcal{G}_i) = 1$$

- Probabilité que $\mathbf{x}$ appartient au groupe $\mathcal{G}_i$, $P(\mathcal{G}_i|\mathbf{x})$

$$P(\mathcal{G}_i|\mathbf{x}) = \frac{P(\mathcal{G}_i) p(\mathbf{x}|\mathcal{G}_i)}{\sum_j P(\mathcal{G}_j) p(\mathbf{x}|\mathcal{G}_j)}$$

Variables indicatrices cachées

- Variables indicatrices cachées $\mathbf{z}^t = \{z_1^t, \ldots, z_K^t\}$
  - z_i^t : association de la donnée $\mathbf{x}^t$ au groupe $\mathcal{G}_i$
  - On ne connait pas les « véritables » valeurs des $\mathcal{Z}$: variables cachées du problème
  - Simplification de la notation : $\pi_i = P(\mathcal{G}_i)$
  - Distribution multinomiale : $z_i^t = 1$ indique si variable $\mathbf{x}^t$ appartient au groupe $\mathcal{G}_i$, et $z_i^t = 0$ autrement

$$P(\mathbf{z}^t) = \prod_{i=1}^{K} \pi_i^{z_i^t}$$

- Vraisemblance d'observation de $\mathbf{x}^t$

$$p(\mathbf{x}^t|\mathbf{z}^t) = \prod_{i=1}^{K} p(\mathbf{x}^t|\mathcal{G}_i)^{z_i^t}$$

- Probabilité jointe $p(\mathbf{x}^t,\mathbf{z}^t)$

$$p(\mathbf{x}^t,\mathbf{z}^t) = P(\mathbf{z}^t)p(\mathbf{x}^t|\mathbf{z}^t)$$

Fonction de vraisemblance

- Fonction de log-vraisemblance de la paramétrisation Φ selon l'association des données de $\mathcal{X}$ aux groupes donnés par $\mathcal{Z}$

$$
\begin{aligned}
L(\Phi|\mathcal{X},\mathcal{Z}) &= \log \prod_t p(\mathbf{x}^t,\mathbf{z}^t|\Phi) = \log \prod_t \left[P(\mathbf{z}^t|\Phi)\, p(\mathbf{x}^t|\mathbf{z}^t,\Phi)\right] \\
&= \log \prod_t \prod_i \left[\pi_i^{z_i^t}\, p(\mathbf{x}^t|\mathcal{G}_i,\Phi)^{z_i^t}\right] \\
&= \sum_t \sum_i \left[\log \pi_i^{z_i^t} + \log p(\mathbf{x}^t|\mathcal{G}_i,\Phi)^{z_i^t}\right] \\
&= \sum_t \sum_i z_i^t \left(\log \pi_i + \log p(\mathbf{x}^t|\mathcal{G}_i,\Phi)\right) \\
&= \sum_t \sum_i z_i^t \left(\log \pi_i + \log \frac{\pi_i P(\mathcal{G}_i|\mathbf{x}^t,\Phi)}{\sum_j \pi_j P(\mathcal{G}_j|\mathbf{x}^t,\Phi)}\right)
\end{aligned}
$$

13.4 Algorithme Espérance-Maximisation

Algorithme Espérance-Maximisation

- Appartenance $h_i^t \equiv P(\mathcal{G}_i|\mathbf{x}^t,\Phi)$: association à un groupe $\mathcal{G}_i$ d'une donnée $\mathbf{x}^t$ selon la paramétrisation Φ (observation variable cachée $\mathbf{z}^t$)
- Log-vraisemblance dépend de la paramétrisation Φ selon l'association des variables cachées $\mathcal{Z}$
  - Similairement, l'association des variables cachées $\mathcal{Z}$ dépend de paramétrisation Φ
  - On ne connait pas le vrai $\mathcal{Z}$ (variables aléatoires cachées) : optimisation de l'**espérance de vraisemblance**
  - Optimisation de l'équation analytique impossible : approche itérative
- Algorithme Espérance-Maximisation (EM)
  - Étape E : calcul de l'espérance des associations aux groupes $h_i^t \equiv P(\mathcal{G}_i|\mathbf{x}^t,\Phi)$ avec paramétrisation Φ actuelle
  - Étape M : obtenir nouvelle paramétrisation Φ^{l+1} maximisant l'espérance de vraisemblance $\mathcal{Q}(\Phi|\Phi^l)$

$$\mathcal{Q}(\Phi|\Phi^l) = \mathbb{E}\left[L(\Phi|\mathcal{X},\mathcal{Z})|\mathcal{X},\Phi^l\right], \qquad \Phi^{l+1} = \underset{\Phi}{\operatorname{argmax}}\, \mathcal{Q}(\Phi|\Phi^l)$$

Étape E

- Étant donné Φ^l, quelle est l'espérance de vraisemblance d'autres paramétrisations Φ possibles ?

$$\begin{aligned}\mathcal{Q}(\Phi|\Phi^l) &= \mathbb{E}\left[L(\Phi|\mathcal{X},\mathcal{Z})|\mathcal{X},\Phi^l\right] \\ &= \sum_t \sum_i \mathbb{E}[z_i^t|\mathcal{X},\Phi^l]\left(\log \pi_i + \log p(\mathbf{x}^t|\mathcal{G}_i,\Phi)\right)\end{aligned}$$

- Espérance d'étiquetage $\mathbb{E}[z_i^t|\mathcal{X},\Phi^l]$ donnée par :

$$\begin{aligned}\mathbb{E}[z_i^t|\mathcal{X},\Phi^l] &= \mathbb{E}[z_i^t|\mathbf{x}^t,\Phi^l] && \mathbf{x}^t \text{ sont iid} \\ &= P(z_i^t = 1|\mathbf{x}^t,\Phi^l) && z_i^t \text{ est booléen} \\ &= \frac{P(z_i^t=1|\Phi^l)p(\mathbf{x}^t|z_i^t=1,\Phi^l)}{p(\mathbf{x}^t|\Phi^l)} && \text{règle de Bayes} \\ &= \frac{\pi_i p(\mathbf{x}^t|\mathcal{G}_i,\Phi^l)}{\sum_j \pi_j p(\mathbf{x}^t|\mathcal{G}_j,\Phi^l)} = \frac{P(\mathcal{G}_i)p(\mathbf{x}^t|\mathcal{G}_i,\Phi^l)}{\sum_j P(\mathcal{G}_j)p(\mathbf{x}^t|\mathcal{G}_j,\Phi^l)} \\ &= P(\mathcal{G}_i|\mathbf{x}^t,\Phi^l) \equiv h_i^t\end{aligned}$$

Espérance de vraisemblance

- Interprétation de h_i^t
  - $h_i^t \equiv \mathbb{E}[z_i^t|\mathcal{X},\Phi^l] = P(\mathcal{G}_i|\mathbf{x}^t,\Phi^l)$ donne la probabilité a posteriori que $\mathbf{x}^t$ appartienne au groupe $\mathcal{G}_i$
  - Observation probabiliste de la variable cachée z_i^t
  - Réinterprétation d'un discriminant pour le clustering
  - h_i^t est une version relaxée de l'appartenance binaire b_i^t des K-means
- Espérance de vraisemblance résultante

$$\begin{aligned}\mathcal{Q}(\Phi|\Phi^l) &= \sum_t \sum_i h_i^t \left[\log \pi_i + \log p(\mathbf{x}^t|\mathcal{G}_i,\Phi^l)\right] \\ &= \sum_t \sum_i h_i^t \log \pi_i + \sum_t \sum_i h_i^t \log p(\mathbf{x}^t|\mathcal{G}_i,\Phi^l)\end{aligned}$$

Étape M

- Étape M : trouver une nouvelle paramétrisation Φ^{l+1} maximisant l'espérance de vraisemblance $\mathcal{Q}(\Phi|\Phi^l)$

$$\begin{aligned}\Phi^{l+1} &= \underset{\Phi}{\text{argmax}}\, \mathcal{Q}(\Phi|\Phi^l) \\ \mathcal{Q}(\Phi|\Phi^l) &= \sum_t \sum_i h_i^t \log \pi_i + \sum_t \sum_i h_i^t \log p(\mathbf{x}^t|\mathcal{G}_i,\Phi^l)\end{aligned}$$

- Maximum aux dérivées partielles nulles
  - π_i est une probabilité, donc contrainte que $\sum_i \pi_i = 1$, résolution par méthode de Lagrange

$$\frac{\partial \mathcal{Q}(\Phi|\Phi^l)}{\partial \pi_j} = \frac{\partial}{\partial \pi_j}\left[\sum_t \sum_i h_i^t \log \pi_i - \lambda\left(\sum_i \pi_i - 1\right)\right] = 0$$

  - Résolution de Φ spécifique à la loi de probabilité

Résolution des probabilités a priori π_i

- Résolution de $\partial \mathcal{Q}(\Phi|\Phi^l)/\partial \pi_i$

$$
\begin{aligned}
\frac{\partial \mathcal{Q}(\Phi|\Phi^l)}{\partial \pi_j} &= \frac{\partial}{\partial \pi_j}\left[\sum_t \sum_i h_i^t \log \pi_i - \lambda\left(\sum_i \pi_i - 1\right)\right] = 0 \\
&= \sum_t \frac{h_j^t}{\pi_j} - \lambda = 0 \\
\pi_i \sum_t \frac{h_i^t}{\pi_i} &= \pi_i \lambda \quad \Rightarrow \quad \sum_i \frac{\pi_i}{\pi_i} \sum_t h_i^t = \lambda \sum_i \pi_i = \lambda \\
\sum_i \frac{\pi_i}{\pi_i} \sum_t h_i^t &= \sum_t \sum_i h_i^t = N \quad \Rightarrow \quad \lambda = N \\
\frac{1}{\pi_i} \sum_t h_i^t - N &= 0 \quad \Rightarrow \quad \pi_i = \frac{\sum_t h_i^t}{N}
\end{aligned}
$$

13.5 Algorithme EM pour loi normale multivariée

Algorithme EM pour loi normale multivariée

- Instance spécifique de l'algorithme EM, $(\mathbf{x}^t|\mathcal{G}_i,\Phi) \sim \mathcal{N}_D(\mathbf{m}_i,\mathbf{S}_i)$
- Résolution du $\mathbf{m}_j$ de $\Phi = \{\pi_i,\mathbf{m}_i,\mathbf{S}_i\}_{i=1}^K$

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{m}_j} \sum_t \sum_i h_i^t \log \frac{1}{(2\pi)^{0,5D}|\mathbf{S}_i|^{0,5}} \exp\left[-\frac{1}{2}(\mathbf{x}^t - \mathbf{m}_i)^\top \mathbf{S}_i^{-1}(\mathbf{x}^t - \mathbf{m}_i)\right] &= 0 \\
\frac{\partial}{\partial \mathbf{m}_j} \sum_t \sum_i h_i^t (\mathbf{x}^t - \mathbf{m}_i)^\top \mathbf{S}_i^{-1}(\mathbf{x}^t - \mathbf{m}_i) &= 0 \\
\sum_t h_j^t (\mathbf{x}^t - \mathbf{m}_j)(-1) &= 0 \\
\sum_t h_j^t \mathbf{x}^t &= \mathbf{m}_j \sum_t h_j^t \\
\mathbf{m}_j &= \frac{\sum_t h_j^t \mathbf{x}^t}{\sum_t h_j^t}
\end{aligned}
$$

Résolution de $\mathbf{S}_j$

- Résolution du $\mathbf{S}_j$ de $\Phi = \{\pi_i, \mathbf{m}_i, \mathbf{S}_i\}_{i=1}^{K}$

$$\frac{\partial}{\partial \mathbf{S}_j} \sum_t \sum_i h_i^t \log \frac{1}{(2\pi)^{0,5D} |\mathbf{S}_i|^{0,5}} \exp\left[-\frac{1}{2}(\mathbf{x}^t - \mathbf{m}_i)^\top \mathbf{S}_i^{-1}(\mathbf{x}^t - \mathbf{m}_i)\right] = 0$$

$$\mathbf{S}_j = \frac{\sum_t h_j^t (\mathbf{x}^t - \mathbf{m}_j)(\mathbf{x}^t - \mathbf{m}_j)^\top}{\sum_t h_j^t}$$

- Résolution de $\mathbf{S}_j$ est subtile, requiert le théorème spectral
  - Pour plus de détails, voir :
    `http://en.wikipedia.org/wiki/Estimation_of_covariance_matrices`

Récapitulatif algorithme EM pour loi normale multivariée

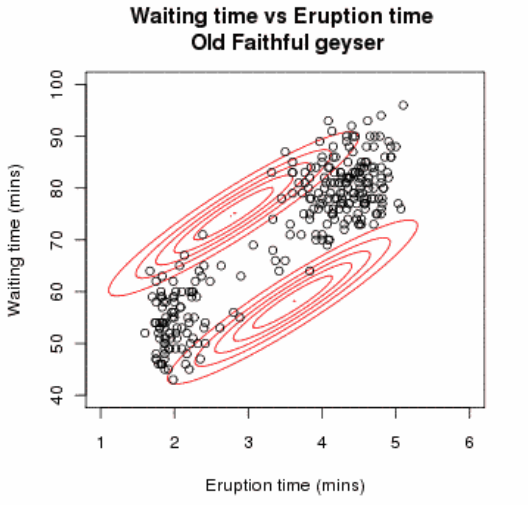
- Étape E : évaluation de $h_i^t,\ i = 1,\ldots,K,\ t = 1,\ldots,N$

$$h_i^t = \frac{\pi_i |\mathbf{S}_i|^{-0,5} \exp\left[-0,5(\mathbf{x}^t - \mathbf{m}_i)^\top \mathbf{S}_i^{-1}(\mathbf{x}^t - \mathbf{m}_i)\right]}{\sum_j \pi_j |\mathbf{S}_j|^{-0,5} \exp\left[-0,5(\mathbf{x}^t - \mathbf{m}_j)^\top \mathbf{S}_j^{-1}(\mathbf{x}^t - \mathbf{m}_j)\right]}$$

- Étape M : évaluation de $\Phi = \{\pi_i, \mathbf{m}_i, \mathbf{S}_i\}_{i=1}^K$

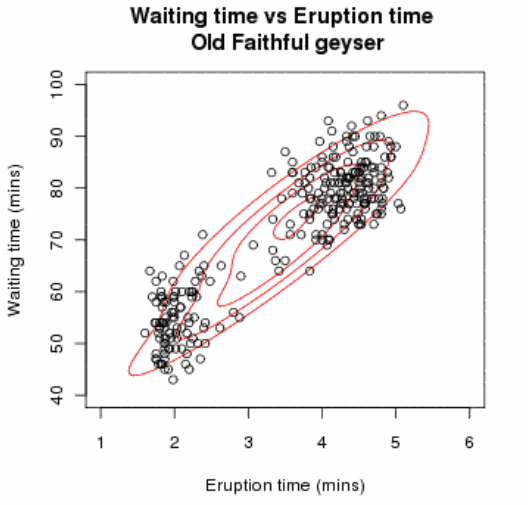
$$\begin{aligned}
\pi_i &= \frac{\sum_t h_i^t}{N} \\
\mathbf{m}_i &= \frac{\sum_t h_i^t \mathbf{x}^t}{\sum_t h_i^t} \\
\mathbf{S}_i &= \frac{\sum_t h_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^\top}{\sum_t h_i^t}
\end{aligned}$$

Illustration de l'algorithme EM



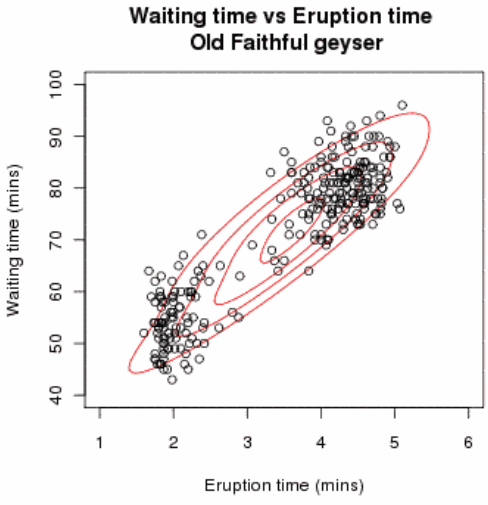
Par Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration de l'algorithme EM



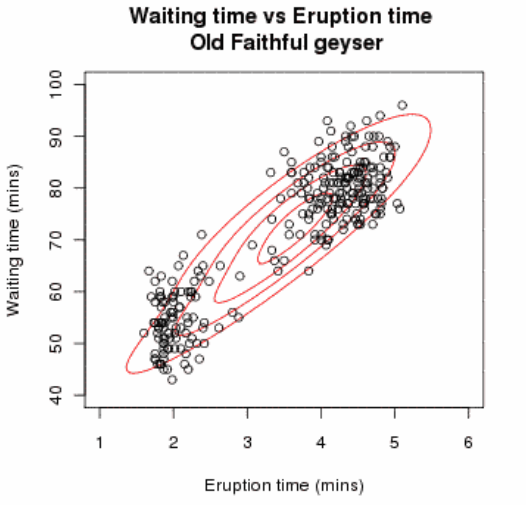
Par Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration de l'algorithme EM



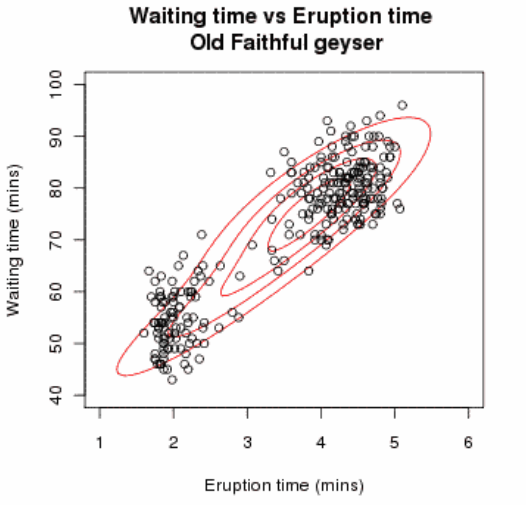
Par Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration de l'algorithme EM



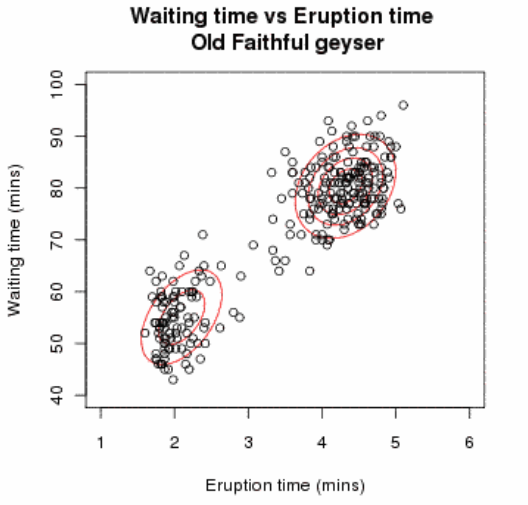
Par Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration de l'algorithme EM



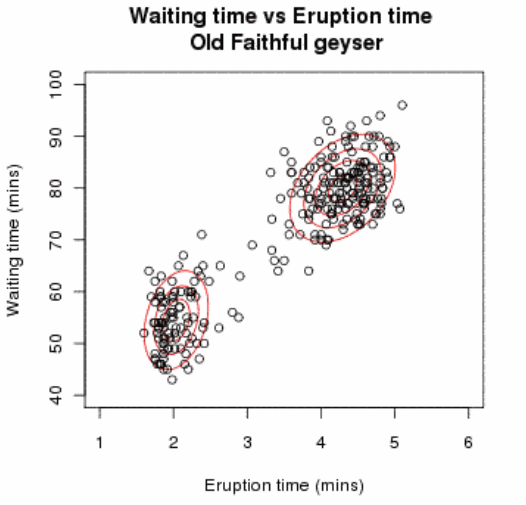
Par Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration de l'algorithme EM



Par Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration de l'algorithme EM



Par Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

13.6 Algorithme EM général

Algorithme EM général

1. Générer une configuration initiale Φ^0
2. Tant que le critère d'arrêt n'est pas atteint, répéter :
   2.1 Étape E : Évaluer les appartenances h_i^t

$$h_i^t = P(\mathcal{G}_i|\mathbf{x}^t,\Phi^l),\ i = 1,\ldots,K,\ t = 1,\ldots,N$$

   2.2 Étape M : Évaluer nouvelle valeur de Φ^{l+1} selon $\mathcal{Q}(\Phi|\Phi^l)$

$$\begin{aligned} \mathcal{Q}(\Phi|\Phi^l) &= \mathbb{E}\left[L(\Phi|\mathcal{X},\mathcal{Z})|\mathcal{X},\Phi^l\right] \\ \Phi^{l+1} &= \underset{\Phi}{\text{argmax}}\, \mathcal{Q}(\Phi|\Phi^l) \end{aligned}$$

3. Retourner le Φ de l'itération finale

Illustration de l'algorithme EM : 2 groupes

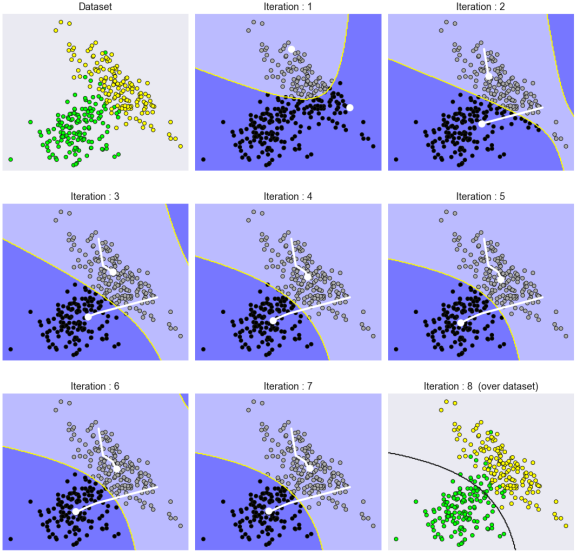
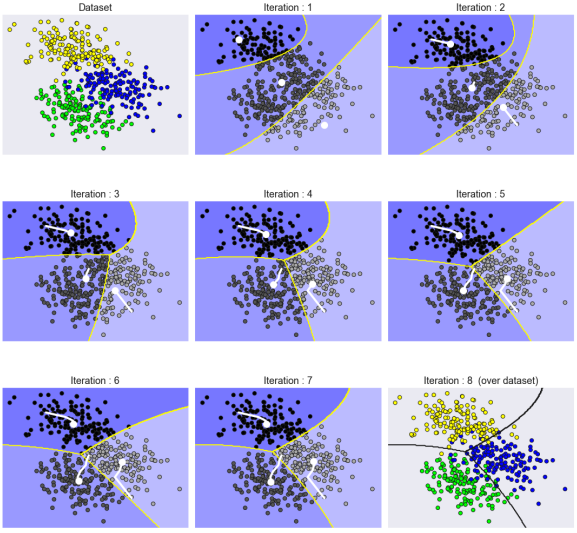


Illustration de l'algorithme EM : 3 groupes



Remarques sur l'algorithme EM

- Initialisation du Φ^0 de l'algorithme avec K-means pour $(\mathbf{x}^t|\mathcal{G}_i,\Phi) \sim \mathcal{N}_D(\mathbf{m}_i,\mathbf{S}_i)$
  - Utiliser estimation des centres par K-means comme $\mathbf{m}_i$ initiaux
  - Calculer matrice de covariance $\mathbf{S}_i$ à partir des associations aux groupes $\mathcal{G}_i$ des données $\mathbf{x}^t$ selon b_i^t obtenus par les K-means
  - Calculer les probabilités a priori selon $\pi_i = \sum_t b_i^t / N$
- Simplifications du modèle à dimensionnalité élevée
  - Partage de la matrice de covariance entre les groupes
  - Matrice de covariance diagonale
  - Matrice de covariance $\sigma\mathbf{I}$

K-means comme algorithme EM

- K-means est un cas particulier de l'algorithme EM
  - Probabilités *a priori* égales pour tous les groupes, $\pi_i = \frac{1}{K}$, $\forall i$.
  - Matrice de covariance partagée $s\mathbf{I}$

$$h_i^t = \frac{\exp\left[-0{,}5s^{-2}\|\mathbf{x}^t - \mathbf{m}_i\|^2\right]}{\sum_j \exp\left[-0{,}5s^{-2}\|\mathbf{x}^t - \mathbf{m}_j\|^2\right]}$$

  - Associations $b_i^t \in \{0,1\}$ sont une version « dure » des $h_i^t \in [0,1]$

$$b_i^t = \begin{cases} 1 & \text{pour } i = \text{argmax}_j\, h_j^t \\ 0 & \text{autrement} \end{cases}$$

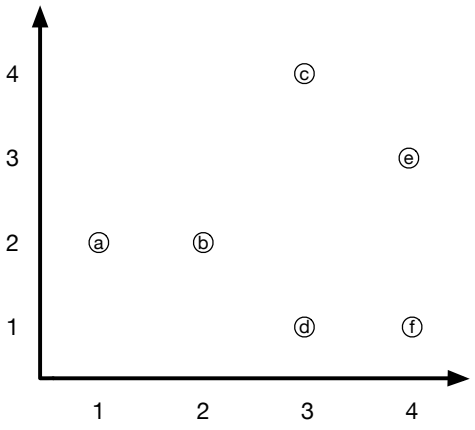
- K-means utilise des densités de probabilité circulaires, alors que EM avec loi normale multivariée utilise des ellipses de forme et orientation quelconques

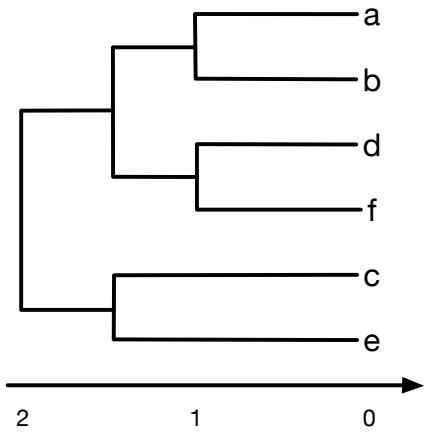
13.7 Clustering hiérarchique

Clustering hiérarchique

- Agglomerations itératives des données
  1. Démarrer avec N groupes, un par observation
  2. Agglomérer les deux groupes les plus similaires et recalculer le centre moyen
  3. Répéter jusqu'à ce qu'un seul groupe soit obtenu
- Divisions itératives des données
  1. Démarrer avec un seul groupe
  2. Diviser en deux groupes les plus différents possibles
  3. Répéter jusqu'à ce que N groupes soient obtenus
- Mesures de similarité pour clustering agglomeratif
  - Clustering en lien simple $d(\mathcal{G}_i,\mathcal{G}_j) = \min_{\mathbf{x}^r\in\mathcal{G}_i,\mathbf{x}^s\in\mathcal{G}_j} D(\mathbf{x}^r,\mathbf{x}^s)$
  - Clustering en lien complet $d(\mathcal{G}_i,\mathcal{G}_j) = \max_{\mathbf{x}^r\in\mathcal{G}_i,\mathbf{x}^s\in\mathcal{G}_j} D(\mathbf{x}^r,\mathbf{x}^s)$

Exemple de clustering hiérarchique





Utilisations du clustering

- Exploration de la structure des données
  - Découvrir des similarités dans les données
  - Organiser les données par groupes similaires
- Experts peuvent nommer ces groupes selon les concepts qu'ils représentent
  - Un concept peut être représenté par différents groupes
- Prétraitement des données
  - Projection dans l'espace des h_i
  - Discrimination dans l'espace des h_i
- Mélange de densités-mélanges pour classement

$$p(\mathbf{x}|C_i) = \sum_{j=1}^{K_i} p(\mathbf{x}|\mathcal{G}_{i,j})P(\mathcal{G}_{i,j})$$

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x}|C_i)P(C_i)$$

Choix du nombre de groupes

- Le choix du nombre de groupes est un paramètre crucial. Comment le déterminer ?
  - Certaines applications l'imposent naturellement
    - Dans l'exemple de la *colormap*, on veut $k = 256$ groupes (couleurs)
  - Tracer les données en 2D, à l'aide d'une ACP, peut permettre d'identifier le nombre de groupes naturels dans les données
  - Un algorithme incrémental peut ajouter dynamiquement des centres, selon un certain critère
  - Vérification/validation des groupes par des experts peut permettre de déterminer si le nombre de groupes est approprié
    - Inspection visuelle d'images
    - Analyse des prototypes de groupes

13.8 Clustering dans scikit-learn

Scikit-learn : K-means

- `cluster.KMeans` : algorithme des K-means
  - Paramètres
    - `n_clusters` (int) : nombre de clusters (défaut : 8)
    - `max_iter` (int) : nombre d'itérations maximal (défaut : 300)
    - `n_init` (int) : nombre de répétitions, la meilleurs solution selon l'*inertie* est conservée (défaut : 10)
    - `init` (string ou ndarray) : initialisation de l'algorithme, `'k-means++'` pour approche « intelligente », `'random'` pour initialisation aléatoire, utiliser un ndarray pour valeurs fournies
    - `tol` (float) : tolérance sur l'inertie avant de déclarer une convergence
  - Attributs
    - `cluster_centers_` (array) : valeurs des centres, $\mathbf{m}_i$ (taille $N \times D$)
    - `labels_` (array) : étiquettes des données, b_i^t
    - `inertia_` (float) : valeur de l'inertie, soit $\sum_t \sum_i b_i^t (\mathbf{x}^t - \mathbf{m}_i)$

Scikit-learn : algorithme EM

- `mixture.GaussianMixture` : EM avec distributions normales multivariées
  - Paramètres
    - `n_components` (int) : nombre de clusters (défaut : 1)
    - `covariance_type` (string) : type de matrice de covariance (défaut : `'full'`)
      - `'full'` : matrices de covariance complètes et distinctes
      - `'tied'` : matrice de covariance complète et partagées
      - `'diag'` : matrices de covariance diagonales et distinctes
      - `'spherical'` : matrices isotropiques ($\mathbf{\Sigma} = \sigma \mathbf{I}$) et distinctes
    - `max_iter` (int) : nombre maximum d'itérations (défaut : 100)
    - `n_init` (int) : nombre de répétitions, la meilleure solution est conservée (défaut : 1)
    - `init_params` (string) : méthode d'initialisation, avec K-means (`'kmeans'`) ou aléatoirement (`'random'`) (défaut : `'kmeans'`)
  - Attributs
    - `weights_` (array) : probabilités a priori de chaque cluster, $P(\mathcal{G}_i)$ (vecteur de taille K)
    - `means_` (array) : vecteurs moyens des clusters (taille $K \times D$)
    - `covariance_` (array) : matrices de covariance

Scikit-learn : clustering hiérarchique

- `cluster.AgglomerativeClustering` : clustering hiérarchique agglomératif
  - Paramètres
    - `n_clusters` (int) : nombre de clusters à trouver (défaut : 2)
    - `affinity` (string ou callable) : mesure d'affinité à utiliser, peut être `'euclidean'`, `'l1'`, `'l2'`, `'manhattan'`, `'cosine'` ou `'precomputed'` (défaut : `'euclidean'`)
    - `'linkage'` (string) : critère de distance entre les clusters (défaut : `'ward'`)
      – `'ward'` : minimiser la variance des clusters agglomérés
      – `'complete'` : en lien complet, maximum de la distance entre deux paires de deux clusters
      – `'average'` : moyenne des distances entre les paires de clusters
  - Attributs
    - `labels_` (array) : étiquettes de clustering
    - `n_leaves_` (int) : nombre de feuilles dans le dendrogramme
    - `children_` (array) : structure du dendrogramme