Clustering

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

Professor: Christian Gagné

Week 13

13.1 Vector quantization

Clustering

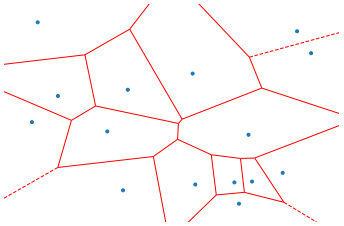
- Supervised learning
  - Class labels available
  - Parametric methods: observations follow a given probability density $p(\mathbf{x}|C_i)$
- One group of data per class
  - According to a normal distribution, mean and covariance law shared by all data from the same class
  - In practice, the data of a class can fit in several groups
    - Cursive writing: different ways of doing 1's and 7's
    - Detecting intrusions in a computer system
- Clustering
  - Identifying "natural" groups in the data

Vector quantization

- Vector quantization
  - Discretize a space $\mathbb{R}^D$, by partitionning it into K regions
- Possible quantization using K reference vectors $\mathbf{m}_i$
  - Assignment of a data $\mathbf{x}^t$ according to the nearest reference vector

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

- Partitioning of the space according to a Voronoi diagram

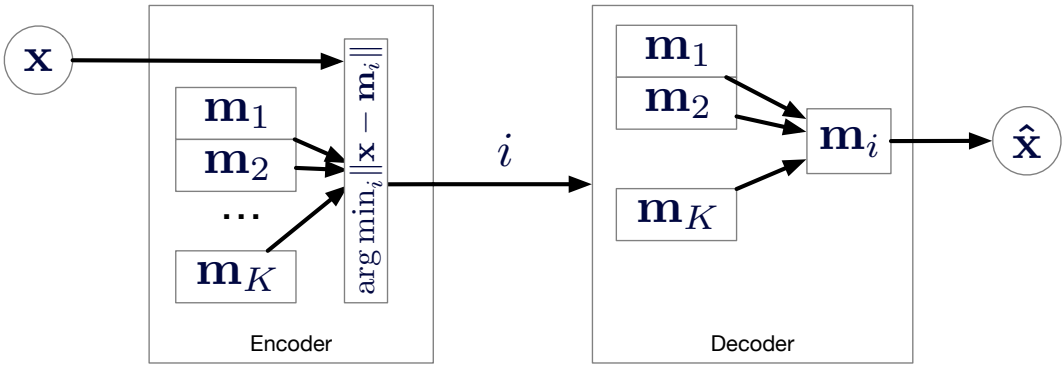


Compression and reconstruction

- Complete compression of space $\mathbb{R}^D$ in K reference vectors $\mathbf{m}_i$
  - Each point in the input space is associated to one of the reference vectors (discrete values)
- *Colormap* example
  - Colour of a pixel in an image: 24 bits
  - Transmit image of 640×400 pixels: more than 6M bits
  - Compression with a *colormap* of 256 different colours
    - The *colormap* fits on 6144 bits
    - Pixels refer to the *colormap*: 8 bits per pixel
    - Image encoded on 2M bits, so, it is a gain of $3:1$.
  - Loss of information if more than 256 different colours in the image
    - Choice of colours minimizing a certain criterion
- Reconstruction error

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

Compression by clustering



13.2 K-means

K-means

- Calculation of the optimum reconstruction error $E(\{\mathbf{m}_i\}_{i=1}^K|\mathcal{X})$ according to the $\mathbf{m}_i$ is impossible analytically
  - Optimal position of the centres $\mathbf{m}_i$ depends on the labels b_i^t
  - Optimal choice of labels b_i^t depends on the position of the centres $\mathbf{m}_i$!
- Iterative resolution, by successive approximations of b_i^t and $\mathbf{m}_i$
  - Estimate $b_i^t(j+1)$ according to $\mathbf{m}_i(j)$
  - Estimate $\mathbf{m}_i(j+1)$ according to $b_i^t(j+1)$
  - Repeat until convergence or resources depletion

Estimation of the centres

- Estimated centres $\mathbf{m}_i$ according to the labels b_i^t
  - $\mathbf{m}_i$ with partial derivative of $E(\{\mathbf{m}_i\}_{i=1}^K|\mathcal{X})$ according to $\mathbf{m}_j$

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

K-means algorithm

1. Initialize centres $\mathbf{m}_i$ randomly
2. As long as the stop criterion is not met, repeat:
   2.1 Estimate data labels b_i^t according to the positions of the centres $\mathbf{m}_i$

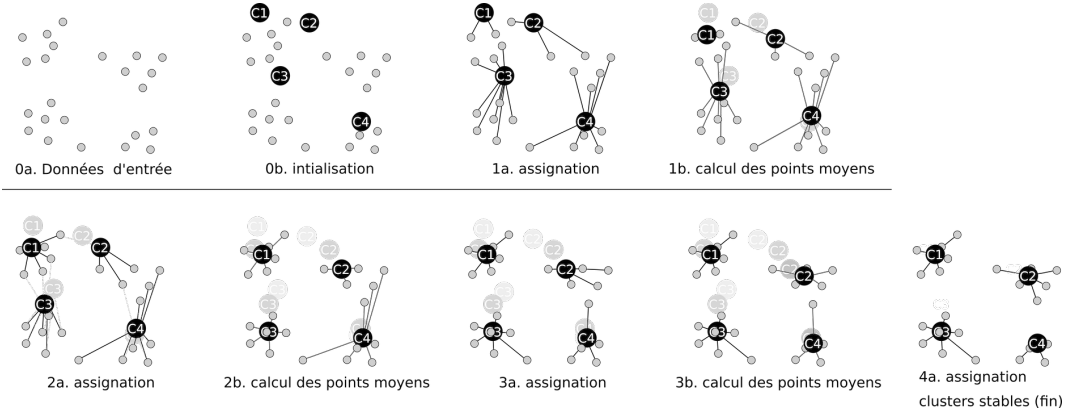
$$b_i^t = \begin{cases} 1 & i = \text{argmin}_j \|\mathbf{x}^t - \mathbf{m}_j\| \\ 0 & \text{otherwise} \end{cases}, \; i = 1, \ldots, K, \; t = 1, \ldots, N$$

   2.2 Optimize the position of the centres $\mathbf{m}_i$ with the new labels 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. Return centre values $\mathbf{m}_i$

Illustration of K-means



Initialization and stop criteria

- Possible approaches to initializing centres $\mathbf{m}_i$
  - Randomly select K instances of $\mathcal{X}$
  - Calculate the mean vector of all the data and initialize K centres around this mean, with slight random variations for each centre
  - Based on the principal component
    1. Calculate the principal component
    2. Project the data on the corresponding line
    3. Partition the data on the line into K groups of equal size
    4. Calculate the average of each of these groups in the space of origin and use them as starting centres
- Stop criteria
  - Maximum number of iterations
  - Variation of the position of the centres is below a given threshold

K-means properties

- No guarantee of convergence towards the global optimum
  - Outcome depends on the choice of the initial positions of the centres
- Relatively fast convergence
- Number of centres to be used fixed in advance
  - Requires knowledge of the number of groups forming the data
  - If number of groups is unknown, empirically determine K
  - *Leader cluster* algorithm: incremental addition of centres when the distance of a data to its centre exceeds a threshold
  - Variation: add a centre when the number of data associated to a centre exceeds a threshold

Illustration of K-means: 2 groups

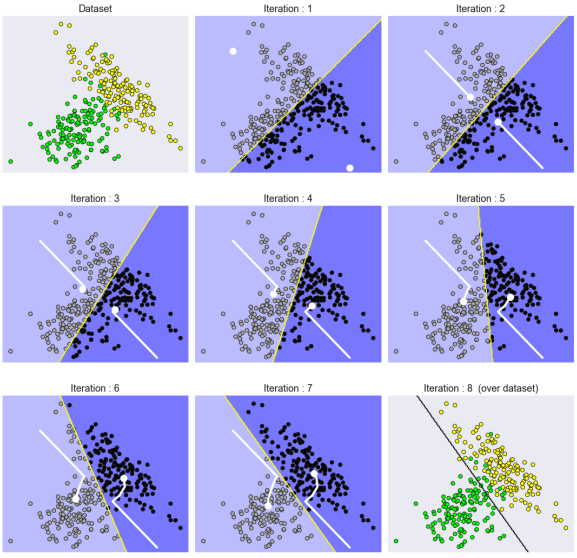
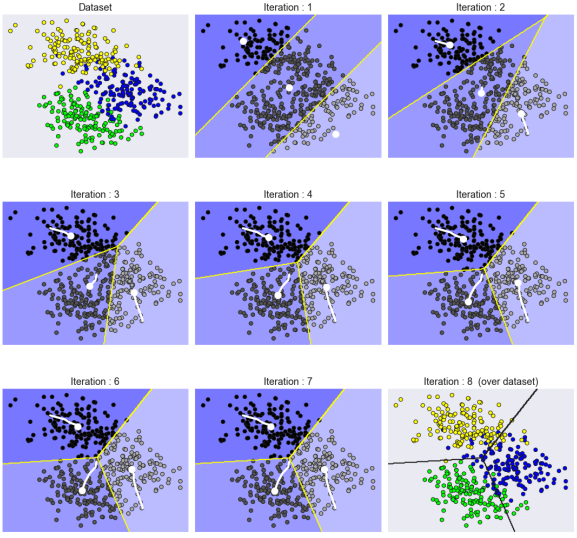
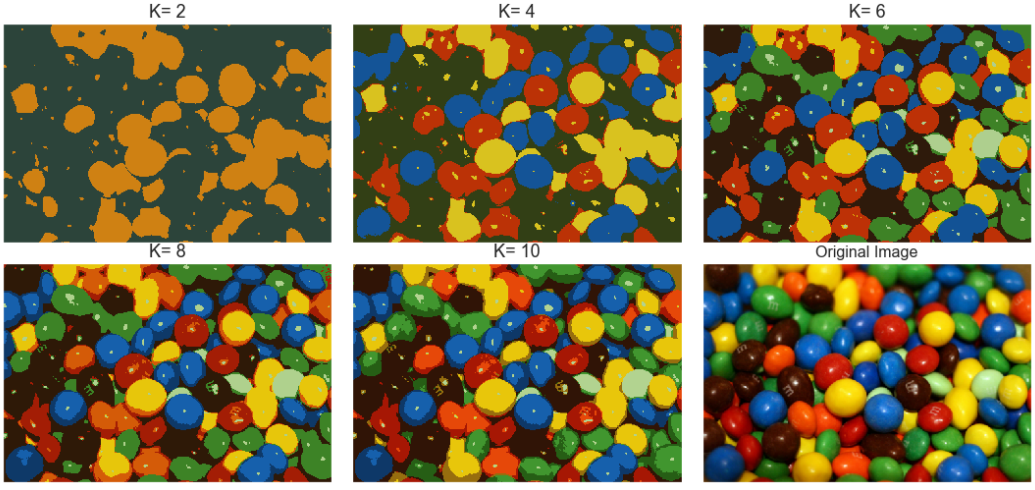


Illustration of K-means: 3 groups



Application: colormap compression



13.3 Mixture density

Mixture density

- Mixture density: combination of density laws associated with several groups

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

- Direct link with the supervised case
  - Similar formulation, but groups are known and identified in the supervised case
  - Can be used with parametric methods, when there are many groups in each class
- Mixture of components according to a multivariate normal law
  - Component density: $(\mathbf{x}|\mathcal{G}_i) \sim \mathcal{N}_D(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
  - Parametrization: $\Phi = \{P(\mathcal{G}_i), \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{K}$
- Uses unlabeled samples, $\mathcal{X} = \{\mathbf{x}^t\}_{t=1}^{N}$

Mixture density probabilities

- Mixture density

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

- Proportion of the group $\mathcal{G}_i$ in the mixture, $P(\mathcal{G}_i)$

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

- Probability that $\mathbf{x}$ belongs to the group $\mathcal{G}_i$, $P(\mathcal{G}_i|\mathbf{x})$

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

Hidden indicator variables

- Hidden indicator variables $\mathbf{z}^t = \{z_1^t, \ldots, z_K^t\}$
  - z_i^t: association of the data $\mathbf{x}^t$ with the group $\mathcal{G}_i$
  - We don't know the "real" values of the $\mathcal{Z}$: hidden variables of the problem
  - Simplification of the notation: $\pi_i = P(\mathcal{G}_i)$
  - Multinomial distribution: $z_i^t = 1$ indicates that variable $\mathbf{x}^t$ belongs to the group $\mathcal{G}_i$, and $z_i^t = 0$ otherwise

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

- Likelihood of observation of $\mathbf{x}^t$

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

- Joint probability $p(\mathbf{x}^t,\mathbf{z}^t)$

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

Likelihood Function

- Log-likelihood function of the parametrization Φ according to the association of the data of $\mathcal{X}$ to the groups given by $\mathcal{Z}$

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

13.4 Expectation–maximization algorithm

Expectation–maximization algorithm

- Membership $h_i^t \equiv P(\mathcal{G}_i|\mathbf{x}^t,\Phi)$: association to a group $\mathcal{G}_i$ of a data $\mathbf{x}^t$ according to the parametrization Φ (hidden variable observation $\mathbf{z}^t$)
- Log-likelihood depends on the parametrization Φ according to the association of hidden variables $\mathcal{Z}$
  - Similarly, the association of the hidden variables $\mathcal{Z}$ depends on parametrization Φ
  - We don't know the real $\mathcal{Z}$ (hidden random variables): optimization of the **likelihood expectation**
  - Optimization of the analytical equation is impossible: iterative approach
- Expectation–maximization algorithm (EM)
  - E-step: calculation of the expectation of associations to groups $h_i^t \equiv P(\mathcal{G}_i|\mathbf{x}^t,\Phi)$ with current Φ parametrization
  - M-step: get new parametrization Φ^{l+1} maximizing the likelihood expectation $\mathcal{Q}(\Phi|\Phi^l)$

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

E-step

- Given Φ^l, what is the likelihood expectation of other possible Φ parametrizations?

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

- Label expectation $\mathbb{E}[z_i^t|\mathcal{X},\Phi^l]$ given by:

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

Likelihood expectation

- Interpretation of h_i^t
  - $h_i^t \equiv \mathbb{E}[z_i^t|\mathcal{X},\Phi^l] = P(\mathcal{G}_i|\mathbf{x}^t,\Phi^l)$ gives the a posteriori probability that $\mathbf{x}^t$ belongs to the group $\mathcal{G}_i$
  - Probabilistic observation of the hidden variable z_i^t
  - Reinterpretation of a discriminant for clustering
  - h_i^t is a relaxed version of the b_i^t binary membership of K-means
- Resulting likelihood expectation

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

M-step

- M-step: find a new parametrization Φ^{l+1} maximizing the likelihood expectation $\mathcal{Q}(\Phi|\Phi^l)$

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

- Maximum where partial derivatives are equal to zero
  - π_i is a probability, therefore $\sum_i \pi_i = 1$, resolution with Lagrange's method

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

  - Resolution of Φ specific to the probability law

Solving the a priori probabilities π_i

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

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

13.5 EM algorithm for multivariate normal distribution

EM algorithm for multivariate normal distribution

- Specific instance of the EM algorithm, $(\mathbf{x}^t|\mathcal{G}_i,\Phi) \sim \mathcal{N}_D(\mathbf{m}_i,\mathbf{S}_i)$
- Solving $\mathbf{m}_j$ for $\Phi = \{\pi_i,\mathbf{m}_i,\mathbf{S}_i\}_{i=1}^K$

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{m}_j} \sum_t \sum_i h_i^t \log \frac{1}{(2\pi)^{0.5D}|\mathbf{S}_i|^{0.5}} \exp\left[-\frac{1}{2}(\mathbf{x}^t-\mathbf{m}_i)^\top \mathbf{S}_i^{-1}(\mathbf{x}^t-\mathbf{m}_i)\right] &= 0 \\
\frac{\partial}{\partial \mathbf{m}_j} \sum_t \sum_i h_i^t (\mathbf{x}^t-\mathbf{m}_i)^\top \mathbf{S}_i^{-1}(\mathbf{x}^t-\mathbf{m}_i) &= 0 \\
\sum_t h_j^t (\mathbf{x}^t-\mathbf{m}_j)(-1) &= 0 \\
\sum_t h_j^t \mathbf{x}^t &= \mathbf{m}_j \sum_t h_j^t \\
\mathbf{m}_j &= \frac{\sum_t h_j^t \mathbf{x}^t}{\sum_t h_j^t}
\end{aligned}
$$

Solving $\mathbf{S}_j$

- Solving $\mathbf{S}_j$ for $\Phi = \{\pi_i, \mathbf{m}_i, \mathbf{S}_i\}_{i=1}^K$

$$\frac{\partial}{\partial \mathbf{S}_j} \sum_t \sum_i h_i^t \log \frac{1}{(2\pi)^{0.5D} |\mathbf{S}_i|^{0.5}} \exp\left[-\frac{1}{2}(\mathbf{x}^t - \mathbf{m}_i)^\top \mathbf{S}_i^{-1}(\mathbf{x}^t - \mathbf{m}_i)\right] = 0$$

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

- Solving $\mathbf{S}_j$ is subtle, requires the spectral theorem
  - For more details, see:
    `http://en.wikipedia.org/wiki/Estimation_of_covariance_matrices`

Summary of EM algorithm for multivariate normal law

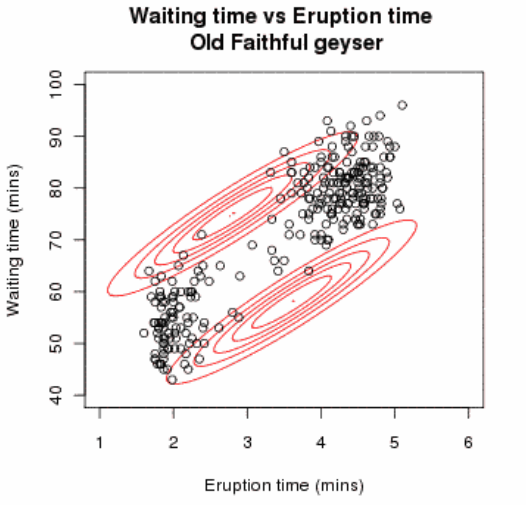
- E-step: evaluation of $h_i^t,\ i = 1,\ldots,K,\ t = 1,\ldots,N$

$$h_i^t = \frac{\pi_i |\mathbf{S}_i|^{-0.5} \exp\left[-0.5(\mathbf{x}^t - \mathbf{m}_i)^\top \mathbf{S}_i^{-1}(\mathbf{x}^t - \mathbf{m}_i)\right]}{\sum_j \pi_j |\mathbf{S}_j|^{-0.5} \exp\left[-0.5(\mathbf{x}^t - \mathbf{m}_j)^\top \mathbf{S}_j^{-1}(\mathbf{x}^t - \mathbf{m}_j)\right]}$$

- M-step: evaluation of $\Phi = \{\pi_i, \mathbf{m}_i, \mathbf{S}_i\}_{i=1}^K$

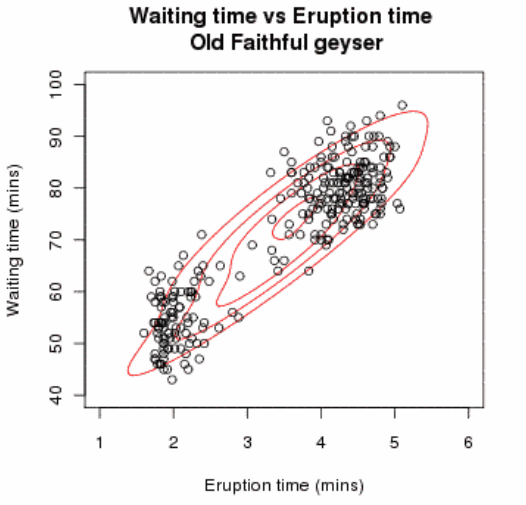
$$\begin{aligned}
\pi_i &= \frac{\sum_t h_i^t}{N} \\
\mathbf{m}_i &= \frac{\sum_t h_i^t \mathbf{x}^t}{\sum_t h_i^t} \\
\mathbf{S}_i &= \frac{\sum_t h_i^t (\mathbf{x}^t - \mathbf{m}_i)(\mathbf{x}^t - \mathbf{m}_i)^\top}{\sum_t h_i^t}
\end{aligned}$$

Illustration of the EM algorithm



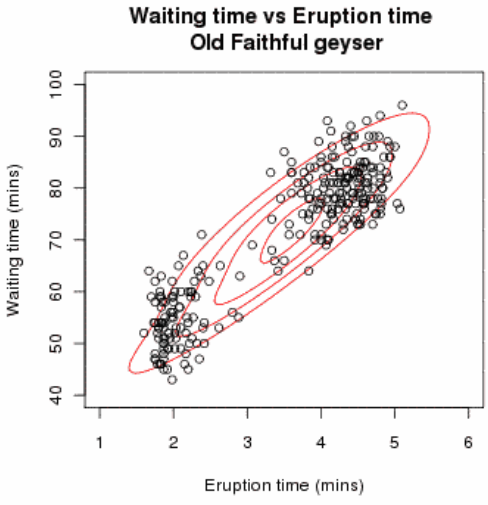
By Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration of the EM algorithm



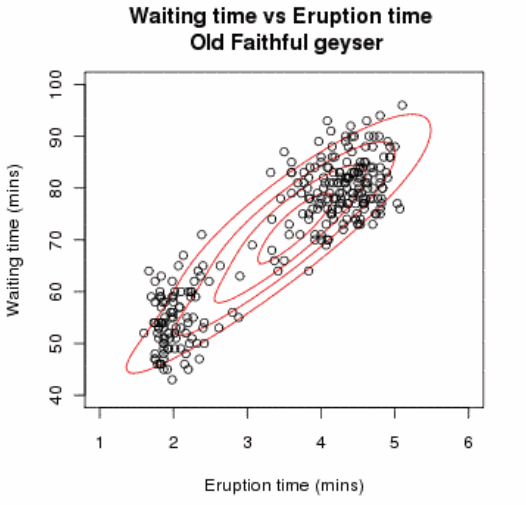
By Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration of the EM algorithm



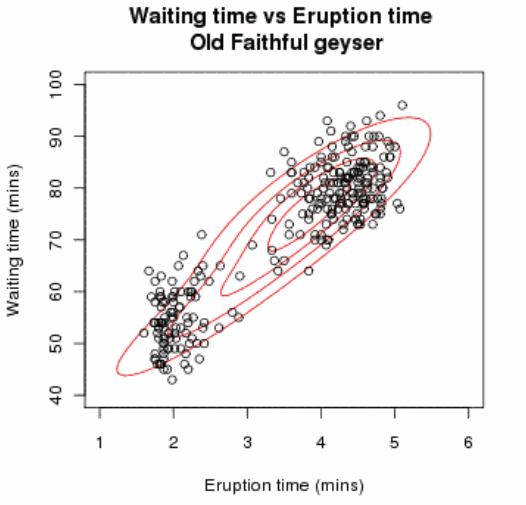
By Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration of the EM algorithm



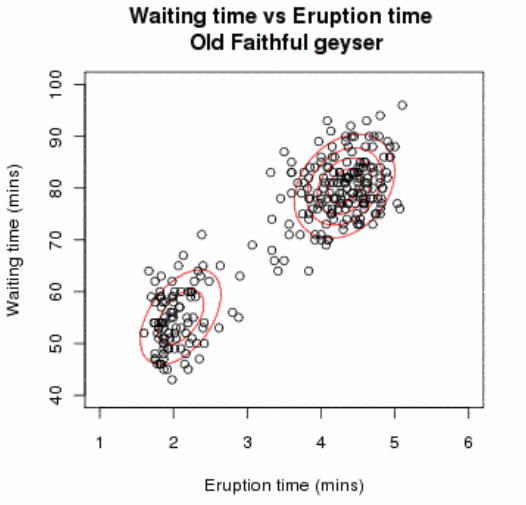
By Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration of the EM algorithm



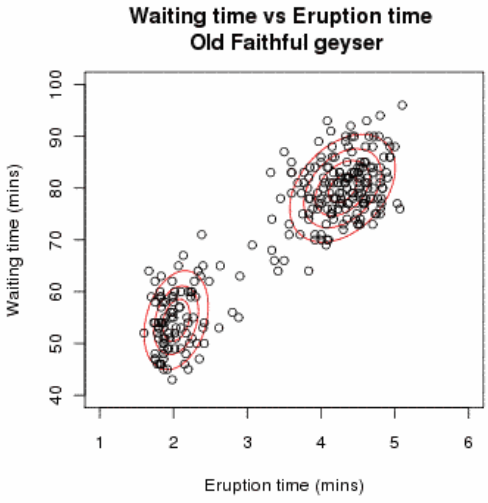
By Mquantin, CC-BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Em_old_faithful.gif.

Illustration of the EM algorithm



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

Illustration of the EM algorithm



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

13.6 General EM algorithm

General EM algorithm

1. Generate an initial configuration Φ^0
2. As long as the stop criterion is not reached, repeat:
   2.1 E-step: Assessing membership 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 M-step: Evaluate new value of Φ^{l+1} according to $\mathcal{Q}(\Phi|\Phi^l)$

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

3. Return the Φ of the final iteration

Illustration of the EM algorithm: 2 groups

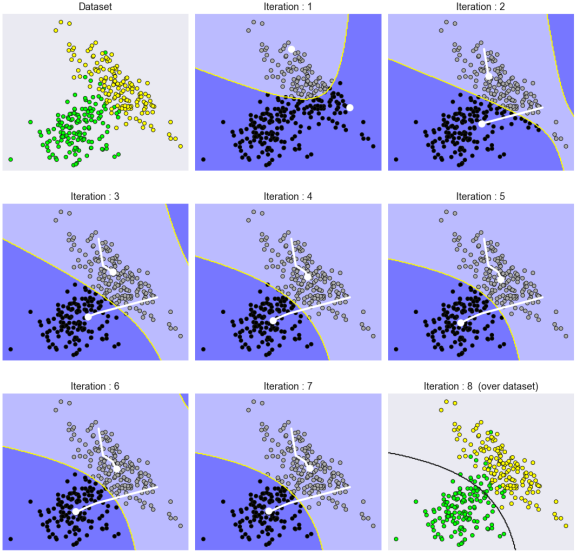
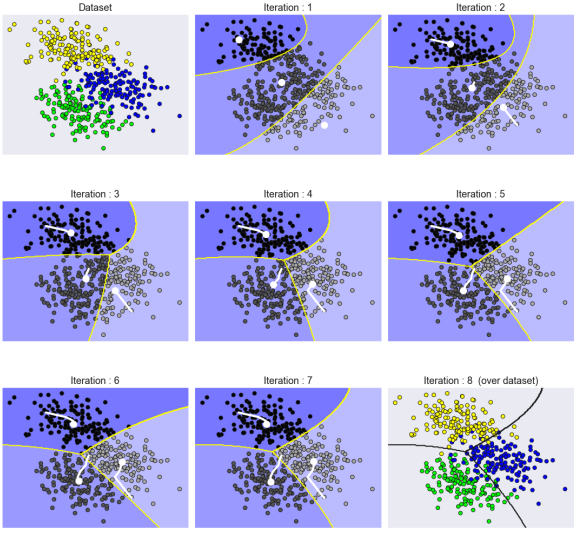


Illustration of the EM algorithm: 3 groups



Notes on the EM algorithm

- Initialization of Φ^0 for the algorithm with K-means when $(\mathbf{x}^t|\mathcal{G}_i,\Phi) \sim \mathcal{N}_D(\mathbf{m}_i,\mathbf{S}_i)$
  - Estimate the centres with K-means for the initial $\mathbf{m}_i$
  - Compute covariance matrix $\mathbf{S}_i$ from associations to groups $\mathcal{G}_i$ of data $\mathbf{x}^t$ according to b_i^t obtained with K-means
  - Calculate the a priori probabilities according to $\pi_i = \sum_t b_i^t / N$
- High dimensional model simplifications
  - Sharing the covariance matrix between groups
  - Diagonal covariance matrix
  - Covariance matrix $\sigma\mathbf{I}$

K-means as EM algorithm

- K-means is a specific case of the EM algorithm
  - *A priori* probabilities equal for all groups, $\pi_i = \frac{1}{K}$, $\forall i$.
  - Shared covariance matrix $s\mathbf{I}$

$$h_i^t = \frac{\exp\left[-0.5s^{-2}\|\mathbf{x}^t - \mathbf{m}_i\|^2\right]}{\sum_j \exp\left[-0.5s^{-2}\|\mathbf{x}^t - \mathbf{m}_j\|^2\right]}$$

  - Associations $b_i^t \in \{0,1\}$ are a "hard" version of $h_i^t \in [0,1]$

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

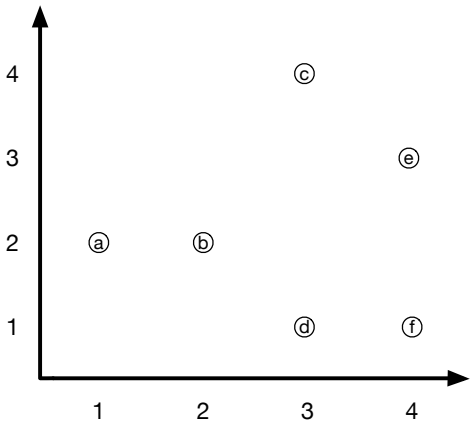
- K-means uses circular probability densities, while EM with multivariate normal distribution uses ellipses of any shape and orientation

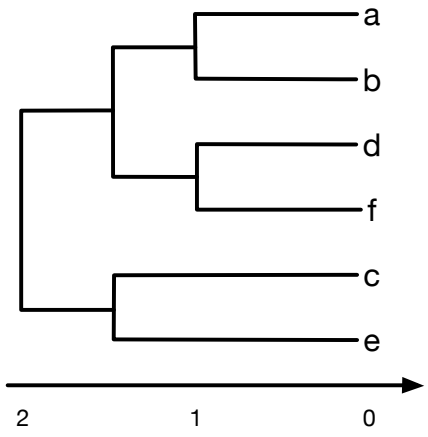
13.7 Hierarchical clustering

Hierarchical clustering

- Iterative data agglomerations
  1. Start with N groups, one per observation
  2. Combine the two most similar groups and recalculate the mean centre
  3. Repeat until only one group is obtained
- Iterative data divisions
  1. Start with one group
  2. Divide into two groups as different as possible
  3. Repeat until N groups are obtained
- Similarity measurement for clustering agglomerative clustering
  - Single-linkage clustering $d(\mathcal{G}_i,\mathcal{G}_j) = \min_{\mathbf{x}^r \in \mathcal{G}_i, \mathbf{x}^s \in \mathcal{G}_j} D(\mathbf{x}^r,\mathbf{x}^s)$
  - Complete-linkage clustering $d(\mathcal{G}_i,\mathcal{G}_j) = \max_{\mathbf{x}^r \in \mathcal{G}_i, \mathbf{x}^s \in \mathcal{G}_j} D(\mathbf{x}^r,\mathbf{x}^s)$

Example of hierarchical clustering





Clustering utilisation

- Exploring data structure
  - Discovering similarities in the data
  - Organize the data into similar groups
- Experts can name these groups according to the concepts they represent
  - A concept can be represented by different groups
- Data preprocessing
  - Projection in the h_i space
  - Discrimination in the h_i space
- Mixture density for classification

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

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

Choosing the number of groups

- The choice of the number of groups is a crucial parameter, how to determine it?
  - Some applications impose it naturally
    - In the example of the *colormap*, we want $k = 256$ groups (colours)
  - Plotting the data in 2D, using PCA, can help identify the number of natural groups in the data
  - An incremental algorithm can dynamically add centres, according to a certain criterion
  - Expert verification/validation of groups can help determine if the number of groups is appropriate
    - Visual image inspection
    - Analysis of group prototypes

13.8 Clustering in scikit-learn

Scikit-learn: K-means

- `cluster.KMeans`: K-means algorithm
  - Parameters
    - `n_clusters` (int): number of clusters (default: 8)
    - `max_iter` (int): maximum number of iterations (default: 300)
    - `n_init` (int): number of repetitions, the best solution according to *inertia* is kept (default: 10)
    - `init` (string or ndarray): initialization of the algorithm, `'k-means++'` for "intelligent" approach, `'random'` for random initialization, use a ndarray for given values
    - `tol` (float): tolerance on inertia before declaring convergence
  - Attributes
    - `cluster_centers_` (array): centre values, $\mathbf{m}_i$ (size $N \times D$)
    - `labels_` (array): data labels, b_i^t
    - `inertia_` (float): value of inertia, which is $\sum_t \sum_i b_i^t (\mathbf{x}^t - \mathbf{m}_i)$

Scikit-learn: EM algorithm

- `mixture.GaussianMixture`: EM with multivariate normal distributions
  - Parameters
    - `n_components` (int): number of clusters (default: 1)
    - `covariance_type` (string): type of covariance matrix (default: `'full'`)
      - `'full'`: complete and distinct covariance matrices
      - `'tied'`: complete and shared covariance matrix
      - `'diag'`: diagonal and distinct covariance matrices
      - `'spherical'`: isotropic and distinct matrices ($\mathbf{\Sigma} = \sigma \mathbf{I}$)
    - `max_iter` (int): maximum number of iterations (default: 100)
    - `n_init` (int): number of repetitions, the best solution is kept (default: 1)
    - `init_params` (string): initialization method, with K-means (`'kmeans'`) or randomly (`'random'`) (default: `'kmeans'`)
  - Attributes
    - `weights_` (array): a priori probabilities of each cluster, $P(\mathcal{G}_i)$ (vector of size K)
    - `means_` (array): average vectors of the clusters (size $K \times D$)
    - `covariance_` (array): covariance matrices

Scikit-learn: hierarchical clustering

- `cluster.AgglomerativeClustering`: hierarchical agglomerative clustering
  - Parameters
    - `n_clusters` (int): number of clusters to find (default: 2)
    - `affinity` (string or callable): affinity measure to use, can be `'euclidean'`, `'l1'`, `'l2'`, `'manhattan'`, `'cosine'` or `'precomputed'` (default: `'euclidean'`)
    - `'linkage'` (string): distance criterion between clusters (default: `'ward'`)
      - `'ward'`: minimize the variance of agglomerated clusters
      - `'complete'`: in complete-linkage, maximum of the distance between two pairs of two clusters
      - `'average'`: average of the distances between the cluster pairs
  - Attributes
    - `labels_` (array): clustering labels
    - `n_leaves_` (int): number of leaves in the dendrogram
    - `children_` (array): structure of the dendrogram