ICT for Health
Clustering

Monica Visintin

Politecnico di Torino

2016/17

Table of Contents

Table of Contents

The K-means algorithm

The assumption is that

$$\mathbf{y}(n) = \mathbf{x}(n) + \nu(n)$$

where $\mathbf{x}(n)$ is one among the set $\mathbf{x}_1, \ldots, \mathbf{x}_K$. Each vector has dimension M. We measured $\mathbf{y}(n)$ for $n = 1, \ldots, N$, we don't know the vectors $\mathbf{x}_k$, $k = 1, \ldots, K$ and we have to guess them: **K-means clustering**

1. we start with an initial guess of $\mathbf{x}_k(0)$, $k = 1, \ldots, K$,
2. at the i-th step we associate to $\mathbf{x}_k(i)$ the points $\mathbf{y}(n)$ which are closer to $\mathbf{x}_k(i)$ than to the other points $\mathbf{x}_h(i)$ (**assignement step**)
3. we evaluate the mean value $\mathbf{m}_k(i)$ of the points $\mathbf{y}(n)$ that have been associated with $\mathbf{x}_k(i)$ (**update step**)
4. we define $\mathbf{x}_k(i+1) = \mathbf{m}_k(i)$, we set $i := i + 1$, we go back to step 2 until the convergence of the algorithm

Some problems with the K-means algorithm

Clustering is different from classification in that the values of $\mathbf{x}_k$ $k = 1, \ldots, K$ are not known (not even K is known, typically). The basic clustering algorithm just described has some problems.

- The K-means algorithm gives different results if the initial guess of $\mathbf{x}_k$ is changed. Should we try several times and then pick the best solution? but what is the best solution?
- In artificial experiments, the K-means algorithm sometimes fails (we generate $\mathbf{x}(n)$ taking it from a known set $\{\mathbf{x}_k\}_{k=1}^K$, we generate $\mathbf{y}(n)$, but then we apply the K-means algorithm as if we did not know which $\mathbf{x}_k$ corresponds to each $\mathbf{y}(n)$, just to check that the algorithm works, and in some cases the values of $\mathbf{x}_k$ found by the K-means algorithm are wrong)
- The performance strictly depends on the specific set of points $\mathbf{y}(n), n = 1, \ldots, N$

Table of Contents

A more accurate model for clustering

We can **model** the clustering problem as follows:

- Vector $\mathbf{y}(n) \in \mathbb{R}^M$ is equal to a vector $\mathbf{x}(n)$, taken from the known set $\mathcal{X} = \{\mathbf{x}_k\}_{k=1}^K$, plus a **Gaussian noise** vector $\boldsymbol{\nu}(n)$: $\mathbf{y}(n) = \mathbf{x}(n) + \boldsymbol{\nu}(n)$; we assume that the components of $\boldsymbol{\nu}(n)$ are statistically independent
- When $\mathbf{x}(n) = \mathbf{x}_k$, then $\boldsymbol{\nu}(n)$ is a Gaussian vector with mean zero and variance σ_k^2
- Once we get $\mathbf{y}(n)$, we have to test these hypotheses:

$$\mathcal{H}_1 : \mathbf{y}(n) = \mathbf{x}_1 + \boldsymbol{\nu}(n), \quad \mathbb{E}\{\nu_i^2(n)\} = \sigma_1^2, i = 1, \ldots, M$$
$$\mathcal{H}_2 : \mathbf{y}(n) = \mathbf{x}_2 + \boldsymbol{\nu}(n), \quad \mathbb{E}\{\nu_i^2(n)\} = \sigma_2^2, i = 1, \ldots, M$$
$$\vdots$$
$$\mathcal{H}_K : \mathbf{y}(n) = \mathbf{x}_K + \boldsymbol{\nu}(n), \quad \mathbb{E}\{\nu_i^2(n)\} = \sigma_K^2, i = 1, \ldots, M$$

and choose “the best” hypothesis (but what is the meaning of “the best”?). Note that in the clustering problem, we do not know $\mathcal{X} = \{\mathbf{x}_k\}_{k=1}^K$ and we don’t know the values of σ_k^2.

The hard K-means algorithm

Remember that we don't know $\mathbf{x}_k$ nor σ_k^2 nor π_k.

1. Start with an initial set of vectors $\mathbf{x}_k(0)$, variances $\sigma_k^2(0)$ (all equal to 1), prior probabilities $\pi_k(0)$ (all equal to $1/K$); set $i = 0$
2. (**assignment step**) $\mathbf{y}(n)$ is given to $\mathcal{R}_k$ if

$$\frac{\pi_k(i)}{(2\pi\sigma_k^2(i))^{N/2}} e^{-\frac{\|\mathbf{y}(n)-\mathbf{x}_k(i)\|^2}{2\sigma_k^2(i)}} \geq \frac{\pi_j(i)}{(2\pi\sigma_j^2(i))^{N/2}} e^{-\frac{\|\mathbf{y}(n)-\mathbf{x}_j(i)\|^2}{2\sigma_j^2(i)}}, \forall j \neq k$$

3. (**update step**) Select vectors $\mathbf{y}(n)$ that have been given to region $\mathcal{R}_k$; call them $\mathbf{w}_n$, $n = 1, \ldots, N_k$. Then define

$$\pi_k(i+1) = \frac{N_k}{N}, \quad \mathbf{x}_k(i+1) = \frac{1}{N_k} \sum_{n=1}^{N_k} \mathbf{w}_n$$

$$\sigma_k^2(i+1) = \frac{1}{(N_k - 1)M} \sum_{n=1}^{N_k} \|\mathbf{w}_n - \mathbf{x}_k(i+1)\|^2$$

4. set $i := i + 1$, go back to step 2

Table of Contents

The "soft" approach [1]

In the Bayesian/MAP approach we evaluate the posterior probabilities

$$\begin{aligned} P(\mathcal{H} = \mathcal{H}_k | \mathbf{y} = \mathbf{u}) &= \frac{P(\mathcal{H} = \mathcal{H}_k, \mathbf{y} = \mathbf{u})}{P(\mathbf{y} = \mathbf{u})} \\ &= \frac{P(\mathbf{y} = \mathbf{u} | \mathcal{H} = \mathcal{H}_k) P(\mathcal{H} = \mathcal{H}_k)}{P(\mathbf{y} = \mathbf{u})} \\ &= \frac{f_{\mathbf{y}|\mathcal{H}_k}(\mathbf{u}) \pi_k}{f_{\mathbf{y}}(\mathbf{u})} \end{aligned}$$

$$\text{Since } f_{\mathbf{y}}(\mathbf{u}) = \sum_{s=1}^{K} \pi_s f_{\mathbf{y}|\mathcal{H}_s}(\mathbf{u}),$$

$$P(\mathcal{H} = \mathcal{H}_k | \mathbf{y} = \mathbf{u}) = \frac{\pi_k f_{\mathbf{y}|\mathcal{H}_k}(\mathbf{u})}{\sum_{s=1}^{K} \pi_s f_{\mathbf{y}|\mathcal{H}_s}(\mathbf{u})} = r_k(\mathbf{u})$$

The "soft" approach [2]

According to the MAP criterion, we should therefore say that the k-th decision region is made of all the points $\mathbf{u}$ for which $P(\mathcal{H} = \mathcal{H}_k | \mathbf{y} = \mathbf{u})$ is maximum:

$$\mathcal{R}_k = \{\mathbf{u} : P(\mathcal{H} = \mathcal{H}_k | \mathbf{y} = \mathbf{u}) \geq P(\mathcal{H} = \mathcal{H}_j | \mathbf{y} = \mathbf{u}), \forall j \neq k\}$$

However, we don't want to make a "hard" decision ($\mathbf{u}$ clearly belongs to one and only one region), we prefer to make a **"soft" decision** in which $P(\mathcal{H} = \mathcal{H}_k | \mathbf{y} = \mathbf{u})$ is just the probability that $\mathbf{u}$ belongs to $\mathcal{R}_k$. The reason is that points $\mathbf{y}(n)$ that are halfway between $\mathbf{x}_k$ and $\mathbf{x}_j$ should not be simply given to $\mathcal{R}_k$ if $\|\mathbf{y}(n) - \mathbf{x}_k\|^2$ is only slightly smaller than $\|\mathbf{y}(n) - \mathbf{x}_j\|^2$; it's better that such points give their contributions to both regions (at least during the phase in which we define the regions).

The "soft" approach [3]

In the literature,

Responsibility

$$P(\mathcal{H} = \mathcal{H}_k | \mathbf{y} = \mathbf{u}) = \frac{\pi_k f_{\mathbf{y}|\mathcal{H}_k}(\mathbf{u})}{\sum_{s=1}^{K} \pi_s f_{\mathbf{y}|\mathcal{H}_s}(\mathbf{u})} = r_k(\mathbf{u})$$

is called the **responsibility** of cluster k for point $\mathbf{u}$.

(cluster k is responsible only for some points $\mathbf{u}$, those that are closer to the cluster and with high values of $r_k(\mathbf{u})$, whereas it is not responsible for the far away points $\mathbf{u}$ with low values of $r_k(\mathbf{u})$, which will be responsibility of another cluster).
In the algorithm, we start from a set of values $\pi_k(0), \mathbf{x}_k(0), \sigma_k^2(0)$ and we evaluate $r_k(\mathbf{u}) = P(\mathcal{H} = \mathcal{H}_k | \mathbf{y} = \mathbf{y}(n))$ for each of the N measured points $\mathbf{y}(n)$ using the parameters $\pi_k(0), \mathbf{x}_k(0), \sigma_k^2(0)$, then we update the parameters.

Update of $\mathbf{x}_k$ [1]

Let us start with the update from $\mathbf{x}_k(0)$ to $\mathbf{x}_k(1)$.
We know how to evaluate the probability density function of $\mathbf{y}$ at $\mathbf{y}(n)$ (i.e. $\mathbf{y}$ is the random vector, $\mathbf{y}(n)$ is the value taken by $\mathbf{y}$ during the n-th experiment):

$$f_{\mathbf{y}}(\mathbf{y}(n)) = \sum_{s=1}^{K} \pi_s f_{\mathbf{y}|\mathcal{H}_s}(\mathbf{y}(n))$$

but this pdf depends on the unknown parameters $\mathbf{x}_k$, $k = 1, \ldots, K$:

$$f_{\mathbf{y}}(\mathbf{y}(n)) = \sum_{s=1}^{K} \pi_s \frac{1}{(2\pi\sigma_s^2)^{M/2}} e^{-\frac{1}{2\sigma_s^2}\|\mathbf{y}(n)-\mathbf{x}_s\|^2} = g(\mathbf{y}(n), \mathbf{x}_1, \ldots, \mathbf{x}_K)$$

If we look at $g(\cdot)$ as a function of $\mathbf{x}_1, \ldots, \mathbf{x}_K$, being $\mathbf{y}(n)$ fixed, we see that it has a maximum when $\mathbf{x}_1 = \cdots = \mathbf{x}_K = \mathbf{y}(n)$.

Update of $\mathbf{x}_k$ [2]

But, if we consider all the N measurements $\mathbf{y}(1), \ldots \mathbf{y}(N)$ (assuming that they are statistically independent) we have a joint pdf

$$f_{\mathbf{y}}(\mathbf{y}(1), \ldots, \mathbf{y}(N)) = \prod_{n=1}^{N} \sum_{s=1}^{K} \pi_s \frac{1}{(2\pi\sigma_s^2)^{M/2}} e^{-\frac{1}{2\sigma_s^2}\|\mathbf{y}(n)-\mathbf{x}_s\|^2}$$
$$= \prod_{n=1}^{N} g(\mathbf{y}(n), \mathbf{x}_1, \ldots, \mathbf{x}_K)$$

which is again a function of , $\mathbf{x}_1, \ldots, \mathbf{x}_K$ and shows now several maxima. Then we can estimate $\mathbf{x}_k, k = 1, \ldots, K$ by maximizing $\log(f_{\mathbf{y}}(\mathbf{y}(1), \ldots, \mathbf{y}(N)))$:

$$G(\mathbf{x}_1, \ldots, \mathbf{x}_K) = \log(f_{\mathbf{y}}(\mathbf{y}(1), \ldots, \mathbf{y}(N))) = \sum_{n=1}^{N} \log(g(\mathbf{y}(n), \mathbf{x}_1, \ldots, \mathbf{x}_K))$$

Update of $\mathbf{x}_k$ [3]

(Note that the maximum likelihood criterion would require that you evaluate $f(\mathbf{y}(n)|\mathbf{x}_k)$ for all k-s and you choose $\mathbf{x}_j$ if it corresponds to the maximum, but you need to know all $\mathbf{x}_k$'s, and therefore it cannot be applied here).

$$\nabla_{\mathbf{x}_k} G(\mathbf{x}_1,\ldots,\mathbf{x}_K) = \sum_{n=1}^{N} \frac{1}{g(\mathbf{y}(n),\mathbf{x}_1,\ldots,\mathbf{x}_K)} \nabla_{\mathbf{x}_k} g(\mathbf{x}_1,\ldots,\mathbf{x}_K)$$

$$= \sum_{n=1}^{N} \frac{1}{g(\mathbf{y}(n),\mathbf{x}_1,\ldots,\mathbf{x}_K)} \frac{\pi_k}{(2\pi\sigma_k^2)^{M/2}} e^{-\frac{1}{2\sigma_k^2}\|\mathbf{y}(n)-\mathbf{x}_k\|^2} \cdot \frac{-2(\mathbf{x}_k-\mathbf{y}(n))}{2\sigma_k^2}$$

$$= \sum_{n=1}^{N} \frac{1}{f_{\mathbf{y}}(\mathbf{y}(n))} \pi_k f_{\mathbf{y}|\mathcal{H}_k}(\mathbf{y}(\mathbf{n})) \frac{\mathbf{y}(n)-\mathbf{x}_k}{\sigma_k^2}$$

$$= \sum_{n=1}^{N} P(\mathcal{H}_k|\mathbf{y}(n)) \frac{\mathbf{y}(n)-\mathbf{x}_k}{\sigma_k^2}$$

$$= \sum_{n=1}^{N} r_k(\mathbf{y}(n)) \frac{\mathbf{y}(n)-\mathbf{x}_k}{\sigma_k^2}$$

Update of $\mathbf{x}_k$ [4]

Then the optimum value of $\mathbf{x}_k$ is the one that solves

$$\nabla_{\mathbf{x}_k} G(\mathbf{x}_1, \ldots, \mathbf{x}_K) = \sum_{n=1}^{N} r_k(\mathbf{y}(n)) \frac{\mathbf{y}(n) - \mathbf{x}_k}{\sigma_k^2} = \mathbf{0}$$

$$\sum_{n=1}^{N} r_k(\mathbf{y}(n)) \mathbf{x}_k = \sum_{n=1}^{N} r_k(\mathbf{y}(n)) \mathbf{y}(n)$$

$$\mathbf{x}_k = \frac{\sum_{n=1}^{N} r_k(\mathbf{y}(n)) \mathbf{y}(n)}{\sum_{n=1}^{N} r_k(\mathbf{y}(n))}$$

Written in this way, it seems that we have a closed form expression for $\mathbf{x}_k$, but it is not so, since $r_k(\mathbf{y}(n))$ depends on $\mathbf{x}_j$ with $j = 1, \ldots, N$. However, we can use the above solution to update the value of $\mathbf{x}_k$:

Update of $\mathbf{x}_k$ [5]

Update of $\mathbf{x}_k$

$$\mathbf{x}_k(1) = \frac{\sum_{n=1}^{N} r_k(\mathbf{y}(n))\mathbf{y}(n)}{\sum_{n=1}^{N} r_k(\mathbf{y}(n))}, \quad k = 1, \ldots, K$$

where the terms $r_k(\mathbf{y}(n))$ on the right side of the equality depend on $\mathbf{x}_j(0), \pi_j(0), \sigma_j^2(0), j = 1, \ldots, K$:

$$r_k(\mathbf{y}(n)) = \frac{\pi_k(0) f_{\mathbf{y}|\mathcal{H}_k}(\mathbf{y}(n))}{\sum_{s=1}^{K} \pi_s(0) f_{\mathbf{y}|\mathcal{H}_s}(\mathbf{y}(n))}$$

$$f_{\mathbf{y}|\mathcal{H}_s}(\mathbf{y}(n)) = \frac{1}{(2\pi\sigma_s^2(0))^{M/2}} e^{-\frac{1}{2\sigma_s^2(0)}\|\mathbf{y}(n)-\mathbf{x}_s(0)\|^2}$$

- If we set the responsibility $r_k(\mathbf{y}(n)) = 1$ if $\mathbf{y}(n)$ is assumed to belong to $\mathcal{R}_k$, and $r_k(\mathbf{y}(n)) = 0$ otherwise, then we get the update rule of the "hard" K-means algorithm

Update of $\mathbf{x}_k$ [6]

- We used a sort of ML criterion to find $\mathbf{x}_k(1)$. The Bayesian approach would require the evaluation of the posterior probabilities $P(\mathbf{x}_k = \mathbf{z}|\mathbf{y}(1), \ldots, \mathbf{y}(N))$, which however requires the prior probability density function $f_{\mathbf{x}_k}(\mathbf{z})$, which we don't know (should it be Gaussian? why?)

Update of σ_k^2

With a similar formula, we update the variance value as follows

Update of σ_k^2

$$\sigma_k^2(1) = \frac{\sum_{n=1}^{N} r_k(\mathbf{y}(n)) \|\mathbf{y}(n) - \mathbf{x}_k(1)\|^2}{M \sum_{n=1}^{N} r_k(\mathbf{y}(n))}, \quad k = 1, \ldots, K$$

Note that we must divide by the dimension M, since, assuming that the hypothesis $\mathcal{H}_k$ is correct,

$$\begin{aligned}
\mathbb{E}\{\|\mathbf{y} - \mathbf{x}_k\|^2 | \mathcal{H} = \mathcal{H}_k\} &= \mathbb{E}\{\|(\mathbf{x}_k + \boldsymbol{\nu}) - \mathbf{x}_k\|^2 | \mathcal{H} = \mathcal{H}_k\} \\
&= \mathbb{E}\{\|\boldsymbol{\nu}\|^2 | \mathcal{H} = \mathcal{H}_k\} \\
&= \mathbb{E}\{\nu_1^2 + \nu_2^2 + \ldots + \nu_M^2 | \mathcal{H} = \mathcal{H}_k\} = M\sigma_k^2
\end{aligned}$$

Update of π_k

If we think of $r_k(\mathbf{y}(n))$ as a soft version of the corresponding hard decision $r_k(\mathbf{y}(n)) = 1$ if $\mathbf{y}(n) \in \mathcal{R}_k$, then the number of measurements $\mathbf{y}(n)$ inside cluster k is $N_k = \sum_{n=1}^{N} r_k(\mathbf{y}(n))$ and the total number of points N can also be written as the sum of the points associated to each cluster $N = \sum_{k=1}^{K} N_k$, so that $\pi_k(1) = N_k/N$ can also be written as

Update of π_k

$$\pi_k(1) = \frac{\sum_{n=1}^{N} r_k(\mathbf{y}(n))}{\sum_{k=1}^{K}\sum_{n=1}^{N} r_k(\mathbf{y}(n))}, \quad k = 1, \ldots, K$$

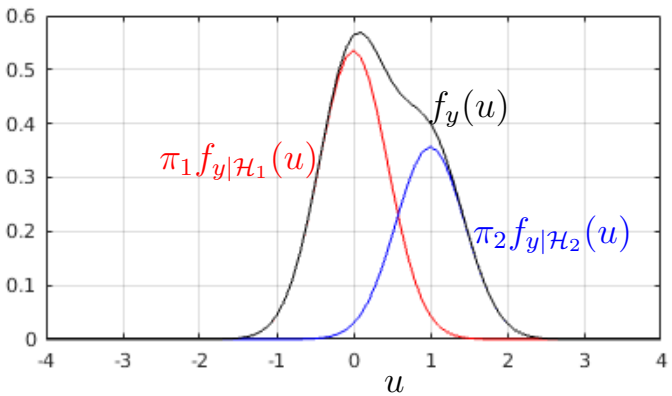
which is valid for both the "soft" and the "hard" versions of the K-means algorithm

Example: the pdfs and the drawn samples

Model ($M = 1$):

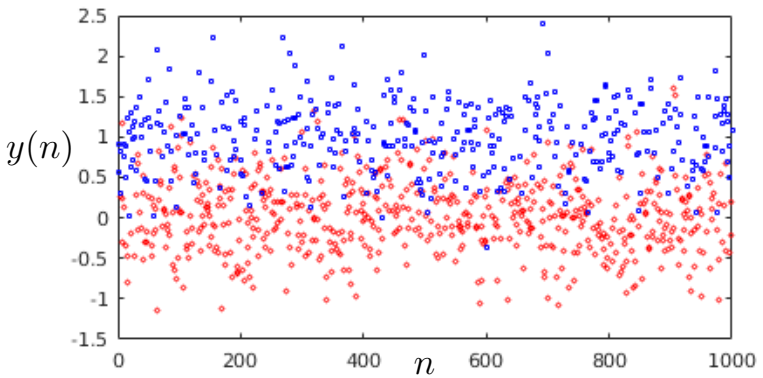
- $\mathcal{H}_1 : y = x_1 + \nu_1, x_1 = 0, \nu \in \mathcal{N}(0, \sigma_1^2 = 0.2), P(\mathcal{H}_1) = 0.6$
- $\mathcal{H}_2 : y = x_2 + \nu_2, x_1 = 1, \nu \in \mathcal{N}(0, \sigma_2^2 = 0.2), P(\mathcal{H}_2) = 0.4$

Number of samples: $N = 1000$

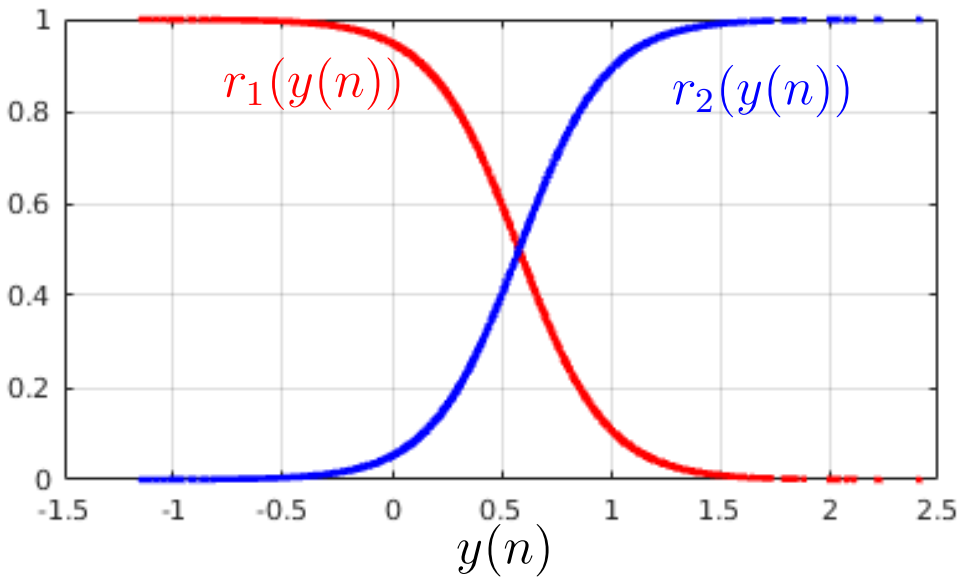


$\pi_1 = 0.6$
$x_1 = 0$
$\sigma_1^2 = 0.2$

$\pi_2 = 0.4$
$x_2 = 1$
$\sigma_2^2 = 0.2$



Example: the responsibilities at the drawn samples

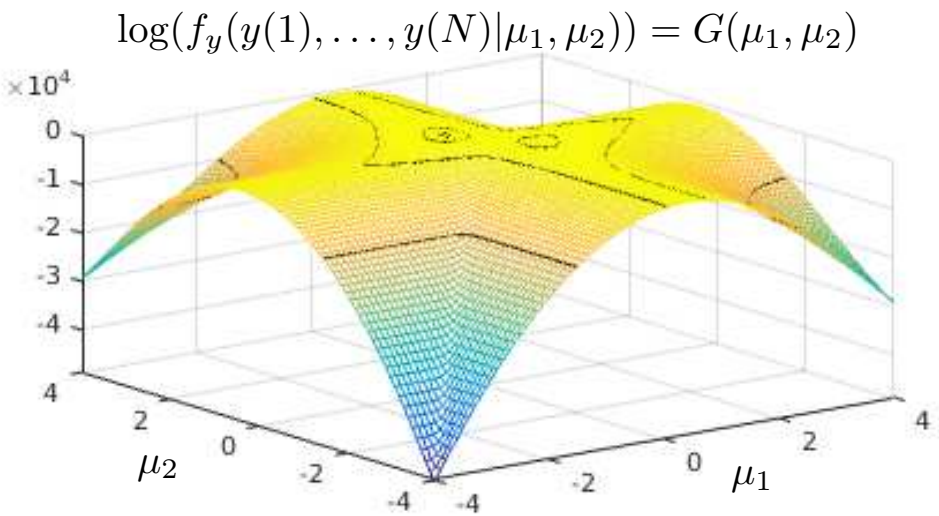


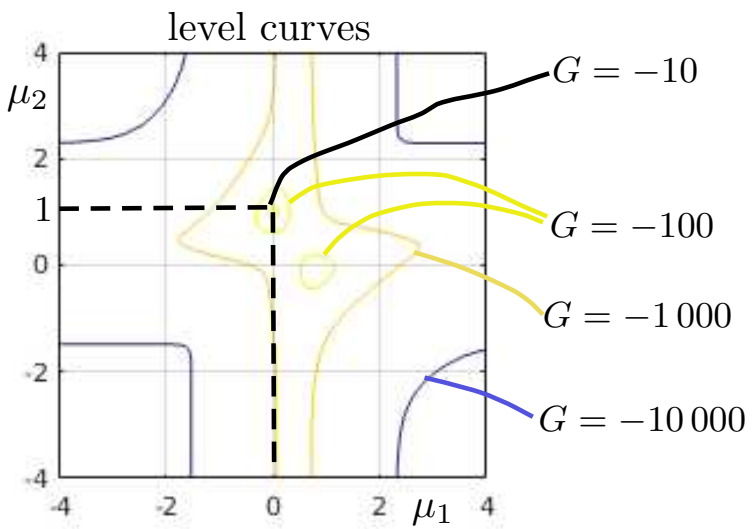
Example: the function used to find x_1 and x_2

On the left: plot of

$$G(\mu_1, \mu_2) = \sum_{n=1}^{N=1000} \log \left[\frac{\pi_1}{(2\pi\sigma_s^2)^{1/2}} e^{-\frac{1}{2\sigma_s^2}(y(n)-\mu_1)^2} + \pi_2 \frac{1}{(2\pi\sigma_s^2)^{1/2}} e^{-\frac{1}{2\sigma_s^2}(y(n)-\mu_2)^2} \right]$$

On the right: the maximum of $G(\mu_1, \mu_2)$ is obtained for $\mu_1 = 0 = x_1$ and $\mu_2 = 1 = x_2$ (we assumed that σ^2, π_1 and π_2 were correct)





The soft K-means algorithm

1. Start from an initial set of values $\mathbf{x}_k(0)$ (random), $\sigma_k^2(0) = 1$, $\pi_k(0) = 1/K$, $k = 1, \ldots, K$. Set $i = 0$
2. **(Assignement step)** for $k = 1, \ldots, K$ and $n = 1, \ldots, N$:

$$r_k(\mathbf{y}(n); i) = \frac{\pi_k(i) \frac{1}{(2\pi\sigma_k^2(i))^{M/2}} e^{-\frac{1}{2\sigma_k^2(i)} \|\mathbf{y}(n) - \mathbf{x}_k(i)\|^2}}{\sum_{s=1}^{K} \pi_s(i) \frac{1}{(2\pi\sigma_s^2(i))^{M/2}} e^{-\frac{1}{2\sigma_s^2(i)} \|\mathbf{y}(n) - \mathbf{x}_s(i)\|^2}}$$

3. **(Update step)**

$$\mathbf{x}_k(i+1) = \frac{\sum_{n=1}^{N} r_k(\mathbf{y}(n); i)\mathbf{y}(n)}{\sum_{n=1}^{N} r_k(\mathbf{y}(n); i)}, \quad k = 1, \ldots, K$$

$$\sigma_k^2(i+1) = \frac{\sum_{n=1}^{N} r_k(\mathbf{y}(n); i)\|\mathbf{y}(n) - \mathbf{x}_k(i+1)\|^2}{M \sum_{n=1}^{N} r_k(\mathbf{y}(n); i)}, \quad k = 1, \ldots, K$$

$$\pi_k(i+1) = \frac{\sum_{n=1}^{N} r_k(\mathbf{y}(n); i)}{\sum_{k=1}^{K} \sum_{n=1}^{N} r_k(\mathbf{y}(n); i)}, \quad k = 1, \ldots, K$$

4. set $i := i + 1$, go back to step 2 until a convergence condition is met.

Comments

- The soft K-means algorithm is at the basis of AutoClass (automatic classification package- Hanson)
- We described the soft K-means algorithm assuming that the Gaussian vector $\boldsymbol{\nu} = [\nu_1, \nu_2, \ldots, \nu_M]$, being correct hypothesis $\mathcal{H}_k$, had $\mathbb{E}\{\nu_1^2\} = \mathbb{E}\{\nu_2^2\} = \cdots = \mathbb{E}\{\nu_M^2\} = \sigma_k^2$. The change to include the assumption $\mathbb{E}\{\nu_1^2\} = \sigma_{1,k}^2$, $\mathbb{E}\{\nu_2^2\} = \sigma_{2,k}^2, \ldots,$ $\mathbb{E}\{\nu_M^2\} = \sigma_{M,k}^2$ with $\sigma_{1,k}^2 \neq \sigma_{2,k}^2$ etc, is straightforward
- The algorithm fails if , at an iteration step i, it turns out that $\mathbf{x}_k(i) = \mathbf{y}(n)$ exactly, because the variance σ_k^2 starts decreasing until it gets to zero and the Gaussian pdf diverges.

Table of Contents

Laboratory # 4 [1]

1. Repeat laboratory #3 using the **hard** K-means algorithm with **2 clusters** (the last column with the arrythmia classification must not be considered; all the patients must be considered); prepare the data by normalizing the features so that each of them has zero mean and variance equal to 1; note that the features are correlated, but this has no effect on the hard K-means algorithm
   - in step 1 start with the vectors $\mathbf{x}_k$ used in the classification problem of laboratory #3, run the hard K-means algorithm and find the difference between the final vectors $\mathbf{x}_k$ and the initial ones; compare now your classification with the classification given by the medical doctors (the case of two classes: arrythmic and healthy);
   - in step 2 start with random initial vectors $\mathbf{x}_k$ as described in the hard K-means algorithm and again compare the final vectors $\mathbf{x}_k$ and the vectors used in the classification problem; compare now your classification with the classification given by the medical doctors (the case of two classes: arrythmic and healthy)

Laboratory # 4 [2]

Note that clustering finds a possible set of clusters and therefore the fact that the classification given by medical doctors is different from the one you find does not mean that your clustering is wrong, it is a clustering that doctors did not point out (they were interested in arrythmia, but patients can be clustered according to the age, for example).

2. then consider **4 clusters**, starting from two different sets of random vectors $\mathbf{x}_k$ and compare the two results that you obtain.
3. Try and implement the **soft** K-means algorithm with 2 clusters. Warning: it is very easy that the algorithm does not converge! at each step check the responsibilities for each patient and force the smallest one to be in any case larger or equal to 10^{-3} (consequently set the other responsibility so that their sum is 1); check the presence of `NAN`s, force the probabilities of the two clusters π_k so that it never occurs that one is zero and the other is one.

Laboratory # 4 [3]

4. Write the report