K-means clustering

BY PHILIPP DÜREN

1 Derivation of the algorithm's essential idea

A random variable $\boldsymbol{X}$ with values in $\mathbb{R}$ exhibiting two clusters is modelled by assuming that it has a probability distribution that is a mixture of two Gaussians. For initial purposes we assume the samples from the individual Gaussians to be labelled accordingly, i.e. each sample is in state $\lambda=1$ or $\lambda=2$, where λ is called the *label of the sample*. Then the generation process of $\boldsymbol{X}$ is as follows:

1. Choose either the first or the second Gaussian by using the (given) distribution of the label, $\mathbb{P}(\lambda=i)$, $i=1,2$. For brevity, we write $\mathbb{P}(\lambda=i)=p_i$, where $p_1+p_2=1$.

2. Generate a sample according to this Gaussian using the common probability distribution function.

Hence the density of the random variable $\boldsymbol{X}$ is:

$$\begin{aligned} f_{\boldsymbol{X}|\boldsymbol{\theta}}(x) &= \sum_{k=1}^{2} p_k \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right) \\ &= \sum_{k=1}^{2} \mathbb{P}(\lambda=k) \cdot f_{\boldsymbol{X}|\boldsymbol{\theta},\lambda=k}(x) \end{aligned}$$

where $\boldsymbol{\theta}=(\mu_1,\mu_2,\sigma)$ is the parameter vector and we denote $\boldsymbol{\mu}=(\mu_1,\mu_2)$.
We would like to get a grip on the probability distribution of the parameters μ_1 and μ_2 (we regard σ as fixed). For that matter, we assume a *prior distribution* $f_{\boldsymbol{\mu}}$ and accept the following black box Bayesian formulas for densities:

- If Y and Z are continuously distributed according to their densities f_Y and f_Z, then

$$f_{Y|Z}=\frac{f_{Z|Y}\cdot f_Y}{f_Z}$$

- If W is discretely distributed with discrete probabilities $\mathbb{P}(W=w_i)=\pi_i$ and Y is continuously distributed according to its density f_Y and also the conditional density $f_{Y|W}$ is known, then

$$\mathbb{P}(W=w_i|Y=y)=\frac{f_{Y|W=w_i}(y)\cdot\pi_i}{f_Y(y)}$$

Then we can write

$$\begin{aligned} \mathbb{P}(\lambda=1|\boldsymbol{\theta},\boldsymbol{X}=x) &= \frac{f_{\boldsymbol{X}|\boldsymbol{\theta},\lambda=1}(x)\cdot p_1}{f_{\boldsymbol{X}|\boldsymbol{\theta}}(x)} \\ &= \frac{\mathbb{P}(\lambda=1)\cdot f_{\boldsymbol{X}|\boldsymbol{\theta},\lambda=1}(x)}{\sum_{k=1}^{2}\mathbb{P}(\lambda=k)\cdot f_{\boldsymbol{X}|\boldsymbol{\theta},\lambda=k}} \\ &= \frac{1}{1+\exp\left(\frac{x(\mu_2-\mu_1)}{\sigma^2}+\frac{\mu_1^2-\mu_2^2}{2\sigma^2}+\log\left(\frac{p_2}{p_1}\right)\right)} \\ \mathbb{P}(\lambda=2|\boldsymbol{\theta},\boldsymbol{X}=x) &= \frac{1}{1+\exp\left(-\frac{x(\mu_2-\mu_1)}{\sigma^2}-\frac{\mu_1^2-\mu_2^2}{2\sigma^2}-\log\left(\frac{p_2}{p_1}\right)\right)} \end{aligned}$$

For brevity we denote

$$p_{k|x} \equiv \mathbb{P}(\lambda=k|\boldsymbol{\theta},\boldsymbol{X}=x)$$

Now we assume that the parameters μ_k are unknown and we wish to infer them from the sample $\{x_n\}_{n=1}^N$. We can derive

$$f_{\boldsymbol{\mu}|\boldsymbol{X}^n=\{x_n\}_{n=1}^N}(\mu_1,\mu_2) \;=\; \frac{f_{\boldsymbol{X}^n|\boldsymbol{\mu}=(\mu_1,\mu_2)}(\{x_n\}_{n=1}^N)\cdot f_{\boldsymbol{\mu}}(\mu_1,\mu_2)}{f_{\boldsymbol{X}^n}(\{x_n\}_{n=1}^N)}$$

and

$$f_{\boldsymbol{X}^n|\boldsymbol{\mu}=(\mu_1,\mu_2)}(\{x_n\}_{n=1}^N) = \prod_{n=1}^N f_{\boldsymbol{X}|\boldsymbol{\mu}}(x_n).$$

We can reason that the most probable guess for $\boldsymbol{\mu}$ is the maximum of the product in the numerator of the fraction above. If we assume a non-committal prior distribution $f_{\boldsymbol{\mu}}(\mu_1,\mu_2)$, we need to maximize the conditional density of $\boldsymbol{X}^n$ given $\boldsymbol{\mu}$, i.e. the *likelihood of* $\boldsymbol{\mu}$. It will be easier to maximize the natural logarithm of this quantity and we denote for brevity

$$L(\boldsymbol{\mu}) \;\equiv\; \log(f_{\boldsymbol{X}^n|\boldsymbol{\mu}=(\mu_1,\mu_2)}(\{x_n\}_{n=1}^N)).$$

To find the maximum of L, we use the Newton method on the gradient of L. For that we have to find the gradient and the Hessian of L first.

$$\begin{aligned}\frac{\partial}{\partial\mu_k}L(\boldsymbol{\mu}) &= \sum_{n=1}^N p_{k|x_n}\cdot\frac{x_n-\mu_k}{\sigma^2}\\ \frac{\partial^2}{\partial\mu_k^2}L(\boldsymbol{\mu}) &= -\sum_{n=1}^N p_{k|x_n}\cdot\frac{1}{\sigma^2}+\sum_{n=1}^N \frac{\partial}{\partial\mu_k}p_{k|x_n}\cdot\frac{x_n-\mu_k}{\sigma^2}\\ &\approx -\sum_{n=1}^N p_{k|x_n}\cdot\frac{1}{\sigma^2}\end{aligned}$$

where we will use the approximation in the last line and the Hessian is thus (approximately) the 2×2 diagonal matrix

$$H\equiv-\sum_{n=1}^N p_{k|x_n}\cdot\frac{1}{\sigma^2}\cdot\mathrm{Id}_2$$

The Newton method thus is

$$\begin{aligned}\boldsymbol{\mu}' &= \boldsymbol{\mu}-H^{-1}\cdot\left(\sum_{n=1}^N p_{k|x_n}\cdot\frac{x_n-\mu_k}{\sigma^2}\right)_{k=1}\\ &= \boldsymbol{\mu}+\frac{\sum_{n=1}^N p_{k|x_n}\cdot x_n}{\sum_{n=1}^N p_{k|x_n}}-\frac{\sum_{n=1}^N p_{k|x_n}\cdot\boldsymbol{\mu}}{\sum_{n=1}^N p_{k|x_n}}\\ &= \frac{\sum_{n=1}^N p_{k|x_n}\cdot x_n}{\sum_{n=1}^N p_{k|x_n}}\end{aligned}$$

Intuitively, this means putting the new cluster centers to the probabilistically weighted center of mass of all data points. The weighing is according to "responsibility" of a cluster for a data point, i.e. data points that are regarded as unrelated to a cluster will not have much influence for its new center.

The algorithm consists of two parts: First, we need to calculate the likelihood that the data set is a result of the current guess of the parameters. This means getting the values of all $p_{k|n}$. We can interpret the $p_{k|n}$ as *responsibilities*: Of course, $p_{1|n}$ and $p_{2|n}y$ add to 1, so if one of them is near 1, we say that this cluster takes *high responsibility* for x_n. This step is also called ***assignment*** as we fuzzily assign clusters (we will in general not have $p_{1|n}=1$ and $p_{2|n}=0$ for most samples, so the responsibility is "fuzzy").

Then we need to update our current guess for $\boldsymbol{\mu}$ by the formula above. This step is called ***update***.

In praxis, we iterate those two steps until our system does not change anymore.

2 Improvements and Generalizations

Now, we model our data set $\{\boldsymbol{x}_n\}_{n=1}^N$, where $\boldsymbol{x}_n \in \mathbb{R}^d$ as a result of a superposition of K multivariate Gaussians $Y_i \sim \mathrm{N}\big(\boldsymbol{\mu}^{(i)}, \Sigma^{(i)}\big)$ with mean $\boldsymbol{\mu}^{(i)} < \mathbb{R}^d$ and covariance matrix $\Sigma^{(i)} < \mathbb{R}^{d \times d}$.

$$\begin{aligned} f_{\boldsymbol{X}\,|\,\boldsymbol{\theta}}(\boldsymbol{x}) &= \sum_{k=1}^{K} p_k \cdot \frac{1}{\sqrt{2\pi \cdot \det\big(\Sigma^{(k)}\big)}} \cdot \exp\Bigg(-\frac{1}{2} \cdot \big(\boldsymbol{x} - \boldsymbol{\mu}^{(k)}\big)^\top \cdot \big[\Sigma^{(k)}\big]^{-1} \cdot \big(\boldsymbol{x} - \boldsymbol{\mu}^{(k)}\big)\Bigg) \\ &= \sum_{k=1}^{K} \mathbb{P}(\lambda = k) \cdot f_{\boldsymbol{X}\,|\,\boldsymbol{\theta},\lambda=k}(\boldsymbol{x}) \end{aligned}$$

Hence the assignment step consists of calculating

$$\begin{aligned} p_{k\,|\,\boldsymbol{x}_n} &= \mathbb{P}(\lambda = k | \boldsymbol{\theta}, \boldsymbol{X} = \boldsymbol{x}_n) \\ &= \frac{p_k \cdot \frac{1}{\sqrt{2\pi \cdot \det(\Sigma^{(k)})}} \cdot \exp\big(-\frac{1}{2} \cdot \big(\boldsymbol{x} - \boldsymbol{\mu}^{(k)}\big)^\top \cdot \big[\Sigma^{(k)}\big]^{-1} \cdot \big(\boldsymbol{x} - \boldsymbol{\mu}^{(k)}\big)\big)}{f_{\boldsymbol{X}\,|\,\boldsymbol{\theta}}(\boldsymbol{x})} \end{aligned}$$

After assigning points, the cluster sizes will change: Perhaps cluster 1 lost a lot of samples to cluster 2. This should find its expression in the weighing of the distributions, i.e. the coefficients p_k. For that we first introduce the notation

$$R^{(k)} = \sum_{n=1}^{N} p_{k\,|\,\boldsymbol{x}_n},$$

i.e. the *total responsibility* of cluster k. Norming those numbers, we get a measure of the proportion of data the cluster k claims for itself:

$$p_k = \frac{R^{(k)}}{\sum_{k=1}^{K} R^{(k)}}$$

Then, using the same arguments as in the simple example, the update step for the cluster centers is

$$\boldsymbol{\mu}^{(k)\prime} = \frac{\sum_{n=1}^{N} p_{k\,|\,\boldsymbol{x}_n} \cdot \boldsymbol{x}_n}{R^{(k)}}$$

It is reasonable to adapt the covariance as well. This can be done using a standard covariance estimator for all data points, again weighed by their responsibilities:

$$\Sigma^{(k)\prime} = \frac{\sum_{n=1}^{N} p_{k\,|\,\boldsymbol{x}_n} \cdot \big[\boldsymbol{x}_n - \boldsymbol{\mu}^{(k)}\big] \cdot \big[\boldsymbol{x}_n - \boldsymbol{\mu}^{(k)}\big]^\top}{\sum_{n=1}^{N} p_{k\,|\,\boldsymbol{x}_n}}$$

3 Conclusion, Caveats and Citations

K-means clustering works well for a reasonable set of problems where data really comes from Gaussian distributions. Of course, a crescent shaped sample will not be modelled appropiately, neither a set of “strings” of data, which a human can easily make out as being clustered intuitively. Also, K-means can blow up: Once a $\boldsymbol{\mu}^{(k)}$ is exactly over one data point $\boldsymbol{x}_n$, the likelihood of that match becomes perfect, yielding a covariance matrix 0. This is a typical flaw of maximum likelihood methods: Overfitting of data is not excluded and can lead to pathological results.
This short overview was shamelessly C&P-ed from [Mac03].

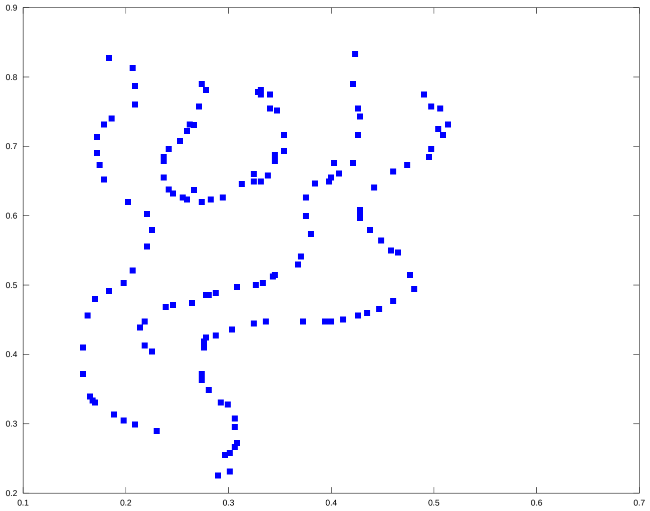

Figure 1. An example where K-means clustering will not work

Bibliography

[Mac03] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.