

Lecture 05: Clustering I

[SCS4049-02] Machine Learning and Data Science

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

Clustering

Just like in classification, each instance gets assigned to a group. However, unlike classification, clustering is an unsupervised task.

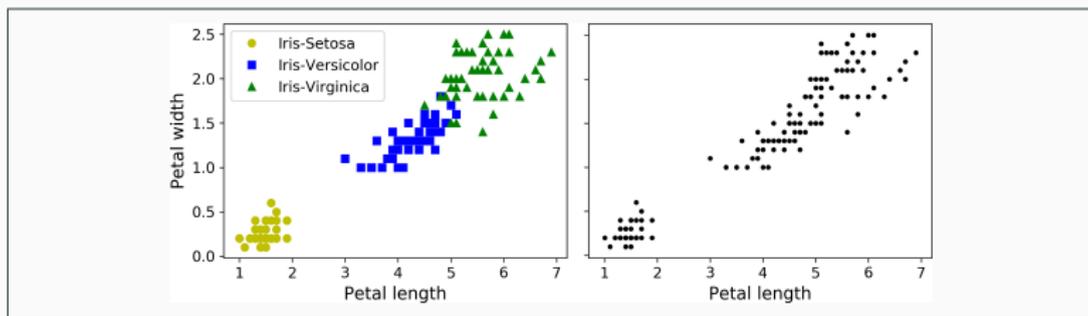


Figure 9-1. Classification (left) versus clustering (right)

Clustering: application

- Customer segmentation
- Data analysis
- Dimensional reduction
- Anomaly detection (outlier detection)
- Semi-supervised learning
- Search engines
- Segmentation of an image

K-means

K-means clustering

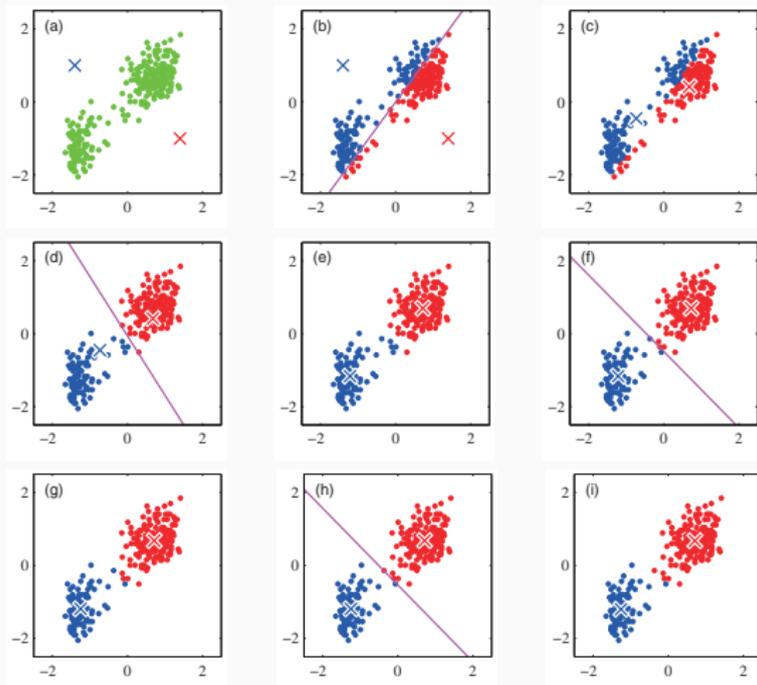


Figure 9.1 Illustration of the K -means algorithm using the re-scaled Old Faithful data set. (a) Green points denote the data set in a two-dimensional Euclidean space. The initial choices for centres μ_1 and μ_2 are shown by the red and blue crosses, respectively. (b) In the initial E step, each data point is assigned either to the red cluster or to the blue cluster, according to which cluster centre is nearer. This is equivalent to classifying the points according to which side of the perpendicular bisector of the two cluster centres, shown by the magenta line, they lie on. (c) In the subsequent M step, each cluster centre is re-computed to be the mean of the points assigned to the corresponding cluster. (d)–(i) show successive E and M steps through to final convergence of

K-means: method

We begin by considering the problem of identifying groups, or clusters, of data points in a multidimensional space. Suppose we have a data set $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ consisting of N observations of a random D -dimensional Euclidean variable \mathbf{x} . Our goal is to partition the data set into some number K of clusters, where we shall suppose for the moment that the value of K is given.

Intuitively, we might think of a cluster as comprising a group of data points whose inter-point distances are small compared with the distances to points outside of the cluster. We can formalize this notion by first introducing a set of D -dimensional vectors $\boldsymbol{\mu}_k$, where $k = 1, \dots, K$, in which $\boldsymbol{\mu}_k$ is a prototype associated with the k -th cluster.

K-means: method

As we shall see shortly, we can think of the μ_k as representing the centres of the clusters. Our goal is then to find an assignment of data points to clusters, as well as a set of vectors $\{\mu_k\}$, such that the sum of the squares of the distances of each data point to its closest vector μ_k , is a minimum.

It is convenient at this point to define some notation to describe the assignment of data points to clusters. For each data point \mathbf{x}_n , we introduce a corresponding set of binary indicator variables $r_{nk} \in \{0, 1\}$, where $k = 1, \dots, K$ describing which of the K clusters the data point \mathbf{x}_n is assigned to, so that if data point \mathbf{x}_n is assigned to cluster k then $r_{nk} = 1$, and $r_{nj} = 0$ for $j \neq k$. This is known as **the 1-of- K coding scheme**.

K-means: method

We can then define an objective function, sometimes called a *distortion measure*, given by

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \quad (1)$$

which represents the sum of the squares of the distances of each data point to its assigned vector $\boldsymbol{\mu}_k$.

Our goal is to find values for the $\{r_{nk}\}$ and the $\{\boldsymbol{\mu}_k\}$ so as to minimize J .

The terms involving different n are independent and so we can optimize for each n separately by choosing r_{nk} to be 1 for whichever value of k gives the minimum value of $\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$. In other words, we simply assign the n -th data point to the closest cluster centre.

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

K-means: method

Now consider the optimization of the $\boldsymbol{\mu}_k$ with the r_{nk} held fixed. The objective function J is a quadratic function of $\boldsymbol{\mu}_k$, and it can be minimized by setting its derivative with respect to $\boldsymbol{\mu}_k$ to zero giving

$$2 \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \quad (3)$$

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{r_{nk}} \quad (4)$$

The denominator in this expression is equal to the number of points assigned to cluster k , and so this result has a simple interpretation, namely set $\boldsymbol{\mu}_k$ equal to the mean of all of the data points \mathbf{x}_n assigned to cluster k .

The two phases of re-assigning data points to clusters and re-computing the cluster means are repeated in turn until there is no further change in the assignments (or until some maximum number of iterations is exceeded).

K-means: python example

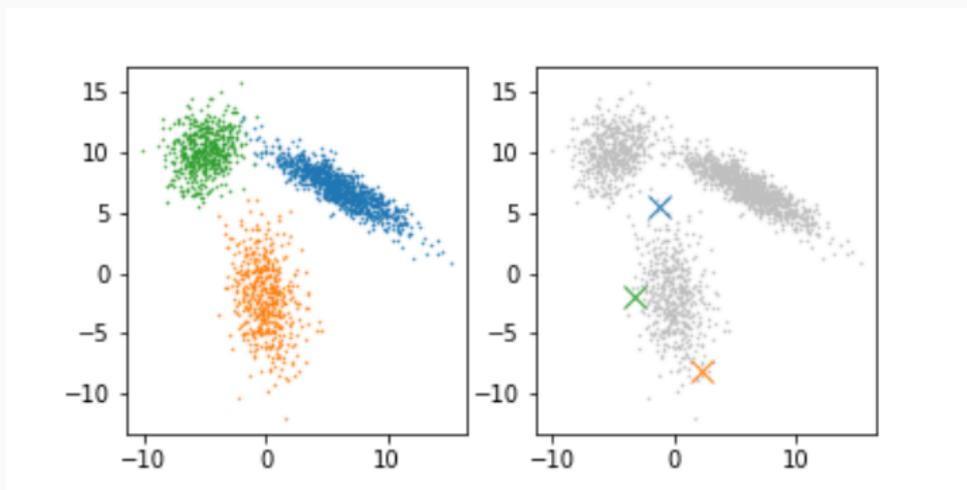


Figure 1: K-means algorithm python example: dataset (left) and the initialization (right).

K-means: python example

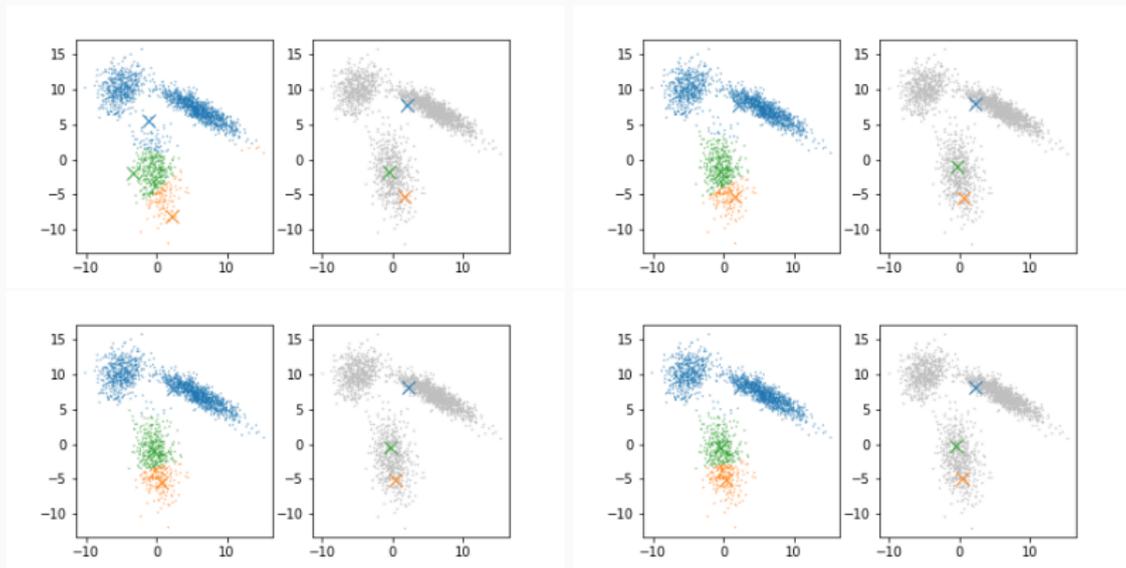


Figure 2: From the 1st to 4th iterations.

K-means: python example

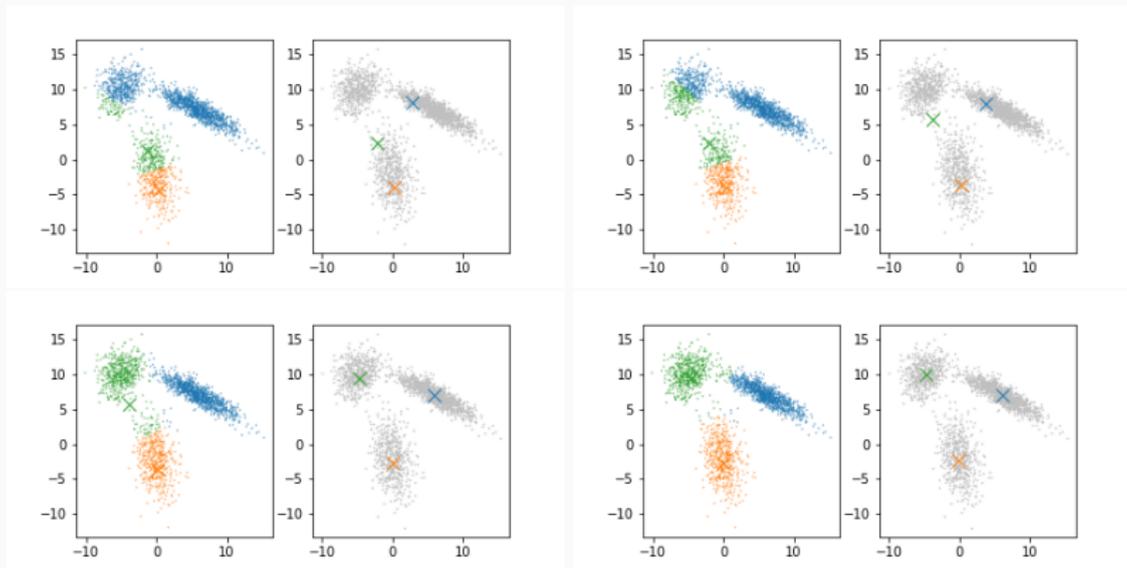


Figure 3: From the 9th to 12th iterations.

K-means: example

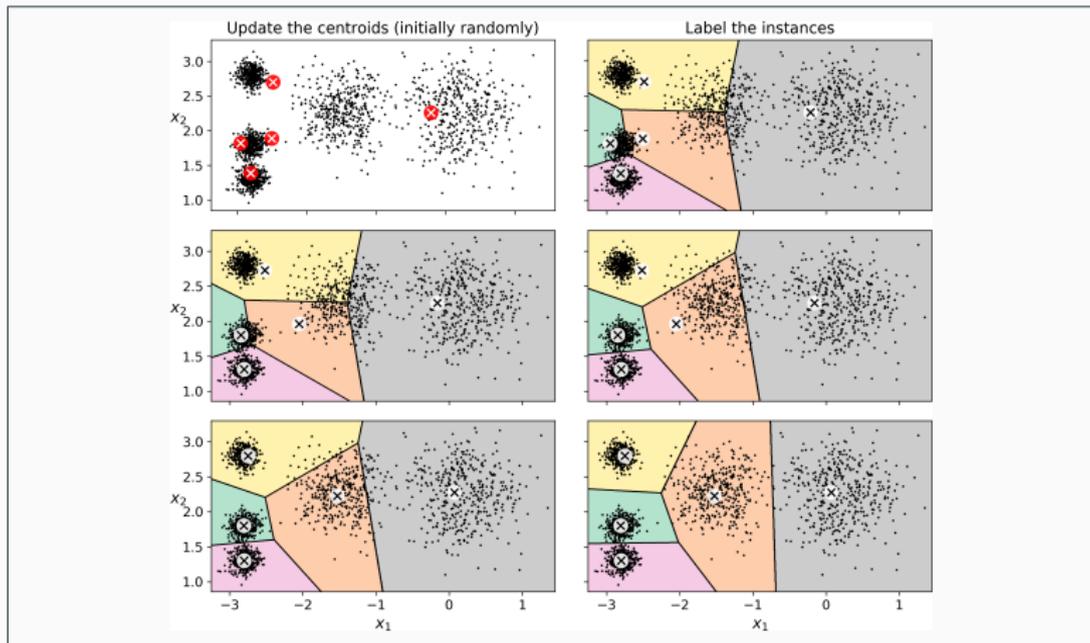


Figure 9-4. The K-Means algorithm

K-means: decision boundary

Decision boundary = Voronoi tessellation

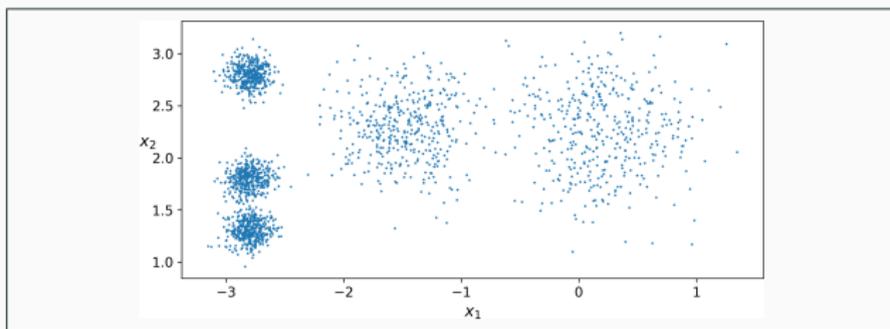


Figure 9-2. An unlabeled dataset composed of five blobs of instances

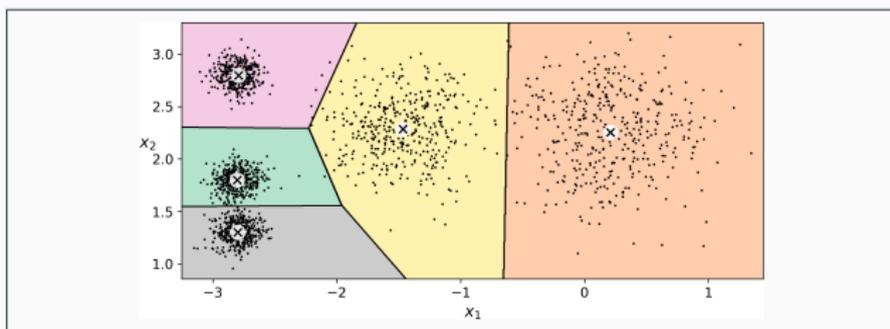


Figure 9-3. K-Means decision boundaries (Voronoi tessellation)

Mixtures of Gaussians

1. **Initialize** the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k .
2. **E-step** Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (5)$$

3. **M-step** Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n \quad (6)$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \quad (7)$$

$$\pi_k = \frac{N_k}{N} \quad (8)$$

4. Evaluate the log likelihood

$$\log p(\mathbf{X} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^N \log \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \quad (9)$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied, return to step 2.

Mixture of Gaussian distributions can be written as a linear superposition of Gaussians.

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (10)$$

Let us introduce K -dimensional binary random variable \mathbf{z} having a 1-of- K representation in which a particular element z_k is equal to 1 and all other elements are equal to 0. The values of z_k therefore satisfy $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$, and we see that there are K possible states for the vector \mathbf{z} according to which element is nonzero.

The marginal distribution over \mathbf{z} is specified in terms of the mixing coefficients π_k , such that

$$p(z_k = 1) = \pi_k \quad (11)$$

where $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^K \pi_k = 1$.

MoG: 1-of-K representation

Figure 9.4 Graphical representation of a mixture model, in which the joint distribution is expressed in the form $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$.



Because \mathbf{z}_k uses a 1-of-K representation, we can also write this distribution in the form

$$p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k} \quad (12)$$

Similarly, the conditional distribution of \mathbf{x} given a particular value for \mathbf{z} is a Gaussian

$$p(\mathbf{x}|\mathbf{z}_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (13)$$

which can also be written in the form

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^K \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k} \quad (14)$$

The joint distribution is given by $p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, and the marginal distribution of \mathbf{x} is then obtained by summing the joint distribution over all possible states of \mathbf{z} to give

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (15)$$

Another quantity that will play an important role is the condition probability of \mathbf{z} given \mathbf{x} . We shall use $\gamma(z_k)$ to denote $p(z_k = 1|\mathbf{x})$, whose value can be found using Bayes' theorem

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(\mathbf{x}|z_k = 1)p(z_k = 1)}{\sum_{j=1}^K p(\mathbf{x}|z_j = 1)p(z_j = 1)} \quad (16)$$

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (17)$$

We shall view π_k as the prior probability of $z_k = 1$, and the quantity $\gamma(z_k)$ as the corresponding posterior probability once we have observed \mathbf{x} . As we shall see later, $\gamma(z_k)$ can also be viewed as the *responsibility* that component k takes for ‘explaining’ the observation \mathbf{x} .

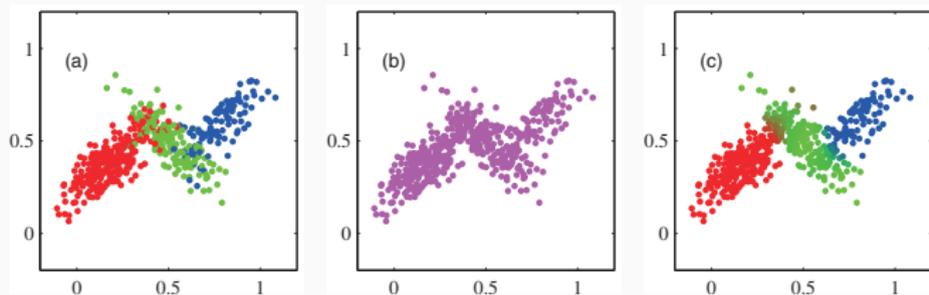


Figure 9.5 Example of 500 points drawn from the mixture of 3 Gaussians shown in Figure 2.23. (a) Samples from the joint distribution $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$ in which the three states of \mathbf{z} , corresponding to the three components of the mixture, are depicted in red, green, and blue, and (b) the corresponding samples from the marginal distribution $p(\mathbf{x})$, which is obtained by simply ignoring the values of \mathbf{z} and just plotting the \mathbf{x} values. The data set in (a) is said to be *complete*, whereas that in (b) is *incomplete*. (c) The same samples in which the colours represent the value of the responsibilities $\gamma(z_{nk})$ associated with data point \mathbf{x}_{n_i} , obtained by plotting the corresponding point using proportions of red, blue, and green ink given by $\gamma(z_{nk})$ for $k = 1, 2, 3$, respectively

MoG: python example

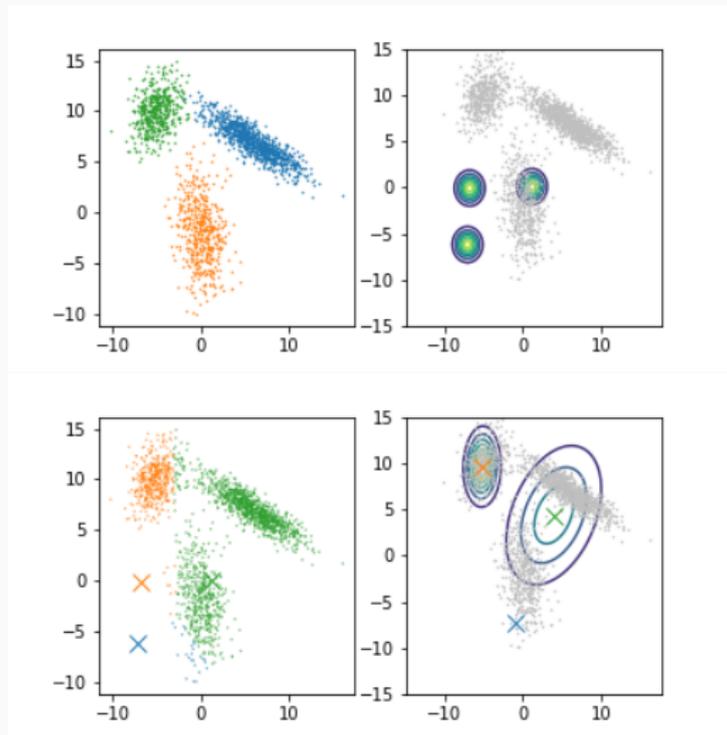


Figure 4: MoG python example: dataset and initialization (top) and the 1st iteration (bottom).

MoG: python example

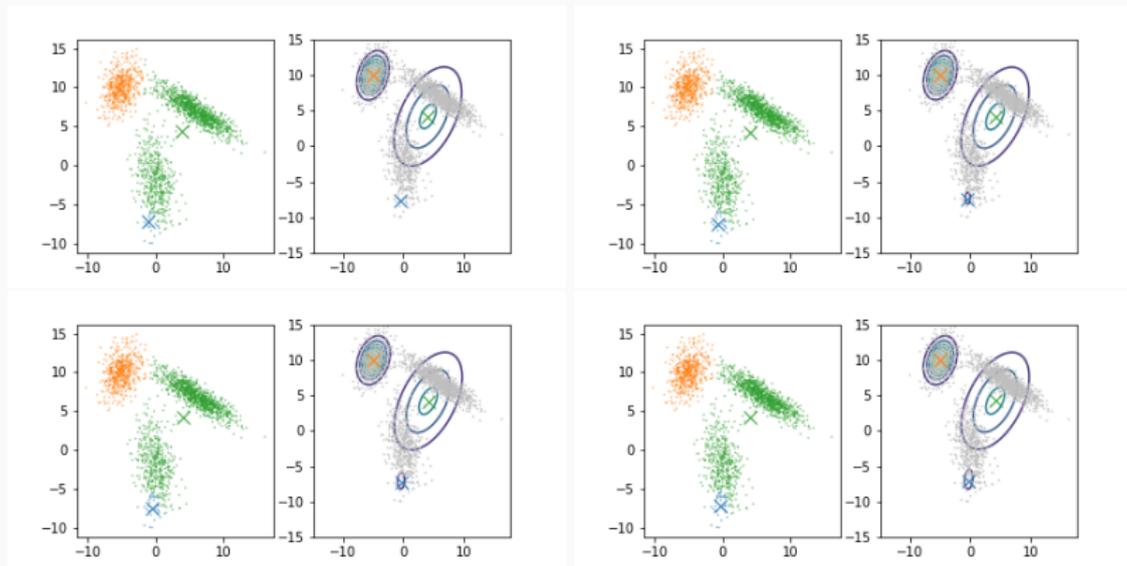


Figure 5: From the 2nd to 5th iterations.

MoG: python example

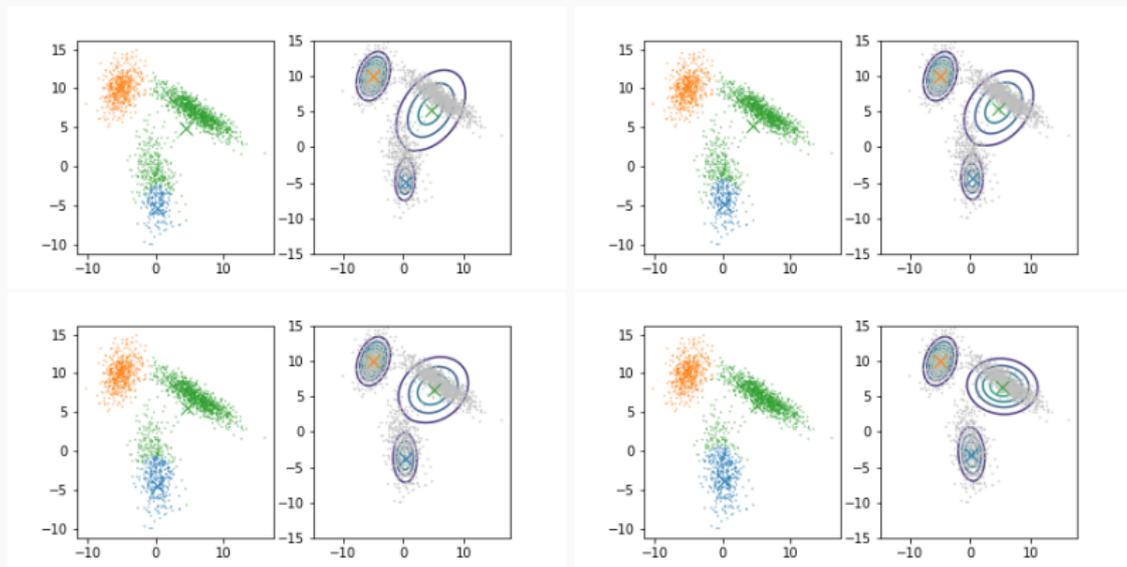


Figure 6: From the 12th to 15th iterations.

MoG: python example

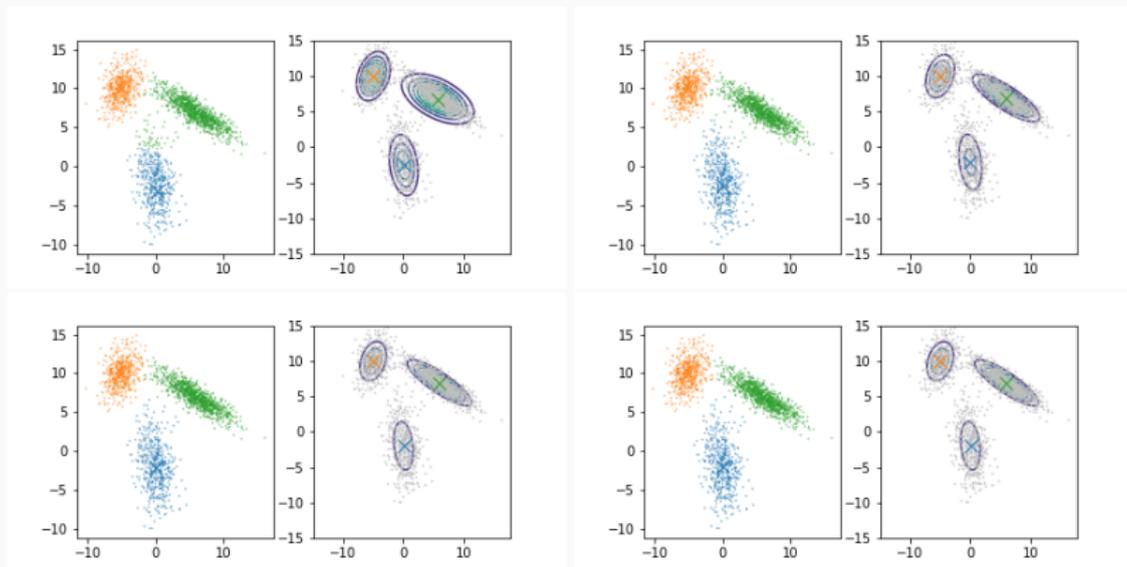


Figure 7: From the 16th to 19th iterations.

Appendix

Reference and further reading

- “Chap 9” of A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
- “Chap 9” of C. Bishop, Pattern Recognition and Machine Learning
- Variational Bayesian mixtures of Gaussians