

Mathematics for Machine Learning

— Expectation Maximization

Joseph Chuang-Chieh Lin

Department of Computer Science & Engineering,
National Taiwan Ocean University

Fall 2025

Credits for the resource

- The slides are based on the textbooks:
 - *Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong: Mathematics for Machine Learning. Cambridge University Press. 2020.*
 - *Arnold J. Insel, Lawrence E. Spence, Stephen H. Friedberg: Linear Algebra, 4th Edition. Prentice Hall. 2013.*
 - *Howard Anton, Chris Rorres, Anton Kaul: Elementary Linear Algebra, 12th Edition. Wiley. 2019.*
- We could partially refer to the monograph:
Francesco Orabona: A Modern Introduction to Online Learning.
<https://arxiv.org/abs/1912.13213>

Outline

- 1 Expectation Maximization (EM) Algorithm
- 2 Latent-Variable Perspective

Outline

- 1 Expectation Maximization (EM) Algorithm
- 2 Latent-Variable Perspective

Motivation

- The previous approach do not give a closed-form solution for the updates of the parameters.
 - \therefore the complex dependency on the parameters.

Motivation

- The previous approach do not give a closed-form solution for the updates of the parameters.
 - \therefore the complex dependency on the parameters.
- The likelihood approach suggests a simple iterative scheme for finding a solution to the parameters estimation problem.

Expectation Maximization

- A general iterative scheme for learning parameters (MLE or MAP) in mixture models and latent-variable models.

Dempster et al. (1977)

Choose initial parameter values (i.e., μ_k, Σ_k, π_k) and **alternate** between the following two steps until convergence:

- **E-step**: Evaluate the responsibilities r_{ik}
 - It can be viewed as the **posterior probability** of data point i belonging to mixture component k .
- **M-step**: Use the **updated responsibilities** to **re-estimate** the parameters.

Expectation Maximization

- A general iterative scheme for learning parameters (MLE or MAP) in mixture models and latent-variable models.

Dempster et al. (1977)

Choose initial parameter values (i.e., μ_k, Σ_k, π_k) and **alternate** between the following two steps until convergence:

- **E-step**: Evaluate the responsibilities r_{ik}
 - It can be viewed as the **posterior probability** of data point i belonging to mixture component k .
 - **M-step**: Use the **updated responsibilities** to **re-estimate** the parameters.
- Intuitive idea: the log-likelihood is increased after each step.

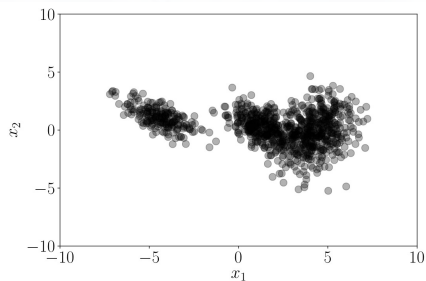
EM algorithm for Estimating parameters of a GMM

- 1 Initialize μ_k, Σ_k, π_k .
- 2 **E-step:** Evaluate r_{ik} for every data point \mathbf{x}_i using the current parameters:

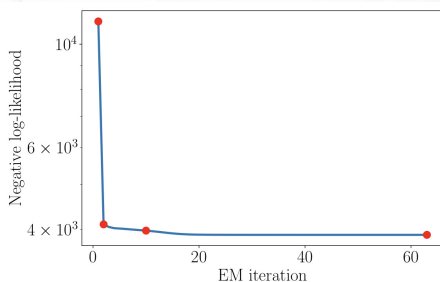
$$r_{ik} = \frac{\pi_k \mathcal{N}(\mathbf{x}_i \mid \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_i \mid \mu_j, \Sigma_j)}$$

- 3 **M-step:** Re-estimate parameters μ_k, Σ_k, π_k using the current responsibilities r_{ik} from the E-step:

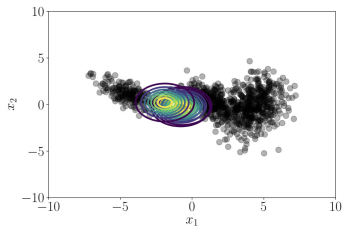
$$\begin{aligned}\mu_k &= \frac{1}{N_k} \sum_{i=1}^N r_{ik} \mathbf{x}_i, \\ \Sigma_k &= \frac{1}{N_k} \sum_{i=1}^N r_{ik} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top, \\ \pi_k &= \frac{N_k}{N}.\end{aligned}$$



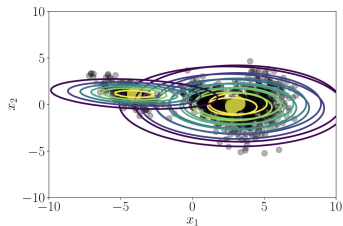
(a) Dataset.



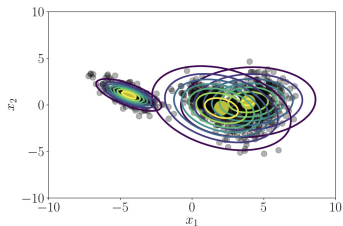
(b) Negative log-likelihood.



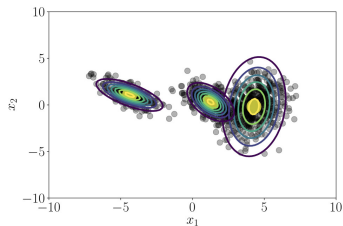
(c) EM initialization.



(d) EM after one iteration.



(e) EM after 10 iterations.



(f) EM after 62 iterations.

Outline

- 1 Expectation Maximization (EM) Algorithm
- 2 Latent-Variable Perspective

Latent-Variable Perspective

- View the GMM from the perspective of a **discrete latent variable** model.
- The latent variable z can attain only a **finite** set of values.

A View of Generative Process

- Consider a GMM as a probabilistic model of generating data.

A View of Generative Process

- Consider a GMM as a probabilistic model of generating data.
- Assume that a mixture model with K components and that a data point \mathbf{x} can be generated by **exactly one** mixture component.

A View of Generative Process

- Consider a GMM as a probabilistic model of generating data.
- Assume that a mixture model with K components and that a data point \mathbf{x} can be generated by **exactly one** mixture component.
- Consider a binary $z_k \in \{0, 1\}$ (whether the k th component is responsible for the data point or not).

A View of Generative Process

- Consider a GMM as a probabilistic model of generating data.
- Assume that a mixture model with K components and that a data point \mathbf{x} can be generated by **exactly one** mixture component.
- Consider a binary $z_k \in \{0, 1\}$ (whether the k th component is responsible for the data point or not).

$$p(\mathbf{x} \mid z_k = 1) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

A View of Generative Process

- Consider a GMM as a probabilistic model of generating data.
- Assume that a mixture model with K components and that a data point \mathbf{x} can be generated by **exactly one** mixture component.
- Consider a binary $z_k \in \{0, 1\}$ (whether the k th component is responsible for the data point or not).

$$p(\mathbf{x} \mid z_k = 1) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

- Define $\mathbf{z} := [z_1, z_2, \dots, z_K]^\top \in \mathbb{R}^K$ as a vector consisting of **exactly one 1 and $K - 1$ many 0s**.
 - **One-hot encoding.**
 - $\mathbf{z} = [z_1, z_2, z_3]^\top = [0, 1, 0]^\top \Rightarrow$ the 2nd mixture component is selected.

Prior on the latent variable

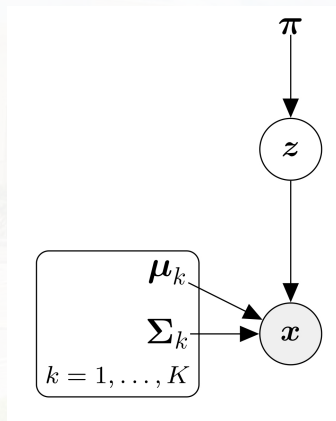
- When the variables z_k are unknown, we can place a prior distribution on \mathbf{z} in practice:

$$p(\mathbf{z}) = \boldsymbol{\pi} = [\pi_1, \pi_2, \dots, \pi_K]^\top, \quad \sum_{k=1}^K \pi_k = 1,$$

where the k th entry $\pi_k = p(z_k = 1)$ describes the prob. that the k th mixture component generated data point \mathbf{x} .

Sampling from a GMM

Ancestral sampling.



A Simple Sampling Procedure

- 1 Sample $z^{(i)} \sim p(\mathbf{z})$.
- 2 Sample $\mathbf{x}^{(i)} \sim p(\mathbf{x} \mid z^{(i)} = 1)$.

Sampling from a GMM

The joint distribution

$$p(\mathbf{x}, z_k = 1) = p(\mathbf{x} \mid z_k = 1)p(z_k = 1) = \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

for $k = 1, \dots, K$. So, we have

$$p(\mathbf{x}, \mathbf{z}) = \begin{bmatrix} p(\mathbf{x}, z_1 = 1) \\ p(\mathbf{x}, z_2 = 1) \\ \vdots \\ p(\mathbf{x}, z_K = 1) \end{bmatrix} = \begin{bmatrix} \pi_1 \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ \pi_2 \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \\ \vdots \\ \pi_K \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K), \end{bmatrix}$$

which fully specifies the probabilistic model.

Likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ in a latent-variable model

Previously, we omitted the parameters $\boldsymbol{\theta}$ of the probabilistic model.

- How to obtain the likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ in a latent-variable model?

Likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ in a latent-variable model

Previously, we omitted the parameters $\boldsymbol{\theta}$ of the probabilistic model.

- How to obtain the likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ in a latent-variable model?
 - Marginalizing out the latent variables.

Likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ in a latent-variable model

Previously, we omitted the parameters $\boldsymbol{\theta}$ of the probabilistic model.

- How to obtain the likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ in a latent-variable model?
 - Marginalizing out the latent variables.
- Summing out (marginalizing out) all latent variables from $p(\mathbf{x}, \mathbf{z})$:

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{\mathbf{z}} p(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{z}) p(\mathbf{z} \mid \boldsymbol{\theta}) \quad ,$$

$$\boldsymbol{\theta} := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k : k = 1, 2, \dots, K\}.$$

Likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ in a latent-variable model

Previously, we omitted the parameters $\boldsymbol{\theta}$ of the probabilistic model.

- How to obtain the likelihood $p(\mathbf{x} \mid \boldsymbol{\theta})$ in a latent-variable model?
 - Marginalizing out the latent variables.
- Summing out (marginalizing out) all latent variables from $p(\mathbf{x}, \mathbf{z})$:

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{\mathbf{z}} p(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{z}) p(\mathbf{z} \mid \boldsymbol{\theta}) = \sum_{k=1}^K p(\mathbf{x} \mid \boldsymbol{\theta}, z_k = 1) p(z_k = 1 \mid \boldsymbol{\theta}),$$

$$\boldsymbol{\theta} := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k : k = 1, 2, \dots, K\}.$$

- There is only one single nonzero entry in each \mathbf{z} , so there are **only K possible configurations** of \mathbf{z} .

So, the desired marginal distribution is

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^K p(\mathbf{x} \mid \boldsymbol{\theta}, z_k = 1) p(z_k = 1 \mid \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

So, the desired marginal distribution is

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^K p(\mathbf{x} \mid \boldsymbol{\theta}, z_k = 1) p(z_k = 1 \mid \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

For the given dataset \mathcal{X} , we have the likelihood

$$p(\mathcal{X} \mid \boldsymbol{\theta}) = \prod_{i=1}^N p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \prod_{i=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

So, the desired marginal distribution is

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^K p(\mathbf{x} \mid \boldsymbol{\theta}, z_k = 1) p(z_k = 1 \mid \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

For the given dataset \mathcal{X} , we have the likelihood

$$p(\mathcal{X} \mid \boldsymbol{\theta}) = \prod_{i=1}^N p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \prod_{i=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

which is exactly the GMM likelihood we have derived before!

So, the desired marginal distribution is

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \sum_{k=1}^K p(\mathbf{x} \mid \boldsymbol{\theta}, z_k = 1) p(z_k = 1 \mid \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

For the given dataset \mathcal{X} , we have the likelihood

$$p(\mathcal{X} \mid \boldsymbol{\theta}) = \prod_{i=1}^N p(\mathbf{x}_i \mid \boldsymbol{\theta}) = \prod_{i=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

which is exactly the GMM likelihood we have derived before!

The latent variable model with latent indicators z_k is an equivalent way of thinking about a Gaussian mixture model.

Posterior Distribution

- Let us look at the posterior distribution on the latent \mathbf{z} .

Posterior Distribution

- Let us look at the posterior distribution on the latent \mathbf{z} .
- By Bayes' theorem,

Posterior Distribution

- Let us look at the posterior distribution on the latent \mathbf{z} .
- By Bayes' theorem,

$$p(z_k = 1 \mid \mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x} \mid z_k = 1)}{p(\mathbf{x})},$$

where the marginal $p(\mathbf{x}) = p(\mathbf{x} \mid \boldsymbol{\theta})$ is we have already derived.

- Hence,

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

Posterior Distribution

- Let us look at the posterior distribution on the latent \mathbf{z} .
- By Bayes' theorem,

$$p(z_k = 1 \mid \mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x} \mid z_k = 1)}{p(\mathbf{x})},$$

where the marginal $p(\mathbf{x}) = p(\mathbf{x} \mid \boldsymbol{\theta})$ is we have already derived.

- Hence,

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

★ The responsibility of the k th mixture component for \mathbf{x} !

Extending to a Full Dataset (1/2)

- Consider a dataset of N data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$.
- Assume that every data point \mathbf{x}_i possesses its own latent variable

$$\mathbf{z}_i = [z_{i1}, z_{i2}, \dots, z_{iK}]^\top \in \mathbb{R}^K.$$

Extending to a Full Dataset (1/2)

- Consider a dataset of N data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$.
- Assume that every data point \mathbf{x}_i possesses its own latent variable

$$\mathbf{z}_i = [z_{i1}, z_{i2}, \dots, z_{iK}]^\top \in \mathbb{R}^K.$$

- Assume that we share the same prior π across all latent variables \mathbf{z}_i .

Extending to a Full Dataset (1/2)

- Consider a dataset of N data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$.
- Assume that every data point \mathbf{x}_i possesses its own latent variable

$$\mathbf{z}_i = [z_{i1}, z_{i2}, \dots, z_{iK}]^\top \in \mathbb{R}^K.$$

- Assume that we share the same prior π across all latent variables \mathbf{z}_i .
- The conditional distribution

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N \mid \mathbf{z}_1, \dots, \mathbf{z}_N) =$$

Extending to a Full Dataset (1/2)

- Consider a dataset of N data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$.
- Assume that every data point \mathbf{x}_i possesses its own latent variable

$$\mathbf{z}_i = [z_{i1}, z_{i2}, \dots, z_{iK}]^\top \in \mathbb{R}^K.$$

- Assume that we share the same prior π across all latent variables \mathbf{z}_i .
- The conditional distribution

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N \mid \mathbf{z}_1, \dots, \mathbf{z}_N) = \prod_{i=1}^N p(\mathbf{x}_i \mid \mathbf{z}_i).$$

Extending to a Full Dataset (2/2)

Consider the posterior distribution $p(z_{ik} = 1 \mid \mathbf{x}_i)$ by applying Bayes' theorem:

$$\begin{aligned} p(z_{ik} = 1 \mid \mathbf{x}_i) &= \frac{p(\mathbf{x}_i \mid z_{ik} = 1)p(z_{ik} = 1)}{\sum_{j=1}^K p(\mathbf{x}_i \mid z_{ij} = 1)p(z_{ij} = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \end{aligned}$$

Extending to a Full Dataset (2/2)

Consider the posterior distribution $p(z_{ik} = 1 \mid \mathbf{x}_i)$ by applying Bayes' theorem:

$$\begin{aligned} p(z_{ik} = 1 \mid \mathbf{x}_i) &= \frac{p(\mathbf{x}_i \mid z_{ik} = 1)p(z_{ik} = 1)}{\sum_{j=1}^K p(\mathbf{x}_i \mid z_{ij} = 1)p(z_{ij} = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \\ &= r_{ik}. \end{aligned}$$

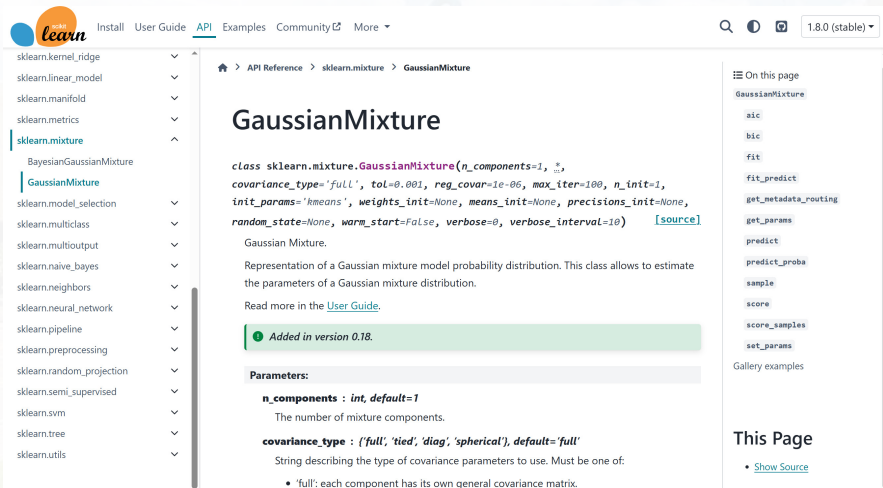
Extending to a Full Dataset (2/2)

Consider the posterior distribution $p(z_{ik} = 1 \mid \mathbf{x}_i)$ by applying Bayes' theorem:

$$\begin{aligned} p(z_{ik} = 1 \mid \mathbf{x}_i) &= \frac{p(\mathbf{x}_i \mid z_{ik} = 1)p(z_{ik} = 1)}{\sum_{j=1}^K p(\mathbf{x}_i \mid z_{ij} = 1)p(z_{ij} = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \\ &= r_{ik}. \end{aligned}$$

- Now, we see that the responsibilities have a mathematically justified interpretation as posterior probabilities.

GaussianMixture — scikit-learn [\[link\]](#)



The screenshot shows the scikit-learn website's API reference for the `GaussianMixture` class. The left sidebar contains a navigation menu with links to various modules, including `sklearn.mixture` which is currently selected. The main content area displays the class name `GaussianMixture` in a large font, followed by its class signature: `class sklearn.mixture.GaussianMixture(n_components=1, *, covariance_type='full', tol=0.001, reg_covar=1e-06, max_iter=100, n_init=1, init_params='kmeans', weights_init=None, means_init=None, precisions_init=None, random_state=None, warm_start=False, verbose=0, verbose_interval=10)`. Below the signature, a brief description states: "Representation of a Gaussian mixture model probability distribution. This class allows to estimate the parameters of a Gaussian mixture distribution." A green callout box indicates "Added in version 0.18." The "Parameters" section lists `n_components` (number of mixture components) and `covariance_type` (string describing the type of covariance parameters to use). The right sidebar provides a list of methods available for the class, such as `aic`, `bic`, `fit`, `fit_predict`, `get_metadata_routing`, `get_params`, `predict`, `predict_proba`, `sample`, `score`, `score_samples`, and `set_params`.

scikit-learn

Install User Guide **API** Examples Community More

1.8.0 (stable)

API Reference > sklearn.mixture > GaussianMixture

GaussianMixture

```
class sklearn.mixture.GaussianMixture(n_components=1, *,
covariance_type='full', tol=0.001, reg_covar=1e-06, max_iter=100, n_init=1,
init_params='kmeans', weights_init=None, means_init=None, precisions_init=None,
random_state=None, warm_start=False, verbose=0, verbose_interval=10)
```

Gaussian Mixture.

Representation of a Gaussian mixture model probability distribution. This class allows to estimate the parameters of a Gaussian mixture distribution.

Read more in the [User Guide](#).

Added in version 0.18.

Parameters:

n_components : *int*, **default=1**
The number of mixture components.

covariance_type : {'full', 'tied', 'diag', 'spherical'}, **default='full'**
String describing the type of covariance parameters to use. Must be one of:

- 'full': each component has its own general covariance matrix.

On this page

- GaussianMixture
- aic
- bic
- fit
- fit_predict
- get_metadata_routing
- get_params
- predict
- predict_proba
- sample
- score
- score_samples
- set_params

Gallery examples

This Page

- [Show Source](#)

Discussions