

[Click to Take the FREE Probability Crash-Course](#)

Search...



# A Gentle Introduction to Expectation-Maximization (EM Algorithm)

by **Jason Brownlee** on [November 1, 2019](#) in **Probability**

[Tweet](#)[Tweet](#)[Share](#)[Share](#)

Last Updated on August 28, 2020

Maximum likelihood estimation is an approach to density estimation for a dataset by searching across probability distributions and their parameters.

It is a general and effective approach that underlies many machine learning algorithms, although it requires that the training dataset is complete, e.g. all relevant interacting random variables are present. Maximum likelihood becomes intractable if there are variables that interact with those in the dataset but were hidden or not observed, so-called latent variables.

The expectation-maximization algorithm is an approach for performing maximum likelihood estimation in the presence of latent variables. It does this by first estimating the values for the latent variables, then optimizing the model, then repeating these two steps until convergence. It is an effective and general approach and is most commonly used for density estimation with missing data, such as clustering algorithms like the Gaussian Mixture Model.

In this post, you will discover the expectation-maximization algorithm.

After reading this post, you will know:

- Maximum likelihood estimation is challenging on data in the presence of latent variables.
- Expectation maximization provides an iterative solution to maximum likelihood estimation with latent variables.
- Gaussian mixture models are an approach to density estimation where the parameters of the distributions are fit using the expectation-maximization algorithm.

[Kick start your project with my new book \*Probability for Machine Learning\* including step by step](#)

- **Update Nov/2019:** Fixed typo in code comment (thanks Daniel)



A Gentle Introduction to Expectation Maximization (EM Algorithm)

Photo by [valcker](#), some rights reserved.

## Overview

This tutorial is divided into four parts; they are:

1. Problem of Latent Variables for Maximum Likelihood
2. Expectation-Maximization Algorithm
3. Gaussian Mixture Model and the EM Algorithm
4. Example of Gaussian Mixture Model



## Problem of Latent Variables for Maximum Likelihood

A common modeling problem involves how to estimate a joint probability distribution for a dataset.

**Density estimation** involves selecting a probability distribution function and the parameters of that distribution that best explain the joint probability distribution of the observed data.

**Maximum Likelihood Estimation** involves treating the problem as an optimization or search problem, where we seek a set of parameters that results in the best fit for the joint probability of the data sample.

A limitation of maximum likelihood estimation is that it assumes that the dataset is complete, or fully observed. This does not mean that the model has access to all data; instead, it assumes that all variables that are relevant to the problem are present.

This is not always the case. There may be datasets where only some of the relevant variables can be observed, and some cannot, and although they influence other random variables in the dataset, they remain hidden.

More generally, these unobserved or hidden variables are referred to as **latent variables**.

“Many real-world problems have hidden variables (sometimes called latent variables), which are not observable in the data that are available for learning.

— Page 816, [Artificial Intelligence: A Modern Approach](#), 3rd edition, 2009.

Conventional maximum likelihood estimation does not work well in the presence of latent variables.

“... if we have missing data and/or latent variables, then computing the [maximum likelihood] estimate becomes hard.

— Page 349, [Machine Learning: A Probabilistic Perspective](#), 2012.

Instead, an alternate formulation of maximum likelihood is required for searching for the appropriate model parameters in the presence of latent variables.

The Expectation-Maximization algorithm is one such approach.

---

## Want to Learn Probability for Machine Learning

Take my free 7-day email crash course now (with sample code).

Click to sign-up and also get a free PDF Ebook version of the course.

Download Your FREE Mini-Course

---

AD

## Expectation-Maximization Algorithm

The Expectation-Maximization Algorithm, or EM algorithm for short, is an approach for maximum likelihood estimation in the presence of latent variables.

“ A general technique for finding maximum likelihood estimators in latent variable models is the expectation-maximization (EM) algorithm.

— Page 424, [Pattern Recognition and Machine Learning](#), 2006.

The EM algorithm is an iterative approach that cycles between two modes. The first mode attempts to estimate the missing or latent variables, called the estimation-step or E-step. The second mode attempts to optimize the parameters of the model to best explain the data, called the maximization-step or M-step.

- **E-Step.** Estimate the missing variables in the dataset.
- **M-Step.** Maximize the parameters of the model in the presence of the data.

The EM algorithm can be applied quite widely, although is perhaps most well known in machine learning for use in unsupervised learning problems, such as density estimation and clustering.

Perhaps the most discussed application of the EM algorithm is for clustering with a mixture model.

## Gaussian Mixture Model and the EM Algorithm

A **mixture model** is a model comprised of an unspecified combination of multiple probability distribution functions.

A statistical procedure or learning algorithm is used to estimate the parameters of the probability distributions to best fit the density of a given training dataset.

The Gaussian Mixture Model, or GMM for short, is a mixture model that uses a combination of Gaussian (Normal) probability distributions and requires the estimation of the mean and standard deviation parameters for each.

There are many techniques for estimating the parameters for a GMM, although a maximum likelihood estimate is perhaps the most common.

Consider the case where a dataset is comprised of many points that happen to be generated by two different processes. The points for each process have a Gaussian probability distribution, but the data is combined and the distributions are similar enough that it is not obvious to which distribution a given point may belong.

The processes used to generate the data point represents a latent variable, e.g. process 0 and process 1. It influences the data but is not observable. As such, the EM algorithm is an appropriate approach to use to estimate the parameters of the distributions.

In the EM algorithm, the estimation-step would estimate a value for the process latent variable for each data point, and the maximization step would optimize the parameters of the probability distributions in an attempt to best capture the density of the data. The process is repeated until a good set of latent values and a maximum likelihood is achieved that fits the data.

- **E-Step.** Estimate the expected value for each latent variable.
-



## Example of Gaussian Mixture Model

We can make the application of the EM algorithm to a Gaussian Mixture Model concrete with a worked example.

First, let's contrive a problem where we have a dataset where points are generated from one of two Gaussian processes. The points are one-dimensional, the mean of the first distribution is 20, the mean of the second distribution is 40, and both distributions have a standard deviation of 5.

We will draw 3,000 points from the first process and 7,000 points from the second process and mix them together.

```
1 ...
2 # generate a sample
3 X1 = normal(loc=20, scale=5, size=3000)
4 X2 = normal(loc=40, scale=5, size=7000)
5 X = hstack((X1, X2))
```

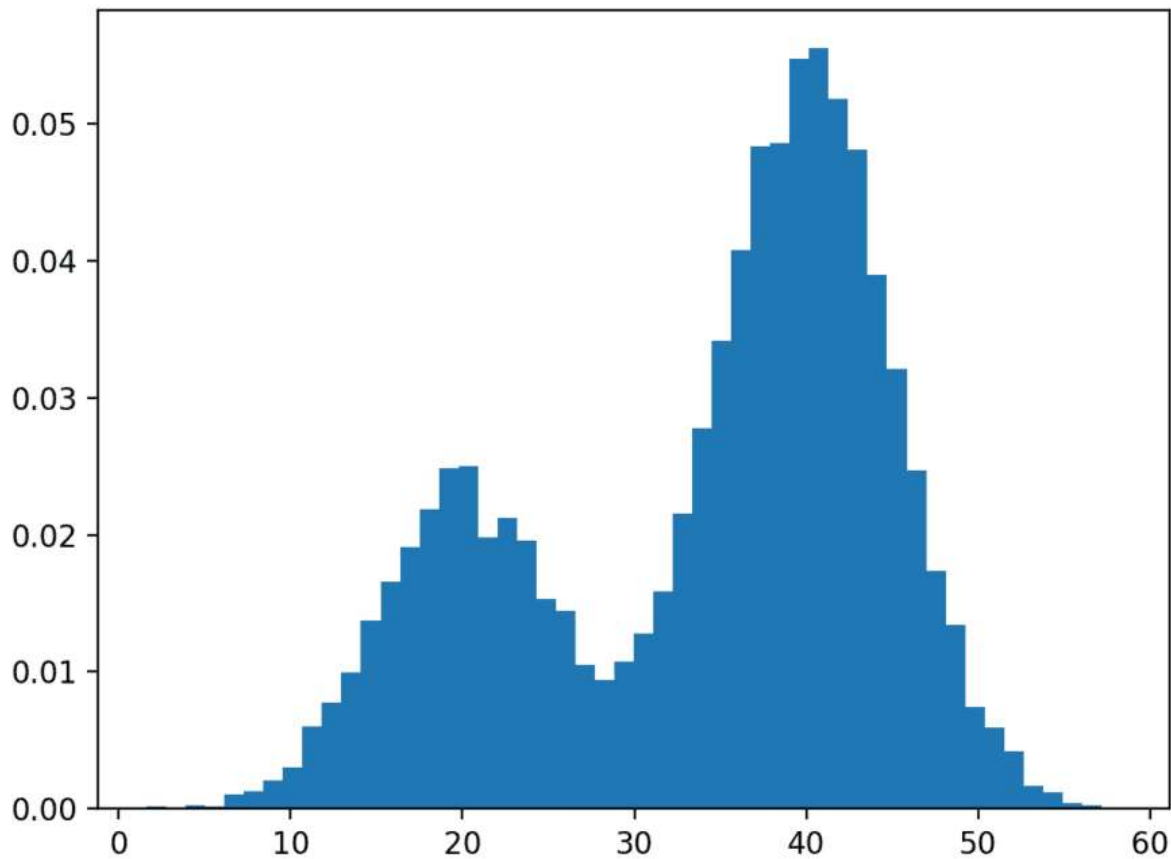
We can then plot a histogram of the points to give an intuition for the dataset. We expect to see a bimodal distribution with a peak for each of the means of the two distributions.

The complete example is listed below.

```
1 # example of a bimodal constructed from two gaussian processes
2 from numpy import hstack
3 from numpy.random import normal
4 from matplotlib import pyplot
5 # generate a sample
6 X1 = normal(loc=20, scale=5, size=3000)
7 X2 = normal(loc=40, scale=5, size=7000)
8 X = hstack((X1, X2))
9 # plot the histogram
10 pyplot.hist(X, bins=50, density=True)
```

The plot clearly shows the expected bimodal distribution with a peak for the first process around 20 and a peak for the second process around 40.

We can see that for many of the points in the middle of the two peaks that it is ambiguous as to which distribution they were drawn from.



Histogram of Dataset Constructed From Two Different Gaussian Processes

We can model the problem of estimating the density of this dataset using a Gaussian Mixture Model.

The [GaussianMixture](#) scikit-learn class can be used to model this problem and estimate the parameters of the distributions using the expectation-maximization algorithm.

The class allows us to specify the suspected number of underlying processes used to generate the data via the `n_components` argument when defining the model. We will set this to 2 for the two processes or distributions.

If the number of processes was not known, a range of different numbers of components could be tested and the model with the best fit could be chosen, where models could be evaluated using scores such as Akaike or Bayesian Information Criterion (AIC or BIC).

```
1 ...
2 # fit model
3 model = GaussianMixture(n_components=2, init_params='random')
4 model.fit(X)
```

Once the model is fit, we can access the learned parameters via arguments on the model, such as the means, covariances, mixing weights, and more.

More usefully, we can use the fit model to estimate the latent parameters for existing and new data points.

For example, we can estimate the latent variable for the points in the training dataset and we would expect the first 3,000 points to belong to one process (e.g. *value=1*) and the next 7,000 data points to belong to a different process (e.g. *value=0*).

```
1 ...
2 # predict latent values
3 yhat = model.predict(X)
4 # check latent value for first few points
5 print(yhat[:100])
6 # check latent value for last few points
7 print(yhat[-100:])
```

Tying all of this together, the complete example is listed below.

```
1 # example of fitting a gaussian mixture model with expectation maximization
2 from numpy import hstack
3 from numpy.random import normal
4 from sklearn.mixture import GaussianMixture
5 # generate a sample
6 X1 = normal(loc=20, scale=5, size=3000)
7 X2 = normal(loc=40, scale=5, size=7000)
8 X = hstack((X1, X2))
9 # reshape into a table with one column
10 X = X.reshape((len(X), 1))
11 # fit model
12 model = GaussianMixture(n_components=2, init_params='random')
13 model.fit(X)
14 # predict latent values
15 yhat = model.predict(X)
16 # check latent value for first few points
17 print(yhat[:100])
18 # check latent value for last few points
19 print(yhat[-100:])
```

Running the example fits the Gaussian mixture model on the prepared dataset using the EM algorithm. Once fit, the model is used to predict the latent variable values for the examples in the training dataset.

**Note:** Your results may vary given the stochastic nature of the algorithm or evaluation procedure, or differences in numerical precision. Consider running the example a few times and compare the average outcome.



[illegible]

AD

## Further Reading

This section provides more resources on the topic if you are looking to go deeper.

## Books

- Section 8.5 The EM Algorithm, [The Elements of Statistical Learning](#), 2016.
- Chapter 9 Mixture Models and EM, [Pattern Recognition and Machine Learning](#), 2006.
- Section 6.12 The EM Algorithm, [Machine Learning](#), 1997.
- Chapter 11 Mixture models and the EM algorithm, [Machine Learning: A Probabilistic Perspective](#), 2012.
- Section 9.3 Clustering And Probability Density Estimation, [Data Mining: Practical Machine Learning Tools and Techniques](#), 4th edition, 2016.
- Section 20.3 Learning With Hidden Variables: The EM Algorithm, [Artificial Intelligence: A Modern Approach](#), 3rd edition, 2009.

AD

## API

- [Gaussian mixture models, scikit-learn API.](#)
- [sklearn.mixture.GaussianMixture API.](#)

## Articles

- [Maximum likelihood estimation, Wikipedia.](#)
- [Expectation-maximization algorithm, Wikipedia.](#)
- [Mixture model, Wikipedia.](#)

## Summary

In this post, you discovered the expectation-maximization algorithm.

Specifically, you learned:

- Maximum likelihood estimation is challenging on data in the presence of latent variables.
- Expectation maximization provides an iterative solution to maximum likelihood estimation with latent variables.
- Gaussian mixture models are an approach to density estimation where the parameters of the distributions are fit using the expectation-maximization algorithm.

Do you have any questions?

Ask your questions in the comments below and I will do my best to answer.

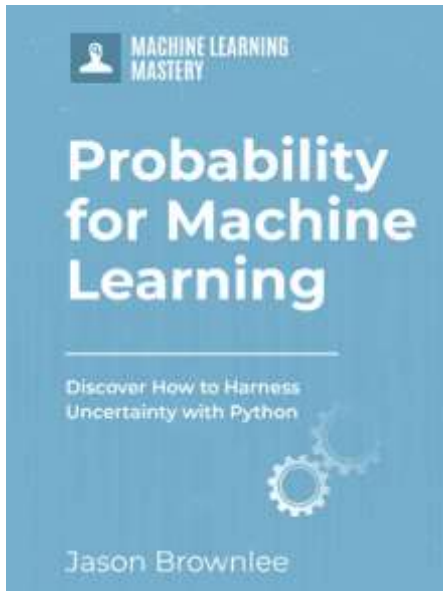
---

## Get a Handle on Probability for Machine Learning!

---

Develop Your Understanding of Probability

---



Discover how in my new Ebook:

[Probability for Machine Learning](#)

It provides **self-study tutorials** and **end-to-end projects** on:  
*Bayes Theorem, Bayesian Optimization, Distributions, Maximum Likelihood,*  
*Cross-Entropy, Calibrating Models*  
and much more...

## Finally Harness Uncertainty in Your Projects

Skip the Academics. Just Results.

[SEE WHAT'S INSIDE](#)

[Tweet](#)

[Tweet](#)

[Share](#)

[Share](#)

## More On This Topic



[A Gentle Introduction to Optimization / Mathematical...](#)



[Why Do I Get Different Results Each Time in Machine...](#)



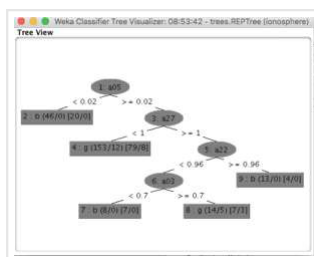
## Stateful and Stateless LSTM for Time Series...



## A Gentle Introduction To Gradient Descent Procedure



## 5 Reasons to Learn Probability for Machine Learning



## How To Use Classification Machine Learning...



## About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

[View all posts by Jason Brownlee](#) →

## < Probabilistic Model Selection with AIC, BIC, and MDL

A Gentle Introduction to Monte Carlo Sampling for Probability &gt;

26 Responses to *A Gentle Introduction to Expectation-Maximization (EM)*



**Ricardo** November 1, 2019 at 10:32 am #

REPLY ↩

this part seems to contradict what follows it: " although it (EM) requires that the training dataset is complete"

---



**Ricardo** November 1, 2019 at 10:34 am #

REPLY ↩

oh, it refers to maximum likelihood, never mind

---



**Jason Brownlee** November 1, 2019 at 1:41 pm #

REPLY ↩

No problem.

---



**Daniel** November 2, 2019 at 10:13 am #

REPLY ↩

Hi Jason, How are you? I hope finding you well!  
I am confused, the first output list should be 0's right? And the second of ones, but they're inverted.  
Thank you in advance!

---



**Jason Brownlee** November 3, 2019 at 5:47 am #

REPLY ↩

The example shows the two different processes were identified, e.g. two class labels.  
It was a typo to expect specific labels to be assigned. I have updated it. Thanks!

---



**Alexander Ligthart** December 4, 2019 at 3:45 am #

REPLY ↩

Dear,

I would like to use a library with the EM algorithm for semi-supervised learning. However, I can't find an existing library in python. Do you know of an exiting one? There are plenty of papers on the subject (Rout et al. 2017, Hassan & Islam 2019) that use EM for semi supervised learning, but their code is not open.

[alexander.ligthart@live.com](mailto:alexander.ligthart@live.com)

Kind regards

---

Typically EM is implemented in service of another task.

e.g. Gaussian mixture models:

<https://scikit-learn.org/stable/modules/mixture.html>



**Amelie** January 23, 2020 at 8:00 pm #

REPLY ↩

Hello,

I have 2 questions please:

- How can we calculate mathematically the center probability of the gaussian  $\mu$  (given that it is the highest probability)
- Is there an automatic exploration method to search for the number of Gaussians for a given stochastic process?

Thank you



**Jason Brownlee** January 24, 2020 at 7:45 am #

REPLY ↩

This is the job of the PDF. Provide a value and get a probability.

Calculate the mean of the distribution and calculate the probability using the pdf.

Not that I'm aware.



**Jonas** May 27, 2022 at 7:42 pm #

REPLY ↩

How about sklearn's `BayesianGaussianMixture` class?



**manjunath** February 29, 2020 at 5:22 am #

REPLY ↩

Can we use this EM algorithm to fill missing values in Time series forecasting ?

Is it effective to fill missing values? if not please suggest some approaches to fill missing values in time series problems



**Jason Brownlee** February 29, 2020 at 7:21 am #

REPLY ↩

See this:

<https://machinelearningmastery.com/handle-missing-timesteps-sequence-prediction-problems->





**ThbML** March 15, 2020 at 9:30 am #

REPLY ↩

Hello. As usual, amazing post ! It helps ! Thank you !

In this example with GMM you use only one feature for clustering.

In my context I use a multi-dimensionnal dataset of about 50 features and 1 000 000 samples (multi parametric image context).

I didn't find any clear answer to if yes or no it is necessary (or better) to scale the features, like in k-means for example with z-score. Can you help me about this ?

If you have a little tip to speed up computation also, It would be great 😊

Thank you.

Cheers.



**Jason Brownlee** March 16, 2020 at 5:51 am #

REPLY ↩

Thanks.

If the input variables have differing units, scaling is a good idea. If they are Gaussian, standardization is a good idea.

Sorry, I don't have suggestions for speeding the computation off hand.



**Nazmus Salehin** March 31, 2020 at 3:15 pm #

REPLY ↩

Hello,

What are some good ways to evaluate the algorithm? One way i could think of is the average number of correct predictions the algorithm is making by summing the correct assignment and divide it by the number of data points. What is your view about it and what are some other ways to evaluate the algorithm?

Thanks



**Jason Brownlee** April 1, 2020 at 5:47 am #

REPLY ↩

Hmmm.

If you are using it for clustering, you could explore clustering specific metrics:  
<https://scikit-learn.org/stable/modules/classes.html#clustering-metrics>

If I understand correctly, the latent parameters in the given example are the Gaussian parameters for each peak?

I guess that we could achieve similar aim as this exercise by fitting appropriate function to histogram data, right?

Regards!



**Jason Brownlee** April 23, 2020 at 5:59 am #

REPLY ↩

More the parameters that define the distribution/s.



**Sene fatou** May 15, 2020 at 11:50 pm #

REPLY ↩

Can I get a python code for expectation maximization in case of estimating parameters in regime switching mean reverting models in financial mathematics



**Jason Brownlee** May 16, 2020 at 6:14 am #

REPLY ↩

Perhaps start with a google search?



**Willwhite** August 3, 2020 at 9:03 pm #

REPLY ↩

Quoting from your text:

E-Step. Estimate the expected value for each latent variable.

M-Step. Optimize the parameters of the distribution using maximum likelihood.

One might misinterpret your post and simply “plug-in” the expected values of the latent variables and then consider them fixed in the M-step.

This is not how the EM works.

The E-step doesn't involve computing the expected value for each latent variable, it involves “computing” the marginal loglikelihood by marginalizing out the latent variables with respect to their conditional distribution given the observed variables and the current value for the estimate.

Also, the usual “estimates” of the latent variables are the maximum a posteriori values, and not their expectation. They coincide if the posterior distribution of the latent variables are symmetric (which is the case in your example), but not in general.

Thanks for your note.



**Schrodinger** August 12, 2020 at 8:21 am #

REPLY ↩

Hi Jason,

I have a question concerning the example you put with gaussian mixture model. If for example I have three combinations of gaussian distribution. Does the model we trained the data estimate the latent variables still with value 0 and 1 or there are 3 possibilities with the value like for example: 0,1, or 2. Thank you so much for your reply.



**Jason Brownlee** August 12, 2020 at 1:34 pm #

REPLY ↩

Yes, I believe you can adapt it for your example.



**Mark Springer** December 19, 2020 at 8:08 am #

REPLY ↩

I don't understand the EM algorithm. Lets take the two gaussians. You have a series of points – do you just pick pairs of gaussians at random, compare their performance, and choose the best? Then you tweak the parameters at random? This can't be what's going on, but you don't explain exactly how the process works.



**Jason Brownlee** December 19, 2020 at 1:28 pm #

REPLY ↩

Not quite, good comment – I need to write a fuller tutorial on the algorithm itself.

As a start, I would recommend some of the references in the “further reading” section.

## Leave a Reply

Email (will not be published) (required)

[SUBMIT COMMENT](#)**Welcome!**I'm *Jason Brownlee* PhDand I **help developers** get results with **machine learning**.[Read more](#)**Never miss a tutorial:**

AD

**Picked for you:**[How to Use ROC Curves and Precision-Recall Curves for Classification in Python](#)[How and When to Use a Calibrated Classification Model with scikit-learn](#)[How to Implement Bayesian Optimization from Scratch in Python](#)



## A Gentle Introduction to Cross-Entropy for Machine Learning

### Loving the Tutorials?

The [Probability for Machine Learning](#) EBook is where you'll find the ***Really Good*** stuff.

>> SEE WHAT'S INSIDE

---

© 2023 Guiding Tech Media. All Rights Reserved.

[LinkedIn](#) | [Twitter](#) | [Facebook](#) | [Newsletter](#) | [RSS](#)

[Privacy](#) | [Disclaimer](#) | [Terms](#) | [Contact](#) | [Sitemap](#) | [Search](#)

[Update Privacy Preferences](#)