

Expectation-maximization algorithm

From Wikipedia, the free encyclopedia

In **statistics**, an **expectation-maximization (EM) algorithm** is an **iterative method** to find **maximum likelihood** or **maximum a posteriori** (MAP) estimates of **parameters** in **statistical models**, where the model depends on unobserved **latent variables**. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the **log-likelihood** evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

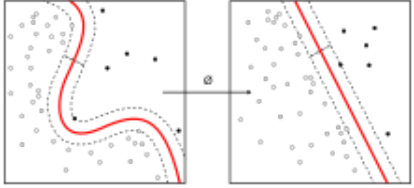
Contents [\[hide\]](#)

- 1 History
- 2 Introduction
- 3 Description
- 4 Properties
- 5 Proof of correctness
- 6 As a maximization-maximization procedure
- 7 Applications
- 8 Filtering and smoothing EM algorithms
- 9 Variants
 - 9.1 α -EM algorithm
- 10 Relation to variational Bayes methods
- 11 Geometric interpretation
- 12 Examples
 - 12.1 Gaussian mixture
 - 12.1.1 E step
 - 12.1.2 M step
 - 12.1.3 Termination
 - 12.1.4 Generalization
 - 12.2 Truncated and censored regression
- 13 Alternatives
- 14 See also
- 15 Further reading
- 16 References
- 17 External links

History [\[edit \]](#)

The EM algorithm was explained and given its name in a classic 1977 paper by [Arthur Dempster](#), [Nan Laird](#), and [Donald Rubin](#).^[1]They pointed out that the method had been "proposed many times in special circumstances" by earlier authors. A very detailed treatment of the EM method for exponential families was published by Rolf Sundberg in his thesis and several papers^{[2][3][4]} following his collaboration with [Per Martin-Löf](#) and [Anders Martin-Löf](#).^{[5][6][7][8][9][10][11]}The Dempster-Laird-Rubin paper in 1977 generalized the method and sketched a convergence analysis for a wider

Machine learning and data mining



Problems

[\[show\]](#)

Supervised learning
(**classification** • **regression**)

[\[show\]](#)

Clustering

[\[show\]](#)

Dimensionality reduction

[\[show\]](#)

Structured prediction

[\[show\]](#)

Anomaly detection

[\[show\]](#)

Neural nets

[\[show\]](#)

Reinforcement Learning


[\[show\]](#)

Theory

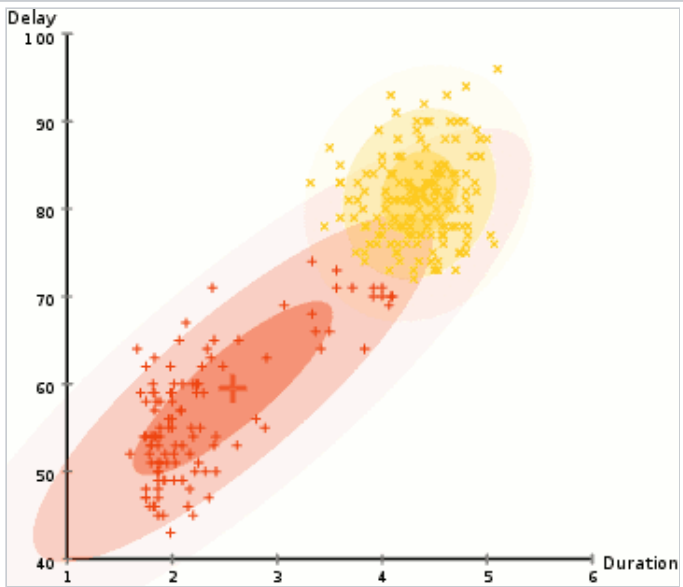
[\[show\]](#)

Machine learning venues

[\[show\]](#)

 **Machine learning portal**

VTE



EM clustering of *Old Faithful* eruption data. The random initial model (which, due to the different scales of the axes, appears to be two very flat and wide spheres) is fit to the observed data. In the first iterations, the model changes substantially, but then converges to the two modes of the *geyser*. Visualized using *ELKI*.

class of problems. Regardless of earlier inventions, the innovative Dempster-Laird-Rubin paper in the *Journal of the Royal Statistical Society* received an enthusiastic discussion at the Royal Statistical Society meeting with Sundberg calling the paper "brilliant". The Dempster-Laird-Rubin paper established the EM method as an important tool of statistical analysis.

The convergence analysis of the Dempster-Laird-Rubin paper was flawed and a correct convergence analysis was published by C.F. Jeff Wu in 1983.^[12] Wu's proof established the EM method's convergence outside of the **exponential family**, as claimed by Dempster-Laird-Rubin.^[12]

Introduction [[edit](#)]

The EM algorithm is used to find (locally) **maximum likelihood** parameters of a **statistical model** in cases where the equations cannot be solved directly. Typically these models involve **latent variables** in addition to unknown **parameters** and known data observations. That is, either **missing values** exist among the data, or the model can be formulated more simply by assuming the existence of further unobserved data points. For example, a **mixture model** can be described more simply by assuming that each observed data point has a corresponding unobserved data point, or latent variable, specifying the mixture component to which each data point belongs.

Finding a maximum likelihood solution typically requires taking the **derivatives** of the **likelihood function** with respect to all the unknown values, the parameters and the latent variables, and simultaneously solving the resulting equations. In statistical models with latent variables, this is usually impossible. Instead, the result is typically a set of interlocking equations in which the solution to the parameters requires the values of the latent variables and vice versa, but substituting one set of equations into the other produces an unsolvable equation.

The EM algorithm proceeds from the observation that the following is a way to solve these two sets of equations numerically. One can simply pick arbitrary values for one of the two sets of unknowns, use them to estimate the second set, then use these new values to find a better estimate of the first set, and then keep alternating between the two until the resulting values both converge to fixed points. It's not obvious that this will work at all, but it can be proven that in this context it does, and that the derivative of the likelihood is (arbitrarily close to) zero at that point, which in turn means that the point is either a maximum or a **saddle point**.^[12] In general, multiple maxima may occur, with no guarantee that the global maximum will be found. Some likelihoods also have **singularities** in them, i.e., nonsensical maxima. For example, one of the *solutions* that may be found by EM in a mixture model involves setting one of the components to have zero variance and the mean parameter for the same component to be equal to one of the data points.

Description [[edit](#)]

Given the **statistical model** which generates a set ***X*** of observed data, a set of unobserved latent data or **missing values** ***Z***, and a vector of unknown parameters ***θ***, along with a **likelihood function** $L(\theta; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z}|\theta)$, the **maximum likelihood estimate** (MLE) of the unknown parameters is determined by the **marginal likelihood** of the observed data

$$L(\theta; \mathbf{X}) = p(\mathbf{X}|\theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\theta)$$

However, this quantity is often intractable (e.g. if ***Z*** is a sequence of events, so that the number of values grows exponentially with the sequence length, making the exact calculation of the sum extremely difficult).

The EM algorithm seeks to find the MLE of the marginal likelihood by iteratively applying these two steps:

Expectation step (E step): Calculate the **expected value** of the log **likelihood function**, with respect to the **conditional distribution** of ***Z*** given ***X*** under the current estimate of the parameters ***θ*^(t)**:

$$Q(\theta|\theta^{(t)}) = \mathbf{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}} [\log L(\theta; \mathbf{X}, \mathbf{Z})]$$

Maximization step (M step): Find the parameter that maximizes this quantity:

$$\boldsymbol{\theta}^{(t+1)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} | \boldsymbol{\theta}^{(t)})$$

In typical models to which EM is applied:

1. The observed data points \mathbf{X} may be **discrete** (taking values in a finite or countably infinite set) or **continuous** (taking values in an uncountably infinite set). Associated with each data point may be a vector of observations.
2. The **missing values** (aka **latent variables**) \mathbf{Z} are **discrete**, drawn from a fixed number of values, and with one latent variable per observed data point.
3. The parameters are continuous, and are of two kinds: Parameters that are associated with all data points, and those associated with a specific value of a latent variable (i.e., associated with all data points which corresponding latent variable has that value).

However, it is possible to apply EM to other sorts of models.

The motive is as follows. If the value of the parameters $\boldsymbol{\theta}$ is known, usually the value of the latent variables \mathbf{Z} can be found by maximizing the log-likelihood over all possible values of \mathbf{Z} , either simply by iterating over \mathbf{Z} or through an algorithm such as the **Viterbi algorithm** for **hidden Markov models**. Conversely, if we know the value of the latent variables \mathbf{Z} , we can find an estimate of the parameters $\boldsymbol{\theta}$ fairly easily, typically by simply grouping the observed data points according to the value of the associated latent variable and averaging the values, or some function of the values, of the points in each group. This suggests an iterative algorithm, in the case where both $\boldsymbol{\theta}$ and \mathbf{Z} are unknown:

1. First, initialize the parameters $\boldsymbol{\theta}$ to some random values.
2. Compute the best value for \mathbf{Z} given these parameter values.
3. Then, use the just-computed values of \mathbf{Z} to compute a better estimate for the parameters $\boldsymbol{\theta}$. Parameters associated with a specific value of \mathbf{Z} will use only those data points which associated latent variable has that value.
4. Iterate steps 2 and 3 until convergence.

The algorithm as just described monotonically approaches a local minimum of the cost function, and is commonly called *hard EM*. The **k-means algorithm** is an example of this class of algorithms.

However, somewhat better methods exist. Rather than making a hard choice for \mathbf{Z} given the current parameter values and averaging only over the set of data points associated with some value of \mathbf{Z} , instead, determine the probability of each possible value of \mathbf{Z} for each data point, and then use the probabilities associated with some value of \mathbf{Z} to compute a **weighted average** over the whole set of data points. The resulting algorithm is commonly called *soft EM*, and is the type of algorithm normally associated with EM. The counts used to compute these weighted averages are called *soft counts* (as opposed to the *hard counts* used in a hard-EM-type algorithm such as *k-means*). The probabilities computed for \mathbf{Z} are **posterior probabilities** and are what is computed in the E step. The soft counts used to compute new parameter values are what is computed in the M step.

Properties [\[edit \]](#)

Speaking of an expectation (E) step is a bit of a **misnomer**. What is calculated in the first step are the fixed, data-dependent parameters of the function Q . Once the parameters of Q are known, it is fully determined and is maximized in the second (M) step of an EM algorithm.

Although an EM iteration does increase the observed data (i.e., marginal) likelihood function, no guarantee exists that the sequence converges to a **maximum likelihood estimator**. For **multimodal distributions**, this means that an EM algorithm may converge to a **local maximum** of the observed data likelihood function, depending on starting values. A variety of heuristic or **metaheuristic** approaches exist to escape a local maximum, such as random-restart **hill climbing** (starting with several different random initial estimates $\boldsymbol{\theta}^{(i)}$), or applying **simulated annealing** methods.

EM is especially useful when the likelihood is an **exponential family**: the E step becomes the sum of expectations of **sufficient statistics**, and the M step involves maximizing a linear function. In such a case, it is usually possible to derive **closed-form expression** updates for each step, using the Sundberg formula (published by Rolf Sundberg using unpublished results of **Per Martin-Löf** and **Anders Martin-Löf**).^{[3][4][7][8][9][10][11]}

The EM method was modified to compute **maximum a posteriori** (MAP) estimates for **Bayesian inference** in the original paper by Dempster, Laird, and Rubin.

Other methods exist to find maximum likelihood estimates, such as **gradient descent**, **conjugate gradient**, or variants of the **Gauss–Newton algorithm**. Unlike EM, such methods typically require the evaluation of first and/or second derivatives of the likelihood function.

Proof of correctness [\[edit \]](#)

Expectation-maximization works to improve $Q(\theta|\theta^{(t)})$ rather than directly improving $\log p(\mathbf{X}|\theta)$. Here is shown that improvements to the former imply improvements to the latter.^[13]

For any \mathbf{Z} with non-zero probability $p(\mathbf{Z}|\mathbf{X}, \theta)$, we can write

$$\log p(\mathbf{X}|\theta) = \log p(\mathbf{X}, \mathbf{Z}|\theta) - \log p(\mathbf{Z}|\mathbf{X}, \theta).$$

We take the expectation over possible values of the unknown data \mathbf{Z} under the current parameter estimate $\theta^{(t)}$ by multiplying both sides by $p(\mathbf{Z}|\mathbf{X}, \theta^{(t)})$ and summing (or integrating) over \mathbf{Z} . The left-hand side is the expectation of a constant, so we get:

$$\begin{aligned} \log p(\mathbf{X}|\theta) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{(t)}) \log p(\mathbf{X}, \mathbf{Z}|\theta) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{(t)}) \log p(\mathbf{Z}|\mathbf{X}, \theta) \\ &= Q(\theta|\theta^{(t)}) + H(\theta|\theta^{(t)}), \end{aligned}$$

where $H(\theta|\theta^{(t)})$ is defined by the negated sum it is replacing. This last equation holds for any value of θ including $\theta = \theta^{(t)}$,

$$\log p(\mathbf{X}|\theta^{(t)}) = Q(\theta^{(t)}|\theta^{(t)}) + H(\theta^{(t)}|\theta^{(t)}),$$

and subtracting this last equation from the previous equation gives

$$\log p(\mathbf{X}|\theta) - \log p(\mathbf{X}|\theta^{(t)}) = Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)}) + H(\theta|\theta^{(t)}) - H(\theta^{(t)}|\theta^{(t)}),$$

However, **Gibbs' inequality** tells us that $H(\theta|\theta^{(t)}) \geq H(\theta^{(t)}|\theta^{(t)})$, so we can conclude that

$$\log p(\mathbf{X}|\theta) - \log p(\mathbf{X}|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)}).$$

In words, choosing θ to improve $Q(\theta|\theta^{(t)})$ beyond $Q(\theta^{(t)}|\theta^{(t)})$ can not cause $\log p(\mathbf{X}|\theta)$ to decrease below $\log p(\mathbf{X}|\theta^{(t)})$, and so the marginal likelihood of the data is non-decreasing.

As a maximization-maximization procedure [\[edit \]](#)

The EM algorithm can be viewed as two alternating maximization steps, that is, as an example of **coordinate ascent**.^{[14][15]} Consider the function:

$$F(q, \theta) := \mathbb{E}_q[\log L(\theta; \mathbf{x}, \mathbf{Z})] + H(q),$$

where q is an arbitrary probability distribution over the unobserved data \mathbf{z} and $H(q)$ is the **entropy** of the distribution q . This function can be written as

$$F(q, \theta) = -D_{\text{KL}}(q \| p_{\mathbf{Z}|\mathbf{X}}(\cdot|\mathbf{x}; \theta)) + \log L(\theta; \mathbf{x}),$$

where $p_{\mathbf{Z}|\mathbf{X}}(\cdot|\mathbf{x}; \theta)$ is the conditional distribution of the unobserved data given the observed data \mathbf{x} and D_{KL} is the **Kullback–Leibler divergence**.

Then the steps in the EM algorithm may be viewed as:

Expectation step: Choose q to maximize F :

$$q^{(t)} = \arg \max_q F(q, \theta^{(t)})$$

Maximization step. Choose θ to maximize F :

$$\theta^{(t+1)} = \arg \max_{\theta} F(q^{(t)}, \theta)$$

Applications [[edit](#)]

EM is frequently used for [data clustering](#) in [machine learning](#) and [computer vision](#). In [natural language processing](#), two prominent instances of the algorithm are the [Baum-Welch algorithm](#) and the [inside-outside algorithm](#) for unsupervised induction of [probabilistic context-free grammars](#).

In [psychometrics](#), EM is almost indispensable for estimating item parameters and latent abilities of [item response theory](#) models.

With the ability to deal with missing data and observe unidentified variables, EM is becoming a useful tool to price and manage risk of a portfolio.^{[[ref?](#)]}

The EM algorithm (and its faster variant [ordered subset expectation maximization](#)) is also widely used in [medical image reconstruction](#), especially in [positron emission tomography](#) and [single photon emission computed tomography](#). See below for other faster variants of EM.

In [structural engineering](#), the Structural Identification using Expectation Maximization (STRIDE) ^[16]algorithm is an output-only method for identifying natural vibration properties of a structural system using sensor data (see [Operational Modal Analysis](#)).

Filtering and smoothing EM algorithms [[edit](#)]

A [Kalman filter](#) is typically used for on-line state estimation and a minimum-variance smoother may be employed for off-line or batch state estimation. However, these minimum-variance solutions require estimates of the state-space model parameters. EM algorithms can be used for solving joint state and parameter estimation problems.

Filtering and smoothing EM algorithms arise by repeating this two-step procedure:

E-step

Operate a Kalman filter or a minimum-variance smoother designed with current parameter estimates to obtain updated state estimates.

M-step

Use the filtered or smoothed state estimates within maximum-likelihood calculations to obtain updated parameter estimates.

Suppose that a [Kalman filter](#) or minimum-variance smoother operates on noisy measurements of a single-input-single-output system. An updated measurement noise variance estimate can be obtained from the [maximum likelihood](#) calculation

$$\hat{\sigma}_v^2 = \frac{1}{N} \sum_{k=1}^N (z_k - \hat{x}_k)^2$$

where \hat{x}_k are scalar output estimates calculated by a filter or a smoother from N scalar measurements z_k . Similarly, for a first-order auto-regressive process, an updated process noise variance estimate can be calculated by

$$\hat{\sigma}_w^2 = \frac{1}{N} \sum_{k=1}^N (\hat{x}_{k+1} - \hat{F}\hat{x}_k)^2$$

where \hat{x}_k and \hat{x}_{k+1} are scalar state estimates calculated by a filter or a smoother. The updated model coefficient estimate is obtained via

$$\hat{F} = \frac{\sum_{k=1}^N (\hat{x}_{k+1} - \hat{F} \hat{x}_k)}{\sum_{k=1}^N \hat{x}_k^2}.$$

The convergence of parameter estimates such as those above are well studied.^{[17][18][19]}

Variants [[edit](#)]

A number of methods have been proposed to accelerate the sometimes slow convergence of the EM algorithm, such as those using [conjugate gradient](#) and modified [Newton's methods](#) (Newton–Raphson).^[20] Also, EM can be used with constrained estimation methods.

Expectation conditional maximization (ECM) replaces each M step with a sequence of conditional maximization (CM) steps in which each parameter θ_i is maximized individually, conditionally on the other parameters remaining fixed.^[21]

This idea is further extended in *generalized expectation maximization (GEM)* algorithm, in which is sought only an increase in the objective function F for both the E step and M step as described in the [As a maximization-maximization procedure](#) section.^[14] GEM is further developed in a distributed environment and shows promising results.^[22]

It is also possible to consider the EM algorithm as a subclass of the **MM** (Majorize/Minimize or Minorize/Maximize, depending on context) algorithm,^[23] and therefore use any machinery developed in the more general case.

α -EM algorithm [[edit](#)]

The Q-function used in the EM algorithm is based on the log likelihood. Therefore, it is regarded as the log-EM algorithm. The use of the log likelihood can be generalized to that of the α -log likelihood ratio. Then, the α -log likelihood ratio of the observed data can be exactly expressed as equality by using the Q-function of the α -log likelihood ratio and the α -divergence. Obtaining this Q-function is a generalized E step. Its maximization is a generalized M step. This pair is called the α -EM algorithm ^[24] which contains the log-EM algorithm as its subclass. Thus, the α -EM algorithm by [Yasuo Matsuyama](#) is an exact generalization of the log-EM algorithm. No computation of gradient or Hessian matrix is needed. The α -EM shows faster convergence than the log-EM algorithm by choosing an appropriate α . The α -EM algorithm leads to a faster version of the Hidden Markov model estimation algorithm α -HMM. ^[25]

Relation to variational Bayes methods [[edit](#)]

EM is a partially non-Bayesian, maximum likelihood method. Its final result gives a [probability distribution](#) over the latent variables (in the Bayesian style) together with a point estimate for θ (either a [maximum likelihood estimate](#) or a posterior mode). A fully Bayesian version of this may be wanted, giving a probability distribution over θ and the latent variables. The Bayesian approach to inference is simply to treat θ as another latent variable. In this paradigm, the distinction between the E and M steps disappears. If using the factorized Q approximation as described above ([variational Bayes](#)), solving can iterate over each latent variable (now including θ) and optimize them one at a time. Now, k steps per iteration are needed, where k is the number of latent variables. For [graphical models](#) this is easy to do as each variable's new Q depends only on its [Markov blanket](#), so local [message passing](#) can be used for efficient inference.

Geometric interpretation [[edit](#)]

For more details on this topic, see [Information geometry](#).

In [information geometry](#), the E step and the M step are interpreted as projections under dual [affine connections](#), called the e-connection and the m-connection; the [Kullback–Leibler divergence](#) can also be understood in these terms.

Examples [\[edit \]](#)

Gaussian mixture [\[edit \]](#)

Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ be a sample of n independent observations from a **mixture** of two **multivariate normal distributions** of dimension d , and let $\mathbf{z} = (z_1, z_2, \dots, z_n)$ be the latent variables that determine the component from which the observation originates.^[15]

$$X_i | (Z_i = 1) \sim \mathcal{N}_d(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \text{ and } \\ X_i | (Z_i = 2) \sim \mathcal{N}_d(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

where

$$\begin{aligned} P(Z_i = 1) &= \tau_1 \text{ and } \\ P(Z_i = 2) &= \tau_2 = 1 - \tau_1 \end{aligned}$$

The aim is to estimate the unknown parameters representing the *mixing* value between the Gaussians and the means and covariances of each:

$$\boldsymbol{\theta} = (\boldsymbol{\tau}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2)$$

where the incomplete-data likelihood function is

$$L(\boldsymbol{\theta}; \mathbf{x}) = \prod_{i=1}^n \sum_{j=1}^2 \tau_j f(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j),$$

and the complete-data likelihood function is

$$L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) = p(\mathbf{x}, \mathbf{z} | \boldsymbol{\theta}) = \prod_{i=1}^n \prod_{j=1}^2 [f(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tau_j]^{\mathbb{I}(z_i=j)}$$

or

$$L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) = \exp \left\{ \sum_{i=1}^n \sum_{j=1}^2 \mathbb{I}(z_i = j) \left[\log \tau_j - \frac{1}{2} \log |\boldsymbol{\Sigma}_j| - \frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_j) - \frac{d}{2} \log(2\pi) \right] \right\}.$$

where \mathbb{I} is an **indicator function** and f is the **probability density function** of a multivariate normal.

To see the last equality, then for each i all indicators $\mathbb{I}(z_i = j)$ are equal to zero, except for one which is equal to one. The inner sum thus reduces to one term.

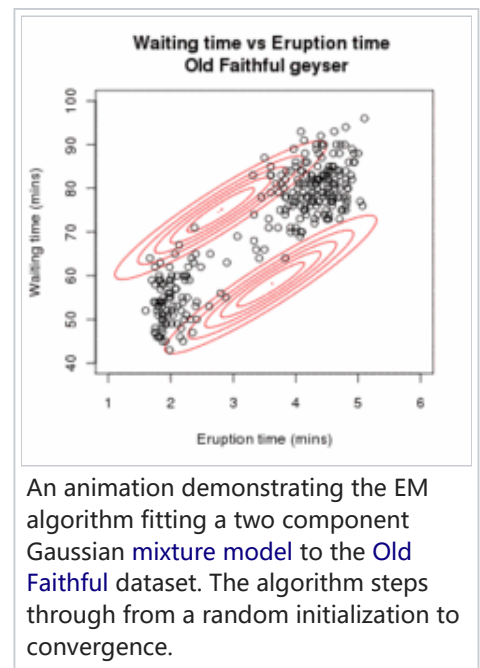
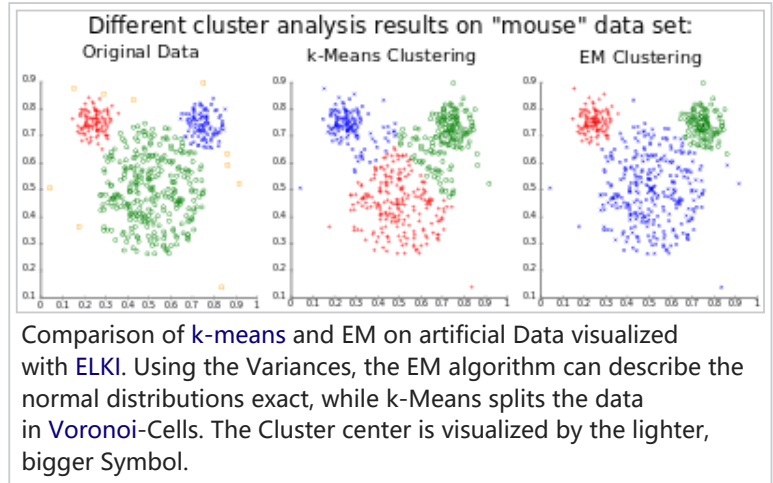
E step [\[edit \]](#)

Given our current estimate of the parameters $\boldsymbol{\theta}^{(t)}$, the conditional distribution of the Z_i s determined by **Bayes theorem** to be the proportional height of the normal **density** weighted by τ .

$$T_{j,i}^{(t)} := P(Z_i = j | X_i = \mathbf{x}_i; \boldsymbol{\theta}^{(t)}) = \frac{\tau_j^{(t)} f(\mathbf{x}_i; \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)})}{\tau_1^{(t)} f(\mathbf{x}_i; \boldsymbol{\mu}_1^{(t)}, \boldsymbol{\Sigma}_1^{(t)}) + \tau_2^{(t)} f(\mathbf{x}_i; \boldsymbol{\mu}_2^{(t)}, \boldsymbol{\Sigma}_2^{(t)})}.$$

These are called the "membership probabilities" which are normally considered the output of the E step (although this is not the Q function of below).

This E step corresponds with this function for Q:



$$\begin{aligned}
Q(\theta|\theta^{(t)}) &= \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}} [\log L(\theta; \mathbf{x}, \mathbf{Z})] \\
&= \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}} [\log \prod_{i=1}^n L(\theta; \mathbf{x}_i, \mathbf{z}_i)] \\
&= \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}} [\sum_{i=1}^n \log L(\theta; \mathbf{x}_i, \mathbf{z}_i)] \\
&= \sum_{i=1}^n \mathbb{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}} [\log L(\theta; \mathbf{x}_i, \mathbf{z}_i)] \\
&= \sum_{i=1}^n \sum_{j=1}^2 P(Z_i = j | X_i = \mathbf{x}_i; \theta^{(t)}) \log L(\theta_j; \mathbf{x}_i, \mathbf{z}_i) \\
&= \sum_{i=1}^n \sum_{j=1}^2 T_{j,i}^{(t)} [\log \tau_j - \frac{1}{2} \log |\Sigma_j| - \frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_j) - \frac{d}{2} \log(2\pi)]
\end{aligned}$$

This full conditional expectation does not need to be calculated in one step, because τ and $\boldsymbol{\mu}/\Sigma$ appear in separate linear terms and can thus be maximized independently.

M step [\[edit \]](#)

$Q(\theta|\theta^{(t)})$ being quadratic in form means that determining the maximizing values of θ is relatively straightforward. Also, τ , $(\boldsymbol{\mu}_1, \Sigma_1)$ and $(\boldsymbol{\mu}_2, \Sigma_2)$ may all be maximized independently since they all appear in separate linear terms.

To begin, consider τ , which has the constraint $\tau_1 + \tau_2 = 1$:

$$\begin{aligned}
\boldsymbol{\tau}^{(t+1)} &= \arg \max_{\boldsymbol{\tau}} Q(\theta|\theta^{(t)}) \\
&= \arg \max_{\boldsymbol{\tau}} \left\{ \left[\sum_{i=1}^n T_{1,i}^{(t)} \right] \log \tau_1 + \left[\sum_{i=1}^n T_{2,i}^{(t)} \right] \log \tau_2 \right\}
\end{aligned}$$

This has the same form as the MLE for the [binomial distribution](#), so

$$\tau_j^{(t+1)} = \frac{\sum_{i=1}^n T_{j,i}^{(t)}}{\sum_{i=1}^n (T_{1,i}^{(t)} + T_{2,i}^{(t)})} = \frac{1}{n} \sum_{i=1}^n T_{j,i}^{(t)}.$$

For the next estimates of $(\boldsymbol{\mu}_1, \Sigma_1)$:

$$\begin{aligned}
(\boldsymbol{\mu}_1^{(t+1)}, \Sigma_1^{(t+1)}) &= \arg \max_{\boldsymbol{\mu}_1, \Sigma_1} Q(\theta|\theta^{(t)}) \\
&= \arg \max_{\boldsymbol{\mu}_1, \Sigma_1} \sum_{i=1}^n T_{1,i}^{(t)} \left\{ -\frac{1}{2} \log |\Sigma_1| - \frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_1)^\top \Sigma_1^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_1) \right\}.
\end{aligned}$$

This has the same form as a weighted MLE for a normal distribution, so

$$\boldsymbol{\mu}_1^{(t+1)} = \frac{\sum_{i=1}^n T_{1,i}^{(t)} \mathbf{x}_i}{\sum_{i=1}^n T_{1,i}^{(t)}} \text{ and } \Sigma_1^{(t+1)} = \frac{\sum_{i=1}^n T_{1,i}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_1^{(t+1)}) (\mathbf{x}_i - \boldsymbol{\mu}_1^{(t+1)})^\top}{\sum_{i=1}^n T_{1,i}^{(t)}}$$

and, by symmetry

$$\boldsymbol{\mu}_2^{(t+1)} = \frac{\sum_{i=1}^n T_{2,i}^{(t)} \mathbf{x}_i}{\sum_{i=1}^n T_{2,i}^{(t)}} \text{ and } \Sigma_2^{(t+1)} = \frac{\sum_{i=1}^n T_{2,i}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_2^{(t+1)}) (\mathbf{x}_i - \boldsymbol{\mu}_2^{(t+1)})^\top}{\sum_{i=1}^n T_{2,i}^{(t)}}.$$

Termination [\[edit \]](#)

Conclude the iterative process if $E_{Z|\theta^{(t)}, \mathbf{x}}[\log L(\theta^{(t)}; \mathbf{x}, \mathbf{Z})] \leq E_{Z|\theta^{(t-1)}, \mathbf{x}}[\log L(\theta^{(t-1)}; \mathbf{x}, \mathbf{Z})] + \epsilon$ for ϵ below some preset threshold.

Generalization [[edit](#)]

The algorithm illustrated above can be generalized for mixtures of more than two [multivariate normal distributions](#).

Truncated and censored regression [[edit](#)]

The EM algorithm has been implemented in the case where an underlying [linear regression](#) model exists explaining the variation of some quantity, but where the values actually observed are censored or truncated versions of those represented in the model.^[26] Special cases of this model include censored or truncated observations from one [normal distribution](#).^[26]

Alternatives [[edit](#)]

EM typically converges to a local optimum, not necessarily the global optimum, with no bound on the convergence rate in general. It is possible that it can be arbitrarily poor in high dimensions and there can be an exponential number of local optima. Hence, a need exists for alternative methods for guaranteed learning, especially in the high-dimensional setting. Alternatives to EM exist with better guarantees for consistency, which are termed *moment-based approaches* or the so-called *spectral techniques*. Moment-based approaches to learning the parameters of a probabilistic model are of increasing interest recently since they enjoy guarantees such as global convergence under certain conditions unlike EM which is often plagued by the issue of getting stuck in local optima. Algorithms with guarantees for learning can be derived for a number of important models such as mixture models, HMMs etc. For these spectral methods, no spurious local optima occur, and the true parameters can be consistently estimated under some regularity conditions.