Expectation-maximization algorithm

An expectation-maximization (EM) algorithm is used in statistics for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. EM alternates between performing an expectation (E) step, which computes an expectation of the likelihood by including the latent variables as if they were observed, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E step. The parameters found on the M step are then used to begin another E step, and the process is repeated.

History

The EM algorithm was explained and given its name in a classic 1977 paper by Arthur Dempster, Nan Laird, and Donald Rubin in the Journal of the Royal Statistical Society (see reference below). They pointed out that method had been "proposed many times in special circumstances" by other authors, but the 1977 paper generalized the method and developed the theory behind it.

Applications

EM is frequently used for data clustering in machine learning and computer vision.

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.

Demonstrations and activities

Various 1D, 2D and 3D demonstrations of EM together with Mixture Modeling are provided as part of the paired SOCR activities and applets. These applets and activities show empirically the properties of the EM algorithm for parameter estimation in diverse settings.

Specification of the EM procedure

Let \( \mathbf{Y} \) denote incomplete data consisting of values of observable variables and \( \mathbf{Z} \) the missing data. Together, \( \mathbf{Z} \) and \( \mathbf{Y} \) form the complete data. \( \mathbf{Z} \) can either be actual missing measurements or a hidden variable that would make the problem easier if its value were known. For instance, in mixture models, the likelihood formula would be much more convenient if mixture components that "generated" the samples were known (see example below).

Let \( \mathbf{P} \) be the joint probability density function (continuous case) or probability mass function (discrete case) of the complete data with parameters given by the vector \( \theta \):
This function can also be considered as the complete data likelihood, that is, it can be thought of as a function of \( \theta \). Further, note that the conditional distribution of the missing data given the observed can be expressed as:

\[
p(z|y, \theta) = \frac{p(y, z|\theta)}{p(y|\theta)} = \frac{p(y|z, \theta)p(z|\theta)}{\int p(y|\hat{z}, \theta)p(\hat{z}|\theta)\,d\hat{z}}
\]

by using the Bayes rule and the law of total probability. (This formulation only requires knowledge of the observation likelihood given the unobservable data \( p(y|Z, \theta) \) and the probability of the unobservable data \( p(z|\theta) \).)

An EM algorithm iteratively improves an initial estimate \( \theta_0 \) by constructing new estimates \( \theta_1, \theta_2, \) and so on. An individual re-estimation step that derives \( \theta_{n+1} \) from \( \theta_n \) takes the following form:

\[
\theta_{n+1} = \arg \max_{\theta} Q(\theta)
\]

where \( Q(\theta) \) is the expected value of the log-likelihood. In other words, we do not know the complete data, so we cannot say what is the exact value of the likelihood, but given the data that we do know (the \( y \)'s), we can find a posteriori estimates of the probabilities for the various values of the unknown \( z \)'s. For each set of \( z \)'s there is a likelihood value for \( \theta \), and we can thus calculate an expected value of the likelihood with the given values of \( y \)'s (and which depends on the previously assumed value of \( \theta \) because this influenced the probabilities of the \( z \)'s).

\( Q \) is given by

\[
Q(\theta) = \sum_z p(z|y, \theta_n) \log p(y, z|\theta)
\]

or more generally

\[
Q(\theta) = E_z[\log p(y, z|\theta) | y],
\]

where it is understood that this denotes the conditional expectation of \( \log p(y, z|\theta) \) being taken with the \( \theta \) used in the conditional distribution of \( Z \) fixed at \( \theta_n \). (The log of the likelihood is often used instead of true likelihood because it leads to easier formulas, but still attains its maximum at the same point as the likelihood.)

In other words, \( \theta_{n+1} \) is the value that maximizes (M) the conditional expectation (E) of the complete data log-likelihood given the observed variables under the previous parameter value. The expectation \( Q(\theta) \) in the continuous case would be given by

\[
Q(\theta) = E_z[\log p(y, z|\theta) | y] = \int_{-\infty}^{\infty} p(z|y, \theta_n) \log p(y, z|\theta) \, dz
\]
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.

The origin of the name comes from the fact that in the paper by Dempster, Laird and Rubin, they first discuss a less general problem in which the probability distribution is of the exponential family, and in that case the so-called E step consists of finding the expected values of certain sufficient statistics of the complete data.

Properties

Part of the reason for the popularity of EM algorithms is that, it can be shown, an EM iteration does not decrease the observed data likelihood function. However, there is no guarantee that the sequence converges to a maximum likelihood estimator. For multimodal distributions, this means that an EM algorithm will converge to a local maximum (or saddle point) of the observed data likelihood function, depending on starting values. There are a variety of heuristic approaches for escaping a local maximum such as using several different random initial estimates, $\theta_0$, or applying simulated annealing.

EM is particularly useful when maximum likelihood estimation of a complete data model is easy. If closed-form estimators exist, the M step is often trivial. A classic example is maximum likelihood estimation of a finite mixture of Gaussians, where each component of the mixture can be estimated trivially if the mixing distribution is known.

"Expectation-maximization" is a description of a class of related algorithms, not a specific algorithm; EM is a recipe or meta-algorithm which is used to devise particular algorithms. The Baum-Welch algorithm is an example of an EM algorithm applied to hidden Markov models. Another example is the EM algorithm for fitting a mixture density model.

An EM algorithm can also find maximum a posteriori (MAP) estimates, by performing MAP estimation in the M step, rather than maximum likelihood.

There are other methods for finding maximum likelihood estimates, such as gradient descent, conjugate gradient or variations of the Gauss-Newton method. Unlike EM, such methods typically require the evaluation of first and/or second derivatives of the likelihood function.

Incremental versions

The classic EM procedure is to replace both $Q$ and $\theta$ with their optimal possible (argmax) values at each iteration. However it can be shown (see Neal & Hinton, 1999) that simply finding $Q$ and $\theta$ to give some improvement over their current value will also ensure successful convergence.
For example, to improve $Q$, we could restrict the space of possible functions to a computationally simple distribution such as a factorial distribution,

$$Q = \prod_i Q_i.$$

Thus at each E step we compute the variational approximation of $Q$.

To improve $\theta$, we could use any hill-climbing method, and not worry about finding the optimal $\theta$, just some improvement. This method is also known as Generalized EM (GEM).

**Relation to variational Bayes methods**

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 $\theta$ (either a maximum likelihood estimate or a posterior mode). We may want a fully Bayesian version of this, giving a probability distribution over $\theta$ as well as the latent variables. In fact the Bayesian approach to inference is simply to treat $\theta$ as another latent variable. In this paradigm, the distinction between the E and M steps disappears. If we use the factorized $Q$ approximation as described above (variational Bayes), we may iterate over each latent variable (now including $\theta$) and optimize them one at a time. There are now $k$ steps per iteration, 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.

**Example: Mixture Gaussian**

Assume that a sample of $m$ vectors (or scalars) $Y_1, \cdots, Y_m$, where $Y_j \in \mathbb{R}^l$, is drawn from one of $n$ Gaussian distributions. Let $z_j$ denote which Gaussian $y_j$ is from, so it can have a value of 1 or 2 or any integer up to $n$. The probability for a particular $y$ if it comes from the $i$-th Gaussian is

$$P(y|z_j = i, \theta) = \mathcal{N}(\mu_i, \sigma_i) = (2\pi)^{-l/2} |\sigma_i|^{-1/2} \exp \left( -\frac{1}{2}(y - \mu_i)^T \sigma_i^{-1} (y - \mu_i) \right)$$

Our task is to estimate the unknown parameters $\theta = \{\mu_1, \ldots, \mu_n, \sigma_1, \ldots, \sigma_n, P(z = 1), \ldots, P(z = n)\}$, that is, the means and standard deviations of each Gaussian and the probability for each Gaussian being drawn for any given point. (Actually it is not clear that we should allow the standard deviations to take any value because then the maximum likelihood may be unbounded as one centres a particular Gaussian on a particular data point and decreases the standard deviation toward zero!)

**E-step**
Estimation of the unobserved $z$'s (which Gaussian is used), conditioned on the observation, using the values from the last maximisation step:

$$p(z_j = i | y_j, \theta_t) = \frac{p(y_j = i, z_j | \theta_t)}{\sum_{k=1}^{n} p(y_j = k, \theta_t) p(z_j = k | \theta_t)}$$

**M-step**

We now want to maximize the expected log-likelihood of the joint event:

$$Q(\theta) = E_z \left[ \ln \prod_{j=1}^{m} p(y_j, z | \theta) | y_j \right]$$

$$= E_z \left[ \sum_{j=1}^{m} \ln p(y_j, z | \theta) | y_j \right]$$

$$= \sum_{j=1}^{m} E_z \left[ \ln p(y_j, z | \theta) | y_j \right]$$

$$= \sum_{j=1}^{m} \sum_{i=1}^{n} p(z_j = i | y_j, \theta_t) \ln p(z_j = i, y_j | \theta)$$

If we expand the probability of the joint event, we get

$$Q(\theta) = \sum_{j=1}^{m} \sum_{i=1}^{n} p(z_j = i | y_j, \theta_t) \ln (p(y_j | z_j = i, \theta) p(z_j = i | \theta))$$

We have the constraint

$$\sum_{i=1}^{n} p(z_j = i | \theta) = 1$$

If we add a Lagrange multiplier, and expand the pdf, we get

$$L(\theta) = \left( \sum_{j=1}^{m} \sum_{i=1}^{n} p(z_j = i | y_j, \theta_t) \left( -\frac{1}{2} \ln(2\pi) - \frac{1}{2} \ln | \sigma_i | - \frac{1}{2} (y_j - \mu_i)^T \sigma_i^{-1} (y_j - \mu_i) + \ln p(z_j = i | \theta) \right) \right) - \lambda \left( \sum_{i=1}^{n} p(z_j = i | \theta) - 1 \right)$$

To find the new estimate $\theta_{t+1}$, we find a maximum where $\frac{\partial L(\theta)}{\partial \theta} = 0$.

New estimate for mean (using some differentiation rules from matrix calculus):
\[
\frac{\partial \mathcal{L}(\theta)}{\partial \mu_i} = \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \left( - \frac{\partial}{\partial \mu_i} \frac{1}{2} (y_j - \mu_i)^T \sigma_i^{-1} (y_j - \mu_i) \right) \\
= \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \left( - \frac{1}{2} \sigma_i^{-1} + \sigma_i^{-T} \right) (y_j - \mu_i)(-1) \\
= \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \left( \sigma_i^{-1}(y_j - \mu_i) \right) \\
= 0 \\
\downarrow \\
\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \sigma_i^{-1} \mu_i = \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \sigma_i^{-1} y_j \\
\downarrow \\
\mu_i = \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) y_j \\
\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \\
\mu_i = \frac{\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) y_j}{\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i)}
\]

New estimate for covariance:

\[
\frac{\partial \mathcal{L}(\theta)}{\partial \sigma_i} = \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \left( - \frac{\partial}{\partial \sigma_i} \frac{1}{2} \ln|\sigma_i| - \frac{\partial}{\partial \sigma_i} \frac{1}{2} (y_j - \mu_i)^T \sigma_i^{-1} (y_j - \mu_i) \right) \\
= \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \left( - \frac{1}{2} \sigma_i^{-T} + \frac{1}{2} \mu_i^{-T}(y_j - \mu_i)(y_j - \mu_i)^T \sigma_i^{-2} \right) \\
= 0 \\
\downarrow \\
\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \sigma_i^{-1} = \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \sigma_i^{-1}(y_j - \mu_i)(y_j - \mu_i)^T \sigma_i^{-1} \\
\downarrow \\
\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) = \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \sigma_i^{-1}(y_j - \mu_i)(y_j - \mu_i)^T \\
\downarrow \\
\sigma_i = \frac{\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i)(y_j - \mu_i)(y_j - \mu_i)^T}{\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i)}
\]

New estimate for class probability:
\[
\frac{\partial \mathcal{L}(\theta)}{\partial p(z_j = i|\theta)} = \left( \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \frac{\partial \ln p(z_j = i|\theta)}{\partial p(z_j = i|\theta)} \right) - \lambda \left( \frac{\partial p(z_j = i|\theta)}{\partial p(z_j = i|\theta)} \right) - 0
\]

\[
\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i) \frac{1}{p(z_j = i|\theta)} = \lambda
\]

\[
p(z_j = i|\theta) = \frac{1}{\lambda} \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i)
\]

Inserting into the constraint:

\[
\sum_{i=1}^{n} p(z_j = i|\theta) = \sum_{i=1}^{n} \frac{1}{\lambda} \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i)
\]

\[
= 1
\]

\[
\lambda = \sum_{i=1}^{n} \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i)
\]

Inserting \( \lambda \) into our estimate:

\[
p(z_j = i|\theta) = \frac{1}{\lambda} \sum_{j=1}^{m} p(z_j = i|y_j, \theta_i)
\]

\[
= \frac{\sum_{j=1}^{m} p(z_j = i|y_j, \theta_i)}{\sum_{k=1}^{n} \sum_{j=1}^{m} p(z_j = k|y_j, \theta_i)}
\]

These estimates now become our \( \theta_{t+1} \), to be used in the next estimation step.