**Tutorial on EM Algorithm**

Loc Nguyen

Sunflower Soft Company, Vietnam

Email: ngphloc@sunflowersoft.net

# **Abstract**

Maximum likelihood estimation (MLE) is a popular method for parameter estimation in both applied probability and statistics but MLE cannot solve the problem of incomplete data or hidden data because it is impossible to maximize likelihood function from hidden data. Expectation maximum (EM) algorithm is a powerful mathematical tool for solving this problem if there is a relationship between hidden data and observed data. Such hinting relationship is specified by a mapping from hidden data to observed data or by a joint probability between hidden data and observed data. In other words, the relationship helps us know hidden data by surveying observed data. The essential ideology of EM is to maximize the expectation of likelihood function over observed data based on the hinting relationship instead of maximizing directly the likelihood function of hidden data. Pioneers in EM algorithm proved its convergence. As a result, EM algorithm produces parameter estimators as well as MLE does. This tutorial aims to provide explanations of EM algorithm in order to help researchers comprehend it. Moreover some improvements of EM algorithm are also proposed in the tutorial such as combination of EM and third-order convergence Newton-Raphson process, combination of EM and gradient descent method, and combination of EM and particle swarm optimization (PSO) algorithm.

**Keywords**: expectation maximization, EM, generalized expectation maximization, GEM, EM convergence.

# **1. Introduction**

Literature of expectation maximization (EM) algorithm in this tutorial is mainly extracted from the preeminent article “Maximum Likelihood from Incomplete Data via the EM Algorithm” by Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin (Dempster, Laird, & Rubin, 1977). For convenience, let **DLR** be reference to such three authors.

We begin a review of EM algorithm with some basic concepts. Before discussing main subjects, there are some conventions. For example, if there is no additional explanation, variables are often denoted by letters such as *x*, *y*, *z*, *X*, *Y*, and *Z* whereas values and constants are often denoted by letters such as *a*, *b*, *c*, *A*, *B*, and *C*. Parameters are often denoted as Greek letters such as *α*, *β*, *γ*, Θ, Φ, and Ψ. Uppercase letters often denote vectors and matrices (multivariate quantities) whereas lowercase letters often denote scalars (univariate quantities). Script letters such as and often denote data samples. Bold and uppercase letters such as ***X*** and ***R*** denote algebraic structures such as spaces, fields, and domains. Moreover, bold and lowercase letters such as ***x***, ***y***, ***z***, ***a***, ***b***, and ***c*** may denote vectors. Bold and uppercase letters such as ***X***, ***Y***, ***Z***, ***A***, ***B***, and ***C*** may denote matrices.

By default, vectors are column vectors although a vector can be column vector or row vector. For example, given two vectors *X* and *Y* and two matrices *A* and *B*:

*X* and *Y* above are column vectors. A row vector is represented as follows:

Matrix *A* is square if *m* = *n*. Matrix Λ is diagonal if it is square and its elements outside the main diagonal are zero:

Let *I* be identity matrix or unit matrix, as follows:

Note, *I* is diagonal and its diagonal elements are 1.

Vector addition and matrix addition are defined like numerical addition:

(if *n* = *k*)

Vector and matrix can be multiplied with a scalar.

Let superscript “*T*” denote transposition operator for vector and matrix, as follows:

Transposition operator is linear with addition operator as follows:

Dot product or scalar product of two vectors can be written with transposition operator, as follows:

However, the product *XYT* results out a matrix as follows:

The length of module of vector *X* in Euclidean space is:

The notation |.| also denotes absolute value of scalar and determinant of square matrix; for example, we have |–1| = 1 and |*A*| which is determinant of given square matrix *A*. Note, determinant is only defined for square matrix. If *A* has nonzero determinant (≠0), there exists its inverse denoted *A*–1 such that:

Where *I* is identity matrix. If matrix *A* has its inverse, *A* is called invertible or non-singular. In general, square matrix *A* is invertible is equivalent to the event that its determinant is nonzero (≠0). There are many documents which guide to determine inverse of invertible matrix.

Given invertible matrix *A*, it is called orthogonal matrix if *A*–1 *= AT*, which means *AA*–1 *= A*–1*A* = *AAT* = *ATA* = *I*.

The product of two matrices (multiplication operator) is:

Square matrix *A* is symmetric if *aij* = *aji* for all *i* and *j*. If *A* is symmetric then, *AT* = *A*. If both *A* and *B* are symmetric then, they are commutative such that *AB* = *BA*. Given invertible matrix *A*, if it is symmetric, its inverse *A*–1 is symmetric too.

Given *N* matrices *Ai* such that their product (multiplication operator) is valid, we have:

Given square matrix *A*, tr(*A*) is trace operator which takes sum of diagonal its elements.

Given invertible matrix *A* (*n* rows and *n* columns), Jordan decomposition theorem (Hardle & Simar, 2013, p. 63) stated that *A* is always decomposed as follows

Where *U* is orthogonal matrix composed of eigenvectors. Hence, *U* is called eigenvector matrix.

There are *n* column eigenvectors *ui* = (*u*11, *u*12,…, *u*1*n*) in *U* and they are mutually orthogonal, *uiTuj* = 0. Where Λ is diagonal matrix composed of eigenvalues. Hence, Λ is called eigenvalue matrix.

Where *λi* are eigenvalues. When invertible matrix *A* is decomposed according to Jordan decomposition, we call *A* is diagonalized. If *A* can be diagonalized, it is called diagonalizable matrix. Of course, if *A* is invertible, *A* is diagonalizable. There are many documents for matrix diagonalization.

Given symmetric matrix *A*, it is positive (negative) definite if and only if *XTAX* > 0 (*XTAX* < 0) for all vector *X*≠**0***T*. It is positive (negative) semi-definite if and only if *XTAX* ≥ 0 (*XTAX* ≤ 0) for all vector *X*. When diagonalizable *A* is diagonalized into *U*Λ*UT*, it is positive (negative) definite if and only if all eigenvalues in Λ are positive (negative). Similarly, it is positive (negative) semi-definite if and only if all eigenvalues in Λ are non-negative (non-positive). If *A* is degraded as a scalar, concepts “positive definite”, “positive semi-definite”, “negative definite”, and “negative semi-definite” becomes concepts “positive”, “non-negative”, “negative”, and “non-positive”, respectively.

Suppose *f*(*X*) is scalar-by-vector function, for instance, *f*: ***R****r* → ***R*** where ***R****r* is *r*-dimensional real vector space. The first-order derivative of *f*(*X*) is gradient vector as follows:

Where is partial first-order derivative of *f* with regard to *xi*. So gradient vector is row vector. The second-order derivative of *f*(*X*) is called Hessian matrix as follows:

If every *xi* is univariate then,

Hessian matrix is square matrix. Function *f*(*X*) is called *n*th-order analytic function or *n*th-order smooth function if there is existence and continuity of *k*th-order derivatives of *f*(*X*) where *k* = 1, 2,…, *n*. Function *f*(*X*) is called smooth enough function if *n* is large enough. According to Schwarz’s theorem (Wikipedia, Symmetry of second derivatives, 2018), if *f*(*X*) is second-order smooth function then, its Hessian matrix is symmetric. When *X* is univariate, gradient vector and Hessian matrix are degraded as scalar values. Without loss of generality, by default, variable *X* in this research is multivariate as vector if there is no additional explanation.

Given *f*(*X*) second-order smooth function, *f*(*X*) is convex (strictly convex) in domain ***X*** if its Hessian matrix is semi-positive definite (positive definite) in ***X***.Similarly, *f*(*X*) is concave (strictly concave) in domain ***X*** if its Hessian matrix is semi-negative definite (negative definite) in ***X***. Extreme point, optimized point, optimal point, or optimizer *X*\* is minimum point (minimizer) of convex function and is maximum point (maximizer) of concave function.

Given second-order smooth function *f*(*X*), function *f*(*X*) has stationary point *X*\* if its gradient vector at *X*\* is zero, *Df*(*X*\*) = **0***T*. The stationary point *X*\* is local minimum point if Hessian matrix at *X*\* that is *D*2*f*(*X*\*) is positive definite. Otherwise, the stationary point *X*\* is local maximum point if Hessian matrix at *X*\* that is *D*2*f*(*X*\*) is negative definite. If a stationary point *X*\* is neither minimum point nor maximum point, it is saddle point in which *Df*(*X*\*) = **0***T* and *D*2*f*(*X*\*) = [**0**] where [**0**] denotes zero matrix whose all elements are zero. Finding extreme point (minimum point or maximum point) is optimization problem. Therefore, if *f*(*X*) is second-order smooth function and its gradient vector *Df*(*X*) and Hessian matrix *D*2*f*(*X*) and are both determined, the optimization problem is processed by solving the equation created from setting the gradient *Df*(*X*) to be zero (*Df*(*X*)=**0***T*) and then checking if the Hessian matrix *Df*(*X*\*) is positive definite or negative definite where *X*\* is solution of equation *Df*(*X*)=**0***T*. If such equation cannot be solved due to its complexity, there are some popular methods to solve optimization problem such as Newton-Raphson (Burden & Faires, 2011, pp. 67-71) and gradient descent (Ta, 2014).

A short description of Newton-Raphson method is necessary because it is helpful to solve the equation *Df*(*X*)=**0***T* for optimization problem in practice, especially in case that there is no algebraic formula for solution of such equation. Suppose *f*(*X*) is second-order smooth function, according to first-order Taylor series expansion of *Df*(*X*) at *X*=*X*0 with very small residual, we have:

Because *f*(*X*) is second-order smooth function, *D*2*f*(*X*0) is symmetric matrix according to Schwarz’s theorem (Wikipedia, Symmetry of second derivatives, 2018), which implies:

*D*2*f*(*X*0) = (*D*2*f*(*X*0))*T*

So, we have:

We expect that *Df*(*X*) = **0***T* so that *X* is a solution.

It implies:

This means:

Therefore, Newton-Raphson method starts with an arbitrary value of *X*0 as a solution candidate and then goes through some iterations. Suppose at *k*th iteration, the current value is *Xk* and the next value *Xk*+1 is calculated based on following equation:

The value *Xk* is solution of *Df*(*X*)=**0***T* if *Df*(*Xk*)=**0***T* which means that *Xk*+1=*Xk* after some iterations. At that time is *Xk*+1 = *Xk* = *X*\* the local optimized point (local extreme point). So, the terminated condition of Newton-Raphson method is *Df*(*Xk*)=**0***T*.

Newton-Raphson method computes second-order derivative *D*2*f*(*X*) but gradient descent (Ta, 2014) does not. This difference is not significant but a short description of gradient descent method is necessary because it is also an important method to solve the optimization problem in case that solving directly the equation *Df*(*X*)=**0***T* is too complicated. Gradient descent method is also iterative method starting with an arbitrary value of *X*0 as a solution candidate. Suppose at *k*thiteration, the next candidate point *Xk*+1 is computed based on the current *Xk* as follows (Ta, 2014):

The direction *Dk* is called descending direction, which is the opposite of gradient of *f*(*X*). Hence, we have *Dk* = –*Df*(*Xk*). The value *tk* is the length of the descending direction *Dk*. The value *tk* is often selected an minimizer (maximizer) of function *g*(*t*) = *f*(*Xk* + *tDk*) for minimization (maximization) where *Xk* and *Dk* are known at *k*thiteration. Alternately, *tk* is selected by some advanced condition such as Barzilai–Borwein condition (Wikipedia, Gradient descent, 2018). After some iterations, point *Xk* converges to the local optimizer *X\** when *Dk* = **0***T*. At that time is we have *Xk*+1 = *Xk* = *X*\*. So, the terminated condition of Newton-Raphson method is *Dk*=**0***T*.

In the case that an optimization problem has some constraints, Lagrange duality (Jia, 2013) is applied to solve this problem. Given first-order smooth function *f*(*X*) and constraints *gi*(*X*) ≤ 0 and *hj*(*X*) = 0, the optimization problem is stated as follows:

A so-called Lagrange function *la*(*X*, *λ*, *μ*) is established as sum of *f*(*X*) and constraints multiplied by Lagrange multipliers *λ* and *μ*. In case of minimization problem, *la*(*X*, *λ*, *μ*) is

In case of maximization problem, *la*(*X*, *λ*, *μ*) is

Where all *λi* ≥ 0. Note, *λ* = (*λ*1, *λ*2,…, *λm*)*T* and *μ* = (*μ*1, *μ*2,…, *μm*)*T* are called Lagrange multipliers and *la*(*X*, *λ*, *μ*) is function of *X*, *λ*, and *μ*. Thus, optimizing *f*(*X*) with subject to constraints *gi*(*X*) ≤ 0 and *hj*(*X*) = 0 is equivalent to optimize *la*(*X*, *λ*, *μ*), which is the reason that this method is called Lagrange duality. If *f*(*X*) is first-order smooth function, *la*(*X*, *λ*, *μ*) is also first-order smooth function. In case of minimization problem, the gradient of *la*(*X*, *λ*, *μ*) with regard to *X* is

In case of maximization problem, the gradient of *la*(*X*, *λ*, *μ*) with regard to *X* is

According to KKT condition (Wikipedia, Karush–Kuhn–Tucker conditions, 2014), a local optimized point (local extreme point) *X*\* is solution of the following equation system:

The last equation in the KKT system above is called complementary slackness. The main task of KKT problem is to solve the first equation *Dla*(*X*, *λ*, *μ*) = **0***T*. Again some practical methods such as Newton-Raphson method can be used to solve the equation *Dla*(*X*, *λ*, *μ*) = **0***T*. Alternately, gradient descent method can be used to optimize *la*(*X*, *λ*, *μ*) with constraints specified in KKT system.

Let *P*(.) denote probability, we need to skim some essential probabilistic rules such as additional rule, multiplication rule, total probability rule, and Bayes’ rule. Given two random events (or random variables) *X* and *Y*, the additional rule (Montgomery & Runger, 2003, p. 33) and multiplication rule (Montgomery & Runger, 2003, p. 44) are expressed as follows:

Where notations and denote union operator and intersection operator in set theory (Wikipedia, Set (mathematics), 2014). Your attention please, when *X* and *Y* are numerical variables, notations and also denote operators “or” and “and” in theory logic (Rosen, 2012, pp. 1-12). The probability *P*(*X*, *Y*) is often known as joint probability.

If *X* and *Y* are mutually exclusive () then, is often denoted as *X*+*Y* and we have:

(Due to *P*(Ø) = 0)

*X* and *Y* are mutually independent if and only if one of three following conditions is satisfied:

When *X* and *Y* are mutually independent, are often denoted as *XY* and we have:

Given a complete set of mutually exclusive events *X*1, *X*2,…, *Xn* such that

The total probability rule (Montgomery & Runger, 2003, p. 44) is specified as follows:

If *X* and *Y* are continuous variables, the total probability rule is re-written in integral form as follows:

Note, *P*(*Y|X*) and *P*(*X*) are continuous functions known as probability density functions mentioned later. The important Bayes’ rule will also be mentioned later.

A variable *X* is called random variable if it conforms a probabilistic distribution which is specified by a probability density function (PDF) or a cumulative distribution function (CDF) (Montgomery & Runger, 2003, p. 64) (Montgomery & Runger, 2003, p. 102) but CDF and PDF have the same meaning and they share interchangeable property when PDF is derivative of CDF; in other words, CDF is integral of PDF. In practical statistics, PDF is used more common than CDF is used and so, PDF is mentioned over the whole report. When *X* is discrete, PDF is degraded as probability of *X*. Note, notation *P*(.) often denotes probability and it can be used to denote PDF but we prefer to use lower case letters such as *f* and *g* to denote PDF. Given a random variable having PDF *f*, we often state that “such variable has distribution *f* or such variable has density function *f*”. Let *F*(*X*) and *f*(*X*) be CDF and PDF, respectively, equation 1.1 is the definition of CDF and PDF.

|  |  |
| --- | --- |
|  | (1.1) |

Equation 1.1 defines CDF and PDF for univariate random variable and so it is easy to expend it for multivariate variable when *X* is vector. Let *X* = (*x*1, *x*2,…, *xn*)*T* be *n-*dimension random vector, its CDF and PDF are re-defined as follows:

|  |  |
| --- | --- |
| Continuous case:  Discrete case: | (1.2) |

Marginal PDF of partial variable *xi* where *xi* is a component of *X* is the integral of *f*(*X*) over all *xj* except *xi*.

Where,

Joint PDF of *xi* and *xj* is defined as the integral of *f*(*X*) over all *xk* except *xi* and *xj*.

Where,

Given random variable *X* and its PDF *f*(*X*), theoretical expectation *E*(*X*) and theoretical variance *V*(*X*) of *X* are:

|  |  |
| --- | --- |
|  | (1.3) |

|  |  |
| --- | --- |
|  | (1.4) |

The expectation *E*(*X*) of *X* is often called theoretical mean. When *X* is multivariate vector, *E*(*X*) is mean vector and *V*(*X*) is covariance matrix. Note, covariance metric is always symmetric and invertible. As usual, *E*(*X*) and *V*(*X*) are often denoted as *μ* and Σ, respectively if they are parameters of PDF. When *X* is univariate, *E*(*X*) and *V*(*X*) are scalars and *V*(*X*) is often denoted *σ*2 (if it is parameter of PDF). For example, if *X* is univariate and follows normal distribution, its PDF is:

If *X* is multivariate and follows multivariate normal distribution, its PDF is:

When *X* = (*x*1, *x*2,…, *xn*)*T* is multivariate, *μ* and Σ have following forms:

Of course, *μ* and Σ are determined by equation 1.3 and equation 1.4, respectively. However, theoretical means and variances of partial variables *xi* can be determined separately. For instance, each *μj* is theoretical mean of partial variable *xj* given marginal PDF .

Each *σij* where *i*≠*j* is theoretical covariance of partial variables *xi* and *xj* given joint PDF .

Each *σii* is theoretical variance of partial variable *xi* given marginal PDF .

Without loss of generality, by default, random variable *X* in this research is multivariate as vector if there is no additional explanation.

Because EM algorithm is essential an advanced version of maximum likelihood estimation (MLE) method, it is necessary to mentioned MLE in short. Suppose random variable *X* conforms to a distribution specified by a PDF denoted *f*(*X* | Θ) with parameter Θ. For example, if *X* is vector and follows normal distribution then,

Where *μ* and Σ are theoretical mean vector and covariance matrix, respectively with note that Θ = (*μ*, Σ)*T*. The notation |.| denotes determinant of given matrix and the notation Σ–1 denotes inverse of matrix Σ. Note, Σ is invertible and symmetric. Parameter of normal distribution is theoretical mean and variance,

But parameters of different distributions may be different from such mean and variance. Anyhow theoretical mean and variance are always determined based on parameter Θ.

For example, suppose *X* = (*x*1, *x*2,…, *xn*)*T* follows multinomial distribution of *K* trials, its PDF is:

Where *xj* are integers and Θ = (*p*1, *p*2,…, *pn*)*T* is the set of probabilities such that

Obviously, the parameter Θ = (*p*1, *p*2,…, *pn*)*T* does not include theoretical mean *E*(*X*) and variance *V*(*X*) but *E*(*X*) and *V*(*X*) of multinomial distribution is determined based on Θ as follows:

Statistic is a function of random variable *X*. Sufficient statistic denoted *τ*(*X*) is the statistic that it has all and only information to estimate parameter Θ. For example, sufficient statistic of the normal PDF above is *τ*(*X*) = (*X*, *XXT*)*T*. In fact, mean *μ* and covariance matrix Σ is totally determined based on only *X* and *XXT* and there is no redundant information in *τ*(*X*) as follows:

Given a sample containing observations, purpose of point estimation is to estimate unknown parameter Θ based on such sample. The result of estimation process is the estimate as approximation of unknown Θ. Formula to calculate based on sample is called estimator of Θ. As usual, estimator of Θ is determined based on sufficient statistics which in turn are functions of observations where observations are considered as random variables. Estimation methods mentioned in this research are MLE, Maximum A Posteriori (MAP), and EM in which MAP and EM are variants of MLE.

According to viewpoint of Bayesian statistics, parameter Θ is also random variable. Equation 1.5 specifies Bayes’ rule in which *f*(Θ) is called prior PDF (prior distribution) of Θ whereas *f*(Θ|*X*) is called posterior PDF (posterior distribution) of Θ given observation *X*.

|  |  |
| --- | --- |
|  | (1.5) |

In Bayes’ rule, the PDF *f*(*X* | Θ) is called likelihood function. If posterior distribution *f*(Θ|*X*) has the same form of prior distribution *f*(Θ), such posterior distribution and prior distribution are called conjugate distributions (conjugate probabilities) and *f*(Θ) is called conjugate prior (Wikipedia, Conjugate prior, 2018) for likelihood function *f*(*X*|Θ). For example, if prior distribution *f*(Θ) is beta distribution and likelihood function *P*(*X*|Θ) follows binomial distribution then, posterior distribution *f*(Θ|*X*) is beta distribution too and hence, *f*(Θ) and *f*(Θ|*X*) are conjugate distributions. Shortly, whether posterior distribution and prior distribution are conjugate distributions depends on prior distribution and likelihood function. In some research, Θ is also called hypothesis.

When *X* is evaluated as observation, let be estimate of Θ. It is calculated as a maximizer of the posterior PDF *f*(Θ|*X*) given *X*. Here data sample has only one observation *X* as = {*X*}.

Because the prior PDF *f*(Θ) is assumed to be fixed and the value is constant with regard to Θ, we have:

Obviously, MLE method determines as a maximizer of the likelihood function *f*(*X* | Θ) with regard to Θ when *X* is evaluated as observation. It is interesting that the likelihood function *f*(*X*|Θ) is the PDF of *X* with parameter Θ. For convenience, MLE maximizes the natural logarithm of the likelihood function denoted *l*(Θ) instead of maximizing the likelihood function.

|  |  |
| --- | --- |
|  | (1.6) |

Where *l*(Θ) = log(*f*(*X* | Θ)) is called log-likelihood function of Θ. Recall that equation 1.6 implies the optimization problem. Note, *l*(Θ) is function of Θ if *X* is evaluated as observation.

|  |  |
| --- | --- |
|  | (1.7) |

Equation 1.6 is the simple result of MLE for estimating parameter based on observed sample. The notation *l*(Θ|*X*) implies that *l*(Θ) is determined based on *X*. If the log-likelihood function *l*(Θ) is first-order smooth function then, from equation 1.6, the estimate can be solution of the equation created by setting the first-order of *l*(Θ) regarding Θ to be zero. Otherwise, some popular methods to solve optimization problem are Newton-Raphson (Burden & Faires, 2011, pp. 67-71), gradient descent (Ta, 2014), and Lagrange duality (Wikipedia, Karush–Kuhn–Tucker conditions, 2014).

For example, suppose *X* = (*x*1, *x*2,…, *xn*)*T* is vector and follows multivariate normal distribution,

Then the log-likelihood function is

Where *μ* and Σ are mean vector and covariance matrix of *f*(*X* | Θ), respectively with note that Θ = (*μ*, Σ)*T*. The notation |.| denotes determinant of given matrix and the notation Σ–1 denotes inverse of matrix Σ. Note, Σ is invertible and symmetric. Because normal PDF is smooth enough function, from equation 1.6, the estimate is solution of the equation created by setting the first-order of *l*(Θ) regarding *μ* and Σ to be zero. The first-order partial derivative of *l*(Θ) with respect to *μ* is (Nguyen, 2015, p. 35):

Setting this partial derivative to be zero, we obtain:

The first-order partial derivative of *l*(Θ) with respect to Σ is:

Due to:

And

Because Bilmes (Bilmes, 1998, p. 5) mentioned:

Where tr(*A*) is trace operator which takes sum of diagonal elements of square matrix, . This implies (Nguyen, 2015, p. 45):

Where Σ is symmetric and invertible matrix. Substituting the estimate into the first-order partial derivative of *l*(Θ) with respect to Σ, we have:

The estimate is the solution of equation formed by setting the first-order partial derivative of *l*(Θ) regarding Σ to zero matrix. Let [**0**] denote zero matrix.

We have:

Finally, MLE results out the estimate for normal distribution given observation *X* as follows:

When then , which implies that the estimate of covariance matrix is arbitrary with constraint that it is symmetric and invertible. This is reasonable because the sample is too small with only one observation *X*. When *X* is replaced by a sample = {*X*1, *X*2,…, *XN*}, it is easy to draw the following result by the similar way with equation 1.10.

In practice, if *X* is observed as particular *N* observations *X*1, *X*2,…, *XN*. Let = {*X*1, *X*2,…, *XN*} be the observed sample of size *N* with note that all *Xi* (s) are mutually independent and identically distributed (iid). Therefore, the likelihood function becomes product of partial likelihood functions as follows:

|  |  |
| --- | --- |
|  | (1.8) |

The log-likelihood function of Θ is:

|  |  |
| --- | --- |
|  | (1.9) |

The notation *l*(Θ|) implies that *l*(Θ) is determined based on . Equation is exact because all *Xi* (s) are iid. We have:

|  |  |
| --- | --- |
|  | (1.10) |

Equation 1.10 is the main result of MLE for estimating parameter based on observed sample. If the log-likelihood function *l*(Θ) is first-order smooth function then, from equation 1.10, the estimate can be solution of the equation created by setting the first-order of *l*(Θ) regarding Θ to be zero. Otherwise, some popular methods to solve optimization problem are Newton-Raphson (Burden & Faires, 2011, pp. 67-71), gradient descent (Ta, 2014), and Lagrange duality (Wikipedia, Karush–Kuhn–Tucker conditions, 2014).

For example, suppose each *Xi* = (*xi*1, *xi*2,…, *xin*)*T* is vector and follows multinomial distribution of *K* trials,

Where *xik* are integers and Θ = (*p*1, *p*2,…, *pn*)*T* is the set of probabilities such that

Given sample = {*X*1, *X*2,…, *XN*} with note that all *Xi* (s) are iid, according to equation 1.9, the log-likelihood function is

Because there is the constraint , we use Lagrange duality method to maximize *l*(Θ). The Lagrange function *la*(Θ, *λ*) is sum of *l*(Θ) and the constraint .

Note, *λ* is called Lagrange multiplier. Of course, *la*(Θ, *λ*) is function of Θ and *λ*. Because multinomial PDF is smooth enough function, the estimate is solution of the equation created by setting the first-order of *la*(Θ) regarding *pj* and *λ* to be zero. The first-order partial derivative of *la*(Θ) with respect to *pj* is:

Setting this partial derivative to be zero, we obtain following equation:

Summing this equation over *n* variables *pj*, we obtain:

Due to

We have

Substitute *λ* = *nN* into equation

We get the estimate as follows:

Quality of estimation is measured by mean and variance of the estimate . The mean of is:

|  |  |
| --- | --- |
|  | (1.11) |

The notation implies the formulation to calculate , which is resulted from MLE, MAP, or EM. Hence, is considered as function of *X* in the integral . The is unbiased estimate if . Otherwise, if then, is biased estimate. As usual, unbiased estimate is better than biased estimate. The condition is the criterion to check if an estimate is unbiased, which is applied for all estimation methods.

For example, given multivariate normal distribution and given sample = {*X*1, *X*2,…, *XN*} where all *Xi* (s) are iid, the estimate from MLE is:

Due to:

Then is unbiased estimate. We also have:

(Due to )

(Due to )

(Let *X* be random variable representing all iid *Xi* (s))

(Due to and the variance )

It is necessary to calculate the variance . In fact, we have:

Therefore, we have:

Hence, we conclude that is biased estimate because of ■

The variance of is:

|  |  |
| --- | --- |
|  | (1.12) |

The smaller the variance , the better the is.

Without loss of generality, suppose parameter Θ is vector, the second-order derivative of the log-likelihood function *l*(Θ) is called likelihood Hessian matrix (Zivot, 2009, p. 7) denoted *S*(Θ).

|  |  |
| --- | --- |
|  | (1.13) |

Suppose Θ = (*θ*1, *θ*2,…, *θr*)*T* where there are *r* partial parameter *θk*, equation 1.13 is expended as follows:

The notation *l*(Θ|*X*) implies that *l*(Θ) is determined based on *X*, according to equation 1.7. The notation *S*(Θ|*X*) implies *S*(Θ) is calculated based on *X*. If sample replaces *X* then,

|  |  |
| --- | --- |
|  | (1.14) |

Where = {*X*1, *X*2,…, *XN*} be the observed sample of size *N* with note that all *Xi* (s) are iid. The notation *l*(Θ|) implies that *l*(Θ) is determined based on , according to equation 1.9. The notation *S*(Θ|) implies *S*(Θ) is calculated based on .

The negative expectation of likelihood Hessian matrix is called information matrix or Fisher information matrix denoted *I*(Θ).

|  |  |
| --- | --- |
|  | (1.15) |

If *S*(Θ) is calculated by equation 1.13 with observation *X* then, *I*(Θ) becomes:

|  |  |
| --- | --- |
|  | (1.16) |

The notation *l*(Θ|*X*) implies that *l*(Θ) is determined based on *X*, according to equation 1.7. The notation *I*(Θ|*X*) implies *I*(Θ) is calculated based on *X*. Note, is considered as function of *X* in the integral .

If *S*(Θ) is calculated by equation 1.14 with observation sample = {*X*1, *X*2,…, *XN*} with note that all *Xi* (s) are iid then, *I*(Θ) becomes:

|  |  |
| --- | --- |
|  | (1.17) |

Where *X* is random variable representing every *Xi*. The notation *I*(Θ|) implies *I*(Θ) is calculated based on . Note, is considered as function of *X* in the integral . Following is proof of equation 1.17.

(The notation *l*(Θ|) implies that *l*(Θ) is determined based on )

(Due to equation 1.7 and iid *Xi* (s))

(Let *X* be random variable representing every *Xi*)

For MLE method, the inverse of estimator information matrix is called Cramer-Rao lower bound denoted .

|  |  |
| --- | --- |
|  | (1.18) |

Where *I*(Θ) is calculated by equation 1.16 or equation 1.17. Any covariance matrix of a MLE estimate has such Cramer-Rao lower bound. Such Cramer-Rao lower bound becomes if and only if is unbiased, (Zivot, 2009, p. 11):

|  |  |
| --- | --- |
|  | (1.19) |

Note, equation 1.18 and equation 1.19 are only valid for MLE method. The sign “≥” implies lower bound. In other words, Cramer-Rao lower bound is variance of the optimal MLE estimate. Moreover, beside the criterion , equation 1.19 can be used as another criterion to check if an estimate is unbiased. However, the criterion is applied for all estimation methods whereas equation 1.19 is only applied for MLE. Suppose Θ = (*θ*1, *θ*2,…, *θr*)*T* where there are *r* partial parameter *θk*, so the estimate is . Each element on diagonal of the Cramer-Rao lower bound is lower bound of variance of , denoted . Let be lower bound of , of course we have:

|  |  |
| --- | --- |
|  | (1.20) |

The sign “≥” implies lower bound. Derived from equation 1.17 and equation 1.18, is specified by equation 1.21.

|  |  |
| --- | --- |
|  | (1.21) |

Where *N* is size of sample = {*X*1, *X*2,…, *XN*} with note that all *Xi* (s) are iid. If there is only one observation *X* then, *N* = 1.

For example, let = {*X*1, *X*2,…, *XN*} be the observed sample of size *N* with note that all *Xi* (s) are iid, given multivariate normal PDF as follows:

Where *n* is dimension of vector *X* and Θ = (*μ*, Σ)*T* with note that *μ* is mean vector and Σ is covariance matrix. From previous example, the MLE estimate given is:

Mean and variance of from previous example are:

We knew that is unbiased estimate with criterion . Now we check again if is unbiased estimate with equation 1.20 as another criterion for MLE. Hence, we firstly calculate the lower bound and then compare it with the variance . In fact, according to equation 1.7, the log-likelihood function is:

The partial first-order derivative of *l*(Θ|*X*) with regard to *μ* is:

The partial second-order derivative of *l*(Θ|*X*) with regard to *μ* is:

According to equation 1.21, the lower bound is:

Due to , is unbiased estimate according to the criterion specified by equation 1.20 ■

MLE ignores the prior PDF *f*(Θ) because *f*(Θ) is assumed to be fixed but Maximum A Posteriori (MAP) method (Wikipedia, Maximum a posteriori estimation, 2017) concerns *f*(Θ) in maximization task as follows:

Where the value is constant with regard to Θ. This implies:

Therefore, the estimate is determined according to MAP as follows:

|  |  |
| --- | --- |
|  | (1.22) |

The log-likelihood function *l*(Θ) is specified by equation 1.7 with observation *X* or equation 1.9 with sample . Good information provided by the prior *f*(Θ) can improve quality of estimation. Essentially, MAP is an improved variant of MLE. Later on, we also recognize that EM algorithm is also a variant of MLE. All of them aim to maximize log-likelihood functions.

Now we skim through an introduction of EM algorithm. Suppose there are two spaces ***X*** and ***Y***, in which ***X*** is *hidden space* (missing space) whereas ***Y*** is *observed space*. We do not know ***X*** but there is a mapping from ***X*** to ***Y*** so that we can survey ***X***by observing ***Y***. The mapping is many-one function *φ*: ***X*** → ***Y*** and we denote *φ*–1(*Y*) = {: *φ*(*X*) = *Y*} as all such that *φ*(*X*) = *Y*. We also denote ***X***(*Y*) = *φ*–1(*Y*). Let *f*(*X* | Θ) be the PDF of random variable and let *g*(*Y* | Θ) be the PDF of random variable . Note, *Y* is also called observation. Equation 1.23 specifies *g*(*Y* | Θ) as integral of *f*(*X* | Θ) over *φ*–1(*Y*).

|  |  |
| --- | --- |
|  | (1.23) |

Where Θ is probabilistic parameter represented as a column vector, Θ= (*θ*1, *θ*2,…, *θr*)*T* in which each *θi* is a particular parameter. As a convention, let Ω be the domain of Θ such that and the dimension of Ω is *r*. For example, normal distribution has two particular parameters such as mean *μ* and variance *σ*2 and so we have Θ= (*μ*, *σ*2)*T*. Note that, Θ can degrades into a scalar as Θ = *θ*. The conditional PDF of *X* given *Y*, denoted *k*(*X* | *Y*, Θ), is specified by equation 1.24.

|  |  |
| --- | --- |
|  | (1.24) |

According to DLR (Dempster, Laird, & Rubin, 1977, p. 1), ***X*** is called *complete data* and the term “incomplete data” implies existence of ***X*** and ***Y*** where ***X*** is not observed directly and ***X*** is only known by the many-one mapping *φ*: ***X*** → ***Y***. In general, we only know ***Y***, *f*(*X* | Θ), and *k*(*X* | *Y*, Θ) and so our purpose is to estimate Θ based on such ***Y***, *f*(*X* | Θ), and *k*(*X* | *Y*, Θ). Like MLE approach, EM algorithm also maximizes the likelihood function to estimate Θ but the likelihood function in EM concerns ***Y*** and there are also some different aspects in EM which will be described later. Pioneers in EM algorithm firstly assumed that *f*(*X* | Θ) belongs to exponential family with note that many popular distributions such as normal, multinomial, and Poisson belong to exponential family (please see table 1.2). Although DLR (Dempster, Laird, & Rubin, 1977) proposed a generality of EM algorithm in which *f*(*X* | Θ) distributes arbitrarily, we should concern exponential family a little bit. Exponential family (Wikipedia, Exponential family, 2016) refers to a set of probabilistic distributions whose density functions have the same exponential form according to equation 1.25 (Dempster, Laird, & Rubin, 1977, p. 3):

|  |  |
| --- | --- |
|  | (1.25) |

Where *b*(*X*) is a function of *X*, which is called base measure and *τ*(*X*) is a vector function of *X*, which is sufficient statistic. For example, the sufficient statistic of normal distribution is *τ*(*X*) = (*X*, *X*2)*T*. Equation 1.25 expresses the canonical form of exponential family. Recall that Ω is the domain of Θ such that . Suppose that Ω is a convex set. If Θ is restricted only to Ω then, *f*(*X* | Θ) specifies a *regular exponential family*. If Θ lies in a curved sub-manifold Ω0 of Ω then, *f*(*X* | Θ) specifies a *curved exponential family*. The *a*(Θ) is *partition function* for variable *X*, which is used for normalization.

The first-order derivative of log(*a*(Θ)) is expectation of *τ*(*X*).

The second-order derivative of log(*a*(Θ)) is (Jebara, 2015):

Where,

Hence (Hardle & Simar, 2013, pp. 125-126),

Where *V*(*τ*(*X*) | Θ) is central covariance matrix of *τ*(*X*). Please read the book “Matrix Analysis and Calculus” by Nguyen (Nguyen, 2015) for comprehending derivative of vector and matrix. Let *a*(Θ | *Y*) be a so-called *observed partition function* for observation *Y*.

Similarly, we obtain that the first-order derivative of log(*a*(***θ*** | *Y*)) is expectation of *τ*(*X*) based on *Y*.

If *f*(*X* | Θ) follows exponential family, the conditional density *k*(*X* | *Y*, Θ) is determined as follows:

If *f*(*X* | Θ) follows exponential family then, *k*(*X* | *Y*, Θ) also follows exponential family. In fact, we have:

Note that *k*(*X* | *Y*, Θ) is determined on . Of course, we have:

The first-order derivative of log(*a*(Θ | *Y*)) is:

The second-order derivative of log(*a*(Θ) | *Y*) is:

Where *V*(*τ*(*X*) | *Y*, Θ) is central covariance matrix of *τ*(*X*) given observed *Y*. Table 1.1 is summary of *f*(*X* | Θ), *g*(*Y* | Θ), *k*(*X* | *Y*, Θ), *a*(Θ), log’(*a*(Θ)), *a*(Θ | *Y*), and log’(*a*(Θ | *Y*)) with exponential family.

|  |
| --- |
|  |

**Table 1.1.** Summary of *f*(*X* | Θ), *g*(*Y* | Θ), *k*(*X* | *Y*, Θ), *a*(Θ), log’(*a*(Θ)), *a*(Θ | *Y*), and log’(*a*(Θ | *Y*)) with exponential family

Table 1.2 shows some popular distributions belonging to exponential family along with their canonical forms (Wikipedia, Exponential family, 2016). In case of multivariate distributions, dimension of random variable *X* = (*x*1, *x*2,…, *xn*)*T* is *n*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Distribution | PDF | Parameter  Θ | *τ*(*X*) | *b*(*X*) | *a*(Θ) |
| Multivariate  normal |  |  |  |  |  |
| Multinomial | Where, , , and . |  |  |  | 1 |

**Table 1.2.** Some popular distributions belonging to exponential family

For example, the multivariate normal distribution with theoretical mean *μ* and covariance matrix Σ of random variable *X* = (*x*1, *x*2,…, *xn*)*T* has following PDF:

Note, Σ is invertible and symmetric. Its canonical form in exponential family is:

Where,

We have:

(Due to |Σ–1| = |Σ|–1)

Therefore,

(Due to Σ is symmetric, *μT*Σ–1*X* = *XT*Σ–1*μ*)

Simply, EM algorithm is iterative process including many iterations, in which each iteration has expectation step (E-step) and maximization step (M-step). E-step aims to estimate sufficient statistic given current parameter and observed data *Y* whereas M-step aims to re-estimate the parameter based on such sufficient statistic by maximizing likelihood function related to *X*. EM algorithm is described in the next section in detail. As an introduction, DLR gave an example for illustrating EM algorithm (Dempster, Laird, & Rubin, 1977, pp. 2-3).

**Example 1.1.** Rao (Rao, 1955) presents observed data *Y* of 197 animals following multinomial distribution with four categories, such as *Y* = (*y*1, *y*2, *y*3, *y*4) = (125, 18, 20, 34). The PDF of *Y* is:

Note, probabilities *py*1, *py*2, *py*3, and *py*4 in *g*(*Y* | *θ*) are 1/2 + *θ*/4, 1/4 – *θ*/4, 1/4 – *θ*/4, and *θ*/4, respectively as parameters. The expectation of any sufficient statistic *yi* with regard to *g*(*Y* | *θ*) is:

Observed data *Y* is associated with hidden data *X* following multinomial distribution with five categories, such as *X* = {*x*1, *x*2, *x*3, *x*4, *x*5} where *y*1 = *x*1 + *x*2, *y*2 = *x*3, *y*3 = *x*4, *y*4 = *x*5. The PDF of *X* is:

Note, probabilities *px*1, *px*2, *px*3, *px*4, and *px*5 in *f*(*X* | *θ*) are 1/2, *θ*/4, 1/4 – *θ*/4, 1/4 – *θ*/4, and *θ*/4, respectively as parameters. The expectation of any sufficient statistic *xi* with regard to *f*(*X* | *θ*) is:

Due to *y*1 = *x*1 + *x*2, *y*2 = *x*3, *y*3 = *x*4, *y*4 = *x*5, the mapping function *φ* between *X* and *Y* is *y*1 = *φ*(*x*1, *x*2) = *x*1 + *x*2. Therefore *g*(*Y* | *θ*) is sum of *f*(*X* | *θ*) over *x*1 and *x*2 such that *x*1 + *x*2 = *y*1 according to equation 1.23. In other words, *g*(*Y* | *θ*) is resulted from summing *f*(*X* | *θ*) over all (*x*1, *x*2) pairs such as (0, 125), (1, 124),…, (125, 0) because of *y*1 = 125 from observed *Y*.

Rao (Rao, 1955) applied EM algorithm into determining the optimal estimate *θ\**. Note *y*2 = *x*3, *y*3 = *x*4, *y*4 = *x*5 are known and so only sufficient statistics *x*1 and *x*2 are not known. Given the *t*th iteration, sufficient statistics *x*1 and *x*2 are estimated as *x*1(*t*) and *x*2(*t*) based on current parameter *θ*(*t*) and *g*(*Y* | *θ*) in E-step below:

Given *py*1 = 1/2 + *θ*/4, which implies that:

Because the probability of *y*1 is 1/2 + *θ*/4 and *y*1 is sum of *x*1 and *x*2, probabilities of *x*1 and *x*2 are based on 1/2 + *θ*/4 such that

We can select *a*1 = 1/2 and *a*2 = *θ*/4, which implies:

When *y*1 is evaluated as *y*1 = 125, we obtain:

Please pay attention that the expectation *y*1(*t*) = *E*(*y*1 | *Y*, *θ*(*t*)) gets value 125 when *y*1 is evaluated as *y*1 = 125 and the probability corresponding to *y*1 gets maximal as 1/2 + *θ*(*t*)/4 = 1.

According to M-step, the next estimate *θ*(*t*+1) is a maximizer of the log-likelihood function related to *X*. This log-likelihood function is:

The first-order derivative of log(*f*(*X* | *θ*) is:

Because *y*2 = *x*3 = 18, *y*3 = *x*4 = 20, *y*4 = *x*5 = 34 and *x*2 is approximated by *x*2(*t*), we have:

As a maximizer of log(*f*(*X* | *θ*), the next estimate *θ*(*t*+1) is solution of the following equation

So we have:

Where,

For example, given the initial *θ*(1) = 0.5, at the first iteration, we have:

After five iterations we gets the optimal estimate *θ\**:

Table 1.3 (Dempster, Laird, & Rubin, 1977, p. 3) lists estimates of *θ* over five iterations (*t* =1, 2, 3, 4, 5) with note that *θ*(1) is initialized arbitrarily and *θ*\* = *θ*(5) = *θ*(6) is determined at the 5th iteration. The third column gives deviation *θ\** and *θ*(*t*) whereas the fourth column gives the ratio of successive deviations. Later on, we will know that such ratio implies convergence rate.

|  |  |  |  |
| --- | --- | --- | --- |
| *t* | *θ*(*t*) | *θ\** – *θ*(*t*) | (*θ\** – *θ*(*t*+1)) /  (*θ\** – *θ*(*t*)) |
| 1 | *θ*(1) = 0.5 | 0.1268 | 0.1465 |
| *θ*(2) = 0.6082 | 0.0186 | 0.1346 |
| 2 | *θ*(2) = 0.6082 | 0.0186 | 0.1346 |
| *θ*(3) = 0.6243 | 0.0025 | 0.1330 |
| 3 | *θ*(3) = 0.6243 | 0.0025 | 0.1330 |
| *θ*(4) = 0.6265 | 0.0003 | 0.1328 |
| 4 | *θ*(4) = 0.6265 | 0.0003 | 0.1328 |
| *θ*(5) = 0.6268 | 0 | 0.1328 |
| 5 | *θ*(5) = 0.6268 | 0 | 0.1328 |
| *θ*(6) = 0.6268 | 0 | 0.1328 |

**Table 1.3.** EM algorithm in simple case

For example, at the first iteration, we have:

# **2. EM algorithm**

Expectation maximization (EM) algorithm has many iterations and each iteration has two steps in which expectation step (E-step) calculates sufficient statistic of hidden data based on observed data and current parameter whereas maximization step (M-step) re-estimates parameter. When DLR proposed EM algorithm (Dempster, Laird, & Rubin, 1977), they firstly concerned that the PDF *f*(*X* | Θ) of hidden space belongs to exponential family. E-step and M-step at the *t*th iteration are described in table 2.1 (Dempster, Laird, & Rubin, 1977, p. 4), in which the current estimate is Θ(*t*), with note that *f*(*X* | Θ) belongs to regular exponential family.

|  |
| --- |
| *E-step*:  We calculate current value *τ*(*t*) of the sufficient statistic *τ*(*X*) from observed *Y* and current parameter Θ(*t*) according to equation 2.6:  *M-step*:  Basing on *τ*(*t*), we determine the next parameter Θ(*t*+1) as solution of equation 2.3:  Note, Θ(*t*+1) will become current parameter at the next iteration ((*t*+1)th iteration). |

**Table 2.1.** E-step and M-step of EM algorithm given regular exponential PDF *f*(*X*|Θ)

EM algorithm stops if two successive estimates are equal, Θ*\** = Θ(*t*) = Θ(*t*+1), at some *t*th iteration. At that time we conclude that Θ*\** is the optimal estimate of EM process. Please see table 1.1 to know how to calculate *E*(*τ*(*X*) | Θ(*t*)) and *E*(*τ*(*X*) | *Y*, Θ(*t*)).

It is necessary to explain E-step and M-step as well as convergence of EM algorithm. Essentially, the two steps aim to maximize log-likelihood function of Θ, denoted *L*(Θ), with respect to observation *Y*.

Where,

Note that log(.) denotes logarithm function. Therefore, EM algorithm is an extension of maximum likelihood estimation (MLE) method. In fact, let *l*(Θ) be log-likelihood function of Θ with respect to *X*.

|  |  |
| --- | --- |
|  | (2.1) |

By referring to table 1.1, the first-order derivative of *l*(Θ) is:

|  |  |
| --- | --- |
|  | (2.2) |

We set the first-order derivative of *l*(Θ) to be zero with expectation that *l*(Θ) will be maximized. Therefore, the optimal estimate Θ*\** is solution of the following equation which is specified in M-step.

The expression *E*(*τ*(*X*) | Θ) is function of Θ but *τ*(*X*) is still dependent on *X*. Let *τ*(*t*) be value of *τ*(*X*) at the *t*th iteration of EM process, candidate for the best estimate of Θ is solution of equation 2.3 according to M-step.

|  |  |
| --- | --- |
|  | (2.3) |

Where,

Thus, we will calculate *τ*(*t*) by maximizing the log-likelihood function *L*(Θ) given *Y*. Recall that maximizing *L*(Θ) is the ultimate purpose of EM algorithm.

Where,

|  |  |
| --- | --- |
|  | (2.4) |

Due to:

It implies:

Because *f*(*X* | Θ) belongs to exponential family, we have:

The log-likelihood function *L*(Θ) is reduced as follows:

By referring to table 1.1, the first-order derivative of *L*(Θ) is:

|  |  |
| --- | --- |
|  | (2.5) |

We set the first-order derivative of *L*(Θ) to be zero with expectation that *L*(Θ) will be maximized, as follows:

It implies:

Let Θ(*t*) be the current estimate at some *t*th iteration of EM process. Derived from the equality above, the value *τ*(*t*) is calculated as seen in equation 2.6.

|  |  |
| --- | --- |
|  | (2.6) |

Where,

Equation 2.6 specifies the E-step of EM process. After *t* iterations we will obtain Θ*\** = Θ(*t*+1) = Θ(*t*) such that *E*(*τ*(*X*) | *Y*, Θ(*t*)) = *E*(*τ*(*X*) | *Y*, Θ*\**) = *τ*(*t*) = *E*(*τ*(*X*) | Θ*\**) = *E*(*τ*(*X*) | Θ(*t*+1)) when Θ(*t*+1) is solution of equation 2.3 (Dempster, Laird, & Rubin, 1977, p. 5). This means that Θ*\** is the optimal estimate of EM process because Θ*\** is solution of the equation:

Thus, we conclude that Θ*\** is the optimal estimate of EM process.

The EM algorithm shown in table 2.1 is totally exact with assumption that *f*(*X*|Θ) belongs to regular exponential family. If *f*(*X*|Θ) is not regular, the maximal point (maximizer) of the log-likelihood function *l*(Θ) is not always the stationary point Θ\* so that the first-order derivative of *l*(Θ) is zero, *l*’(Θ\*) = 0. However, if *f*(*X*|Θ) belongs to curved exponential family, the M-step of the EM algorithm shown in table 2.1 is modified as follows (Dempster, Laird, & Rubin, 1977, p. 5):

|  |  |
| --- | --- |
|  | (2.7) |

Where *τ*(*t*) is calculated by equation 2.6 in E-step. This means that, in more general manner, the maximizer Θ(*t*+1) will be found by some way. Recall that if Θ lies in a curved sub-manifold Ω0 of Ω where Ω is the domain of Θ then, *f*(*X* | Θ) belongs to curved exponential family.

In general, given exponential family, within simple EM algorithm, E-step aims to calculate the current sufficient statistic *τ*(*t*) that maximizes the log-likelihood function *L*(Θ) given *Y* at current Θ(*t*) whereas M-step aims to maximize the log-likelihood function *l*(Θ) given *τ*(*t*), as seem in table 2.2. Note, in table 2.2, *f*(*X*|Θ) belongs to curved exponential family but it is not necessary to be regular.

|  |
| --- |
| *E-step*:  We calculate current value *τ*(*t*) of the sufficient statistic *τ*(*X*) from observed *Y* according to equation 2.6.  Where,  The log-likelihood function *L*(Θ(*t*)) given *Y* at current Θ(*t*) gets maximal with such *τ*(*t*).Note, *L*(Θ) is specified by equation 2.4.  *M-step*:  Basing on *τ*(*t*), we determine the next parameter Θ(*t*+1) by maximizing the log-likelihood function *l*(Θ) given *τ*(*t*), where *l*(Θ) is specified by equation 2.1. Actually, the sufficient statistic *τ*(*t*) calculated in E-step is substituted for unobserved *X* in *l*(Θ) so that it is possible to maximize *l*(Θ) with subject to Θ. |

**Table 2.2.** E-step and M-step of EM algorithm given exponential PDF *f*(*X*|Θ)

EM algorithm stops if two successive estimates are equal, Θ*\** = Θ(*t*) = Θ(*t*+1), at some *t*th iteration. At that time, Θ*\** is the optimal estimate of EM process, which is an optimizer of *L*(Θ).

Going back example 1.1, given the *t*th iteration, sufficient statistics *x*1 and *x*2 are estimated as *x*1(*t*) and *x*2(*t*) based on current parameter *θ*(*t*) in E-step according to equation 2.6.

Given *py*1 = 1/2 + *θ*/4, which implies that:

This suggests us to select:

Essentially, equation 2.3 specifying M-step is result of maximizing the log-likelihood function *l*(Θ).

Hence, the next estimate *θ*(*t*+1) is a maximizer of such log-likelihood function *l*(Θ).

Table 1.3 (Dempster, Laird, & Rubin, 1977, p. 3) show resulted estimation ■

For further research, DLR gave a preeminent generality of EM algorithm (Dempster, Laird, & Rubin, 1977, pp. 6-11) in which *f*(*X* | Θ) specifies arbitrary distribution. In other words, there is no requirement of exponential family. They define the conditional expectation *Q*(Θ’ | Θ) according to equation 2.8 (Dempster, Laird, & Rubin, 1977, p. 6).

|  |  |
| --- | --- |
|  | (2.8) |

The two steps of generalized EM (*GEM*) algorithm aim to maximize *Q*(Θ | Θ(*t*)) at some *t*th iteration as seen in table 2.3 (Dempster, Laird, & Rubin, 1977, p. 6).

|  |
| --- |
| *E-step*:  The expectation *Q*(Θ | Θ(*t*)) is determined based on current parameter Θ(*t*), according to equation 2.8. Actually, *Q*(Θ | Θ(*t*)) is formulated as function of Θ.  *M-step*:  The next parameter Θ(*t*+1) is a maximizer of *Q*(Θ | Θ(*t*)) with subject to Θ. Note that Θ(*t*+1) will become current parameter at the next iteration (the (*t*+1)th iteration). |

**Table 2.3.** E-step and M-step of GEM algorithm

DLR proved that GEM algorithm converges at some *t*th iteration. At that time, Θ*\** = Θ(*t*+1) = Θ(*t*) is the optimal estimate of EM process, which is an optimizer of *L*(Θ).

It is deduced from E-step and M-step that *Q*(Θ | Θ(*t*)) is increased after every iteration. How to maximize *Q*(Θ | Θ(*t*)) is the optimization problem which is dependent on applications. For example, some popular methods to solve optimization problem are Newton-Raphson (Burden & Faires, 2011, pp. 67-71), gradient descent (Ta, 2014), and Lagrange duality (Wikipedia, Karush–Kuhn–Tucker conditions, 2014). GEM algorithm still aims to maximize the log-likelihood function *L*(Θ) specified by equation 2.4, which will be explained in next section. The next section focuses on convergence of GEM algorithm proved by DLR (Dempster, Laird, & Rubin, 1977, pp. 7-10) but firstly we should discuss some features of *Q*(Θ’ | Θ). In special case of exponential family, *Q*(Θ’ | Θ) is modified by equation 2.9.

|  |  |
| --- | --- |
|  | (2.9) |

Where,

Following is a proof of equation 2.9.

Because *k*(*X* | *Y*, Θ) belongs exponential family, the expectation *E*(*τ*(*X*) | *Y*, Θ) is function of Θ, denoted *τ*Θ. It implies:

If *f*(*X*|Θ) belongs to regular exponential family, *Q*(Θ’ | Θ) gets maximal at the stationary point Θ\* so that the first-order derivative of *Q*(Θ’ | Θ) is zero. By referring to table 1.1, the first-order derivative of *Q*(Θ’ | Θ) with regard to Θ’ is:

Let *τ*(*t*) be the value of *τ*Θ at the *t*th iteration. The next parameter Θ(*t*+1) is determined at M-step as solution of the following equation.

The equation above is indeed equation 2.3. If *f*(*X*|Θ) belongs to curved exponential family, Θ(*t*+1) is determined as follows:

The equation above is indeed equation 2.7. Therefore, GEM shown in table 2.3 degrades into EM shown in table 2.1 if *f*(*X*|Θ) belongs to exponential family. Of course, this recognition is trivial. Example 1.1 is also a good example for GEM when multinomial distribution belongs to exponential family and then we apply equation 2.7 into maximizing *Q*(Θ’ | Θ).

In practice, if *Y* is observed as particular *N* observations *Y*1, *Y*2,…, *YN*. Let = {*Y*1, *Y*2,…, *YN*} be the observed sample of size *N* with note that all *Yi* (s) are mutually independent and identically distributed (iid). Given an observation *Yi*, there is an associated random variable *Xi*. All *Xi* (s) are iid and they are not existent in fact. Each is a random variable like *X*. Of course, the domain of each *Xi* is ***X***. Let = {*X*1, *X*2,…, *XN*} be the set of associated random variables. Because all *Xi* (s) are iid, the joint PDF of is determined as follows:

Because all *Xi* (s) are iid and each *Yi* is associated with *Xi*, the conditional joint PDF of given is determined as follows:

The conditional expectation *Q*(Θ’ | Θ) given samples ***X*** and ***Y*** is determined as follows:

(Suppose *f*(*Xi* | Θ) and *k*(*Xj* | *Yj*, Θ) are analytic functions)

(Suppose *f*(*Xi* | Θ) and *k*(*Xj* | *Yj*, Θ) are analytic functions)

By taking Riemann integral on , we have:

As a result, equation 2.10 specifies the conditional expectation *Q*(Θ’ | Θ) given an observed sample = {*Y*1, *Y*2,…, *YN*} and a set of associated random variables = {*X*1, *X*2,…, *XN*}. Note, all *Xi* (s) are iid and they are not existent in fact.

Because all *Xi* are iid, let *X* be the random variable representing every *Xi* and the equation of *Q*(Θ’ | Θ) is re-written as follows:

|  |  |
| --- | --- |
|  | (2.10) |

The similar proof of equation 2.10 in case that *Xi* (s) are discrete is found in (Bilmes, 1998, p. 4). In case that *f*(*X* | Θ) and *k*(*X* | *Yi*, Θ) belong to exponential family, equation 2.10 becomes equation 2.11 with an observed sample = {*Y*1, *Y*2,…, *YN*}.

|  |  |
| --- | --- |
|  | (2.11) |

Where,

Please combine equation 2.9 and equation 2.10 to comprehend how to derive equation 2.11. Note, is dependent on both Θ and *Yi*.

DLR (Dempster, Laird, & Rubin, 1977, p. 1) called ***X*** as *complete data* because the mapping *φ*: ***X*** → ***Y*** is many-one function. There is another case that the complete space ***Z*** consists of hidden space ***X*** and observed space ***Y*** with note that ***X*** and ***Y*** are separated. There is no explicit mapping *φ* from ***X*** and ***Y*** but there exists a PDF of as the joint PDF of and .

In this case, the equation 2.8 is modified with the joint PDF *f*(*X*, *Y* | Θ). The PDF of *Y* becomes:

The PDF *f*(*Y*|Θ) is equivalent to the PDF *g*(*Y*|Θ) mentioned in equation 1.23. Although there is no explicit mapping from ***X*** to ***Y***, the PDF of *Y* above implies an implicit mapping from ***Z*** to ***Y***. The conditional PDF of *X* given *Z* is specified according to Bayes’ rule as follows:

The conditional PDF *f*(*X*|*Y*, Θ) is equivalent to the conditional PDF *k*(*X*|*Y*, Θ) mentioned in equation 1.24. Of course, given *Y*, we always have:

Equation 2.12 specifies the conditional expectation *Q*(Θ’ | Θ) in case that there is no explicit mapping from ***X*** to ***Y*** but there exists the joint PDF of *X* and *Y*.

|  |  |
| --- | --- |
|  | (2.12) |

Where,

Note, ***X*** is separated from ***Y*** and the complete data ***Z*** = (***X***, ***Y***) is composed of ***X*** and ***Y***. For equation 2.12, the existence of the joint PDF *f*(*X*, *Y* | Θ) can be replaced by the existence of the conditional PDF *f*(*Y*|*X*, Θ) and the prior PDF *f*(*X*|Θ) due to:

In applied statistics, equation 2.8 is often replaced by equation 2.12 because specifying the joint PDF *f*(*X*, *Y* | Θ) is more practical than specifying the mapping *φ*: ***X*** → ***Y***. However, equation 2.8 is more general equation 2.12 because the requirement of the joint PDF for equation 2.12 is stricter than the requirement of the explicit mapping for equation 2.8. In case that *X* and *Y* are discrete, equation 2.12 becomes:

In case that *X* and *Y* are discrete, *P*(*X*, *Y* | Θ) is the joint probability of *X* and *Y* whereas *P*(*X* | *Y*, Θ) is the conditional probability of *X* given *Y*. Mixture model mentioned in subsection 5.1 is a good example for GEM without explicit mapping from ***X*** to ***Y***.

Equation 2.12 can be proved alternately without knowledge related to complete data. In fact, given hidden space ***X***, observed space ***Y***, and a joint PDF *f*(*X*, *Y* | Θ), the likelihood function *L*(Θ’) is re-defined here as log(*f*(*Y* | Θ’)). The maximizer is:

Suppose the current parameter is Θ after some iteration. Next we must find out the new estimate that maximizes the next log-likelihood function *L*(Θ’). In other words, it maximizes the deviation between current log-likelihood *L*(Θ) and next log-likelihood *L*(Θ’) with regard to Θ’.

Suppose the total probability of observed data can be determined by marginalizing over hidden data:

The expansion of *f*(*Y* | Θ’)is total probability rule. The deviation *L*(Θ’) –*L*(Θ) is re-written:

Because hidden *X* is the complete set of mutually exclusive variables, the sum of conditional probabilities of *X* is equal to 1 given *Y* and Θ.

Where,

Applying Jensen’s inequality (Sean, 2009, pp. 3-4)

into the deviation *L*(Θ’) –*L*(Θ), Sean (Sean, 2009, p. 6) proved that:

Where,

Because *C* is constant with regard to Θ’, it is possible to eliminate *C* in order to simplify the optimization criterion as follows:

Let

We have the proof ■

In practice, suppose *Y* is observed as a sample = {*Y*1, *Y*2,…, *YN*} of size *N* with note that all *Yi* (s) are mutually independent and identically distributed (iid). The observed sample is associated with a a hidden set (latent set) = {*X*1, *X*2,…, *XN*} of size *N*. All *Xi* (s) are iid and they are not existent in fact. Let be the random variable representing every *Xi*. Of course, the domain of *X* is ***X***. Equation 2.13 specifies the conditional expectation *Q*(Θ’ | Θ) given such .

|  |  |
| --- | --- |
|  | (2.13) |

Equation 2.13 is a variant of equation 2.10 in case that there is no explicit mapping between *Xi* and *Yi* but there exists the same joint PDF between *Xi* and *Yi*. Please see the proof of equation 2.10 to comprehend how to derive equation 2.13.

If *X* is discrete such that *f*(*X*, *Y* | Θ) = *P*(*X*|Θ)*f*(*Y* | *X*, Θ) then, according to the total probability rule, we have:

Note, when *X* is discrete, its PDF *f*(*X*|Θ) becomes the probability *P*(*X*|Θ). Therefore, equation 2.14 is the discrete variant of equation 2.13, as follows:

|  |  |
| --- | --- |
|  | (2.14) |

Where *P*(*X* | *Yi*, Θ) is determined by Bayes’ rule, as follows:

Equation 2.14 is the base for estimating the probabilistic mixture model by EM algorithm, which will be described later in detail.

GEM is now described in general. Here it is necessary to explain GEM by the viewpoint of lower bound (Sean, 2009, pp. 7-8). The main purpose of GEM algorithm is to maximize the log-likelihood *L*(Θ) = log(*g*(*Y* | Θ)) with observed data *Y*. However, it is too difficult to maximize log(*g*(*Y* | Θ)) because *g*(*Y* | Θ) is not well-defined when *g*(*Y* | Θ) is integral of *f*(*X* | Θ) given a general mapping function. DLR solved this problem by an iterative process which is an instance of GEM algorithm. The lower-bound (Sean, 2009, pp. 7-8) of *L*(Θ) is maximized over many iterations of the iterative process so that *L*(Θ) is maximized finally. Such lower-bound is determined indirectly by the condition expectation *Q*(Θ | Θ(*t*)) so that maximizing *Q*(Θ | Θ(*t*))is the same to maximizing the lower bound. Suppose Θ(*t*+1) is a maximizer of *Q*(Θ | Θ(*t*)) at *t*th iteration, which is also a maximizer of the lower bound at *t*th iteration. The lower bound is increased after every iteration. As a result, the maximizer Θ*\** of the final lower-bound after many iterations will be expected as a maximizer of *L*(Θ) in final.

For more explanations, let *lb*(Θ | Θ(*t*)) be lower bound of *L*(Θ) at the *t*th iteration (Sean, 2009, p. 7). From equation 3.2, we have:

*lb*(Θ | Θ(*t*)) = *Q*(Θ | Θ(*t*)) – *H*(Θ(*t*) | Θ(*t*))

Please see equation 3.1 for definition of *H*(Θ’ | Θ). Due to equation 3.2 and equation 3.3

*L*(Θ) = *Q*(Θ | Θ(*t*)) – *H*(Θ | Θ(*t*))

*H*(Θ | Θ(*t*)) ≤ *H*(Θ(*t*) | Θ(*t*))

We have:

*lb*(Θ | Θ(*t*)) ≤ *L*(Θ)

The lower bound *lb*(Θ | Θ(*t*)) has following property (Sean, 2009, p. 7):

*lb*(Θ(*t*) | Θ(*t*)) = *Q*(Θ(*t*) | Θ(*t*)) – *H*(Θ(*t*) | Θ(*t*)) = *L*(Θ(*t*))

Therefore, the two steps of GEM is interpreted with regard to the lower bound *lb*(Θ | Θ(*t*)) as seen in table 2.4.

|  |
| --- |
| *E-step*:  The lower bound *lb*(Θ | Θ(*t*)) is re-calculated based on *Q*(Θ | Θ(*t*)).  *M-step*:  The next parameter Θ(*t*+1) is a maximizer of *Q*(Θ | Θ(*t*)) which is also a maximizer of *lb*(Θ | Θ(*t*)) because *H*(Θ(*t*) | Θ(*t*)) is constant. Note that Θ(*t*+1) will become current parameter at the next iteration so that the lower bound is increased in the next iteration. |

**Table 2.4.** An interpretation of GEM with lower bound

Because *Q*(Θ | Θ(*t*)) is defined fixedly in E-step, most variants of EM algorithm focus on how to maximize *Q*(Θ’ | Θ) in M-step more effectively so that EM is faster or more accurate. Figure 2.1 (Borman, 2004, p. 7) shows relationship between the log-likelihood function and its lower-bound .

A screenshot of a cell phone

Description automatically generated

**Figure 2.1.** Relationship between the log-likelihood function and its lower-bound

Convergence of GEM will be mentioned in next section.

# **3. Convergence of EM algorithm**

Recall that DLR proposed GEM algorithm which aims to maximize the log-likelihood function *L*(Θ) by maximizing *Q*(Θ’ | Θ) over many iterations. This section focuses on mathematical explanation of the convergence of GEM algorithm given by DLR (Dempster, Laird, & Rubin, 1977, pp. 6-9). Recall that we have:

Let *H*(Θ’ | Θ) be another conditional expectation which has strong relationship with *Q*(Θ’ | Θ) (Dempster, Laird, & Rubin, 1977, p. 6).

|  |  |
| --- | --- |
|  | (3.1) |

From equation 2.8 and equation 3.1, we have:

|  |  |
| --- | --- |
|  | (3.2) |

Following is a proof of equation 3.2.

**Lemma3.1** (Dempster, Laird, & Rubin, 1977, p. 6). For any pair (Θ’, Θ) in Ω x Ω,

|  |  |
| --- | --- |
|  | (3.3) |

The equality occurs if and only if *k*(*X* | *Y*, Θ’) = *k*(*X* | *Y*, Θ) almost everywhere ■

Following is a proof of lemma 3.1 as well as equation 3.3. The log-likelihood function *L*(Θ’) is re-written as follows:

Due to

By applying Jensen’s inequality (Sean, 2009, pp. 3-4) with concavity of logarithm function,

Sean (Sean, 2009, p. 6) proved that:

It implies:

According to Jensen’s inequality (Sean, 2009, pp. 3-4), the equality occurs if and only if *k*(*X* | *Y*, Θ’) is linear or *f*(*X* | Θ’) is constant. In other words, the equality occurs if and only if *k*(*X* | *Y*, Θ’) = *k*(*X* | *Y*, Θ) almost everywhere when *f*(*X* | Θ) is not constant and *k*(*X* | *Y*, Θ’) is a PDF.

Let be a sequence of estimates of Θ resulted from iterations of EM algorithm. Let Θ → *M*(Θ) be the mapping such that each estimation Θ(*t*) → Θ(*t*+1) at any given iteration is defined by equation 3.4 (Dempster, Laird, & Rubin, 1977, p. 7).

|  |  |
| --- | --- |
|  | (3.4) |

**Definition 3.1** (Dempster, Laird, & Rubin, 1977, p. 7). An iterative algorithm with mapping *M*(Θ) is a GEM algorithm if

|  |  |
| --- | --- |
|  | (3.5) |

Of course, specification of GEM shown in table 2.3 satisfies the definition 3.1 because Θ(*t*+1) is a maximizer of *Q*(Θ | Θ(*t*)) with regard to variable Θ in M-step.

**Theorem 3.1** (Dempster, Laird, & Rubin, 1977, p. 7). For every GEM algorithm

|  |  |
| --- | --- |
|  | (3.6) |

Where equality occurs if and only if *Q*(*M*(Θ) | Θ) = *Q*(Θ | Θ) and *k*(*X* | *Y*, *M*(Θ)) = *k*(*X* | *Y*, Θ) almost everywhere ■

Following is the proof of theorem 3.1 (Dempster, Laird, & Rubin, 1977, p. 7):

Because the equality of lemma 3.1 occurs if and only if *k*(*X* | *Y*, Θ’) = *k*(*X* | *Y*, Θ) almost everywhere and the equality of the definition 3.1 is *Q*(*M*(Θ) | Θ) = *Q*(Θ | Θ), we deduce that the equality of theorem 3.1 occurs if and only if *Q*(*M*(Θ) | Θ) = *Q*(Θ | Θ) and *k*(*X* | *Y*, *M*(Θ)) = *k*(*X* | *Y*, Θ) almost everywhere. It is easy to draw corollary 3.1 and corollary 3.2 from definition 3.1 and theorem 3.1.

**Corollary 3.1** (Dempster, Laird, & Rubin, 1977). Suppose for some , *L*(Θ*\**) ≥ *L*(Θ) for all then for every GEM algorithm:

1. *L*(*M*(Θ*\**)) = *L*(Θ*\**)
2. *Q*(*M*(Θ*\**) | Θ*\**) = *Q*(Θ*\** | Θ*\**)
3. *k*(*X* | *Y*, *M*(Θ*\**)) = *k*(*X* | *Y*, Θ*\**) ■

*Proof*. From theorem 3.1 and the assumption of corollary 3.1, we have:

This implies:

As a result,

From theorem 3.1, we also have:

**Corollary 3.2** (Dempster, Laird, & Rubin, 1977). If for some , *L*(Θ*\**) > *L*(Θ) for all such that Θ ≠ Θ\*, then for every GEM algorithm:

*M*(Θ*\**) = Θ*\** ■

*Proof*. From corollary 3.1 and the assumption of corollary 3.2, we have:

If *M*(Θ*\**) ≠ Θ*\**, there is a contradiction *L*(*M*(Θ*\**)) = *L*(Θ*\**) > *L*(*M*(Θ*\**)). Therefore, we have *M*(Θ*\**) = Θ*\** ■

**Theorem 3.2** (Dempster, Laird, & Rubin, 1977, p. 7). Suppose is the sequence of estimates resulted from GEM algorithm such that:

1. The sequence is bounded above, and
2. *Q*(Θ(*t*+1) | Θ(*t*)) – *Q*(Θ(*t*) | Θ(*t*)) ≥ *ξ*(Θ(*t*+1) – Θ(*t*))*T*(Θ(*t*+1) – Θ(*t*)) for some scalar *ξ* > 0 and all *t*.

Then the sequence converges to some Θ*\** in the closure of Ω ■

*Proof*. The sequence is non-decreasing according to theorem 3.1 and is bounded above according to the assumption 1 of theorem 3.2 and hence, the sequence converges to some *L\** < +∞. According to Cauchy criterion (Dinh, Pham, Nguyen, & Ta, 2000, p. 34), for all *ε* > 0, there exists a *t*(*ε*) such that, for all *t* ≥ *t*(*ε*) and all *v* ≥ 1:

By applying equation 3.2 and equation 3.3, for all *i* ≥ 1, we obtain:

(Due to *L*(Θ(*t*+*i*–1)) = *Q*(Θ(*t*+*i*–1) | Θ(*t*+*i*–1)) – *H*(Θ(*t*+*i*–1) | Θ(*t*+*i*–1)) according to equation 3.2)

It implies

By applying *v* times the assumption 2 of theorem 3.2, we obtain:

It means that

Where,

Notation |.| denotes length of vector and so |Θ(*t*+*i*) – Θ(*t+i* –1)| is distance between Θ(*t*+*i*) and Θ(*t*+*i* –1). Applying triangular inequality, for any *ε* > 0, for all *t* ≥ *t*(*ε*) and all *v* ≥ 1, we have:

According to Cauchy criterion, the sequence converges to some Θ*\** in the closure of Ω.

Theorem 3.1 indicates that *L*(Θ) is non-decreasing on every iteration of GEM algorithm and is strictly increasing on any iteration such that *Q*(Θ(*t*+1) | Θ(*t*)) > *Q*(Θ(*t*) | Θ(*t*)). The corollaries 3.1 and 3.2 indicate that the optimal estimate is a fixed point of GEM algorithm. Theorem 3.2 points out convergence condition of GEM algorithm. However, there is still no assertion of convergence yet and so we need mathematical tools of derivative and differential to prove convergence of GEM. We assume that *Q*(Θ’ | Θ), *L*(Θ), *H*(Θ’ | Θ), and *M*(Θ) are smooth enough. As a convention for derivatives of bivariate function, let *Dij* denote as the derivative (differential) by taking *i*th-order partial derivative (differential) with regard to first variable and then, taking *j*th-order partial derivative (differential) with regard to second variable. If *i* = 0 (*j* = 0) then, there is no partial derivative with regard to first variable (second variable). For example, following is an example of how to calculate the derivative *D*11*Q*(Θ(*t*) | Θ(*t*+1)).

* Firstly, we determine
* Secondly, we substitute Θ(*t*) and Θ(*t*+1) for such *D*11*Q*(Θ’ | Θ) to obtain *D*11*Q*(Θ(*t*) | Θ(*t*+1)).

Equation 3.1 shows some derivatives (differentials) of *Q*(Θ’ | Θ), *H*(Θ’ | Θ), *L*(Θ), and *M*(Θ).

|  |
| --- |
|  |

**Table 3.1.** Some differentials of *Q*(Θ’ | Θ), *H*(Θ’ | Θ), *L*(Θ), and *M*(Θ)

When Θ’ and Θ are vectors, *D*10(…) is gradient vector and *D*20(…) is Hessian matrix. As a convention, let **0** = (0, 0,…, 0)*T* be zero vector.

**Lemma 3.2** (Dempster, Laird, & Rubin, 1977, p. 8). For all Θ in Ω,

|  |  |
| --- | --- |
|  | (3.7) |

|  |  |
| --- | --- |
|  | (3.8) |

|  |  |
| --- | --- |
|  | (3.9) |

|  |  |
| --- | --- |
|  | (3.10) |

|  |  |
| --- | --- |
|  | (3.11) |

|  |  |
| --- | --- |
|  | (3.12) |

Note, *VN*(.) denotes non-central covariance matrix. Followings are proofs of equation 3.7, equation 3.8, equation 3.9, equation 3.10, equation 3.11, and equation 3.12. In fact, we have:

It implies:

We also have:

It implies:

We also have:

It implies:

From equation 3.2, we have:

We also have:

It implies:

We have:

It implies:

Due to:

We have:

**Lemma 3.3** (Dempster, Laird, & Rubin, 1977, p. 9). If *f*(*X* | Θ) and *k*(*X* | *Y*, Θ) belong to exponential family, for all Θ in Ω, we have:

|  |  |
| --- | --- |
|  | (3.13) |

|  |  |
| --- | --- |
|  | (3.14) |

|  |  |
| --- | --- |
|  | (3.15) |

|  |  |
| --- | --- |
|  | (3.16) |

*Proof*. If *f*(*X* | Θ’) and *k*(*X* | *Y*, Θ’) belong to exponential family, from table 1.1 we have:

And,

And,

And,

Hence,

We have:

We have:

We have:

**Theorem 3.3** (Dempster, Laird, & Rubin, 1977, p. 8). Suppose the sequence is an instance of GEM algorithm such that

Then for all *t*, there exists a Θ0(*t*+1) on the line segment joining Θ(*t*) and Θ(*t*+1) such that

Furthermore, if *D*20*Q*(Θ0(*t*+1) | Θ(*t*)) is negative definite, and the sequence is bounded above then, the sequence converges to some Θ*\** in the closure of Ω ■

Note, if Θ is a scalar parameter, *D*20*Q*(Θ0(*t*+1) | Θ(*t*)) degrades as a scalar and the concept “negative definite” becomes “negative” simply. Following is a proof of theorem 3.3.

*Proof*. Second-order Taylor series expending for *Q*(Θ | Θ(*t*)) at Θ = Θ(*t*+1) to obtain:

Where Θ0(*t*+1) is on the line segment joining Θ and Θ(*t*+1). Let Θ = Θ(*t*), we have:

If *D*20*Q*(Θ(*t*+1) | Θ(*t*)) is negative definite then,

Whereas,

So there exists some *ξ* > 0 such that

In other words, the assumption 2 of theorem 3.2 is satisfied and hence, the sequence converges to some Θ*\** in the closure of Ω if the sequence is bounded above ■

**Theorem 3.4** (Dempster, Laird, & Rubin, 1977, p. 9). Suppose the sequence is an instance of GEM algorithm such that

1. The sequence converges to Θ*\** in the closure of Ω.
2. *D*10*Q*(Θ(*t*+1) | Θ(*t*)) = **0***T* for all *t*.
3. *D*20*Q*(Θ(*t*+1) | Θ(*t*)) is negative definite for all *t*.

Then *DL*(Θ*\**) = **0***T*, *D*20*Q*(Θ*\** | Θ*\**) is negative definite, and

|  |  |
| --- | --- |
|  | (3.17) |

The notation “–1” denotes inverse of matrix. Note, *DM*(Θ*\**) is differential of *M*(Θ) at Θ = Θ\*, which implies convergence of GEM algorithm. Followings are proofs of theorem 3.4.

From equation 3.2, we have:

When *t* approaches +∞ such that Θ(*t*) = Θ(*t*+1) = Θ*\** then, *D*10*H*(Θ*\** | Θ*\**) is zero according to equation 3.7 and so we have:

*DL*(Θ*\**) = **0***T*

Of course, *D*20*Q*(Θ*\** | Θ*\**) is negative definite because *D*20*Q*(Θ(*t*+1) | Θ(*t*)) is negative definite, when *t* approaches +∞ such that Θ(*t*) = Θ(*t*+1) = Θ*\**.

By first-order Taylor series expansion for *D*10*Q*(Θ2 | Θ1) as a function of Θ1 at Θ1 = Θ*\** and as a function of Θ2 at Θ2 = Θ*\**, respectively, we have:

Where *R*1(Θ1) and *R*2(Θ2) are remainders. By summing such two series, we have:

By substituting Θ1 = Θ(*t*) and Θ2 = Θ(*t*+1), we have:

Due to *D*10*Q*(Θ(*t*+1) | Θ(*t*)) = **0***T*, we obtain:

It implies:

Multiplying two sides of the equation above by *D*20*Q*(Θ*\** | Θ(*t*))–1 and letting *M*(Θ(*t*)) = Θ(*t*+1), *M*(Θ\*) = Θ\*, we obtain:

Let *t* approach +∞ such that Θ(*t*) = Θ(*t*+1) = Θ*\**, we obtain *DM*(Θ*\**) as differential of *M*(Θ) at Θ*\** as follows:

|  |  |
| --- | --- |
|  | (3.18) |

Due to, when *t* approaches +∞, we have:

The derivative *D*11*Q*(Θ’ | Θ) is expended as follows:

It implies:

(Due to theorem 3.4)

(Due to equation 3.8)

Therefore, equation 3.18 becomes equation 3.17.

Finally, theorem 3.4 is proved. By combination of theorems 3.2 and 3.4, corollary 3.3 is a criterion of convergence of GEM.

**Corollary 3.3.** If an algorithm satisfies three following assumptions:

1. *Q*(*M*(Θ(*t*)) | Θ(*t*)) > *Q*(Θ(*t*) | Θ(*t*)) for all *t*.
2. The sequence is bounded above.
3. *D*10*Q*(Θ*\** | Θ*\**) = **0***T* and *D*20*Q*(Θ*\** | Θ*\**) negative definite.

Then,

1. Such algorithm is an GEM and converges to a local maximizer Θ*\** of *L*(Θ) such that *DL*(Θ*\**) = **0***T* and *D*2*L*(Θ*\**) negative definite.
2. Equation 3.17 is obtained ■

The assumption 1 of corollary 3.3 implies that the given algorithm is a GEM according to definition 3.1. From such assumption, we also have:

So there exists some *ξ* > 0 such that

In other words, the assumption 2 of theorem 3.2 is satisfied and hence, the sequence converges to some Θ*\** in the closure of Ω when the sequence is bounded above according to the assumption 2 of corollary 3.3. From equation 3.2, we have:

When *t* approaches +∞ such that Θ(*t*) = Θ(*t*+1) = Θ*\** then,

*DL*(Θ*\**) = *D*10*Q*(Θ*\** | Θ*\**) – *D*10*H*(Θ*\** | Θ*\**)

*D*10*H*(Θ*\** | Θ*\**) is zero according to equation 3.7. Hence, along with the assumption 3 of corollary 3.3, we have:

*DL*(Θ*\**) = *D*10*Q*(Θ*\** | Θ*\**) = **0***T*

Due to *DL*(Θ*\**) = 0, we only assert here that the given algorithm converges to Θ*\** as a stationary point of *L*(Θ). Later on, we will prove that Θ*\** is a local maximizer of *L*(Θ) when *Q*(*M*(Θ(*t*)) | Θ(*t*)) > *Q*(Θ(*t*) | Θ(*t*)), *DL*(Θ*\**) = 0, and *D*20*Q*(Θ*\** | Θ*\**) negative definite. Due to *D*10*Q*(Θ*\** | Θ*\**) = **0***T*, we obtain equation 3.17 ■

By default, suppose all GEM algorithms satisfy the assumptions 2 and 3 of corollary 3.3. Thus, we only check the assumption 1 to verify whether a given algorithm is a GEM which converges to local maximizer Θ*\**. Note, if the assumption 1 of corollary 3.3 is replaced by “*Q*(*M*(Θ(*t*)) | Θ(*t*)) ≥ *Q*(Θ(*t*) | Θ(*t*)) for all *t*” then, Θ*\** is only asserted to be a stationary point of *L*(Θ) such that *DL*(Θ*\**) = **0***T*. Wu (Wu, 1983) gave a deep research on convergence of GEM in her/his article “On the Convergence Properties of the EM Algorithm”. Please read this article for more details about convergence of GEM.

Because *H*(Θ’ | Θ) and *Q*(Θ’ | Θ) are smooth enough, *D*20*H*(Θ*\** | Θ*\**) and *D*20*Q*(Θ*\** | Θ*\**) are symmetric matrices according to Schwarz’s theorem (Wikipedia, Symmetry of second derivatives, 2018). Thus, *D*20*H*(Θ*\** | Θ*\**) and *D*20*Q*(Θ*\** | Θ*\**) are commutative:

*D*20*H*(Θ*\** | Θ*\**)*D*20*Q*(Θ*\** | Θ*\**) = *D*20*Q*(Θ*\** | Θ*\**)*H*20*Q*(Θ*\** | Θ*\**)

Suppose both *D*20*H*(Θ*\** | Θ*\**) and *D*20*Q*(Θ*\** | Θ*\**) are diagonalizable then, they are simultaneously diagonalizable (Wikipedia, Commuting matrices, 2017). Hence there is a (orthogonal) eigenvector matrix *U* such that (Wikipedia, Diagonalizable matrix, 2017) (StackExchange, 2013):

Where *He\** and *Qe\** are eigenvalue matrices of *D*20*H*(Θ*\** | Θ*\**) and *D*20*Q*(Θ*\** | Θ*\**), respectively, according to equation 3.19 and equation 3.20. Of course, *h*1*\**, *h*2*\**,…, *hr\** are eigenvalues of *D*20*H*(Θ*\** | Θ*\**) whereas *q*1*\**, *q*2*\**,…, *qr\** are eigenvalues of *D*20*Q*(Θ*\** | Θ*\**).

|  |  |
| --- | --- |
|  | (3.19) |

|  |  |
| --- | --- |
|  | (3.20) |

From equation 3.17, *DM*(Θ*\**) is decomposed as seen in equation 3.21.

|  |  |
| --- | --- |
|  | (3.21) |

Let *Me\** be eigenvalue matrix of *DM*(Θ*\**), specified by equation 15. As a convention *Me\** is called convergence matrix.

|  |  |
| --- | --- |
|  | (3.22) |

Of course, all *mi\** = *hi\** / *qi\** are eigenvalues of *DM*(Θ*\**) with assumption *qi\** < 0 for all *i*. We will prove that 0 ≤ *mi\** ≤ 1 for all *i* by contradiction. Conversely, suppose we *always* have *mi\** > 1 or *mi\** < 0 for some *i*. When Θ degrades into scalar as Θ = *θ* with note that scalar is 1-element vector, equation 3.17 is re-written as equation 3.23:

|  |  |
| --- | --- |
|  | (3.23) |

From equation 3.23, the next estimate *θ*(*t*+1) approaches*θ\** when *t* → +∞ and so we have:

So equation 3.24 is a variant of equation 3.23 (McLachlan & Krishnan, 1997, p. 120).

|  |  |
| --- | --- |
|  | (3.24) |

Because the sequence is non-decreasing, the sequence is monotonous. This means:

Or

It implies

So we have

However, this contradicts the converse assumption “there always exists *mi\** > 1 or *mi\** < 0 for some *i*”. Therefore, we conclude that 0 ≤ *mi\** ≤ 1 for all *i*. In general, if Θ*\** is stationary point of GEM then*, D*20*Q*(Θ*\** | Θ*\**) and *Qe\** are negative definite, *D*20*H*(Θ*\** | Θ*\**) and *He\** are negative semi-definite, and *DM*(Θ*\**) and *Me\** are positive semi-definite, according to equation 3.25.

|  |  |
| --- | --- |
|  | (3.25) |

As a convention, if GEM algorithm fortunately stops at the first iteration such that Θ(1) = Θ(2) = Θ*\** then, *mi\** = 0 for all *i*.

Suppose Θ(*t*) = (*θ*1(*t*), *θ*2(*t*),…, *θr*(*t*)) at current *t*th iteration and Θ*\** = (*θ*1*\**, *θ*2*\**,…, *θr\**), each *mi\** measures how much the next *θi*(*t*+1) is near to *θi*\*. In other words, the smaller the *mi\** (s) are, the faster the GEM is and so the better the GEM is. This is why DLR (Dempster, Laird, & Rubin, 1977, p. 10) defined that the convergence rate *m\** of GEM is the maximum one among all *mi\**, as seen in equation 3.26. The convergence rate *m\** implies lowest speed.

|  |  |
| --- | --- |
|  | (3.26) |

From equation 3.2 and equation 3.17, we have (Dempster, Laird, & Rubin, 1977, p. 10):

Where *I* is identity matrix:

By the same way to draw convergence matrix *Me\** with note that *D*20*H*(Θ*\** | Θ*\**), *D*20*Q*(Θ*\** | Θ*\**), and *DM*(Θ*\**) are symmetric matrices, we have:

|  |  |
| --- | --- |
|  | (3.27) |

Where *Le\** is eigenvalue matrix of *D*2*L*(Θ*\**). From equation 3.27, each eigenvalue *li\** of *Le\** is proportional to each eigenvalues *qi\** of *Qe\** with ratio 1–*mi\** where *mi\** is an eigenvalue of *Me\**. Equation 3.28 specifies a so-called speed matrix *Se\**:

|  |  |
| --- | --- |
|  | (3.28) |

From equation 3.25 and equation 3.28, we have 0 ≤ *si\** ≤ 1. Equation 3.29 specifies *Le\** which is eigenvalue matrix of *D*2*L*(Θ*\**).

|  |  |
| --- | --- |
|  | (3.29) |

From equation 3.28, suppose Θ(*t*) = (*θ*1(*t*), *θ*2(*t*),…, *θr*(*t*)) at current *t*th iteration and Θ*\** = (*θ*1*\**, *θ*2*\**,…, *θr\**), each *si\** = 1–*mi\** is really the speed that the next *θi*(*t*+1) moves to *θi*\*. From equation 3.26 and equation 3.28, equation 3.30 specifies the speed *s\** of GEM algorithm.

|  |  |
| --- | --- |
| Where, | (3.30) |

As a convention, if GEM algorithm fortunately stops at the first iteration such that Θ(1) = Θ(2) = Θ*\** then, *s\** = 1.

For example, when Θ degrades into scalar as Θ = *θ*, the fourth column of table 1.1 (Dempster, Laird, & Rubin, 1977, p. 3) gives sequences which approaches *Me\** = *DM*(*θ\**) through many iterations by the following ratio to determine the limit in equation 3.23 with *θ*\* = 0.6268.

In practice, if GEM is run step by step, *θ\** is not known yet at some *t*th iteration when GEM does not converge yet. Hence, equation 3.24 (McLachlan & Krishnan, 1997, p. 120) is used to make approximation of *Me\** = *DM*(*θ\**) with unknown *θ\** and *θ*(*t*) ≠ *θ*(*t*+1).

It is required only two successive iterations because both *θ*(*t*) and *θ*(*t*+1) are determined at *t*th iteration whereas *θ*(*t*+2) is determined at (*t*+1)th iteration. For example, in table 1.1, given *θ*(1) = 0.5, *θ*(2) = 0.6082, and *θ*(3) = 0.6243, at *t* = 1, we have:

Whereas the real *Me\** = *DM*(*θ\**) is 0.1465 shown in the fourth column of table 1.1 at *t* = 1.

We will prove by contradiction that if definition 3.1 is satisfied strictly such that *Q*(*M*(Θ(*t*)) | Θ(*t*)) > *Q*(Θ(*t*) | Θ(*t*)) then, *li\** < 0 for all *i*. Conversely, suppose we *always* have *li\** ≥ 0 for some *i* when *Q*(*M*(Θ(*t*)) | Θ(*t*)) > *Q*(Θ(*t*) | Θ(*t*)). Given Θ degrades into scalar as Θ = *θ* with note that scalar is 1-element vector, when *Q*(*M*(Θ(*t*)) | Θ(*t*)) > *Q*(Θ(*t*) | Θ(*t*)), the sequence is strictly increasing, which in turn causes that the sequence is strictly monotonous. This means:

Or

It implies

So we have

From equation 3.29, we deduce that *D*2*L*(*θ\**) = *Le\** = *Qe\*Se\** < 0 where *Qe\** = *D*20*Q*(*θ\** | *θ\**) < 0. However, this contradicts the converse assumption “there always exists *li\** ≥ 0 for some *i* when *Q*(*M*(Θ(*t*)) | Θ(*t*)) > *Q*(Θ(*t*) | Θ(*t*))”. Therefore, if *Q*(*M*(Θ(*t*)) | Θ(*t*)) > *Q*(Θ(*t*) | Θ(*t*)) then, *li\** < 0 for all *i.* In other words, at that time, *D*2*L*(Θ*\**) = *Le\** is negative definite. Recall that we proved that *DL*(Θ*\**) = 0 for corollary 3.3. Now we have *D*2*L*(Θ*\**) negative definite, which means that Θ*\** is a local maximizer of *L*(Θ*\**) in corollary 3.3. In other words, corollary 3.3 is proved.

Recall that *L*(Θ) is the log-likelihood function of observed *Y* according to equation 2.3.

Both –*D*20*H*(Θ*\** | Θ*\**) and –*D*20*Q*(Θ*\** | Θ*\**) are information matrices (Zivot, 2009, pp. 7-9) specified by equation 3.31.

|  |  |
| --- | --- |
|  | (3.31) |

*IH*(Θ*\**) measures information of *X* about Θ*\** with support of *Y* whereas *IQ*(Θ*\**) measures information of *X* about Θ*\**. In other words, *IH*(Θ*\**) measures observed information whereas *IQ*(Θ*\**) measures hidden information. Let *VH*(Θ*\**) and *VQ*(Θ*\**) be covariance matrices of Θ*\** with regard to *IH*(Θ*\**) and *IQ*(Θ*\**), respectively. They are inverses of *IH*(Θ*\**) and *IQ*(Θ*\**) according to equation 3.32.

|  |  |
| --- | --- |
|  | (3.32) |

Equation 3.33 is a variant of equation 3.17 to calculate *DM*(Θ*\**) based on information matrices:

|  |  |
| --- | --- |
|  | (3.33) |

If *f*(*X* | Θ), *g*(*Y* | Θ) and *k*(*X* | *Y*, Θ) belong to exponential family, from equation 3.14 and equation 3.16, we have:

Hence, equation 3.34 specifies *DM*(Θ*\**) in case of exponential family.

|  |  |
| --- | --- |
|  | (3.34) |

Equation 3.35 specifies relationships among *VH*(Θ*\**), *VQ*(Θ*\**), *V*(*τ*(*X*) | *Y*, Θ*\**), and *V*(*τ*(*X*) | Θ*\**) in case of exponential family.

|  |  |
| --- | --- |
|  | (3.35) |

# **4. Variants of EM algorithm**

The main purpose of EM algorithm (GEM algorithm) is to maximize the log-likelihood *L*(Θ) = log(*g*(*Y* | Θ)) with observed data *Y* by maximizing the condition expectation *Q*(Θ’ | Θ). Such *Q*(Θ’ | Θ) is defined fixedly in E-step. Therefore, most variants of EM algorithm focus on how to maximize *Q*(Θ’ | Θ) in M-step more effectively so that EM is faster or more accurate.

## **4.1. EM algorithm with prior probability**

DLR (Dempster, Laird, & Rubin, 1977, pp. 6, 11) mentioned that the convergence rate *DM*(Θ*\**) specified by equation 3.17 can be improved by adding a prior probability *π*(Θ) in conjugation with *f*(*X* | Θ), *g*(*Y* | Θ) or *k*(*X* | *Y*, Θ) according to maximum a posteriori probability (MAP) method (Wikipedia, Maximum a posteriori estimation, 2017). For example, if *π*(Θ) in conjugation with *g*(*Y* | Θ) then, the posterior probability *π*(Θ | *Y*) is:

Because is constant with regard to Θ, the optimal likelihood-maximization estimate Θ*\** is a maximizer of *g*(*Y* | Θ)*π*(Θ). When π(Θ) is conjugate prior of the posterior probability *π*(Θ | *X*) (or *π*(Θ | *Y*)), both *π*(Θ) and *π*(Θ | *X*) (or *π*(Θ | *Y*)) have the same distributions (Wikipedia, Conjugate prior, 2018); for example, if *π*(Θ) is distributed normally, *π*(Θ | *X*) (or *π*(Θ | *Y*)) is also distributed normally.

For GEM algorithm, the log-likelihood function associated MAP method is specified by equation 4.1.1 with note that *π*(Θ) is non-convex function.

|  |  |
| --- | --- |
|  | (4.1.1) |

It implies from equation 3.2 that

Let,

|  |  |
| --- | --- |
|  | (4.1.2) |

GEM algorithm now aims to maximize *Q*+(Θ’ | Θ) instead of maximizing *Q*(Θ’ | Θ). The proof of convergence for *Q*+(Θ’ | Θ) is not changed in manner but determining the convergence matrix *Me* for *Q*+(Θ’ | Θ) is necessary. Because *H*(Θ’ | Θ) is kept intact whereas *Q*(Θ’ | Θ) is replaced by *Q*+(Θ’ | Θ), we expect that the convergence rate *m\** specified by equation 3.26 is smaller so that the convergence speed *s\** is increased and so GEM algorithm is improved with regard to *Q*+(Θ’ | Θ). Equation 4.1.3 specifies *DM*(Θ*\**) for *Q*+(Θ’ | Θ).

|  |  |
| --- | --- |
|  | (4.1.3) |

Where *Q*+(Θ’ | Θ) is specified by equation 4.1.2 and *D*20*Q*+(Θ’ | Θ) is specified by equation 4.1.4.

|  |  |
| --- | --- |
|  | (4.1.4) |

Where,

Because *Q*(Θ’ | Θ) and *π*(Θ’) are smooth enough, *D*20*Q*(Θ*\** | Θ*\**) and *D*20*L*(*π*(Θ*\**)) are symmetric matrices according to Schwarz’s theorem (Wikipedia, Symmetry of second derivatives, 2018). Thus, *D*20*Q*(Θ*\** | Θ*\**) and *D*20*L*(*π*(Θ*\**)) are commutative:

*D*20*Q*(Θ*\** | Θ*\**)*D*20*L*(*π*(Θ*\**)) = *D*20*L*(*π*(Θ*\**))*D*20*Q*(Θ*\** | Θ*\**)

Suppose both *D*20*Q*(Θ*\** | Θ*\**) and *D*20*L*(*π*(Θ*\**)) are diagonalizable then, they are simultaneously diagonalizable (Wikipedia, Commuting matrices, 2017). Hence there is a (orthogonal) eigenvector matrix *V* such that (Wikipedia, Diagonalizable matrix, 2017) (StackExchange, 2013):

Where *Qe\** and Π*e\** are eigenvalue matrices of *D*20*Q*(Θ*\** | Θ*\**) and *D*20*L*(*π*(Θ*\**)), respectively. Note *Qe\** and its eigenvalues are mentioned in equation 3.20. Because *π*(Θ*\**) is non-convex function, eigenvalues *π*1*\**, *π*2*\**,…, *πr\** of Π*e\** are non-positive.

From equation 4.1.2, *D*20*Q*+(Θ*\** | Θ*\**) is decomposed as below:

So eigenvalue matrix of *D*20*Q*+(Θ*\** | Θ*\**) is (*Qe\** + Π*e\**) and eigenvalues of *D*20*Q*+(Θ*\** | Θ*\**) are *qi\** + *πi\**, as follows:

According to equation 3.19, the eigenvalue matrix of *D*20*H*(Θ*\** | Θ*\**) is *He\** fixed as follows:

Due to *DM*(Θ*\**) = *D*20*H*(Θ*\** | Θ*\**)*D*20*Q*+(Θ*\** | Θ*\**), equation 3.21 is re-calculated:

As a result, the convergence matrix *Me\** which is eigenvalue matrix of *DM*(Θ*\**) is re-calculated by equation 4.1.5.

|  |  |
| --- | --- |
|  | (4.1.5) |

The convergence rate *m\** of GEM is re-defined by equation 4.1.6.

|  |  |
| --- | --- |
|  | (4.1.6) |

Because all *hi\**, *qi\**, and *πi\** are non-positive, we have:

Therefore, by comparing equation 4.1.6 and equation 3.26, we conclude that *m\** is smaller with regard to *Q*+(Θ’ | Θ). In other words, the convergence rate is improved with support of prior probability *π*(Θ). In literature of EM, the combination of GEM and MAP with support of *π*(Θ) results out a so-called MAP-GEM algorithm.

## **4.2. EM algorithm with Newton-Raphson method**

In the M-step of GEM algorithm, the next estimate Θ(*t*+1) is a maximizer of *Q*(Θ | Θ(*t*)), which means that Θ(*t*+1) is a solution of equation *D*10*Q*(Θ | Θ(*t*)) = **0***T* where *D*10*Q*(Θ | Θ(*t*)) is the first-order derivative of *Q*(Θ | Θ(*t*)) with regard to variable Θ. Newton-Raphson method (McLachlan & Krishnan, 1997, p. 29) is applied into solving the equation *D*10*Q*(Θ | Θ(*t*)) = **0***T*. As a result, M-step is replaced a so-called Newton step (N-step).

N-step starts with an arbitrary value Θ0 as a solution candidate and also goes through many iterations. Suppose the current parameter is Θ*i*, the next value Θ*i* +1 is calculated based on equation 4.2.1.

|  |  |
| --- | --- |
|  | (4.2.1) |

N-step converges after some *i*th iteration. At that time, Θ*i*+1 is solution of equation *D*10*Q*(Θ | Θ(*t*)) = 0 if Θ*i*+1=Θ*i*. So the next parameter of GEM is Θ(*t*+1) = Θ*i*+1. The equation 4.2.1 is Newton-Raphson process. Recall that *D*10*Q*(Θ | Θ(*t*)) is gradient vector and *D*20*Q*(Θ | Θ(*t*)) is Hessian matrix. Following is a proof of equation 4.2.1.

According to first-order Taylor series expansion of *D*10*Q*(Θ | Θ(*t*)) at Θ = Θ*i* with very small residual, we have:

Because *Q*(Θ | Θ(*t*)) is smooth enough, *D*20*Q*(Θ | Θ(*t*)) is symmetric matrix according to Schwarz’s theorem (Wikipedia, Symmetry of second derivatives, 2018), which implies:

*D*20*Q*(Θ | Θ(*t*)) = (*D*20*Q*(Θ | Θ(*t*)))*T*

So we have:

Let Θ = Θ*i*+1 and we expect that *D*10*Q*(Θ*i*+1 | Θ(*t*)) = **0***T* so that Θ*i*+1 is a solution.

It implies:

This means:

Rai and Matthews (Rai & Matthews, 1993) proposed a so-called EM1 algorithm in which Newton-Raphson process is reduced into one iteration, as seen in table 4.2.1 (Rai & Matthews, 1993, pp. 587-588). Rai and Matthews assumed that *f*(*x*) belongs to exponential family but their EM1 algorithm is really a variant of GEM in general. In other words, there is no requirement of exponential family for EM1.

|  |  |  |
| --- | --- | --- |
| *E-step*:  The expectation *Q*(Θ | Θ(*t*)) is determined based on current Θ(*t*), according to equation 2.8. Actually, *Q*(Θ | Θ(*t*)) is formulated as function of Θ.  *M-step*:  The next parameter Θ(*t*+1) is:   |  |  | | --- | --- | |  | (4.2.2) | |

**Table 4.2.1.** E-step and M-step of EM1 algorithm

Rai and Matthews proved convergence of EM1 algorithm by their proposal of equation 4.2.2. Second-order Taylor series expending for *Q*(Θ | Θ(*t*)) at Θ = Θ(*t*+1) to obtain:

Where Θ0(*t*+1) is on the line segment joining Θ and Θ(*t*+1). Let Θ = Θ(*t*), we have:

By substituting equation 4.2.2 for *Q*(Θ(*t*+1) | Θ(*t*)) – *Q*(Θ(*t*) | Θ(*t*)) with note that *D*20*Q*(Θ | Θ(*t*)) is symmetric matrix, we have:

Let,

Because *Q*(Θ’ | Θ) is smooth enough, *D*20*Q*(Θ(*t*) | Θ(*t*)) and *D*20*Q*(Θ0(*t*+1) | Θ(*t*)) are symmetric matrices according to Schwarz’s theorem (Wikipedia, Symmetry of second derivatives, 2018). Thus, *D*20*Q*(Θ(*t*) | Θ(*t*)) and *D*20*Q*(Θ0(*t*+1) | Θ(*t*)) are commutative:

*D*20*Q*(Θ(*t*) | Θ(*t*))*D*20*Q*(Θ0(*t*+1) | Θ(*t*)) = *D*20*Q*(Θ0(*t*+1) | Θ(*t*))*D*20*Q*(Θ(*t*) | Θ(*t*))

Suppose both *D*20*Q*(Θ(*t*) | Θ(*t*)) and *D*20*Q*(Θ0(*t*+1) | Θ(*t*)) are diagonalizable then, they are simultaneously diagonalizable (Wikipedia, Commuting matrices, 2017). Hence there is a (orthogonal) eigenvector matrix *V* such that (Wikipedia, Diagonalizable matrix, 2017) (StackExchange, 2013):

Where *Qe*(*t*) and *Qe*(*t*+1) are eigenvalue matrices of *D*20*Q*(Θ(*t*) | Θ(*t*)) and *D*20*Q*(Θ0(*t*+1) | Θ(*t*)), respectively. Matrix *A* is decomposed as below:

(Because *Qe*(*t*) and *Qe*(*t*+1) are commutative)

Hence, eigenvalue matrix of *A* is also *Qe*(*t*+1). Suppose *D*20*Q*(Θ0(*t*+1) | Θ(*t*)) is negative definite, *A* is negative definite too. We have:

Because *D*20*Q*(Θ(*t*) | Θ(*t*)) is negative definite, we have:

Because *A* is negative definite, we have:

As a result, we have:

Hence, EM1 surely converges to a local maximizer Θ*\** according to corollary 3.3 with assumption that *D*20*Q*(Θ0(*t*+1) | Θ(*t*)) and *D*20*Q*(Θ(*t*) | Θ(*t*)) are negative definite for all *t*.

Rai and Matthews made experiment on their EM1 algorithm (Rai & Matthews, 1993, p. 590). As a result, EM1 algorithm saved a lot of computations in M-step. In fact, by comparing GEM (table 2.3) and EM1 (table 4.2.1), we conclude that EM1 increases *Q*(Θ | Θ(*t*)) after each iteration whereas GEM maximizes *Q*(Θ | Θ(*t*)) after each iteration. However, EM1 will maximizes *Q*(Θ | Θ(*t*)) at the last iteration when it converges. EM1 gains this excellent and interesting result because of Newton-Raphson process specified by equation 4.2.2.

Because equation 3.17 is not changed with regard to EM1, the convergence matrix of EM1 is not changed.

Therefore, EM1 does not improve convergence rate in theory as MAP-GEM algorithm does but EM1 algorithm really speeds up GEM process in practice because it saves computational cost in M-step.

In equation 4.2.2, the second-order derivative *D*20*Q*(Θ(*t*) | Θ(*t*)) is re-computed at every iteration for each Θ(t). If *D*20*Q*(Θ(*t*) | Θ(*t*)) is complicated, it can be fixed by *D*20*Q*(Θ(1) | Θ(1)) over all iterations where Θ(1) is arbitrarily initialized for EM process so as to save computational cost. In other words, equation 4.2.2 is replaced by equation 4.2.3 (Ta, 2014).

|  |  |
| --- | --- |
|  | (4.2.3) |

In equation 4.2.3, only *D*10*Q*(Θ(*t*) | Θ(*t*)) is re-computed at every iteration whereas *D*20*Q*(Θ(1) | Θ(1)) is fixed. Equation 4.2.3 implies a pseudo Newton-Raphson process which still converges to a local maximizer Θ*\** but it is slower than Newton-Raphson process specified by equation 4.2.2 (Ta, 2014).

Newton-Raphson process specified by equation 4.2.2 has second-order convergence. I propose to use equation 4.2.4 for speeding up EM1 algorithm. In other words, equation 4.2.2 is replaced by equation 4.2.4 (Ta, 2014), in which Newton-Raphson process is improved with third-order convergence. Note, equation 4.2.4 is common in literature of Newton-Raphson process.

|  |  |
| --- | --- |
| Where, | (4.2.4) |

The convergence of equation 4.2.4 is same as the convergence of equation 4.2.2. Following is a proof of equation 4.2.4 by Ta (Ta, 2014).

Without loss of generality, suppose Θ is scalar such that Θ = *θ*, let

Let *r*(*θ*) represents improved Newton-Raphson process.

Suppose *ω*(*θ*) has first derivative and we will find *ω*(*θ*). According to Ta (Ta, 2014), the first-order derivative of *η*(*θ*) is:

According to Ta (Ta, 2014), the second-order derivative of *η*(*θ*) is:

If is solution of equation *q*(*θ*) = 0, Ta (Ta, 2014) gave:

In order to achieve , Ta (Ta, 2014) selected:

According to Ta (Ta, 2014), Newton-Raphson process is improved as follows:

This means:

As a result, equation 4.2.4 is a generality of the equation above when Θ is vector.

I propose to apply gradient descent method (Ta, 2014) into M-step of GEM so that Newton-Raphson process is replaced by gradient descent process with expectation that descending direction which is the opposite of gradient vector *D*10*Q*(Θ | Θ(*t*)) speeds up convergence of GEM. Table 4.2.2 specifies GEM associated with gradient descent method, which is called GD-GEM algorithm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *E-step*:  The expectation *Q*(Θ | Θ(*t*)) is determined based on current Θ(*t*), according to equation 2.8. Actually, *Q*(Θ | Θ(*t*)) is formulated as function of Θ.  *M-step*:  The next parameter Θ(*t*+1) is:   |  |  | | --- | --- | |  | (4.2.5) |   Where *γ*(*t*) > 0 is length of the descending direction. As usual, *γ*(*t*) is selected such that   |  |  | | --- | --- | |  | (4.2.6) |   Where, |

**Table 4.2.1.** E-step and M-step of GD-GEM algorithm

Note, gradient descent method is used to solve minimization problem but its use for solving maximization problem is the same. Second-order Taylor series expending for *Q*(Θ | Θ(*t*)) at Θ = Θ(*t*+1) to obtain:

Where Θ0(*t*+1) is on the line segment joining Θ and Θ(*t*+1). Let Θ = Θ(*t*), we have:

By substituting equation 4.2.5 for *Q*(Θ(*t*+1) | Θ(*t*)) – *Q*(Θ(*t*+1) | Θ(*t*)), we have:

Due to:

As a result, we have:

Hence, GD-GEM surely converges to a local maximizer Θ*\** according to corollary 3.3 with assumption that *D*20*Q*(Θ0(*t*+1) | Θ(*t*)) is negative definite.

It is not easy to solve the maximization problem with regard to *γ* according to equation 4.2.6. So if *Q*(Θ | Θ(*t*)) satisfies Wolfe conditions (Wikipedia, Wolfe conditions, 2017) and concavity and *D*10*Q*(Θ | Θ(*t*)) is Lipschitz continuous (Wikipedia, Lipschitz continuity, 2018) then, equation 4.2.6 is replaced by equation 4.2.7 (Wikipedia, Gradient descent, 2018).

|  |  |
| --- | --- |
|  | (4.2.7) |

Where |.| denotes length or module of vector.

## **4.3. EM algorithm with Aitken acceleration**

According to Lansky and Casella (Lansky & Casella, 1992), GEM converges faster by combination of GEM and Aitken acceleration. Without loss of generality, suppose Θ is scalar such that Θ = *θ*, the sequence is monotonous. From equation 3.23

We have the following approximate with *t* large enough (Lambers, 2009, p. 1):

We establish the following equation from the above approximation, as follows (Lambers, 2009, p. 1):

Hence, *θ\** is approximated by (Lambers, 2009, p. 1)

We construct Aitken sequence such that (Wikipedia, Aitken's delta-squared process, 2017)

|  |  |
| --- | --- |
|  | (4.3.1) |

Where Δ is forward difference operator,

And

When Θ is vector as Θ = (*θ*1, *θ*2,…, *θr*)*T*, Aitken sequence is defined by applying equation 4.3.1 into its components *θi* (s) according to equation 4.3.2:

|  |  |
| --- | --- |
|  | (4.3.2) |

Where,

According theorem of Aitken acceleration, Aitken sequence approaches Θ*\** faster than the sequence with note that the sequence is instance of GEM.

Essentially, the combination of GEM and Aitken acceleration is to replace the sequence by Aitken sequence as seen in table 4.3.1.

|  |
| --- |
| *E-step*:  The expectation *Q*(Θ | Θ(*t*)) is determined based on current Θ(*t*), according to equation 2.8. Actually, *Q*(Θ | Θ(*t*)) is formulated as function of Θ. Note that *t* = 1, 2, 3,… and Θ(0) = Θ(1).  *M-step*:  Let Θ(*t*+1) = (*θ*1(*t*+1), *θ*2(*t*),…, *θr*(*t*+1))*T* be a maximizer of *Q*(Θ | Θ(*t*)). Note Θ(*t*+1) will become current parameter at the next iteration ((*t*+1)th iteration).  Aitken parameter is calculated according to equation 4.3.2.  If then, the algorithm stops and we have . |

**Table 4.3.1.** E-step and M-step of GEM algorithm combined with Aitken acceleration

Because Aitken sequence converges to Θ*\** faster than the sequence does, the convergence of GEM is improved with support of Aitken acceleration method.

In equation 4.3.2, parametric components *θi* (s) converges separately. Guo, Li, and Xu (Guo, Li, & Xu, 2017) assumed such components converges together with the same rate. So they replaced equation 4.3.2 by equation 4.3.3 (Guo, Li, & Xu, 2017, p. 176) for Aitken sequence .

|  |  |
| --- | --- |
|  | (4.3.3) |

## **4.4. ECM algorithm**

Because M-step of GEM is complicated, Meng and Rubin (Meng & Rubin, 1993) proposed a so-called Expectation Conditional Expectation (ECM) algorithm in which M-step is replaced by several computationally simpler Conditional Maximization (CM) steps. Each CM-step maximizes *Q*(Θ | Θ(*t*)) on given constraint. ECM is very useful in the case that maximization of *Q*(Θ | Θ(*t*)) with constraints is simpler than maximization of *Q*(Θ | Θ(*t*)) without constraints as usual.

Suppose the parameter Θ is partitioned into *S* sub-parameters Θ = {Θ1, Θ2,…, Θ*S*} and there are *S* pre-selected vector function *gs*(Θ):

|  |  |
| --- | --- |
|  | (4.4.1) |

Each function *gs*(Θ) represents a constraint. Support there is a sufficient enough number of derivatives of each *gs*(Θ). In ECM algorithm (Meng & Rubin, 1993, p. 268), M-step is replaced by a sequence of CM-steps. Each CM-step maximizes *Q*(Θ | Θ(*t*)) over Θ but with some function *gs*(Θ) fixed at its previous value. Concretely, there are *S* CM-steps and every *s*th CM-step finds Θ(*t*+*s*/*S*) that maximizes *Q*(Θ | Θ(*t*)) over Θ subject to the constraint *gs*(Θ) = *gs*(Θ(*t*+(*s*–1)/*S*)). The next parameter Θ(*t*+1) is the output of the final CM-step such that Θ(*t*+1) = Θ(*t*+*s*/*S*). Table 4.4.1 (Meng & Rubin, 1993, p. 272) shows E-step and CM-steps of ECM algorithm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *E-step*:  As usual, *Q*(Θ | Θ(*t*)) is determined based on current Θ(*t*) according to equation 2.8. Actually, *Q*(Θ | Θ(*t*)) is formulated as function of Θ.  *CM-steps*:  There are *S* CM-steps. In every *s*th CM step (*s* =1, 2,…, *S*), finding   |  |  | | --- | --- | |  | (4.4.2) |   The next parameter Θ(*t*+1) is the output of the final CM-step (*S*th CM-step):   |  |  | | --- | --- | |  | (4.4.3) |   Note, Θ(*t*+1) will become current parameter at the next iteration ((*t*+1)th iteration). |

**Table 4.3.1.** E-step and CM-steps of ECM algorithm

ECM algorithm stops at some *t*th iteration such that Θ(*t*) = Θ(*t*+1) = Θ\*. CM-steps depend on how to define pre-selected functions in *G*. For example, if *gs*(Θ) consists all sub-parameters except Θ*s* then, the *s*th CM-step maximizes *Q*(Θ | Θ(*t*)) with regard to Θ*s* whereas other sub-parameters are fixed. If *gs*(Θ) consists only Θ*s* then, the *s*th CM-step maximizes *Q*(Θ | Θ(*t*)) with regard to all sub-parameters except Θ*s*. Note, definition of ECM algorithm is specified by equation 4.4.2 and equation 4.4.3

From equation 4.4.2 and equation 4.4.3, we have:

Hence, the convergence of ECM is asserted according to corollary 3.3. However, Meng and Rubin (Meng & Rubin, 1993, pp. 274-276) provided some conditions for convergence of ECM to a maximizer of *L*(Θ).

# **5. Applications of EM**

## **5.1. Mixture model and EM**

As usual, let ***X*** be the hidden or latent space and let ***Y*** be the observed space. Especially, the random variable *X* in ***X*** represents latent class or latent component of random variable *Y* in ***Y***. Suppose *X* is discrete and ranges in ***X*** = {1, 2,…, *K*}. The so-called probabilisticfinite *mixture model* is represented by the PDF of *Y*, as seen in equation 5.1.1.

|  |  |
| --- | --- |
|  | (5.1.1) |

Where,

Note, *Y* can be discrete or continuous. Recall that the ultimate purpose of EM algorithm is to maximize *f*(*Y*|Θ) with subject to Θ. Each *fX*(*Y*|*θX*) is called the *X*th partial PDF of *Y* whose partial parameter is *θX*. Each *fX*(*Y*|*θX*) is also called the *X*th observational PDF of *Y*. It is really the conditional PDF of *Y* given *X*, as seen in equation 5.1.2.

|  |  |
| --- | --- |
|  | (5.1.2) |

From equation 5.1.1, the mixture model *f*(*Y*|Θ) is the mean of *K* partial PDFs. The variable *X* implies which partial PDF “generates” *Y* (Bilmes, 1998, p. 5).

Each *αX* is called mixture coefficient. It is really the probability of discrete *X*, as seen in equation 5.1.3. However, in mixture model, each *αX* is also considered as parameter, which is belongs to the compound parameter Θ.

|  |  |
| --- | --- |
|  | (5.1.3) |

The joint probabilistic distribution of *X* and *Y*, which implies the implicit mapping between ***X*** and ***Y***, is product of the mixture coefficient *αX* and the *X*th PDF of *Y*, as seen in equation 5.1.4.

|  |  |
| --- | --- |
|  | (5.1.4) |

This implies:

|  |  |
| --- | --- |
|  | (5.1.5) |

Equation 5.1.6 specifies the conditional probability of *X* given *Y*. Please pay attention to this important probability.

|  |  |
| --- | --- |
|  | (5.1.6) |

Following is the proof of equation 5.1.6. According to Bayes’ rule, we have:

Applying equation 5.1.3 and equation 5.1.4, we have:

In other words, equation 5.1.6 is established■

Now GEM algorithm is applied into mixture model for estimating the parameter Θ. Derived from equation 2.12, the conditional expectation *Q*(Θ’|Θ) of mixture model becomes:

|  |  |
| --- | --- |
|  | (5.1.7) |

In practice, suppose *Y* is observed as a sample = {*Y*1, *Y*2,…, *YN*} of size *N* with note that all *Yi* (s) are mutually independent and identically distributed (iid). The observed sample is associated with a a hidden set (latent set) = {*X*1, *X*2,…, *XN*} of size *N*. All *Xi* (s) are iid and they are not existent in fact. Let be the random variable representing every *Xi*. Of course, the domain of *X* is ***X***. Derived from equation 2.14, equation 5.1.8 specifies *Q*(Θ’|Θ) given such .

|  |  |
| --- | --- |
|  | (5.1.8) |

Equation 5.1.8 is the general case of equation 5.1.7. At the *t*th iteration of GEM, given current parameter Θ(*t*) = (*α*1(*t*), *α*2(*t*),…, *αK*(*t*), *θ*1(*t*), *θ*2(*t*),…, *θK*(*t*))*T*, the conditional expectation specified by equation 5.1.8 is written as follows:

Thus, the unknown of *Q*(Θ|Θ(*t*)) is Θ = (*α*1, *α*2,…, *αK*, *θ*1, *θ*2,…, *θK*)*T*. Because *X* is discrete and ranges in {1, 2,…, *K*}, the conditional expectation *Q*(Θ|Θ(*t*)) is re-written as equation 5.1.9 for convenience.

|  |  |
| --- | --- |
|  | (5.1.9) |

Where the conditional probability *P*(*k* | *Y*, Θ(*t*)) is determined by equation 5.1.10 which is indeed equation 5.1.6.

|  |  |
| --- | --- |
|  | (5.1.10) |

At M-step of the current *t*th iteration, *Q*(Θ|Θ(*t*)) specified by equation 5.1.9 is maximized with subject to Θ. How to maximize *Q*(Θ|Θ(*t*)) with subject to Θ is dependent on types of partial PDFs *fk*(*Yi*|*θk*).

Because there is the constraint , we use Lagrange duality method to maximize to maximize *Q*(Θ|Θ(*t*)). The Lagrange function *la*(Θ, *λ* | Θ(*t*)) is sum of *Q*(Θ|Θ(*t*)) and the constraint , which is specified by equation 5.1.11.

|  |  |
| --- | --- |
|  | (5.1.11) |

Note, *λ* ≥ 0 is called Lagrange multiplier. Of course, *la*(Θ, *λ* | Θ(*t*)) is function of Θ and *λ*. The next parameters *αk*(*t*+1) that maximizes *Q*(Θ|Θ(*t*)) is solution of the equation formed by setting the first-order partial derivative of Lagrange function regarding *αk* and *λ* to be zero with suppose that the Lagrange function is first-order smooth function.

This implies:

|  |  |
| --- | --- |
|  | (5.1.12) |

Summing equation 5.1.12 over *K* classes {1, 2,…, *K*}, we have (Bilmes, 1998, p. 5):

Substituting *λ = N* into equation 5.1.12, the next parameters *αk*(*t*+1) is totally determined by equation 5.1.13.

|  |  |
| --- | --- |
|  | (5.1.13) |

Note, the conditional probability *P*(*k* | *Yi*, Θ(*t*)) is determined by equation 5.1.10.

When parameters *αk*(*t*+1) and *λ* are determined, the Lagrange function *la*(Θ, *λ* | Θ(*t*)) is now function of parameters *θk* as *la*(*θk*|*θk*(*t*)). The next parameters *θk*(*t*+1) is solution of the equation formed by setting the first-order partial derivative of Lagrange function regarding *θk* to be zero with suppose that the Lagrange function is first-order smooth function.

Thus, the next parameters *θk*(*t*+1) is solution of the equation 5.1.14.

|  |  |
| --- | --- |
|  | (5.1.14) |

The two steps of GEM algorithm for constructing mixture model at some *t*th iteration are shown in table 5.1.1. Note, suppose the Lagrange function is first-order smooth function.

|  |
| --- |
| *E-step*:  The conditional probability *P*(*k* | *Yi*, Θ(*t*)) is calculated based on current parameter Θ(*t*) = (*α*1(*t*), *α*2(*t*),…, *αK*(*t*), *θ*1(*t*), *θ*2(*t*),…, *θK*(*t*))*T*, according to equation 5.1.10.  *M-step*:  The next parameter Θ(*t*+1) = (*α*1(*t*+1), *α*2(*t*+1),…, *αK*(*t*+1), *θ*1(*t*+1), *θ*2(*t*+1),…, *θK*(*t*+1))*T*, which is a maximizer of *Q*(Θ | Θ(*t*)) with subject to Θ, is calculated by equation 5.1.13 and equation 5.1.14. Note, *θk*(*t*+1) is solution of the equation 5.1.14. |

**Table 5.1.1.** E-step and M-step of GEM algorithm for constructing mixture model regarding first-order smooth Lagrange function

GEM algorithm converges at some *t*th iteration. At that time, Θ*\** = Θ(*t*+1) = Θ(*t*) is the optimal estimate of mixture model regarding first-order smooth Lagrange function.

Suppose that each PDF *fk*(*Yi*|*θk*) ) belongs to regular exponential family and then, solving equation 5.1.4 is easier as follows:

(Due to *fk*(*Yi*|*θk*) ) belongs to exponential family, please see table 1.1)

(Due to log’(*a*(*θk*)) = *E*(*τ*(*Y*|*θk*)), please see table 1.1)

In general, the next parameters *θk*(*t*+1) is solution of the equation 5.1.15 within regular exponential family.

|  |  |
| --- | --- |
|  | (5.1.15) |

Where *Y* is the random variable representing all *Yi* (s) and,

The two steps of GEM algorithm for constructing mixture model at some *t*th iteration are shown in table 5.1.2 with suppose that each partial PDF *fX*(*Y*|*θX*) is assumed to belong regular exponential family.

|  |
| --- |
| *E-step*:  The conditional probability *P*(*k* | *Yi*, Θ(*t*)) is calculated based on current parameter Θ(*t*) = (*α*1(*t*), *α*2(*t*),…, *αK*(*t*), *θ*1(*t*), *θ*2(*t*),…, *θK*(*t*))*T*, according to equation 5.1.10.  *M-step*:  The next parameter Θ(*t*+1) = (*α*1(*t*+1), *α*2(*t*+1),…, *αK*(*t*+1), *θ*1(*t*+1), *θ*2(*t*+1),…, *θK*(*t*+1))*T*, which is a maximizer of *Q*(Θ | Θ(*t*)) with subject to Θ, is calculated by equation 5.1.13 and equation 5.1.15. Note, *θk*(*t*+1) is solution of the equation 5.1.15. |

**Table 5.1.1.** E-step and M-step of GEM algorithm for constructing mixture model regarding regular exponential family

GEM algorithm converges at some *t*th iteration. At that time, Θ*\** = Θ(*t*+1) = Θ(*t*) is the optimal estimate of mixture model regarding regular exponential family.

There is a special case that each *fk*(*Yi*|*θk*) is normal distribution, which is popular in domain of mixture model, with note that normal distribution belongs to regular exponential family. Thus, let *Y* be random variable representing all *Yi*. Without loss of generality, suppose *Y* is vector so that each *fk*(*Y*|*θk*) is multivariate normal distribution. Recall that each *fk*(*Y*|*θk*) is called the *k*th partial PDF of *Y* or the *k*th observational PDF of *Y*. In this case, the mixture model is called *normal mixture model* (Gaussian mixture model) and it is easy to solve equation 5.1.14 or equation 5.1.15 for *θk*. Suppose random variable *Y* is vector of size *n*.

|  |  |
| --- | --- |
|  | (5.1.16) |

Where *μk* and Σ*k* are mean vector and covariance matrix of *fk*(*Y*|*θk*), respectively. The notation |.| denotes determinant of given matrix and the notation Σ*k*–1 denotes inverse of matrix Σ*k*. Note, Σ*k* is invertible and symmetric. Now we find other parameters *θk*(*t*+1) = (*μk*(*t*+1), Σ*k*(*t*+1))*T* by solving directly equation 5.1.14 or equation 5.1.15. Recall that each *Yi* conforms to multivariate normal distribution, according to equation 5.1.16.

Where *μk* and Σ*k* are mean and covariance matrix of *fk*(*Yi*|*θk*), respectively. The Lagrange function is re-written as follows:

Where *p* is the dimension of *Yi*; in other words, *p* is the dimension of space ***Y***.

The first-order partial derivative of Lagrange function with respect to *μk* is (Nguyen, 2015, p. 35):

The next parameter *μk*(*t*+1) that maximizes *Q*(Θ|Θ(*t*)) is solution of the equation formed by setting the first-order partial derivative of Lagrange function with regard to *μk* to be **0***T*. Note that **0** = (0, 0,…, 0)*T* is zero vector.

This implies equation 5.1.17 to specify the next parameter *μk*(*t*+1).

|  |  |
| --- | --- |
|  | (5.1.17) |

Note, the conditional probability *P*(*k* | *Yi*, Θ(*t*)) is determined by equation 5.1.10.

The first-order partial derivative of Lagrange function with respect to Σ*k* is:

Due to:

And

Because Bilmes (Bilmes, 1998, p. 5) mentioned:

Where tr(*A*) is trace operator which takes sum of diagonal elements of matrix .

This implies (Nguyen, 2015, p. 45):

Where Σ*k* is symmetric and invertible matrix. Substituting the next parameter *μk*(*t*+1) specified by equation 5.1.16 into the first-order partial derivative of Lagrange function with respect to Σ*k*, we have:

The next parameter Σ*k*(*t*+1) that maximizes *Q*(Θ|Θ(*t*)) is the solution of equation formed by setting the first-order partial derivative of Lagrange function regarding Σ*k* to zero matrix. Let [**0**] denote zero matrix.

We have:

This implies equation 5.1.18 to specify the next parameter Σ*k*(*t*+1).

|  |  |
| --- | --- |
|  | (5.1.18) |

Note, the conditional probability *P*(*k* | *Yi*, Θ(*t*)) is determined by equation 5.1.10 and the next parameter *μk*(*t*+1) is specified by equation 5.1.17.

As a result, the solution *θk*(*t*+1) = (*μk*(*t*+1), Σ*k*(*t*+1))*T* of equation 5.1.14 or equation 5.1.15 is specified by equation 5.1.17 and equation 5.1.18 when each *fk*(*Y*|*θk*) is multivariate normal distribution within normal mixture model. The two steps of GEM algorithm for constructing normal mixture model at some *t*th iteration are refined in table 5.1.3 (Bilmes, 1998, p. 7).

|  |
| --- |
| *E-step*:  The conditional probability *P*(*k* | *Yi*, Θ(*t*)) is calculated based on current parameter Θ(*t*) = (*α*1(*t*), *α*2(*t*),…, *αK*(*t*), *θ*1(*t*), *θ*2(*t*),…, *θK*(*t*))*T*, according to equation 5.1.10. Note, in normal mixture model, each observational PDF *fk*(*Y*|*θk*) is (multivariate) normal distribution with mean vector *μk* and covariance matrix Σ*k* such that *θk* = (*μk*, Σ*k*)*T*.  *M-step*:  The next parameter Θ(*t*+1) = (*α*1(*t*+1), *α*2(*t*+1),…, *αK*(*t*+1), *θ*1(*t*+1), *θ*2(*t*+1),…, *θK*(*t*+1))*T*, which is a maximizer of *Q*(Θ | Θ(*t*)) with subject to Θ, is calculated by equation 5.1.13, equation 5.1.17, and equation 5.1.18 with current parameter Θ(*t*). |

**Table 5.1.3.** E-step and M-step of GEM algorithm for constructing normal mixture model

GEM algorithm converges at some *t*th iteration. At that time, Θ*\** = Θ(*t*+1) = Θ(*t*) is the optimal estimate of normal mixture model.

# **6. Discussions**

The convergence of GEM is based on the assumption that *Q*(Θ’ | Θ) is smooth enough but *Q*(Θ’ | Θ) may not be smooth in practice when *f*(*X* | Θ) may be discrete probability function. For example, when *f*(*X* | Θ) and *k*(*X* | *Y*, Θ) are discrete, equation 2.8 becomes

This discussion section goes beyond traditional variants of GEM algorithm when *Q*(Θ’ | Θ) is not smooth. Therefore, heuristic optimization methods which simulate social behavior, such as particle swarm optimization (PSO) algorithm (Poli, Kennedy, & Blackwell, 2007) and artificial bee colony (ABC) algorithm, are useful in case that there is no requirement of existence of derivative. Moreover, these heuristic methods aim to find global optimizer. I propose an association of GEM and PSO which produces a so-called quasi-PSO-GEM algorithm in which M-step is implemented by one-time PSO (Wikipedia, Particle swarm optimization, 2017). Given current *t*th iteration, Θ(*t*) is modeled as swarm’s best position. Suppose there are *n* particles and each particle *i* has current velocity *Vi*(*t*), current positions Ψ*i*(*t*), and best position Φ*i*(*t*). At each iteration, it is expected that these particles move to swarm’s new best position which is the next parameter Θ(*t*+1). The swarm’s best position at the final iteration is expected as Θ*\**. Table 6.2 is the proposal of quasi-PSO-GEM algorithm.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *E-step*:  As usual, *Q*(Θ | Θ(*t*)) is determined based on current Θ(*t*) according to equation 2.8. Actually, *Q*(Θ | Θ(*t*)) is formulated as function of Θ.  *M-step* includes four sub-steps:   1. Calculating the next velocity *Vi*(*t*+1) of each particle based on its current velocity *Vi*(*t*), its current positions Ψ*i*(*t*), its best positions Φ*i*(*t*), and the swarm’s best position Θ(*t*):  |  |  | | --- | --- | |  | (6.1) |   Where *ω*, *ϕ*1, and *ϕ*2 are particular parameters of PSO (Poli, Kennedy, & Blackwell, 2007, pp. 3-4) whereas *r* is a random number such that 0 < *r* < 1 (Wikipedia, Particle swarm optimization, 2017).   1. Calculating the next position Ψ*i*(*t*+1) of each particle based on its current position Ψ*i*(*t*) and its current velocity *Vi*(*t*):  |  |  | | --- | --- | |  | (6.2) |  1. If *Q*(Φ*i*(*t*) | Θ(*t*)) < *Q*(Ψ*i*(*t*+1) | Θ(*t*)) then, the next best position of each particle *i* is re-assigned as Φ*i*(*t*+1) = Ψ*i*(*t*+1). Otherwise, such next best position is kept intact as Φ*i*(*t*+1) = Φ*i*(*t*). 2. The next parameter Θ(*t*+1) is the swarm’s new best position over the best positions of all particles:  |  |  | | --- | --- | |  | (6.3) |   If the bias |Θ(*t*+1) – Θ(*t*)| is small enough, the algorithm stops. Otherwise, Θ(*t*+1) and all *Vi*(*t*+1), Ψ*i*(*t*+1), Φ*i*(*t*+1) become current parameters in the next iteration. |

**Table 6.1.** E-step and M-step of the proposed quasi-PSO-GEM

At the first iteration, each particle is initialized with Ψ*i*(1) = Φ*i*(1) = Θ(1) and uniformly distributed velocity *Vi*(1). Note, Θ(1) is initialized arbitrarily. Other termination criteria can be used, for example, *Q*(Θ | Θ(*t*)) is large enough or the number of iterations is large enough.

We cannot prove mathematically convergence of quasi-PSO-GEM but we expect that Θ(*t*+1) resulted from equation 6.3 is an approximation of Θ*\** at the last iteration after a large enough number of iterations. However, quasi-PSO-GEM tendentiously approaches global maximizer of *L*(Θ), regardless of whether *L*(Θ) is concave. Hence, it is necessary to make experiment on quasi-PSO-GEM.

There are many other researches which combine EM and PSO but the proposed quasi-PSO-GEM algorithm has different ideology when it one-time PSO is embed into M-step to maximize *Q*(Θ | Θ(*t*)) and so the ideology of quasi-PSO-GEM is near to the ideology of Newton-Raphson process. With different viewpoint, some other researches combine EM and PSO in order to solving better a particular problem instead of improving EM itself. For example, Ari and Aksoy (Ari & Aksoy, 2010) used PSO to solve optimization problem of the clustering algorithm based on mixture model and EM. Rajeswari and Gunasundari (Rajeswari & Gunasundari, 2016) proposed EM for PSO based weighted clustering. Zhang, Zhuang, Gao, Luo, Ran, and Du (Zhang, et al., 2014) proposed a so-called PSO-EM algorithm to make optimum use of PSO in partial E-step in order solve the difficulty of integrals in normal compositional model. Golubovic, Olcan, and Kolundzija (Golubovic, Olcan, & Kolundzija, 2007) proposed a few modifications of the PSO algorithm which are applied to EM optimization of a broadside antenna array. Tang, Song, and Liu (Tang, Song, & Liu, 2014) proposed a hybrid clustering method based on improved PSO and EM clustering algorithm to overcome drawbacks of EM clustering algorithm. Tran, Vo, and Lee (Tran, Vo, & Lee, 2013) proposed a novel clustering algorithm for image segmentation by employing the arbitrary covariance matrices that uses PSO for the estimation of Gaussian mixture models.

# **References**

Ari, C., & Aksoy, S. (2010). Maximum Likelihood Estimation of Gaussian Mixture Models Using Particle Swarm Optimization. *The 20th International Conference on Pattern Recognition (ICPR 2010)* (pp. 746-749). Istanbul: IEEE. Retrieved February 21, 2018, from www.cs.bilkent.edu.tr/~saksoy/papers/icpr10\_clustering.pdf

Bilmes, J. A. (1998). *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models.* International Computer Science Institute, Department of Electrical Engineering and Computer Science. Berkeley: University of Washington. Retrieved from http://melodi.ee.washington.edu/people/bilmes/mypubs/bilmes1997-em.pdf

Borman, S. (2004). *The Expectation Maximization Algorithm - A short tutorial.* University of Notre Dame, Department of Electrical Engineering. South Bend, Indiana: Sean Borman's Home Page.

Burden, R. L., & Faires, D. J. (2011). *Numerical Analysis* (9th Edition ed.). (M. Julet, Ed.) Brooks/Cole Cengage Learning.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. (M. Stone, Ed.) *Journal of the Royal Statistical Society, Series B (Methodological), 39*(1), 1-38.

Dinh, L. T., Pham, D. H., Nguyen, T. X., & Ta, P. D. (2000). *Univariate Analysis - Principles and Practices.* (K. H. Ha, T. V. Ngo, & D. H. Pham, Eds.) Hanoi, Vietnam: Hanoi National University Publisher. Retrieved from http://www.ebook.edu.vn/?page=1.14&view=11156

Golubovic, R. M., Olcan, D. I., & Kolundzija, B. M. (2007). Particle Swarm Optimization Algorithm and Its Modifications Applied to EM Problems. In B. D. Milovanović (Ed.), *The 8th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (ELSIKS 2007)* (pp. 427-430). Nis, Serbia: IEEE. doi:10.1109/TELSKS.2007.4376029

Guo, X., Li, Q.-y., & Xu, W.-l. (2017, February). Acceleration of the EM Algorithm Using the Vector Aitken Method and Its Steffensen Form. *Acta Mathematicae Applicatae Sinica, 33*(1), 175-182. doi:10.1007/s10255-017-0648-3

Hardle, W., & Simar, L. (2013). *Applied Multivariate Statistical Analysis.* Berlin, Germany: Research Data Center, School of Business and Economics, Humboldt University.

Jebara, T. (2015). *The Exponential Family of Distributions.* Columbia University, Computer Science Department. New York: Columbia Machine Learning Lab. Retrieved April 27, 2016, from http://www.cs.columbia.edu/~jebara/4771/tutorials/lecture12.pdf

Jia, Y.-B. (2013). *Lagrange Multipliers.* Lecture notes on course “Problem Solving Techniques for Applied Computer Science”, Iowa State University of Science and Technology, USA.

Lambers, J. (2009). *Accelerating Convergence.* University of Southern Mississippi, Department of Mathematics. Hattiesburg: University of Southern Mississippi. Retrieved February 15, 2018, from http://www.math.usm.edu/lambers/mat460/fall09/lecture13.pdf

Lansky, D., & Casella, G. (1992). Improving the EM Algorithm. *Computing Science and Statistics*, 420-424. doi:10.1007/978-1-4612-2856-1\_67

McLachlan, G., & Krishnan, T. (1997). *The EM Algorithm and Extensions.* New York, NY, USA: John Wiley & Sons. Retrieved from https://books.google.com.vn/books?id=NBawzaWoWa8C

Meng, X.-L., & Rubin, D. B. (1993, June 1). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika, 80*(2), 267-278. doi:10.2307/2337198

Montgomery, D. C., & Runger, G. C. (2003). *Applied Statistics and Probability for Engineers* (3rd Edition ed.). New York, NY, USA: John Wiley & Sons, Inc.

Nguyen, L. (2015). *Matrix Analysis and Calculus* (1st ed.). (C. Evans, Ed.) Hanoi, Vietnam: Lambert Academic Publishing. Retrieved from https://www.shuyuan.sg/store/gb/book/matrix-analysis-and-calculus/isbn/978-3-659-69400-4

Poli, R., Kennedy, J., & Blackwell, T. (2007, June). Particle swarm optimization. (M. Dorigo, Ed.) *Swarm Intelligence, 1*(1), 33-57. doi:10.1007/s11721-007-0002-0

Rai, S. N., & Matthews, D. E. (1993, June). Improving the EM Algorithm. (C. A. McGilchrist, Ed.) *Biometrics, 49*(2), 587-591. doi:10.2307/2532570

Rajeswari, J., & Gunasundari, R. (2016, December). EMPWC: Expectation Maximization with Particle Swarm Optimization based Weighted Clustering for Outlier Detection in Large Scale Data. (C.-H. Lien, & T.-L. Liao, Eds.) *International Journal of Control Theory and Applications (IJCTA), 9*(36), 517-531. Retrieved February 21, 2018, from http://serialsjournals.com/articlesview.php?volumesno\_id=1131&article\_id=14367&volumes\_id=848&journals\_id=268

Rao, R. C. (1955, June). Estimation and tests of significance in factor analysis. *Psychometrika, 20*(2), 93-111. doi:10.1007/BF02288983

Rosen, K. H. (2012). *Discrete Mathematics and Its Applications* (7nd Edition ed.). (M. Lange, Ed.) McGraw-Hill Companies.

Sean, B. (2009). *The Expectation Maximization Algorithm - A short tutorial.* University of Notre Dame, Indiana, Department of Electrical Engineering. Sean Borman's Homepage.

StackExchange. (2013, November 19). *Eigenvalues of the product of 2 symmetric matrices*. (Stack Exchange Network) Retrieved February 9, 2018, from Mathematics StackExchange: https://math.stackexchange.com/questions/573583/eigenvalues-of-the-product-of-2-symmetric-matrices

Ta, P. D. (2014). *Numerical Analysis Lecture Notes.* Vietnam Institute of Mathematics, Numerical Analysis and Scientific Computing. Hanoi: Vietnam Institute of Mathematics. Retrieved 2014

Tang, Z., Song, Y.-Q., & Liu, Z. (2014). Medical Image Clustering Based on Improved Particle Swarm Optimization and Expectation Maximization Algorithm. *The 6th Chinese Conference on Pattern Recognition (CCPR 2014).* *II*, pp. 360-371. Changsha, China: Springer. doi:10.1007/978-3-662-45643-9\_38

Tran, A.-K., Vo, Q.-N., & Lee, G. (2013). Maximum Likelihood Estimation of Gaussian Mixture Models Using PSO for Image Segmentation. In J. Chen, A. Cuzzocrea, & L. T. Yang (Ed.), *The 2013 IEEE 16th International Conference on Computational Science and Engineering (CSE 2013)* (pp. 501-507). Sydney, NSW, Australia: IEEE. doi:10.1109/CSE.2013.81

Wikipedia. (2014, August 4). *Karush–Kuhn–Tucker conditions*. (Wikimedia Foundation) Retrieved November 16, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/Karush–Kuhn–Tucker\_conditions

Wikipedia. (2014, October 10). *Set (mathematics)*. (A. Rubin, Editor, & Wikimedia Foundation) Retrieved October 11, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/Set\_(mathematics)

Wikipedia. (2016, March September). *Exponential family*. (Wikimedia Foundation) Retrieved 2015, from Wikipedia website: https://en.wikipedia.org/wiki/Exponential\_family

Wikipedia. (2017, May 25). *Aitken's delta-squared process*. (Wikimedia Foundation) Retrieved February 15, 2018, from Wikipedia website: https://en.wikipedia.org/wiki/Aitken%27s\_delta-squared\_process

Wikipedia. (2017, February 27). *Commuting matrices*. (Wikimedia Foundation) Retrieved February 9, 2018, from Wikipedia website: https://en.wikipedia.org/wiki/Commuting\_matrices

Wikipedia. (2017, November 27). *Diagonalizable matrix*. (Wikimedia Foundation) Retrieved February 10, 2018, from Wikipedia website: https://en.wikipedia.org/wiki/Diagonalizable\_matrix#Simultaneous\_diagonalization

Wikipedia. (2017, March 2). *Maximum a posteriori estimation*. (Wikimedia Foundation) Retrieved April 15, 2017, from Wikipedia website: https://en.wikipedia.org/wiki/Maximum\_a\_posteriori\_estimation

Wikipedia. (2017, March 7). *Particle swarm optimization*. (Wikimedia Foundation) Retrieved April 8, 2017, from Wikipedia website: https://en.wikipedia.org/wiki/Particle\_swarm\_optimization

Wikipedia. (2017, May 8). *Wolfe conditions*. (Wikimedia Foundation) Retrieved February 20, 2018, from Wikipedia website: https://en.wikipedia.org/wiki/Wolfe\_conditions

Wikipedia. (2018, January 15). *Conjugate prior*. (Wikimedia Foundation) Retrieved February 15, 2018, from Wikipedia website: https://en.wikipedia.org/wiki/Conjugate\_prior

Wikipedia. (2018, January 28). *Gradient descent*. (Wikimedia Foundation) Retrieved February 20, 2018, from Wikipedia website: https://en.wikipedia.org/wiki/Gradient\_descent

Wikipedia. (2018, February 17). *Lipschitz continuity*. (Wikimedia Foundation) Retrieved February 20, 2018, from Wikipedia website: https://en.wikipedia.org/wiki/Lipschitz\_continuity

Wikipedia. (2018, January 7). *Symmetry of second derivatives*. (Wikimedia Foundation) Retrieved February 10, 2018, from Wikipedia website: https://en.wikipedia.org/wiki/Symmetry\_of\_second\_derivatives

Wu, J. C. (1983, March). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics, 11*(1), 95-103. Retrieved from https://projecteuclid.org/euclid.aos/1176346060

Zhang, B., Zhuang, L., Gao, L., Luo, W., Ran, Q., & Du, Q. (2014, May 14). PSO-EM: A Hyperspectral Unmixing Algorithm Based On Normal Compositional Model. (A. Plaza, Ed.) *IEEE Transactions on Geoscience and Remote Sensing, 52*(12), 7782 - 7792. doi:10.1109/TGRS.2014.2319337

Zivot, E. (2009). *Maximum Likelihood Estimation.* Lecture Notes on course "Econometric Theory I: Estimation and Inference (first quarter, second year PhD)", University of Washington, Seattle, Washington, USA.

**Contents**

[Abstract 1](#_Toc51165771)

[1. Introduction 1](#_Toc51165772)

[2. EM algorithm 24](#_Toc51165773)

[3. Convergence of EM algorithm 35](#_Toc51165774)

[4. Variants of EM algorithm 50](#_Toc51165775)

[4.1. EM algorithm with prior probability 50](#_Toc51165776)

[4.2. EM algorithm with Newton-Raphson method 52](#_Toc51165777)

[4.3. EM algorithm with Aitken acceleration 57](#_Toc51165778)

[4.4. ECM algorithm 58](#_Toc51165779)

[5. Applications of EM 59](#_Toc51165780)

[5.1. Mixture model and EM 59](#_Toc51165781)

[6. Discussions 66](#_Toc51165782)

[References 68](#_Toc51165783)