Multivariate "Diagonal" Mixture Models

Serge Iovleff

September 15, 2014

Abstract

This document resume the different "diagonal" mixture models that are, or will be, implemented in the stk++ Clustering project and used in the MixtComp project.

1 A Short Course on Mixture Modeling

Let $\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ be n independent vectors in $\mathbb{R}^d$ such that each $\mathbf{x}_i$ arises from a probability distribution with density

$$f(\mathbf{x}_i|\theta) = \sum_{k=1}^{K} p_k h(\mathbf{x}_i|\boldsymbol{\lambda}_k, \boldsymbol{\alpha}) \tag{1}$$

where the p_k's are the mixing proportions ($0 < p_k < 1$ for all $k = 1, ..., K$ and $p_1 + ... + p_K = 1$), $h(\cdot|\boldsymbol{\lambda}_k, \boldsymbol{\alpha})$ denotes a d-dimensional distribution parameterized by $\boldsymbol{\lambda}_k$ and $\boldsymbol{\alpha}$. The parameters $\boldsymbol{\alpha}$ do not depend from k and is common to all the components of the mixture. The vector parameter to be estimated is $\theta = (p_1, \ldots, p_K, \boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_K, \boldsymbol{\alpha})$ and is chosen to maximize the observed log-likelihood

$$L(\theta|\mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \ln \left(\sum_{k=1}^{K} p_k h(\mathbf{x}_i, \boldsymbol{\lambda}_k, \boldsymbol{\alpha}) \right). \tag{2}$$

It is well known that for a mixture distribution, a sample of indicator vectors or *labels* $\mathbf{z} = \{\mathbf{z}_1, ..., \mathbf{z}_n\}$, with $\mathbf{z}_i = (z_{i1}, \ldots, z_{iK})$, $z_{ik} = 1$ or 0, according to the fact that $\mathbf{x}_i$ is arising from the kth mixture component or not, is associated to the observed data $\mathbf{x}$. The sample $\mathbf{z}$ is *unknown* so that the maximum likelihood estimation of mixture models is performed via the EM algorithm [Dempster et al., 1997] or by a stochastic version of EM called SEM (see [Mclachlan and Peel, 2000]), or by a k-means like algorithm called CEM.

1.1 The algorithms

1.1.1 The EM algorithm

Starting from an initial arbitrary parameter θ^0, the mth iteration of the EM algorithm consists of repeating the following E and M steps.

- **E step:** The current conditional probabilities that $z_{ik} = 1$ for $i = 1, \ldots, n$ and $k = 1, \ldots, K$ are computed using the current value θ^{m-1} of the parameter:

$$t_{ik}^m = t_k^m(\mathbf{x}_i|\theta^{m-1}) = \frac{p_k^{m-1} h(\mathbf{x}_i|\boldsymbol{\lambda}_k^{m-1}, \boldsymbol{\alpha}^{m-1})}{\sum_{l=1}^K p_l^{m-1} h(\mathbf{x}_i|\boldsymbol{\lambda}_l^{m-1}, \boldsymbol{\alpha}^{m-1})}. \tag{3}$$

- **M step:** The m.l. estimate θ^m of θ is updated using the conditional probabilities t_{ik}^m as conditional mixing weights. It leads to maximize

$$L(\theta|\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{t}^m) = \sum_{i=1}^n \sum_{k=1}^K t_{ik}^m \ln\left[p_k h(\mathbf{x}_i|\boldsymbol{\lambda}_k, \boldsymbol{\alpha})\right], \tag{4}$$

  where $\mathbf{t}^m = (t_{ik}^m, i = 1, \ldots, n, k = 1, \ldots, K)$. Updated expression of mixture proportions are, for $k = 1, \ldots, K$,

$$p_k^m = \frac{\sum_{i=1}^n t_{ik}^m}{n}. \tag{5}$$

  Detailed formula for the updating of the $\boldsymbol{\lambda}_k$'s and $\boldsymbol{\alpha}$ are depending of the component parameterization and will be detailed in the next section.

The EM algorithm may converge to a local maximum of the observed data likelihood function, depending on starting values.

1.1.2 The SEM algorithm

The SEM algorithm is a stochastic version of EM incorporating between the E and M steps a restoration of the unknown component labels $\mathbf{z}_i$, $i = 1, \ldots, n$, by drawing them at random from their current conditional distribution. Starting from an initial parameter θ^0, an iteration of SEM consists of three steps.

- **E step:** The conditional probabilities t_{ik}^m $(1 \leq i \leq n, 1 \leq k \leq K)$ are computed for the current value of θ as done in the E step of EM.
- **S step:** Generate labels $\mathbf{z}^m = \{\mathbf{z}_1^m, ..., \mathbf{z}_n^m\}$ by assigning each point $\mathbf{x}_i$ at random to one of the mixture components according to the categorical distribution with parameter $(t_{ik}^m, 1 \leq k \leq K)$.
- **M step:** The m.l. estimate of θ is updated using the generated labels by maximizing

$$L(\theta|\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{t}^m) = \sum_{i=1}^n \sum_{k=1}^K z_{ik}^m \ln\left[p_k h(\mathbf{x}_i|\boldsymbol{\lambda}_k, \boldsymbol{\alpha})\right], \tag{6}$$

SEM does not converge point wise. It generates a Markov chain whose stationary distribution is more or less concentrated around the m.l. parameter estimator. A natural parameter estimate from a SEM sequence $(\theta^r)_{r=1,\ldots,R}$ is the mean $\sum_{r=b+1}^R \theta^r/(R-b)$ of the iterates values where the first b burn-in iterates have been discarded when computing this mean. An alternative estimate is to consider the parameter value leading to the highest likelihood in a SEM sequence.

1.1.3 The CEM algorithm

This algorithm incorporates a classification step between the E and M steps of EM. Starting from an initial parameter θ^0, an iteration of CEM consists of three steps.

- **E step:** The conditional probabilities t_{ik}^m $(1 \leq i \leq n, 1 \leq k \leq K)$ are computed for the current value of θ as done in the E step of EM.
- **C step:** Generate labels $\mathbf{z}^m = \{\mathbf{z}_1^m, ..., \mathbf{z}_n^m\}$ by assigning each point $\mathbf{x}_i$ to the component maximizing the conditional probability $(t_{ik}^m, 1 \leq k \leq K)$.
- **M step:** The m.l. estimate of θ are computed as done in the M step of SEM.

CEM is a *K-means*-like algorithm and contrary to EM, it converges in a finite number of iterations. CEM is not maximizing the observed log-likelihood L (2) but is maximizing in θ and $\mathbf{z}_1, \ldots, \mathbf{z}_n$ the complete data log-likelihood

$$CL(\theta, \mathbf{z}_1, \ldots, \mathbf{z}_n | \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \ln[p_k h(\mathbf{x}_i | \boldsymbol{\lambda}_k)]. \tag{7}$$

where the missing component indicator vector $\mathbf{z}_i$ of each sample point is included in the data set. As a consequence, CEM is not expected to converge to the m.l. estimate of θ and yields inconsistent estimates of the parameters especially when the mixture components are overlapping or are in disparate proportions (see [Mclachlan and Peel, 2000], Section 2.21).

1.2 Strategies

1.2.1 SEM Strategies with missing values

The SEM algorithm is well suited for estimating mixture models in the context of missing values. Let use denote by $\tilde{\mathbf{x}}$ the set of missing values among the observed data $\mathbf{x}$. The strategy to use in this case is a two steps strategy. In the first step, the parameters $\boldsymbol{\theta}$ are estimated. In the second step, the missing values $(\tilde{\mathbf{x}}, \mathbf{z})$ are estimated using a Monte-Carlo algorithm.

Step 1 Estimate the parameters $\boldsymbol{\theta}$ by generating a sequence $(\boldsymbol{\theta}^t, \tilde{\mathbf{x}}^t, \mathbf{z}^t)_{t=1}^{B+T}$ using the SEM algorithm (see section 1.1.2). The method is the following

- perform B "burning" SEM-steps and discard the values generated,
- perform T SEM-steps and store the sequence $(\boldsymbol{\theta}^t)_{t=B+1}^{T+B}$ generated,
- Compute the maximum likelihood estimates of $\boldsymbol{\theta}$ using
  1. for the continuous parameters the mean over the iterations (and additionally give the standard-deviation)
  2. for the categorical (discrete) parameters the mode of the generated values (and additionally, give the most frequent other values).

Step 2 Estimate the missing values $((\tilde{\mathbf{x}}, \mathbf{z}))$ by generating a sequence $(\tilde{\mathbf{x}}^t, \mathbf{z}^t)_{t=1}^{B'+T}$ using a Monte-Carlo algorithm :

- perform B' "burning" S-steps and discard the values generated,
- perform T' S-steps and store the sequence $(\tilde{\mathbf{x}}^t, \mathbf{z}^t)_{t=B'+1}^{T'+B'}$ generated,
- compute the estimates of $\boldsymbol{\theta}$ using
  1. for the continuous missing values the mean over the iterations (and additionally give the standard-deviation)
  2. for the categorical (discrete) missing values the mode of the generated values (and additionally, give the most frequent other values).

2 Multivariate Gaussian Mixture Models

A Gaussian density on $\mathbb{R}_+$ is a density of the form:

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \quad \sigma > 0. \tag{8}$$

A joint Gaussian density on ${\mathbb{R}^d}_+$ is a density of the form:

$$h(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}) = \prod_{j=1}^{d} f(x^j; \mu^j, \sigma^j) \quad \sigma^j > 0. \tag{9}$$

The parameters $\boldsymbol{\mu} = (\mu^1, \ldots, \mu^d)$ are the position parameters and the parameters $\boldsymbol{\sigma} = (\sigma^1, \ldots, \sigma^d)$ are the standard-deviation parameters. Assumptions on the position and standard-deviation parameters among the variables and the components lead to define four families of mixture model.

Let us write a multidimensional Gaussian mixture model in the from `gaussian_s*` with `s*`, the different ways to parametrize the standard-deviation parameters of a Gaussian mixture:

- `sjk` means that we have one standard-deviation parameter for each variable and for each component,
- `sk` means that the standard-deviation parameters are the same for all the variables inside a component,
- `sj` means that the standard-deviation parameters are different for each variable but are equals between the components,
- and finally `s` means that the standard-deviation parameters are all equals.

The `gaussian_sjk` model is the most general model and have a density function of the form

$$f(\mathbf{x}_i|\theta) = \sum_{k=1}^{K} p_k \prod_{j=1}^{d} g(x_i^j|\mu_k^j, \sigma_k^j). \tag{10}$$

3 Multivariate categorical Mixture Models

A Categorical probability distribution on a finite space $\mathcal{C} = \{\infty, \ldots, \mathcal{L}\}$ is a probability distribution of the form:

$$P(x = l) = p_l \quad p_l > 0,\, l \in \mathcal{C}, \tag{11}$$

with the constraint $p_1 + \ldots + p_L = 1$.

A joint Categorical probability distribution on $\mathcal{X}^d$ is a probability distribution of the form:

$$P(\mathbf{x} = (x_1, \ldots, x_d)) = \prod_{j=1}^{d} p^j_{x_j} \tag{12}$$

The parameters $\boldsymbol{\mu} = (\mu^1, \ldots, \mu^d)$ are the position parameters and the parameters $\boldsymbol{\sigma} = (\sigma^1, \ldots, \sigma^d)$ are the standard-deviation parameters. Assumptions on the position and standard-deviation parameters among the variables and the components lead to define four families of mixture model.

4 Multivariate Gamma Mixture Models

A gamma density on $\mathbb{R}_+$ is a density of the form:

$$g(x; a, b) = \frac{(x)^{a-1}\, e^{-x/b}}{\Gamma(a)\,(b)^a} \quad a > 0, \quad b > 0. \tag{13}$$

A joint gamma density on $\mathbb{R}^d{}_+$ is a density of the form:

$$h(\mathbf{x}; \mathbf{a}, \mathbf{b}) = \prod_{j=1}^{d} g(x^j; a^j, b^j) \quad a^j > 0, \quad b^j > 0. \tag{14}$$

The parameters $\mathbf{a} = (a^1, \ldots, a^d)$ are the shape parameters and the parameters $\mathbf{b} = (b^1, \ldots, b^d)$ are the scale parameters. Assumptions on the scale and shape parameters among the variables and the components lead to define twelve families of mixture model. Let us write a multidimensional gamma mixture model in the from `gamma_a*_b*` with `a*` (resp. `b*`), the different ways to parametrize the shape (resp. scale) parameters of a gamma mixture:

- `ajk` (resp. `bjk`) means that we have one shape (resp. scale) parameter for each variable and for each component,
- `ak` (resp. `bk`) means that the shape (resp. scale) parameters are the same for all the variables inside a component,
- `aj` (resp. `bj`) means that the shape (resp. scale) parameters are different for each variable but are equals between the components,
- and finally `a` (resp. `b`) means that the shape (resp. scale) parameters are the same for all the variables and all the components.

	ajk	ak	aj	a
bjk	`gamma_ajk_bjk` (2dK)	`gamma_ak_bjk` (dK + K)	`gamma_aj_bjk` (dK+d)	`gamma_a_bjk` (dK+1)
bk	`gamma_ajk_bk` (dK+K)	`gamma_ak_bk` (2K)	`gamma_aj_bk` (K+d)	`gamma_a_bk` (K+1)
bj	`gamma_ajk_bj` (dK+d)	`gamma_ak_bj` (K+d)	NA	NA
b	`gamma_ajk_b` (dK+1)	`gamma_ak_b` (K+1)	NA	NA

Table 1: The twelve multidimensional gamma mixture models. In parenthesis the number of parameters of each model.

The models we can build in this way are summarized in the table 1, in parenthesis we give the number of parameters of each models. The tested and implemented model we present in this article are in the gray cells.

The `gamma_ajk_bjk` model is the most general and have a density function of the form

$$f(\mathbf{x}_i|\theta) = \sum_{k=1}^{K} p_k \prod_{j=1}^{d} g(x_i^j|a_k^j, b_k^j). \tag{15}$$

All the other models can be derived from this model by dropping the indexes in j and/or k from the expression (15). For example the mixture model `gamma_aj_bk` have a density function of the form

$$f(\mathbf{x}_i|\theta) = \sum_{k=1}^{K} p_k \prod_{j=1}^{d} g(x_i^j|a^j, b^k). \tag{16}$$

5 Multivariate Beta Mixture Models

A beta density on $(0, 1)$ is a density of the form:

$$b(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha, \beta)} \quad \alpha > 0, \quad \beta > 0. \tag{17}$$

A joint beta density on $(0, 1)^d$ is a density of the form:

$$h(\mathbf{x}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{j=1}^{d} b(x^j; \alpha^j, \beta^j) \quad \alpha^j > 0, \quad \beta^j > 0. \tag{18}$$

The parameters $\boldsymbol{\alpha} = (\alpha^1, \ldots, \alpha^d)$ and the parameters $\boldsymbol{\beta} = (\beta^1, \ldots, \beta^d)$ are the shape parameters. Assumptions on these parameters among the variables and the components lead to define twelve families of mixture models. Let us write a multidimensional beta mixture model in the form `beta_a*_b*` with `a*` (resp. `b*`). The models we can build are summarized in the table 2 below

The `b_ajk_bjk` model is the most general and has a density function of the form

$$f(\mathbf{x}_i|\theta) = \sum_{k=1}^{K} p_k \prod_{j=1}^{d} b(x_i^j|\alpha_k^j, \beta_k^j). \tag{19}$$

	ajk	ak	aj	a
bjk	`b_ajk_bjk`	`b_ak_bjk`	`b_aj_bjk`	`b_a_bjk`
bk	`b_ajk_bk`	`b_ak_bk`	`b_aj_bk`	`b_a_bk`
bj	`b_ajk_bj`	`b_ak_bj`	NA	NA
b	`b_ajk_b`	`b_ak_b`	NA	NA

Table 2: The twelve multidimensional beta mixture models

All the other models can be derived from this model by dropping the indexes in j and/or k from the expression (19). For example the mixture model `gamma_aj_bk` has a density function of the form

$$f(\mathbf{x}_i|\theta) = \sum_{k=1}^{K} p_k \prod_{j=1}^{d} b(x_i^j|\alpha^j, \beta^k). \tag{20}$$

References

[Brent, 1973] Brent, R. P. (1973). Some efficient algorithms for solving systems of nonlinear equations. 10(2):327–344.

[Dempster et al., 1997] Dempster, A., Laird, N., and Rubin, D. (1997). Maximum likelihood from incomplete data with the em algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39:1.

[Mclachlan and Peel, 2000] Mclachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley-Interscience, 1 edition.

A M step computation for the Gaussian models

For all the M Step, the mean is updated using the following formula

$$\boldsymbol{\mu}_k = \frac{1}{t_{.k}} \sum_{i=1}^{n} t_{ik} \mathbf{x}_i,$$

with $t_{.k} = \sum_{i=1}^{n} t_{ik}$, for $k = 1, \ldots, K$.

A.1 M Step of the gaussian_sjk model

Using the equation (4) and dropping the constant, we obtain that we have to maximize in $\boldsymbol{\sigma} = (\sigma_k^j)^2$, for $j = 1, \ldots, d$ and $k = 1, \ldots, K$ the expression

$$l(\boldsymbol{\sigma}|\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{t}^m) = \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik}^m \sum_{j=1}^{d} \left[-\frac{1}{(\sigma_k^j)^2} (x_i^j - \hat{\mu}_j^k)^2 - \log((\sigma_k^j)^2) \right]. \tag{21}$$

For this model, the variance is updated using the formula:

$$(\hat{\sigma}_k^j)^2 = \frac{1}{t_{.k}} \sum_{i=1}^{n} t_{ik} (x_i^j - \hat{\mu}_k^j)^2.$$

A.2 M Step of the gaussian_sk model

For this model, the variance is updated using the formula:

$$(\hat{\sigma}_k)^2 = \frac{1}{dt_{.k}} \sum_{j=1}^{d} \sum_{i=1}^{n} t_{ik}(x_i^j - \hat{\mu}_k^j)^2.$$

A.3 M Step of the gaussian_sj model

For this model, the variance is updated using the formula:

$$(\hat{\sigma}^j)^2 = t_{ik}(x_i^j - \mu_k^j)^2.$$

A.4 M Step of the gaussian_s model

For this model, the variance is updated using the formula:

$$\hat{\sigma}^2 = \frac{1}{nd} \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2.$$

B M step computation for the Gamma models

In this section, given the array $\mathbf{t}$ of conditional probabilities, we will write $t_{.k} = \sum_{i=1}^{n} t_{ik}$, for $k = 1, \ldots, K$ and will denote

$$\bar{x}_k^j = \frac{1}{t_{.k}} \sum_{i=1}^{n} t_{ik} x_i^j,$$

the k-th pondered mean of the j-th observation, and by

$$(\overline{\log(x)})_k^j = \frac{1}{t_{.k}} \sum_{i=1}^{n} t_{ik} \log(x_i^j),$$

the k-th pondered log-mean of the j-th observation.

Replacing h by its expression in the equation (4) and summing in i, the M-step for the twelve gamma mixture models defined in table (1) is equivalent to maximize the following expression

$$l(A, B) = \sum_{k=1}^{K} \sum_{j=1}^{d} t_{.k} \left(A(\overline{\log(x)})_k^j - \frac{\bar{x}_k^j}{B} - \log(\Gamma(A)) - A \log(B) \right), \qquad (22)$$

with $A \in \{a, a^j, a_k, a_k^j\}$ and $B \in \{b, b^j, b_k, b_k^j\}$.

We now describe the various derivatives and for each models explicit the maximum likelihood equations to solve. Taking the derivative with respect to B:

- If $B = b_k^j$ then

$$\frac{dl}{db_k^j} = t_{.k} \left(\frac{\bar{x}_k^j}{b^2} - \frac{A}{b} \right) \text{ and thus } \hat{b}_k^j = \frac{\bar{x}_k^j}{A}$$

- If $B = b_k$ then

$$\frac{dl}{db_k} = t_{.k} \sum_{j=1}^{d} \left(\frac{\bar{x}_k^j}{b_k^2} - \frac{A}{b_k} \right) \text{ and thus } \hat{b}_k = \frac{\sum_{j=1}^{d} \bar{x}_k^j}{\sum_{j=1}^{d} A}$$

- If $B = b^j$ then

$$\frac{dl}{db^j} = \sum_{k=1}^{K} t_{.k} \left(\frac{\bar{x}_k^j}{(b^j)^2} - \frac{A}{b^j} \right) \text{ and thus } \hat{b}^j = \frac{\sum_{k=1}^{K} t_{.k} \bar{x}_k^j}{\sum_{k=1}^{K} t_{.k} A}$$

- If $B = b$ then

$$\frac{dl}{db} = \sum_{k=1}^{K} \sum_{j=1}^{d} t_{.k} \left(\frac{\bar{x}_k^j}{b^2} - \frac{A}{b} \right) \text{ and thus } \hat{b} = \frac{\sum_{k=1}^{K} \sum_{j=1}^{d} t_{.k} \bar{x}_k^j}{\sum_{k=1}^{K} \sum_{j=1}^{d} t_{.k} A}$$

Taking now the derivative with respect to A:

1. If $A = a_k^j$, then

$$\frac{dl}{da_k^j} = t_{.k} \left((\overline{\log(x)})_k^j - \log(B) \right) - t_{.k} \Psi(a_k^j).$$

and thus

- if $B = b_k^j$ (model `gamma_ajk_bjk`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}_k^j) & = & (\overline{\log(x)})_k^j - \log(\hat{b}_k^j) \\ \hat{b}_k^j & = & \frac{\bar{x}_k^j}{\hat{a}_k^j}, \end{array} \right. \tag{23}$$

- if $B = b_k$ (model `gamma_ajk_bk`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}_k^j) & = & (\overline{\log(x)})_k^j - \log(\hat{b}_k) \\ \hat{b}_k & = & \frac{\sum_{j=1}^{d} \bar{x}_k^j}{\sum_{j=1}^{d} a_k^j} \end{array} \right. \tag{24}$$

- if $B = b^j$ (model `gamma_ajk_bj`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}_k^j) & = & (\overline{\log(x)})_k^j - \log(\hat{b}^j) \\ \hat{b}^j & = & \frac{\sum_{k=1}^{K} t_{.k} \bar{x}_k^j}{\sum_{k=1}^{K} t_{.k} a_k^j} \end{array} \right. \tag{25}$$

- if $B = b$ (model `gamma_ajk_b`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}_k^j) & = & (\overline{\log(x)})_k^j - \log(\hat{b}) \\ \hat{b} & = & \frac{\sum_{j=1}^{d} \sum_{k=1}^{K} t_{.k} \bar{x}_k^j}{\sum_{j=1}^{d} \sum_{k=1}^{K} t_{.k} a_k^j} \end{array} \right. \tag{26}$$

2. If $A = a_k$, then

$$\frac{dl}{da_k} = t_{.k} \sum_{j=1}^{d} \left((\overline{\log(x)})_k^j - \log(B) \right) - t_{.k} d\Psi(a_k).$$

and thus

- if $B = b_k^j$ (model `gamma_ak_bjk`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}_k) & = & \frac{1}{d}\sum_{j=1}^{d}\left((\overline{\log(x)})_k^j - \log(\hat{b}_k^j)\right) \\ \hat{b}_k^j & = & \frac{\bar{x}_k^j}{\hat{a}_k}, \end{array} \right. \tag{27}$$

- if $B = b_k$ (model `gamma_ak_bk`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}_k) & = & \frac{1}{d}\sum_{j=1}^{d}\left((\overline{\log(x)})_k^j - \log(\hat{b}_k)\right) \\ \hat{b}_k & = & \frac{\sum_{j=1}^{d}\bar{x}_k^j}{da_k} \end{array} \right. \tag{28}$$

- if $B = b^j$ (model `gamma_ak_bj`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}_k) & = & \frac{1}{d}\sum_{j=1}^{d}\left((\overline{\log(x)})_k^j - \log(\hat{b}^j)\right) \\ \hat{b}^j & = & \frac{\sum_{k=1}^{K} t_{.k}\bar{x}_k^j}{\sum_{k=1}^{K} t_{.k}a_k} \end{array} \right. \tag{29}$$

- if $B = b$ (model `gamma_ak_b`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}_k) & = & \frac{1}{d}\sum_{j=1}^{d}\left((\overline{\log(x)})_k^j - \log(\hat{b})\right) \\ \hat{b} & = & \frac{\sum_{j=1}^{d}\sum_{k=1}^{K} t_{.k}\bar{x}_k^j}{d\sum_{k=1}^{K} t_{.k}a_k} \end{array} \right. \tag{30}$$

3. If $A = a^j$, then

$$\frac{dl}{da^j} = \sum_{k=1}^{K} t_{.k}\left((\overline{\log(x)})_k^j - \log(B)\right) - n\Psi(a^j).$$

and thus

- if $B = b_k^j$ (model `gamma_aj_bjk`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}^j) & = & \frac{1}{n}\sum_{k=1}^{K} t_{.k}\left((\overline{\log(x)})_k^j - \log(\hat{b}_k^j)\right) \\ \hat{b}_k^j & = & \frac{\bar{x}_k^j}{\hat{a}^j}, \end{array} \right. \tag{31}$$

- if $B = b_k$ (model `gamma_aj_bk`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}^j) & = & \frac{1}{n}\sum_{k=1}^{K} t_{.k}\left((\overline{\log(x)})_k^j - \log(\hat{b}_k)\right) \\ \hat{b}_k & = & \frac{\sum_{j=1}^{d}\bar{x}_k^j}{\sum_{j=1}^{d}\hat{a}^j} \end{array} \right. \tag{32}$$

4. If $A = a$, then

$$\frac{dl}{da} = \sum_{k=1}^{K}\sum_{j=1}^{d} t_{.k}\left((\overline{\log(x)})_k^j - \log(B)\right) - nd\Psi(a).$$

and thus

- if $B = b_k^j$ (model `gamma_a_bjk`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}) & = & \frac{1}{nd} \sum_{j=1}^{d} \sum_{k=1}^{K} t_{.k} \left((\overline{\log(x)})_k^j - \log(\hat{b}_k^j) \right) \\ \hat{b}_k^j & = & \frac{\bar{x}_k^j}{\hat{a}}, \end{array} \right. \tag{33}$$

- if $B = b_k$ (model `gamma_a_bk`)

$$\left\{ \begin{array}{lcl} \Psi(\hat{a}) & = & \frac{1}{nd} \sum_{j=1}^{d} \sum_{k=1}^{K} t_{.k} \left((\overline{\log(x)})_k^j - \log(\hat{b}_k) \right) \\ \hat{b}_k & = & \frac{\sum_{j=1}^{d} \bar{x}_k^j}{d\hat{a}} \end{array} \right. \tag{34}$$

In the next sections, we will describe for some models the way to estimate A and B when $A = a_k^j$.

B.1 First algorithm for the M Step of the gamma models

Among the twelve models, we can find six models from whom it is possible to estimate in a single pass of the Brent's method the value of A and then to estimate the value of B. For example for the `gamma_ajk_bjk` model, using (23) gives $\hat{a}_k^j$ solution in a of the following equation

$$(\overline{\log(x)})_k^j - \Psi(a) - \log(\bar{x}_k^j) + \log(a) = 0 \tag{35}$$

whom solution can be found using Brent's method [Brent, 1973].

Having found the estimator of the a_k^j, the estimator of b_k^j can be computed.

B.2 Second algorithm for the M Step of the gamma models

For the other models we have to iterate in order to find the ML estimators. For example for the `gamma_ajk_bj` model, the set of non-linear equations (25) can be solved using an iterative algorithm:

- **Initialization:** Compute an initial estimator of the $\mathbf{a}_k$, $k = 1, \ldots K$ and $\mathbf{b}$ using moment estimators.
- **Repeat until convergence :**
  - **a step:** For fixed b^j solve for each a_k^j, the equation:

$$\Psi(a) - (\overline{\log(x)})_k^j + \log(b^j) = 0.$$

  - **b step:** Update b^j using equation (25).

This algorithm minimize alternatively the log-likelihood in $\mathbf{a}_k$, $k = 1, \ldots n$ and in $\mathbf{b}$ and converge in few iterations.