BIAS OF STANDARD ERRORS IN LATENT CLASS MODEL APPLICATIONS USING NEWTON-RAPHSON AND EM ALGORITHMS

Liberato Camilleri
Department of Statistics and Operations Research
University of Malta
Msida (MSD 06) Malta
E-mail: liberato.camilleri@um.edu.mt

The EM algorithm is a popular method for computing maximum likelihood estimates. It tends to be numerically stable, reduces execution time compared to other estimation procedures and is easy to implement in latent class models. However, the EM algorithm fails to provide a consistent estimator of the standard errors of maximum likelihood estimates in incomplete data applications. Correct standard errors can be obtained by numerical differentiation. The technique requires computation of a complete-data gradient vector and Hessian matrix, but not those associated with the incomplete data likelihood. Obtaining first and second derivatives numerically is computationally very intensive and execution time may become very expensive when fitting Latent class models using a Newton-type algorithm. When the execution time is too high one is motivated to use the EM algorithm solution to initialize the Newton Raphson algorithm. We also investigate the effect on the execution time when a final Newton-Raphson step follows the EM algorithm after convergence. In this paper we compare the standard errors provided by the EM and Newton-Raphson algorithms for two models and analyze how this bias is affected by the number of parameters in the model fit.

Keywords: EM algorithm; Numerical differentiation; Incomplete data; Proportional odds model; Maximum likelihood estimation; Latent class model.

1. Introduction

A limitation of the EM algorithm is that the estimated information matrix, in contrast to the case for gradient methods such as Newton-Raphson, is not a direct by-product of maximization. Procedures for obtaining the information matrix within the EM algorithm have been suggested by several authors.

An approach for computing the Fisher information matrix within the EM framework was suggested by (Louis 1982). His methodology is based on a result by (Fisher 1925) that showed that, given the incomplete data, incomplete data scores are conditional expectations of the complete data scores. The author derives a procedure for extracting the observed information matrix when the EM algorithm is used to find maximum likelihood estimates in incomplete data problems. The technique requires the computation of the complete data gradient vector and the Hessian matrix but does not require those associated with the incomplete data log- likelihood function. A criticism of this approach is that the procedure is often computationally demanding and hard to implement because it requires the computation of both a complete-data score vector and second derivative matrix.

An alternative approach for computing the Fisher information matrix using gradients only was suggested by (Meilijson 1989). Methods that only require gradients are easier to compute analytically and less demanding to compute numerically. An appealing advantage of this procedure, in contrast to the approach suggested by (Louis 1982), is that once the individual scores have been identified there is no additional analysis to perform. Meilijson's methodology is based on a result by (Fisher 1925) in which the evaluation of individual score vectors of the incomplete data is a by-product of the application of the E-step of the EM algorithm. The Fisher information matrix may be consistently estimated by the empirical variance-covariance matrix of these individual score vectors and the M step may be replaced by a Newton-type step. This permits a unification of EM methodology and Newton methods. A demerit of Meilijson's technique is that it applies only to specialized cases in which the observed data are independent and identically distributed samples.

Another approach for computing the observed information matrix is the well-known supplemented EM (SEM) algorithm, suggested by (Meng and Rubin 1991). The SEM algorithm numerically differentiates the EM operator $M(\varphi)$ and uses a result by (Dempster, Laird and Rubin 1977) that relates the Jacobian of $M(\varphi)$ to the Hessian matrix $H(\varphi)$, both evaluated at $\hat{\varphi}$. The authors claim that their algorithm can be applied to any problem to which EM has been applied, assuming that one has access to the complete-data asymptotic variance-covariance matrix. (Segal, Bacchetti and Jewell 1994) point out that the SEM algorithm requires very accurate estimates of $\hat{\varphi}$ and so they can be much more expensive to obtain than the EM estimates. (McCulloch 1998) remarks that for many problems the method of obtaining standard errors using the SEM algorithm can be numerically unstable. (Jamshidian and Jennrich 2000) point out that, algorithms that numerically differentiate $M(\boldsymbol{\varphi})$ may suffer from the error magnification problem when the EM algorithm is slow. The authors remark that algorithms that numerically differentiate the score vector $\mathbf{g}(\boldsymbol{\varphi})$ are appropriate for all maximum likelihood applications and they do not suffer from the error magnification problem.

The variance-covariance matrix can be obtained by other techniques that do not use numerical differentiation. Bootstrapping uses computer intensive resampling and treats a given sample as the population. An empirical probability distribution is constructed from the sample of size *n* in which the probability of each observation is 1/*n*. *K* random samples each of size *n* are drawn with replacement from this empirical distribution where some of the observations in a sample may be duplicated. The EM algorithm is then performed on each sample to calculate the vector of parameters $\hat{\boldsymbol{\varphi}}_k$. Hence a probability distribution is constructed from all the resampled parameter estimates in which the probability of each $\hat{\boldsymbol{\varphi}}_k$ is 1/*K*. This distribution is the bootstrapped estimate of the sampling distribution of $\hat{\boldsymbol{\varphi}}$ which can be used to provide estimates for the standard errors. The primary advantage of bootstrapping is that no assumptions about the shape of the sampling distribution are made. Jackknifing is a different resampling technique in which a single observation is omitted at a time. Thus, each sample consists of *n*-1 observations formed by deleting a different observation from the sample. A jackknifed estimate of the sampling distribution of $\hat{\boldsymbol{\varphi}}$ can be obtained in a similar way to the bootstrap procedure. (Agresti 2002) remarks that bootstrap and jackknife procedures are useful tools for estimating standard errors when samples are small or data is sparse.

2. A General Model

The purpose of this study is to fit latent class models that analyze ordinal categorical responses using both the EM algorithm and a Newton-type algorithm to assess the bias between the standard errors of these two maximization procedures.

A latent class model relates a set of observed multivariate categorical variables to a latent variable which is discrete. Latent class analysis, unlike cluster analysis, uses a model-based approach that combines conventional statistical estimation methods to classical clustering techniques. In this methodology latent classes are defined by the criterion of conditional independence where the observed variables within each segment are statistically independent. The assumption of conditional independence has been widely used in latent class modelling. It is directly analogous to the assumption, in the factor analysis model, that observed variables are conditionally independent given the factors. This implies that the observed correlations between the items are due to the clustered nature of the population, whereas within a cluster, the items are independent.

Let $\boldsymbol{\varphi} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi})$ be the vector comprising the parameters of the latent class model with *K* segments. The n^{th} density function is of the form

$$P(\mathbf{Y}_n = \mathbf{y}_n | \boldsymbol{\varphi}) = \sum_{k=1}^{K} \pi_k . P\left(\mathbf{Y}_n = \mathbf{y}_n | \boldsymbol{\alpha}, \boldsymbol{\beta}_k\right) \qquad (1)$$

π_k are the unconditional probabilities that sum to 1 and represent the proportion of respondents that are allocated to each segment. The marginal or conditional probability $P(y_{jn} = r | \boldsymbol{\alpha}, \boldsymbol{\beta}_k)$ follows the Proportional Odds model suggested by (McCullagh 1980)

$$P\left(y_{jn} = r | \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = F\left(\alpha_r + \mathbf{x}_j^{'}\boldsymbol{\beta}\right) - F\left(\alpha_{r-1} + \mathbf{x}_j^{'}\boldsymbol{\beta}\right) \qquad (2)$$

In this model y_{jn} is a rating response elicited by the n^{th} respondent for the j^{th} item; $\boldsymbol{\alpha}$ is a vector of threshold parameters; $\boldsymbol{\beta}$ is a vector of regression parameters and $\mathbf{x}_j$ are item covariates. The choice of $F(.)$ is the Logistic distribution which leads to the logit link.

The likelihood function of the data set is obtained by taking the product of the *N* density functions.

$$L(\boldsymbol{\varphi}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k . P\left(\mathbf{Y}_n = \mathbf{y}_n | \boldsymbol{\alpha}, \boldsymbol{\beta}_k\right) \qquad (3)$$

The log-likelihood function is given by:

$$l(\boldsymbol{\varphi}) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k . P\left(\mathbf{Y}_n = \mathbf{y}_n | \boldsymbol{\alpha}, \boldsymbol{\beta}_k\right) \qquad (4)$$

Maximum likelihood estimation can be carried out via standard numerical optimization routines such as the Newton Raphson method or alternatively using the EM algorithm. The popularity of the EM algorithm arises from its computational elegance, particularly for latent class models. The idea behind the EM algorithm is to augment the observed data by introducing unobserved data, λ_{nk} indicating whether the n^{th} respondent belongs to the k^{th} segment.

An effective procedure to fit a latent class model with *K* segments is to maximize the expected complete log-likelihood function using the iterative EM algorithm.

$$L\left(\boldsymbol{\varphi} \mid \boldsymbol{\Lambda}\right) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\pi_k . P\left(\mathbf{Y}_n = \mathbf{y}_n | \boldsymbol{\alpha}, \boldsymbol{\beta}_k\right)\right]^{\lambda_{nk}} \qquad (5)$$

The complete log likelihood $l\left(\boldsymbol{\varphi} \mid \boldsymbol{\Lambda}\right)$ is given by:

$$l(\boldsymbol{\varphi} | \boldsymbol{\Lambda}) = \sum_{n=1}^{N} \sum_{k=1}^{K} [\lambda_{nk} . \ln P(\mathbf{Y}_n = \mathbf{y}_n | \boldsymbol{\alpha}, \boldsymbol{\beta}_k) + \lambda_{nk} . \ln(\pi_k)] \qquad (6)$$

The complete log-likelihood function $l(\boldsymbol{\varphi}|\boldsymbol{\Lambda})$ has simpler form compared to $l(\boldsymbol{\varphi})$ given by (4) and the derivatives are easier to compute.

Each iteration is composed of two steps: an E-step and an M-step. In the E-step, $E[l(\boldsymbol{\varphi}|\boldsymbol{\Lambda})]$ is calculated with respect to the conditional distribution of the unobserved data $\boldsymbol{\Lambda} = (\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, ..., \boldsymbol{\lambda}_N)$ given the vector of observed responses $\mathbf{y}_n$ and using the provisional parameter estimates $\boldsymbol{\varphi}$. This is achieved by using Bayes' theorem to estimate λ_{nk}.

$$E\left(\lambda_{nk}\right) = \frac{\pi_k . P\left(\mathbf{Y}_n = \mathbf{y}_n \left| \boldsymbol{\alpha}, \boldsymbol{\beta}_k \right.\right)}{\sum_{k=1}^{K} \pi_k . P\left(\mathbf{Y}_n = \mathbf{y}_n \left| \boldsymbol{\alpha}, \boldsymbol{\beta}_k \right.\right)} = p_{nk} \qquad (7)$$

satisfying the constraint $\sum_{k=1}^{K} p_{nk} = 1$.

In the M-step, $E[l(\boldsymbol{\varphi}|\boldsymbol{\Lambda})]$ is maximized with respect to $\boldsymbol{\varphi}$. This is achieved by replacing λ_{nk} by their expected posterior probabilities p_{nk}. So

$$E[l(\boldsymbol{\varphi}|\boldsymbol{\Lambda})] = \sum_{n=1}^{N}\sum_{k=1}^{K}[p_{nk}.\ln P(\mathbf{Y}_n = \mathbf{y}_n|\boldsymbol{\alpha}, \boldsymbol{\beta}_k) + p_{nk}.\ln(\pi_k)] \qquad (8)$$

The two terms on the right hand side of the expression can be maximized separately. The maximization of $E[l(\boldsymbol{\varphi}|\boldsymbol{\Lambda})]$ with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}_k$ is performed by transforming the polychotomous responses as a vector of 0-1 indicators. This allows the use of Poisson likelihood in model fitting by considering each term of $\sum_1^N \sum_1^K p_{nk}.\ln P(\mathbf{Y}_n = \mathbf{y}_n|\boldsymbol{\alpha}, \boldsymbol{\beta}_k)$ as a weighted Poisson log-likelihood function.

The maximization of $E[l(\boldsymbol{\varphi}|\boldsymbol{\Lambda})]$ with respect to π_k subject to the constraint $\sum_1^K \pi_k = 1$, is obtained by maximizing the augmented function.

$$\sum_{k=1}^{K}\sum_{n=1}^{N} p_{nk} \ln \pi_k - \delta(\sum_{k=1}^{K} \pi_k - 1) \qquad (9)$$

δ is the Lagrange multiplier. Setting the derivative with respect to π_k equal to zero yields

$$\hat{\pi}_k = \frac{1}{N}\sum_{n=1}^{N} p_{nk} \quad \text{for} \quad k = 1, 2, ..., K \qquad (10)$$

Since the probabilities, p_{nk} are unknown then the iterative procedure is initiated by setting random assignment to these probabilities. The algorithm alternately updates the parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\pi}$ and the prior weights, p_{nk} until the process converges.

Maximum likelihood estimation via a Newton-Raphson algorithm uses numerical first and second derivatives of the likelihood function. The algorithm is computationally demanding and time expensive even with few model parameters. The Newton-Raphson algorithm can be derived by considering an approximation of $\partial l(\boldsymbol{\varphi})/\partial\boldsymbol{\varphi}$ using a first order Taylor series expansion around the parameter $\boldsymbol{\varphi}^m$ evaluated at the m^{th} iteration.

$$\frac{\partial l\left(\boldsymbol{\varphi}\right)}{\partial\boldsymbol{\varphi}} \approx \frac{\partial l\left(\boldsymbol{\varphi}^m\right)}{\partial\boldsymbol{\varphi}} + \frac{\partial^2 l\left(\boldsymbol{\varphi}^m\right)}{\partial\boldsymbol{\varphi}\partial\boldsymbol{\varphi}'}\left(\boldsymbol{\varphi} - \boldsymbol{\varphi}^m\right) \qquad (11)$$

Gradient methods are iterative and updated parameters can be evaluated by setting $\partial l\left(\boldsymbol{\varphi}\right)/\partial\boldsymbol{\varphi}$ to zero. Denoting the gradient vector and Hessian matrix by $\mathbf{g}(\boldsymbol{\varphi}^m)$ and $\mathbf{H}(\boldsymbol{\varphi}^m)$, the updated parameters are given by:

$$\boldsymbol{\varphi}^{m+1} = \boldsymbol{\varphi}^m - \mathbf{H}\left(\boldsymbol{\varphi}^m\right)^{-1}\mathbf{g}\left(\boldsymbol{\varphi}^m\right) \qquad (12)$$

If the log-likelihood is quadratic in the parameters, as in the case of linear regression models, the equations can be solved in closed form and maximum likelihood estimates $\hat{\boldsymbol{\varphi}}$ are found in a single iteration.

3. Application

To illustrate the procedure, 310 respondents were asked to rate a number of items (profiles) on an ordinal scale. The items described different combinations of car attributes, namely the brand, price and door feature. We utilized a full profile method of collecting respondent evaluations. Since the study compared 4 brands, 4 price values and 2 door features, a complete design yielded 32 combinations of attribute levels. Presenting respondents with 32 product profiles to assess their worth was not considered feasible because the information overload very often leads to the temptation on the part of the respondent to adopt patterned types of responses. To simplify the task, the chosen design had two blocks of 16 items each. Each respondent was provided with a set of 16 items to compare with random assignment to block. The rating responses had 7 categories where 1 corresponded to 'worst' and 7 corresponded to 'best'.

Two latent class models were fitted using both the EM and Newton Raphson algorithms. In the first model, the linear predictor included brand as a sole main effect. The latent variable, segment, was interacted with each level of brand and the model was estimated with two latent classes, four latent variables and a logit link function. In the second model, the linear predictor included brand and door feature as main effects and the interaction of a quadratic function of price with brand. Each term was interacted with the latent variable, segment. The model was estimated with two latent classes, thirteen latent variables and a logit link function.

The EM algorithm for fitting latent class models, proposed by Dempster et al (1977), is equivalent to iterative fitting of a weighted GLM, where the posterior probabilities are recalculated at each iteration. This model is implemented as a set of GLIM (Generalized linear interactive models) macros. Being a non-linear model, the proportional odds model is accommodated using the OWN model facilities of GLIM. A problem associated with the application of the EM algorithm to latent class models is its convergence to local maxima. It is caused by the likelihood being multimodal, so that the algorithm becomes sensitive to the starting values used. One way of addressing this problem is to perturb the posterior probabilities at each iteration to widen the search for the global maximum. This is achieved by adding to each probability a pseudo-random real value from a uniform distribution in the range [0, 1] multiplied by a scalar. The posterior probabilities are rescaled such that $\sum_{k=1}^{K} p_{nk} = 1$. The scalar is then reduced systematically so that the iterative procedure will finally converge. An alternative way of tackling this problem is to consider several starting values chosen from a wide range of seed numbers. The selected solution is the one that yields the smallest log-likelihood.

The Newton-Raphson algorithm is implemented using the facilities of GLLAMM (Generalized linear latent and mixed models). This software, which accommodates a large class of models including multilevel, item response, structural equation, longitudinal and latent class models, uses numerical first and second derivatives of the log-likelihood function and produce standard errors by maximizing the marginal log-likelihood. GLLAMM software can fit proportional odds models by specifying the `family` to be binomial and the `link` to be ologit. This link corresponds to the logit link functions appropriate for ordinal data. The syntax `nrf` specifies the number of latent variables; the syntax `nip` specifies the number of latent classes (segments) and the syntax `ip(fn)` yields non-centred latent classes. Some of the terms in the GLIM output were intrinsically aliased. In order to get a similar solution using GLLAMM we had to constrain these parameters to zero using the `constraint define` command in GLLAMM.

4. Results of the study

Although the EM algorithm yields maximum likelihood estimates of the parameters it fails to provide standard errors of these parameter estimates as a by-product of the iterative algorithm. On the other hand, a Newton-type algorithm provides correct standard errors; however, there is a computing cost associated with our patience in waiting for an output. It is well known that Newton-type methods require good starting values and a fast convergence is only guaranteed if these starting values are near the solution. Another problem is that obtaining first and second derivatives numerically is computationally intensive and a Newton-type algorithm may become very expensive particularly when fitting models with a considerable number of parameters. This paper compares the standard errors of the parameters provided by the EM and Newton-Raphson algorithms for the two models and contrast execution times when using the two estimation methods.

It was noted that estimation with GLLAMM using a Newton-type algorithm took about fifty times longer compared to GLIM using an EM algorithm. For problems with large numbers of parameters and latent variables, Newton-type methods can become infeasible and computationally demanding. When the computer cost is too high one is motivated to use GLIM's EM algorithm solution to initialize GLLAMM's Newton Raphson algorithm. This reduces considerably the execution time for GLLAMM. It was noted that when a final Newton-Raphson step was applied to GLIM's EM solution after convergence the algorithm always converged in at most three iterations yielding a solution which was concave. In spite of this improvement, estimation with GLLAMM still took about five times longer compared to GLIM.

Table 1 displays the parameter estimates and standard errors of the first model using both the EM and Newton-Raphson algorithms. Six threshold (cut-point) parameters were estimated since a 7-point likert scale was used for the rating scores. The GLIM solution required 34 iterations and took 3 minutes to converge. The log-likelihood of this solution was 9807.98. The parameter estimates elicited from the EM algorithm were then used as starting values for the Newton-Raphson algorithm. GLLAMM required three iterations and took 9 minutes to converge. The log-likelihood of the GLLAMM solution was 9807.62.

Term	**EM algorithm**		**Newton-Raphson algorithm**	
	Estimate	**St Error**	**Estimate**	**St Error**
Cutp1	-4.061	0.134	-4.063	0.177
Cutp2	-2.816	0.127	-2.814	0.171
Cutp3	-1.858	0.124	-1.856	0.169
Cutp4	-0.927	0.122	-0.925	0.168
Cutp5	0.118	0.121	0.119	0.167
Cutp6	1.362	0.126	1.364	0.168
Brand(1).Seg(1)	-2.871	0.177	-2.870	0.274
Brand(1).Seg(2)	-1.149	0.140	-1.148	0.191
Brand(2).Seg(1)	-0.636	0.174	-0.636	0.270
Brand(2).Seg(2)	-0.603	0.139	-0.603	0.189
Brand(3).Seg(1)	-2.628	0.176	-2.629	0.332
Brand(3).Seg(2)	-1.360	0.140	-1.360	0.190
Brand(4).Seg(1)	-2.541	0.177	-2.541	0.273
Brand(4).Seg(2)	Aliased	Aliased	Aliased	Aliased

Table 1: Parameter estimates and standard errors elicited the EM and EM+NR algorithms.

Another interesting observation is that GLIM provided deflated standard errors where the deflation for each standard error varied from 24% to 47%. The cause for this deflation is that the EM algorithm has to estimate *KN* missing or unobserved values λ_{nk} together with the model parameters.

Table 2 displays the parameter estimates and standard errors of the second model using both estimation methods. The GLIM solution required 34 iterations and took 10 minutes to converge. The log-likelihood of this solution was 9004.64. Using GLIM's parameter estimates as initial values, GLLAMM required 3 iterations that took 36 minutes to converge. The log-likelihood of the GLLAMM solution was 9003.24 and the amount of deflation of GLIM's standard errors compared to GLLAMM's varied from 0% to 19%.

Term	EM algorithm		Newton-Raphson algorithm	
	Estimate	St Error	Estimate	St Error
Cutp1	-0.631	0.843	-0.634	0.877
Cutp2	0.043	0.843	0.045	0.877
Cutp3	0.604	0.843	0.602	0.877
Cutp4	1.181	0.843	1.180	0.877
Cutp5	1.802	0.843	1.803	0.877
Cutp6	2.513	0.843	2.513	0.877
Door(1).Seg(1)	-1.295	1.135	-1.297	1.214
Door(1).Seg(2)	-0.314	0.044	-0.312	0.053
Door(2).Seg(1)	-0.799	1.135	-0.798	1.214
Door(2).Seg(2)	Aliased	Aliased	Aliased	Aliased
Brand(2).Seg(1)	-0.436	1.079	-0.434	1.090
Brand(2).Seg(2)	1.082	1.188	1.080	1.213
Brand(3).Seg(1)	-0.275	1.078	-0.273	1.090
Brand(3).Seg(2)	0.625	1.189	0.623	1.215
Brand(4).Seg(1)	-0.569	1.083	-0.567	1.104
Brand(4).Seg(2)	1.597	1.186	1.597	1.233
Brand(1).Price.Seg(1)	0.410	0.213	0.411	0.218
Brand(1).Price.Seg(2)	0.406	0.234	0.405	0.244
Brand(2).Price.Seg(1)	0.598	0.212	0.598	0.218
Brand(2).Price.Seg(2)	0.319	0.233	0.317	0.244
Brand(3).Price.Seg(1)	0.515	0.212	0.515	0.216
Brand(3).Price.Seg(2)	0.246	0.234	0.246	0.241
Brand(4).Price.Seg(1)	0.494	0.212	0.494	0.218
Brand(4).Price.Seg(2)	0.133	0.233	0.131	0.244
Brand(1).PriceS.Seg(1)	-0.017	0.014	-0.017	0.014
Brand(1).PriceS.Seg(2)	-0.043	0.016	-0.043	0.016
Brand(2).PriceS.Seg(1)	-0.030	0.014	-0.030	0.014
Brand(2).PriceS.Seg(2)	-0.037	0.015	-0.037	0.016
Brand(3).PriceS.Seg(1)	-0.026	0.014	-0.026	0.014
Brand(3).PriceS.Seg(2)	-0.033	0.016	-0.033	0.016
Brand(4).PriceS.Seg(1)	-0.023	0.014	-0.023	0.014
Brand(4).PriceS.Seg(2)	-0.023	0.015	-0.023	0.016

Table 2: Parameter estimates and standard errors elicited the EM and EM+NR algorithms.

An interesting observation is that when complex models are fitted the discrepancy between GLIM's standard errors compared to GLLAMM's was smaller. An explanation for this occurrence is that the proportion of model parameters compared to the proportion of missing values increases when more terms are included in the model fit.

4 Conclusion

Newton-type algorithms are essential to elicit correct standard errors for the parameter estimates; however, these algorithms are extremely slow since they use numerical first and second derivatives of the log-likelihood. This execution time problem becomes more severe when the number of latent variables in the latent class model is increased. Estimation with a Newton-type algorithm may take fifty times longer compared to estimation with an EM algorithm. The study proposes using the EM algorithm solution as an initialization step. Equipped with very good starting values the final Newton-Raphson step converges quickly. This procedure guarantees correct standard errors of the parameters estimates and reduces execution times considerably. Another interesting finding is that the bias between the correct and incorrect standard errors obtained respectively by Newton-type and EM algorithms becomes less conspicuous as the model complexity increases.

References:

[1] Aitkin, M., Anderson, D., Francis, B. and Hinde, J. (1994), *Statistical Modelling in GLIM,* Oxford Science Publications.

[2] Camilleri, L. and Green, M. (2004), Statistical Models for Market Segmentation, *Proceedings of the 19th International Workshop Statistical Modelling, Florence*. 120-124.

[3] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977), Maximum Likelihood from Incomplete Data via the EM algorithm, *Journal of the Royal Statistical Society*, B, 39, 1-38.

[4] Fisher, R.A. (1925), Theory of Statistical Estimation. *Proc. Camb. Phil. Society*., 22, 700-725.

[5] Francis, B., Green, M. and Payne, C. (1993), *The GLIM 4 manual*, Oxford Science Publications.

[6] Green, M. (2000), Statistical Models for Conjoint Analysis, *Proceedings of the 15th International Workshop on Statistical Modelling, Bilbao*. 216-222.

[7] Green, P.J. (1984), Iteratively Reweighted Least Square for Maximum Likelihood Estimation *Journal of Royal Statistical Society*, B, 46, 149-192.

[8] Jamshidian, M. and Jennrich, R.I. (1997), Acceleration of the EM algorithm using quasi-Newton methods, *Journal of the Royal Statistical Society* B, 569-587.

[9] Jamshidian, M. and Jennrich, R.I. (2000), Standard Errors for EM Estimation, *Journal of the Royal Statistical Society* B, 257-270.

[10] Louis, T.A. (1982), Finding the Observed Information Matrix when using the EM algorithm, *Journal of the Royal Statistical Society*, 44, 226-233.

[11] McCullagh P. (1980) Regression Models for Ordinal Data, *J.R. Statistical .Soc B*, 42, 109-142.

[12] McCulloch, C.E. (1998), Maximum Likelihood Variance components estimation for Binary Data, *Journal of the American Statistical Association*, 89, 330-335.

[13] Meilijson, I. (1989), A Fast Improvement to the EM algorithm on its Own Term, *Journal of the Royal Statistical Society*, 51, 127-138.

[14] Meng, X.L. and Rubin, D.B. (1991), Using EM to obtain asymptotic variance-covariance matrices: the SEM algorithm, *Journal of American Statistical Association*, 86, 899-909.

[15] Nelder, J.A., Wedderburn, R.W.M. (1972), Generalized Linear Models, *Journal of the Royal Statistical Society*, A, 135, 370-384.

[16] Rabe-Hesketh, S., Pickles, A. and Skrondal, A. (2001), GLLAMM: A General Class of Multilevel Models and STATA, *Multilevel Modelling Newsletter*, 13, 17-23.

[17] Segal, M.R., Bacchetti, P. and Jewell, N.P. (1994), Variance for Maximum Penalized Likelihood estimates obtained via the EM algorithm, *Journal of the Royal Statistical Society* B, 56, 345-352.

[18] Skrondal, A. and Rabe-Hesketh, S. (2004), *Generalized Latent Variable Modelling*, Chapman & Hall/CRC.

[19] Vermunt, J.K. (2004), An EM algorithm for the estimation of parametric and non-parametric hierarchical non-linear models, *Statistica Neerlandica*, 58, 220-233.