ISSN: 2414-1895 DOI: 10.6919/ICJE.202405_10(5).0062

An Improved Monte Carlo EM Acceleration Algorithm

Yugui Lu[1,2], Qunlan Li[3]

[1] School of Mathematics and Physics, Hechi University, Yizhou 546300, China

[2] School of Economics and Trade, Guangxi Vocational Teachers College, Nanning 530007, China

[3] School of Teacher Education, Hechi University, Yizhou 546300, China

Abstract

In this paper, an improved Monte Carlo EM (MCEM) acceleration algorithm is proposed, aiming at solving the problems of slow convergence and difficult integral calculation of the traditional MCEM algorithm when dealing with complex models. The article first reviews the basic concepts and application background of EM algorithms and MCEM algorithms, and then points out the limitations of MCEM algorithms with high-dimensional data and complex models. To overcome these challenges, the authors introduce a new algorithm that approximates the solution of N-R step integrals by means of a Monte Carlo simulation method, which improves the convergence speed of the algorithm and maintains the quadratic convergence property. Specifically, the improved MCEM acceleration algorithm consists of the following key steps: random sampling in the E1 step, computation of the expectation in the E2 step, maximization of the objective function in the M step, and approximation of the integral using the Monte Carlo method in the improved N-R step. Through numerical examples, the authors demonstrate the advantages of the improved algorithm over the original MCEM algorithm and the MCEM accelerated algorithm in terms of accuracy of parameter estimation and speed of convergence. In addition, the article discusses the effect of random number selection on the performance of the algorithm and provides a method for choosing the appropriate number of random numbers. In conclusion, the improved MCEM acceleration algorithm effectively improves the computational efficiency and accuracy of the results when dealing with complex data models, which has important practical application value for statistical analysis in the era of big data.

Keywords

MCEM Algorithm; Newton-Raphson Algorithm; Monte Carlo Simulation.

1. Introduction

The EM algorithm is a commonly used algorithm for dealing with missing data, initially proposed by Dempster et al. (1977), consisting of an expectation step (E step) and a maximization step (M step) [1]. In practical applications, it is sometimes difficult, or even impossible, to obtain the explicit expression of the E step expectation in the EM algorithm. To solve this problem, Louis (1982) proposed the use of Monte Carlo simulation to complete the solution of the E step, leading to the development of the generalized EM algorithm known as the MCEM algorithm [2]. The convergence speed of the EM algorithm is inherently controlled by the reciprocal of the missing data information and is generally linear. With high rates of data missing, the EM algorithm converges more slowly, and the MCEM algorithm, which relies on Monte Carlo simulation to solve for the expectation, converges even more slowly compared to the EM algorithm. To address the slow convergence speed

Volume 10 Issue 5, 2024
DOI: 10.6919/ICJE.202405_10(5).0062

of the MCEM algorithm, Luo ji (2008) proposed an MCEM acceleration algorithm that combines the MCEM algorithm with the N-R algorithm, giving the MCEM algorithm a quadratic convergence speed on top of its original linear convergence speed, effectively increasing the operational speed of obtaining results with the MCEM algorithm [3]. In the era of big data, the complexity and redundancy of high-dimensional data have a certain impact on the convergence speed of the MCEM acceleration algorithm, and the increased complexity of the model also poses a challenge to the calculation of the N-R step integral in the MCEM acceleration algorithm. Therefore, based on previous work, this paper proposes an improvement to the difficult-to-solve N-R step integral using the Monte Carlo simulation method to find its integral approximation. By ensuring the quadratic convergence speed of the MCEM acceleration algorithm, the difficulty of solving the N-R step integral in the MCEM acceleration algorithm is addressed, further expanding the applicability of the MCEM acceleration algorithm.

2. The MCEM Algorithm

The MCEM algorithm is a method that utilizes Monte Carlo simulation to effectively implement the integration of the E step in the EM algorithm [4-6], dividing the E step into the following E1 and E2 steps:

Step E1: Draw m random numbers $Z_1, Z_2, \cdots, Z_m$ randomly from $p(Z|\theta^{(i+1)}, Y)$.

Step E2: Calculate $\hat{Q}(\theta|\theta^{(i)}, Y) = \frac{1}{m}\sum_{j=1}^{m} \log p(\theta|z_j, Y)$.

By the law of large numbers, as long as m is sufficiently large, $\hat{Q}(\theta|\theta^{(i)}, Y)$ and $Q(\theta|\theta^{(i)}, Y)$ will be very close, thus we can maximize a in the M step.

3. MCEM Acceleration Algorithm

As an extension of the EM algorithm, the MCEM algorithm has stronger applicability. However, like the EM algorithm, its convergence speed is controlled by the reciprocal of the missing information, and with a higher proportion of missing data, the calculations generated by the Monte Carlo simulations during expectation estimation cause the MCEM algorithm to converge more slowly than the EM algorithm. The N-R algorithm has a quadratic convergence speed near the posterior mode. To improve the convergence speed of the MCEM algorithm and avoid slow convergence at high data missing rates, Luo J (2008) combined the MCEM algorithm with the N-R algorithm, resulting in the MCEM acceleration algorithm that has a quadratic convergence speed. The computational logic of the MCEM acceleration algorithm is: first, use the MCEM algorithm to implement the first step of iteration to obtain the optimal solution $\theta_{EM}^{(i)}$, then use the N-R method to correct it, obtaining a new optimal solution, denoted as $\theta^{(i+1)}$. At this point, complete the i step of iteration, and continue iterating until the algorithm converges. The specific steps of the MCEM acceleration algorithm are as follows:

Step E1:Draw m random numbers $Z_1, Z_2, \cdots, Z_m$ randomly from $p(Z|\theta^{(i+1)}, Y)$.

Step E2:Calculate.

$$\hat{Q}(\theta|\theta^{(i)}, Y) = \frac{1}{m}\sum_{j=1}^{m} \log p(\theta|z_j, Y)$$

Step M:Let's maximize $\hat{Q}(\theta|\theta^{(i)}, Y)$,We get solution $\theta_{EM}^{(i)}$.

Step N-R:Let.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202405_10(5).0062

$$\begin{aligned}
\theta^{(i+1)} &= \theta^{(i)} + \left[\nabla^2 Q(\theta^{(i)} \middle| \theta^{(i)}, Y)\right]^{-1} \nabla Q(\theta^{(i)} \middle| \theta^{(i)}, Y) \\
&= \theta^{(i)} + \left[-\left.\frac{\partial^2 \ln p(\theta|Y)}{\partial \theta^2}\right|_{\theta^{(i)}} \right]^{-1} \left[\int \frac{\partial^2 \ln p(Z|Y,\theta)}{\partial \theta^2} p(Z|Y,\theta) dZ \Big|_{\theta^{(i)}} \right] (\theta_{EM}^{(i)} - \theta^{(i)}) \\
&= \theta^{(i)} + \left[\int \frac{\partial^2 \ln p(\theta|Y,Z)}{\partial \theta^2} p(Z|Y,\theta) dZ \Big|_{\theta^{(i)}} - \int \frac{\partial^2 \ln p(Z|Y,\theta)}{\partial \theta^2} p(Z|Y,\theta) dZ \Big|_{\theta^{(i)}} \right]^{-1} \cdot \left[Var\left(\frac{\partial \ln p(\theta|Y,Z)}{\partial \theta} \right)\Big|_{\theta^{(i)}} \right] \cdot (\theta_{EM}^{(i)} - \theta^{(i)}) \\
&= \theta^{(i)} + \left[\int \frac{\partial^2 \ln p(\theta|Y,Z)}{\partial \theta^2} p(Z|Y,\theta) dZ \Big|_{\theta^{(i)}} - Var\left(\frac{\partial \ln p(\theta|Y,Z)}{\partial \theta} \right)\Big|_{\theta^{(i)}} \right]^{-1} \left[\int \frac{\partial^2 \ln p(Z|Y,\theta)}{\partial \theta^2} p(Z|Y,\theta) dZ \Big|_{\theta^{(i)}} \right] \cdot (\theta_{EM}^{(i)} - \theta^{(i)})
\end{aligned} \tag{1}$$

Where $\nabla^2 Q(\theta^{(i)} | \theta^{(m)}, X)$ and $\nabla Q(\theta^{(i)} | \theta^{(m)}, X)$ represent the Hesse matrix and gradient of the function $\nabla Q(\theta | \theta^{(m)}, X)$ at $\theta^{(i)}$.

Thus an iteration is formed: $\theta^{(i)} \to \theta^{(i+1)}$ .Iterate through the steps E1, E2, M, and N-R above until $\left\| \theta^{(i+1)} - \theta^{(i)} \right\|$ or $\left\| Q(\theta^{(i+1)} | \theta^{(i)}, Y) - Q(\theta^{(i)} | \theta^{(i)}, Y) \right\|$ is sufficiently small.

4. The Improved MCEM Acceleration Algorithm

The MCEM acceleration algorithm effectively improves the convergence speed of the algorithm, but for more complex models, the integral calculation of the N-R step is too complex, and it is difficult to give its explicit expression, similar to the difficulty of obtaining the explicit expression of the E-step expectation in the EM algorithm. Based on this, this paper proposes an improved MCEM acceleration algorithm that combines the Monte Carlo method with the N-R algorithm, providing an approximation for equation (1) and solving the difficulty of integral calculation in the N-R step. The computational logic of the improved MCEM acceleration algorithm is: while maintaining the conditions of the E1 step, E2 step, and M step of the MCEM acceleration algorithm, the N-R step is replaced with an improved N-R step.

Given that when sufficiently large, the following equations (2) and (3) hold:

$$\begin{aligned}
\int \frac{\partial^2 \ln p(\theta|Y,Z)}{\partial \theta^2} p(Z|Y,\theta) dZ \Big|_{\theta^{(i)}} &\approx -\frac{1}{m} \sum_{j=1}^{m} \left.\frac{\partial^2 \ln p(\theta|Y,z_j)}{\partial \theta^2}\right|_{\theta^{(i)}} \\
&= C_1(\theta)\big|_{\theta^{(i)}}
\end{aligned} \tag{2}$$

$$\begin{aligned}
Var\left(\frac{\partial \ln p(\theta|Y,Z)}{\partial \theta} \right)\Big|_{\theta^{(i)}} &\approx -\frac{1}{m} \sum_{j=1}^{m} \left(\left.\frac{\partial \ln p(\theta|Y,z_j)}{\partial \theta}\right|_{\theta^{(i)}} \right)^2 - \left[\frac{1}{m} \sum_{j=1}^{m} \left.\frac{\partial \ln p(\theta|Y,z_j)}{\partial \theta}\right|_{\theta^{(i)}} \right]^2 \\
&= C_2(\theta)\big|_{\theta^{(i)}}
\end{aligned} \tag{3}$$

Thus, the N-R step of the MCEM acceleration algorithm can be improved as equation (4):

$$\theta^{(i+1)} = \theta^{(i)} + \left(C_1(\theta)\big|_{\theta^{(i)}} - C_2\big|_{\theta^{(i)}} \right)^{-1} \cdot \left(C_1\big|_{\theta^{(i)}} \right) \cdot (\theta_{EM}^{(i)} - \theta^{(i)}) \tag{4}$$

According to the law of large numbers, as long as the random number m is sufficiently large, the value of the improved N-R step is very close to that of the N-R step. Therefore, the improved MCEM acceleration algorithm can maintain the advantage of quadratic convergence speed while solving the problem of difficult N-R step integral calculations in the accelerated algorithm.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202405_10(5).0062

5. Numerical Simulation

Suppose there are now 1000 respondents, divided according to gender and color blindness into four categories: normal male, color-blind male, normal female, color-blind female [7], i.e.,

$$Y=(y_1,y_2,y_3,y_4)=(442,38,514,6),$$

According to the genetic model, the unit probability of the data is:

$$\left(\frac{\theta}{2},\frac{1-\theta}{2},\frac{\theta^2}{2}+\theta(1-\theta),\frac{(1-\theta)^2}{2}\right),\text{where}\,\theta\in(0,1).$$

To estimate parameter θ, introduce unobserved data Z, dividing the third unit into two units, then the augmented data are as follows:

$$X=\left(y_1,y_2,Z,y_3-Z,y_4\right),$$

And the unit probability of the augmented data is:

$$\left(\frac{\theta}{2},\frac{1-\theta}{2},\frac{\theta^2}{2},\theta(1-\theta),\frac{(1-\theta)^2}{2}\right)$$

And under the condition of a flat prior,

$$\log p(\theta|Y,Z)=(y_1+y_3+Z)\log(\theta)+(y_2+y_3+2y_4-Z)\log(1-\theta).$$

Thus, it follows that:

$$\frac{\partial\ln p(\theta|Y,Z)}{\partial\theta}=\frac{y_1+y_3+Z}{\theta}-\frac{y_2+y_3+2y_4-Z}{1-\theta}$$
$$\frac{\partial^2\ln p(\theta|Y,Z)}{\partial\theta^2}=-\frac{y_1+y_3+Z}{\theta^2}-\frac{y_2+y_3+2y_4-Z}{(1-\theta)^2}$$

Below, we use the improved MCEM acceleration algorithm to estimate the value of θ:

Step E1: Draw m random numbers $Z_1,Z_2,\cdots,Z_m$ randomly from $B(514,\theta^{(i)}/(2-\theta^{(i)}))$,then:

$$\bar{z}=\frac{1}{m}\sum_{j=1}^{m}z_j\ ,\ \overline{z^2}=\frac{1}{m}\sum_{j=1}^{m}z_j^2$$

Step E2:

$$\hat{Q}(\theta|\theta^{(i)},Y)=(y_1+y_3+\overline{Z})\log(\theta)+(y_2+y_3+2y_4-\overline{Z})\log(1-\theta)$$

Step M: Let's maximize $\hat{Q}(\theta|\theta^{(i)},Y)$,Then there is:

$$\theta_{EM}^{(i)}=\frac{y_1+y_3+\overline{Z}}{y_1+y_2+2y_3+2y_4}.$$

Step improved N-R:

$$\theta^{(i+1)}=\theta^{(i)}+\left(C_1(\theta)\Big|_{\theta^{(i)}}-C_2\Big|_{\theta^{(i)}}\right)^{-1}\cdot\left(C_1\Big|_{\theta^{(i)}}\right)\cdot(\theta_{EM}^{(i)}-\theta^{(i)})$$

where $C_1(\theta)\Big|_{\theta^{(i)}}=\frac{y_1+y_3+\overline{z}}{(\theta^{(i)})^2}-\frac{y_1+y_3+2y_4-\overline{z}}{(1-\theta^{(i)})^2}$; $C_2\Big|_{\theta^{(i)}}=\frac{\overline{z^2}-\left(\overline{z}\right)^2}{(\theta^{(i)}(1-\theta^{(i)}))^2}$.

Below, we set m =1000, using $\left\|\theta^{(i+1)}-\theta^{(i)}\right\|<10^{-4}$ as the convergence criterion, successively using the MCEM algorithm, MCEM acceleration algorithm, and the improved MCEM algorithm to find the maximum likelihood estimation of parameter θ, with the iteration results and the number of iterations shown in Table 1.

Table 1. Iteration Results for Parameter Estimation

	MCEM		MCEM acceleration algorithm		The improved MCEM acceleration algorithm
$\theta^{(0)}=0.5000$	0.7271	0.9117	0.8007	0.9125	0.8898
	0.8217	0.9122	0.8580	0.9130	0.9095
	0.8648	0.9124	0.8840	0.9131	0.9125
	0.8865	0.9127	0.8970	0.9130	0.9129
	0.8983	0.9130	0.9040	0.9130	0.9123
	0.9045	0.9133	0.9080		0.9128
	0.9082	0.9130	0.9106		0.9128
	0.9101	0.9129	0.9117		
	0.9113	0.9129	0.9122		
iterations	18		14		7

From the results in Table 1, it is easy to see that to estimate the value of parameter θ using the MCEM algorithm requires 18 iterations, with a relatively slow convergence speed, whereas the MCEM acceleration algorithm significantly outperforms the MCEM algorithm in terms of convergence speed. The MCEM acceleration algorithm converges after 14 iterations, four less than the unaccelerated MCEM algorithm, while the improved MCEM acceleration algorithm only requires 7 iterations to converge. Its convergence speed is significantly faster than both the MCEM algorithm and the MCEM acceleration algorithm, indicating that the improved MCEM acceleration algorithm compensates for the convergence speed sacrificed by the MCEM algorithm's use of Monte Carlo

Volume 10 Issue 5, 2024
DOI: 10.6919/ICJE.202405_10(5).0062

simulation in the E step within the improved N-R step. Therefore, the improved MCEM acceleration algorithm is superior to the MCEM acceleration algorithm.

6. Selection of Random Numbers

In the MCEM algorithm, the selection of random numbers m in step E1 significantly affects the stability and convergence of the algorithm's iterative results. Values too small cannot achieve stable convergence, and values too large reduce the algorithm's computational efficiency. In the numerical simulation example, we select $m=1000$; below, we discuss the reasons for this choice. First, we take $m=10,100,1000$ respectively, and using $\left\|\theta^{(i+1)}-\theta^{(i)}\right\|<10^{-4}$ as the convergence criterion, we perform the MCEM algorithm and plot as follows:

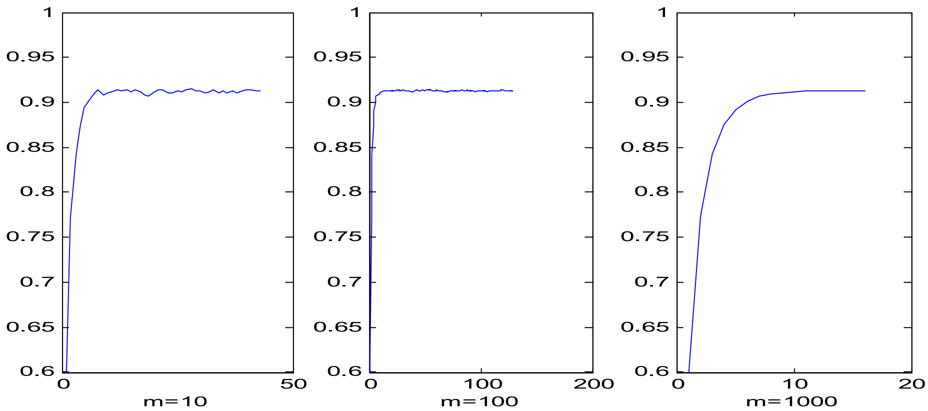

Figure 1. The iteration process of the MCEM algorithm with m ranging from 10 to 1000

From Figure 1, it can be seen that when the random number m is set to 10, the fluctuation of the estimated values remains significant after numerous iterations, leading to unstable estimates and poor iteration performance; when m is set to 100, despite a reduction in the fluctuation amplitude after several iterations, it requires more than 100 iterations to achieve stable onvergence, resulting in average iteration performance; when m is set to 1000, the estimated values tend to stabilize after a few iterations and converge after just over 10 steps. Although selecting a larger random number could yield higher precision, it would undoubtedly increase the computation time of the algorithm. Therefore, for the numerical simulation case in this paper, choosing a random number m of 1000 is appropriate.

7. Conclusion

As an extension of the EM algorithm, the MCEM algorithm has broader applicability. However, its use of Monte Carlo simulation in the E-step sacrifices the convergence speed of the algorithm, which is inherently linear. In response to this, this paper has improved the convergence speed of the MCEM algorithm, achieving a square convergence speed and then appropriately amplifying the convergence speed once more, resulting in the improved MCEM acceleration algorithm that enhances the iteration efficiency of the algorithm. The example calculations demonstrate the superiority of the improved MCEM acceleration algorithm. Finally, the paper concludes with an analysis and discussion on the method of selecting random numbers in the MCEM algorithm.

Acknowledgments

Funding: Guangxi Philosophy and Social Science Planning Research Project "Research on Guangxi's Comprehensive Collaborative Innovation Strategy with Leading Regional Innovation Highlands as the Core" (22CJL004); Guangxi Universities Young and Middle-aged Teachers' Research Basic

Ability Improvement Project "Statistical Inference Methods and Their Applications for Semiparametric Panel Models with Missing Data" (2021KY0618).

References

[1] Dempster A P, Laird N M, Rubin D B. Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion) [J]. Journal of the Royal Statistical Society, Series B, 1977, 39:1-38.

[2] Louis T A. Finding the observed information matrix when using the EM algorithm [J]. Journal of the Royal Statistics Society, Series B, 1982, 44:226-233.

[3] Luo J. Accelerated Monte Carlo EM Algorithm [J]. Applied Probability and Statistics, 2008, 24(3): 311-318.

[4] Meng X L, Dyk D V. The EM algorithm-an old folk-song sung to a fast new tune [J]. Journal of the Royal Statistical Society, Series B, 1997, 59(3), 511-567.

[5] Fessler J A, Hero A O. Space-alternating generalized expectation-maximization algorithm [J]. IEEE Transactions on Signal Processing, 1994, 42(10):2664-2677.

[6] Walker S. An EM algorithm for nonlinear random effects models [J]. Biometrics, 1996, 52(3):934-944.

[7] Mao S S, Wang J L, Pu X L. Advanced Mathematical Statistics [M]. Higher Education Press (Third Edition), 2022.