The EM Algorithm in Practice: A Methodological Survey of Applications, Extensions, and Computation

Hanson Yu

Jincheng College, Sichuan University, Chengdu, Sichuan 611731

Abstract: ***This paper presents the fundamental principles of the Expectation-Maximization (EM) algorithm, a well-established iterative method for parameter estimation in statistical models involving latent variables. A representative dataset is selected to illustrate the practical application of the algorithm. Implementations of the EM algorithm are developed in both C and Python programming languages, enabling a comparative assessment of computational performance and accessibility. The corresponding experimental results are systematically presented and analyzed, with particular emphasis on the algorithm's convergence behavior and the accuracy of parameter estimates. The findings provide insights into the practical utility and robustness of the EM algorithm across different programming environments.***

Keywords: Algorithms; EM; Experiments.

1. INTRODUCTION

In our daily life, we often encounter incomplete missing data. In the course of data mining, we often deal with missing data. There are many ways to deal with data loss, but replenishing missing data in a big data environment is very tedious. The EM algorithm is an algorithm that can find reliable missing values very consistently, and although it has some disadvantages, its advantage is that the algorithm is simple and the problem is very broad, so it is also widely used.

Zhang et al. [1] established and applied a flow field evaluation system after early polymer injection in thick reservoirs in 2025. Ding [2] developed and validated a multispectral vision system for in-situ detection of pesticide residues on agricultural produce in 2025. Junxi, Wang, and Chen [3] proposed a graph convolutional network based on matrix factorization (GCN-MF) for recommendation in 2024. Hu, Zhang, and Sun [4] unveiled new directions in text sentiment analysis using multiscale deep neural networks in 2024. Ge and Wu [5] conducted an empirical study on the adoption of ChatGPT for bug fixing among professional developers in 2023. Ma [6] proposed a unified framework for congestion diagnosis and dynamic mitigation in complex networks in 2025. Luo [7] performed an integration analysis of computer application technology and information management in 2026. Ya [8] studied EDA technology in digital circuit design with a focus on application methodologies in 2025. Wang [9] researched the application of computer science and technology in the context of big data in 2026. Peng et al. [10] exploited aggregation and segregation of representations for domain adaptive human pose estimation in 2025. Peng et al. [11] introduced RAIN, a regularization method on input and network for black-box domain adaptation, in 2023. Narouei et al. [12] examined the effects of germicidal far-UVC on ozone and particulate matter in a conference room in 2025. Shan, Xu, Xia, and Lin [13] rethought wine tasting for Chinese consumers using a service design approach enhanced by multimodal personalization in 2025. Tang et al. [14] designed and optimized a shallow-angle grating coupler for vertical emission from indium phosphide devices in 2020. Sun [15] addressed accessibility challenges and solutions in designing inclusive interfaces for digital products in 2025. Zhou [16] proposed a digital precision distribution strategy for social media content on private domain platforms in the automotive industry using a collaborative filtering model based on user behavior in 2025. Yang, Zheng, and Lu [17] constructed a multi-dimensional network credit-related transaction risk map with early warning by integrating graph neural networks in 2025. Yuan et al. [18] introduced TA-Mem, a tool-augmented autonomous memory retrieval method for large language models in long-term conversational question answering, in 2026. Finally, Yang et al. [19] developed a recursive multi-agent trading system for iterative optimized portfolio strategy under geopolitical uncertainty in 2026.

2. THEORY OF EM ALGORITHM

The EM arithmetic is called the maximum expected arithmetic. It is very important to find the missing value in data mining. This algorithm is an iterative algorithm whose main application scenario is to be in a model that is a probability parameter model and contains unknown variables, also called hidden variables. In such a model, there is a maximum likelihood estimation, which is also known as the maximum a posteriori probability estimation. In the field of software engineering, EM algorithm is widely used in machine learning, and very hot.

2.1 Algorithmic Processes

Em algorithm is a strategy to optimize parameters, which is calculated by iteration Its optimization calculation can be mainly divided into two steps, called e step and m step, respectively, and can also be called the expected step and the maximum step, so this algorithm is also called the em algorithm, which is a shorthand combination of these two optimization step.[2]The basic idea of this model algorithm is to first accurately estimate the mean value of the model parameters based on the previously provided observation data, also called target observation data, by analyzing and processing the data set. Next, use the parameter value as a base point based on the estimate of the previous parameter model, continue to estimate the next missing data value, and then continue to combine the previous estimate with the missing data again. Based on previous observations, repeatedly iterate and estimate the parameter value. During the iterations, this estimate will be constrained. When this constrained occurs, the iteration ends and the program ends [3].

2.2 Jensen inequality

The Jensen inequality is defined as follows:

(1) Assuming that a domain-defined range of the function f is as a real number, if any real number x can be chosen and the second-order derivative of f (x) is greater than 0, then f is a scalar function.

(2) If there is a convex function f, then e [f (x)] is much greater than f (e [x]) when there is x which is not a random variable and x is not a random constant. E [f (x)] is equivalent to f (e [x]) if and only if x (x) is constant. Where e (x) represents the expectation of x for a mathematical prediction.

(3) when jensen's inequality is widely applied to concave functions, the inverse direction of the sign is a reverse direction, that is, e [f (x)] is less than f (e [x]). E [f (X)] is equal to f (E [X]) if and only if x is constant. Where e (x) means x for a mathematical expectation.
(4) The main use of the Jensen inequality is to prove the convergence of the EM algorithm[4]。

3. EXPERIMENTAL DESIGN

The experiment chooses c + + language and python programming language environment, this paper uses a simple example to understand and learn the process of em algorithm. The Em algorithm is mainly an estimation method proposed by Dempster, Lad and Rubin in 1977. The main purpose of this estimation method is to obtain estimates of parameters that are very similar to each other and can be obtained from an unknown situation. The values of the numbers can be placed in an incomplete data set to estimate the values of each parameter, which is also a simple and very useful machine learning algorithm. This technique has been widely studied and applied in data mining. At present, our main goal is to collect and process those defective data, censored data [5]

3.1 Dataset selection and initialization parameters

The data for this experiment are shown in Table 1, which is a transaction data set. T100-T500 represents the cases where 5 customers purchased merchandise I1-I5, 1 means purchase and 0 means no.

Table 1: Trading Data Set

	L1	L2	L3	L4	L5
T100	0	0	1	1	1
T200	1	1	0	0	0
T300	0	0	0	1	1
T400	0	1	1	1	1
T500	1	1	0	0	0

3.2 Experiment in C + + language environment

3.2.1 Initialization parameters

Parameter initialization:

doublearr[2][5]={{0.3,0.4,0.3,0.7,0.4},{0.6,0.5,0.3,0.4,0.4}};/ / The probability of men and women buying various goods doublex[5][5]={{0,0,1,1,1},{1,1,0,0,0},{0,0,0,1,1},{0,1,1,1,1},{1,1,0,0,0}};/ / Dataset

3.2.2 Step

Calculating the expectation (E), calculating the maximum likelihood estimate of the hidden variable using the existing estimate of the hidden variable;

voidm_step (doublearr [2] [5], doublex [5] [5], doublea1 [5], doublea 2 [5]) {doublesum1=0.0, sum2=0.0; for (inti=0; i<5; i++) {sum1+=a1 [i]; sum2+=a2 [i]; cout<<a1[i]<<" "<<a2[i]<<endl;} // Update the probability of men and womenk1=sum1/5; k2=sum2/5; // Probability of updating products for(intj=0; j<5; j++) {doubles1=0.0, s2=0.0; for(intk=0; k<5; k++) {s1+=a1 [k]*x [k] [j]; s2+=a2 [k]*x [k] [j];} arr [0] [j]=s1/sum 1; arr [1] [j]=s2/sum2;}}

At each step E, the first step is initialized, and then the & theta; And update all & theta; The value of that is.

3.2.3 M-step

Maximization (M) is the maximum likelihood value obtained by using the theory described above on the basis of maximization in E step, and the value of the parameter is calculated through this value.

voidm_step(doublearr[2][5],doublex[5][5],doublea1[5],doublea2[5]){doublesum1=0.0,sum2=0.0; for(inti=0;i<5;i++){sum1+=a1[i];sum2+=a2[i];cout<]<<""<]<}

Every time you enter the M step, it updates & pi; The value is.

3.2.4 Iteration

An estimate of a parameter found at step M is used repeatedly at step E to calculate this value, and the process continues over and over again until the iterations are too many or aggregated

While (flag) {e_step (arr, x, a1, a2); m_step (arr, x, a1, a2); print (a1, a2, arr); cout<<endl<< "k1=" <<k1<< "k2=" <<k2<<endl; cout<<" "<<endl; F--; if (F<=0||K1=k1 & & K2=k2) {flag=false;}

The reason for stopping too many iterations is to avoid improper selection of the data set and the program entering a dead cycle. At the same time, the program can also optimize the selection of conditions for the end of an iteration because the conditions currently selected are too precise. In fact, by observing the iterative process, the accuracy of the iteration results can be 0.01, not 100 percent, so this is space for the program to be optimized for efficiency.

3.3 Experiments in Python environment

Symbol interpretation:

(1) π K represents the probability of occurrence of the kth component (k = 2, which can be understood as the male component and the female component) where the sum of the probabilities of both sexes should be 1. Use the brackets in the code _ 1 for males and _ 2 for females.

(2) X (i) j = 1 (or 0) indicates that customer i purchased the product j (or not), e.g. x (1) 1 = 1 indicates that customer 1 purchased the product 1. The code uses two dictionaries to represent the shopping situation of five users P1-P5. It is divided into vertical and horizontal representations.

(3) θ Kj represents the probability that the jth variable is observed to have a value of 1 in the kth component weight. This can be understood here as the preference of men and women for five items.

(4) p (k | i) indicates the probability that the data in article i belongs to the k component. This can be understood as the probability that the i-th customer is male and female.

In Python, the maximum number of bits that can be represented by a float calculation is limited, and in probability operations, it will eventually stop at 1.0 or 0.0. Accordingly, conditions of cessation may be established on this basis. The p (k | i) representation in the EM algorithm represents the probability that the i data belongs to the k th component. That is, there are five pieces of data for boys and girls, so when the number of iterations is enough, the value of p is only 1.0 or 0.0. If the number of either 1.0 or 0.0 in a statistical array adds up to 10, then the iteration stops.

After analyzing the variables, you can consider writing a two-step implementation of E, M, and E step to calculate the probability of the fourth variable, that is, the five roles of gender. The probability was initially set at 0.4 for boys and 0.6 for girls.

Defexpectation (): foriinrange(5): forjinrange(5): p [0] [i]*=theta [0] [j]**x [i] [j]* (1-theta [0] [j]) ** (1-x [i] [j]) p [1] [i]*=theta [1] [j]**x [i] [j]* (1-theta [1] [j])**(1-x[i] [j]) foriinrange (5): p [0] [i]= (p [0] [i]*PI1) / (p [0] [i]*PI1+p [1] *PI2) P [1] [i] = 1-p [0] [i] print'p:') print (p [0]) print (p [1])

The above is the E step code implementation.

Step M updates the probability of gender and the degree of preference for items between men and women.

defmaximization(): sum1=0.0sum2=0.0 foriinrange(5): sum1+=p[0][i]

Computer System Networks and Telecommunications:

sum2+=p[1] [i] PI1=sum1/5 PI2=sum2/5 foriinrange (5): s1=0.0s2=0.0 forjinrange (5): s1+=p[0] [j]*x[j] [i]s2+=p [1] [j]*x[j] [i] theta [0] [i]=s1/sum1 theta[1] [i]=s2/sum2print ('П1:',PI1,'П2:',PI2) print (':') print (theta [0]) print (theta[1])

Here design when new and old pi； The five digits after the decimal point of k do not judge that the iteration is stopped when the change occurs. The above is the M step code implementation process.

4. EXPERIMENTAL RESULTS

After completing an instance of the EM algorithm in both languages, run the program, and according to the output in Figure 1, we can see that after the third iteration, k1 and k2 have stabilized and are no longer changed.

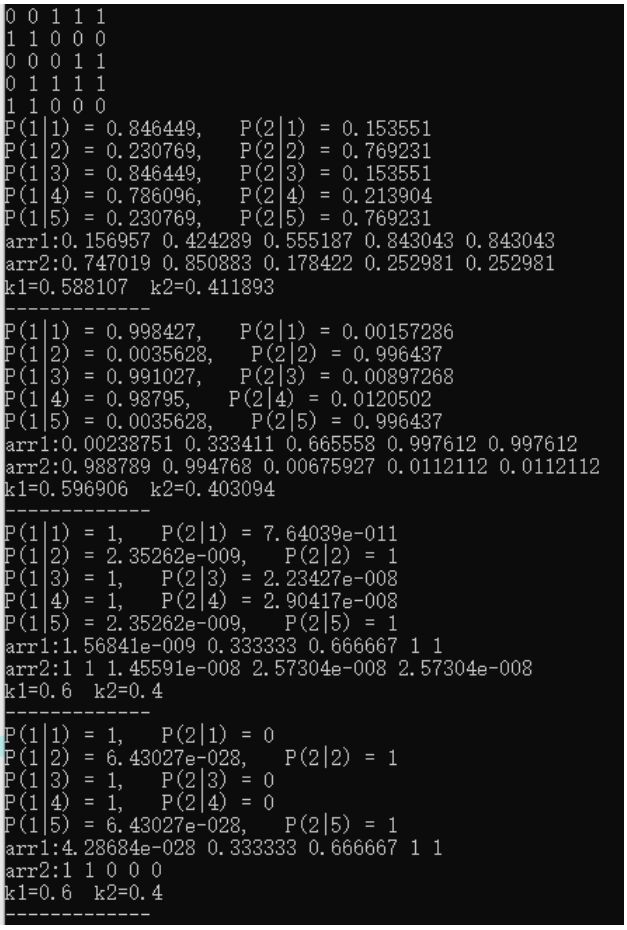

Figure 1: EM algorithm execution output diagram

REFERENCES

[1] Zhang, J., Liu, Y., Chen, X., Zheng, B., Li, J., & Ye, L. (2025). Establishment and Application of Flow Field Evaluation System after Early Polymer Injection in Thick Reservoir. International Journal of Advance in Applied Science Research, 4(12), 54-63.

[2] Ding, G. (2025). Development and Validation of a Multispectral Vision System for In-Situ Detection of Pesticide Residues on Agricultural Produce. International Journal of Advance in Applied Science Research, 4(8), 98-102.

[3] Junxi, Y., Wang, Z., & Chen, C. (2024). GCN-MF: A graph convolutional network based on matrix factorization for recommendation. Innovation & Technology Advances, 2(1), 14–26. https://doi.org/10.61187/ita.v2i1.30

[4] Hu, H., Zhang, J., & Sun, Y. (2024). The Multiscale Deep Neural Networks: Unveiling New Directions in Text Sentiment Analysis. Innovation & Technology Advances, 2(2), 34–45. https://doi.org/10.61187/ita.v2i2.65

[5] Ge, H., & Wu, Y. (2023). An Empirical Study of Adoption of ChatGPT for Bug Fixing among Professional Developers. Innovation & Technology Advances, 1(1), 21–29. https://doi.org/10.61187/ita.v1i1.19

[6] Ma, J. (2025). A Unified Framework for Congestion Diagnosis and Dynamic Mitigation in Complex Networks. International Journal of Advance in Applied Science Research, 4(11), 36-41.

[7] Luo, R. (2026). The Integration Analysis of Computer Application Technology and Information Management. International Journal of Advance in Applied Science Research, 5(2), 27-30.

[8] Ya, L. (2025). EDA Technology in Digital Circuit Design: A Study on Application Methodologies. International Journal of Advance in Applied Science Research, 4(12), 6-10.

[9] Wang, J. (2026). Research on the Application of Computer Science and Technology in the Context of Big Data. International Journal of Advance in Applied Science Research, 5(1), 72-77.

[10] Peng, Qucheng, Ce Zheng, Zhengming Ding, Pu Wang, and Chen Chen. "Exploiting Aggregation and Segregation of Representations for Domain Adaptive Human Pose Estimation." In 2025 IEEE 19th International Conference on Automatic Face and Gesture Recognition (FG), pp. 1-10. IEEE, 2025.
[11] Peng, Qucheng, et al. "RAIN: regularization on input and network for black-box domain adaptation." Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. 2023.
[12] Narouei, F. H., Tang, Z., Wang, S. I., Hashmi, R. H., Welch, D., Sethuraman, S., ... & McNeill, V. F. (2025). Effects of germicidal far-UVC on ozone and particulate matter in a conference room. Plos one, 20(8), e0328224.
[13] Shan, X., Xu, Y., Xia, T., & Lin, Y. S. (2025, October). Rethinking Wine Tasting for Chinese Consumers: A Service Design Approach Enhanced by Multimodal Personalization. In 2025 International Conference on Content-Based Multimedia Indexing (CBMI) (pp. 1-5). IEEE.
[14] Tang, Yingheng, et al. "Design and Optimization of Shallow-Angle Grating Coupler for Vertical Emission from Indium Phosphide Devices." (2020).
[15] Sun, Lingxin. "Designing Inclusive Interfaces: Accessibility Challenges and Solutions in Digital Products." Proceedings of the 2025 International Conference on Artificial Intelligence and Sustainable Development. 2025.
[16] Zhou, Z. (2025, November). Digital precision distribution strategy for social media content on private domain platforms in the automotive industry: a collaborative filtering model based on user behavior. In Proceedings of the 2025 International Conference on Digital Society and Intelligent Computing (pp. 516-521).
[17] Yang, X., Zheng, X., & Lu, Q. (2025, October). Construction and early warning of multi-dimensional network credit-related transaction risk maps by integrating graph neural network (GNN). In Proceedings of the 2025 2nd International Conference on Digital Economy and Computer Science (pp. 919-923).
[18] Yuan, M., Liu, J., Yang, J., Li, X., Yan, W., Wu, Y., & Liang, P. (2026, March). TA-Mem: Tool-Augmented Autonomous Memory Retrieval for LLM in Long-Term Conversational QA. In 2026 9th International Conference on Advanced Algorithms and Control Engineering (ICAACE) (pp. 2684-2688). IEEE.
[19] Yang, J., Wu, Y., Liu, J., Liang, P., Yuan, M., Li, X., & Yan, W. (2026). Recursive Multi-Agent Trading System: Iterative Optimized Portfolio Strategy Under Geopolitical Uncertainty. arXiv preprint arXiv:2605.25311.