DOI: 10.5604/20830157.1130179

EFFECTIVE EXPECTATION MAXIMIZATION ALGORITHM IMPLEMENTATION USING MULTICORE COMPUTER SYSTEMS

Alexei Kasitskij[1], Peter Bidyuk[1], Alexander Gozhyi[2]

[1]National Technical University "Kyiv Polytechnic Institute", Institute for Applied System Analysis, [2] Petro Mohyla Black Sea State University, Department of Information Technologies and Program Systems

Abstract. *A popular expectation maximization algorithm that is widely used in modern data processing systems to solve various problems including optimization and parameter estimation is considered in the paper. The task of the study was to enhance effectiveness of the algorithm execution in time. An enhancement of execution rate for the EM algorithm using multicore architecture of modern computer systems was carried out. Necessary modifications aimed at better parallelism were proposed for implementation of the EM algorithm. An efficiency of the software implementation was tested on the classic problem of Gaussian random variables mixture separation. It is shown that in the mixture separation problem EM algorithm performance degrades when the distance between mean values of distributions is less than three standard deviations, which is totally in the spirit of three sigma law. In such cases, it is very important to have an efficient EM algorithm implementation to be able to process such test cases in a reasonable time.*

Keywords: expectation maximization algorithm, multicore architecture, parallelism, problem of Gaussian random variables mixture separation, three sigma law

EFEKTYWNY ALGORYTM EM I JEGO IMPLEMENTACJA PRZY UŻYCIU WIELORDZENIOWYCH SYSTEMÓW KOMPUTEROWYCH

Streszczenie. *W artykule opisany jest popularny algorytm EM (expectation maximization), który jest powszechnie stosowany w nowoczesnych systemach przetwarzania danych do rozwiązywania różnych problemów, w tym optymalizacji i estymacji parametrów. Celem badań było zwiększenie efektywności czasu wykonywania algorytmu. Zwiększenie szybkości wykonania algorytmu EM użyto wielordzeniowy architektury nowoczesnych systemów komputerowych. Zostały zaproponowane niezbędne modyfikacje mające na celu lepszą równoległość realizacji algorytmu EM. Skuteczność implementacji programu była testowana na klasycznym problemie separacji Gaussowskich zmiennych losowych. Wykazano, że w przypadku rozdziału mieszaniny wydajność algorytmu EM ulega degradacji, kiedy odległość między średnimi wartościami rozkładu wynosi mniej niż trzy odchylenia standardowe, co jest całkowicie zgodnie z regułą trzech sigm. W takich przypadkach, jest bardzo ważne, aby mieć efektywną realizację algorytmu EM móc przetworzyć takie przypadki w rozsądnym czasie.*

Słowa kluczowe: Algorytm EM, architektury wielordzeniowe, równoległość, separacji Gaussowskich zmiennych losowych, reguła trzech sigm

Introduction

Fast developments in the area of modern informational computer technologies require new methods and techniques for data processing, implementation of data mining systems for large databases processing, and for solving many other practical problems. Today especially important and useful are new methods for solving optimization problems as well as the tasks of mathematical modeling, image and speech recognition, forecasting and control. All these procedures usually create a kernel of data processing units in modern informational decision support systems (IDSS) [3, 7].

An expectation maximization algorithm (EM algorithm) is widely used in mathematical and applied statistics, optimization theory and its multiple applications for computing unknown model parameters, imputing lost measurements, finding the minima and maxima values for various functions etc. One of its applications is directed towards maximum likelihood estimation of unknown parameters for probabilistic models including the cases when some variables cannot be measured directly.

The algorithm is functioning iteratively in two steps. At the E-step (expectation step) an expected value of likelihood function is computed using current approximation of non-measurable variables. The M-step is used for computing the model parameter estimates that maximize the expected likelihood generated at the E-step. The EM algorithm is also often used for data clustering, machine learning, in computer vision systems and natural language processing (known as special case: Baum-Welch algorithm). Due to the possibility of functioning in the conditions of lost data the EM algorithm creates very useful instruments for portfolio risks estimation in analysis of various financial data [1]. Among other known applications are medical image processing: for example, positron emission tomography and single-photon emission computed tomography.

We use the EM algorithm to find the maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. Typically these models involve latent variables in addition to unknown parameters and known data observations. That is, either there are missing values among the data, or the model can be formulated more simply by assuming the existence of additional unobserved data points.

1. Problem statement

A mixture separation problem can be defined as follows. Given a set of N points in a D-dimensional space, $x_1, x_2, ..., x_N \in R^D$ and a family of probability densities F on R^D, find the probability density, such, that the probability of generating the set, $x_1, x_2, ..., x_N$, using this density is maximized. The common technique for definition of the probability density family is assigning densities to construct a common algebraic form with different parameters θ [4, 5].

2. Problem solution

As shown in [5], this problem has the following solution:

$$m_k = \frac{\sum_{n=1}^{N} p(k|n) x_n}{\sum_{n=1}^{N} p(k|n)}$$

$$\sigma_k = \sqrt{\frac{1}{D} \frac{\sum_{n=1}^{N} p(k|n) \| x_n - m_k \|^2}{\sum_{n=1}^{N} p(k|n)}}$$

$$p_k = \frac{1}{N} \sum_{n=1}^{N} p(k|n) \, .$$

The first two equations are quite easy for accepting, because m_k and σ_k are the sample mean and the sample variance, weighted by a probability of a given point to be generated from the k-th model. The third equation for the mixture is not so obvious. Since the values of p_k are obtained as a sample mean of a conditional probabilities $p(k|n)$, third equation is not so hard to interpret.

3. Iterative computing procedure

These equations are strongly connected, because $p(k|n)$ on the right sides of both equations are dependent on all variables on the left side. This makes it hard to directly solve these equations. Nevertheless, EM-algorithm allows us to build an iterative procedure for solving this system [8].

EM-algorithm is an iterative process, with the following steps:

1) E-step: estimate conditional expectation

$$Q(\theta|\theta_n) = E_{Z|X,\theta_n}\{\ln P(X,z|\theta)\}.$$

2) M-step: maximization of the Q in relation to θ

$$\theta_{n+1} = \arg\max_{\theta} Q(\theta|\theta_n)$$

Computations of the sum of current membership probabilities for each point, conditional membership probability calculation loop, expectation estimation loop and variance estimation loop, ware parallelized using Open MP technology. Program implementation in C++ is done using Open MP, which allows to analyze algorithm efficiency can be tested experimentally.

To validate the correctness and measure the performance for all tests we used the following two simple metrics:

$$Eps = \frac{1}{K}\sum_{k=1}^{K}(m_k - m_k^*)^2 + \sum_{k=1}^{K}(\sigma_k - \sigma_k^*)^2,$$

and ***Time*** – the wall-clock time of the program execution.

All tests were executed on a PC with Intel Core i7 2600k CPU. This CPU has 4 cores with Hyper Threading technology, which makes it possible to execute 8 parallel threads.

Table 1. Characteristics of EM algorithm for separating mixtures of two Gaussians (single thread)

No.	α	Eps	Time, s
1	1024	1.36e-6	0.359
2	256	3.99e-6	0.359
3	64	5.09e-6	0.359
4	16	6.82e-7	0.312
5	4	1.80e-6	0.530
6	1	4.04e-6	4.680
7	0.25	6.21e-5	867.0

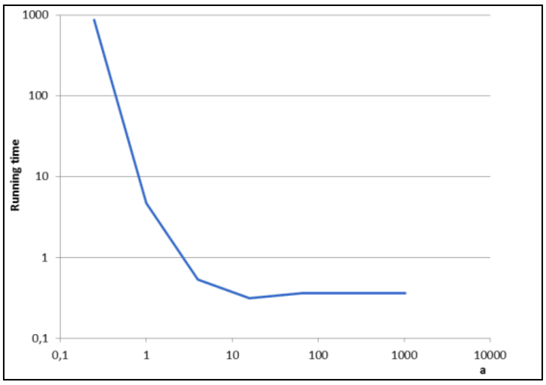


Fig. 1. EM algorithm running time for single thread implementation

As one can see, EM algorithm keeps good precision even when the distance between means is less than the variance. From the other side, algorithm running time increases exponentially in such cases (Tab. 1, Fig, 1). Now take a closer look at the value range of α, which are less than four.

Table 2. Characteristics of EM algorithm for separating mixtures of two Gaussians (single thread, and $\alpha \le 4$)

No.	α	Eps	Time,s
1	4	1.37e-6	0.421
2	2	1.13e-5	2.839
3	1	1.39e-5	4.696
4	½	1.13e-4	48.98
5	¼	1.34e-3	1857
6	1/8	3.66e-3	3115
7	1/16	3.15e-3	34469

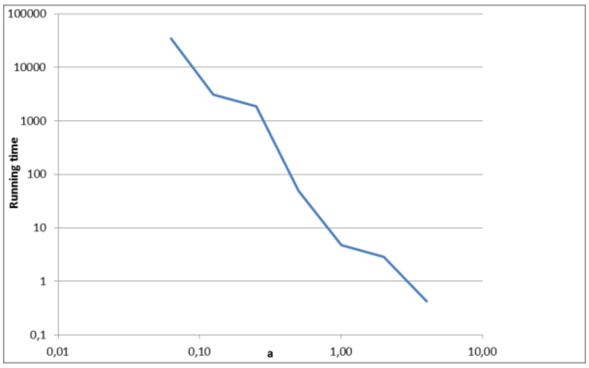


Fig. 2. EM algorithm running time

As shown in the Tab. 2 and Fig. 2, algorithm performance degrades when the distance between mean values is less than 3 standard deviation values, which can be easily explained with a three sigma rule. Nevertheless, algorithm precision degrades slowly, while the running time increases exponentially.

4. Separation of two Gaussians with equal means and different variances

Separation of two Gaussians mixture with equal means $m_1 = -m_2 = \alpha$, and different variance values $\sigma_1 = \sigma_2 = 1$; sample size is 1000000 (Table 3, Fig. 3). Algorithm works in a single thread. $m_1 = m_2 = 0$ (single thread)

Table 3. EM algorithm characteristics in a case $m_1 = m_2 = 0$ (single thread)

No.	*S*	Eps	Time,s
1	64	3.84e-3	3.051
2	32	1.36e-3	3.429
3	16	2.24e+2	5.802
4	8	1.94e-6	6.640
5	4	9.01e+0	9.757
6	2	9.98e-1	11.57
7	1	2.14e-4	33.564

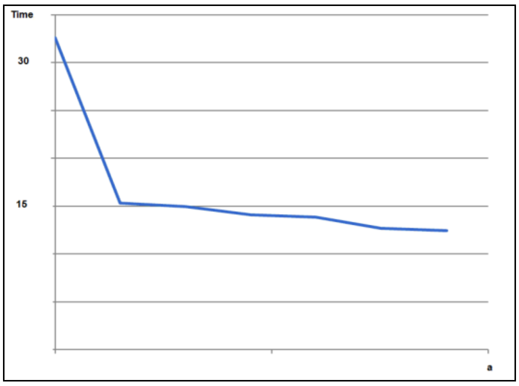


Fig. 3. EM algorithm running time in a case

5. Multithreaded algorithm version

Multithreaded version of the algorithm was implemented using Open MP technology [9, 10]. Similar tests will be used for testing multithreaded version of the algorithm. The only difference is that threads used (***Threads#***) column will replace distribution parameters (*a*, *s*) column.

6. Separating mixture of two gaussians with equal variance and different means which are far apart

Separating the mixture of two Gaussians with equal variance $\sigma_1 = 1, \sigma_2 = s$, and different means m_1=-m_2=100; sample size is 100 000 000 (Tab. 4, and Fig. 4).

Table 4. EM algorithm characteristics in multithreaded version

No.	Threads#	Eps	Time, s
1	1	3.5e-8	43.66
2	2	3.5e-8	24.16
3	3	3.5e-8	19.20
4	4	3.5e-8	16.14
5	5	3.5e-8	14.36
6	6	3.5e-8	13.38
7	7	3.5e-8	12.87
8	8	3.5e-8	12.46

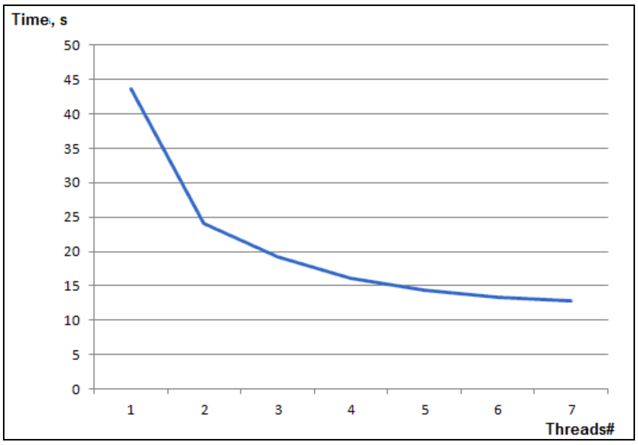


Fig. 4. EM algorithm running time for multithreaded version

7. Effectiveness of parallelization

All tests in this paragraph were done with hiring as a data easy separable mixtures, what is meaning that the distances between mean values were significantly greater than standard deviation. The mixture of two Gaussians with equal variance $\sigma_1=\sigma_2=1$, and different means $m_1=-m_2=100$; sample size is 100 000 000 (Fig. 5).

Parallelization effectiveness [1] is calculated to the relation of single-threaded program running time with formula

$$E=\frac{R}{p}=\frac{T_1}{pT_p}.$$

For higher quality of time measurements all calculations were performed 10 times, and then averaged.

As shown in Fig. 5, EM algorithm parallelization effectiveness drops with the increase of threads number. This is caused by the fact that EM-algorithm is using large amount of memory and this impacts performance when the number of used cores is greater than the number of memory channels.

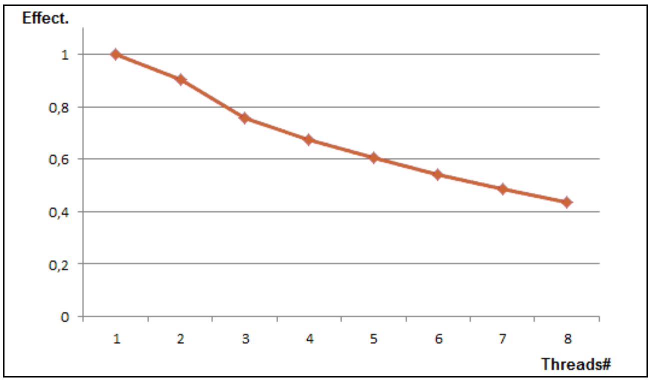


Fig. 5. EM algorithm parallelization effectiveness

Conclusions

The problem of random variables mixtures separation was considered with the use of expectation maximization algorithm running on fast processors. Running time estimations were determined and presented as a result of computing experiments. As an example a Gaussian mixture separation quality was measured for different sizes of the mixture samples.

It is shown that in the mixture separation problem performance of the EM algorithm constructed degrades when the distance between mean values of mixture clusters was less than three standard deviations, what is totally in spirit of the three sigma law. In such cases, it is very important to have an efficient EM algorithm implementation to be able to process such test cases in a reasonable time. To measure effectiveness of the computer C++ program parallelization we used the expression

$$E=\frac{T_1}{pT_p},$$

where E is effectiveness; p – number of used cores; T_1, T_p are program running times on one and p cores respectively. As the carried out experiments show, EM algorithm parallelization effectiveness is almost linear (E = 0.9) when the number of cores is not greater than memory channels, and degrades with the increase of used cores number.

It is reasonable to direct the future studies to the implementation of selected EM algorithms on other types of modern computing systems, say popular and not expensive graphic processors.

References

[1] Ben-Ari M.: Principles of Concurrent and Distributed Programming (2nd ed.). Addison-Wesley, 2006.

[2] Bidyuk P. I., Gozhij O. P., Korshevnyuk L.O.: Computer based decision support systems. Chornomorsky State University named after Petro Mogyla, Mykolaiv, 2012.

[3] Borman S.: The expectation maximization algorithm a short tutorial. http://www.seanborman.com/ publications /EM_algorithm.pdf

[4] Chapman B., Jost G. R., Kuck D. J.: Using Open MP: Portable Shared Memory Parallel Programming. The MIT Press, Boston 2007.

[5] Dellaert F.: The expectation maximization algorithm. Techn. paper, Georgia Institute of Technology, 2002.

[6] Dinov I. D.: Expectation maximization and mixture modeling tutorial. http://www.stat.ucla.edu/~dinov/courses_students.dir/04/Spring/Stat233.dir/STAT233_notes.dir/EM_Tutorial.pdf

[7] Hollsapple C. W., Winston A. B.: Decision support systems. West Publishing Company, New York 1996.

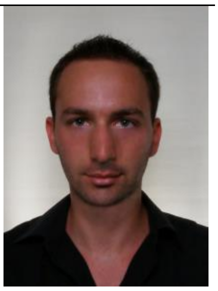
[8] Korbicz J.(Ed.): Measurements, models, systems and design. Wydawnictwa Komunikacji i łączności, 2007.

[9] Quinn M. J.: Parallel Programming in C with MPI and Open MP. McGraw-Hill Inc. 2004.

[10] Shuicheng Y., Chang S. F., Johnson M. H., Xi Z., Xiaodan Z., Huang T. S.: Sift-bag kernel for video event analysis. Proceeding of the 16th ACM international conference on Multimedia, 2008, 229-238.

M.Sc. Alexei Kasitskij
e-mail: pbidyuke_00@ukr.net

National Technical University "Kyiv Polytechnic Institute", Institute for Applied System Analysis.
Postgraduate student at Institute for Applied System Analysis. Currently his research interests include, fundamentals of Artificial Intelligence systems, theory of algorithms, forecasting methods, probabilistic and statistical modeling.



D.Sc. Petr Bidyuk
e-mail: pbidyuke_00@ukr.net

National Technical University "Kyiv Polytechnic Institute", Institute for Applied System Analysis.
His research interests areas include multicriteria optimization, multiobjective optimization, Methods for analysis of nonlinear systems, forecasting methods, probabilistic and statistical modeling.

Ph.D. Alexander Gozhyi
e-mail: alex_daos@mail.ru

Petro Mohyla Black Sea State University, department of information technologies and program systems. Associate Professor, Ph.D., Dean of Computer Science Faculty at Petro Mohyla Black Sea State University. Currently his research interests include multicriteria optimization, functional programming, fundamentals of Artificial Intelligence systems, theory of algorithms.

otrzymano/received: 2014.08.19 *przyjęto do druku/accepted: 2014.11.10*