

基于 Fork/Join 框架的并行期望最大算法实现与仿真

孙晓静¹ 王天赐² 关智超² 朱红磊^{2*}

SUN Xiaojing WANG Tianci GUAN Zhichao ZHU Honglei

摘要

为解决期望最大算法(expectation-maximum, EM)在处理大规模数据时存在计算时间开销过长问题,文章设计了一种基于 Fork/Join 框架实现并行多任务快速处理大规模数据的 EM 方法。该方法引入并应用 Fork/Join 在 EM 算法的每一次迭代过程中,通过创建 ForkJoinTask 任务和线程池,采用对任务进行拆分和并行处理方式,实现数据的并行计算处理。实验结果表明,采用所设计方法在计算似然估计和期望结果时,比常规 EM 算法减少计算时间约 2/3,极大提高了算法处理大规模数据的效率,降低了时间开销。

关键词

期望最大;并行多任务;Fork/Join 框架;大规模数据;仿真

doi: 10.3969/j.issn.1672-9528.2026.04.029

0 引言

期望最大化(expectation-maximum, EM)算法是一种通过迭代进行极大似然估计的优化算法,广泛应用于对含有隐变量的概率密度模型(如混合高斯模型)的参数估计^[1-2]。当数据缺失、隐变量存在等数据不完整,有未知模型参数、无法直接用极大对数似然函数求解时,便需使用期望最大算法^[3-4]。EM 算法对处理含隐变量的数据有非常好的效果,通过迭代收敛即可得到最终模型参数,是最常见的隐变量估计算法,在机器学习、数据挖掘中有着广泛用途^[5-11]。EM 算法在多场景下具有较好结果,但是当进行含有隐变量的参数估计时,若样本数据量过大,计算运行耗时较长。因此,研究如何提高算法计算效率,减少运行时间具有重要意义,针对 EM 算法在处理大规模数据时耗时过长问题,常用的思想是联合多处理机、结合并行算法实现任务的并行处理,加快处理速度^[12-15]。如文献[16]提出一种并行参数 EM 算法来学习具有缺省数据的贝叶斯网络参数,实验表明可有效降低参数学习的时间复杂性。多处理器并行计算可缩短任务执行时间,提高处理机的利用率和任务执行效率,目前 Java 中的 Fork/join 是高并发处理任务的并行框架^[17-18],可充分利用多核处理机的优势,提高计算处理效率,但目前尚未使用 Fork/join 并行处理提升 EM 算法效率的成果。因此,本文提出一种基

于并行多任务的 EM 算法,该算法结合 Fork/join 框架,在迭代过程中,通过对任务进行拆分和并行处理的方式实现提升计算效率,解决了 EM 算法处理大规模数据运行耗时较长问题。

1 基础知识

1.1 EM 算法

期望最大化算法是一种从无标签数据中求解模型参数或隐变量的估计方法,它首先对模型随机生成参数;然后,根据上一步所得的隐变量值重新对模型参数估计,再将得到的模型参数进行隐变量估计,对该过程反复迭代,直至参数收敛,从而实现对数据参数最优求解。其中,样本分析处理过程主要分为两步:E 步,初始化参数,极大似然估计隐变量,计算期望;M 步,利用 E 步计算的期望,更新模型参数。对存在隐变量或未知参数的数据,极大似然估计无法直接处理,而 EM 算法通过初始化参数,结合现有数据计算极大似然估计,估计隐变量并重新计算分布从而更新参数,通过持续迭代不断优化参数得到最终解。

1.2 Fork/Join

Fork/Join 是 JAVA 的 JUC 并行包中用来并行执行任务的框架。该框架将大任务拆分为多个小任务,拆分的子任务在线程池中并行执行,通过创建多个任务队列由不同线程并行执行,同时利用工作窃取算法,让空闲线程获取其他线程中未完成任务并执行,最后把子任务执行结果汇总得到最终结果,充分利用多核处理器的优势,提高 CPU 利用率,加快任务执行。

Fork/Join 框架实现对任务拆分并通过多线程并行执行子任务,充分利用多核处理机的空闲核,提高任务处理效率

1. 中国核电工程有限公司郑州分公司 河南郑州 450002

2. 河南中医药大学信息技术学院 河南郑州 450046

[基金项目] 国家自然科学基金项目“一类新型并行系统互连网络及其关键性质的研究”(61902113);河南省高等教育教学改革研究与实践项目“面向河南省高校‘中医药+人工智能’交叉学科高质量发展研究与实践”(2025SJGLX220Y)

和 CPU 利用率。对于大型任务，可调用 Fork/Join 框架实现并行处理任务，减少任务处理的时间消耗。因此，本文应用该框架设计并行 EM 算法，并实例测试对比原始算法与并行 EM 算法性能。

2 并行改进期望最大算法设计

2.1 设计思想

期望最大算法通过循环迭代策略，有效处理含隐变量的样本数据、估计模型参数，在最大期望算法实际应用时，大规模或超大规模样本数据易造成处理时间过长，因此，利用 Fork/Join 框架通过将期望最大化算法中任务分割，实现任务的并行处理，减少算法运算时间。

2.2 实现步骤

并行期望最大化算法处理数据步骤如算法 1 所示。

算法 1 并行期望最大化算法步骤

Step1: 输入样本数据，随机初始化模型参数 θ^0 。

Step2: 循环，假设最大迭代次数 j :

(a) 创建 ForkJoinTask 任务，设置任务拆分阈值并继承相应类。

if(问题规模 < 阈值) // 阈值默认为 1000

 覆写并执行 compute() 方法处理数据，

 即 E 步:

 (i) 由模型参数 θ^j 计算极大似然估计求隐变量值和期望

 (ii) 将部分期望值及相关统计量保存，为后面更新参数处理。

else

 将该任务拆分为两个子任务，调用 invokeAll() 或 Fork() 方法将子任务提交至线程池并行执行，后汇总子任务处理所得期望值及相关统计量。

(b) M 步: 参数更新，得到 θ^{j+1} 。汇总 compute() 方法每个子任务完成后所得的部分隐变量值和期望值，统一归化处理，得到 θ^{j+1} ，实现参数更新。

(c) 如果 θ^{j+1} 收敛或达到最大迭代次数，则算法结束。否则回到步骤 (a) 继续迭代。

Step3: 输出模型参数 θ 。

使用 EM 算法处理大规模数据，每次迭代求解期望 (E 步) 需要进行大量运算，该过程最为耗时，是造成算法处理时间过长的主要原因，所以对此进行改进: 引入 Fork/Join 框架，创建任务对象，设置任务分割阈值，若任务足够小，则直接

执行任务，计算并返回期望及相关统计量; 若任务规模大于阈值，则根据当前任务规模，新建两个子任务，子任务规模为当前任务规模的一半，实现任务拆分，然后调用 Fork() 或 invokeAll() 方法将子任务提交至线程池，并行处理子任务，最后合并子任务结果并返回。子任务处理结果是部分数据的期望，不能进行参数更新，因此需汇总所有子任务处理结果。并行处理得到期望及相关统计量后，执行 M 步: 统一归化处理，更新参数。改进算法将并行处理应用在每次 EM 的迭代运算中，实现数据并行计算，同时充分调用空闲处理机，加快任务处理速度，减少数据的计算处理时间，大幅提高了计算效率。

3 实验仿真

为测试改进并行 EM 算法在数据处理时的性能，进行仿真实验如下:

实验环境: 软件工具 Eclipse, jdk 版本 1.8。使用三个测试集数据，规模分别为测试集一 (412 行 $\times$ 88 列)，测试集二 (601 行 $\times$ 114 列)，测试集三 (1 087 行 $\times$ 130 列)，三台计算机，机器 1 (联想 Y7000, 英特尔 i7-8750H)，机器 2 (台式 ThinkCentre, 英特尔 i5-2400)，机器 3 (英特尔 i7-8550U)。

实验内容: 设计 JAVA 程序，程序一为经典 EM 算法处理数据; 程序二为改进并行处理 EM 算法，对程序设置计时函数，统计数据处理时间。在三个不同型号机器上分别使用两种方法对小中大三种规模数据集进行处理，对计算运行耗时进行对比分析。

测试集格式如表 1 所示，每一行代表一名患者，每列代表患者该症状的表现有无 (0 表示无此症状，1 表示有此症状)。对测试集可进行优化处理，若存在重复行则只记录为一行，并添加 counts 记录重复数，处理重复行可优化数据集，便于 EM 算法处理。

表 1 测试数据集格式

	症状 1	症状 2	症状 3	...	症状 n
患者 1	0	1	1	...	0
患者 2	1	0	1	...	1
患者 3	1	1	0	...	1
...	...	...	...	...	...
患者 m	1	0	1	...	0

程序一使用经典 EM 算法处理数据; 程序二使用改进并行 EM 算法处理，其中相关条件设置如下:

(1) 创建 ForkJoinTask 任务继承 RecursiveTask 类。

(2) 数据集总行数作为任务规模大小，以行为标准对

数据集拆分处理，设置任务分割阈值为 12。

为验证 Fork/Join 框架的并行处理任务时调用 CPU 的利用率，在机器 1 处理测试集一时，记录 60 s 内 CPU 利用率变化作为对比，如图 1~2 所示。

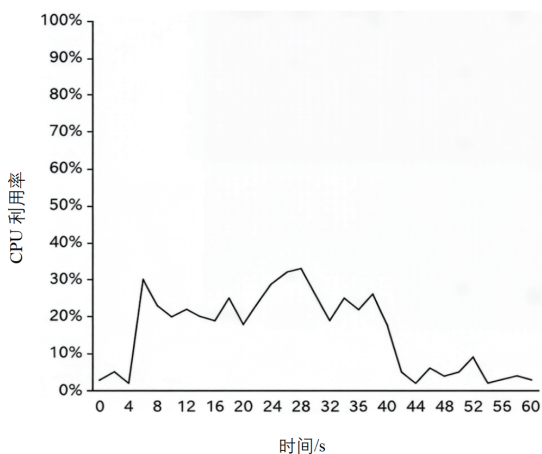


图 1 经典 EM 算法 CUP 利用率

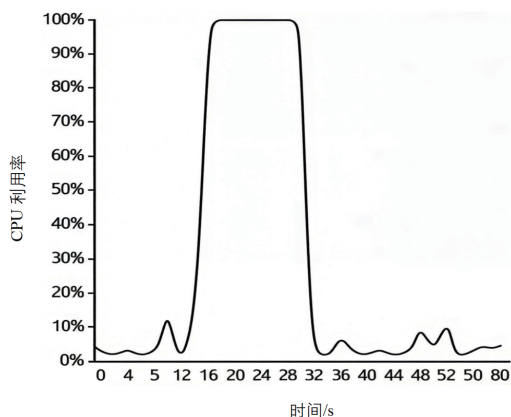


图 2 并行 EM 算法 CPU 利用率

通过图 1~2 可以看出在程序运行过程中，使用 Fork/Join 框架的并行 EM 算法在任务处理期间 CPU 利用率高达 100%，充分调动所有空闲处理机，而经典 EM 算法处理时的 CPU 利用率仅为 25% 左右，并行与非并行处理数据时 CPU 利用率有较大差别，并行 EM 算法显著提高了 CPU 利用率和效率，实现了任务并发处理。

为验证改进并行 EM 算法在处理不同规模数据时的差异，对上述三台电脑分别使用改进并行 EM 算法和经典算法，在三种不同规模数据集上进行仿真对比测试。得到结果如表 2 所示，三台机器在不同测试集的表现分别如图 3~5 所示。经典 EM 算法处理数据时，需单线程逐行计算极大似然估计和期望并更新参数，而并行 EM 算法通过 Fork/Join 框架算法将该数据集按行拆分处理，使得每个子任务的行数均小于 12（阈值），拆分的多个子任务在多核处理机上并行执行，充分利用空闲处理机，从而缩短程序运行的时间。

表 2 三台机器在不同规模数据集的测试对比

程序处理		机器 1 测试时长 /s	机器 2 测试时长 /s	机器 3 测试时长 /s
测试集一	经典 EM 算法	32	41	32
	并行 EM 算法	7	12	9
测试集二	经典 EM 算法	97	124	102
	并行 EM 算法	23	42	25
测试集三	经典 EM 算法	617	910	898
	并行 EM 算法	183	263	312

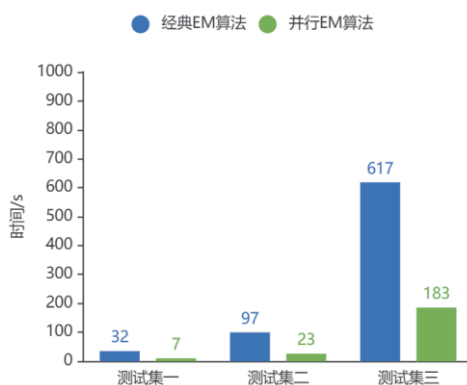


图 3 机器 1 并行测试结果对比

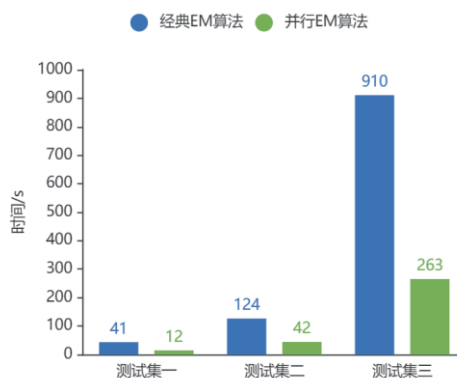


图 4 机器 2 并行测试结果对比

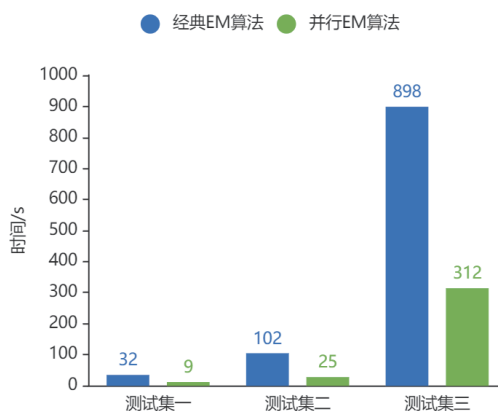


图 5 机器 3 并行测试结果对比

结合图 3~5 可知,当测试集数据规模依次增大,改进并行 EM 算法比经典 EM 算法处理数据所消耗时间明显减少,且随着数据规模增大,改进并行 EM 算法处理数据效果越来越好。在数据集三的测试上,改进并行 EM 算法的计算效率约是经典算法的三倍,因为改进并行算法能减少程序分析处理的时间,在实际的数据分析应用中,其计算效率也将较大提升。

4 结语

本文设计一种 Fork/Join 应用在期望最大算法的并行多任务处理数据方法,对期望最大算法在每次迭代计算似然估计和期望时将处理数据集任务拆分,利用空闲的处理机并行处理任务,实现并行多任务快速处理。改进的方法充分利用系统资源,大大减少了程序运行时的时间消耗,解决了期望最大算法在处理大数据任务时的时间开销过大问题。仿真测试显示,与经典 EM 的单线程处理相比,使用改进并行期望最大算法在不同机器处理不同规模数据时,能大大减少数据处理时间,显著提高算法的计算效率。

参考文献:

- [1] 刘铭,于子奇.一种改进的期望最大化算法[J].吉林大学学报(理学版),2022,60(5):1176-1182.
- [2] 王中钰,钱晓东.基于改进期望最大化算法的供应链网络边连接规则优化[J].计算机应用,2024,44(11):3386-3395.
- [3] 陈昊,杜新纲,葛得辉,等.基于期望最大化算法的电能表误差估计技术[J].中国测试,2023,49(3):47-52.
- [4] 褚润聪,武岩波,朱敏.模式耦合期望最大向量近似消息传递频域迭代均衡[J].声学学报,2025,50(4):1015-1030.
- [5] 宋禹昕,常中宇,高进,等.采用期望最大化算法的半滑舌鲷性逆转性状高效遗传解析[J].山东农业大学学报(自然科学版),2024,55(4):531-539.
- [6] 刘劲宏,吴晨韵,徐劲,等.基于期望最大算法的空间事件及异常值探测[J].空间科学学报,2022,42(6):1185-1192.
- [7] 孙家豪,刘策伦.一种基于 EM 算法的码辅助载波同步算法[J].计算机仿真,2024,41(10):306-310.
- [8] 陈志,白思俊,郭云涛.随机活动工期下求解资源约束项目最大期望净现值的动态调度算法[J].运筹与管理,2022,31(6):17-24.
- [9] 石怡萱.面向 DNA 大数据集的改进期望最大化模型发现算法研究[D].西安:西安电子科技大学,2024.
- [10] 魏语轩,李昕闻,陈兴国.基于效用函数的期望最大搜索算法[J].软件导刊,2023,22(8):86-92.
- [11] 朱小婷,张君,王璐,等.基于小波奇异特征约束的期望最大时延估计算法[J].兵工学报,2024,45(4):1108-1116.
- [12] 万开成.大数据处理中的并行计算与分布式存储技术研究[J].信息与电脑(理论版),2024,36(19):166-168.
- [13] 吴宗明,曹继军,汤强.基于深度强化学习的在线并行 SDN 路由优化算法研究[J].计算机科学,2025,52(S1):795-803.
- [14] 张菊,高国灵,王乾伟.并行处理技术在场监雷达录取器中的应用[J].科技创新与应用,2025,15(7):192-196.
- [15] 罗爱宝.梯度下降类和 EM 类迭代算法的并行化研究[D].南京:南京大学,2012.
- [16] 俞奎,王浩,吴信东,等.贝叶斯网络的并行 EM 学习算法[J].模式识别与人工智能,2008,21(5):670-676.
- [17] 陈永青.浅析 JDK1.7 中的 Fork/Join 框架[J].中国新通信,2016,18(19):94-95.
- [18] 鲍婷婷,焦圣明,殷笑茹,等.基于 Fork/Join 框架的等值面快速生成并行算法[J].计算机技术与发展,2020,30(3):187-193.

【作者简介】

孙晓静(1986—),女,河南郑州人,硕士研究生,高级研究员,研究方向:控制理论、高性能计算。

王天赐(2005—),男,河南周口人,在读本科,研究方向:人工智能。

关智超(2005—),男,河南焦作人,在读本科,研究方向:人工智能。

朱红磊(1983—),通信作者(email:zhuhonglei@hactcm.edu.cn),男,河南周口人,博士研究生,讲师,研究方向:中医药人工智能。

(收稿日期:2025-09-12 修回日期:2026-04-17)