

引用格式:李仲,吴玉平,陈岚,等. 基于机器学习的版图热点检测并行算法[J]. 微电子学与计算机, 2019, 36(1): 27-31. [LI Z, WU Y P, CHEN L, et al. Parallel Algorithm in Hotspot Detectionbased on Machine Learning[J]. Microelectronics & Computer, 2019, 36(1): 27-31.]

基于机器学习的版图热点检测并行算法

李 仲^{1,2,3}, 吴玉平¹, 陈 岚¹, 张学连¹

(1 中国科学院 微电子研究所, 北京 100029; 2 三维及纳米集成电路设计自动化技术
北京市重点实验室, 北京 100029; 3 中国科学院大学, 北京 100049)

摘 要: 针对基于机器学习的版图热点训练过程中降维算法耗时长和大多数训练算法没有利用多核资源的问题, 分别提出了基于 MPI 的 PCA 并行降维算法和基于 OpenMP 的 AdaBoost 并行训练算法. 首先采用 QR 分解优化奇异值求解特征矩阵, 再结合 MPI 实现 PCA 的并行降维计算, 最后将降维后的数据利用多核 CPU 进行训练, 达到减小训练时间的目的. 实验结果表明, PCA 并行降维算法加速比达 4.7 倍, AdaBoost 并行训练算法加速比达 4.9 倍, 验证了并行化的可行性.

关键词: 版图热点; PCA; AdaBoost; 并行计算

中图分类号: TN401

文献标识码: A

文章编号: 1000-7180(2019)01-0027-05

Parallel Algorithm in Hotspot Detection Based on Machine Learning

LI Zhong^{1,2,3}, WU Yu-ping¹, CHEN Lan¹, ZHANG Xue-lian¹

(1 Institute of Microelectronics, Chinese Academy of Sciences, Beijing 10029, China;
2 Beijing Key Laboratory of Three-Dimensional and Nanometer Integrated Circuit Design Automation Technology,
Beijing 100029, China; 3 University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: The training phase of hotspot detection based on Machine Learning is time-consuming. In this paper, a parallel PCA algorithm based on MPI and a parallel AdaBoost algorithm based on OpenMP are proposed to reduce the training time. Firstly, the QR decomposition is used to optimize SVD, and then the MPI is used to implement the PCA parallel algorithm. Finally, the reduced dimension data is trained using multi-core CPU parallel algorithm. The experimental results show that the PCA parallel algorithm could achieve a 4.7 times speed-up ratio, and the AdaBoost parallel algorithm could achieve a 4.9 times speed-up ratio, which verifies the feasibility of parallelization.

Key words: hotspot detection; PCA; AdaBoost; parallel computing

1 引言

近年来,随着深亚微米工艺开发,版图图形之间的相互影响变得复杂,版图热点检测不能用简单的设计规则处理时,研究者们开始将机器学习的方法引入到版图热点检测当中,但由于版图热点类型多且复杂,往往不能得到高质量的分类模型,使得检测结果的准确率较低. 为了简化版图热点数据,文献

[1]提出了一种基于密度的特征提取算法,文献[2]对热点数据进行了分类处理从而有针对性的进行特征提取和分类训练,文献[3]则在特征提取之后利用 PCA 降维算法进一步简化热点数据,虽然准确率得到了较好的提高,但随着对版图热点数据简化处理的步骤增多,加上版图检测中的机器学习训练方法大都是基于 CPU 的串行算法,这就导致了大量的时间耗在机器学习的训练阶段. 本文通过分析已有

收稿日期: 2018-04-19; 修回日期: 2018-05-20

基金项目: 北京市科技专项: 科技新星与领军人才(Z171100001117147)

的研究成果,结合目前流行的并行计算模型,分别提出了基于 MPI 的 PCA 并行降维算法和基于 OpenMP 的 AdaBoost 并行训练算法。

2 AdaBoost 训练算法

AdaBoost (Adaptive Boosting) 是 Freund 和 Schapire 在 1995 年提出的一种可将弱分类器提升为强分类器的机器学习算法^[4]。AdaBoost 训练算法是一个弱分类器逐步提升的过程,即针对同一个训练集训练不同的弱分类器,然后将这些弱分类器最终组合成的一个强分类器。

在版图热点检测中,AdaBoost 算法与其他常见的机器学习训练算法相比,不用担心支持向量机 (SVM) 出现过拟合的问题,没有人工神经网络 (ANN) 复杂的计算且易于实现。本文选择 AdaBoost 作为训练算法,具体算法描述如下:给定训练样本集 $D(X, Y) = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ 和训练次数 T , 其中 $x_N \in X$ 是样本的特征属性, $y_i \in \{-1, +1\}$ 是样本的分类标签, 分别代表非版图热点和版图热点。

(1) 初始化样本权重 $D_1(x) = 1/N$;

(2) 进行 $t = 1, \dots, T$ 轮训练

① 训练一个弱分类器 h_t ;

② 估计 h_t 的误差

$$\epsilon_t = P(h_t(x_i) \neq f(x));$$

③ 计算分类器 h_t 的权重

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right);$$

④ 更新样本权重

$$D_{t+1}(x) = \frac{D_t(x) \exp(-\alpha_t f(x) h_t(x))}{z_t};$$

(3) 输出强分类器

$$(H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x)));$$

由算法描述我们可以了解到,AdaBoost 算法的主体部分是特征值选取的迭代过程,而迭代过程是对数据集中不同的元素进行相同计算的过程,就可以分配到不同线程上进行并行化计算。

3 数据处理

3.1 定位热点

首先我们需要知道版图热点的位置信息,因版图样本的准备过程不同,部分学者利用 Calibre™ 软件先对版图进行 OPC 处理及光刻仿真,然后定位出其中的版图热点。版图样本本质上是由多个多边形

组成,本文利用多边形相互之间的投影,找到多边形之间的间距小于规定最小间距的冲突区域来定位版图热点。

3.2 拓扑分类

版图热点的类型有很多,如果不加以分类进行训练,会得到比较单一的分类模型,一定程度上降低了分类模型的质量。所以,有必要将具有相似几何特征的版图图形分类在一起,从而有针对性的训练分类模型。此外,分类好的版图热点数据可以根据工艺不同的需求进行调整,具有很高的灵活性。基于字符串的拓扑分类算法^[2],通过二进制 0、1 的形式将版图图形表示出来,使其与大小、位置无关,再转为十进制形式,最后利用字符串匹配原则将相似的图形进行分类。

3.3 特征提取

特征提取是版图热点检测中非常关键的阶段,特征提取的目的在于将版图热点的几何信息描述为方便计算的向量维度形式。基于密度的特征提取算法^[1]是常用的特征提取方法中提取准确率相对较高的一种,通过在版图热点位置上建立长度为 W 的提取窗口和长度为 S 的滑动窗口将版图图形均等分成小的方块区域,计算每个方块区域中版图图形的占有面积,最后按照“Z”字型扫描形成一个 K 维的有序向量序列。

$$K = (W/S) * (W/S)$$

3.4 特征降维

通常情况下,每个训练样本在特征提取之后被表示为几十维或者上百维的特征向量,面对高维度且冗长的特征向量,往往容易陷入维度灾难中,特征降维就显得尤为重要。我们希望特征降维在保证原有版图图形信息损失最少的情况下,一方面可以提取图形的关键信息降低维度,另一方面能够加快 AdaBoost 的学习速度,减少空间开销。

基于 PCA 的特征降维算法^[3],即主成分分析法 (Principle Component Analysis, PCA),是目前应用最为广泛的降维算法,通过正交变换,将向量 X 映射到低维子空间从而形成一个新的向量 P ($P < X$),即

$$P = X(m * n) \times U_k(n * k)$$

式中, X 是 m 行(样本个数) n 列(向量维度)的版图热点特征矩阵; U_k 是选取协方差矩阵 $X^T X$ 的前 K 个特征向量组成的向量矩阵。在计算方法上,本文选用了基于双边 Jacobi 的奇异值分解法 (Singular Value Decomposition, SVD) 来求解协方差矩阵的

特征值和特征向量.

3.5 运行时间

数据采用实验室 40 nm 标准单元库 785 个金属层版图热点,利用不同面积的提取窗口和滑动窗口形成多维度的特征向量,对版图定位、拓扑分类、特征提取和特征降维四个算法进行用时统计,各算法运行时间如表 1 所示.我们从表中可以了解到,版图定位和拓扑分类的运行速度很快,只与版图样本的数量有关,平均运行时间分别为 0.005s 和 0.011s;基于密度特征提取的运行时间随着提取窗口的面积每增加 100 nm² 而增加约 600 s,呈线性增长的趋势;而基于 PCA 特征降维的运行时间随着向量维度的升高呈指数增长的趋势,这对于训练几千或几万且维度较高的版图样本数据是不能忍受的,所以有必要对 PCA 降维算法进行改进.

表 1 各算法处理数据的运行时间

版图定位 /s	拓扑分类 /s	提取窗口 /nm	滑动窗口 /nm	向量维度 /d	特征提取 /s	特征降维 /s
3.876	8.792	300	50	36	705.543	52.241
		320	25	144	711.75315	306.494
		400	50	64	1 260.048	904.783
		400	100	16	1 261.215	0.996
		420	35	144	1 396.18815	422.302
		500	50	100	1 988.691	5 226.564

4 算法改进

4.1 并行计算模型

并行技术在科学研究、大型工程计算等领域有着广泛的应用,并行程序设计是使用程序设计语言显式地说明计算中不同部分如何在不同处理器上同时执行,是并行技术的核心问题^[5].近年来,基于 MPI 和 OpenMP 是主流的并行程序设计方法.

MPI(Message Passing Interface)消息传递接口,是进程级并行计算模型,通过通信在进程之间进行消息传递.

OpenMP(Open Multiprocessing)共享存储标准,线程级并行计算模型,通过专用的 pragma 来指定程序并行化.

4.2 基于 MPI 的 PCA 降维算法

我们通过对 PCA 降维串行算法进行分析,发现大部分时间耗费在利用 Jacobi 方法求解协方差矩阵的特征值和特征向量上,这是由于双边 Jacobi 算法在每一次对矩阵进行旋转变换时,遍历矩阵寻找

绝对值最大的非对角线元素需要花费 $O(n^2)$ 的时间,随着样本维度升高,构建方程组规模变大,带来计算速度较慢的问题.

本文引入正交三角分解法(QR)来优化基于双边 Jacobi 的奇异值分解法^[6].如果直接对矩阵 $\mathbf{X}_{m \times n}$ 进行奇异值分解, Jacobi 矩阵旋转变换是在 m 维空间中进行.利用 QR 分解可以将矩阵 $\mathbf{X}_{m \times n}$ 分解成一个正交矩阵 $\mathbf{Q}_{m \times n}$ 和一个上三角矩阵 $\mathbf{R}_{n \times n}$,我们对上三角矩阵 $\mathbf{R}_{n \times n}$ 进行 SVD 分解,再进行 Jacobi 矩阵旋转变换就是在 n 维空间中进行,从而减少运算量.

为了进一步提高运行速度,结合 MPI 并行计算模型,提出基于 MPI 使用 QR 优化的 PCA 降维算法.使用 MPI 并程序程序设计的主要思路为:首先,设置 MPI 运行环境,包括初始化 MPI、获取进程数及进程 ID 等;然后进行数据划分,假设有 P 个进程,则将原始矩阵分割成 n/p 个部分,每个进程都会负责矩阵相邻的 n/p 个元素;接下来就是进行数据间的通信,我们让 0 号进程(根进程)负责读入数据和向其他进程发送数据,每个进程执行完相应计算后,再由 0 号进程将计算结果进行收集和连接;最后结束 MPI 运行环境,否则程序的运行结果是不可预知的.具体算法描述如下:

- (1)初始化 MPI 运行环境;
- (2)获取进程数量 P ,矩阵的行数 m ,矩阵的列数 $n, m \gg n$;
- (3)将数据划分为 N 部分, $N=n/p$;
- (4)0 号进程读取样本数据并发送给不同的进程,每个进程:
- ①数据中心化处理

$$x=(X-X_{\text{mean}})/\sqrt{n-1};$$

②计算协方差矩阵 $\text{Cov}=x x^T$;

③进行 QR 分解, $\text{Cov}=\mathbf{QRR}^T\mathbf{Q}^T$;

(5)0 号进程收集每个进程的计算结果;

(6)对 R^T 进行奇异值分解 $R^T=\mathbf{USV}^T$ 从而求解特征值和特征向量;

(7)投影降维;

(8)0 号进程输出降维结果;

(9)结束 MPI 运行环境;

4.3 基于 OpenMP 的 AdaBoost 训练算法

AdaBoost 算法的本质是一种迭代算法, AdaBoost 每训练一个弱分类器的时间与样本数目、每个样本的特征值维度和训练次数都有关,使得 AdaBoost 算法在上万个样本的情况下训练速度不可能很快.

为了缩短训练时间,改善基于单核 CPU 的训练环境,本文选取了基于 OpenMP 的多核计算模型实现 AdaBoost 的并行计算. 使用基于 OpenMP 的并行计算有两点好处,第一,OpenMP 计算模型对循环较多的算法具有良好的加速效果. AdaBoost 算法可并行的步骤正好在于进行训练的 T 轮迭代之中,即“每一个样本训练成一个弱分类器”和“更新每一个样本的权重”,可以将这两个循环步骤分配到不同线程上进行并行计算. 第二,OpenMP 计算模型不需要数据间的相互通信,只需在原串行算法的基础上作出相应的并行命令即可实现 AdaBoost 的并行化,具体实现步骤如下:

- (1)初始化样本权重
- (2)进行 T 轮训练

① #pragma omp for 每一个样本
训练一个弱分类器 h_t ;

②估计 h_t 的误差;

③计算分类器 h_t 的权重;

④ #pragma omp for 每一个样本
更新样本权重
- (3)输出强分类器

5 实验结果

本文采用 C++ 语言在 Red Hat Linux 平台上实现对 PCA 降维算法和 AdaBoost 训练算法的并行设计.

(1)实验环境

硬件配置: Intel® Xeon® CPU E5620, 2.4 GHz; 进程数 Processor: 0 ~ 15, 共 16 个进程; 线程数 Threads = CPU 总核数 = 物理 CPU 个数 * 每颗物理 CPU 的核数 = 2 * 4, 共 8 个线程.

软件配置: Linux Version 2.6.18; Red Hat Version 4.1.2; MPI Release 3.2; GCC 4.1.2 (OpenMP).

(2)实验数据

实验数据来源于实验室 40 nm 标准单元库 785 个金属层版图热点. 为了更好的体现并行算法的加速性能和得到平均的加速比,我们将数据分为 16 维、36 维、64 维、100 维和 144 维五组进行测试. 对于 PCA 降维过程 K 值的选取,本文选择 $K=1$,降维数据保留原数据 99% 的贡献率.

(3)测试结果(单位: s)

首先,使用 QR 优化后 SVD 求解 PCA 降维算法与基于 Jacobi 的 SVD 求解 PCA 算法进行对比,

运行结果如表 2 所示.

表 2 改进的 PCA 降维算法运行时间对比

方法	16 维	36 维	64 维	100 维	144 维
Jacobi	0.996	52.24	904.7	5 226.6	15 414
QR 优化	0.296	1.185	3.502	8.399	17.439

可以看到,改进的 PCA 降维算法在高维度的计算上,运行速度有了大幅度的提升. 然后,将改进的 PCA 降维算法与 MPI 并行计算模型相结合,运行结果如表 3 所示.

表 3 基于 MPI 的 PCA 并行算法运行时间

进程数	16 维	36 维	64 维	100 维	144 维
$P=1$	0.296	1.185	3.502	8.399	17.439
$P=2$	0.15	0.613	1.837	4.485	9.454
$P=4$	0.092	0.337	1.025	2.574	5.649
$P=6$	0.09	0.253	0.769	1.98	4.537
$P=8$	0.07	0.229	0.689	1.742	4.099
$P=10$	0.088	0.292	0.84	2.179	5.081
$P=12$	0.098	0.305	0.941	2.661	6.489
$P=14$	0.103	0.308	0.939	2.676	6.553
$P=16$	0.134	0.313	0.943	2.634	6.683
$P=1/P=8$	4.229	5.175	5.083	4.821	4.254
Ave	4.712 4				

表 3 可以看到,改进的 PCA 降维算法结合 MPI 并行计算之后,速度得到了进一步的提升. 当进程数为 $P=2$ 时,运行时间比串行时间缩短几乎一倍;当进程数为 $P=8$ 时,并行加速效果达到最优,加速比平均为 4.7 倍;但当进程数超过 8 以后,由于多进程间的通信也要作为并行算法的开销,运算速度被降低. 所以,在进程数的选取上,并不是越多加速效果越好.

最后,将 PCA 降维之后的数据用于基于 OpenMP 的 AdaBoost 训练,运行结果如表 4 所示.

表 4 基于 OpenMP 的 AdaBoost 并行算法运行时间

线程数	15 维	27 维	44 维	64 维	79 维
$T=1$	1.689	2.639	3.974	5.566	6.786
$T=2$	0.957	1.431	2.174	3.009	3.603
$T=3$	0.714	1.128	1.475	2.085	2.513
$T=4$	0.573	0.808	1.199	1.721	2.108
$T=5$	0.494	0.7	1.023	1.4	1.744
$T=6$	0.476	0.67	0.941	1.27	1.54
$T=7$	0.379	0.612	0.864	1.181	1.416
$T=8$	0.347	0.58	0.785	1.085	1.334
$T=1/T=8$	4.867	4.55	5.062	5.13	5.087
Ave	4.939 2				

从表 4 可以看到,当线程数为 $T=2$ 的时,运行时间同样比串行时间缩短几乎一倍;随着线程数的不断增多,运行时间也在不断的减小. 当线程数为 $T=8$ 时,并行加速效果达到最优,加速比平均为 4.9 倍.

6 结束语

本文通过分析版图热点数据处理过程中每个算法的用时,针对耗时较多的 PCA 降维算法和基于单核 CPU 训练的 AdaBoost 算法,将目前流行的并行技术应用在基于机器学习的版图热点检测之中,分别提出了基于 MPI 使用 QR 优化的 PCA 并行降维算法和基于 OpenMP 多核的 AdaBoost 并行训练算法. 实现结果验证了并行化的可行性,为解决训练过程耗时问题提供了一个较好的解决方法.

参考文献:

[1] Wu J, Pikus Y. Detecting context sensitive hotspots in standard cell libraries[J]. Proc. of SPIE, 2009, (7275):265-271.

[2] Yu Y T. Machine-learning-based hotspot detection using topological classification and critical feature extraction[J]. IEEE/ACM Design Automation Conference (DAC), 2013 , 34 (3) :1-6.

[3] Matsunawa T, Gao J R, Yu B, et al. A new lithography hotspot detection framework based on AdaBoost classifier and simplified feature extraction[J]. Spie Advanced Lithography, 2015 :94270S-94270S-11.

[4] Yoav F Schapire, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of Computer and System Sciences, 1997,55(1): 119-139.

[5] Quinn M J. MPI 与 OpenMP 并行程序设计:C 语言版[M]. 陈文光,武永卫,译. 北京:清华大学出版社,2004.

[6] Drmac Z. A posteriori computation of the singular vectors in a preconditioned Jacobi SVD algorithm[J]. Ima Journal of Numerical Analysis, 1999, 19(2):191-213.

作者简介:

李 仲 男,(1992-),硕士研究生.研究方向为计算机技术. E-mail:lizhong@ime.ac.cn.

吴玉平 男,(1968-),硕士,研究员,硕士生导师.研究方向为混合信号集成电路设计技术与方法.

陈 岚 女,(1968-),博士,研究员,博士生导师.研究方向为超深亚微米芯片设计方法学及计算机体系结构.

张学连 女,(1987-),硕士,工程师.研究方向为混合信号电路设计方法学及 EDA 技术.

(上接第 26 页)

[2] Xiang D, Liu G, Chakrabarty K, et al. Thermal-aware test scheduling for NOC-based 3D integrated circuits[C]// International Conference on Very Large Scale Integration. [s. l.],IEEE, 2013:96-101.

[3] 欧阳一鸣,贺超,梁华国,等. NoC 架构下异构 IP 核的并行测试方法[J]. 电子学报, 2013, 41 (12): 2391-2396.

[4] 许川佩,刘洋,莫玮. 带分复用的三维片上网络测试规划研[J]. 仪器仪表学报, 2015,35(9):2120-2128.

[5] Zhao R, Tang W. Monkey algorithm for global numerical optimization[J]. Journal of Uncertain Systems, 2008, 2(3):165-176.

[6] Jia J, Feng S, Liu W. A triaxial accelerometer monkey algorithm for optimal sensor placement in structural health monitoring[J]. Measurement Science & Technology, 2015, 26(6):65104-65115.

[7] 许川佩,凌景,胡聪. 动态带分复用的三维片上网络协同优化研究[J]. 仪器仪表学报, 2016, 37(12): 2821-2828.

[8] 许川佩,陈家栋,万春霆. 基于云模型进化算法的硅通孔数量受约束的 3D NoC 测试规划研究[J]. 电子与信息学报, 2015, 37(2):477-483.

[9] 许川佩,李克梅. 基于粒子群算法的多约束 3D NoC 协同测试规划[J]. 仪器仪表学报, 2017, 38(3): 765-772.

作者简介:

许川佩 女,(1968-),博士,教授.研究方向为集成电路测试.

陈 玄(通讯作者) 男,(1993-),硕士研究生.研究方向为集成电路测试. E-mail:chy811255@163.com.