

基于云计算的数据流系综分类算法研究

钱 琳, 秦亮曦

(广西大学 计算机与电子信息学院, 广西 南宁 530004)

摘 要: 综合分析了数据流分类算法以及云计算的基本理论, 提出了基于 Hadoop 框架的数据流系综分类算法, 算法采用 MapReduce 并行编程模型对传统基于动态权重系综模型进行改进, 以提升算法的分类效率. 分析结果表明, 该算法在处理快速海量到达的数据流时, 其执行效率远高于传统系综算法.

关键词: 云计算; 数据流; 系综分类; MapReduce

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2012)02-0099-04

Study on an Ensemble Classification Algorithm for Data Streams with Cloud Computing

QIAN Lin, QIN Liang-xi

(School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China)

Abstract: According to comprehensive analysis on data streams classification algorithms and the basic theory of cloud computing, it is proposed an ensemble classification algorithm for data streams running on Hadoop framework, and it takes MapReduce parallel programming model to improve traditional dynamic weight-based ensemble, finally speed up classification efficiency. Results show that the algorithm for high speed massive data stream has much better running efficiency than traditional ensemble algorithm.

Key words: cloud computing; data streams; ensemble classification; MapReduce

1 引言

随着互联网经济的快速发展, 社会生活的各个领域产生了大规模海量的数据. 而这些海量数据往往表现出快速到达的、变化的、潜在无限^[1]等数据流特征. 根据数据流的特点, 这种特殊数据形式的挖掘算法面临着一遍扫描数据, 有限内存, 概念漂移等一系列富有挑战性的问题. 作为数据挖掘领域的基本理论之一, 数据的分类一直是学术界重点研究的对象, 设计出了诸如决策树、贝叶斯、可支持向量机等等一系列经典的算法. 但这些传统算法并不能有效地应对数据流的分类问题, 于是快速决策树、基于动态权重的系综分类等算法应运而生, 而这些算法也并不能从根本上解决执行效率和分类精度之间的矛盾.

1995 年 Google 公司提出了基于 Hadoop 云计算平台的软件架构——MapReduce^[2], 用于大规模数据集的并行化处理. 由于云计算平台分类挖掘的高效廉价和系综分类的高精度低效率, 本文创新的提出了云计算平台上的基于动态权重分类器系综算法. 利用 MapReduce 编程模型并行化传统系综算法, 将其应用于快速到达的概念漂移数据流分类中, 以达到提升分类效率, 进而提高分类精度的目的.

2 Hadoop 云计算框架

Hadoop^[3] 云计算框架是由 Apache 基金会设计的开源分布式系统基础架构. 用户可以在不了解分布式底层细节的情况下开发分布式程序, 这样就可以充分利用集群的能力进行高速运算和存储. 简单

收稿日期: 2011-06-02; **修回日期:** 2011-08-03

基金项目: “十一五”国家科技支撑计划课题(2009BAH53B03)

地说来, Hadoop 是一个可以更容易开发、处理大规模数据的软件平台。

从技术上看, Hadoop 由两项关键服务构成: 采用 Hadoop 分布式文件系统 (HDFS) 的可靠数据存储服务, 以及利用一种叫做 MapReduce 技术的高性能并行数据处理服务. 这两项服务的共同目标是, 提供一个使对结构化和复杂数据的快速、可靠分析变为现实的方法。

(1) HDFS: HDFS 是 Google File System 的开源实现, 具有高容错性的特点. 将它部署在低廉的硬件上, 可以提供高传输率来快速访问应用程序的数据, 使其适合那些有着超大数据集的应用程序. 如图 1 所示, HDFS 把节点分成两类: NameNode 和 DataNode. NameNode 是唯一的, 程序与之通信, 然后从 DataNode 上存取文件. 这些操作是透明的, 与普通的文件系统 API 没有区别。

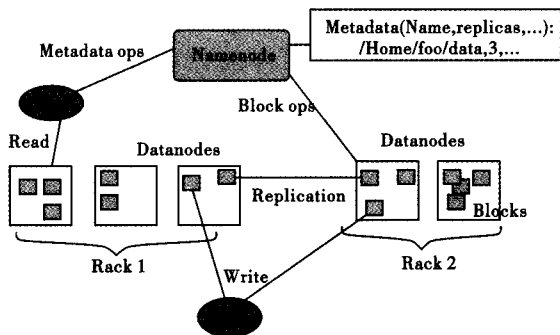


图 1 HDFS 架构示意图

(2) MapReduce^[4] 编程模型: 云计算平台的核心技术就是并行化计算, 为了使用户能够更轻松地获取云计算带来的服务, Hadoop 云计算平台采用了开源的 Google MapReduce 计算模型来实现, 将复杂的大规模集群并行计算过程高度抽象到 Map 和 Reduce 两个基本函数。

MapReduce 的处理流程: 首先将输入的海量数据集分割成 M 个数据块, 由 master 分配器来将 M 个 Map 任务分配给各空闲结点. 被分配了 Map 任务的结点接受一组数据, 从中解析出一个键/值 (key/value) 对, Map 函数接受这个键/值 (key/value) 对, 同时产生一个中间键/值 (key/value) 对列表. 将列表中具有相同 key 值的 value 值集合在一起传递给 Reduce 函数. 由 master 分配器将 Reduce 任务分配给空闲结点, Reduce 函数将这些 value 值合并成一个较小的 value 值列表, 并将结果输出. 图 2 展示了 MapReduce 的整个运作流程。

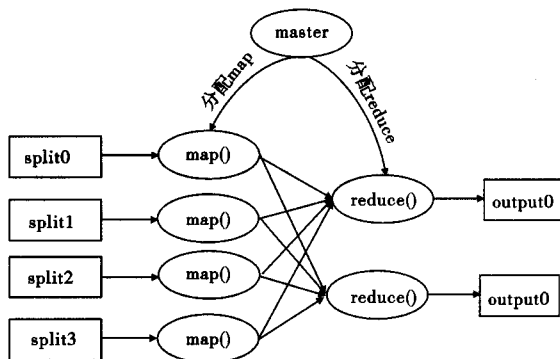


图 2 MapReduce 运作流程

3 云计算数据流系综分类算法

在传统静态海量数据挖掘算法中, 分类器系综是一种从根本上提升分类精度的方法. 它采用相继训练一系列分类器, 并根据其在随时间变化的环境中期望分类准确率加权, 保留权值最高的 k 个分类器, 并加权投票做出决策. 而数据流往往表现出快速到达、概念漂移等特征, 使得针对静态数据设计的系综算法分类效率低下, 进而影响了系综分类的整体精度. 为此, Wang 等人在文献[5]中提出了基于动态权重的系综分类算法, 有效地降低了概念漂移对于数据流分类精度的影响. 而频繁的权重调整, 基分类器的集成与淘汰产生了巨大的计算开销, 使得系综分类不能及时处理新到的数据流. 为此, 我们引入 Hadoop 云计算平台中的 MapReduce 编程模型并行优化基于权重的数据流系综算法, 从根本上加强算法的实际应用价值。

3.1 基于动态权重的数据流系综分类算法描述

令 t 为数据块到达的时间戳, D_t 为在时间 t 到达的数据块, 则数据流可以描述为 $\{\dots D_{t-1}, D_t, D_{t+1}, \dots\}$, 从数据流中截取前 n 个数据块作为基础训练集表述为 $\{D_1, D_2, \dots, D_t, \dots, D_n\}$, 并且令下一时刻即将到来的数据块为 D_{n+1} 。

基于动态权重的数据流系综分类过程如下:

(1) 在截取的前 n 个数据块上分别训练 n 个分类器作为基础分类器集合 $C_t (t=1 \dots n)$, 基础分类器的选择标准为根据各种数据流分类器的特点, 尽量选择差异性强的分类器进行集成如 (VFDT, CVFDT, SVM, 朴素贝叶斯等)。

(2) 给每个基础分类器 C_t 赋予一个权值 w_t , 此权值的大小的调整反比于每个基础分类器对新到数据块 D_{n+1} 分类的均方误差率 $MSE_t = \frac{1}{|S_n|} \sum_{(x, c) \in S_n} (1 - f_c(x))^2$ 为了方便表述令 $w_t = MSE_r - MSE_t$, 其

中 $MSE_r = \sum_c p(c)(1-p(c))^2$ 为丢弃阈值,可动态根据原分类器的随机预测概率 $p(c)$ 作出调整。

(3) 若出错率超过丢弃阈值则将这样的分类器丢弃,并在新的数据块上生成新的分类器以权值 w_i 加入基础分类器集合,否则继续调整现有基础分类器集合权值。

(4) 调用加权投票算法获得最终决策分类器。

3.2 MapReduce 云计算数据流系综分类算法描述

算法描述如下:

(1) 将数据流中被采样为训练数据的部分进行划分,分成 n 个规模相当的数据块。

(2) 首先并行在 n 个训练数据块上产生 n 个基础分类器,然后把这 n 个数据块进行格式化操作,解析出 $\langle \text{key}_1, \text{value}_1 \rangle$ 对。在我们这个具体的情况下这个对的格式为 $\langle \text{Block}, \text{Classifier} \rangle$, 其中 Block 为块号,Classifier 为基础分类器。伪代码段如下:

Input: Data streams

Output: $\langle \text{Block}, \text{Classifier} \rangle$

Sample Data streams as training data and partition it into n blocks with MapReduce

Parallel do produce n classifiers on $\{D_1, D_2, \dots, D_i, \dots, D_n\}$ // 并行产生基础分类器集合

Emit $\langle \text{Block}, \text{Classifier} \rangle$ // 产生 $\langle \text{key}_1, \text{value}_1 \rangle$ 键值对

(3) 根据集群计算机的个数 n , 由 Master 分配进程将先到的 n 个数据块分别发送至分配了 Map 任务的 n 个结点。

(4) 每个分配了 Map 任务的结点接收到 $\langle \text{key}_1, \text{value}_1 \rangle$ 对后, 由 Map 函数控制在相应的训练数据块上产生单分类器, 将新到的一个数据块放入共享资源池, 使得所有分配了 Map 任务的结点并行计算分类器在新到数据块上的 MSE_r 、 MSE_i 以及权值 $w_i = MSE_r - MSE_i$, 并产生中间 $\langle \text{key}_2, \text{value}_2 \rangle$ 对, 其具体值为 $\langle \text{Classifier}, w \rangle$ 。若产生的 $\langle \text{Classifier}, w \rangle$ 对中的 $w=0$, 则在新到的数据块上产生替代基分类器并以随机权值 w 加入系综决策, 同时丢弃失效基础分类器。以下为 Map 函数的伪代码段:

Map(Block, Classifier)

// Block 为块号, Classifier 为基分类器

{

Receive $\langle \text{Block}, \text{Classifier} \rangle$ // 接收 $\langle \text{key}_1, \text{value}_1 \rangle$ 键值对

For $t=1 \sim n$ parallel do // 并行计算权值

{

Read D_{t+1} ; // 并行读取共享资源池中的新到数据块

Compute $w_i = MSE_r - MSE_i$;

}

If ($w=0$)

{

Discard C^i ;

produce new classifier C_{t+1} on D_{t+1} ;

Ensemble $C^S \cup C_{t+1}$;

}

Emit $\langle \text{Classifier}, w \rangle$;

}

(5) 加强基础分类器的差异性, 可以最大程度地发挥系综分类器在处理概念漂移数据流时高精度的特性。可以调用 MapReduce 中一个可选用的 Combiner 函数, 将特性相同的分类器进行合并。函数采用同一类型的众多分类器中分类精度最高的输出, 这样不但节约了宝贵的网络带宽、降低了网络延迟, 还能有效提高分类精度。例如, 若并行产生的基分类器中, 产生了快速决策树、概念自适应决策树、增量决策树等决策树类算法可将其用 Combiner 函数合并输出一个精度最高的分类器。伪代码如下:

Combiner(Classifier r_i, w_i)

{

Parallel do if ($\text{Classifier}_i = \text{Classifier}_{i+1}$) select highest w ; // 若分类器种类相同则选权重最高的加入基础分类器集合

Emit intermediate $\langle \text{Classifier}_K, w_K \rangle$; // 输出优化过的 $\langle \text{Classifier}, w \rangle$ 键值对

}

(6) 由分配了 Reduce 任务的结点接收经由 Combiner 函数处理后的 $\langle \text{Classifier}, w \rangle$ 对, 并调用加权投票算法, 最终得到用于下一个新到数据块分类的分类器。

Reduce(Classifier K, w_K)

{

For each w_K in w list

Parallel do Voting;

Emit Classifier;

}

注: 对于高速数据流在时间戳 t 时刻可能流过多个新到的数据块, 我们利用分区函数 $\text{Hash}(\text{key}) \bmod R$ 将 Combiner 函数产生的中间键值对分成 R 个不同的分区, 再把每个分区映射到 R 个分配了 Reduce 任务的节点上, 使其并行执行加权投票算法, 加快系综分类器的性能。

4 算法时间复杂度分析

假设使用大小为 block 的数据块 S 上构建一个基础分类器的时间复杂度为 $O(f(\text{block}))$. 令计算单分类器权值的时间复杂度为 $O(CW(i))$, 则系统分类器整体权值计算复杂度为 $O(\sum_{i=1}^n CW(i))$, 加权投票的时间复杂度为 $O(1)$ 与以上相比可忽略不计. 传统的分类器系统算法串行的在 n 个数据块上构建 n 个基础分类器并计算分类器权值, 故整体时间复杂度为 $O(n * f(\text{block})) + O(\sum_{i=1}^n CW(i)) = O(f(s) + \sum_{i=1}^n CW(i))$. 而基于 MapReduce 的云计算系统分类算法提供了并行构建基分类器的机制, 故其整体时间复杂度仅为 $O(f(\text{block}) + \frac{1}{n} \sum_{i=1}^n CW(i) + T(n))$, 其中 $T(n)$ 为云计算系统的通讯开销. 当分布式结点的通讯开销 $T(n) \ll f(\text{block}) + \frac{1}{n} \sum_{i=1}^n CW(i)$ 时, 与传统系统分类算法相比, 云计算平台上的系统分类算法时间复杂度大大降低, 提高了算法的执行效率. 反之, 说明当云计算系统中的结点个数为 n 时, 通讯开销对算法效率的影响很大, 算法在这个特定云环境下不具有可扩展性. 但仅升级云计算平台的软硬件配置, 并不能解决云计算自身的弊端, 这也是未来需进一步完善的方面.

5 结束语

本文提出了一种在云计算环境下以 MapRe-

duce 为基本编程模型的数据流系统分类算法. 此算法将传统基于动态权重的数据流系统分类算法进行 MapReduce 并行化改造, 并应用到 Hadoop 云计算平台来提升算法处理大规模快速数据流分类问题的能力. 通过对算法复杂度进行理论分析可知, 算法的执行效率大大超过了串行算法. 但是我们同样注意到由于云计算环境受网络带宽、通信开销等诸多因素影响, 如何优化现有 MapReduce 编程模式来降低云平台固有的缺陷仍然有待进一步的研究.

参考文献:

- [1] 程转流, 胡为成. 基于 DSFCI-tree 的分布式数据流频繁闭合模式挖掘 [J]. 微电子学与计算机, 2007, 24(9): 120-125.
- [2] 刘鹏. 云计算 [M]. 北京: 电子工业出版社, 2010: 1-4.
- [3] White T. Hadoop, the definitive guide [M]. O'Reilly Media, Inc, 2009.
- [4] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. [C] // Proc of the 6th Symposium on Operating Systems Design and Implementation. San Francisco: Google Inc, 2004.
- [5] Wang H, Fan W, Yu P S, et al. Mining concept-drifting data streams using ensemble classifiers. [C] Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington D C, USA: ACM, 2003: 226-235.

作者简介:

钱琳 男, (1984—), 硕士. 研究方向为数据挖掘与知识发现.

(上接第 98 页)

- [2] Wu Cheng-Shing, Wu An-Yeu. Modified vector rotational CORDIC (MVR-CORDIC) algorithm and architecture [J]. IEEE Transactions on Circuits and Systems- II: Analog and Digital Signal Processing, 2001, 48(6): 548-561.
- [3] 张尊, 李绍荣. 基于改进的 CORDIC 算法的 NCO 实现 [J]. 信息通信, 2009(4): 11-13.
- [4] 杨小牛, 楼才义, 徐建良. 软件无线电技术与应用 [M]. 北京: 北京理工大学出版社, 2010.
- [5] 何宾. FPGA 数字信号处理实现原理及方法 [M]. 北京: 清华大学出版社, 2010.

- [6] 施国勇. 数字信号处理 FPGA 电路设计 [M]. 北京: 高等教育出版社, 2010.

作者简介:

段文伟 女, (1987—), 硕士研究生. 研究方向为扩频通信、卫星导航.

于龙洋 男, (1989—), 硕士研究生. 研究方向为扩频通信、卫星导航.

李署坚 男, (1952—), 高级工程师, 硕士生导师. 研究方向为扩频通信.