云计算平台上的增量分类研究*

李 曼
(南京邮电大学 计算机学院,江苏 南京 210003)

摘 要:针对已有增量分类算法只是作用于小规模数据集或者在集中式环境下进行的缺点,提出一种基于 Hadoop 云计算平台的增量分类模型,以解决大规模数据集的增量分类。为了使云计算平台可以自动地对增量的训练样本进行处理,基于模块化集成学习思想,设计相应 Map 函数对不同时刻的增量样本块进行训练,Reduce 函数对不同时刻训练得到的分类器进行集成,以实现云计算平台上的增量学习。仿真实验证明了该方法的正确性和可行性。

关键词:增量分类;Hadoop;云计算
中图分类号:TP391 **文献标识码**:A **文章编号**:1674-7720(2011)18-0065-04

Incremental classification method based on cloud computing

Li Man
(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: To alleviate some issues about the current incremental learning algorhtms, such as only for small-scale data sets and work in a centralized environment, an incremental classification algorithm based on Hadoop cloud computing platform is proposed to deal with large data sets. In order to automatically process the incremental training samples on cloud computing platform, we design Map function to train incremental data blocks at different times, and design reduce function to integrate different classifiers based on modular ensemble learning. The simulation results indicate that the proposed method is correct and feasible.

Key words: incremental classification; Hadoop; cloud computing

随着信息技术和生物技术突飞猛进的发展,科学研究和实际应用中产生了海量数据,并且这些数据每天都在增加,为了将每天产生的新数据纳入到新的学习系统,需要利用增量学习。增量学习比较接近人类自身的学习方式,可以渐进地进行知识的更新,修正和加强以前的知识,使得更新后的知识能适应更新后的数据,而不必重新学习全部数据,从而降低了对时间和空间的需求。模块化是扩展现有增量学习能力的有效方法之一[1],而集成学习(Ensemble Learning)一直是机器学习领域的一个研究热点[2-6],许多模块化增量分类算法[7-9]正是基于二者提出的。

云计算(Cloud Computing)这一新名词从 2007 年第 3 季度诞生起就在学术界和产业界引起了轰动,Google、IBM、百度、Yahoo 等公司都开始进行“云计算”的部署工作。云计算是分布式计算(Distributed Computing)、并行计算(Parallel Computing)和网格计算(Grid Computing)的发展与延伸。在云计算环境下,互联网用户只需要一个终端就可以享用非本地或远程服务集群提供的各种服务(包括计算、存储等),真正实现了按需计算,有效地提高了云端各种软硬件资源的利用效率。随着云计算技术的日益成熟,云计算也为解决海量数据挖掘所面临的问题提供了很好的基础[10]。虽然在机器学习领域,对增量学习进行了较深入的研究,但是在云计算环境下,还没有相关文献讨论利用增量分类提高云计算环境下海量数据挖掘的效率问题。本文基于模块化的集成学习思想,研究在开源云计算平台 Hadoop[11]上的增量分类方法。

1 Hadoop 云平台的体系结构

在现有的云计算技术中,Apache 软件基金会(Apache Software Foundation)组织下的开源项目 Hadoop 是一个很容易支持开发和并行处理大规模数据的分布式

* 基金项目:国家自然科学基金(61073114)

云计算平台，具有可扩展、低成本、高效和可靠性等优点。程序员可以使用 Hadoop 中的 Streaming 工具(Hadoop 为简化 Map/Reduce 的编写，为让不熟悉 Java 的程序员更容易在 Hadoop 上开发而提供的一个接口)使用任何语言编写并运行一个 Map/Reduce 作业。Hadoop 项目包括多个子项目，但主要是由 Hadoop 分布式文件系统 HDFS (Hadoop Distributed File System) 和映射/化简引擎(Map/Reduce Engine)两个主要的子项目构成。

1.1 分布式文件系统 HDFS

Hadoop 实现了一个分布式文件系统(Hadoop DistributedFile System)，简称 HDFS。HDFS 采用 Master/Slave 架构，一个 HDFS 集群由一个 NameNode 节点和若干 DataNode 节点组成。NameNode 节点存储着文件系统的元数据，这些元数据包括文件系统的名字空间等，并负责管理文件的存储等服务，程序使用的实际数据并存放在 DataNode 中，Client 是获取分布式文件系统 HDFS 文件的应用程序。图 1 是 HDFS 结构图。

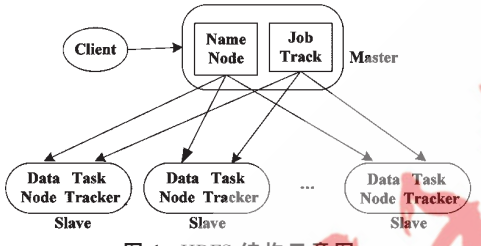


图 1 HDFS 结构示意图

图 1 中，Master 主要负责 NameNode 及 JobTracker 的工作，JobTracker 的主要职责是启动、跟踪和调度各个 Slave 任务的执行。还会有多台 Slave，每一台 Slave 通常具有 DataNode 的功能并负责 TaskTracker 的工作。TaskTracker 根据应用要求来结合本地数据执行 Map 任务以及 Reduce 任务。

1.2 Map/Reduce 分布式并行编程模型

Hadoop 框架中采用了 Google 提出的云计算核心计算模式 Map/Reduce，它是一种分布式计算模型，也是简化的分布式编程模式[12]。Map/Reduce 把运行在大规模集群上的并行计算过程抽象成两个函数：Map 和 Reduce，其中，Map 把任务分解成多个任务，Reduce 把分解后的多个任务处理结果汇总起来，得到最终结果。图 2 介绍了用 Map/Reduce 处理数据的过程。一个 Map/Reduce 操作分为两个阶段：映射和化简。

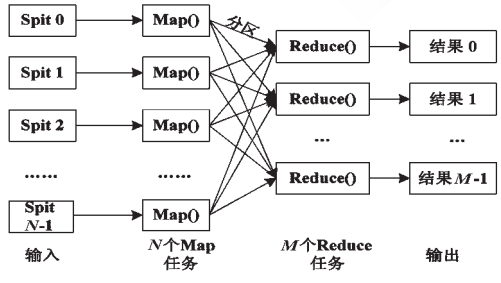


图 2 Map/Reduce 处理数据的过程

在映射阶段(Map 阶段)，Map/Reduce 框架将用户输入的数据分割为 N 个片段，对应 N 个 Map 任务。每一个 Map 的输入是数据片段中的键值对$<K1, V1>$集合，Map 操作会调用用户定义的 Map 函数，输出一个中间态的键值对$<K2, V2>$。然后，按照中间态 $K2$ 将输出的数据进行排序，形成$<K2, \text{list}(V2)>$元组，这样可以使对应于同一个键的所有值的数据都集合在一起。最后，按照 $K2$ 的范围将这些元组分割成 M 个片段，从而形成 M 个 Rdeuce 任务。

在化简阶段(Reduce 阶段)，每一个 Reduce 操作的输入是 Map 阶段的输出，即$<K2, \text{list}(V2)>$片段，Reduce 操作调用用户定义的 Reduce 函数，生成用户需要的结果$<K3, V3>$进行输出。

2 基于 Map/Reduce 的模块化增量分类模型

基于 Map/Reduce 的增量分类模型，主要思想是 Map 函数对训练数据进行训练，得到基于不同时刻增量块的分类器，Reduce 函数利用 Map 训练好的分类器对测试样本进行预测，并且将不同时刻训练得到的分类器进行集成，得到最终的分类结果。基于 Map/Reduce 的增量分类模型如图 3 所示。当 t_1 时刻有海量的训练样本到达时，通过设置 Map 任务的个数使得云平台自动地对到达的海量样本进行划分，每个 Map 的任务就是对基于划分所得的样本子集进行训练得到一个基分类器。同一时刻的不同 Map 之间可以并行训练，从而得到 t_1 时刻的增量分类系统。当 t_T 时刻的训练样本到达以后，采取相同的步骤，得到 t_T 时刻的不同基分类器，然后将这些分类器加入到 t_{T-1} 时刻的增量分类系统以构成 t_T 时刻的增量分类系统。再采用 Reduce 函数将当前增量分类系统里所有分类器进行集成，集成方法可以采用投票法 Majority Voting(MV)进行。

2.1 Map 过程

Map 函数的主要功能就是建立不同时刻的增量分类系统。当某一时刻有新的训练样本到达时，Map 便从 HDFS 将其读取。通过设置 Map 任务的个数使得云平台自动地对大规模的训练样本进行划分，每一个 Map 任务完成基于一个划分块的分类训练，划分后的不同块可以并行训练，从而得到基于该时刻增量样本集的不同分类器，然后将这些分类器加入上一时刻的增量分类系统以构成当前时刻的增量分类系统。Map 函数伪代码如下：

```
void Map( )
  {…
```

(1)初始化：训练分类算法相关参数；初始训练数据集：$TR_1=\{\{X_i, y_i\} | X_i \in R^D, y_i \in \{1, 2, \cdots, c\}, 1 \leqslant i \leqslant N_1\}$，$X_i$ 表示样本，y_i 表示样本类别，$\{1, 2, \cdots, c\}$表示类别标志集合，N_1 表示初始训练样本数，D 表示样本特征空间维数。

(2) 设置 Map 任务给 n 个数，将 TR_1 自动划分成 n 块，使得：$TR_1=\{TR_{11}, TR_{12}, \cdots, TR_{1n}\}$

(3)FOR $m=1, 2, \cdots, n$

①基于 TR_{1m} 训练得到分类器 C_{1m}

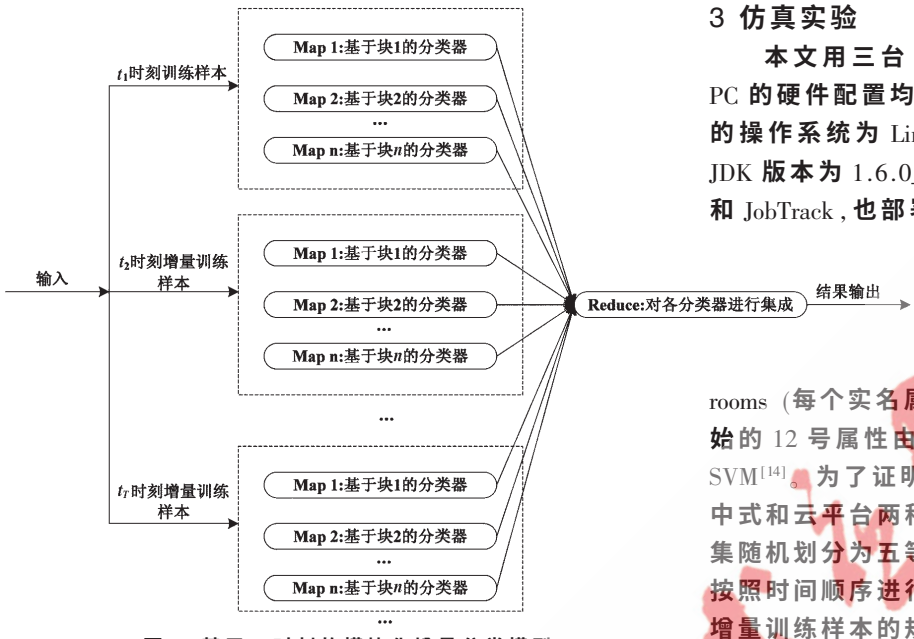


图 3 基于 t_T 时刻的模块化增量分类模型

②将分类器 C_{1m} 加入初始时刻的增量分类系统

END

(4)FOR $j=2,3,\cdots,T$

①将在 t_j 时刻采集的训练集增量块 TR_j 划分，使得：$TR_j=\{TR_{j1},TR_{j2},\cdots,TR_{jn}\}$；

②FOR $m=1,2,\cdots,n$

(a)用 TR_{jm} 训练得到基分类器 C_{jm}；

(b)将分类器 C_{jm} 加入 t_{j-1} 时刻的增量分类系统，以构成 t_j 时刻的增量学习系统；

END

END

}

2.2 Reduce 过程

Reduce 函数的主要功能是对当前时刻增量分类系统的分类器进行集成。集成策略可以采用简单的投票法 Majority Voting(MV)。也就是将 t_j 时刻增量分类系统中各个基分类器的分类结果进行多数投票，得到票数多的类别作为测试样本的类别。如果出现票数相等的情形，则进行随机猜测。Reduce 函数的伪代码如下：

```
void Reduce( )
 {…
(1)初始化：当前增量数据采集时刻数 T；测试样本 X_test
(2)FOR j=1,2,…,T
    FOR m=1,2,…,n
        求分类器 C_jm 对 X_test 的分类结果 C_jm(X_test)；
    END
END
```

(3)若所有 $C_{jm}(X_{test})(j=1,2,\cdots,T,m=1,2,\cdots,n)$ 均相同，则得到 X_{test} 的类别 $C_{jm}(X_{test})$，否则则根据多数投票的分类器组合策略输出 X_{test} 的类别。

}

3 仿真实验

本文用三台 PC 搭建了 Hadoop 云计算平台，三台 PC 的硬件配置均为 2GRAM 和 AMD 双核 CPU，各节点的操作系统为 Linux Centos 5.4，Hadoop 版本为 0.19.2，JDK 版本为 1.6.0_12。实验中一台 PC 既部署 NameNode 和 JobTrack，也部署 DataNode 和 TaskTrack，另两台 PC 均部署 DataNode 和 TaskTrack。

实验对两个数据集进行了仿真，第一个数据集是来自 UCI 的 Adult[13] 数据集，第二个是来自 UCI 的 Mushrooms（每个实名属性都被分解成若干个二进制属性，初始的 12 号属性由于丢失未使用）数据集。分类器采用 SVM[14]。为了证明该方法的正确性，每个实验分别在集中式和云平台两种环境下进行。两种环境都将训练样本集随机划分为五等份以构成 5 个增量训练子集，也就是按照时间顺序进行了 5 次增量训练。由于现实中采集的增量训练样本的规模可能很大，所以在云平台环境中，通过设置 Map 的个数对样本进行分解。本次实验中 Map 的个数设为 2，这样每个增量训练子集都会被云平台自动划分成两块，各块之间可以进行并行训练。为了与云平台环境进行对比，在集中式环境中将每个子集手动均分成两块，每块用来训练一个分类器。每个实验均采用了式(1)和式(2)两种核函数。

RBF 核函数：$K(X_i,X)=\exp(-\gamma\|X_i-X\|^2)$ (1)

Sigmoid 核函数：$K(X_i,X)=\tanh(\beta_0 X_i^T+\beta_1)$ (2)

其中，相关参数 $\gamma=0.008$，$\beta_0=0.009$，$\beta_1=0$，惩罚因子 $C=1$。实验所用的两个数据集的数据分布如表 1 所示。

表 1 两个数据集的数据特性

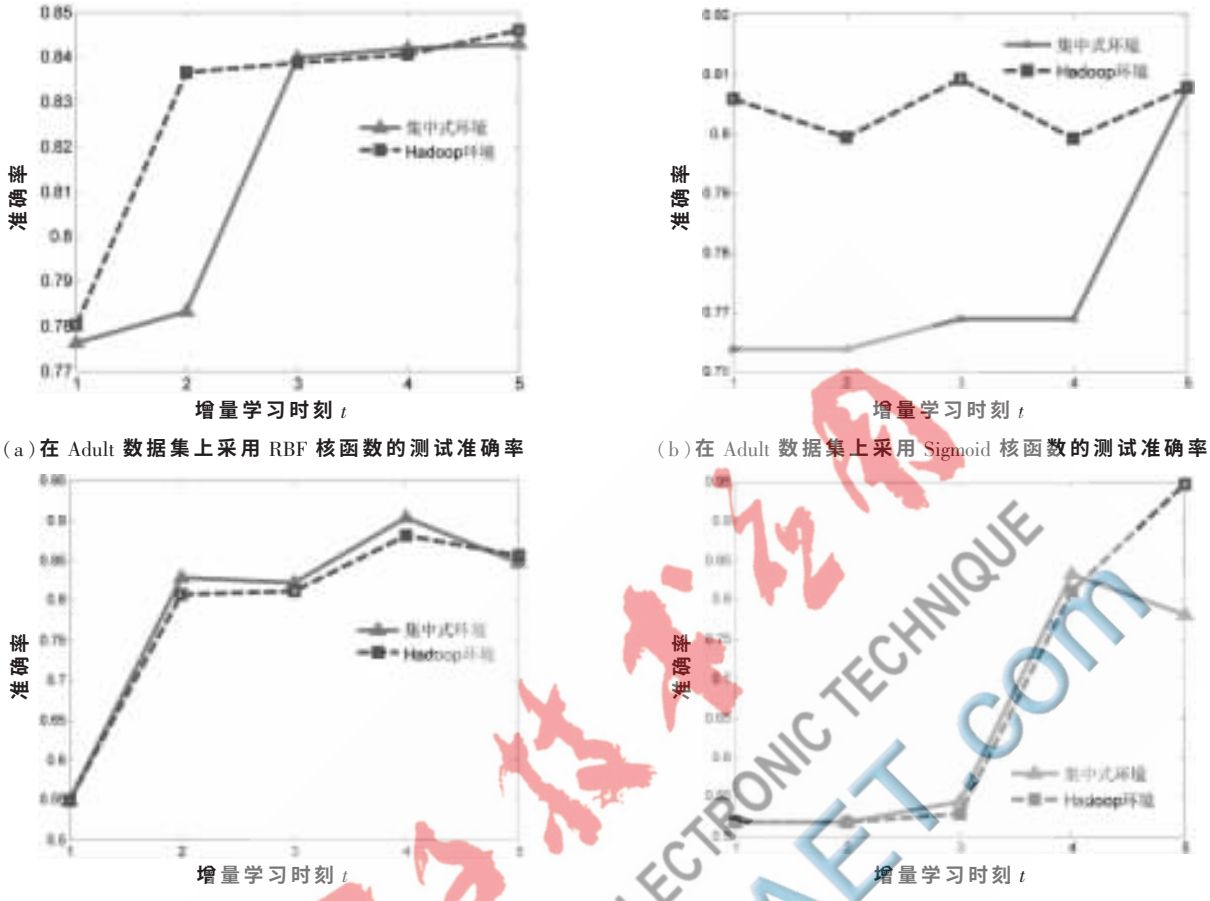
数据集	特征维数	训练样本数	测试样本数
Adult	14	32 561	16 281
Mushrooms	21	8 124	8 124

两个增量分类系统在不同数据集上的分类性能如图 4 所示。通过比较可知，集中式增量分类的准确率和 Hadoop 云平台上增量分类准确率较为接近，证明了本文所提出的在 Hadoop 云平台上实现增量分类方法的可行性和正确性。由于 MV 方法本身具有较大的波动性，故集中式和 Hadoop 云平台环境中随着训练样本的增加，增量分类系统的学习能力是曲折上升的。

本文提出了一种基于 Hadoop 云平台的增量分类方法，仿真实验表明，基于 Hadoop 云平台的增量分类是可行的。与其他增量分类方法相比，该模型简单，易于实现。通过设置平台中 Map 任务的个数让云平台自动地对海量训练样本进行划分，划分后的各个任务相互独立，可以进行并行训练。这提高了海量数据的处理速度，基本实现了实时的增量分类。

参考文献

[1] 罗四维，温津伟.神经场整体性和增殖性研究与分析[J].



(a)在 Adult 数据集上采用 RBF 核函数的测试准确率　(b)在 Adult 数据集上采用 Sigmoid 核函数的测试准确率

(c)在 Mushrooms 数据集上采用 RBF 核函数的测试准确率　(d)在 Mushrooms 数据集上采用 Sigmoid 核函数的测试准确率

图 4　两个增量系统在不同数据集上的分类性能

计算机研究与发展,2003,40(5):668-674.

[2] 周志华,陈世福.神经网络集成[J].计算机学报,2002,25(1):1-8.

[3] 王珏,石纯一.机器学习研究[J].广西师范大学学报,2003,21(2):1-15.

[4] LU B L,ITO M.Task decomposition and module combination based on class relations:a modular neural networks for pattern classification[J].IEEE Trans.Neural Networks,1999,10(5):1244-1256.

[5] HUANG Y S,SUEN C Y.A method of combining multiple experts for the recognition of unconstrained handwritten numerals[J]. IEEE Trans. Pattern Analysis and Machine Intelligence,1995,17(1):90-94.

[6] WOODS K, KEGELMEYER W P,BOWYER K.Combination of multiple classifiers using local accuracy estimates[J]. IEEE Trans. Pattern Analysis and Machine Intelligence, 1997, 19(4):405-410.

[7] POLIKAR R,UDPA L,UDPA S S,et al. Learn++:an incremental learning algorithm for supervised neural networks [J]. IEEE Trans. System, Man, and Cybernetic, 2001,31(4):497-508.

[8] LU B L,ICHIKAWA M. Emergent online learning in min-max modular neural networks[J]. In:Proc. of Inter'l Conference on Neural Network(IJCNN'01), Washington, DC, USA, 2001:2650-2655.

[9] 文益民,杨旸,吕宝粮.集成学习算法在增量学习中的应用研究[J].计算机研究与发展, 2005, 42(增刊):222-227.

[10] COPPOCK H W,FREUND J E.All-or-none versus incremental learning of errorless shock escapes by the rat[J]. Science, 1962, 135(3500):318-319.

[11] Hadoop.[EB/OL].2008-12-16.http://hadoop.apache.org/core/.

[12] DEAN J,GHEMAWAT S.MapReduce:simplified data processingon large clusters[J]. Communications of the ACM, 2008, 51(1):107-113.

[13] PLATT J C. Fast training of support vector machines using sequential minimal optimization[D].SCHOLKOPF B, BURGES C J C, SMOLA A J, editors.Advances in kernel methods-support vector learning.Cambridge, MA,MIT Press,1998.

[14] 邓乃扬,田英杰.数据挖掘中的新方法:支持向量机[D].北京:科学出版社, 2004.

(收稿日期:2011-05-19)

作者简介:

李曼,男,1986 年生,硕士研究生,主要研究方向:计算机应用与技术。