### 机器学习【大数据 > *volume*】

1. 概览

Big Data【5V】：*Volume——大量* ；Velocity——速度；

Variety——种类；Value——价值；Veracity——精确

Big data machine learning:

所谓大数据，即为这种情况：数据量高于One Computer的内存；

所以，很有必要采用：分布式存储【数据挖掘/机器学习】；

但是，对于GPU-Based machine learning不予考虑。

**分布式系统：**

1. 2.

3.

处理器

内存

处理器

处理器

处理器

内存

From small to big data——两点不同：

1. *对大数据进行分析的方法还没有完全准备好，甚至还没有一个 完备的Tools。*
2. 而大量unilmited的数据使我们更容易挖掘信息。

分布式数据分析的可能优势：

1. 平行数据加载：（Parallel data loading）
2. 从磁盘加载TB级数据很慢；但如果使用100个机器的话，那么每个机器的磁盘就会存储1/100数据

**1/100的加载时间**

2.但是如何在100个机器存储数据则是另一个问题。

1. 故障容差：（fault tolerance）

一些数据会在100个机器上重复。那么，如果one加 载失败，会有others顶替。

分布式数据分析的可能劣势：

3.工作流NOT打断：（workflow not interrupted）

如果数据已经分布式存储的话，那么试图减少一个机器 去分析数据不会很方便。

1. More：

你可以迭代地抽取样本（subsample）到自己的机器

> > 更深度的分析

1. 沟通 & 同步：（communication & synchronization）

移动计算数据（moving computation to data）并不是如 想象中的那么容易。

Doing 分布式 or not：

并不是所有的数据集都适合hadoop。现在flash drive，disk，以及RAM的容量都很大，有很大可能性一次去处理TB级数据。

**而二者之间的抉择：与具体的应用直接相关(application dependent)。**

Challenge：配置分布式开发环境

Now Consider: 1. 计算时间（in the past）

1. 加载时间
2. 沟通 & 同步时间

加载时间：（loading time）

事实上，往往影响时间大小的因素不是计算时间，而是其他。

计算时间现已不是唯一的考量因素。

常常：加载时间 >> 计算时间

Why ？ 取决于复杂度（complexity）N

假定存储层仅仅包含disk，而实例的数量为N；

那么**加载时间**：N\*（一个big常量）；

**运行时间**：N^q \*（一些常量） 其中q>=1；

【1传统情况下（q>1），运行时间会更大，由于采 用非线性的算法；2但是当N很大时，我们可能会 使用一个线性的算法（q=1）以提高效率

***此时，加载时间可能会处于支配地位***。】

Small 分析 & Big 分析：

对于big data，究竟采用small分析还是big分析为好？

如果你需要从一个huge数据集中抽取单条记录，那是相当简单；

例如：存储一条“高速公路上预定的房间”是很fast 的。

如果你想分析整个数据集，那是较为困难的。

例如：多次存储数据。

1. 分布式【数据分析算法】

引子：一台机器上的existing data mining/machine learning算法

它们在设计时并没有考虑数据的存储以及与中间结果的交流；

并且在假定合理的情况下，迭代地使用数据。

【所以，在一个分布式系统中做最小二乘回归算法并不是很容易的。】

在分布式系统中的数据分析算法：

大致有两种类型的方法：

1.使existing（单个机器）算法并行化；

2.特意设计new算法for分布式设置；

3.处于二者之间。

1. 一个例子：分布式的*牛顿方法for逻辑回归 ——LIBLINEAR库*
2. 其他实现：Vowpal\_wabbit(一种机器学习系统)

它刚开始在单个机器上作为一个线性分类器；6.0版本之后，得到Hadoop 支持。它最初是令随机梯度平行化，之后转向LBFGS（类似牛顿方法）。

Sibyl from Google(一种分布式机器学习系统)

它是基于：Minimize Ai（wi）, in parallel



3.成功的应用：计算广告是一个想法来使用分布式线性分类器。

CTR预测：一个二分类问题

其中事件的分类为：(not )clicked features

3.讨论与结论

1.分布式训练的设计思想：

在一台机器上，我们经常采用批处理而不是在线学习。

而对于大数据应用来说，在线&流学习可能更有用。在之前的牛顿方法的例子中，我 们采用的就是平行&同步算法。

1. 系统要素：针对大数据平台的开发正处于初期；仍有一些技术细节需要考虑。

例如在开发语言的选取上（spark&Scala）；mapPartitions 与map in spark 之间的不同；迭代的执行&函数式的并行。

1. 工作流因素：数据分析仅仅是构建一个大数据应用的一部分。其工作流包括从清洗数 据到使用结论；而其余的中间步骤消耗的时间可能更多。

最后，我们要综合考虑算法、系统、应用这三方面因素。