一种基于统计推断的交互式大数据探索系统

计算机科学与技术（实验班） 12070001 柯伟辰

指导老师：蒋宗礼 林庆维

摘要

随着大数据时代的到来，大数据当中的商业价值日益为各家企业所重视。通过对大数据的分析，企业可以发现数据当中的规律，从中获取经济效益。然而，分析大数据是一项十分困难的任务，企业决策者们在大数据上进行分析的时候，即使使用成熟的分布式系统进行运算，也不得不忍受漫长的查询时间。这严重影响了决策者们的工作效率。事实上，比起完全精确的结果，决策者们对于一个能够迅速得到的“大致精确”的结果更有兴趣。基于这样的现象，本文旨在设计一套能够支持交互式探索大数据的系统。该系统利用采样推断对查询进行结果估计，从而快速地返回查询结果，同时支持在大数据上进行验证查询结果。实验表明该系统的查询速度完全支持交互式操作，并保证了较高的估计准确度。

关键词：大数据，商业智能，统计推断

Abstract

With the coming of “The era of big data”, business value of the massive data is receiving increasing attention from companies. More revenue could be made from the patterns in the data with proper analytic methods. However, analyzing big data remains to be a very hard problem. Decision makers have to put up with the slow response for their queries on big data, even if they are executed on distributed systems. The slow response time is a critical restriction to them. In fact, decision makers do not require accurate answers to the queries: an approximate answer is enough for them. With respect to this phenomenon, we designed a system that gives approximate answers of queries in a short time on big data based on statistics inference, and supports verifying the result on the actual data. Evaluations on the system shows that it is quick enough for interactive queries while preserving high accuracy of the estimated answers.

Key Words: Big data, Business intelligence, Statistics inference

1. 绪论

如今，我们生活在一个“数据驱动”的时代中，近年以来，全世界都在对大数据分析投入大量精力进行研究，试图挖掘出大数据背后的商业价值。在企业中，对收集来的大数据进行分析与可视化，对企业决策者来说是重要的工作，对企业的未来发展有着重大的战略意义。

但是，我们知道对大数据的分析是极其困难的，因为大数据的体积远远超过一般机器所能处理的能力。为此，类似于 Map-Reduce 的分布式大数据存储和计算框架被陆续提出，解决了大数据的处理问题。然而，这样的框架并不能满足企业决策者们的需要，因为其响应时间依然过长。事实上，企业决策者们不需要花费如此长的时间得到一个精确的结果，对于他们来讲，一个短时间内的“大致精确”的结果就已经足够做出结论了。例如，一个决策者更需要在几秒钟内知道“2011 年的销量大约是 5 亿美元”，而不是花几个小时以后知道“2011 年的销量是 500,000,011.24 美元”。因此，也有很多有损压缩数据的方法被提出，例

如随机采样，直方图，小波变换等等。但是，这些方法却都有其不能解决的问题：有的不能保证误差，有的预处理时间过长，有的不适用于稀疏数据等等。

事实上， 从决策者的需求考虑的话，他们需要的是这样一种系统：首先能够提供交互式的界面，能够提供对原数据快速的探索式查询，系统在这个阶段提供的是对大数据上准确结果的估计值，这样决策者能够很快发现其感兴趣的结论；然后由于这种结论存在误差，该系统也应当允许决策者在大数据上实际运行这个查询，以便于验证其结论正确性。

本文的目的就是构建一个这样的系统，首先使用一种分层采样算法对原数据进行抽样，之后的查询都在这份样本上进行，并且对于查询给出推断的准确值以及置信区间；同时，系统也连接到一个分布式集群，支持对大数据的直接查询。

本文是作者在微软亚洲研究院软件分析组实习期间完成的，整个系统的版权归微软公司所有，部分数据因为涉及机密信息而进行了一定的屏蔽。

2. 采样推断模块设计

2.1 分层采样

对原数据进行采样的时候，最简单的方法就是简单随机采样，即对原数据当中每一条都以等概率决定是否舍弃。但是这样的简单随机采样对于一些数据效果会打很大的折扣，其中最重要的问题就是丢组的问题。为了解决这一问题，我们可以使用分层采样的方法。所谓分层采样，就是在每组当中按比例进行随机抽样，这样就可以保证不丢失任何一个组别。分层采样方法的原理是非常简单的，但是使用分层采样算法时，要解决的最大的问题就是以哪一列或哪几列分组。如果一个分层采样的样本是以某个列组合 S 分组的话，那么询问任何 S 当中的列或者列组合时，该样本都能保证不丢组。同时，这份分层采样的样本也能处理关于 S 之外的列的询问，只是不丢组的性质就不能保证了。为了叙述方便，我们在后面的实验中将 S 叫做“预设组”，将只涉及 S 当中列的询问叫“预设组询问”。

事实上，根据业界最新的研究成果 BlinkDB 的论文，对于某张高维数据表，绝大部分的查询所涉及到的列都是固定的几列。他们调查了 Facebook 一周的查询历史记录，发现有 90%的查询都落在了 20%的列组合上，并且这些列组合即使随着时间的推移也是相对稳定的。因此，我们完全可以让用户自己给定自己相对感兴趣的列，这些列数并不会很多，然后我们以这些列为组进行分组采样。即使用户对于自己的选择在一段时间之内不甚满意，也可以重新指定另外的一些列重新进行采样，因为分层采样的复杂度虽然略高于一般随机采样，但是相对于其他数据压缩方法也非常低，因此对原数据进行多份采样也是可以接受的。

2.2 分层采样算法的设计

如上节所述，分层采样的首要任务是要指定一些列，将这些列视为组。指定列的工作在本系统中由用户完成。其次是确定每组的采样率。虽然标准的分层采样是组内的采样率固定为总体的采样率，但是因为我们要做的是统计推断，如果一个组的条目数过少的话，会影响推断的准确度。因此我们做如下设计：

- 若某一组预定的采样条数（即大数据条数×采样率）小于给定的阈值 K，那么该组固定采样 K 条，不足 K 条则全部保留；
- 否则，按采样率导出的样本条数进行采样。

2.3 采样推断公式

由于篇幅限制，本部分中仅给出各聚合函数的概率密度函数推导的结果。另外，由于每个组的采样率可能不同，所以最终结果需要根据每一组的采样率进行一定的修正。

2.3.1 Count 聚合函数

定理 2.1（Count 真实值的分布）设在小数据中，第 i 组数据出现了$\mathrm{n_i}$条，那么大数据中第 i 组数据的条数$\mathrm{N_i}$服从$B(N,p_i)$，其中$\mathrm{p_i}=\frac{n_i}{n}$为估计的第 i 组的出现概率。

当我们已经求得$\mathrm{N_i}$的分布之后，利用二分查找的方法就很容易求得上下界$[u_l,u_r]$了。若$\mathrm{p_i}$很小，也可以将二项分布近似为泊松分布或正态分布，从而利用查表等方法更加简单地求得置信区间。

2.3.2 Avg 聚合函数

定理 2.2（Avg 真实值的近似分布）：当样本容量足够大的时候，每组的 Avg 真实值近似服从于正态分布，且其均值$\mu=\overline{\mathrm{v_i}}$，方差$\sigma^2=\sum_{\mathrm{n_i}=0}^{\mathrm{n}}\mathrm{B(n,p_i)}\left(\overline{\mathrm{v_i}}^2+\frac{\mathrm{S_i^2}}{\mathrm{n_i}}\right)-\overline{\mathrm{v_i}}^2$，其中$\overline{v_i}$是样本中第 i 组的平均值，$S_i^2$是样本中第 i 组的方差，$n_i$是样本中第 i 组的条目数，$n$是样本总条目数，$p_i=\frac{n_i}{n}$是估计的第 i 组在在大数据中出现的频率。

2.3.3 Sum 聚合函数

我们知道，$\mathrm{Sum_i}=N_i*Avg_i$，其中N_i是大数据中第 i 组的条目数，而在上一节中我们已经得出$Avg_i\sim\mathrm{N}(\mu,\sigma^2)$，所以我们可以得到$Sum_i\sim N(N_i\mu,N_i^2\sigma^2)$，$N_i$可以通过$N_i=n_i*\frac{N}{n}$来估计。这样我们就推导出了 Sum 的概率密度函数。

3. 系统设计与实现

本系统主要有 3 个部分构成：前端界面，后端服务器，以及负责运算和存储的集群。其中，UI 部分使用 C#实现，服务器部分使用 Scala 和 Java 混合实现，UI 和服务器之间通过 JSON 通信。服务器后端连接一个 Hadoop 集群，其中大数据存储部分用 HDFS 完成，查询部分用 Spark 框架完成。

HDFS 是一个分布式大数据存储框架，其基本原理是将原数据切分成很多份，每一份可以有一个或多个副本，这些副本分别放在不同的机器上，从而实现分布式存储和容灾机制；Spark 是一个比 Map-Reduce 更加高效的分布式运算系统，其主要的原理也是先将数据切分，将切分后的数据划分为一组任务分配给多台机器并行执行，最后再合并结果。由于其将中间结果写入内存而不是硬盘，所以其速度比 Map-Reduce 高很多。在 Spark 核心之上的 Spark SQL 组件也提供了很多非常适合本系统调用的接口。

服务器部分主要的系统结构图 1 所示，UI 发来的 JSON 请求将首先被接口层的 JSON 解析器解析，之后根据其请求类型交给相应的 Handler。其中，小数据事务 Handler 主要负责调用推断算法模块，处理针对采样进行的统计推断请求；大数据事务 Handler 主要负责调用 Spark 引擎处理新建采样请求和大数据验证请求。处理完成后再经由接口层将结果序列化为 JSON 返回 UI。

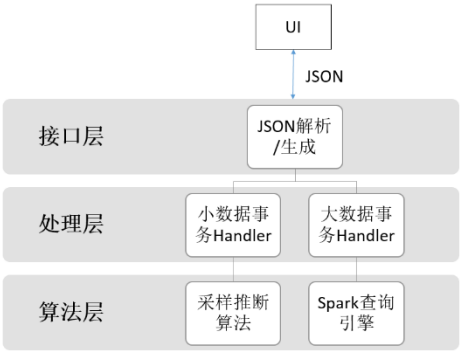

图 1 服务器的系统结构

4. 实验结果与分析

评价本系统的实验主要性能测试和准确度测试两项。性能测试中主要测试系统的响应速度，准确度测试中主要测试采样推断模块的准确率。

在性能测试中我们选择了一份 10GB 的实际数据，分别用我们的采样推断算法和 Spark 引擎运行相同的 10 条查询。实验结果表明，统计推断算法在小数据上执行查询的速度相当快，平均每条查询仅需要 2.3 秒，用户几乎无需等待，就可以立刻观察到结果；而 Spark 引擎对于每条查询都要耗费平均 5 分钟左右，远远超出了用户能忍耐的时间。所以，使用采样推断的方法，能够显著减少用户等待的时间，满足用户“交互”探索数据的需求。

在准确度中我们采用与上部分完全相同的测试数据，并分别用分层采样算法和简单随机抽样算法制作该数据的两份样本。我们在置信度 90%的情况下，分别在两份样本上运行了相同的 100 条查询，其中 80 条查询是在分层采样算法的预设列上进行的。实验结果表明，对于预设列询问，分层采样算法的准确度明显好于随机采样，分层采样算法的正确率在 88%左右，而随机采样算法在 75%左右；从相对误差的角度来看，两种采样算法的平均相对误差在预设列询问上相差极大。分层采样在预设列上的相对误差为 15.3%，而随机采样算法的相对误差高达 48.8%，其最主要的原因就是简单随机采样算法在预设列上的丢组问题，这个问题导致其对于小类根本无法准确估计，以至于产生了 100%的相对错误。

结论

本文讨论了一种基于采样推断的大数据处理系统，该系统主要分为采样推断和大数据验证两大部分。用户可以通过先对原数据进行采样，然后在原数据上的样本上进行交互式探索来快速发现其感兴趣的结论，进而再调用大数据验证模块有目的地进行查询，验证自己的假设，从而得出结论。实验结果表明，该系统的查询效率比直接利用大数据处理引擎要提高很多，采样推断准确度比起一般的随机采样有所提高，能够适应用户交互式探索大数据的需求。同时，该系统还可以与数据挖掘算法结合，自动地给出数据中潜在的规律，进一步方便决策者的使用。

但是，本文的研究内容尚有继续挖掘的空间。例如，在目前的系统中，我们是让用户自己选择预设列组合的，但是如果我们能够研究出一套学习算法，使得系统可以自动选择最佳的预设列组合的的话，毫无疑问可以让系统的准确度更上一层楼，更加改善用户的使用体验。这只是本文相关的未来工作的一例，如能在相关领域进行进一步研究的话，这套系统一定会变得更加智能、准确。

参考文献

[1] Cukier K. Data, data everywhere: A special report on managing information[M]. Economist Newspaper, 2010.

[2] Spark [EB/OL]. Apache. 2016. http://spark.apache.org/

[3] Cormode G, Garofalakis M, Haas P J, et al. Synopses for massive data: Samples, histograms, wavelets, sketches[J]. Foundations and Trends in Databases, 2012, 4(1–3): 1-294.

[4] 王松桂，张忠占，程维虎. 概率论与数理统计（第三版）[M]. 科学出版社, 2013：146-156.

[5] Agarwal S, Mozafari B, Panda A, et al. BlinkDB: queries with bounded errors and bounded response times on very large data[C]//Proceedings of the 8th ACM European Conference on Computer Systems. ACM, 2013: 29-42.