# 一个直接处理csv文件的方法

汪兴元

2015 年 7 月 21 日

**摘要**

我们常常需要处理体积很大的csv数据或日志文件。一般来讲，导入RDBMS来处理是常用的一种办法，但此办法也不完美，尤其是SQL这种描述型的语言。在表达算法的时候不如一般的程序设计语言的表达力强，某些应用场合虽然能够实现，但是难度太高，优化不易；批量处理大数据，通过在RDBMS上运行SQL效率并不高。

另一种办法就是使用Hadoop或Spark，或导入NoSQL中，再使用MapReduce。但是使用类似Hadoop之类的工具又可能太重，数据量不够运作一个Hadoop集群，还得承受其代价。

本文界绍了直接处理csv文件的一套办法，作为另一个数据处理的选项，示例代码用python。在从原始数据到最终的目标结果，需要组合本文提到的各种算法，才能达到目的。通过管道-过滤器连接每一个算法，能够将整个处理算法分而治之，同时得到比较好的性能。

本文假设要处理的整个数据集无法一次装入机器的内存。

# 目录

# 1 校验数据

csv文件可能存在错误。在正常情况下，csv文件不应存在错误。但是写入csv的过程并非事务型的，不象RDBMS那样有很好的ACID保证，故磁盘耗尽，进程异常退出、挂起等因素，直接导致csv文件的格式并非预期。

检查数据没有统一的方法，要视数据自身的特点来做检查。一般是检查一些数据正确所需的必要条件，但必要并不一定充分。一般的检查方法有：

1. 检查列数。如果列数不等于预期值，可以确定此行数据错误。
2. 检查每列数据的格式，比如，数字，日期，或其它满足指定格式的文本串。可以偿试将其解析为对应的类型，看能否成功；或用正则表达式验证。
3. 校验字段的值。前两项都对的情况下，可以检查数据值的取值范围。比如健在的人的年龄不能是负数，也不能是数百以上；历史数据中的记录时间不可能超过当前的日历时钟。

如果数据是已经通过有严格校验的系统中生成或导出的，原始数据本身无误，一般做以上三项检查可以查出我们能见到的全部错误。但是如果数据是人工填写或是有故障的机器生成的，这三项检查仍不能确保检查出全部错误。