R 中大规模数据的整理与分析

邱怡轩

统计之都

2012 年 5 月 26 日

第五届中国 R 语言会议

概要

第五届中国 R 语言会议

1. 大数据?
2. R ☯ 数据库
3. R @ ff
4. 一个你可能感兴趣的例子

概要

第五届中国 R 语言会议

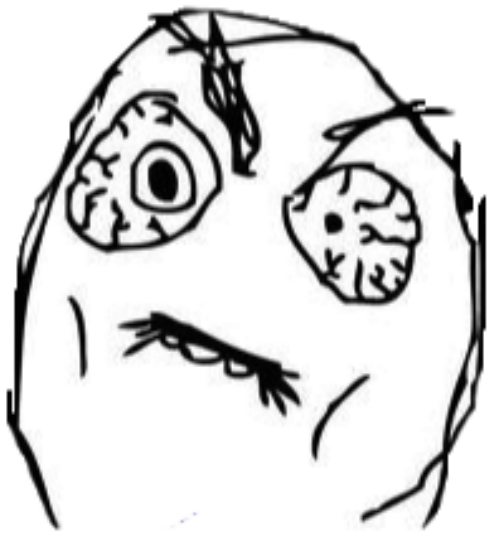
你眼中的大数据?

你眼中的大数据?

你眼中的大数据?

useR眼中的大数据

```
> n = 5000000;
> p = 5;
> x = matrix(rnorm(n * p), n, p);
> x = cbind(1, x);
> bet = c(2, rep(1, p));
> y = c(x %*% bet) + rnorm(n);
> lm(y ~ 0 + x);
错误: 无法分配大小为38.1 Mb的矢量
此外: 警告信息:
1: In lm.fit(x, y, offset = of
  Reached total allocation of
2: In lm.fit(x, y, offset = of
  Reached total allocation of
3: In lm.fit(x, y, offset = of
  Reached total allocation of
4: In lm.fit(x, y, offset = of.
  Reached total allocation of
>
```



概要

第五届中国 R 语言会议

2 R ☯ 数据库

简介

- 如果原始数据很大，但用来建模的数据较小，则可以先在数据库中进行整理，然后通过 R 与数据库的接口提取数据
- 数据库适合存放和整理比较规整的数据，和 R 中的数据框有良好的对应关系，这也是 R 中绝大多数统计模型的标准数据结构
- 数据库中大部分的运算都不需要消耗很大的内存

R 中的数据库接口

- RODBC
- DBI 系列 (ROracle, RMySQL, RPostgreSQL, RSQLite)
  - 在 R 的层面上遵循相同或类似的语法
- RMySQL
  - http://biostat.mc.vanderbilt.edu/wiki/Main/RMySQL
  - http://www.google.com.hk/search?q=site:http://cos.name/cn/%20RMySQL
- RPostgreSQL
  - http://www.road2stat.com/cn/r_language/rpostgresql.html

数据库对比

重量级 (1G+)，完整的商业支持。

中量级 (30M~50M)，功能强大。

轻量级 (~300K)，最大 (？) 的优势在于数据库直接保存为单个文件，便携性强。在 R 中无需另外安装。

连接数据库的主要步骤

1. 打开数据库

2. 提交 SQL 语句

3. 提取数据

4. 关闭数据库

打开数据库连接

MySQL™ & PostgreSQL

```
# 使用 RPostgreSQL 时将 MySQL() 改成 dbDriver("PostgreSQL")
con = dbConnect(MySQL(), user = "root",
    password = "donttellyou", dbname = "my_db")
```


```
con = dbConnect(dbDriver("SQLite"), dbname = "my_db.db")
```

查看表名、字段名


```
table.names = dbListTables(con)
fields.names = dbListFields(con, "my_table")
```

提交查询


```
# 结果集
res = dbSendQuery(con, "select * from my_table")
# 获取记录，n = -1 表示获取结果集中的所有记录
dat = fetch(res, n = 100)
```

一步到位：

```
dat = dbGetQuery(con, "select * from my_table limit 100")
```

关闭数据库连接


```
dbDisconnect(con)
```

小结

- 只取出你需要的数据到内存
- 重点是编写 SQL 语句
- 一个用 R ☯ SQL 做回归的例子
  - http://yixuan.cos.name/en/2011/10/how-to-run-regression-on-large-datasets-in-r/
- 可以使用 `biglm` 软件包直接对数据库中的表拟合广义线性模型，包括普通回归、Logistic 回归等

概要

第五届中国 R 语言会议

ff 简介

- 简单地说，ff 软件包的作用就是构建“硬盘版”的向量、矩阵和数据框
- 内存中的对象可以通过 ff() 函数转存到硬盘中，而硬盘中的对象也可以通过取下标的形式返回内存
- ff 软件包及其扩展 ffbase 可以用很小的内存对数据进行读写、排序、汇总、更新等操作，其作用类似于数据库，但与 R 联系紧密，操作方式与 **R** 中类似，无需编写复杂的 SQL 语句

快速入门

- 创建一个长度为 1 亿、存储类型为 double 的 ff 向量

```
library(ff)
v = ff(vmode = "double", length = 1e+08)
```

- 向量的真实存储位置，对象所占内存 vs. 对象所占硬盘

```
filename(v)
object.size(v)  # 内存
file.info(filename(v))$size  # 硬盘
```

- 对向量元素进行赋值

```
v[1:1000] = rnorm(1000)
v[c(1:100, length(v))]
```

快速入门 [2]

- 生成 ff 数据框

```
v1 = ff(vmode = "integer", length = 100)
v2 = ff(rnorm(100))
dat = ffdf(v1 = v1, v2 = v2)
```

- 取元素子集

```
dat[1:5, ]; dat[, 2] # data.frame 对象
dat[1]               # ffdf 对象
dat[[1]]; dat$v2     # ff 对象
```

- 读/写纯文本文件

```
dat2 = read.csv.ffdf(file = "my_data.csv")
write.table.ffdf(dat, file = "export.txt", sep = "\t")
```

快速入门 [3]

- 数据汇总

```
library(ffbase)
v = ff(rnorm(1e6)); mean(v); quantile(v)
# 目前可能有 bug，只能通过下述方法调用，修复后可直接 hist(v)
ffbase:::hist.ff(v)
```

- 大数据抽样

```
bigsample(1e+08, 10)
bigsample(v, 10)
```

- 排序

```
v.sorted = ffsort(v)
print(v); print(v.sorted)
```

进阶

- 每一个 `ff` 对象都包含 R 中的对象和硬盘数据两部分
  - 硬盘中存储真实的数据
  - 内存中记录相关的文件信息，相当于一个“指针”
  - 因此，`ff` 对象是“按引用”传递的

```
x = ff(rnorm(10))
y = x
y[1:10] = 0
x
```

- 要复制对象，需要使用 `clone()` 函数

```
y = clone(x)
y[1] = 1
x
```

进阶 [2]

- 将对象保存至工作空间，以便下次使用

```
obj = ff(rbinom(10000, 100, 0.5), vmode = "integer")
ffsave(obj, file = "D:/tmp/my_obj")
```

- 删除对象的硬盘数据文件，再删除整个对象

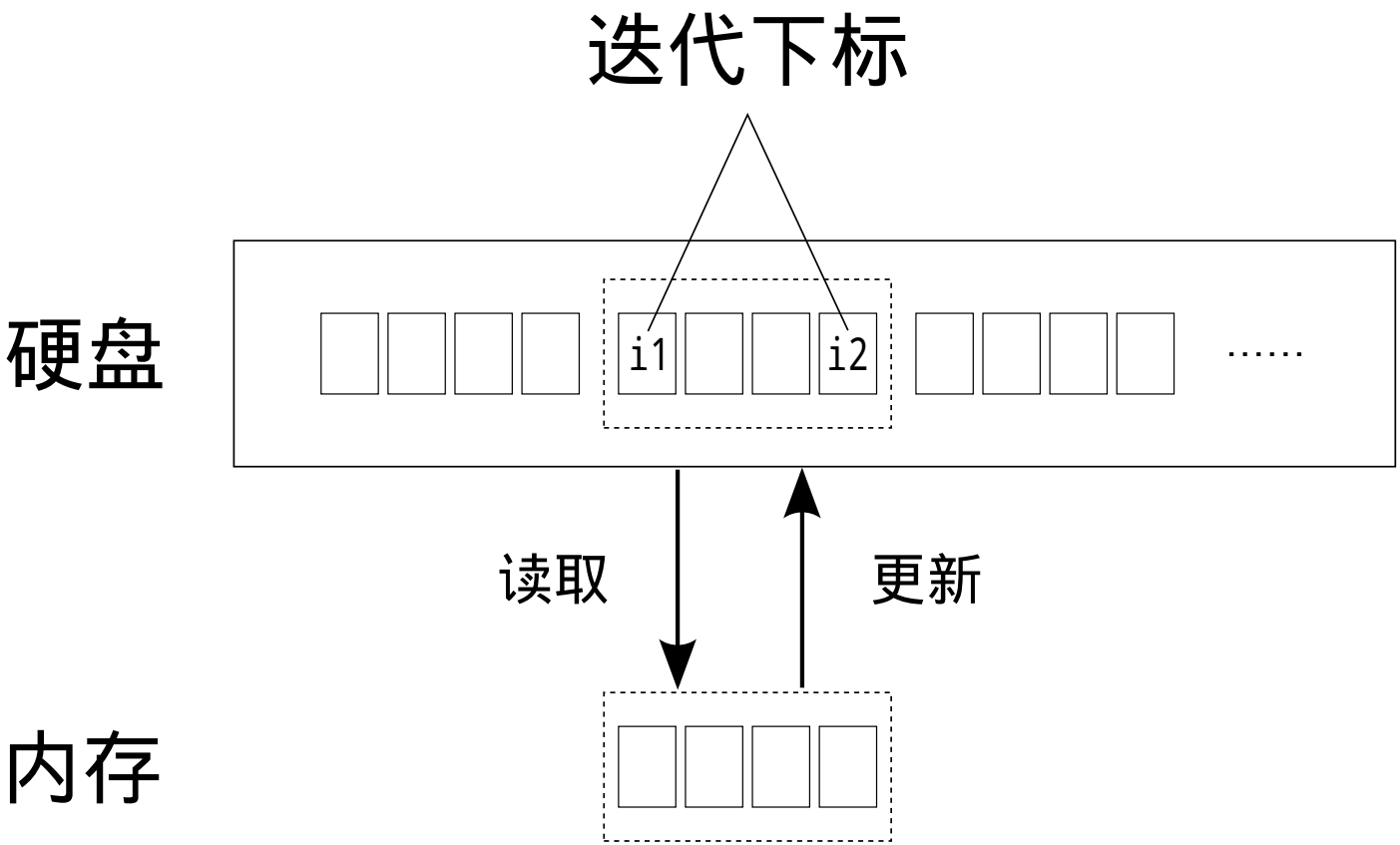
```
# 顺序不能反!
delete(obj)
rm(obj)
```

- 加载之前已经保存的对象，并将其打开

```
ffload("D:/tmp/my_obj")
open(obj)
obj
```

精华

- 大数据处理的一般思路：分而治之



精华 [2]

- 生成长度为 1 千万的正态随机数变量，每次对 10 万个元素赋值

```
v = ff(vmode = "double", length = 1e+07)
ffvecapply(v[i1:i2] <- rnorm(i2 - i1 + 1), X = v,
    BATCHSIZE = 1e+05, VERBOSE = TRUE)
```

- 求向量前 20 万个元素之和

```
s = 0
ffvecapply(s <- s + sum(v[i1:i2]), X = v, N = 2e+05)
```

精华 [2]

- 生成长度为 1 千万的正态随机数变量，每次对 10 万个元素赋值

```
v = ff(vmode = "double", length = 1e+07)
ffvecapply(v[i1:i2] <- rnorm(i2 - i1 + 1), X = v,
    BATCHSIZE = 1e+05, VERBOSE = TRUE)
```

- 求向量前 20 万个元素之和

```
s = 0
ffvecapply(s <- s + sum(v[i1:i2]), X = v, N = 2e+05)
```

- 更多的用法，发挥你的想像力吧

小结

- `ff` 可以将 R 中的向量、矩阵和数据框存储到硬盘，并与内存对象进行交互
- 利用 `ff` 特有的 `apply` 系列函数可以方便地对 `ff` 对象进行循环、迭代、汇总等操作
- `ff` 对象暂时不支持字符型向量，完整的类型支持列表请？`vector.vmode`

概要

第五届中国 R 语言会议

4 一个你可能感兴趣的例子

微博主题词

- 有人的地方就有江湖，互联网也不例外
- 你总是可以看到“××大战”和“××之争”这类的话题
- 对于争战的双方，网民们都在谈论什么？

微博主题词

- 有人的地方就有江湖，互联网也不例外
- 你总是可以看到“××大战”和“××之争”这类的话题
- 对于争战的双方，网民们都在谈论什么？

- 我们想找到与某一个主题相关的“关键词”

分析框架

- 以“方韩”为例，我们想知道与他们各自相关的主题词都有哪些
- 分析思路
  - 获取与“方韩”相关的微博各约 2 万条
  - 对微博内容进行分词，构造词频矩阵，即每条微博是一个观测，语料库中每个词语出现的频率是自变量，微博的主题 (关于“方”-0 或关于“韩”-1) 是二分因变量
  - 对因变量做回归并进行变量选择，找到那些最能将因变量区分开来的词语，并判断系数的正负，负的认为是“方”的主题词，正的认为是“韩”的主题词

分析框架

- 以“方韩”为例，我们想知道与他们各自相关的主题词都有哪些
- 分析思路
  - 获取与“方韩”相关的微博各约 2 万条
  - 对微博内容进行分词，构造词频矩阵，即每条微博是一个观测，语料库中每个词语出现的频率是自变量，微博的主题 (关于“方”-0 或关于“韩”-1) 是二分因变量
  - 对因变量做回归并进行变量选择，找到那些最能将因变量区分开来的词语，并判断系数的正负，负的认为是“方”的主题词，正的认为是“韩”的主题词
- 重点在于展示大规模数据的分析过程，分析结果仅供参考 ☺

主要流程

- 访问http://s.weibo.com/weibo/<search_content>&rd=MjAxN&page=<page_number>抓取微博的 HTML 文本 (感谢 @ 波波头一头)，注意抓取网页的频率限制
- 利用 XML 软件包解析 HTML，获取微博的文本内容
- 利用 rmmseg4j 软件包进行中文分词
- 将数据导入数据库，进行简单的处理 (大写转小写，删去某些词语)

主要流程

- 利用 `ff` 软件包读取数据，并结合 `Matrix` 软件包构造稀疏词频矩阵
  - 32147 条微博，10711 个词语，完整的词频矩阵大小为 2.57G，稀疏化后仅占据 11.76M
- 使用 Lasso 对回归进行变量选择，核心算法用 `Rcpp` 软件包和 C++ 实现
- 结果使用 `wordcloud` 软件包进行可视化展示

结果


真不
就能
哈
bigfounder
科学
薛
算
生
专门
这次
知道
嘛
抽
亮
城
楠
了
障碍
同时
想
欠
假
罗永浩
之
老
取消
打
不
韩寒
娘
妈
达
教
更
妨碍
斗士
古人
底线
风
舟子
南
建
咬
试验
发言
辩
福
道歉
主持人
菊花
特色
薛蛮子
怎么
罗
方面
恶劣
是个
据说
默念


韓寒
我爱
学会
文章
自然
好戏
起了
大眼
看完
甚至
昨晚
错的
提到
就做
虽然
时候
角度
民主
台湾
马英九
本
也是
米
文化
這
中
在
少年
事件
量身
想做
即使
锥
方舟子
也有
严
十
巴
大陆
此事
10
靠
文
那么
可
博
目
孙
们
所
左
书
电影
认为
先是
健康
戴
是因为
身
猪
敬
搞得
图
171
置疑
朋友
风
如
删
判断
林
青春
惊奇
三重
169
我以为

CAPITAL OF STATISTICS

PROFESSION, HUMANITY & INTEGRITY

谢谢！