用 bignmf 进行非负矩阵分解

潘岚锋 邱怡轩 魏太云

- 一个具有统计味道的算法
- 一个软件包
- 两个例子

非负矩阵分解在图像处理和文本挖掘领域有广泛的应用。非负约束导致分解得到的矩阵具有强稀疏性，使结果易于解释。

$$V^{m\times m} \approx W^{m\times r} H^{r\times m}$$

其中 $V, W, H \geq 0$， $r \ll \min(m, n)$

使用最简单的平方和损失函数，

$$\min_{W,H\geq 0} \|V - WH\|^2$$

为什么非负？

如果没有非负约束，平方和损失下的最优解就是奇异值分解的前 r 个特征根对应的特征向量。但是奇异值分解得到的矩阵通常看不出来实际意义，仅仅是方差最大的因子 1，方差第二大的因子 2。。。

非负矩阵分解的思路与因子分析类似，以因子的非负线性组合来最大程度的表达原数据。

非负约束的结果一：因子是稀疏的，每个因子只与部分样本有关，易于解释；二：因子的系数是稀疏的，是数据本身也可以得到解释。

如果已知这些因子，那么非负矩阵分解就是一个简单的多响应变量回归问题 (要求系数非负)。

在因子未知的情况下，将因子设为随机值，得到系数，再根据系数求因子，然后继续求系数。。。

重复这个过程会发现，因子不再是随机的，而是包含了原数据的信息的。

已有算法

非负矩阵分解已经有了很多算法，例如 Multiplicative Update，Projected Gradient Method，Gradient Descent。交替最小二乘法比较符合上面的解释，具有更好的统计意义，而且收敛性质很好，计算速度快。

已有的交替最小二乘法都使用冷冰冰的二次规划算法求解非负回归，我们使用的最小角回归的思路解非负回归，更有统计的感觉，而且速度更快 (至少不会慢)。

交替最小二乘

1. 初始化 W
2. 针对 V 的每一列 $V_{\cdot j}$，以 W 为自变量求
   $h_j = \arg\min_{h_j \geq 0} f(V_{\cdot j}, Wh_j)$，得到 $H = (h_1, h_2, \ldots, h_m)$
3. 针对 V 的每一行 $V_{\cdot j}$，以 H 为自变量求
   $w_i = \arg\min_{w_i \geq 0} f(V_{i\cdot}^T, H^T w_i)$，得到 $W = (w_1, w_2, \ldots, w_n)^T$
4. 重复 2、3 步直至收敛条件满足

bignmf 包

bignmf 包就是求解上面说的问题。只有两个函数：

- bignmf 用于一般矩阵的分解
- bignmfsp 用于稀疏矩阵的分解，接受用 Matrix 包生成的稀疏矩阵

用法都非常简单，主要两个参数，一个设定矩阵，一个设定秩

例子

```
v_mat <- matrix(rexp(60000,2), 200, 300)
system.time(re <- bignmf(v_mat, 5))
v_mat <- matrix(rexp(6000000,2), 2000, 3000)
v_mat[v_mat < quantile(v_mat, .1)] <- 0
system.time(re <- bignmf(v_mat, 20))
```

输出

输出长度为 3 的列表：

W $n \times r$ 的矩阵

H $r \times m$ 的矩阵

iteration 迭代次数

为什么叫 bignmf?

- R 中已经有包叫 NMF 做非负矩阵分解，它的速度不敢恭维
- 字母少
- 理论上可以处理比较大的数据，目前尚未完全实现

C++ 实现

借助伟大的 Rcpp 和 RcppEigen 包，算法内层用 C++ 代码实现

- cdouterloop 函数负责用坐标下降法解非负回归，内部各种显式循环
- wupdate 把 V 按行拆成 n 个小的回归问题，以 H 为自变量，求解 W
- hupdate 把 V 按列拆成 m 个小的回归问题，以 W 为自变量，求解 H
- spwupdate 和 sphupdate 分别是稀疏版本

稀疏矩阵

C++ 库 Eigen 支持稀疏矩阵的运算。RcppEigen 可以把 Matrix 包生成的稀疏矩阵传递到 C++ 中。

直接把 Matrix 包得到的系数矩阵输入函数 bignmfsp 中就可以

支持并行

非负矩阵分解可以很好的支持并行运算

wupdate 和 hupdate 都是把 V 拆成多个小的回归问题，这些小的问题可以分散到多个 cpu 上计算。

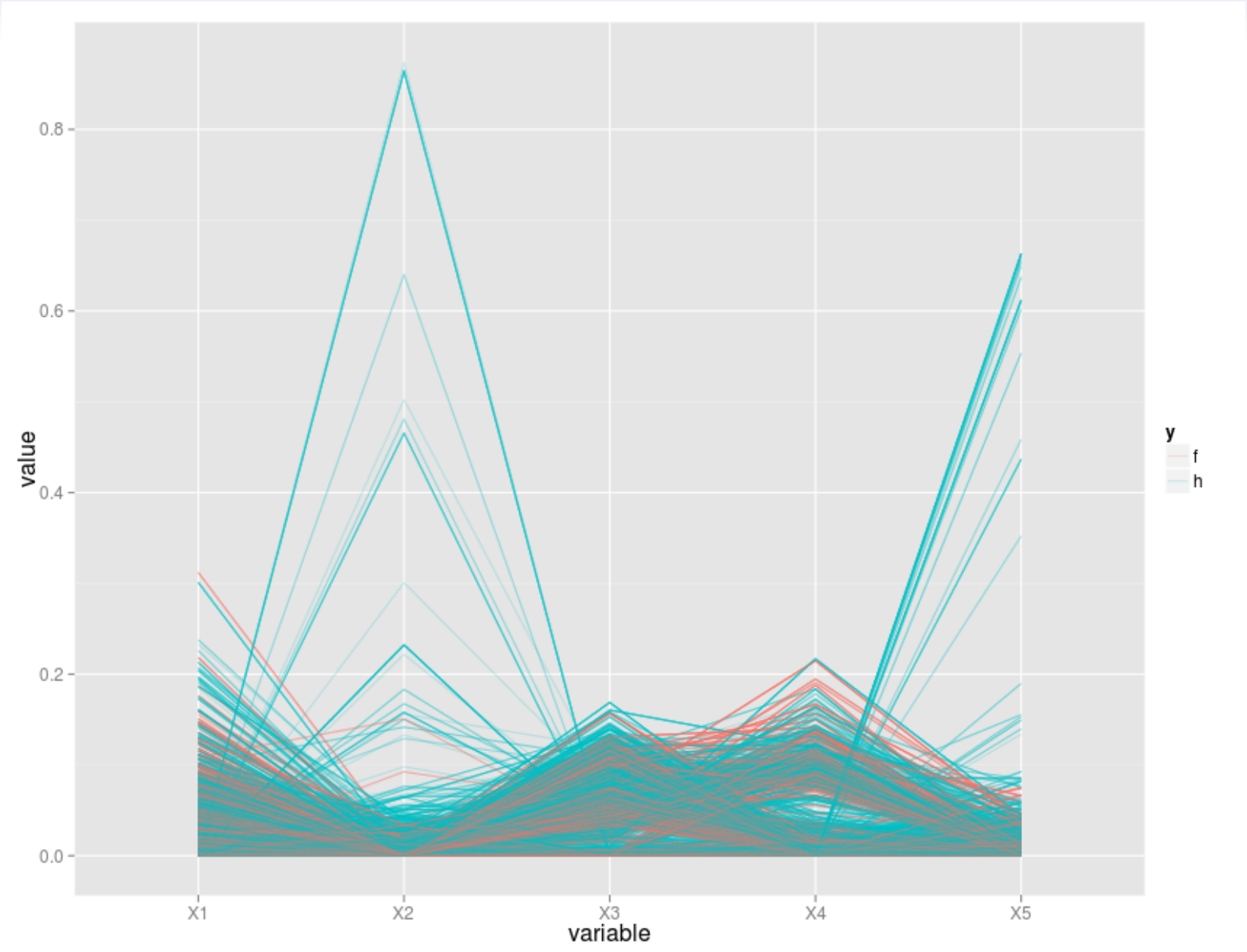
更大？

即使 V， W， H 三个矩阵的大小都远远超过了内存也没有关系。把原矩阵和结果矩阵分散成许多个小部分，每次读取一小部分可以处理的数据，分别计算。

1. 每次读取 V 和 W 的一行或几行，计算 W^TW 和 W^TV
2. 读取 W^TV 的一列或多列，计算出 H 对应的列；在得到 H 的列的同时，可以开始计算 HH^T 和 VH^T
3. 读取 VH^T 的一行或多行，计算 W 对应的行；同时，可以开始计算 W^TW 和 W^TV
4. ……

方韩之争微博数据

将数据合并成一个 32147×10711 的词频矩阵，前 18944 行为搜索”韩寒“的微博，后 13203 行为搜索“方舟子”得到的微博。共涉及 10711 个词。取秩为 5，对这个矩阵进行分解得到 W 和 H。


0.8
0.6
0.4
0.2
0.0
value
X1
X2
X3
X4
X5
variable
y
f
h

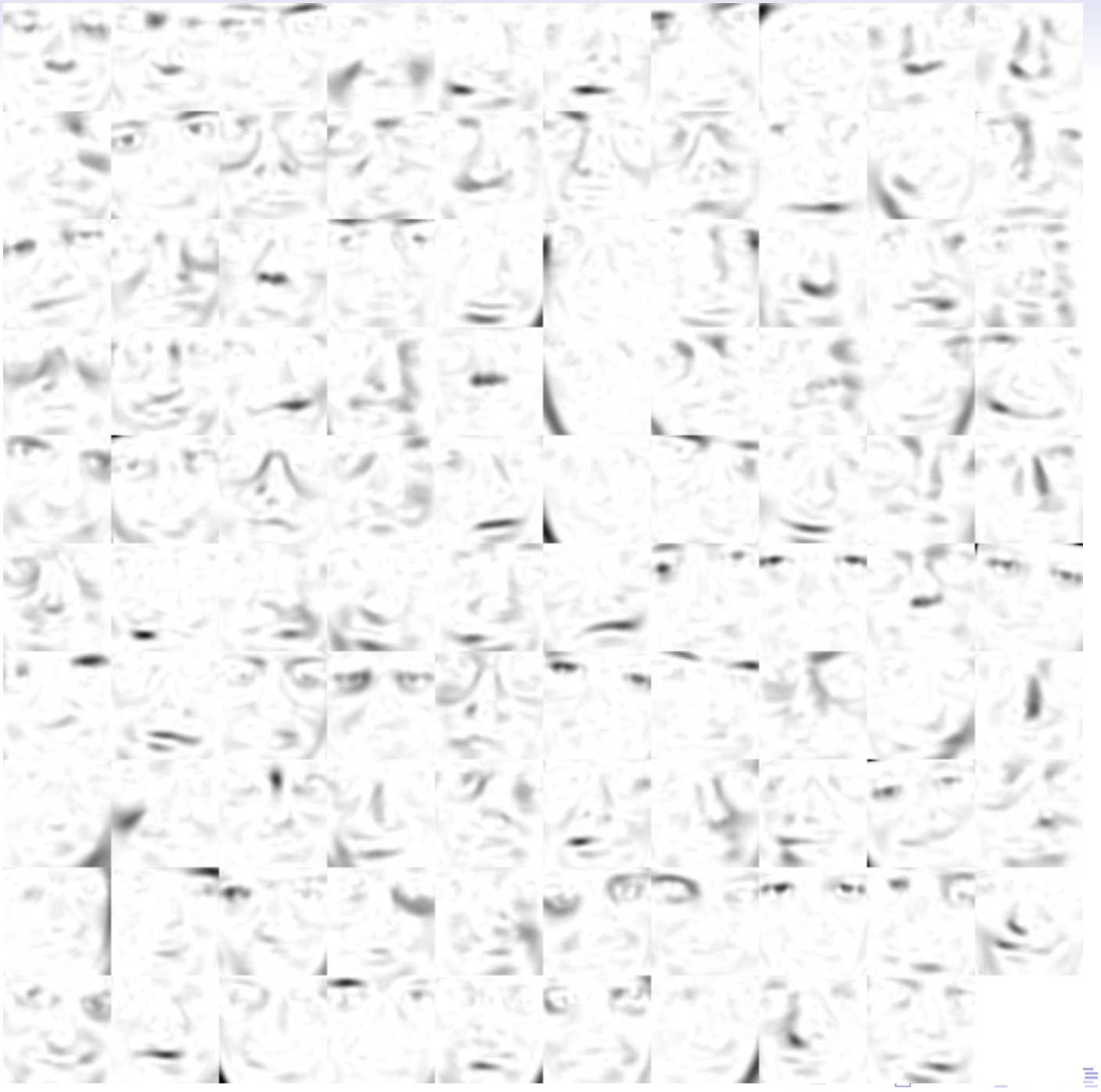
方韩之争微博数据

韩寒有独特的特征词，而方舟子没有很特殊的词。

“一部分, 不, 不只, 中, 九, 也是, 人民日, 令人, 作家, 創, 助人, 動, 司, 善良, 大, 已, 常, 心, 意, 感, 拾, 文, 文化, 日常生活, 昧, 最近, 核心, 機, 歷, 灣, 熱, 物, 生活, 發, 眼, 知名, 程, 經, 老, 與, 英, 融入, 行, 表, 親, 計, 訪, 認, 讓, 身, 車, 這, 都, 都是, 鏡, 闆, 陸, 震撼, 韓寒”

”171,5,http, 一, 上, 不少, 为, 举行, 也, 人, 作者, 六六, 其, 博, 厘米, 发出, 吗, 图片, 在, 墙, 宴会, 宴请, 家里, 席, 引发, 当时, 微, 手里, 拿着, 据, 日前, 昨日, 有, 此举, 测, 测量, 热, 特地, 真的, 站着, 等, 网友, 蜗居, 袜子, 认为, 议, 请来, 贴, 赤脚, 身高, 还, 还有, 透露, 那么, 重要, 量身, 随后, 高的”

人脸数据



- 一个具有统计味道的算法
- 一个软件包，两个函数
- 两个例子