Alternating Minimization Allgorithms (ALS) and other alternatives

Khaled Fouda

2023-12-18

```
knitr::opts_chunk$set(echo = F, cache = FALSE)
library(kableExtra)
library(magrittr)
library(tidyverse)
knitr::opts_knit$set(root.dir="/mnt/campus/math/research/kfouda/main/HEC/Youssef/HEC_MAO_COOP/")
```

References:

[1] Hastie (2015) Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares [2]

Building on our model in the document "Soft Impute With Covariates", we explore alternative ways of computing the low-rank matrix "B" while achieving the same results. The alternative ways are much faster that they outperform Mao's model.

We begin by introducing the ALS algorithm for estimating B, first without covariates:

Consider our usual model (model 8 in Soft Impute With Covariates):

$$minimize_B \quad \frac{1}{2}\|Y-B\|_F^2+\lambda||B||_*$$

which as the following soluion

$$\hat{B}=S_\lambda(Y)=UD_\lambda V' \quad with \quad D_\lambda=diag[(d_1-\lambda)_+\cdots(d_r-\lambda)_+]$$

where r acts as an upper bound to the rank of $\hat{B}$. Since we don't require the full SVD decomposition to compute $\hat{B}$ and we only need the rows/columns upto r, the authors have discussed the possibility of using alternative methods to do partial svd decomposition. Specifically, they discussed the use of the algorithm "PROPACK" which is an state-of-the-art method of computing SVD decomposition upto certain number of Eigen Values. However, they didn't do it in their implementation and they used full SVD decomposition instead.

A second trick they proposed to avoid the high computations is that they rewrote $\hat{Y}$ as

$$\hat{Y}=[P_\Omega(Y)-P_\Omega(\hat{B})]+\hat{B}$$

$\hat{Y}$ ubove refers to the filled matrix where observed values are left as they are and missing values are filled from $\hat{B}$. The first term above is a very sparse matrix since the missing values are left as 0. The second term is not sparse but is low-rank. Using the two terms instead of $\hat{Y}$ should improve the speed of matrix operations. Again, they proposed it but didn't use it. I attempt to use it but it provided no significant improvement over using $\hat{Y}$ as is. Maybe this will be different for extremely large matrices as I tested with dimensions up-to 1000×1000.