Power Method Examples

David Elsheimer

2/26/2020

Power Method Examples

Usage of the `power_method_dense` function to compute the largest singular value of the following matrices, comparing results with what is provided by the `svd` function in R.

```r
set.seed(12345)
n <- 1e3
p <- 2e3
A <- matrix(rnorm(n*p), n, p)
B <- matrix(rnorm(n*p), n, p)
C <- matrix(rnorm(n*p), n, p)

x<-rnorm(p)
k<-1000
tol<-1e-6
xapproxA<- RStudio2020::power_method_dense(A,x,k,tol)
xapproxB<- RStudio2020::power_method_dense(B,x,k,tol)
xapproxC<- RStudio2020::power_method_dense(C,x,k,tol)

round(norm(A%*%xapproxA,type="2"),3)
```

```
## [1] 76.063
```

```r
round(max(svd(A)$d),3)
```

```
## [1] 76.063
```

```r
round(norm(B%*%xapproxB,type="2"),3)
```

```
## [1] 76.468
```

```r
round(max(svd(B)$d),3)
```

```
## [1] 76.468
```

```r
round(norm(C%*%xapproxC,type="2"),3)
```

```
## [1] 75.928
```

```r
round(max(svd(C)$d),3)
```

```
## [1] 75.928
```

Usage of the `power_method_sparse` function to compute the largest singular value of the following matrices, comparing results with what is provided by the `svd` function in R.

```r
library(Matrix)
set.seed(12345)
n <- 1e3
nnz <- 0.1*n
ix <- sample(1:n, size = nnz, replace = FALSE)
A <- Matrix(0, nrow=n, ncol=n, sparse=TRUE)
```

```r
A[ix] <- rnorm(nnz)
A[1,1] <- 10

ix <- sample(1:n, size = nnz, replace = FALSE)
B <- Matrix(0, nrow=n, ncol=n, sparse=TRUE)
B[ix] <- rnorm(nnz)
B[1,1] <- 10

ix <- sample(1:n, size = nnz, replace = FALSE)
C <- Matrix(0, nrow=n, ncol=n, sparse=TRUE)
C[ix] <- rnorm(nnz)
C[1,1] <- 10

x<-rnorm(ncol(A))
k<-1000
tol<-1e-6

xapproxA<- RStudio2020::power_method_sparse(A,x,k,tol)
xapproxB<- RStudio2020::power_method_sparse(B,x,k,tol)
xapproxC<- RStudio2020::power_method_sparse(C,x,k,tol)

round(norm(A%*%xapproxA,type="2"),3)
```

```
## [1] 13.584
```

```r
round(max(svd(as.matrix(A))$d),3)
```

```
## [1] 13.584
```

```r
round(norm(B%*%xapproxB,type="2"),3)
```

```
## [1] 13.498
```

```r
round(max(svd(as.matrix(B))$d),3)
```

```
## [1] 13.498
```

```r
round(norm(C%*%xapproxC,type="2"),3)
```

```
## [1] 14.081
```

```r
round(max(svd(as.matrix(C))$d),3)
```

```
## [1] 14.081
```

Usage of the `power_method_low_rank` function to compute the largest singular value of the matrices $\mathbf{U}_1\mathbf{V}_1^\intercal$, $\mathbf{U}_2\mathbf{V}_2^\intercal$, and $\mathbf{U}_3\mathbf{V}_3^\intercal$ defined below, compared to what is provided by the `svd` function in R.

```r
set.seed(12345)
n <- 1e3
k <- 10
U1 <- V1 <- matrix(rnorm(n*k), n, k)

U2 <- V2 <- matrix(rnorm(n*k), n, k)

U3 <- V3 <- matrix(rnorm(n*k), n, k)


x<-rnorm(n)
k<-1000
tol<-1e-6
xapproxU1V1<- RStudio2020::power_method_low_rank(U1,V1,x,k,tol)
xapproxU2V2<- RStudio2020::power_method_low_rank(U2,V2,x,k,tol)
xapproxU3V3<- RStudio2020::power_method_low_rank(U3,V3,x,k,tol)

norm((U1%*%t(V1))%*%xapproxU1V1,type="2")
```

```
## [1] 1161.955
```

```r
max(svd(U1%*%t(V1))$d)
```

```
## [1] 1161.955
```

```r
norm((U2%*%t(V2))%*%xapproxU2V2,type="2")
```

```
## [1] 1156.733
```

```r
max(svd(U2%*%t(V2))$d)
```

```
## [1] 1156.733
```

```r
norm((U3%*%t(V3))%*%xapproxU3V3,type="2")
```

```
## [1] 1147.554
```

```r
max(svd(U3%*%t(V3))$d)
```

```
## [1] 1147.554
```

The largest value of the following matrix is estimated below.

$$\mathbf{A} = \mathbf{S} + \mathbf{u}\mathbf{u}^\intercal.$$

```r
set.seed(12345)
n <- 1e7
nnz <- 1e-5*n
ix <- sample(1:n, size = nnz, replace = FALSE)
S <- Matrix(0, nrow=n, ncol=n, sparse=TRUE)
S[ix] <- rnorm(nnz)
S[1,1] <- 10

u <- matrix(rnorm(n), ncol=1)
```

```r
x<-rnorm(n)
k<-1000
tol<-1e-6

max(svd(S+u%*%t(u))$d)
```

```
## Error: cannot allocate vector of size 745058.1 Gb
```

```r
RStudio2020::power_method_dense(S+u%*%t(u),x,k,tol)
```

```
## Error: cannot allocate vector of size 745058.1 Gb
```

```r
xapproxsplr<-RStudio2020::power_method_sparse_plus_low_rank(S,u,u,x,k,tol)

###Largest value
sqrt(t(xapproxsplr)%*%as.matrix((Matrix::t(S)%*%S)%*%xapproxsplr +
                     u%*%(t(u)%*%u)%*%(t(u)%*%xapproxsplr) +
                     u%*%(t(u)%*%(S%*%xapproxsplr)) +
                     (Matrix::t(S)%*%u)%*%(t(u)%*%xapproxsplr)))
```

```
##          [,1]
## [1,] 10004815
```

Both `svd` and `power_method_dense` result in errors, as the matrix is too large and will cause storage issues. Thus `power_method_sparse_plus_low_rank` is what has to be used to compute the result.