Untitled

Shuang Song ss3756

10/23/2019

First we reproduce the example.

```
n <- 1000; p <- 25
beta <- c(1, rep(0, p-1))
X  <- matrix(rnorm(n * p), ncol = p)
svals <- svd(X)$d
max(svals)/min(svals)
```

```
## [1] 1.353864
```

```
N <- 1e4; l2_errors <- rep(0, N)
for (k in 1:N) {
  y <- X %*% beta + rnorm(n)
  betahat <- casl_ols_svd(X, y)
l2_errors[k] <- sqrt(sum((betahat - beta)^2))
}
mean(l2_errors)
```

```
## [1] 0.1581489
```

Now, let us replace the first column of X with a linear combination of the original first column and the second column.

```
alpha <- 0.001
X[,1] <- X[,1] * alpha + X[,2] * (1 - alpha)
svals <- svd(X)$d
max(svals) / min(svals)
```

```
## [1] 1998.964
```

```
N <- 1e4; l2_errors <- rep(0, N)
for (k in 1:N) {
  y <- X %*% beta + rnorm(n)
  betahat <- solve(crossprod(X), crossprod(X, y))
  l2_errors[k] <- sqrt(sum((betahat - beta)^2))
}
mean(l2_errors)
```

```
## [1] 35.95189
```

Now we use ridge regression.

```
lambda <- 0.4
svals <- svd(X)$d
(max(svals) +  lambda) / (min(svals) + lambda)
```

```
## [1] 106.6925
```

```
N <- 1e4; `12_errors` <- rep(0, N)
for (k in 1:N){
  y <- X %*% beta + rnorm(n)
  betahat <- solve(crossprod(X) + lambda, crossprod(X,y))
  `12_errors`[k] <- sqrt(sum((betahat - beta)^2))
```

```
}
mean(`12_errors`)
```

```
## [1] 35.19231
```

We can see that the variance is substancially decreased. By introducing lambda, we avoid dividing by a small σ_{min}, so the condition number is decreased.