sgl

```
library(tidyverse)
library(faux)
library(ggpubr)
```

```
# simulate data with 10 groups and 10 elements per group with within-group correlation of 0.2
set.seed(1)
n <- 200
p <- 100
g <- 10
X <-  matrix(double(n*p), nrow = n)
for (i in seq(1,p,g)) {
   X[, i : (i+g-1)] <- rnorm_multi(n, g, r = 0.2, as.matrix = T)
 }
#beta <- c(sample(c(-1, 1), 50, replace = TRUE), rep(0, 50))
nonzero <- c(10,8,6,4,2,1)
beta <- numeric(100)
j <- 1
for (i in seq(1, 60, 10)){
  a <- sort(sample(c(i:(i+10-1)))[1:nonzero[j]])
  beta[a] <- sample(c(-1,1), nonzero[j], replace = TRUE)
  j <- j + 1
}
# do svd on X for orthonormality
for (i in seq(1, p, g)) X[, i:(i + g - 1)] <- svd(X[, i:(i + g - 1)])$u
y <- X %*% beta + rnorm(n, sd = 4)
```

```
lasso_factory <- function(X, y, lambda, alpha) {
  XX <- crossprod(X)
  Xy <- crossprod(X,y)
  n <- length(y)
  helper <- function(beta){
    reg <- c()
    for (i in seq(1, p, by = g)){
      a <- sum(beta[i:(i+g-1)]^2)
      reg <- c(reg, a)
    }
    reg
  }
  obj <- function(beta){
    base <- helper(beta)
    sum_all <- sum(base^(1/2))
    0.5 *  mean((y - X %*% beta)^2) + lambda*(1-alpha)*sum_all + alpha*lambda * sum(abs(beta))
  }

  soft <- function(thresh, x) pmax(1-thresh, 0)*x
```

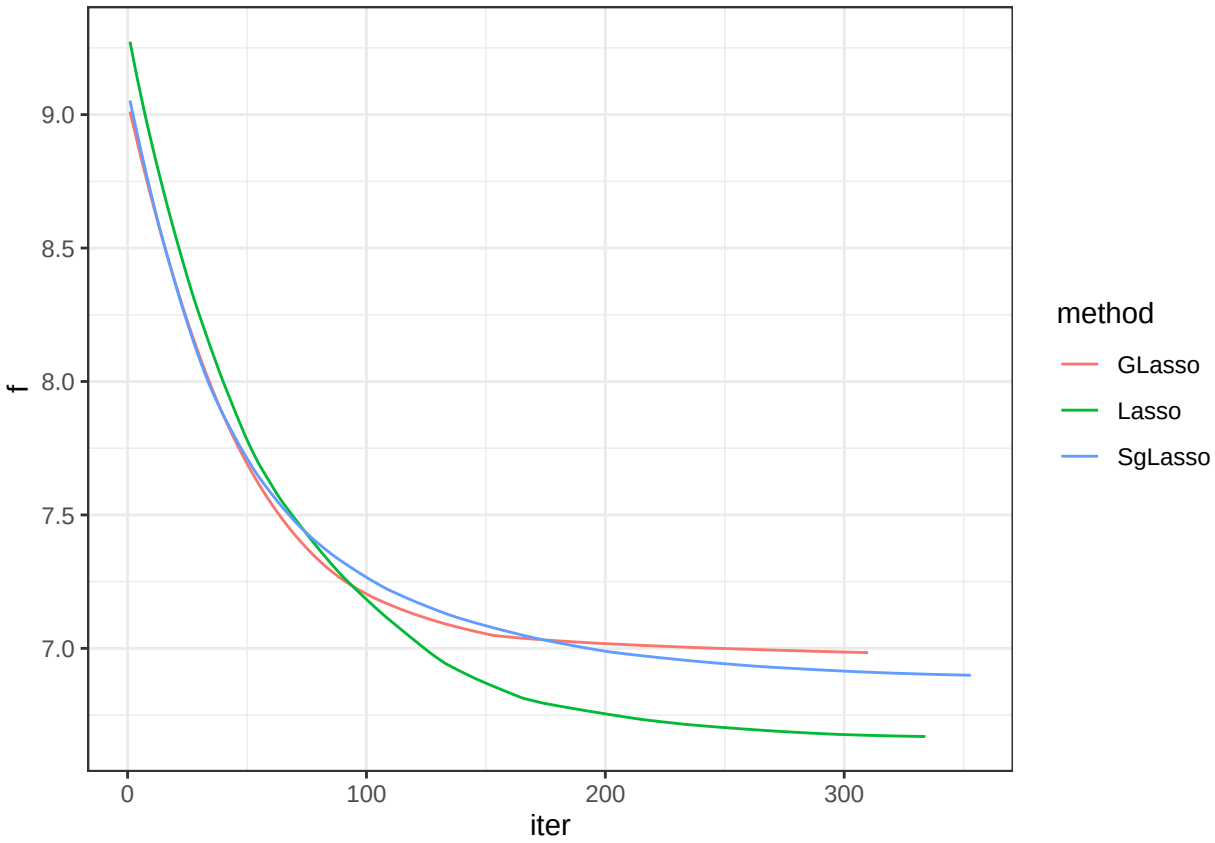
```
  prox_h <- function(beta, x) {
    sign(beta) * pmax(abs(beta) - x, 0)
  }

  maxnorm <- function(x) max(abs(x))

  cd <- function(beta){
    function(beta, gamma, maxit, epsilon){
      r <- y-X%*%beta
      for (i in seq(1, p, by = g)){
        r_k <- r+X[, i:(i+g-1)]%*%beta[i: (i+g-1)]

        if ((sum((prox_h(t(X[, i:(i+g-1)])%*%r_k, alpha*lambda))^2))^(1/2) <= lambda*(1-alpha)){
          beta[i:(i+g-1)] <- 0
        }

        else{
          a <- Inf
          while(maxnorm(beta[i:(i+g-1)] - a) > epsilon){
            s <- prox_h(beta[i:(i+g-1)] - gamma*(-1/n)*t(X[, i:(i+g-1)])%*%r_k, gamma*alpha*lambda)
            if ((sum(s^2))^(1/2) <= gamma*(1-alpha)*lambda){
              beta[i:(i+g-1)] <- 0
              a <- 0
            }
            else{
              a <- soft(gamma*(1-alpha)*lambda / (sum(s^2))^(1/2), s)
              j <- j + 1
              beta[i:(i+g-1)] <- a
            }
          }
        }
       r <- r_k - X[, i:(i+g-1)] %*% beta[i:(i+g-1)]
      }
    beta
  }
}

  return(list(obj=obj,soft = soft, prox_h = prox_h, cd = cd, maxnorm = maxnorm))
}
```

```
lasso_cd <- function(method, maxit = 1000L, epsilon = 1e-4){
  fofx <- numeric(maxit)
  beta <- rnorm(p)
  beta0 <- beta
  maxnorm <- function(x) max(abs(x))
  for (i in seq_len(maxit)){
    update_beta <- method$cd(beta)
    beta <-  update_beta(beta0, 0.5, 200, 1e-4)
    fofx[i] <- method$obj(beta)
    if (maxnorm(beta - beta0) < epsilon) break
    if (i > 1 && abs(fofx[i] - fofx[i-1]) < epsilon) break
    beta0 <- beta
  }
```

```
  if (i == maxit) warning("Maximum iterations (", maxit, ") reached.")
  return(list(x = beta, fofx = fofx[1:i]))
}
```

```
glasso <- lasso_factory(X, y, 0.06, 0)
sglasso <- lasso_factory(X, y, 0.035, 0.5)
lasso <- lasso_factory(X, y, 0.02, 1)
```

```
GLasso <- lasso_cd(glasso)
SgLasso <- lasso_cd(sglasso)
Lasso <- lasso_cd(lasso)
df <- dplyr::bind_rows(
                 Lasso = tibble(f = Lasso$fofx),
                 SgLasso = tibble(f = SgLasso$fofx),
                 GLasso = tibble(f = GLasso$fofx),
                 .id = "method") %>%
  group_by(method) %>%
  mutate(iter = row_number())
df %>%
  ggplot(aes(iter, f, color=method)) +
  geom_line() +
  theme_bw()
```



```r
classification.plot <- function(strategy){
  df <- dplyr::bind_rows(
                Lasso = tibble(beta.pred = sign(Lasso$x)),
                SgLasso = tibble(beta.pred = sign(SgLasso$x)),
                GLasso = tibble(beta.pred = sign(GLasso$x)),
                .id = "method") %>%
                 group_by(method) %>%
  mutate(beta.actual = beta) %>%
  mutate(predictor = seq(1,100)) %>%
  filter(method == strategy)
  plt <- df %>%
  ggplot(aes(x = predictor, y = beta.pred)) +
  geom_point(color = "#00AFBB", size = 1)+
  geom_point(aes(x = predictor, y= beta.actual), shape =1)+
  theme_bw()+
  labs(y = "Coefficient", x = "Predictor", title = strategy) +
  scale_x_continuous(breaks=seq(0,100,10))
  return(plt)
}
glasso.plot <- classification.plot("GLasso")
sglasso.plot <- classification.plot("SgLasso")
lasso.plot <- classification.plot("Lasso")
ggarrange(glasso.plot, sglasso.plot, lasso.plot, nrow = 3)
```