EnetOrRidgeLasso

Niharika

27 September 2016

some initialization

```
q = .01
n=100
p=10
err.sigma = 1
s = 8 #number of active features
true.beta = c(rep(1,s),rep(0,p-8))
```

Method for Data generation.

```
########generate data
generate.data = function(p, n, true.beta, err.sigma)
{
  x.train=matrix(0,n,p)
  x.valid =matrix(0,n,p)

  Z1 = rnorm(n)

  #correlated features
  for(i in 1:5)
  {
    x.train[,i] = Z1 + rnorm(n,0,q)
    x.valid[,i] = Z1 + rnorm(n,0,q)

  }

  #independent features
  for(i in 5:p)
  {
    x.train[,i] = rnorm(n)
    x.valid[,i]=rnorm(n)
  }

  y.train = drop (x.train %*% true.beta) + err.sigma*rnorm (n)
  y.valid = drop (x.valid %*% true.beta) + err.sigma*rnorm (n)

  result = list(x.train = x.train,  y.train = y.train, x.valid = x.valid, y.valid=y.valid)
  return(result)
}
```

generate data

```
d = generate.data(p, n, true.beta, err.sigma)
x = d$x.train
y = d$y.train
```

one set of lambda, alpha

```
lamda_1 = 1
lamda_2 = 2
alpha = lamda_1/(lamda_1+lamda_2)
lambda = lamda_1+lamda_2
```

elastic net, using direct glmnet

```
res.enet = glmnet(x, y, alpha = alpha, lambda = lambda)
```

compute enet applying Lasso on the modified design matrix

```
d = diag(sqrt(lamda_2),p)
x1 = rbind(x, d) #Augment the design matrix with d
y1 =  c(y, rep(0,p)) #Augment the response vector with 0
res.lasso =  glmnet(x1, y1, alpha = 1, lambda = lamda_1)
```

Compare both the estimates

```
res.enet$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             s0
## V1  0.69227984
## V2  0.68127147
## V3  0.69092988
## V4  0.69751775
## V5  0.07860283
## V6  .
## V7  .
## V8  0.04903522
## V9  .
## V10 .
```

```
res.lasso$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##               s0
## V1  0.6820055668
## V2  0.5592687045
## V3  0.8475913365
## V4  0.8979156302
```

```
## V5  0.0748122558
## V6  .
## V7  .
## V8  0.0006395835
## V9  .
## V10 .
```

Compare output with another set of lambda, alpha

```
lamda_1 = 0.5
lamda_2 = 1.5
```

Compare both the estimates

```
res.enet$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##            s0
## V1  0.8232708
## V2  0.8113790
## V3  0.8214587
## V4  0.8281501
## V5  0.4535209
## V6  0.3274273
## V7  0.3213348
## V8  0.3895354
## V9  .
## V10 .
```

```
res.lasso$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##            s0
## V1  0.7907010
## V2  0.7108635
## V3  1.0016791
## V4  1.0051485
## V5  0.5907084
## V6  0.3961987
## V7  0.3910906
## V8  0.4703945
## V9  .
## V10 .
```