

DALMKL Discussion

Kaiqiao Li

August 31, 2018

Discussion

This package allows us to call the C++ code for DALMKL algorithm efficiently. Here we allowed two loss function for binary classification (logistic loss and hinge loss). The objective function for DALMKL model is as follow:

$$\operatorname{argmin}_{\alpha, b} \sum_{i=1}^N l \left(y_i, \sum_{m=1}^M K_m \alpha_m + b \right) + C \sum_{m=1}^M \|\alpha_m\|_{K_m},$$

whereas the loss function

$$l(y, f)$$

can be written as

$$\log(1 + \exp(-yf))$$

for logistic loss and

$$(1 - yf)_+$$

for high loss. To optimize this problem we can use the following R function.

```
library(RMKL)
library(kernlab)
data(benchmark.data)
dd <- data[[1]]
trtrain <- sample(nrow(dd), 140)
k_train <- simplify2array(list(kernelMatrix(rbfddot(sigma = 0.05),
      dd[trtrain, 1:2]), kernelMatrix(vanilladot(), dd[trtrain, 1:2])))
k_test <- simplify2array(list(kernelMatrix(rbfddot(sigma = 0.05),
      dd[-trtrain, 1:2], dd[trtrain, 1:2]),
      kernelMatrix(vanilladot(), dd[-trtrain, 1:2],
      dd[trtrain, 1:2])))

y_train <- dd[trtrain, 3]
cri_out <- .01
cri_in <- .000001
maxiter_out <- maxiter_in <- 500
C <- .5
calpha <- 10
mod_hinge <- SpicySVM(k_train, y_train, C, cri_out, cri_in, maxiter_out, maxiter_in,
      calpha)
alpha_hinge <- mod_hinge$alpha
b_hinge <- mod_hinge$b
prediction_hinge <- predictspicy(alpha_hinge, b_hinge, k_test)

mod_logistic <- SpicyLogit(k_train, y_train, C, cri_out, cri_in, maxiter_out, maxiter_in,
      calpha)
alpha_logistic <- mod_logistic$alpha
b_logistic <- mod_logistic$b
prediction_logistic <- predictspicy(alpha_logistic, b_logistic, k_test)
```

Here we can call the C++ implementation for hinge or logistic loss as above. Here the `maxiter_out` and `maxiter_in` are the maximum iteration numbers we tried for the outer and inner cycle. The `cri_out` and `cri_in` are the convergence criterion values in outer and inner cycle. The criterion for inner cycle must be smaller than outer cycle like 0.0001 ratio and we suggested values as 0.01 and 0.000001. The `C` paramter is the tuning parameters, which must be positive, for the block one norm penalty. The `calpha` parameter is proximity parameters using for proximal minimization and also need to be positive. Here we suggest value 10 for this parameter.