561

Shuo Wang

1/27/2022

```
# calculate MSE for a particular model
cal_MSE = function(n, p, rho, signal){
  MSE_AIC = c()
  MSE_BIC = c()
  MSE_LOOCV = c()
  for(i in 1:1000){
    data = get_data(n, p, rho, signal)
    # AIC
    stage1 = ic.glmnet(data[,-1], data[,"y"], crit = "aic")
    penalty.factor = 1 / abs(coef(stage1)[-1] + 1/sqrt(nrow(data)))
    Adap_lasso_aic = ic.glmnet(data[,-1], data[,"y"], crit = "aic",
                               penalty.factor = penalty.factor)
    MSE_AIC = c(MSE_AIC, mean((Adap_lasso_aic$residuals)^2))
    # BIC
    stage1 = ic.glmnet(data[,-1], data[,"y"], crit = "bic")
    penalty.factor = 1 / abs(coef(stage1)[-1] + 1/sqrt(nrow(data)))
    Adap_lasso_bic = ic.glmnet(data[,-1], data[,"y"], crit = "bic",
                               penalty.factor = penalty.factor)
    MSE_BIC = c(MSE_BIC, mean((Adap_lasso_bic$residuals)^2))
    # LOOCV
    stage1 = cv.glmnet(as.matrix(data[,-1]), unlist(data[,"y"]), type.measure = "mse",
                       nfold = n, alpha = 1, grouped = FALSE)
    penalty.factor = 1 / abs(coef(stage1, s = stage1$lambda.min)[-1] + 1/sqrt(nrow(data)))
    Adap_lasso_loocv = cv.glmnet(as.matrix(data[,-1]), unlist(data[,"y"]),
                                 type.measure = "mse", nfold = n, alpha = 1,
                                 penalty.factor = penalty.factor,
                                 keep = TRUE, grouped = FALSE)
    pred = predict(Adap_lasso_loocv, newx = as.matrix(data[,-1]),
                   s = Adap_lasso_loocv$lambda.min)
    MSE_LOOCV = c(MSE_LOOCV, mean((unlist(data[,"y"])-pred)^2))
  }
  return(data.frame(AIC.MSE = mean(MSE_AIC), BIC.MSE = mean(MSE_BIC),
                    LOOCV.MSE = mean(MSE_LOOCV)))
}
```

```
set.seed(1)

n = 100
P = c(10, 25, 50)
RHO = c(0, 0.25, 0.5)
result = data.frame(p = NA, rho = NA, Estimator = "Adaptive_Lasso",
                    Signal = NA, AIC.MSE = NA, BIC.MSE = NA, LOOCV.MSE = NA)
# sparse
```

```
for(p in P){
  for(rho in RHO){
    temp = cal_MSE(n, p, rho, "sparse")
    result = rbind(result, cbind(data.frame(p = p, rho = rho, Estimator = "Adaptive_Lasso",
                                            Signal = "Sparse"), temp))
  }
}
# dense
for(p in P){
  for(rho in RHO){
    temp = cal_MSE(n, p, rho, "dense")
    result = rbind(result, cbind(data.frame(p = p, rho = rho, Estimator = "Adaptive_Lasso",
                                            Signal = "Dense"), temp))
  }
}
saveRDS(result, "result.Rda")
```

```
result
```

```
##     p  rho      Estimator Signal    AIC.MSE    BIC.MSE  LOOCV.MSE
## 1  10 0.00 Adaptive_Lasso Sparse 0.02792182 0.02884820 0.02820078
## 2  10 0.25 Adaptive_Lasso Sparse 0.03853452 0.03966114 0.03886868
## 3  10 0.50 Adaptive_Lasso Sparse 0.05157621 0.05301133 0.05203097
## 4  25 0.00 Adaptive_Lasso Sparse 0.04389139 0.04764674 0.04536777
## 5  25 0.25 Adaptive_Lasso Sparse 0.06483782 0.06975987 0.06708715
## 6  25 0.50 Adaptive_Lasso Sparse 0.09966520 0.10516304 0.10236509
## 7  50 0.00 Adaptive_Lasso Sparse 0.05379002 0.06545047 0.06102982
## 8  50 0.25 Adaptive_Lasso Sparse 0.08642461 0.10016835 0.09464900
## 9  50 0.50 Adaptive_Lasso Sparse 0.13837683 0.15845732 0.15045993
## 10 10 0.00 Adaptive_Lasso  Dense 0.08705605 0.08995909 0.08713886
## 11 10 0.25 Adaptive_Lasso  Dense 0.11753506 0.12139267 0.11771886
## 12 10 0.50 Adaptive_Lasso  Dense 0.16723401 0.17198740 0.16762749
## 13 25 0.00 Adaptive_Lasso  Dense 0.08034100 0.10415231 0.08394331
## 14 25 0.25 Adaptive_Lasso  Dense 0.11049729 0.13408863 0.11386973
## 15 25 0.50 Adaptive_Lasso  Dense 0.16082142 0.18235877 0.16386566
## 16 50 0.00 Adaptive_Lasso  Dense 0.06622898 0.11831192 0.08130785
## 17 50 0.25 Adaptive_Lasso  Dense 0.09466444 0.14971899 0.10851570
## 18 50 0.50 Adaptive_Lasso  Dense 0.14124502 0.19638209 0.15355975
```