wqDoug Woodward

CS613 HW 10

|  |  |
| --- | --- |
| # | Answer |
| 2 | 1. iii 2. iii 3. ii |
| 4 | 1. iii 2. ii 3. Iv 4. Iii 5. v |
| 5 | N = 2, p =2, x11 = x12, x21 = x22  Y1+y2 = 0  X11+x21 = 0  X12 + x22 = 0  Beta = 0   1. (y1 - Betahat1x11-betaHat2x12)^2 + (y2 - Betahat1x21-betaHat2x22)^2 + lambda(betahat1^2 + betahat2^2) 2. Derivative of above: Bhati = x1+y1+x2y2-Bhati(x1^2+x2^2)/lambda(betahat1^2 + betahat2^2) with the symmetry showing that Bhat1 = Bhat 2 as Bhati can be either Bhat1 or Bhat2 3. (y1 - Betahat1x11-betaHat2x12)^2 + (y2 - Betahat1x21-betaHat2x22)^2 + lambda(|betahat1^2| + |betahat2^2|) 4. From B, we can see that the value for Bhati can actually be anything, and therefore we can have many possible solutions. |
|  | library(ISLR)  library(glmnet)  set.seed(4221)  d = College  test = sample(1:dim(College)[1], dim(College)[1]/3)  train = -test  College.train = College[train,]  College.test = College[test,]  lm.fit = lm(Apps~., data=College.train)  lm.pred = predict(lm.fit, College.test)  mean((College.test[, "Apps"] - lm.pred)^2)  x = model.matrix(Apps~., data=College.train)  y = model.matrix(Apps~., data=College.test)  grid = 10 ^ seq(4, -2, length=100)  ridge = cv.glmnet(x, College.train[,"Apps"] ,alpha=0, lambda = grid)  ridge.lambda = ridge$lambda.min  ridge.pred = predict(ridge, newx=x, s=ridge.lambda)  mean((College.test[, "Apps"] - ridge.pred)^2)  lasso = cv.glmnet(x, College.train[,"Apps"] ,alpha=1, lambda = grid)  lasso.lambda = lasso$lambda.min  lasso.pred = predict(ridge, newx=x, s=lasso.lambda)  mean((College.test[, "Apps"] - lasso.pred)^2)  predict(lasso, s=lasso.lambda, type="coefficients")  [1] 1665477  [1] 32543554  [1] 32617312  19 x 1 sparse Matrix of class "dgCMatrix"  1  (Intercept) -7.874329e+02  (Intercept) .  PrivateYes -6.093633e+02  Accept 1.228820e+00  Enroll .  Top10perc 5.035133e+01  Top25perc -1.299642e+01  F.Undergrad 5.099811e-02  P.Undergrad 3.165198e-03  Outstate -5.589079e-02  Room.Board 2.265919e-01  Books 1.650927e-03  Personal 5.598210e-02  PhD -4.486963e+00  Terminal -5.866285e+00  S.F.Ratio 9.729152e+00  perc.alumni -3.827068e+00  Expend 7.956845e-02  Grad.Rate 6.453269e+00  The approaches are really about the same overall, just slight differences. |
| 11 | library(MASS) library(glmnet) set.seed(4221)  test = sample(1:dim(Boston)[1], dim(Boston)[1]/3) train = -test Boston.train = Boston[train,] Boston.test = Boston[test,]  x = model.matrix(crim ~ . - 1, data = Boston) y = Boston$crim cv.lasso = cv.glmnet(x, y, type.measure = "mse") plot(cv.lasso)  cv.ridge = cv.glmnet(x, y, type.measure = "mse", alpha=0) plot(cv.ridge) |