Data Project

Peiran Wang

December 4th, 2023

Contents

Download the dataset and load the data in R.

```
load('pathway_scores.RData')
load('pc_scores.rda')
load('survival.RData')

X1 = pathway.scores
X2 = pc_scores
```

Variable Selection

```
variable_selection1 = function(X, Y) {
  # TODO: fancier variable selection logic
  return(abs(cor(X, Y)) >= 0.35)
  # return a vector of booleans indicating which covariates are selected.
}

variable_selection2 = function(X, Y) {
  # TODO: fancier variable selection logic
  fit = glmnet(X,Y,alpha=0.8,lambda = 5)
  selected = coef(fit)!=0
  return(selected[-1])
  # return a vector of booleans indicating which covariates are selected.
}
```

Prediction model using one set of covariates

```
prediction_model1 = function(X, Y, X_new) {
  # TODO: fancier prediction model
  cv_model <- cv.glmnet(X, Y, alpha = 0.2)
  best_lambda <- cv_model$lambda.min
  best_model <- glmnet(X, Y, alpha = 0.2, lambda = best_lambda)
  return(predict(best_model, newx = X_new))
}
```

```
prediction_model2 = function(X, Y, X_new) {
  # TODO: fancier prediction model
  model = lm(Y ~ 1, data = data.frame(X))
  full_model = lm(Y ~ ., data = data.frame(X))
  step_model = stepAIC(model, scope = list(lower = model, upper = full_model), direction = "forward",tra
  return(predict(step_model, newx=X_new))
  }
```

Combined Model

```
combined_model = function(X1, X2, Y, X1_new, X2_new) {
  # TODO: fancier model that combines two sets of covariates
  X0 = X1[,1]*X2
  for (i in 2:dim(X1)[2]) {
    temp = X1[,i]*X2
    X0 = cbind(X0, temp)
  }
  fit = cv.glmnet(X0, Y, nfolds = 10, family = "gaussian", alpha = 0.2)
  best_lambda = fit$lambda.min
  best_fit = glmnet(X0, Y, family = "gaussian", alpha = 0.2, lambda = best_lambda)
  X0_new = X1_new[,1]*X2_new
  for (i in 2:dim(X1_new)[2]) {
    temp = X1_new[,i]*X2_new
    X0_new = cbind(X0_new, temp)
  }
  return(predict(fit, newx = X0_new))
}
```

Estimate prediction error of the two-stage procedure

For example, using leave-one-out cross validation.

```
Z = Y
predictions = sapply(1:length(Y), function(i) {
  # leave one out
  Xi1 = X1[-i, ]
  Xi2 = X2[-i, ]
  Zi = Z[-i]

  # variable selection
  s1 = variable_selection1(Xi1, Zi)
  s2 = variable_selection2(Xi2, Zi)
  Xi1_s = Xi1[, s1]
  Xi2_s = Xi2[, s2]

  # left out data for testing
  X1_new = X1[i, s1, drop = FALSE]
  X2_new = X2[i, s2, drop = FALSE]

  # evaluation of prediction model using first dataset
  pred1 = prediction_model1(Xi1_s, Zi, X1_new)

  # evaluation of prediction model using second dataset
  pred2 = prediction_model2(Xi2_s, Zi, X2_new)

  # evaluation of prediction model using combined dataset
  pred_combined = combined_model(Xi1_s, Xi2_s, Zi, X1_new, X2_new)

  return(rbind(
    pred1,
    pred2,
    pred_combined
  ))
})
```

```
# Evaluation

# Model using first dataset
sqrt(mean((predictions[1, ] - Y)^2))
```

```
[1] 34.20361
```

```
# Model using second dataset
sqrt(mean((predictions[2, ] - Y)^2))
```

```
[1] 33.87671
```

```
# Model using combined dataset
sqrt(mean((predictions[3, ] - Y)^2))
```

```
[1] 31.44185
```