

STA 35C – Homework 5

Submission due: Thu, November 5 at 11:59 PM PT

Instructor: Dogyoon Song

Instructions: Upload a **single PDF file** to Gradescope via Canvas (“Homework 5” under “Assignments”). Name the file using the prefix of your UC Davis email ID and the homework number (e.g., `dgsong_hw5.pdf`). Include “STA 35C,” your name, and the last four digits of your student ID on the front page. For coding problems, include your R code, relevant output, and any requested plots/tables in the PDF. No late submissions will be accepted. Any submission received after the deadline will receive 0 points. For full information about submission requirements and the late submission policy, see the syllabus.

Problem 1: Cross-validation for classification (20 points total + 5 bonus points).

We will explore cross-validation for a *classification* (logistic regression) setting on a simulated dataset. Use the code below to generate X (numeric) and Y (binary):

```
set.seed(2026)
n <- 100
X <- rnorm(n)
log_odds <- -2 - 3*X
probs <- 1 / (1 + exp(-log_odds))
Y <- rbinom(n, 1, probs)
df <- data.frame(X, Y)
```

Keep this dataset fixed throughout parts (a)–(d). Fit an intercept in every logistic model and predict class 1 when its fitted probability is at least 0.5. Use `predict(..., type = "response")` to obtain probabilities. For LOOCV and k -fold CV, divide the total number of held-out misclassifications by n .

- (a) **(5 points)** Plot Y against X , with X on the horizontal axis and colors identifying the two classes.
- (b) **(5 points)** Set `set.seed(35)` once before making 100 random 50/50 training/validation splits. For each split, fit `glm(Y ~ X, family = binomial, ...)` using only the training observations and compute the validation classification error. Plot a histogram of the 100 errors and report their mean and standard deviation.
- (c) **(5 points)** Compute the LOOCV classification error, refitting the model without each observation before predicting it. Compare it with the mean validation error from part (b), noting the different training sample sizes.
- (d) **(5 points)** Perform 5-fold and 10-fold CV, refitting the model on the training observations in each fold. For each choice of k , use `set.seed(35)` followed by `fold <- sample(rep(1:k, length.out = n))`.

Report both errors and compare them with LOOCV in 2–3 sentences. Is a monotone ordering of these errors guaranteed?

- (e*) **(5 bonus points)** Generate a new dataset of $n = 100$ independent observations using `set.seed(2027)`, $X \sim \mathcal{N}(0, 1)$, and

$$\log \frac{\Pr(Y = 1 \mid X = x)}{\Pr(Y = 0 \mid X = x)} = -2 - 3x + x^2.$$

Compare logistic models $Y \sim X$ and $Y \sim X + I(X^2)$ using only 5-fold CV. Use the same folds for both models, generated as in part (d) with $k = 5$. Report their classification errors at threshold 0.5 and briefly interpret the difference.

Problem 2: Subset selection (30 points total + 5 bonus points).

- (a) **(10 points)** In 4–6 sentences, explain the purpose and main steps of best subset selection. Compare its search path and computational cost with forward stepwise selection.
- (b) **(20 points total)** Load `Credit` from `ISLR2`. Use `Balance` as the response and encode the remaining variables with:

```
data("Credit", package = "ISLR2")
x <- model.matrix(Balance ~ ., data = Credit)[, -1]
y <- Credit$Balance
p <- ncol(x)
```

Here subset size k counts columns of $\mathbf{x}$, including individual dummy columns. Always include an intercept, which does not count toward k . You may use `leaps::regsubsets(x, y, nvmax = p, method = "exhaustive")` and its `summary` and `coef` methods. Handle $k = 0$ separately as the intercept-only model.

- (i) **(5 points)** For each $k = 0, \dots, p$, find the subset with the lowest training RSS. Report the selected columns, RSS, and adjusted R^2 in one table, and plot the minimum RSS against k . You do not need to report or plot every candidate subset.
- (ii) **(5 points)** Select the model with the largest adjusted R^2 . Explain why minimizing training RSS alone cannot favor a smaller model over the full model.
- (iii) **(10 points)** Select subset size by 5-fold CV using `set.seed(35)` and `fold <- sample(rep(1:5, length.out = nrow(x)))`. Repeat the best-subset search *within each training fold* before predicting its held-out observations. For $k = 0$, predict the training-fold mean. Report CV MSE for each k , computed from all held-out squared errors. Choose the minimizing size and report its best subset fitted on the full dataset. Compare it with the adjusted- R^2 choice in 1–2 sentences.

For $k \geq 1$, the following helper predicts from a `regsubsets` fit:

```
predict_subset <- function(fit, newx, k) {
  b <- coef(fit, id = k)
  z <- cbind("(Intercept)" = 1, newx)
  drop(z[, names(b), drop = FALSE] %*% b)
}
```

- (c*) **(5 bonus points)** Run forward stepwise selection on the same $\mathbf{x}$ and $\mathbf{y}$. Report the selected columns at each size and identify any sizes where its subset or RSS differs from best subset selection. No additional CV is required. (*Hint:* use `method = "forward"`.)

Problem 3: Regularization (30 points in total).

- (a) **(5 points)** In 3–5 sentences, explain regularization and compare the penalties used by ridge regression and the lasso. Why can shrinkage improve prediction even when it increases training RSS?
- (b) **(10 points)** Consider a single coefficient b with unpenalized loss $(3 - b)^2$. Ridge regression minimizes $(3 - b)^2 + \lambda b^2$, while the lasso minimizes $(3 - b)^2 + \lambda |b|$. For $\lambda \geq 0$, you may use

$$\hat{b}_{\text{ridge}} = \frac{3}{1 + \lambda}, \quad \hat{b}_{\text{lasso}} = \max\{3 - \lambda/2, 0\}.$$

Report both estimates for $\lambda = 0, 2, 6$ in one table. Explain how the table illustrates shrinkage and the difference between small and exactly zero coefficients.

- (c) **(15 points)** This is a guided version of [JWHT21, Chapter 6, Exercise 9(a)–(d), (g)]. Predict `Apps` from the other variables in `ISLR2::College`. Create one training/test split:

```
data("College", package = "ISLR2")
x <- model.matrix(Apps ~ ., data = College)[, -1]
y <- College$Apps
set.seed(2026)
train_id <- sample.int(nrow(x), floor(2*nrow(x)/3))
set.seed(35)
fold_id <- sample(rep(1:5, length.out = length(train_id)))
```

Fit least squares, ridge regression, and the lasso on the training observations. For ridge and lasso, choose `lambda.min` by 5-fold CV *within the training set*, using the same `fold_id` for both. Use an unpenalized intercept and `standardize = TRUE` in `glmnet`. Do not standardize using the test data. Report a three-row table of test MSEs, the selected λ for each penalized model, and the number of nonzero lasso slopes. Compare the test errors in 2–3 sentences. Do not use the test set to tune λ . (*Hints:* `glmnet::cv.glmnet` uses `alpha = 0` for ridge and `alpha = 1` for lasso. Pass `foldid = fold_id` and use `s = "lambda.min"` when predicting.)

Problem 4: Multiple hypothesis testing (20 points in total).

- (a) **(5 points)** In at most 3 sentences, explain a Type I error and what it means for a test to have level α . Distinguish a probability bounded by α from one equal to α .
- (b) **(5 points)** Suppose all m null hypotheses are true, the tests are independent, and each is rejected with probability exactly α . Let V be the total number of false rejections. Give its distribution and PMF, and calculate $\mathbb{E}[V]$ and $\text{Var}(V)$.
- (c) **(5 points)** Under the assumptions in part (b), derive $\Pr(V \geq 1)$, the family-wise error rate (FWER). Evaluate it for $m = 20$ and $\alpha = 0.05$. Compare this with the error probability for one test.
- (d) **(5 points)** Now consider hypotheses $H_1, \dots, H_4$ with observed p -values 0.010, 0.030, 0.031, and 0.200, respectively. Do not assume that all four nulls are true. Assume independent tests with valid null p -values. Identify the rejected hypotheses under Bonferroni at FWER level 0.05 and Benjamini–Hochberg at FDR level 0.05. Show the thresholds used. Briefly distinguish the two error criteria being controlled.

References

- [JWHT21] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*, volume 112 of *Springer Texts in Statistics*. Springer, New York, NY, 2nd edition, 2021.