Homework 2

Mari Sanders

2025-02-24

Problem 1

Linear Regression

- Uses continuous outcome variables.
- Assumes a linear relationship between the outcome and predictors.
- Coefficients represent the change in the dependent variable for a one-unit change in an independent variable.
- Errors are assumed to be normally distributed.
- Output is continuous and can take any real number.
- Uses RMSE or MSE to evaluate model fit

Logistic Regression

- Uses binary or categorical outcome variables.
- Models non-linear relationships using the logit function and probabilities.
- Coefficients are expressed as log odds, meaning a unit change in an independent variable affects the log odds of the outcome.
- Errors follow a binomial distribution.
- Output is a probability between 0 and 1.
- Uses deviance and score to evaluate model fit

Problem 2

$Odds = \frac{pi}{1-pi}$, which is the probability of an event occurring over the probability of the event not occurring. If you do e^{β} to the coefficients in logistic regression, you will get the odds ratio. This is interpretable because it is in terms of the original equation and also easy to understand.

$log(odds) = log(pi/1 - pi)$, which is the probability of odds ratio given that the values are in terms of log, which is less interpretable. The coefficients β in logistic regression are in terms of log odds, such that a change in log-odds for a one-unit increase in the outcome.

Problem 3

L1 Regularization: Adds the absolute value of the sum of coefficients as a penalty term to the RSS. Lasso makes some of the coefficients go to zero.

L2 Regularization: Adds the squared sum of coefficients as the penalty term to the RSS. Ridge/L2 regularization shrinks coefficients toward each other so that they can borrow strength from each other, but does not shrink any coefficients to zero.

Problem 4

$$Y_i \sim Ber(\pi_i)$$

$$f(y_i, \pi_i) = \binom{m_i}{y_i} \pi_i^{y_i} (1-\pi_i)^{1-y_i}$$

$$l(y_i, \pi_i) = \log\left(\binom{m_i}{y_i} \pi_i^{y_i} (1-\pi_i)^{1-y_i}\right)$$

$$= \log\left(\binom{m_i}{y_i}\right) + y_i \log(\pi_i) + (1-y_i)\log(1-\pi_i)$$

$$l(\beta) = \sum_{i=1}^{n} \left[y_i \log\left(\frac{\pi_i}{1-\pi_i}\right) + \log(1-\pi_i) + \log\binom{m_i}{y_i} \right]$$

$$x_i^T \beta = g(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right)$$

$$l(\beta) = \sum_{i=1}^{n} y_i x_i^T \beta - \log\left(1 + e^{x_i^T \beta}\right)$$

$$S(\beta) = \frac{\partial}{\partial \beta}\left(l(\beta)\right) = \sum_{i=1}^{n} y_i \underbrace{\frac{\partial}{\partial \beta}(x_i^T \beta)}_{x_i} - \frac{\partial}{\partial \beta} \log\left(1 + e^{x_i^T \beta}\right)$$

$$= \sum_{i=1}^{n} y_i x_i - \frac{1}{1+e^{x_i^T \beta}} \cdot e^{x_i^T \beta} \cdot x_i$$

$$S(\beta) = \sum_{i=1}^{n} y_i x_i - \frac{x_i e^{x_i^T \beta}}{1+e^{x_i^T \beta}} = \sum_{i=1}^{n} (y_i - \pi_i) x_i$$

$$I(\beta) = -E\left(\frac{\partial^2 l(\beta)}{\partial \beta^2}\right)$$

$$= \frac{\partial}{\partial \beta} \sum_{i=1}^{n} y_i x_i - \frac{x_i e^{x_i^T \beta}}{1+e^{x_i^T \beta}}$$

$$I(\beta) = -\sum_{i=1}^{n} x_i \frac{e^{x_i \beta} x_i}{(1+e^{\beta x_i})^2} = \sum_{i=1}^{n} x_i \left(\frac{1}{(1+e^{x_i \beta})^2}\right) \cdot e^{x_i^T \beta} \cdot x_i$$

$$I(\beta) = \Sigma x_i x_i^T \underbrace{\frac{e^{x_i^T \beta}}{1+e^{x_i^T \beta}}}_{\pi} \cdot \underbrace{\frac{1}{1+e^{x_i^T \beta}}}_{(1-\pi)}$$

Problem 5

Logit

```
dose <- c(0,1,2,3,4)
dying <- c(2, 8, 15, 23, 27)
tested <- rep(30,5)
data <- data.frame(dose, tested, dying)

resp <- cbind(died = data$dying, alive = tested - data$dying)
pred <- data$dose

fitlogit <- glm(resp~pred,family=binomial(link='logit'),data= data)
summary(fitlogit)
```

```
##
## Call:
## glm(formula = resp ~ pred, family = binomial(link = "logit"),
##     data = data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.3238     0.4179  -5.561 2.69e-08 ***
## pred          1.1619     0.1814   6.405 1.51e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 64.76327  on 4  degrees of freedom
## Residual deviance:  0.37875  on 3  degrees of freedom
## AIC: 20.854
##
## Number of Fisher Scoring iterations: 4
```

```
beta_1 <- exp(fitlogit$coefficient[2])
confint(fitlogit)
```

```
## Waiting for profiling to be done...
```

```
##                  2.5 %    97.5 %
## (Intercept) -3.2060617 -1.557314
## pred         0.8301789  1.546129
```

```
confint_logit <- exp(confint(fitlogit))
```

```
## Waiting for profiling to be done...
```

```r
devfitlogit <- sum(residuals(fitlogit,type='deviance')^2)
predictionlogit <- predict(fitlogit, data.frame(pred = 0.01), se.fit=TRUE,type='response')

cat("Estimate:", exp(fitlogit$coefficients[2]), "\n",
    "CI:", c(confint_logit[2,1], confint_logit[2,2]), "\n",
    "Deviance:", devfitlogit, "\n",
    "P(dying|x=0.01):", predictionlogit$fit, "\n")
```

```
## Estimate: 3.195984
##  CI: 2.293729 4.693269
##  Deviance: 0.3787483
##  P(dying|x=0.01): 0.09011997
```

Probit

```r
fitprobit <- glm(resp~pred,family=binomial(link='probit'),data= data)
summary(fitprobit)
```

```
##
## Call:
## glm(formula = resp ~ pred, family = binomial(link = "probit"),
##     data = data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.37709    0.22781  -6.045 1.49e-09 ***
## pred         0.68638    0.09677   7.093 1.31e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 64.76327  on 4  degrees of freedom
## Residual deviance:  0.31367  on 3  degrees of freedom
## AIC: 20.789
##
## Number of Fisher Scoring iterations: 4
```

```r
confint(fitprobit)
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %     97.5 %
## (Intercept) -1.8436290 -0.9442144
## pred         0.5033779  0.8840139
```

```r
confint_probit <- exp(confint(fitprobit))
```

```
## Waiting for profiling to be done...
```

```
devprobit <- sum(residuals(fitprobit,type='deviance')^2)

predictionprobit <- predict(fitprobit, data.frame(pred = 0.01), se.fit = TRUE,type = 'response')
cat("Estimate:", exp(fitprobit$coefficients[2]), "\n",
    "CI:", c(confint_probit[2,1], confint_probit[2,2]), "\n",
    "Deviance:", devprobit, "\n",
    "P(dying|x=0.01):", predictionprobit$fit, "\n")
```

```
## Estimate: 1.986512
##  CI: 1.6543 2.420596
##  Deviance: 0.3136684
##  P(dying|x=0.01): 0.0853078
```

Cloglog

```
fitcloglog <- glm(resp~pred, family = binomial(link = "cloglog"), data = data)
confint_cloglog <- exp(confint(fitcloglog))
```

```
## Waiting for profiling to be done...
```

```
devcloglog <- sum(residuals(fitcloglog,type = 'deviance')^2)

predictcloglog <- predict(fitcloglog, data.frame(pred =0.01), se.fit=TRUE,type='response')

cat("Estimate:", exp(fitcloglog$coefficients[2]), "\n",
    "CI:", c(confint_cloglog[2,1], confint_cloglog[2,2]), "\n",
    "Deviance:", devcloglog, "\n",
    "P(dying|x=0.01):", predictcloglog$fit, "\n")
```

```
## Estimate: 2.110277
##  CI: 1.726303 2.624218
##  Deviance: 2.230479
##  P(dying|x=0.01): 0.1281601
```

b)

```
LD50 <- function(fit) {
  beta0 = fit$coefficients[1]
  beta1 = fit$coefficients[2]

  betacov = vcov(fit)

  x0 = -beta0/beta1


  varx0 = betacov[1,1]/(beta1^2)+betacov[2,2]*(beta0^2)/(beta1^4)-2*betacov[1,2]*beta0/(beta1^3)

  cat("Estimate:", exp(x0), "\n",
```

```
    "CI:", exp((x0+c(qnorm(0.05),-qnorm(0.05))*sqrt(varx0))), "\n")

}
```

Logit Model: Probit Model: Cloglog:

Problem 6

```
amount <- seq(from = 10, to = 90, by = 5)
offers <- c(4, 6, 10, 12, 39, 36, 22, 14, 10, 12, 8, 9, 3, 1, 5, 2, 1)
enrolls <- c(0, 2, 4, 2, 12, 14, 10, 7, 5, 5, 3, 5, 2, 0, 4, 2, 1)
declined <- offers - enrolls
data <- data.frame(amount, offers, enrolls, declined)

mphfit <- glm(cbind(enrolls, declined) ~ amount,
              family=binomial(link='logit'),data= data)
summary(mphfit)
```

```
##
## Call:
## glm(formula = cbind(enrolls, declined) ~ amount, family = binomial(link = "logit"),
##     data = data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.64764    0.42144  -3.910 9.25e-05 ***
## amount       0.03095    0.00968   3.197  0.00139 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21.617  on 16  degrees of freedom
## Residual deviance: 10.613  on 15  degrees of freedom
## AIC: 51.078
##
## Number of Fisher Scoring iterations: 4
```

a)

```
devmph <- sum(residuals(mphfit,type='deviance')^2)

sum(residuals(mphfit,type='pearson')^2)
```

```
## [1] 8.814299
```

```r
pval=1-pchisq(devmph,17-2)


hl <- hoslem.test(mphfit$y, fitted(mphfit), g=10)
hl
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mphfit$y, fitted(mphfit)
## X-squared = 1.6111, df = 8, p-value = 0.9907
```

The p value is 0.7795345, which means that we fail to reject the null hypothesis that there is no relationship between the predictors and the response. This means that the model fits the data ok. ### b)

```r
confint(mphfit)
```

```
## Waiting for profiling to be done...

##                   2.5 %      97.5 %
## (Intercept) -2.50117202 -0.84287573
## amount       0.01245819  0.05060401
```

```r
exp(confint(mphfit))
```

```
## Waiting for profiling to be done...

##                  2.5 %    97.5 %
## (Intercept) 0.08198885 0.4304708
## amount      1.01253611 1.0519063
```

For a unit increase in amount, the odds of enrollment increases by 1.031434.

We are 95% confident that the the odds ratio is between 1.01253611 and 1.0519063 meaning that a unit increase in the amount of money increases the odds of enrollment by 2%-5%. Since the confidence interval does not include 1, we can say that the effect is statistically significant.

c)

```r
beta0 <- mphfit$coefficients[1]
beta1 <- mphfit$coefficients[2]
betacov <- vcov(mphfit)
log_odds <- log(0.4 / (1 - 0.4))
scholarship_needed <- (log_odds - beta0) / beta1

var_x <- betacov[1,1] / (beta1^2) +
        betacov[2,2] * (beta0 - log_odds)^2 / (beta1^4) -
        2 * betacov[1,2] * (beta0 - log_odds) / (beta1^3)
```

```r
se_x <- sqrt(var_x)

ci_lower <- scholarship_needed - 1.96 * se_x
ci_upper <- scholarship_needed + 1.96 * se_x

cat("Estimate:", scholarship_needed, "\n",
    "Standard Error:", se_x, "\n",
    "CI Lower:", ci_lower, "\n",
    "CI Upper:", ci_upper, "\n")
```

```
## Estimate: 40.13429
##  Standard Error: 4.873174
##  CI Lower: 30.58286
##  CI Upper: 49.68571
```