Model selection and multimodel inference

QERM 514 - Homework 5

1 May 2020

R Markdown file

You can find the R Markdown file used to create this answer key here.

Background

This week's home work will require you to use all of the information you have learned so far in class. Your task is to analyze some data on the concentration of nitrogen in the soil at 41 locations on the island of Maui in the Hawaiian Archipelago. Along with the nitrogen measurements, there are 4 possible predictor variables that may help to explain the variation in soil nitrogen concentration. The accompanying data file `soil_nitrogen.csv` has the following 5 columns of data:

- `nitrogen`: concentration of soil nitrogen (mg nitrogen kg^{-1} soil)
- `temp`: average air temperature (°C)
- `precip`: average precipitation (cm)
- `slope`: slope of the hillside (degrees)
- `aspect`: aspect of the hillside (N, S)

As you work through the following problems, be sure to show all of the code necessary to produce your answers.

Problems

a) Begin by building a global model that contains all four of the predictors plus an intercept. Show the resulting ANOVA table, and report the multiple and adjusted R^3 values. Also report the estimate of the residual variance $\hat{\sigma}^2$.

```
## get data
soil_N <- read.csv("soil_nitrogen.csv")

## sample size (for later)
nn <- nrow(soil_N)

## fit full model
mod_full <- lm(nitrogen ~ temp + precip + slope + aspect, data = soil_N)
summary(mod_full)
```

```
##
## Call:
## lm(formula = nitrogen ~ temp + precip + slope + aspect, data = soil_N)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0914 -1.0880 -0.2175  1.0337  3.7394
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 47.63347    2.47280  19.263  < 2e-16 ***
## temp         0.03736    0.09292   0.402   0.6900
## precip       0.11302    0.01054  10.726 9.23e-13 ***
## slope       -0.78304    0.06029 -12.987 3.76e-15 ***
## aspectS      1.25273    0.52804   2.372   0.0231 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.664 on 36 degrees of freedom
## Multiple R-squared:  0.9075, Adjusted R-squared:  0.8972
## F-statistic: 88.28 on 4 and 36 DF,  p-value: < 2.2e-16
```

The multiple $R^2 = 0.907$ and the adjusted $R^2 = 0.897$. The estimated residual variance ($\hat{\sigma}^2$) is 2.77. (Note that the `Residual standard error: 1.664` listed in the table is $\hat{\sigma}$ rather than $\hat{\sigma}^2$.

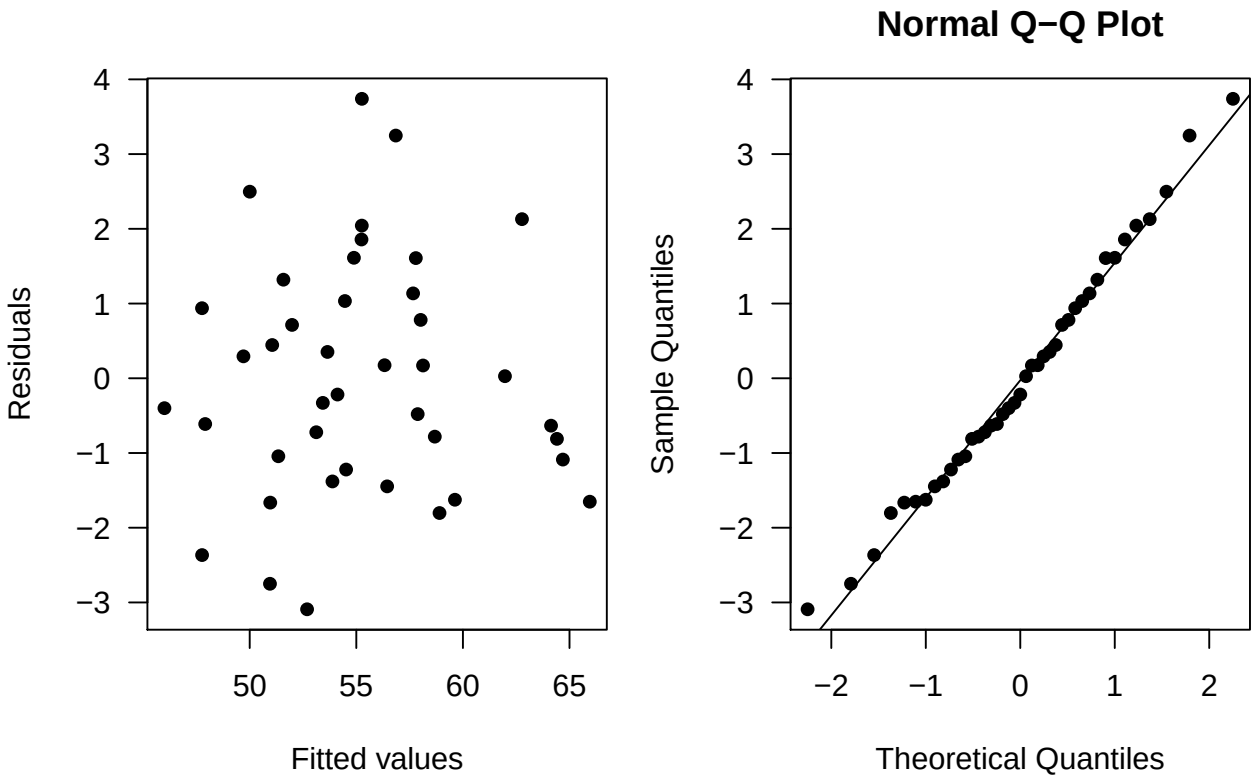
---

b) Check the residuals from your full model for possible violations of the assumption that the $e_i \sim \mathrm{N}(0, \sigma^2)$.

There are a number of checks we can make, but the most obvious are a plot of the residuals against the fitted values ($\hat{y}$) and a Q-Q plot. As there is no indication that the data are from a time series, or that they were collected at locations very close to one another, there is no reason (or way) to check for autocorrelation.

```
## get residuals
ee <- residuals(mod_full)

par(mfrow = c(1, 2),
    mai = c(0.9, 0.9, 0.6, 0.1),
    omi = c(0, 0, 0, 0))
## residuals vs fitted
plot(fitted(mod_full), ee, las = 1, pch = 16,
     ylab = "Residuals", xlab = "Fitted values")
## Q-Q plot
qqnorm(ee, las = 1, pch = 16)
qqline(ee)
```



Both of these plots look good. I do not see any abnormal patterns in the residuals as a function of the fitted values (ie, they look homoscedastic), nor do I see any egregious deviations from the assumption of normality.

---

c) Does this seem like a reasonable model for these data? Why or why not?

Yes, this model seems to be okay. The overall fit looks rather promising and there is no indication that our assumptions about the model errors have been violated.

---

d) Now fit various models using all possible combinations of the 4 predictors, including an intercept-only model (ie, there should be a total of 16 models). Compute the AIC, AICc, and BIC for each of your models and compare the relative rankings of the different models.

```
## data frame specifying predictors to include
df <- as.data.frame(matrix(c(FALSE, TRUE), 2, 4))
## add col names
cov_names <- colnames(df) <- colnames(soil_N)[-1]

## create set of all possible combinations
model_set <- as.matrix(expand.grid(df))

## number of models in our set
n_mods <- nrow(model_set)
```

```r
## empty matrix for storing results
mod_res <- matrix(NA, n_mods, 3)
colnames(mod_res) <- c("AIC", "AICc", "BIC")

## fit models & store IC
for(i in 1:n_mods) {
  ## create model formula
  if(i == 1) {
    fmla <- "nitrogen ~ 1"
  } else {
    fmla <- paste("nitrogen ~", paste(cov_names[model_set[i,]], collapse = " + "))
  }
  ## fit model
  mod_fit <- lm(as.formula(fmla), data = soil_N)
  ## get AIC
  mod_res[i,"AIC"] <- AIC(mod_fit)
  ## get AICc
  ## number of parameters in the model
  k <- 1 + length(coef(mod_fit))
  ## calculate penalty term
  pterm <- (2 * k * (k + 1)) / (nn - k - 1)
  ## get AICc
  mod_res[i,"AICc"] <- AIC(mod_fit) + pterm
  ## get BIC
  mod_res[i,"BIC"] <- BIC(mod_fit)
}

## find top-ranked model(s) (ie, those with lowest IC)
(best_mod <- apply(mod_res, 2, which.min))
```

```
##  AIC AICc  BIC
##   15   15   15
```

```r
## scale IC to delta-values & round them
min_IC <- apply(mod_res, 2, min)
(delta_IC <- round(t(t(mod_res) - min_IC), 1))
```

```
##        AIC AICc  BIC
##  [1,] 91.4 90.0 86.3
##  [2,] 93.4 92.3 90.0
##  [3,] 69.5 68.4 66.0
##  [4,] 71.3 70.7 69.6
##  [5,] 58.1 57.1 54.7
##  [6,] 59.4 58.8 57.7
##  [7,]  3.9  3.3  2.1
##  [8,]  5.8  5.8  5.8
##  [9,] 91.1 90.1 87.7
## [10,] 93.1 92.5 91.4
## [11,] 69.2 68.6 67.5
```

```
## [12,] 71.1 71.1 71.1
## [13,] 57.5 56.9 55.8
## [14,] 58.6 58.6 58.6
## [15,]  0.0  0.0  0.0
## [16,]  1.8  2.6  3.5
```

All three of the information criteria point to model 15 as the "best" of the bunch, which has 3 predictors: temp, precip, slope. However, AIC also indicates that model 16 containing all 4 predictors is also pretty close (ie, it's within 2 units).

---

e) Conduct a leave-one-out cross-validation for all of the models in part (d), using the root mean squared prediction error (RMSPE) as your scale-dependent measure of fit. Report your results alongside your results from part (d). Do all of the methods agree on which of these models is the best?

```
## empty vector for predictions
loo_res <- rep(NA, nn)
## empty vector for MSPE
rmspe <- rep(NA, n_mods)

## loop over all possible model combinations
for(i in 1:n_mods) {
  ## create model formula
  if(i == 1) {
    fmla <- "nitrogen ~ 1"
  } else {
    fmla <- paste("nitrogen ~", paste(cov_names[model_set[i,]], collapse = " + "))
  }
  ## loop over number of observations
  for(j in 1:nn) {
    ## drop one observation and fit the model
    fm <- lm(as.formula(fmla), soil_N[-j,])
    ## predict the missing value
    loo_res[j] <- predict(fm, newdata = data.frame(soil_N[j,]))
  }
  ## calculate RMSPE for the predictions
  rmspe[i] <- sqrt(sum((soil_N$nitrogen - loo_res)^2) / nn)
}

## add RMSPE values to above table for IC
(tbl_results <- cbind(delta_IC, RMSPE = round(rmspe, 2)))
```

```
##        AIC AICc  BIC RMSPE
##  [1,] 91.4 90.0 86.3  5.25
##  [2,] 93.4 92.3 90.0  5.36
##  [3,] 69.5 68.4 66.0  4.02
##  [4,] 71.3 70.7 69.6  4.11
##  [5,] 58.1 57.1 54.7  3.49
```

```
##  [6,] 59.4 58.8 57.7  3.58
##  [7,]  3.9  3.3  2.1  1.80
##  [8,]  5.8  5.8  5.8  1.86
##  [9,] 91.1 90.1 87.7  5.24
## [10,] 93.1 92.5 91.4  5.36
## [11,] 69.2 68.6 67.5  4.01
## [12,] 71.1 71.1 71.1  4.12
## [13,] 57.5 56.9 55.8  3.48
## [14,] 58.6 58.6 58.6  3.55
## [15,]  0.0  0.0  0.0  1.72
## [16,]  1.8  2.6  3.5  1.78
```

The estimated RMSPE values generally agree with the IC results, although there is not as clear a separation among the models.

---

f) Given some uncertainty that one of these models is the true data-generating model, compute the weights of evidence for each of the models in your set. Which model has the greatest support from the data? What are the odds against the intercept-only model compared to the best model?

The weights of evidence should be based upon the AIC values obtained in part (d).

```
## numerator
num <- exp(-0.5 * tbl_results[,"AIC"])
## denominator
dem <- sum(num)
## Akaike weights
wts <- num / dem
## evidence ratios
ER <- exp(0.5 * tbl_results[,"AIC"])
## data frame with our results
data.frame(model = seq(n_mods),
           weights = round(wts, 3),
           ER = floor(ER))
```

```
##    model weights           ER
## 1      1   0.000 7.034898e+19
## 2      2   0.000 1.912284e+20
## 3      3   0.000 1.235189e+15
## 4      4   0.000 3.038074e+15
## 5      5   0.000 4.132898e+12
## 6      6   0.000 7.916735e+12
## 7      7   0.089 7.000000e+00
## 8      8   0.034 1.800000e+01
## 9      9   0.000 6.054993e+19
## 10    10   0.000 1.645918e+20
## 11    11   0.000 1.063137e+15
## 12    12   0.000 2.748963e+15
## 13    13   0.000 3.061726e+12
```

```
## 14    14   0.000 5.306746e+12
## 15    15   0.623 1.000000e+00
## 16    16   0.253 2.000000e+00
```

The results also indicate the model 15 is the "best" of our model set because it has ~60% of the total weights. Model 16, which includes all of the predictors has ~25% of the total weights. The evidence against the intercept-only model being better than model 15 is ~7e19:1, which are overwhelming odds against it.

---

g) Calculate the model-averaged parameters across all models in your set. Use these parameters to predict what the soil nitrogen concentration would be on the nearby island of Moloka'i if the average precipitation was 150 cm, the average temperature was 22 °C, and the hillside faced south with a slope of 11 degrees.

The trick here is to recognize that `aspect` is coded as a `0` for `N` and a `1` for `S`.

```
## empty matrix for storing coefficients
## we'll fill it with 0's and replace them with the param estimates
mod_coef <- matrix(0, n_mods, 1 + ncol(df))
colnames(mod_coef) <- c("Intercept", colnames(df))

## fit models & store AIC & BIC
for(i in 1:n_mods) {
  if(i == 1) {
    fmla <- "nitrogen ~ 1"
  } else {
    fmla <- paste("nitrogen ~", paste(cov_names[model_set[i,]], collapse = " + "))
  }
  mod_fit <- lm(as.formula(fmla), data = soil_N)
  mod_coef[i, c(TRUE, model_set[i,])] <- coef(mod_fit)
}

## calculate weighted parameters
wtd_coef <- mod_coef * wts
(avg_coef <- colSums(wtd_coef))
```

```
##  Intercept       temp     precip      slope     aspect
## 48.1928509  0.0103904  0.1136534 -0.7829092  1.0923140
```

```
## compute model-averaged prediction
X <- matrix(c(Intercept = 1, temp = 22, precip = 150, slope = 11, aspect = 1), nrow = 1)
(y_hat_avg <- X %*% as.matrix(avg_coef, ncol = 1))
```

```
##          [,1]
## [1,] 57.94976
```

---

h) Compare your prediction from part (g) to a prediction from the model identified as the best in part (e), using the same inputs. How much do they differ from one another?

Because our top model does not have a term for `temp`, we can ignore that value in our prediction from the top model (or alternatively you could just set $\beta_{\text{temp}} = 0$.

```
## get coefficients from best model w/o `temp`
beta_best <- matrix(coef(lm(nitrogen ~ precip + slope + aspect, data = soil_N)), ncol = 1)
## compute prediction from best model w/o `temp`
(y_hat_best <- X[-2] %*% beta_best)
```

```
##          [,1]
## [1,] 57.99948
```