Lecture 21: Model Selection

1 Choosing Models

Sometimes we need to choose from two or more possible models. For example, we might want to consider these two models:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

and

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \beta_4 X_1^2 + \beta_5 X_2^2 + \epsilon.$$

How do we decide which model to choose?

We can't just use the MSE. The second model is guaranteed to have a smaller MSE. (Do you see why?) We really want to use the model that will predict future observations well. That is, we want the model with the smallest generalization error.

2 Generalization and Optimism

We estimated our model by minimizing the mean squared error — or training error — on our data. In other words, $\widehat{\beta}$ was chosen to minimize

$$\frac{1}{n}(\mathbf{Y} - \mathbf{Xb})^T(\mathbf{Y} - \mathbf{Xb}).$$

Let

$$\widehat{m}(x) = \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \cdots + \widehat{\beta}_1 x_p$$

be our fitted model. How well will $\widehat{m}$ predict in the future? We define the **Generalization Error** or **Prediction error** as follows. Imagine a new data point (X, Y) where $X = (X_1, \ldots, X_p)$. Our prediction of Y using our model is $\widehat{m}(X)$. The **Generalization Error** or **Prediction error** is

$$G = \mathbb{E}[(Y - \widehat{m}(X))^2]. \tag{1}$$

Recall that the MSE or training error is

$$T = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{m}(X_i))^2. \tag{2}$$

In general, T is a poor estimate of G. In fact, we usually have that $T < G$.

For choosing models, we should use G instead of T. But to do this, we need a way to estimate G. Before we explain how to estimate G, let's first try to understand why the training error underestimates the generalization error.

3 Why is the Training Error Smaller than the Generalization Error?

The generalization error measures how well we can predict a future observation. An easier task is to predict new data at the same values of X_i as our traning data. We will show that even for this easier task, the training error is a poor estimate of generalization error.

Our original data were generated from the model

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon.$$

From these data we constructed our estimator $\widehat{\beta}$. Our fitted model is $\widehat{m}(x) = \widehat{\beta}_0 + \sum_j \widehat{\beta}_j X_j$. The fitted values are $\widehat{Y}_i = \widehat{m}(X_i)$. The training error — also called the *in-sample error* — is

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2.$$

Now imagine that we got a new data set at the same X_i's but with new errors:

$$\mathbf{Y}' = \mathbf{X}\beta + \epsilon'$$

where ϵ and ϵ' are independent but identically distributed. The design matrix is the same, the true parameters β are the same, but the noise is different. How well can we predict these new $Y_i's$? The predicted values using our model are still $\widehat{Y}_i = \widehat{m}(X_i)$. Define the *out-of-sample prediction error* by

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i' - \widehat{Y}_i)^2.$$

We will show that

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2\right] < \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i' - \widehat{Y}_i)^2\right]. \tag{3}$$

In fact, we will prove that

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i' - \widehat{Y}_i)^2\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2\right] + \frac{2}{n}\sigma^2(p+1). \tag{4}$$

Notice that $\widehat{Y}_i$ is a function of of all the $Y_i's$ so these are dependent random variables. On the other hand, $\widehat{Y}_i$ and Y_i' are completely statistically independent. (Remember that we're holding $\mathbf{X}$ fixed.)

Now

$$\begin{aligned}
\mathbb{E}\left[(Y_i - \widehat{Y}_i)^2\right] &= \mathrm{Var}\left[Y_i - \widehat{Y}_i\right] + \left(\mathbb{E}\left[Y_i - \widehat{Y}_i\right]\right)^2 && (5)\\
&= \mathrm{Var}\left[Y_i\right] + \mathrm{Var}\left[\widehat{Y}_i\right] - 2\mathrm{Cov}\left[Y_i, \widehat{Y}_i\right] + \left(\mathbb{E}\left[Y_i\right] - \mathbb{E}\left[\widehat{Y}_i\right]\right)^2. && (6)
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\mathbb{E}\left[(Y_i' - \widehat{Y}_i)^2\right] &= \mathrm{Var}\left[Y_i' - \widehat{Y}_i\right] + \left(\mathbb{E}\left[Y_i' - \widehat{Y}_i\right]\right)^2 && (7)\\
&= \mathrm{Var}\left[Y_i'\right] + \mathrm{Var}\left[\widehat{Y}_i\right] - 2\mathrm{Cov}\left[Y_i', \widehat{Y}_i\right] + \left(\mathbb{E}\left[Y_i'\right] - \mathbb{E}\left[\widehat{Y}_i\right]\right)^2. && (8)
\end{aligned}$$

Now Y_i' is independent of Y_i, but has the same distribution. This tells us that $\mathbb{E}\left[Y_i'\right] = \mathbb{E}\left[Y_i\right]$, $\mathrm{Var}\left[Y_i'\right] = \mathrm{Var}\left[Y_i\right]$, but $\mathrm{Cov}\left[Y_i', \widehat{Y}_i\right] = 0$. So

$$\begin{aligned}
\mathbb{E}\left[(Y_i' - \widehat{Y}_i)^2\right] &= \mathrm{Var}\left[Y_i\right] + \mathrm{Var}\left[\widehat{Y}_i\right] + \left(\mathbb{E}\left[Y_i\right] - \mathbb{E}\left[\widehat{Y}_i\right]\right)^2 && (9)\\
&= \mathbb{E}\left[(Y_i - \widehat{Y}_i)^2\right] + 2\mathrm{Cov}\left[Y_i, \widehat{Y}_i\right]. && (10)
\end{aligned}$$

Averaging over data points,

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i' - \widehat{Y}_i)^2\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2\right] + \frac{2}{n}\sum_{i=1}^{n}\mathrm{Cov}\left[Y_i, \widehat{Y}_i\right].$$

For a linear model, it can be shown that $\mathrm{Cov}\left[Y_i, \widehat{Y}_i\right] = \sigma^2 H_{ii}$. So,

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i' - \widehat{Y}_i)^2\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2\right] + \frac{2}{n}\sigma^2 \,\mathrm{tr}\,\mathbf{H}$$

and we know that with p predictors and one intercept, $\mathrm{tr}\,\mathbf{H} = p + 1$. Hence,

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i' - \widehat{Y}_i)^2\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2\right] + \frac{2}{n}\sigma^2(p+1).$$

Thus we have proved (4).

The term $(2/n)\sigma^2(p+1)$ is called the **optimism** of the model — the amount by which its in-sample MSE systematically under-estimates its true expected squared error. Notice that the optimism:

- Grows with σ^2: more noise gives the model more opportunities to seem to fit well by capitalizing on chance.
- Shrinks with n: at any fixed level of noise, more data makes it harder to pretend the fit is better than it really is.
- Grows with p: every extra parameter is another control which can be adjusted to fit to the noise.

Minimizing the in-sample MSE completely ignores the bias from optimism, so it is guaranteed to pick models which are too large and predict poorly out of sample.

4 Cross-Validation

The best way to estimate generaliation error is cross-validation. There are two main flavors of cross-validation: K-fold cross-validation and leave-one-out cross-validation. K-fold cross-validation is better but leave-one-out cross-validation is faster.

4.1 K-fold Cross-Validation

K-fold cross-validation goes as follows.

- Randomly divide the data into K equally-sized parts, or "folds". A common choice is $K = 5$ or $K = 10$.
- For each fold
  - Temporarily hold out that fold, calling it the "testing set".

- Call the other $K-1$ folds, taken together, the "training set".
- Estimate each model on the training set.
- Calculate the MSE of each model on the testing set.

- Average MSEs over folds.

We then pick the model with the lowest MSE, averaged across testing sets.

In other words, divide the data into K groups $B_1, \ldots, B_K$. For $j \in \{1, \ldots, K\}$, estimate $\widehat{m}$ from the data $\{B_1, \ldots, B_{j-1}, B_{j+1}, \ldots, B_K\}$. Then let

$$\widehat{G}_j = \frac{1}{n_j} \sum_{i \in B_j} (Y_i - \widehat{m}(X_i))^2$$

where n_j is the number of points in B_j. Finally, we estimate the generalization error by

$$\widehat{G} = \frac{1}{K} \sum_{j=1}^{K} \widehat{G}_j.$$

4.2 Leave-one-out Cross-Validation (LOOCV)

Let $\widehat{Y}_i^{(-i)}$ be the predicted value when we leave out (X_i, Y_i) from the dataset. The **leave-one-out cross-validation score** (LOOCV) is

$$LOOCV = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \widehat{Y}_i^{(-i)})^2.$$

Computing LOOCV sounds painful. Fortunately, there is a simple shortcut formula:

$$LOOCV = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{Y_i - \widehat{Y}_i}{1 - H_{ii}} \right)^2.$$

So computing LOOCV is fast.

It also interesting to note the following. We know that $\text{tr}(\mathbf{H}) = p + 1$. So the average value of the $H_{ii}'s$ is $\gamma \equiv (p+1)/n$. If we approximate each H_{ii} with γ we have

$$LOOCV \approx \frac{1}{n} \sum_{i=1}^{n} \left(\frac{Y_i - \widehat{Y}_i}{1 - \gamma} \right)^2 ..$$

By doing a Taylor series we see that $(1-\gamma)^{-2} \approx 1 + 2\gamma$. Hence,

$$LOOCV \approx \frac{1 + 2\gamma}{n} \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2 \tag{11}$$

$$= \frac{1}{n} \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2 + 2\gamma \frac{1}{n} \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2 \tag{12}$$

$$= \text{training error } + \frac{2\widehat{\sigma}^2}{n}(p+1). \tag{13}$$

5 Mallow's C_p Statistic

The Mallows C_p statistic just substitutes in a feasible estimator of σ^2 into the optimism. Usually we take $\widehat{\sigma}^2$ *from the largest model we consider*. This will be an unbiased estimator of σ^2 if the real model is smaller (contains a strict subset of the predictor variables), but not vice versa. That is, for a linear model with $p+1$ coefficients fit by OLS,

$$C_p = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2 + \frac{2\widehat{\sigma}^2}{n}(p+1). \tag{14}$$

Notice how similar this is to (13). The selection rule is to pick the model which minimizes C_p.

We can think of C_p as having two parts,

$$C_p = MSE + (\text{penalty})$$

From one point of view, the penalty is just an estimate of the bias. From another point of view, it's a cost we're imposing on models for having extra parameters. Every new parameter has got to pay that cost by reducing the MSE by at least a certain amount; if it doesn't, the extra parameter isn't worth it.

For comparing models, we really care about differences:

$$\Delta C_p = MSE_1 - MSE_2 + \frac{2}{n}\widehat{\sigma}^2(p_1 - p_2) \tag{15}$$

Alternate form of C_p. You will find many references which define C_p somewhat differently:

$$\frac{nMSE}{\widehat{\sigma}^2} - n + 2p \tag{16}$$

and say that the optimal value is close to p, not close to 0. To see that this selects exactly the same models as the rule given above, take a difference between two models, with MSE's MSE_1, MSE_2 and p_1, p_2 predictors. We get

$$\frac{n(MSE_1 - MSE_2)}{\widehat{\sigma}^2} + 2(p_1 - p_2)$$

Dividing by n and multiplying by $\widehat{\sigma}^2$ gives us back Eq. 15. There are reasons to assert that Eq. 16 should indeed be close to p for the right model (if the Gaussian noise assumption holds), but Eq. 14 is a good estimate of the out-of-sample error, and a good model selection rule, much more broadly.

6 R^2 and Adjusted R^2 (OPTIONAL)

Recall that

$$R^2 = 1 - \frac{MSE}{s_Y^2}$$

Picking a model by maximizing R^2 is thus equivalent to picking a model by minimizing MSE. It is therefore bad for exactly the same reasons that minimizing MSE across models is bad.

Recall that the adjusted R^2 is

$$R^2_{adj} = 1 - \frac{MSE\frac{n}{n-p-1}}{s^2_Y}$$

That is, it's R^2 with the unbiased estimator of σ^2. Maximizing adjusted R^2 therefore corresponds to minimizing that unbiased estimator. What does that translate to?

$$\begin{aligned} MSE\frac{n}{n-p-1} &= MSE\frac{1}{1-(p+1)/n} && (17)\\ &\approx MSE\left(1+\frac{p+1}{n}\right) && (18)\\ &= MSE + MSE\frac{p+1}{n} && (19)\end{aligned}$$

where the approximation becomes exact as $n \to \infty$ with p fixed. Even for the completely right model, where MSE is a consistent estimator of $\widehat{\sigma}^2$, the correction or penalty is only half as big as we've seen it should be. Selecting models using adjusted R^2 is not completely stupid, as maximizing R^2 is, but it is still not going to work very well.

7 Akaike Information Criterion (AIC)

The great Japanese statistician Hirotugu Akaike proposed a famous model selection rule which also has the form of "in-sample performance plus penalty". What has come to be called the **Akaike information criterion** (AIC) is

$$AIC(S) \equiv L_S - \dim(S)$$

where L_S is the log likelihood of the model S, evaluated at the maximum likelihood estimate, and $\dim(S)$ is the dimension of S, the number of adjustable parameters it has. Akaike's rule is to pick the model which maximizes AIC.

The reason for this definition is that Akaike showed AIC/n is an unbiased estimate of the expected log-probability the estimated parameters will give to a new data point which it hasn't seen before, if the model is right. This is the natural counterpart of expected squared error for more general distributions than the Gaussian. IF we do specialize to linear-Gaussian models, then we have

$$L = -\frac{n}{2}(1+\log 2\pi) - \frac{n}{2}\log MSE$$

and the dimension of the model is $p+2$ (because σ^2 is also an adjustable parameter). Notice that $-\frac{n}{2}(1+\log 2\pi)$ doesn't involve the parameters at all. If we compare AICs for two models, with mean squared errors in-sample of MSE_1 and MSE_2, and one with p_1 predictors and the other with p_2, the difference in AICs will be

$$\Delta AIC = -\frac{n}{2}\log MSE_1 + \frac{n}{2}\log MSE_2 - (p_1 - p_2).$$

To relate this to C_p, let's write $MSE_2 = MSE_1 + \Delta MSE$. Then

$$\begin{aligned} \Delta AIC &= -\frac{n}{2}\log MSE_1 + \frac{n}{2}\log MSE_1\left(1+\frac{\Delta MSE}{MSE_1}\right) - (p_1-p_2) && (20)\\ &= -\frac{n}{2}\log\left(1+\frac{\Delta MSE}{MSE_1}\right) - (p_1-p_2). && (21)\end{aligned}$$

Now let's suppose that model 1 is actually the correct model, so $MSE_1 = \widehat{\sigma}^2$, and that ΔMSE is small compared to $\widehat{\sigma}^2$, so

$$\Delta AIC \approx -\frac{n}{2}\frac{\Delta MSE}{\widehat{\sigma}^2} - (p_1 - p_2) \tag{22}$$

$$\frac{-2\widehat{\sigma}^2}{n}\Delta AIC \approx \Delta MSE + \frac{2}{n}\widehat{\sigma^2}(p_1 - p_2) = \Delta C_p. \tag{23}$$

So, if one of the models we're looking at is actually the correct model, and the others aren't too different from it, picking by maximizing AIC will give the same answer as picking by minimizing C_p.

Other Uses of AIC AIC can be applied whenever we have a likelihood. It is therefore used for tasks like comparing models of probability distributions, or predictive models where the whole distribution is important. C_p, by contrast, really only makes sense if we're trying to do regression and want to use squared error.

7.1 BIC

Another model selection criterion is BIC (Bayesian Information Criterion) developed by Schwarz. The BIC for a model S is

$$BIC(S) = L_S - \frac{\log n}{2}\dim(S).$$

This is a stronger penalty than AIC applies, and this has consequences:

> As $n \to \infty$, if the true model is among those BIC can select among, BIC will tend to pick the true model.

Of course there are various conditions attached to this, some of them quite technical, but it's generally true for IID samples, for regression modeling, for many sorts of time series model, etc. Unfortunately, the model selected by BIC will tend to predict less well than the one selected by leave-one-out cross-validation or AIC.

8 Summary

Cross-validation, AIC and C_p all have the same goal: try to find a model that predicts well. They tend to choose similar models. BIC is quite different and tends to choose smaller models. Cross-validation is very general can be used in more settings than the others. There are theorems that say that cross-validation is very effective at estimating generalization error. These theorems make very few assumptions.

9 Inference after Selection

All of the inferential statistics we have done in earlier lectures presumed that our choice of model was completely fixed, and not at all dependent on the data. If different data sets would lead us to use different models, and our data are (partly) random, then which model we're using is also

random. This leads to some extra uncertainty in, say, our estimate of the slope on X_1, which is *not* accounted for by our formulas for the sampling distributions, hypothesis tests, confidence sets, etc.

A very common response to this problem, among practitioners, is to ignore it, or at least hope it doesn't matter. This can be OK, if the data-generating distribution forces us to pick one model with very high probability, or if all of the models we might pick are very similar to each other. Otherwise, ignoring it leads to nonsense.

Here, for instance, I simulate 200 data points where the Y variable is a standard Gaussian, and there are 100 independent predictor variables, all also standard Gaussians, independent of each other *and of* Y:

```
n = 200
p = 100
y = rnorm(n)
x = matrix(rnorm(n*p),nrow=n)
df = data.frame(y=y,x)
mdl = lm(y~., data=df)
```

Of the 100 predictors, 5 have t-statistics which are significant at the 0.05 level or less. (The expected number would be 5.) If we select the model using just those variables we get

```
##
## Call:
## lm(formula = y ~ ., data = df[, c(1, stars)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.53035 -0.75081  0.03042  0.58347  2.63677
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03084    0.07092   0.435   0.6641
## X21         -0.13821    0.07432  -1.860   0.0644
## X25          0.12472    0.06945   1.796   0.0741
## X41          0.13696    0.07279   1.882   0.0614
## X83         -0.03067    0.07239  -0.424   0.6722
## X88          0.14585    0.07040   2.072   0.0396
##
## Residual standard error: 0.9926 on 194 degrees of freedom
## Multiple R-squared:  0.06209,Adjusted R-squared:  0.03792
## F-statistic: 2.569 on 5 and 194 DF,  p-value: 0.02818
```

Notice that final over-all F statistic: it's testing whether including those variables fits better than an intercept-only model, and saying it thinks it does, with a definitely significant p-value. This is the case even though, by construction, the response is *completely independent* of *all* predictors. This is not a fluke: if you re-run my simulation many times, your p-values in the full F test will not be uniformly distributed (as they would be on all 100 predictors), but rather will have a distribution

strongly shifted over to the left. Similarly, if we looked at the confidence intervals, they would be much too narrow.

These issues do not go away if the true model isn't "everything is independent of everything else", but rather has some structure. Because we picked the model to predict well on this data, if we then run hypothesis tests on that same data, they'll be too likely to tell us everything is significant, and our confidence intervals will be too narrow. Doing statistical inference on the same data we used to select our model is just broken. It may not always be as spectacularly broken as in my demo above, but it's still broken.

There are three ways around this. One is to pretend the issue doesn't exist; as I said, this is popular, but it's got nothing else to recommend it. Another, is to not do tests or confidence intervals. The third approach, which is in many ways the simplest, is to use data splitting.

Data splitting is (for regression) a very simple procedure:

- Randomly divide your data set into two parts.
- Calculate your favorite model selection criterion for all your candidate models using only the first part of the data. Pick one model as the winner.
- Re-estimate the winner, and calculate all your inferential statistics, using only the other half of the data.

(Division into two equal halves is optional, but usual.)

Because the winning model is statistically independent of the second half of the data, the confidence intervals, hypothesis tests, etc., can treat it as though that model were fixed *a priori*. Since we're only using $n/2$ data points to calculate confidence intervals (or whatever), they will be somewhat wider than if we really had fixed the model in advance and used all n data points, but that's the price we pay for having to select a model based on data.

10 R Practicalities

You can get the LOOCV from R as follows:

```
out = lm(y ~ x)
LOOCV = mean(((y - fitted(out))/(1-hatvalues(out)))^2)
```

To get C_p I suggest you write your own function. To get AIC use `glm` instead out `lm` like this:

```
out = glm(y ~ x)
out$aic
```

To get K-fold cross-validation you can either write your own code (excellent idea!) or use the `boot` package together with `glm` as follows:

```
library(boot)
out = glm(y ~ x,data = D)
cv.glm(D,out,K=5)$delta[1]
```

It looks pretty strange, but that will give you the value that you want. Here is an example. We will generate data from a quadratic. We will then fit polynomials up to order 10. Then we will plot the LOOCV and the K-fold cross-validation using $K = 5$.

```
library("boot")

## generate the data
n = 100
x = runif(n)
y = 2 + x - 3*x^2 + rnorm(n,0,.1)
D = data.frame(x=x,y=y)

## plot the data
pdf("PolynomialExample1.pdf")
plot(x,y)
dev.off()
```

```
pdf("PolynomialExample2.pdf")
LOOCV = rep(0,10)
KFoldCV = rep(0,10)

## fit polynomials and get the cross-validation scores
for(j in 1:10){
     out = glm(y ~ poly(x,j),data=D)
     print(summary(out))
     LOOCV[j] = mean(((y - fitted(out))/(1-hatvalues(out)))^2)
     KFoldCV[j] = cv.glm(D,out,K=5)$delta[1]
     }

## plot them
plot(1:10,LOOCV,type="l",lwd=3)
lines(1:10,KFoldCV,lwd=3,col="blue")
dev.off()
```

11 History

Cross-validation goes back in statistics into the 1950s, if not earlier, but did not become formalized as a tool until the 1970s, with the work of Stone (1974). It was adopted, along with many other statistical ideas, by computer scientists during the period in the late 1980s–early 1990s when the modern area of "machine learning" emerged from (parts of) earlier areas called "artificial intelligence", "pattern recognition", "connectionism", "neural networks", or indeed "machine learning". Subsequently, many of the scientific descendants of the early machine learners forgot where their

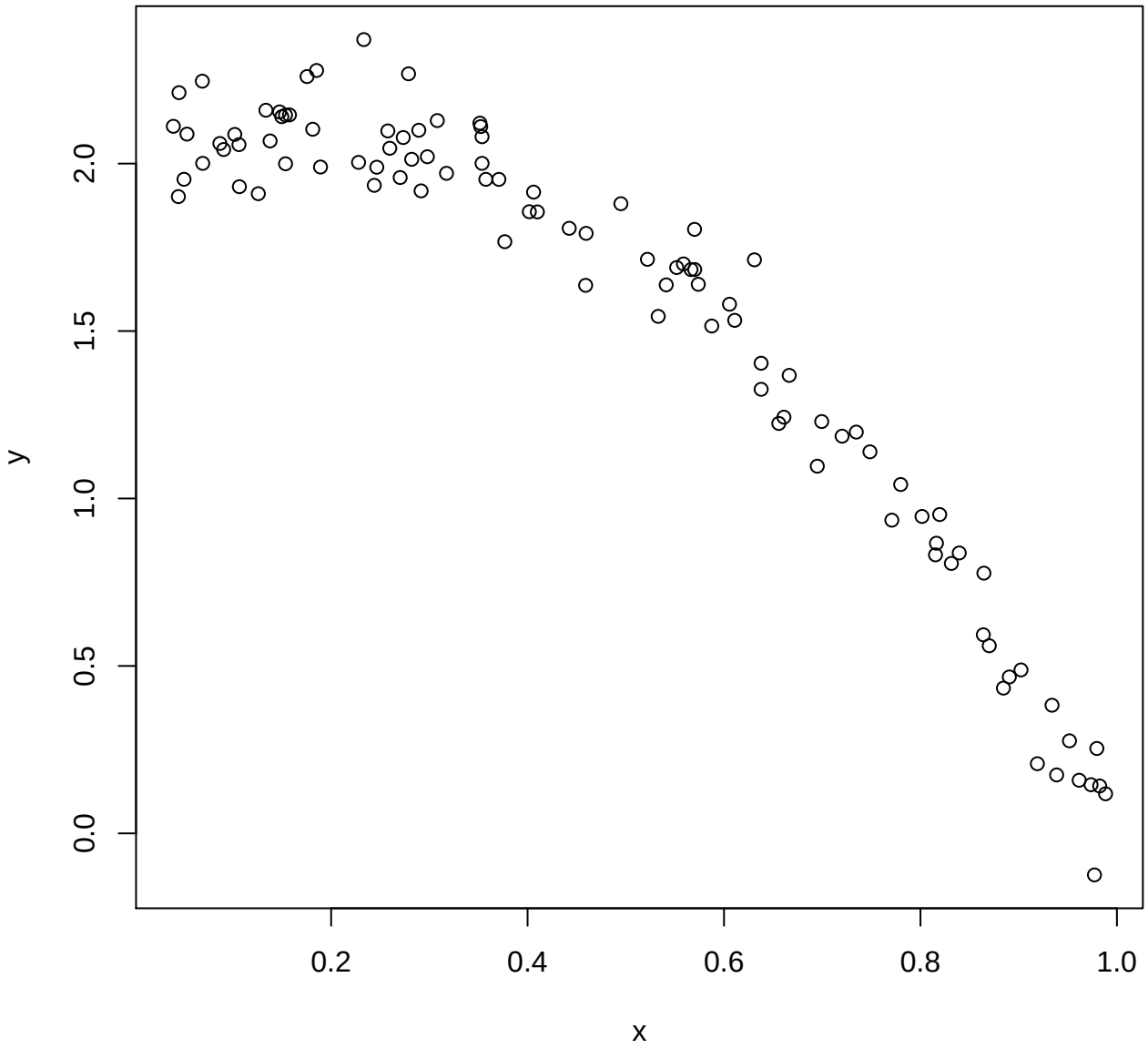

FIGURE 1: *The data.*

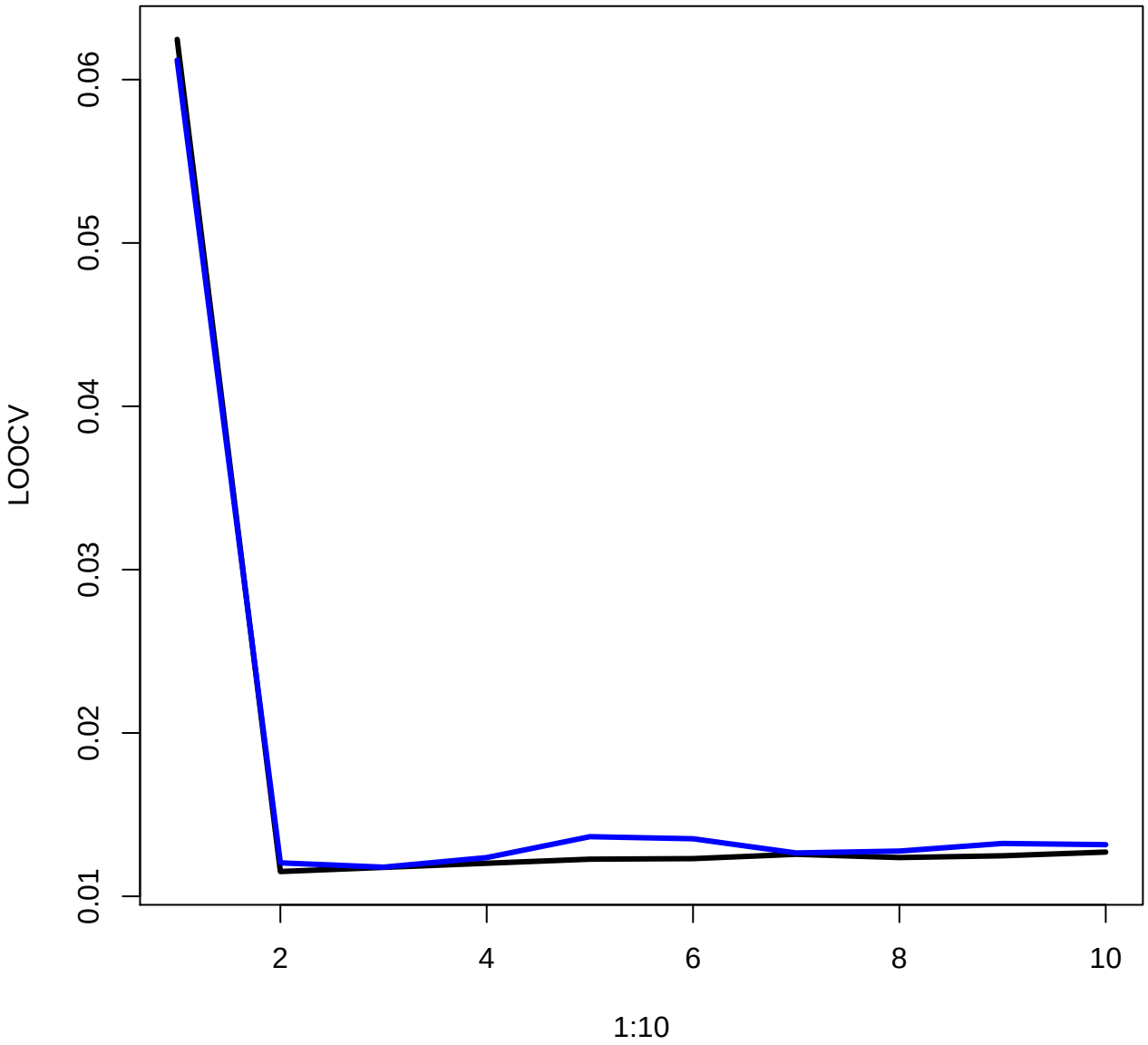

FIGURE 2: *LOOCV is in black. K-fold is in blue. The x-axis is the order of the polynomial.*

ideas came from, to the point where many people now think cross-validation is something computer science contributed to data analysis.

12 More on AIC (OPTIONAL)

Akaike had a truly brilliant argument for subtracting a penalty equal to the number of parameters from the log-likelihood, which is too pretty not to at least sketch here.[1]

Generically, say that the parameter vector is θ, and its true value is θ^*. (For linear regression with Gaussian noise, θ consists of all $p+1$ coefficients plus σ^2.) The length of this vector, which is $\dim(S)$, is let's say d. (For linear regression with Gaussian noise, $d = p+2$.) The maximum likelihood estimate is $\widehat{\theta}$. We know that the derivative of the likelihood is zero at the MLE:

$$\nabla L(\widehat{\theta}) = 0$$

Let's do a Taylor series expansion of $\nabla L(\theta)$ around the true parameter value θ^*:

$$\nabla L(\theta) = \nabla L(\theta^*) + (\theta - \theta^*)\nabla\nabla L(\theta^*)$$

Here $\nabla\nabla L(\theta^*)$ is the $d \times d$ matrix of second partial derivatives of L, evaluated at θ^*. This is called the **Hessian**, and would traditionally be written $\mathbf{H}$, but that would lead to confusion with the hat matrix, so I'll call it $\mathbf{K}$. Therefore the Taylor expansion for the gradient of the log-likelihood is

$$\nabla L(\theta) = \nabla L(\theta^*) + (\theta - \theta^*)\mathbf{K}$$

Applied to the MLE,

$$\mathbf{0} = \nabla L(\theta^*) + (\widehat{\theta} - \theta^*)\mathbf{K}$$

or

$$\widehat{\theta} = \theta^* - K^{-1}\nabla L(\theta^*)$$

What is the *expected* log-likelihood, on new data, of $\widehat{\theta}$? Call this expected log-likelihood ℓ (using a lower-case letter to indicate that it is non-random). Doing another Taylor series,

$$\ell(\theta) \approx \ell(\theta^*) + (\theta - \theta^*)^T\nabla\ell(\theta^*) + \frac{1}{2}(\theta - \theta^*)^T\nabla\nabla\ell(\theta^*)(\theta - \theta^*)$$

However, it's not hard to show that the expected log-likelihood is always[2] maximized by the true parameters, so $\nabla\ell(\theta^*) = 0$. (The same argument also shows $\mathbb{E}\left[\nabla L(\theta^*)\right] = 0$.) Call the Hessian in this Taylor expansion $\mathbf{k}$. (Again, notice the lower-case letter for a non-random quantity.) We have

$$\ell(\theta) \approx \ell(\theta^*) + \frac{1}{2}(\theta - \theta^*)^T\mathbf{k}(\theta - \theta^*)$$

Apply this to the MLE:

$$\ell(\widehat{\theta}) \approx \ell(\theta^*) + \frac{1}{2}\nabla L(\theta^*)\mathbf{K}^{-1}\mathbf{k}\mathbf{K}^{-1}\nabla L(\theta^*)$$

Taking expectations,

$$\mathbb{E}\left[\ell(\widehat{\theta})\right] \approx \ell(\theta^*) + \frac{1}{2}\operatorname{tr}\mathbf{K}^{-1}\mathbf{k}\mathbf{K}^{-1}\mathbf{J}$$

[1] Nonetheless, this subsection is optional.

[2] Except for quite weird models.

where $\mathrm{Var}\left[\nabla L(\theta^*)\right] = \mathbf{J}$. For large n, $\mathbf{K}$ converges on $\mathbf{k}$, so this simplifies to

$$\mathbb{E}\left[\ell(\widehat{\theta})\right] \approx \ell(\theta^*) + \frac{1}{2}\operatorname{tr}\mathbf{k}^{-1}\mathbf{J}$$

This still leaves things in terms of $\ell(\theta^*)$, which of course we don't know, but now we do another Taylor expansion, this time of L around $\widehat{\theta}$:

$$L(\theta^*) \approx L(\widehat{\theta}) + \frac{1}{2}(\theta^* - \widehat{\theta})^T \nabla\nabla L(\widehat{\theta})(\theta^* - \widehat{\theta})$$

so

$$L(\theta^*) \approx L(\widehat{\theta}) + \frac{1}{2}(\mathbf{K}^{-1}\nabla L(\theta^*))^T \nabla\nabla L(\widehat{\theta})(\mathbf{K}^{-1}\nabla L(\theta^*))$$

For large n, $\nabla\nabla L(\widehat{\theta}) \to \nabla\nabla L(\theta^*) \to \mathbf{k}$. So, again taking expectations,

$$\ell(\theta^*) \approx \mathbb{E}\left[L(\widehat{\theta}\right] + \frac{1}{2}\operatorname{tr}\mathbf{k}^{-1}\mathbf{J}$$

Putting these together,

$$\mathbb{E}\left[\ell(\widehat{\theta})\right] \approx \mathbb{E}\left[L(\widehat{\theta}\right] + \operatorname{tr}\mathbf{k}^{-1}\mathbf{J}$$

An unbiased estimate is therefore

$$L(\widehat{\theta}) + \operatorname{tr}\mathbf{k}^{-1}\mathbf{J}$$

Finally, a fundamental result (the "Fisher identity") says that for well-behaved models, *if* the model is correct, then

$$\mathrm{Var}\left[\nabla L(\theta^*)\right] = -\nabla\nabla\ell(\theta^*)$$

or $\mathbf{J} = -\mathbf{k}$. Hence, if the model is correct, our unbiased estimate is just

$$L(\widehat{\theta}) - \operatorname{tr}\mathbf{I}$$

and of course $\operatorname{tr}\mathbf{I} = d$.

There, as you'll notice, several steps where we're making a bunch of approximations. Some of these approximations (especially those involving the Taylor expansions) can be shown to be OK asymptotically (i.e., as $n \to \infty$) by more careful math. The last steps, however, where we invoke the Fisher identity, are rather more dubious. (After all, all of the models we're working with can hardly contain the true distribution.) A somewhat more robust version of AIC is therefore to use as the criterion

$$L(\widehat{\theta}) + \operatorname{tr}\mathbf{K}\mathbf{J}$$