Regression extensions and model selection

Chrysafis Vogiatzis

Lecture 33

Learning objectives

After lecture 33, we will be able to:

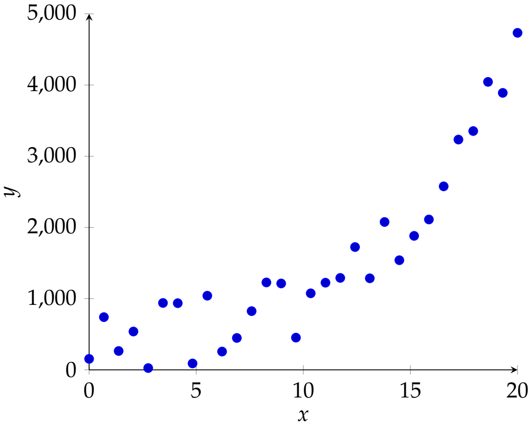
- Perform and interpret polynomial regression.
- Perform and interpret simple nonlinear regression.
- Build regression models with multiple predictors using:
  - all subsets selection.
  - backwards selection.
  - forwards selection.
- Describe and implement an "80-20" validation strategy.
- Describe and implement a K-fold validation strategy.

Motivation: Higher degree terms

What if our relationships is not linear, but is instead a more general **polynomial**? For example, what if I am sure that the yield of a crop is related to the square of the temperature? How could we incorporate this information into our regression models?

Or, what if I plot my data in a scatter plot and get an image like the one in Figure 1? How can I use regression to fit this data to a line?

Figure 1: The scatter plot containing all of our data points.

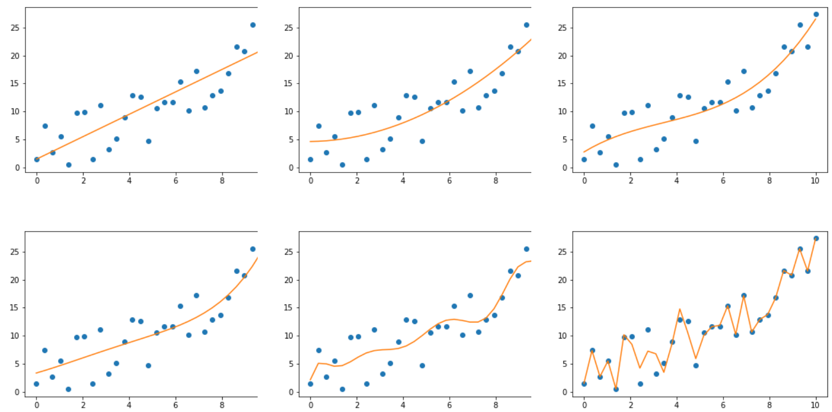


Motivation: Model building

Ok, so we have seen how to build models using 1 or $k > 1$ predictor variables. But given many possible predictor variables, how can we find the combination that works best?

Polynomial regression

Let's start with a question. Which of the following regression models do you believe best captures the data?



The first model (shown on the top left) is your typical simple linear regression. The other five models add some "curvature" by allowing higher degrees in the regression. For example, the second model is a quadratic term, whereas the last two are regressions that includes terms at the power of 10 and 25!

So, assume you have tried simple linear regression and the results have been underwhelming. You would like, instead to try the following line:

$$y = \beta_0 + \beta_1 x + \beta_{11} x_1^2.$$

A couple of notes:

1. We only consider simple linear regression for simplicity: we could very easily extend this to multiple linear regression.

2. There is one predictor variable: but it appears twice in our regression, one with degree 1 and one with degree 2. This is a quadratic regression!

How can we deal with a regression like this? Well, we can follow the next steps:

1. Create a "new" predictor variable (let us call it x_2).

2. Set x_2 equal to x_1^2: $x_2 = x_1^2$.
3. Set up a **multiple linear regression** using matrix X based on *two* predictor variables: x_1 and $x_2 = x_1^2$.
4. Find $\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_{11} \end{bmatrix} = \left(X^T X\right)^{-1} X^T y$.

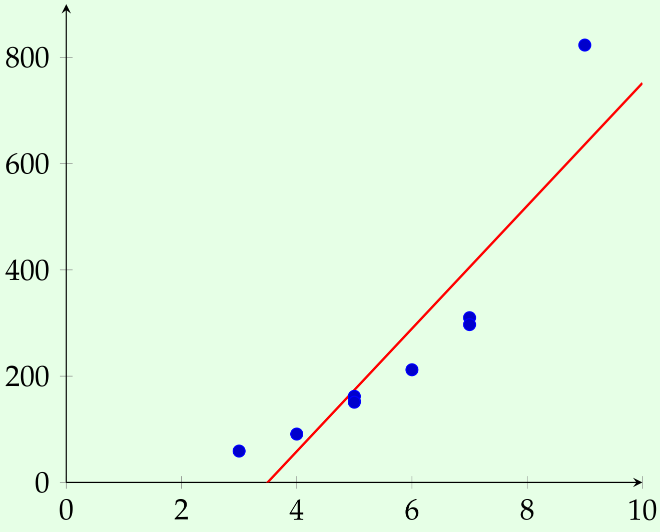
Let us put this to the test right away.

A small quadratic regression model

Consider the following data:

x	y
7	310
3	59
5	153
5	162
4	91
6	212
7	297
5	151
9	823

We tried a linear regression and got the line $y = 7.2404x - 2.2194$.



Since it does not look great, we decide to try a second degree regression polynomial of the form: $y = \beta_0 + \beta_1 x + \beta_{11} x^2$. What are $\hat{\beta}_0, \hat{\beta}_1, b\hat{e}ta_{11}$?

A small quadratic regression model

1. Add a new column in your data that is equal to x^2.

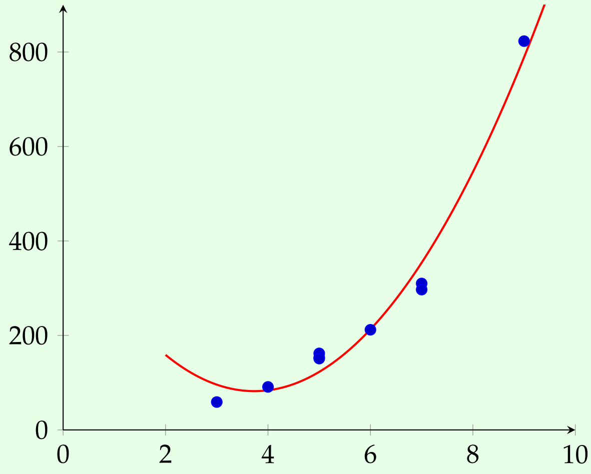
x	x^2	y
7	49	310
3	9	59
5	25	153
5	25	162
4	16	91
6	36	212
7	49	297
5	25	151
9	81	823

2. Construct X:

$$X = \begin{bmatrix} 1 & 7 & 49 \\ 1 & 3 & 9 \\ 1 & 5 & 25 \\ 1 & 5 & 25 \\ 1 & 4 & 16 \\ 1 & 6 & 36 \\ 1 & 7 & 49 \\ 1 & 5 & 25 \\ 1 & 9 & 81 \end{bmatrix}$$

3. Solve for $\hat{\beta}$:

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_{11} \end{bmatrix} = \left(X^T X\right)^{-1} X^T y = \begin{bmatrix} 437.74 \\ -190.47 \\ 25.5 \end{bmatrix}$$



Look at how much nicer this looks like!

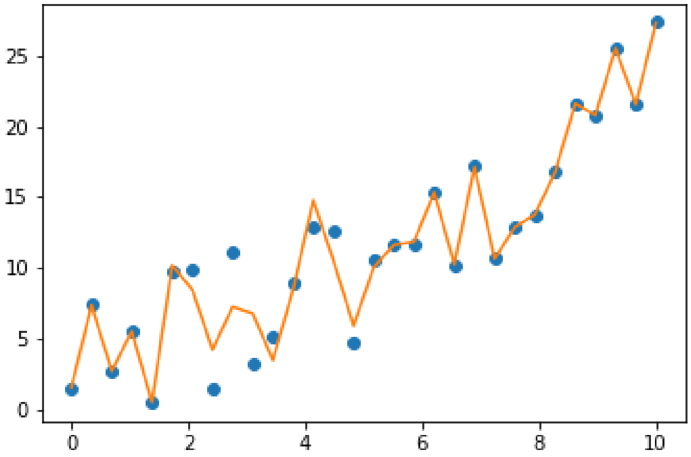
We can follow the same logic with other nonlinear functions!

Some nonlinear transformation examples

- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2$
  - Introduce new variable $x_{12} = x_1 x_2$ and solve.
- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{123} x_1 x_2 x_3$
  - Introduce new variable $x_{123} = x_1 x_2 x_3$ and solve.
- We can even do that with other nonlinear functions: for example $y = \beta_0 + \beta_1 x_1 + \beta_2 cos(x_1)$.
  - Introduce new variable $x_2 = cos(x_1)$ and solve.
- Or $y = \beta_0 + \beta_1 x_1 + \beta_2 \log x_1$.
  - Introduce new variable $x_2 = \log x_1$ and solve.

Finally, what is the appropriate model? Now that we can go nonlinear, we could (if we wanted to) make almost all residuals equal to zero! See for example the regression curve in Figure 2.

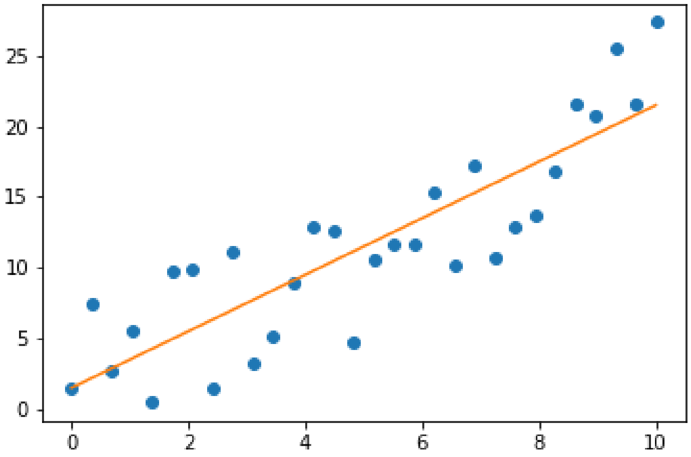
Figure 2: An example of **overfitting**. Here, we end up following the given data too closely, not allowing for any randomness at all.



Of course, the opposite route is still very much possible. We may decide that the simplest, linear regression may be the way to go. The previous two cases are called **overfitting** and **underfitting**.

- Overfitting is an issue because we end up getting too caught up on past information, and hence we lose our edge to predict the future if it doesn't look exactly like the past.

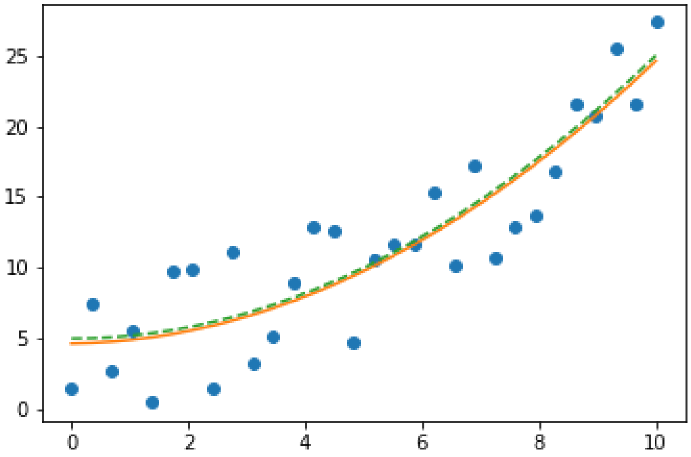
Figure 3: An example of **underfitting**. Here, we end up with a simple linear regression that does not seem to follow the data as well.



- Underfitting, on the other hand, is an issue of oversimplification: our model does not predict well because it is missing information.

We would like to do an **appropriate model selection**. How?

Figure 4: An appropriate model which balances past information and flexibility to new data.



Before we see the how, a couple of quotes that can help us drive the point of model selection home:

1. Paul Valéry (philosopher, 1942)

   > "Ce qui est simple est toujours faux. Ce qui ne l'est pas est inutilisable." [1]

2. George Box (statistician, 1978)

   > "All models are wrong, but some are useful."

Model selection

In most problems, we have *many* potential variables to consider. To make things worse, we can include different *functions* of the variables themselves! Which ones should be included?

Which to include?

We want to build a regression model using any combination of three factors x_1, x_2, x_3. We can build any of the following models:

1. x_1 alone, or x_2 alone, or x_3 alone. 3 models.
2. x_1 and x_2 but not x_3, or x_1 and x_3 but not x_2, or x_2 and x_3 but not x_2. 3 models.
3. x_1, x_2, and x_3 toghether. 1 model.
4. None of them! 1 model.

The last case happens when all three factors are not significant to the regression.

To make matters worse, assume if we could also add *their squares*: $x_1, x_2, x_3, x_1^2, x_2^2, x_3^2$ for a total of $2^6 = 64$ possible models. Which one should we build?

Hopefully by now you are motivated and you see how this is an important problem that needs to be addressed. Even more so nowadays, with the advent of big data. It is imperative we find out a way to trim the model so that only significant factors are included. In the next few pages, we discuss three model building approaches, called:

1. **all subsets** selection.
2. **backwards** selection.
3. **forwards** selection.

[1] "What is simple is always wrong. What is not simple is impossible to use."

All subsets selection

All subsets selection is a term to signal that we need to consider *all* possible subsets we can create with our factors: these can be quite many. They actually grow exponentially and with k predictor variables, we already have 2^k possible subsets. [2]

Among all 2^k possible subsets, pick the subset of predictor variables/factors that leads to the largest R^2_{adj}.

Back to the realtor.com example

Consider the realtor.com example from last time. We assumed a house's price depends on area (sq. ft.), the year built, the garage spots, the number of bedrooms, and the number of bathrooms. Which subset of variables gives us the best regression model?

We have 32 combinations to consider (including the empty set, which implies that neither factor is significant). Some are presented here:

- $(x_1, x_2, x_3, x_4, x_5)$: $R^2_{adj} = 0.827$
- (x_1, x_2, x_3, x_4): $R^2_{adj} = 0.882$
- (x_2, x_3, x_4, x_5): $R^2_{adj} = 0.831$
- (x_1, x_3, x_4, x_5): $R^2_{adj} = 0.883$
- ...
- (x_1, x_2, x_3): $R^2_{adj} = 0.910$
- ...
- (x_1, x_3, x_4): $R^2_{adj} = 0.909$
- (x_1, x_3, x_5): $R^2_{adj} = 0.910$
- ...
- (x_1, x_2): $R^2_{adj} = 0.919$
- (x_1, x_3): $R^2_{adj} = 0.926$
- ...
- (x_1): $R^2_{adj} = 0.927$
- $\emptyset$: $R^2_{adj} = 0.857$

Among them, pick the one with the largest R^2_{adj}. In our case, that would be the model with **only** x_1.

Backwards selection

We immediately see the issue with the previous case: too many combinations to consider, even for few predictor variables. To avoid enu-

[2] Why?

merating fully all subsets, we investigate two heuristic approaches. With the term heuristic approach, we mean an approach that is not guaranteed to give us the optimal subset; that said, we expect its solution to be obtained faster. We proceed to describe the apporach.

1. First, start by including **all** predictor variables in your regression model.
2. Do a hypothesis test for significance of each individual factor among the predictor variables in your current regression.
3. Check if all P-values are above some threshold (say $p > 0.10$).
4. If not, find the one factor with the *largest* P-value.
   - This is the "least significant" predictor.
5. Remove it from consideration and calculate the new R^2_{adj}. If it is lower than the previously obtained R^2_{adj}, stop. Otherwise, iterate (go back to step 1) after removing the factor.

If P-values are not readily available, we may compare each T-test value $|T_0|$ to $t_{\alpha/2,n-k-1}$ and see if you'd accept/reject the hypothesis. Then, pick the variable which is the farthest from the rejection area and remove it from consideration instead.

Back to the realtor.com example

In the notes from Lecture 32, we did individual hypothesis tests for each of the factors. We had gotten:

1. x_1: $T_0 = 0.964$ P-value $= 0.437$.
2. x_2: $T_0 = 0.195$ P-value $= 0.864$.
3. x_3: $T_0 = -0.234$ P-value $= 0.837$.
4. x_4: $T_0 = 0.220$ P-value $= 0.846$.
5. x_5: $T_0 = 0.237$ P-value $= 0.835$.

The R^2_{adj} for the full model with all five variables is equal to 0.827. Use the backwards selection heuristic approach to find a good regression model.

Back to the realtor.com example

We remove the one with the largest P-value (the one with the T_0 test statistic value that is farthest from rejection): x_2. We then ran the new regression with the remaining four variables to get that :

1. x_1: $T_0 = 1.391$ P-value $= 0.258$.
2. x_3: $T_0 = -0.393$ P-value $= 0.720$.
3. x_4: $T_0 = 0.250$ P-value $= 0.819$.
4. x_5: $T_0 = 0.291$ P-value $= 0.790$.

The new R^2_{adj} is equal to 0.883: since it has improved, continue with the next iteration. From the remaining factors, we now remove x_4 (the highest P-value). The new model (including x_1, x_3, x_5) leads to $R^2_{adj} = 0.910$. This is an improvement, so we continue. Again, the new model includes:

1. x_1: $T_0 = 5.699$ P-value $= 0.005$.
2. x_3: $T_0 = -0.850$ P-value $= 0.443$.
3. x_5: $T_0 = 0.249$ P-value $= 0.816$.

x_5 is set to be removed, leaving us with a model including only x_1, x_3. The new R^2_{adj} is 0.926 – again improving the previous one. Hence, we get:

1. x_1: $T_0 = 7.535$ P-value $= 0.0007$.
2. x_3: $T_0 = -0.993$ P-value $= 0.366$.

Note how x_3 has a P-value above 0.1: let's try removing it and keep a model with **only** x_1. Its R^2_{adj} is equal to 0.927 – another improvement! Removing x_1, though, we obtain the empty model with $R^2_{adj} = 0.857$. Since it worsens, we stick with the model with x_1 alone.

Forwards selection

Forwards selection is – you guessed it – the opposite idea!

1. First, start by including **none** of the predictor variables.
   - That is, we have a line based only on the intercept β_0.

2. Then, run k separate regression models, one for each of the predictor variables.

3. Check whether any of the P-values in each individual test is below some threshold (say $p < 0.10$).

4. Pick the regression variable that leads to the *smallest* P-value. Equivalently, you may check the variable whose addition increases R^2_{adj} the most.

   - This is the "most significant" predictor.

5. Add it to the model and continue to run $k - 1$ separate regression lines: each with the variable from the first part, and one of the remaining variables.

6. Iterate and stop when no more variables have a P-value that is lower than 0.10 (or when no addition leads to an increased R^2_{adj}).

Back to the realtor.com example

Let us solve the same problem, but using forwards selection now.

- First, perform five different regressions, one per variable.

  1. x_1: $T_0 = 9.492$ P-value $= 7.79 \cdot 10^{-5}$.
  2. x_2: $T_0 = -3.354$ P-value $= 0.015$.
  3. x_3: $T_0 = 4.396$ P-value $= 0.005$.
  4. x_4: $T_0 = 6.343$ P-value $= 0.0007$.
  5. x_5: $T_0 = 1.750$ P-value $= 0.131$.

- We add the one with the smallest P-value (the one with the T test statistic value that is the easiest to reject): x_1. The current model has $R^2_{adj} = 0.926$.

- We then ran the new regression with the one variable from earlier (x_1) plus each of the remaining four:

  1. (x_1, x_2): $T_0 = 0.632$ P-value $= 0.555$.
  2. (x_1, x_3): $T_0 = -0.993$ P-value $= 0.366$.
  3. (x_1, x_4): $T_0 = 0.741$ P-value $= 0.492$.
  4. (x_1, x_5): $T_0 = 0.386$ P-value $= 0.716$.

All P-values are above 0.1, so we stop. The model obtained from forwards selection is the one including x_1 alone.

Validation

We have build a regression model based on past data and we now want to put it to the test. But before we do that, we want to check how confident should we be in our model? How can we validate our model selection?

Traditionally, the main idea has been to split our data (the data that we would normally use to build our model!) in two parts: **training** data and **testing** data. The common split between these two is 80%-20% in favor of training. Now, we:

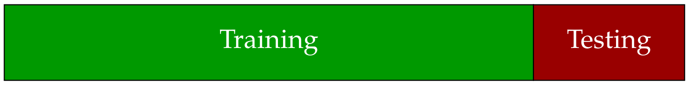
1. Use the training data to build the regression model.

2. Use the testing data to evaluate how well the regression is doing.

    - We quantify the performance through the mean square error:

$$MS_E = \frac{1}{n-2}\sum\left(y_i^{test} - \hat{y}_i^{test}\right)^2.$$

Visually, this is the traditional 80-20 split:



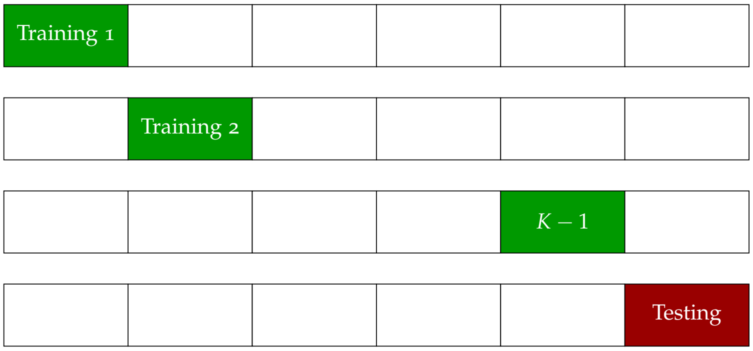
So, what is K-fold validation? K-fold validation involves splitting the data into K parts, typically of equal size. $K-1$ of them are used as training data, with 1 part of them being used as testing data. Then, we:

1. Use the training data to create $K-1$ regression models, one for each of the parts.

2. Use the testing data to test how well **each** of the regression models are performing. Again, you may use the MS_E as defined earlier.

3. Select and return the best model amongst them.

Or, visually:


Training 1
Training 2
$K-1$
Testing