Multicollinearity and Validation

CIVL 7012/8012

In Today’s Class

- Recap
- Multicollinearity
- Model Validation

MULTICOLLINEARITY

1. Perfect Multicollinearity
2. Consequences of Perfect Multicollinearity
3. Imperfect Multicollinearity
4. Consequences of Imperfect Multicollinearity
5. Detecting Multicollinearity
6. Resolving Multicollinearity

Multicollinearity

When the explanatory variables are very highly correlated with each other (correlation coefficients either very close to 1 or to -1) then the problem of multicollinearity occurs.

Perfect Multicollinearity

- When there is a perfect linear relationship.
- Assume we have the following model:

$$Y=\beta_1+\beta_2X_2+ \beta_3X_3+e$$

where the sample values for X_2 and X_3 are:

X_2	1	2	3	4	5	6
X_3	2	4	6	8	10	12

Perfect Multicollinearity

- We observe that $X_3=2X_2$
- Therefore, although it seems that there are two explanatory variables in fact it is only one.
- This is because X_2 is an exact linear function of X_3 or because X_2 and X_3 are perfectly collinear.

Perfect Multicollinearity

When this occurs then the equation:

$$\delta_1 X_1 + \delta_2 X_2 = 0$$

can be satisfied for non-zero values of both δ_1 and δ_2.

In our case we have that

$$(-2)X_1 + (1)X_2 = 0$$

So $\delta_1 = -2$ and $\delta_2 = 1$.

Consequences of Perfect Multicollinearity

- Under Perfect Multicollinearity, the OLS estimators simply **do not exist**. (*prove on board*)
- If you try to estimate an equation in Eviews and your equation specifications suffers from perfect multicollinearity Eviews will not give you results but will give you an error message mentioning multicollinearity in it.

Imperfect Multicollinearity

- Imperfect multicollinearity (or near multicollinearity) exists when the explanatory variables in an equation are correlated, but this correlation is **less than** perfect.
- This can be expressed as:

$$X_3 = X_2 + v$$

where *v* is a random variable that can be viewed as the 'error' in the exact linear releationship.

Consequences of Imperfect Multicollinearity

- In cases of imperfect multicollinearity the OLS estimators can be obtained and they are also BLUE.
- However, although linear unbiassed estimators with the minimum variance property to hold, the OLS variances are often larger than those obtained in the absence of multicollinearity.

Detecting Multicollinearity

- The easiest way to measure the extent of multicollinearity is simply to look at the matrix of correlations between the individual variables.
- In cases of more than two explanatory variables we run the auxiliary regressions. If near linear dependency exists, the auxiliary regression will display a small equation standard error, a large R^2 and statistically significant F-value.

Resolving Multicollinearity

- Approaches, such as the ridge regression or the method of principal components. But these usually bring more problems than they solve.

- Some econometricians argue that if the model is otherwise OK, just ignore it. Note that you will always have some degree of multicollinearity, especially in time series data.

Resolving Multicollinearity

- The easiest ways to “cure” the problems are

(a) drop one of the collinear variables

(b) transform the highly correlated variables into a ratio

(c) go out and collect more data e.g.

(d) a longer run of data

(e) switch to a higher frequency

Multiple Regression Analysis: Estimation

Linear relationships among the independent variables

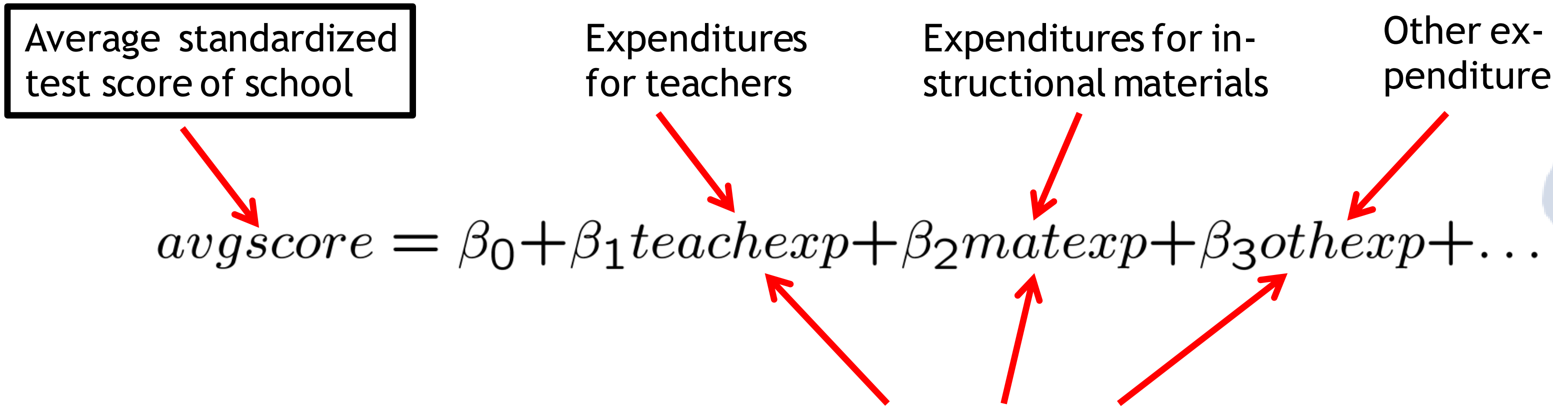
Regress x_j on all other independent variables (including a constant)

The R-squared of this regression will be the higher the better x_j can be linearly explained by the other independent variables

- Sampling variance of $\widehat{\beta}_j$ will be the higher the better explanatory variable x_j can be linearly explained by other independent variables
- The problem of almost linearly dependent explanatory variables is called <u>multicollinearity</u> (i.e. $R_j \rightarrow 1$ for some j)

Multiple Regression Analysis: Estimation

- **An example for multicollinearity**



Average standardized test score of school

Expenditures for teachers

Expenditures for instructional materials

Other expenditures

$$avgscore = \beta_0 + \beta_1 teachexp + \beta_2 matexp + \beta_3 othexp + \ldots$$

The different expenditure categories will be strongly correlated because if a school has a lot of resources it will spend a lot on everything.

It will be hard to estimate the differential effects of different expenditure categories because all expenditures are either high or low. For precise estimates of the differential effects, one would need information about situations where expenditure categories change differentially.

As a consequence, sampling variance of the estimated effects will be large.

Multiple Regression Analysis: Estimation

- **Discussion of the multicollinearity problem**
  - In the above example, it would probably be better to lump all expen-diture categories together because effects cannot be disentangled
  - In other cases, dropping some independent variables may reduce multicollinearity (but this may lead to omitted variable bias)
  - Only the sampling variance of the variables involved in multicollinearity will be inflated; the estimates of other effects may be very precise
  - Note that multicollinearity is not a violation of MLR.3 in the strict sense
  - Multicollinearity may be detected through “variance inflation factors“

$$VIF_j = 1/(1 - R_j^2)$$

As an (arbitrary) rule of thumb, the variance inflation factor should not be larger than 10

Resolving Multicollinearity

- The easiest ways to "cure" the problems are

(a) drop one of the collinear variables

(b) transform the highly correlated variables into a ratio

(c) go out and collect more data e.g.

(d) a longer run of data

(e) switch to a higher frequency

Validation

- *Cross-Validation*
  - Used to estimate test set prediction error rates associated with a given model to evaluate its performance, or to select the appropriate level of model flexibility.
- *Bootstrap*
  - Used most commonly to provide a measure of accuracy of a parameter estimate or of a given method.

Validation

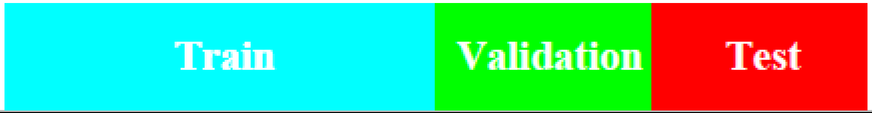
- *Test Error*
  - The average error that results from using a machine learning method to predict the response on a new observation.
  - The prediction error over an independent test sample.
- *Training Error*
  - The average loss over the training sample:
- **Note:** The training error rate can dramatically *underestimate* the test error rate

Model Assessment

- If we are in a data-rich situation, the best approach for both *model selection* and *model assessment* is to randomly divide the dataset into three parts: training set, validation set, and test set.

- The *training set* is used to fit the models. The *validation set* is used to estimate prediction error for model selection. The *test set* is used for assessment of the prediction error of the final chosen model.

- A typical split might by 50% for training, and 25% each for validation and testing.



Train

Validation

Test

Model Assessment (cont.)

- **Best solution:** use a large designated test set, sometimes not available. For the methods presented here, there is insufficient data to split it into three parts.

- Here, we consider cross-validation (CV) methods that estimate the test error by *holding out* a subset of the training observations from the fitting process, and then applying the regression method to those held out observations.

Overview: Model Selection

- By far, the most important use of validation is for *model selection*, which we will discuss in greater detail next week.

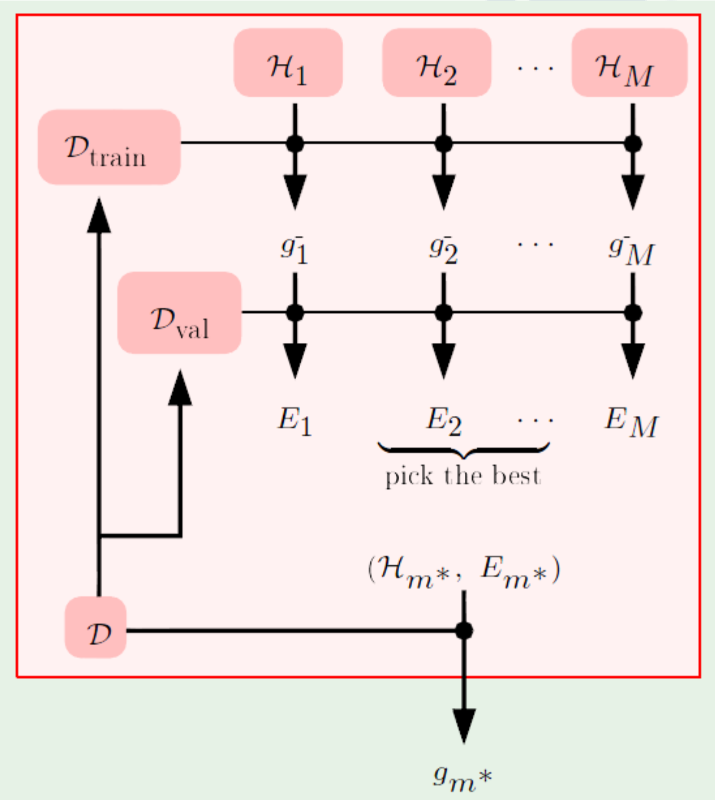
- This could be the choice between a linear model and a nonlinear model, the choice of the order of polynomial in a model, the choice of a regularization parameter, or any other choice that affects the learning process.

- In almost every learning situation, there are some choices to be made and we need a principled way of making these choices.

- The leap is to realize that *validation* can be used to estimate the out-of-sample error for more than one model.

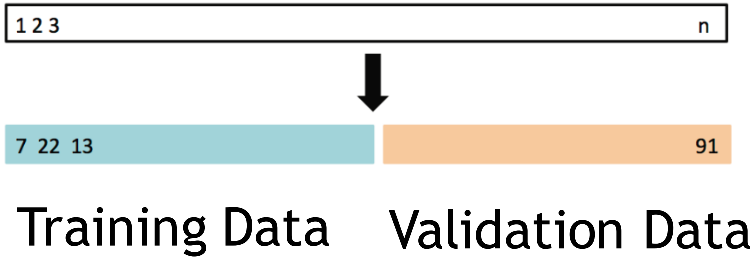
Overview: Model Selection (cont.)

- Suppose we have *M* models; validation can be used to select one of these models.
- We use the training data to fit the model, and we evaluate each model on the validation set to obtain the validation errors.
- It is now a simple matter to select the model with the lowest validation error.



Validation Set Approach

- Suppose that we would like to find a set of variables that give the lowest *validation error rate* (an estimate of the *test error rate*).
- If we have a large data set, we can achieve this goal by randomly splitting the data into separate training and validation data sets.
- Then, we use the training data set to build each possible model and select the model that gives the lowest error rate when applied to the validation data set.



Training Data Validation Data

Validation Set Approach: Example

- **Example:** we want to predict *mpg* from *horsepower*
- Two models:
  - mpg ~ horsepower
  - mpg ~ horsepower + horspower^2
- Which model gives a better fit?
  - We randomly split (50/50) *392 observations* into training and validation data sets, and we fit both models using the training data.
  - Next, we evaluate both models using the validation data set.
  - **Winner** = model with the *lowest* validation (testing) MSE

Measures of error

- MSE

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

- RMSE: sqrt (MSE)
- Mean absolute deviation
- % Mean absolute deviation

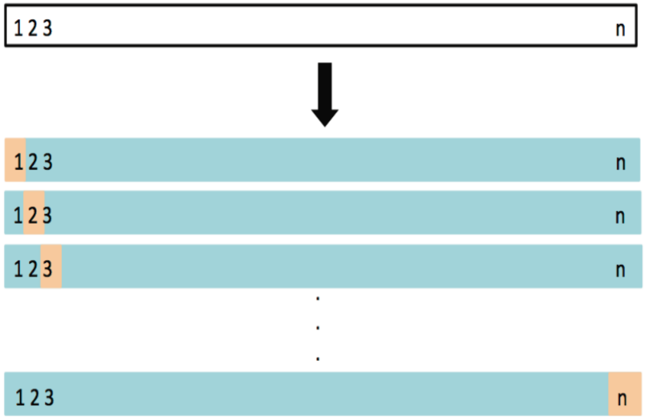
Validation Set Approach: Review

- **Advantages:**
  - Conceptually simple and easy implementation.

- **Drawbacks:**
  - The validation set error rate (MSE) can be highly variable.
  - Only a subset of the observations (those in the training set) are used to fit the model.
  - Machine learning methods tend to perform worse when trained on fewer observations.
  - Thus, the validation set error rate may tend to *overestimate* the test error rate for the model fit on the entire data set.

Leave-One-Out Cross-Validation

- Instead of creating two subsets of comparable size, a single observation is used for the validation set and the remaining observations (*n* – 1) make up the training set.



$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{MSE}_i$$

- **LOOCV Algorithm:**
  - Split the entire data set of size *n* into:
    - Blue = training data set
    - Beige = validation data set
  - Fit the model using the training data set
  - Evaluate the model using validation set and compute the corresponding MSE.
  - Repeat this process *n* times, producing *n* squared errors. The average of these *n* squared errors estimates the test MSE.

Validation Set Approach vs. LOOCV

- LOOCV has far less bias and, therefore, tends not to overestimate the test error rate.

- Performing LOOCV multiple times always yields the same results because there is no randomness in the training/validation set splits.

- LOOCV is computationally intensive because the model has to be fit *n* times.

K-Fold Cross-Validation

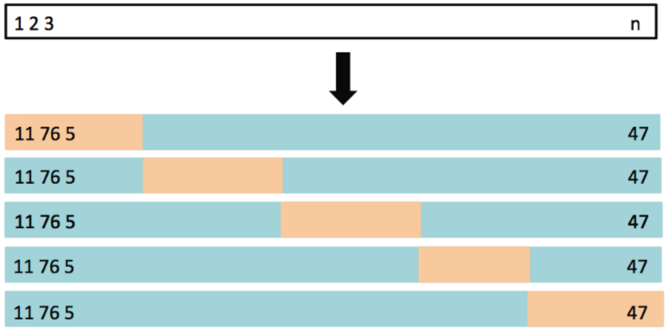
- Probably the simplest and most widely used method for estimating prediction error.

- This method directly estimates the average prediction error when the regression method is applied to an independent test sample.

- Ideally, if we had enough data, we would set aside a validation set (as previously described) and use it to assess the performance of our prediction model.

- To finesse the problem, *K*-fold cross-validation uses part of the available data to fit the model, and a different part to test it.

K-Fold Cross-Validation (cont.)



- We use this method because LOOCV is computationally intensive.

- We randomly divide the data set of into *K* folds (typically *K* = 5 or 10).

- The first fold is treated as a validation set, and the method is fit on the remaining *K* – 1 folds. The MSE is computed on the observations in the *held-out* fold. The process is repeated *K* times, taking out a different part each time.
- By averaging the *K* estimates of the test error, we get an estimated validation (test) error rate for new observations.

K-Fold Cross-Validation (cont.)

- Let the *K* folds be C_1, … , C_K, where C_k denotes the indices of the observations in fold *k*. There are n_k observations in fold *k*: if *N* is a multiple of *K*, then $n_k = n \ / \ K$.

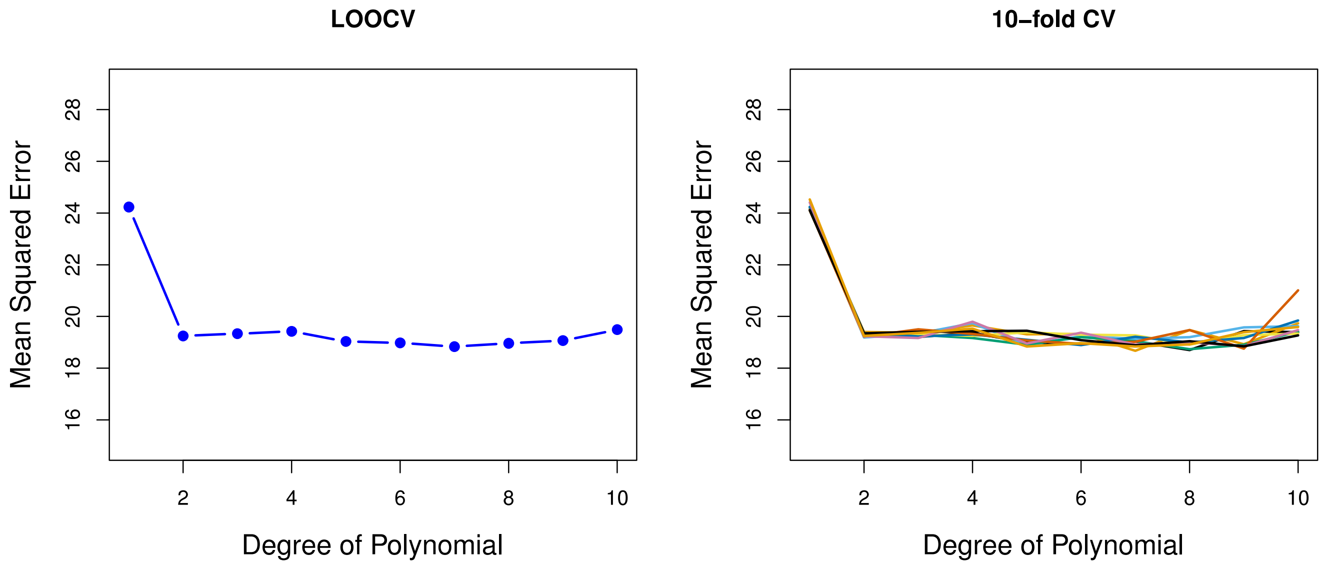
- Compute: $\mathrm{CV}_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \mathrm{MSE}_k$

where $\mathrm{MSE}_k = \frac{1}{n_k} \sum_{i \in C_k} (Y_i - \hat{Y}_i)^2$ and $\hat{Y}_i$ is the fitted value for observation *i*, obtained from the data with fold *k* removed.

1	2	3	4	5
Train	Train	Validation	Train	Train

Note: LOOCV is a special case of *K*-fold, where *K* = *n*

K-Fold Cross-Validation vs. LOOCV



- **Left Panel:** LOOCV Error Curve
- **Right Panel:** *10*-fold CV run nine separate times, each with a different random split of the data into ten parts.
- **Note:** LOOCV is a special case of *K*-fold, where *K* = *n*

Bias-Variance Trade-off for *K*-Fold Cross-Validation

- Which is better, LOOCV or *K*-fold CV?
  - LOOCV is more computationally intensive than *K*-fold CV
  - From the perspective of bias reduction, LOOCV is preferred to *K*-fold CV (when $K < n$)
  - However, LOOCV has higher variance than *K*-fold CV (when $K < n$)
  - Thus, we see the bias-variance trade-off between the two resampling methods

- We tend to use *K*-fold CV with $K = 5$ or $K = 10$, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance

Cross-Validation on Classification Problems

- We will cover classification problems in more detail later in the course, but we briefly show how CV can be used when *Y* is qualitative (categorical) as opposed to quantitative. Here, rather than use MSE to quantify test error, we instead use the number of misclassified observation.

- We use CV as follows:
  - Divide data into *K* folds; hold-out one part and fit using the remaining data (compute error rate on hold-out data); repeat *K* times.
  - CV Error Rate: average over the *K* errors we have computed

AIC and BIC

- AIC: $2k - 2*\ln(L)$
- BIC: $k*\ln(n) - 2*\ln(L)$
- Lower AIC and BIC of models are preferred