# Lecture 6: Evaluation measures and model selection

## Contents



## Last class

We discussed different ways to evaluate the goodness of fit of a LR and we introduced statistical tests to compare nested models (including the full versus the null model).

## Today:

1. Model Evaluation
2. Variable Selection

## Lecture 6 Learning Objectives

- Recognize different metrics to evaluate a LR
- Choose an appropriate measure to evaluate the LR when inference is the primary goal
- Choose an appropriate measure to evaluate the LR when prediction is the primary goal
- Distinguish between measures computed on the training data versus those computed on the test data

- Recognize the limitations of the coefficient of determination (R-square on training set) to evaluate
- Recognize the limitations of the coefficient of determination to compare nested models
- Choose an appropriate measure to select variables of a LR when inference is the primary goal
- Choose an appropriate measure to select variables of a LR when prediction is the primary goal
- Understand the algorithms (best, forward, backward) to automate a variable selection process in LR
- Interpret the results of a model evaluation and a selection process
- Identify LASSO as a variable selection tool

```
library(tidyverse)
library(repr)
options(repr.plot.width=7, repr.plot.height=4)
library(ggplot2)
library(broom)
library(gridExtra)
library(moderndive)
library(latex2exp)
library(leaps)

dat <- read.csv("data/Assessment_2015.csv")
dat <- dat %>% filter(ASSESSCLAS=="Residential")  %>%
        mutate(assess_val = ASSESSMENT / 1000, age=2020-YEAR_BUILT)


# Our sample
set.seed(561)
dat_s <- sample_n(dat, 1000, replace = FALSE)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ─
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.1     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.1
✔ purrr     1.0.2
```

```
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ─
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all con
```

```
Attaching package: 'gridExtra'
```

```
The following object is masked from 'package:dplyr':

    combine
```

***Have we fit a "good" model?***

This is a very general question which we can (and should) break down into more specific questions.

> Intuitively: if the model is good, $\hat{y}_i$ should be close to $y_i$.

# 1.1 Inference vs. Prediction

To assess or evaluate the model there is a more essencial question to answer:

**What is your goal?**

The evaluation metrics are different depending on the goal of the analysis

> The estimation methodologies can also differ although we focus on LS in the course

- **Inference**: your primary goal is to understand the relation between a response variable $Y$ and a set of explanatory variables $X_1, \ldots, X_p$
  - you estimate the LR using a sample to understand how variables are associated (in the population)
  - you use methods from Inference to draw conclusions about the population from the results obtained in the sample

**Examples of inference problems**:

- A real estate agent wants to identify factors that are related to the assessed values of homes (e.g., size of houses, age, amenities, etc)
- Biologists want to verify empirically the central dogma of biology that relates mRNA to protein values

- **Prediction**: your primary goal is to make predictions about the response $Y$, and you are not so concerned about how you get those predictions
    - you estimate the LR using a sample to make predictions of the response for new units (houses, subjects, counties, etc) from the population

**Examples of prediction problems**:

- A real estate agent is interested in determining if a house is under- or over-valued given its characteristics (prediction problem)
- Biologists want to use mRNA data to predict protein values of different genes

# 1.2 Evaluation metrics

Many metrics used to evaluate LR measure how far $Y$ is from $\hat{Y}$:

**Mean Squared Error**: MSE = $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

- *Training MSE*: if we use the original sample $y_1, \ldots, y_n$ and their predicted values (formula above)
- *Test MSE*: if we predict new responses using the estimated LR (similar formula but averaging the squared errors in the test set)

**Note**: we can argue that these are *residuals* and not *errors*. However, many software and articles call this measure the MSE.

**Residuals Sum of Squares**: RSS = $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

- sum of the squared residuals (small is good)
- residuals are measured in the *training* set

**Residuals Standard Error**: RSE = $\sqrt{\frac{1}{n-p}\text{RSS}}$

- estimates the standard deviation of the error term $\varepsilon$ (the RSS is divided by the appropriate degrees of freedom to give a "good" estimate of $\sigma = \sqrt{Var(\varepsilon)}$)
- needed to estimate the standard errors of $\hat{\beta}_j$ in classical theory! (for inference)
- $p$ is the number of estimated parameters including the intercept
- a measure based on *training* data to evaluate the fit of the model (for inference)

- gives an idea of the size of the *irreducible* error, very similar to the RSS, small is good

**Note**: again, *errors* is used to defined this measure that is based on *residuals*.

All the measures listed above are *absolute* measures and it is not easy to judge if these are small enough.

Two important relative measures:

**(i) Coefficient of Determination**, $R^2 = 1 - \frac{RSS}{TSS}$

- it is a common measure of goodness of fit: how well does the model fit the *training* data?
- it doesn't have a known distribution so you can't use it to test a statistical hypothesis
- it also compares $Y$ vs $\hat{Y}$ using the RSS but relative to the TSS (sum of the squared residuals of the null model, no variables, just an intercept)
- the $R^2 = cor(Y, \hat{Y})^2$, if $\hat{Y}$ is a predition obtained from a LR with an intercept estimated by LS
- the $R^2$ increases as we include more variables in the model so it is not useful to compare models of different sizes

# Can we compute the $R^2$ on the test set?

Yes, as we mentioned for the MSE, the $R^2$ can be computed for new responses in a test set $y_{new}$ compared to the predicted values obtained using the trained LR, $\hat{y}_{new}$.

- Some functions compute the $R^2$ from a validation set or using cross validation (perhaps seen in other courses).
- However, note that it is *no longer the coefficient of determination*. It measures the correlation between the true and the predicted responses in a test set.

**Note**: in other courses and articles you may find other measures under the name of $R^2$.

**(ii) The F-statistic**: $F = \frac{(TSS - RSS)/q}{RSS/(n-p)}$

- tests if there is at least one of the predictors $X_1, X_2, \ldots, X_q$ useful in predicting the response, $H_0 : \beta_1 = \beta_2 = \ldots = \beta_q = 0$

- compares a full model ($p$ parameters, including the intercept) versus a null (intercept-only) model
- it also compares $Y$ vs $\hat{Y}$ from the *training* set using the RSS
- the form of the statistic is such that (asymptotically) it follows an $F$ distribution

**Class Discussion: Think, Pair, Share**

Take a moment to compare and contrast the $F$-statistic and $R^2$. What is similar about them? What are the differences between them?

- The $F$ statistic and the $R^2$ both depend on the RSS and the TSS so there is a formula that relates them.
- However, the former has a known $F$ distribution (under certain assumptions) so we can use it to make probabilistic statements.

## Conclusion for Part 1

It is important that you understand:

- what you are computing
- which set you are using (training vs test)
- what is the purpose of the model

# Do we need all the available predictors in the model?

Some datasets contain *many* variables but not all are relevant and you may want to identify the *most relevant* variables to build a model. But again:

## What is your goal? (*Inference vs. Prediction*)

To decide if a variable (or set of variables) is relevant or not we need to choose an evaluation metric

As we have just discussed, the evaluation metric used depends on the goal of the analysis!

# 2.1 Inference

Do all the predictors help to explain the response, or is only a subset of the predictors useful?

Here are some ways we can attempt to answer this question:

- The $F$-test: to compare and test nested models

> `anova`

- The $t$- test: to test the contribution of individual variables to explain the response. Thus, we can use these tests to evaluate variables one at a time

> `lm` and `tidy`

*Caution*: if there are many variables in the model (i.e., $p$ is large) using individual $t$-tests may result in many false discoveries (i.e., reject a true $H_0$, by chance)

- The $R^2$ (or the RSS):

The RSS decreases as more variables are included in the model!

> You can use the $R^2$ or the $RSS$ to compare models of *equal sizes*

- The adjusted $R^2$:

To overcome this problem the $R^2$ is penalized by the number of variables in the model ($p$):

$$\text{adjusted } R^2 = 1 - \frac{RSS/(n-p)}{TSS/(n-1)}$$

- Note that the RSS is penalized by the degrees of freedom.

- You can use the adjusted $R^2$ to compare models of *different sizes* (not necessarily nested)

Caution note: the *training* set is used (over an over) to select so it can't be used again to assess the final significance of the model.

> This problem is known as the *post-inference* problem

**Question**

The coefficient of determination, $R^2 = 1 - RSS/TSS$ can be used to compare the fit of two nested models. True or False?

A. TRUE

B. FALSE

# Example:

- Suppose we compare a model with 2 explanatory variables with another model with 3 additional variables, using $n = 100$ observations.
- The RSS of the larger model (RSS5) are smaller than those of the reduced models (RSS2).
- The adjusted $R^2$ divides the smaller RSS5 by 94 (100-5-1) and larger RSS2 by 97 (100-2-1), so it is not obvious anymore which of the adjusted ones will be smaller.
- It can go either way depending on how relevance of the additional variables to remarkably decrease the residuals.

```
fit2 <-  lm(assess_val ~ age + BLDG_METRE , data=dat_s)

fit5 <-  lm(assess_val ~ age + BLDG_METRE + FIREPLACE + GARAGE + BASEMENT , da


glance(fit2)
glance(fit5)
```

A tibble: 1 × 12

| r.squared | adj.r.squared | sigma | statistic | p.value | df | logLik | |
|---|---|---|---|---|---|---|---|
| <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <d |
| 0.6690865 | 0.6684227 | 108.674 | 1007.936 | 3.781472e-240 | 2 | -6105.789 | 1221 |

A tibble: 1 × 12

| r.squared | adj.r.squared | sigma | statistic | p.value | df | logLik | A |
|---|---|---|---|---|---|---|---|
| <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dl |
| 0.678893 | 0.6772777 | 107.2131 | 420.3082 | 2.991024e-242 | 5 | -6090.748 | 1219 |

# 2.2 Prediction

Do all the predictors help to *predict* the response of new *test* samples? or is only a subset of the predictors useful?

As we discussed before, we need a different metric to evaluate, the $F$-test can not be used to compare *out-of-sample* predictions from different models!

- The test MSE

> if an independent test set is not available, we can use cross-validation to generate *test* sets

## Caution note: the set that has been used to *select* can not be used to evaluate the prediction of the selected model

In the past, a cross-validation study was not always feasible, in particular for large datasets. Thus, different measures were proposed to "approximate" the test MSE.

- Estimates of the test MSE

The **Mallow's Cp, Akaike information criterion (AIC) and Bayesian information criterion(BIC)** add different penalties to the training RSS to adjust for the fact that the training error tends to underestimate the test error

> the penalty increases as the number of predictors in the model increases

> these quantities can be computed for models more complex than LR (more in Regression II)

## An automated proceedure

When we don't have any idea about which variables should be included in the model. Ideally, you want to select the best model out of *all possible models* of all possible sizes.

> For example: if the dataset has 2 explanatory variables $X_1$ and $X_2$, there are 4 models to compare: (1) an intercept-only model, (2) a model with only $X_1$, (3) a model with only $X_2$, and (4) a model with both $X_1$ and $X_2$.

However, the number of *all possible* models become too large rapidely, even for small subset of variables

> there are a total of $2^p$ models from a set of $p$ variables

> if $p = 20$ (20 available explanatory variables) we need to evaluate more than a million models!

There are methods to search more efficiently for a good model (although it may not find the "best" one out of all possible):

- **Forward selection**: Image from ISLR

**Algorithm 6.2** *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.

1. Start with the intercept only model: $y_i = \beta_0 + \varepsilon_i$

- Remember that $\hat{\beta}_0 = \bar{y}$ from the training samples, so $\hat{y}_0 = \bar{y}$ for any observation (from the training or the test set)

2. Select the best model of each size (without counting the intercept)

**Size 1** Evaluate all models of size 1, choose the best model of size 1 (based on RSS, equal size models), call it $\mathcal{M}_1$.

⋮

**Size 2** Starting with the best size 1 model, add 1 variable and evaluate all the resulting models of size 2. Choose the best model of size 2 (based on RSS), call it $\mathcal{M}_2$ (note *all* these models share 1 variable selected in previous step)

. . . continue until you reach the full model

**Size p** there's only one full model, call it $\mathcal{M}_p$.

Note that we can stop this iteration earlier if we want a model of a predetermined size

3. Now we have to select the best out of $p$ models: $\mathcal{M}_1$ (the best model of size 1), $\mathcal{M}_2$ (the best model of size 2 given the selection in previous step), . . . , $\mathcal{M}_p$ (the full model of size $p$)

> You can't use the RSS to compare models of different sizes, but you can use a cross-validation or validation MSE, the $C_p$ (proportional to AIC), the BIC, or the adjusted $R^2$ depending on the data available and the goal of the study.

Other selection procedures include:

- **Backward selection**: start with the full model and remove variables, one at a time
- **Hybrid selection**: after adding a variable, the method may also remove variables

# Example

- Forward selection with the assessment value data via the `regsubsets()` function in the package `leaps`
- Selection Criteria: Mallows $C_p$
- Note: The best model is the one that shows the smallest $C_p$

```
# Split data into train/test sets

set.seed(561)

dat_train <- sample_n(dat_s, size=nrow(dat_s)*0.75, replace=FALSE)
dat_test <- anti_join(dat_s, dat_train, by="the_geom")
```

```
# note: you could use lm(y~.) notation if you want to consider all possible ex

dat_forward <- regsubsets(
assess_val ~ age + BLDG_METRE + FIREPLACE + GARAGE + BASEMENT,
data = dat_train, nvmax = 5, method = "forward",
)

fwd_summary <- summary(dat_forward)

fwd_summary

fwd_summary$cp
```

```
Subset selection object
Call: regsubsets.formula(assess_val ~ age + BLDG_METRE + FIREPLACE +
    GARAGE + BASEMENT, data = dat_train, nvmax = 5, method = "forward",
    )
5 Variables  (and intercept)
           Forced in Forced out
age            FALSE      FALSE
BLDG_METRE     FALSE      FALSE
FIREPLACEY     FALSE      FALSE
GARAGEY        FALSE      FALSE
BASEMENTY      FALSE      FALSE
1 subsets of each size up to 5
Selection Algorithm: forward
         age BLDG_METRE FIREPLACEY GARAGEY BASEMENTY
1  ( 1 ) " " "*"        " "        " "     " "
2  ( 1 ) "*" "*"        " "        " "     " "
3  ( 1 ) "*" "*"        " "        " "     "*"
4  ( 1 ) "*" "*"        " "        "*"     "*"
5  ( 1 ) "*" "*"        "*"        "*"     "*"
```

23.3035072471821 · 12.8565291757996 · 5.86568771125292 · 4.70146960044644 · 6

```
fwd_summary_table <- data.frame(
    Model = 1:5,
    RSQ = fwd_summary$rsq,
    ADJ.RSQ = fwd_summary$adjr2,
    RSS = fwd_summary$rss,
    Cp = fwd_summary$cp,
    BIC = fwd_summary$bic
    )

fwd_summary_table
```

A data.frame: 5 × 6

| Model | RSQ | ADJ.RSQ | RSS | Cp | BIC |
|---|---|---|---|---|---|
| <int> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> |
| 1 | 0.6906317 | 0.6902181 | 7144086 | 23.303507 | -866.6769 |
| 2 | 0.6956371 | 0.6948222 | 7028498 | 12.856529 | -872.2907 |
| 3 | 0.6992527 | 0.6980433 | 6945005 | 5.865688 | -874.6334 |
| 4 | 0.7005252 | 0.6989173 | 6915621 | 4.701470 | -871.1933 |
| 5 | 0.7008073 | 0.6987966 | 6909107 | 6.000000 | -865.2800 |

```
coef(dat_forward, 4)
coef(dat_forward, 5)
```

**(Intercept):** 50.9525743083722 **age:** -0.680575404742103 **BLDG_METRE:** 2.44501212626386 **GARAGEY:** 25.7743017375825 **BASEMENTY:** 64.6837216663365

**(Intercept):** 45.5291054528736 **age:** -0.638775834961519 **BLDG_METRE:** 2.4332424185864 **FIREPLACEY:** 8.28400506276642 **GARAGEY:** 24.3276191625825 **BASEMENTY:** 65.1444860050404

However, a word of caution regarding stepwise model selection: although widely discussed, these approaches have many potential issues such as multiple comparison problems resulting in $p$-values that are too low. See page 68 of Regression Modelling Strategies for a more detailed list of problems associated with stepwise model seleciton.
Regularization offers an alternative way to select a model by penalizing the RSS.

- For example: ridge, lasso, elastic net
- These are different *shrinkage estimation methods* that shrink coefficient estimates towards zero
- Alternatives to variable selection with $p > n$ and it is less time consuming

LASSO (also called L1-regularization) is a very popular approach for variable selection. The model minimizes

$$\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j))^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

where $\lambda$ is the penalty term. Note that when $\lambda = 0$ we are left with a least squares regression model. In R, the `glmnet` package is commonly used to perform LASSO regression.

# In-Class Exercise

Using the teaching score data from UT Austin from last class, perform backwards model selection to select the "best" model based on Mallow's $C_p$. Think about whether we can/should assess the interaction model using `regsubsets()`.