

# Machine Learning for Social Sciences

## Part 1: Regularisation

---

Felix Hagemeister

TUM School of Social Sciences and Technology

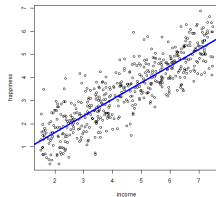
April 2022

# Regression

Linear regression minimizes the in-sample sum of squared residuals (“deviance”).

That is, it finds a  $\hat{\beta}$  that maximizes in-sample  $R^2$ .

$$\text{dev}_{IS}(\hat{\beta}) \propto \sum_{i=1}^n (y_i - X_i' \hat{\beta})^2 \quad (1)$$



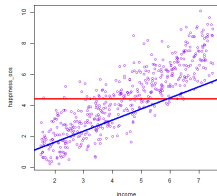
# Regression

All that matters for prediction is the out-of-sample (OOS) deviance.

For OOS  $R^2$ ,  $\hat{\beta}$  is still the same (still fit with observations 1...n), but deviance is now calculated over new observations:

$$\text{dev}_{\text{OOS}}(\hat{\beta}) \propto \sum_{i=n+1}^{n+m} (y_i - X_i' \hat{\beta})^2 \quad (2)$$

OOS  $R^2$  will be positive if it performs better than the null model (simple average).



# Exercise 1

How do we assess ou-of-sample (OOS) fit?

Let's cover different types of cross validation using the code [here](#) based on [this](#) article.

# K-Fold Out-Of-Sample (OOS) Cross Validation (CV)

Given a dataset of  $n$  observations,  $\{[X_i, y_i]\}_{i=1}^n$  :

- Split the data into  $K$  evenly random subsets (*folds*).
- For  $k = 1 \dots K$ :
  - Fit the coefficients  $\hat{\beta}$  using all but the  $k$ th fold of data
  - Record  $R^2$  on the left-out  $k$ th fold.

This will yield a sample of  $K$  OOS  $R^2$  values. This sample is an *estimate* of the distribution of your model's predictive performance on new data.

Information criteria (AIC, BIC, AICc) are analytic approximations for these estimates. However, CV is the better choice.

## Exercise 2

Let's try to predict survival of people on board of the titanic.

- The data [titanic\\_sample.csv](#) is a sample of people on board and provides some (historically true) individual information.
- The R code [here](#) predicts survival based on characteristics with linear and logistic regression, and perform k-fold OOS validation. Note that it uses manual coding for many steps to improve understanding (we will later use wrapper packages for many of these).

## Poor prediction properties of OLS

- **Overfitting:** analysis corresponds too closely to data
- **Multiplicity:** false discovery rate (FDR)<sup>1</sup> might be high if small share of covariates is irrelevant
- **Added noise:** too many irrelevant controls reduce quality of relevant predictors

---

1. FDR = expectation of false positives among significant tests.

How do we decide which model to build?

- Regression setting with  $p$  potential covariates have  $2^p$  different possible models (with 20 covariates already  $> 1\text{M}$ ).

**Regularization:** Penalizing model complexity to come up with promising candidate models.

Approaches to be avoided:

- *Backward stepwise regression:* Looking at full model fit and cutting down, e.g. using p-values with Benjamin Hochberg algorithm.
- *Forward stepwise regression:* Adding covariates stepwise that most increase OOS  $R^2$ .

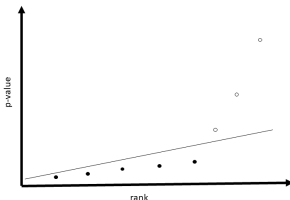


# Controlling expected false discovery rate (FDR)

## Benjamin-Hochberg (BH) FDR control algorithm:

For  $N$  tests, with p-values  $p_1 \dots p_n$  and target FDR  $q$ :

- Order your p-values from smallest to largest as  $p_1 \dots p_n$ .
- Set the p-value cutoff as  $p^* = q \frac{k}{N}$
- Select those features as significant with p-values above  $p^*$



Minimize a *penalized* deviance:

- **OLS:**  $L_{OLS}(\hat{\beta}) = \sum_{i=1}^n (y_i - x_i' \hat{\beta})^2$
- **Ridge:**  $L_{Ridge}(\hat{\beta}) = \sum_{i=1}^n (y_i - x_i' \hat{\beta})^2 + \lambda \sum_{j=1}^m \hat{\beta}_j^2$
- **Lasso:**  $L_{Lasso}(\hat{\beta}) = \sum_{i=1}^n (y_i - x_i' \hat{\beta})^2 + \lambda \sum_{j=1}^m |\hat{\beta}_j|$
- **Elastic Net:**  $L_{ElasticNet}(\hat{\beta}) = \frac{\sum_{i=1}^n (y_i - x_i' \hat{\beta})^2}{2n} + \lambda \left( \frac{1-\alpha}{2} \sum_{j=1}^m \hat{\beta}_j^2 + \alpha \sum_{j=1}^m |\hat{\beta}_j| \right)$

where

- $\alpha$  is mixing parameter between ridge and lasso.
- $\lambda$  is the penalty strength and a *tuning parameter*

**Lasso** is a fantastic default because

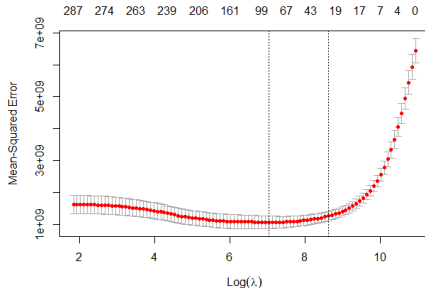
- it gives least possible amount of bias while preserving stability
- it yields automatic variable screening (some of the solved  $\hat{\beta}$  are exactly zero)

**Ridge** penalty  $\hat{\beta}_j^2$  places heavy penalty on large vales of  $\beta$ . Use only if you think that all covariates have small effects and there are no big dominating effects.

# Regularisation

**Lasso** alone cannot do model selection, but *enumerates* a number of possible candidate models for different  $\lambda$ .

- Use k-fold cross validation for each candidate model.
- Choose the model with the smallest OOS sum of squared residuals (“mean squared prediction error”).



## Practical Hints

- Factor reference level matters under penalisation: get rid of reference level and create separate dummies for each factor level
- Size of coefficient matters: standardize covariates or standardise  $\beta$ s in the cost function by multiplying coefficients with standard deviation of corresponding covariates
- Note that you might still want to avoid standardisation if you have indicator covariates (penalty would mechanically be higher on common categories)

## Dealing with missing data

- **Categorical variables:** Treat “missing” as separate category: use *naref()* function provided [here](#)
- **Numerical variables:** Replace depending on sparsity: If variable has many zeros, replace with zero. Otherwise replace with mean. Use *mzimpute()* function provided [here](#)

# Bias-Variance Trade-Off

Consider a model  $y_i = f(x_i) + \epsilon_i$

where

- $\text{Var}(\epsilon_i) = \sigma^2$
- $E(\epsilon_i) = 0$
- $\epsilon_i$  independent across  $i$
- $\epsilon_i$  and  $x_i$  independent

Suppose we fit a mapping  $\hat{f}$  from sample  $D$  and use it to predict a value for  $y_0$  at some  $x_0$

$$\text{Mean Square Error of Prediction} = E[(y_0 - \hat{f}(x_0))^2]$$

$$\text{MSE} = \sigma^2 + \text{Var}(\hat{f}(x_0)) + \text{Bias}(\hat{f}(x_0))^2$$

- Adding relevant variables to a regression reduces bias.
  - Adding any variable to a regression increases variance of each estimated coefficient.
- with many covariates, OLS is unbiased but has high variance
- idea behind regularisation is to introduce (small) bias into coefficient estimates and to reduce variance.



## Exercise 3

Let's use some meta-packages to implement lasso and elastic net:

- [Here](#) is a coding example using *glmnet*.
- [Here](#) is a coding example using *caret*.
- [Here](#) is a coding example for elastic net, also using *caret*.

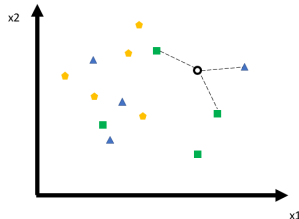
**Classification:** Predicting response variable  $y$  that represents membership in one of many categories.

Let's cover

- K Nearest Neighbours
- Logistic regression (again)
- Performance metrics
- Distributed Multinomial Regression

# Classification

**K nearest neighbours:** predicted class is most common class in the set of  $k$  nearest neighbours.



Good idea for intuition, but too crude to be useful in practice.

**Logistic Regression:** Estimate probability  $p$  for binary response variable to be 1.

A *classification rule*, or cutoff, is the probability  $p$  at which you predict

- $\hat{y}_f = 0$  for  $p_j \leq p$
- $\hat{y}_f = 1$  for  $p_j > p$ .

Such a rule involves two types of errors (false positive, and false negatives), which can be converted into rates. Note that these statistics are normalized by classification.

- **False Positive Rate** =  $\frac{\text{expected \# false positives}}{\text{\#classified positive}}$
- **False Negative Rate** =  $\frac{\text{expected \# false negatives}}{\text{\#classified negative}}$

Measures of classification performance:

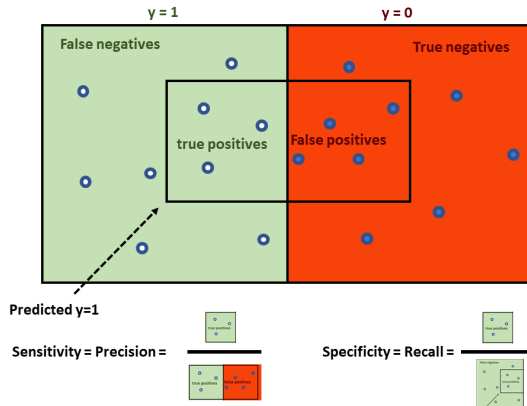
- **Accuracy** =  $(TP+TN)/(TP+FP+FN+TN)$
- **Precision** =  $TP/(TP+FP)$
- **Sensitivity** = **Recall** =  $TP/(TP+FN)$
- **Specificity** =  $TN/(TN+FP)$

where

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

# Classification

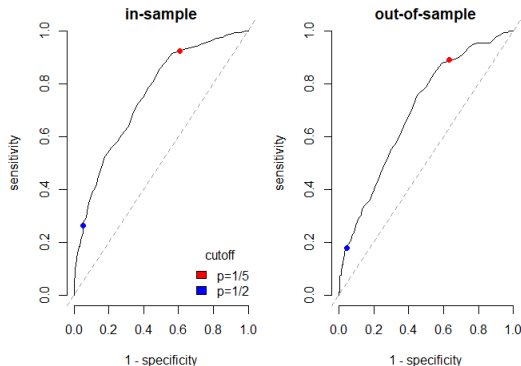
The **F1 Score** combines the precision and recall of a classifier into a single metric by taking their harmonic mean: 
$$F1 \text{ Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$



# Classification

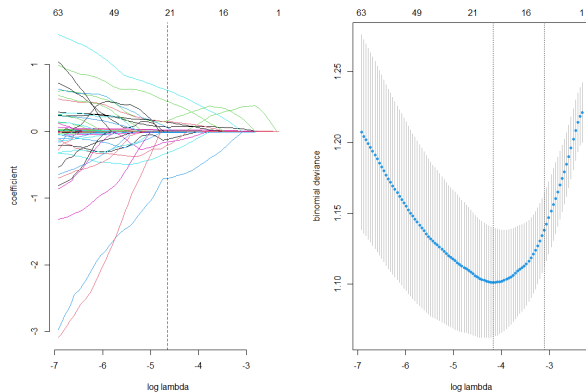
A nice visual summary of potential classification rules is the ROC curve that plots sensitivity against  $1 - \text{specificity}$ .<sup>2</sup>

The area under the curve of the out-sample ROC plot is often used as performance measure for a classification model.



# Classification

You can plot the regularisation path and CV results:





## Exercise 4

Let's build a model to predict default on loans using a real dataset on loans and credit from a set of local lenders in Germany.

- The data can be downloaded [here](#).
- [Here](#) is a coding example for classification, using Matt Taddy's *gamlr* package.

What if our outcome is not binary, but has multiple classes (one of  $K$  categories)?

- We can use *multinomial logistic regression*.
- But multinomial regressions can be slow...

## **Solution: Distributed multinomial regression (DML)**

- Trick: multinomial logistic regression coefficients will be – for all practical purposes – similar to those we can get through *independent* poisson estimation for each of the log-linear equations  $E[y_{ik}|\mathbf{x}_i] = \exp(\mathbf{x}_i'\beta_k)$ .

Let's try to predict glass type from glass features.

- [This](#) coding example implements multinomial logistic regression with parallel computing.

# References

---

- Bansak, Kirk, Jeremy Ferwerda, Jens Hainmueller, Andrea Dillon, Dominik Hangartner, Duncan Lawrence, and Jeremy Weinstein. 2018. "Improving refugee integration through data-driven algorithmic assignment." *Science* 359 (6373): 325–329.
- Deryugina, Tatyana, Garth Heutel, Nolan H. Miller, David Molitor, and Julian Reif. 2019. "The Mortality and Medical Costs of Air Pollution: Evidence from Changes in Wind Direction." *American Economic Review* 109, no. 12 (December): 4178–4219.
- Kavanagh, Nolan M., Anil Menon, and Justin E. Heinze. 2021. "Does Health Vulnerability Predict Voting for Right-Wing Populist Parties in Europe?" *American Political Science Review* 115 (3): 1104–1109.
- Mullainathan, Sendhil, and Ziad Obermeyer. 2021. "Diagnosing Physician Error: A Machine Learning Approach to Low-Value Health Care\*." Qjab046, *The Quarterly Journal of Economics* (December). ISSN: 0033-5533.