Quantitative Text Analysis

Bruno Castanho Silva

Day 4 - Machine Learning

Cologne Center for
Comparative Politics

Any questions from yesterday?

Machine learning

Prediction

- We’re interested in making *predictions!*
- For that, we create models that *learn* from existing data.
- Two main kinds: supervised and unsupervised learning;

Supervised learning

- Is it a sheepdog or a mop in each picture?

Unsupervised learning

- Is there something (meaningful) in all these pictures?

Prediction v. Inference

- Statistical analysis in social sciences is almost always concerned with inference:
  - How does X impact Y?
  - Is there a relation between Z, X, and Y?
- “The coefficients are the object of interest. i.e., the statistical model itself is the object of interest” (Cranmer and Desmarais 2017);

Prediction v. Inference

- Methods under the “machine learning” umbrella can be used for inference, but they are most concerned with prediction;
- Given a vector of values of X for an observation, what’s my best guess about its value in Y?
    - How X1 or X2 are related to Y? Often not relevant, but can be investigated.

The “Learning” in Machine Learning

- We do not start with a theoretically specified model and make adjustments.
- Instead, let algorithms decide, purely based on the data, what combination of parameters and hyperparameters gives the best predictions.
- It **learns** from the data how to make ever more accurate predictions.
    - The more data, the more accurate predictions. Usually.

The Black Box

- This means that, in trying to get more accurate predictions, we often go for less interpretable models;
  - No one knows how IV's are related to the DV in a neural network. But they're still stopping self-driving cars from crashing into one another. Usually.
- That *does not* mean we shouldn't have a basic understanding of what each technique or algorithm is doing.
  - Even if we can't program said algorithm ourselves.

Our models...

- Linear regression is a good way to start thinking about prediction;
- We want to find a model that, applied to the data, yields the most accurate predictions of the outcome variable;
- Models can be more or less flexible.
- In a familiar regression context, we add flexibility by, for example, adding quadratic terms.

More flexible vs. More rigid models

- More flexible:
  - Better accuracy (usually);
  - Less interpretable;
- More rigid:
  - Lower accuracy (usually);
  - More interpretable.

Parameters v. Hyperparameters

- In common regression we use OLS (or MLE) to estimate model parameters. There, they are the coefficients, intercepts, and residuals. They are what we learn from the data.
- Hyperparameters are characteristics **of the model** we run and are decided **before estimation**.

Parameters v. Hyperparameters

- For example: the presence or not of a quadratic term in a regression is a hyperparameter.
- *Given we include it*, the estimated coefficient of the quadratic term is a parameter.
- Hyperparameters are what determines how rigid or flexible a model is.

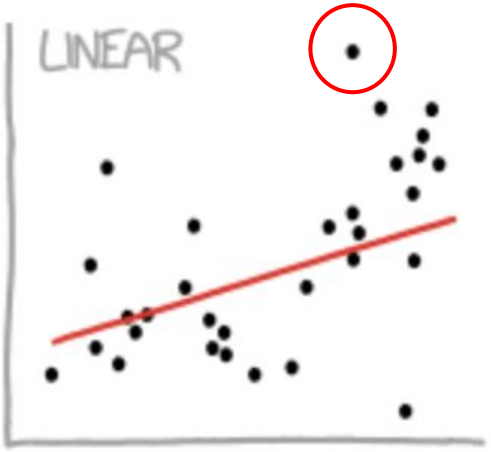
Parameters v. Hyperparameters

- Finding the best set of parameters is called *training*.
  - That’s what the computer does for us.
- Finding the best set of hyperparameters is called *tuning*.
  - That’s what we have to do

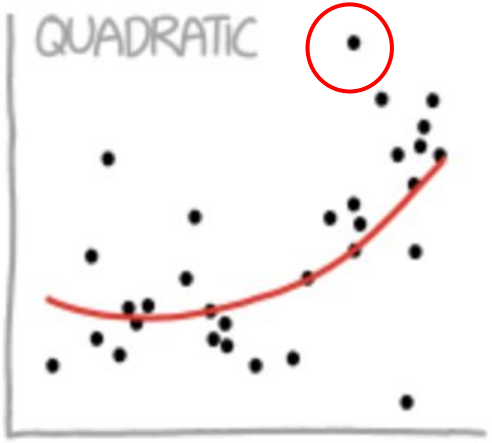
Bias-Variance trade-off

- **Bias** is the error in predictions, in relation to observed values;
- **Variance** is how much the model changes if we train it on different data;

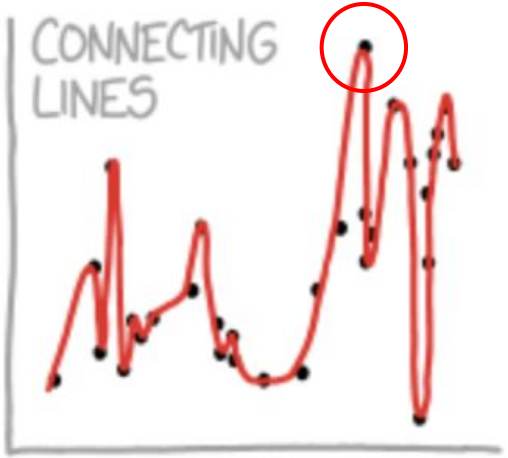
CURVE-FITTING METHODS
AND THE MESSAGES THEY SEND



"HEY, I DID A REGRESSION."



"I WANTED A CURVED LINE, SO I MADE ONE WITH MATH."



"I CLICKED 'SMOOTH LINES' IN EXCEL."

Bias-variance trade-off

- **Bias** is the error in predictions, in relation to observed values;
- **Variance** is how much the model changes if we train it on different data;
- Rigid models tend to have higher bias and low variance;
- Flexible models tend to have lower bias and high variance;
- The goal is finding the combined optimal minimum.

Problems

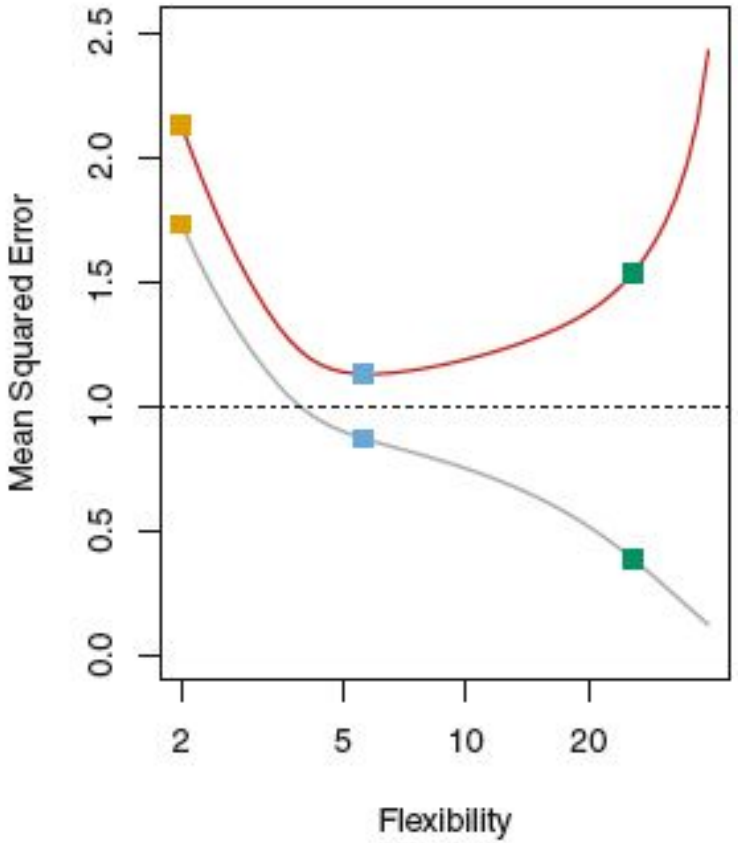
- The most flexible models will almost always give better predictions;
- However, it’s easy to start modeling noise instead of the true relationship.
- That’s called **overfitting!**
- It means that the model works great on data used to train it, but has poor performance on new data.

Training and Test Errors

- First defining feature of ML:
- We’re interested in making predictions about *new* observations.
- Training set: used to estimate (train, tune,...) the model;
- Test set: new observations for which we may not know the value in Y.
    - We apply the model developed on the training set to make predictions about observations on the test set.

Flexibility vs. Rigidity

- Increasing the flexibility will **almost always** reduce the training set error.
- However, eventually it will start **increasing the test set error** because of overfitting;



Validation set approach

- Split the data into training and test sets.
  - No rule-of-thumb on proportions, but rather have more obs on training than testing (say, 80%-20%)
- Tune and train the model using the training set. Once you’re satisfied with its performance, make predictions about the observations on the test set and see how it performed.

How do I select the model from the training data?

- Try out various hyperparameters, and check their error rates.
- Parsimony: look for the most rigid model that gives an acceptable (low) error.

Problems of the validation set approach

- Splitting only once into training and validation set can still lead to bias;
  - All extreme observations/outliers end up in one of the sets...
- If we repeat several iterations of training/testing, soon our estimate of the test set error becomes akin to our training set error. We’re using the test set to train the model.
- Enters **cross validation** (more in the R script later)

Assumption for out-of-sample prediction

- Our data is a random sample from the unknown observations to which we want to predict.

Thanks to machine-learning algorithms,
the robot apocalypse was short-lived.

Lasso/Ridge Regressions

The MSE

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

Regression Penalties

- Two popular methods to improve the predictive power of linear regression:
- Ridge and Lasso, and a combination of the two: elastic-net

Ridge Regression

- Add a penalty term to the RSS - the quantity we try to minimize in a regression.
- Ridge: $\lambda\beta_j^2$. Lambda is the *tuning parameter*. It's chosen through CV;
- Higher lambdas mean that coefficients are shrunk towards zero, because this term gets larger.
- Lambda = 0 returns the exact same result as OLS.

The Ridge penalizer

$$\mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

Lasso

- Slightly different penalizer: $\lambda|\beta_j|$
- This time, increasing lambda leads to some coefficients being shrunk *to* zero.
- The lasso performs variable selection. IV's that have little impact on the outcome are effectively removed (their coefficients become 0).
- The lasso is a good first step for $p > n$ problems - meaning, text analysis

The lasso penalizer

$$\mathrm{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

Pros and Cons

- While they are still in a linear regression framework, both perform well even in non-linear problems;
  - Can be applied with binary or multinomial functions.
- If the number of predictors is very large, lasso is a better option;
- If all predictors are minimally related to the outcome, ridge might be better.
- But, why not unite the two?

The elastic net

- Merges both penalizers using a second tuning parameter: α, so that $0 \leq \alpha \leq 1$.
- Both the ridge and the lasso penalizers are added to the RSS. But one is multiplied by α, and the other by $(1-\alpha)$.
- Therefore, if $\alpha = 1$ or 0, only one is used. But any value in between combines both: both coefficient shrinkage *and* variable reduction.

Elastic Net

- Standard option now for regularized regression;
- Cross-validate both alpha and lambda;
- Suitable for continuous, dichotomous, and multinomial outcomes.
- Works well with sparse matrices and a large number of predictors.
- Interpretable: we get coefficient estimates ordered by substantive significance.

There’s much more options

- Tree-based methods (e.g. Random forests or boosting algorithms) are very powerful and flexible
- All sorts of deep learning models
- Etc etc etc

- Overview Courses 2022
- Brain & Behaviour
- Business & Economics
- Contemporary Discussions
- Education
- Healthcare
- History, Philosophy & Culture
- Language & Communication
- Law & Politics
- Science
- Skills Training
- Social Research Methods
- Social Sciences

Social Research Methods: Introduction to Machine Learning for Social Sciences

Date: 27 June - 1 July 2022

Early bird fee: €518 or €431(deadline 1 April 2022)

Regular fee: €575

Application deadline: 1 May 2022

APPLY

MethodsNET

Course Description

This course is offered as part of the Summer School in Social Research Methods, which is developed and coordinated by MethodsNET in collaboration with the Nijmegen School of Management, Radboud University. MethodsNET is a global methods excellence network that offers world class training in social research methods, through its top instructors from renowned universities worldwide. This course of the Summer School in Social Research Methods has a unique approach: the morning is fully dedicated to the course topics mentioned in the course description. In the afternoon, you can choose to take part in a rich set of extra optional training activities to broaden or deepen your skills and knowledge.