Machine learning 2

Understanding methods, model selection and interpretation

ISRIC Spring School “Hands on Digital Soil Mapping”

Thursday 9-10:30, 31 May 2018

Madlene Nussbaum

For any subsequent questions:

Madlene Nussbaum
madlene.nussbaum@bfh.ch

BFH-HAFL
Bern University of Applied Sciences
School for Agriculture, Forestry and Food Sciences
Länggasse 85
CH-3052 Zollikofen
Switzerland

Objectives ...

- Make sense of **terms** and **concepts** often heard
- Get an **overview** of machine learning (ML) methods and their strategies
- Learn to **apply** at least one further ML technique
- Know at least two ways of **model selection**
- Know two ways of **model interpretation**, move away from **ML = black box**
- Machine learning will not solve all your problems by one click, so **be critical!**

List of references in slides (besides reading list):

Behrens, T., Schmidt, K., Ramirez-Lopez, L., Gallant, J., Zhu, A.-X., and Scholten, T.: Hyper-scale digital soil mapping and soil formation analysis, Geoderma, 213, 578–588, doi:10.1016/j.geoderma.2013.07.031, 2014.

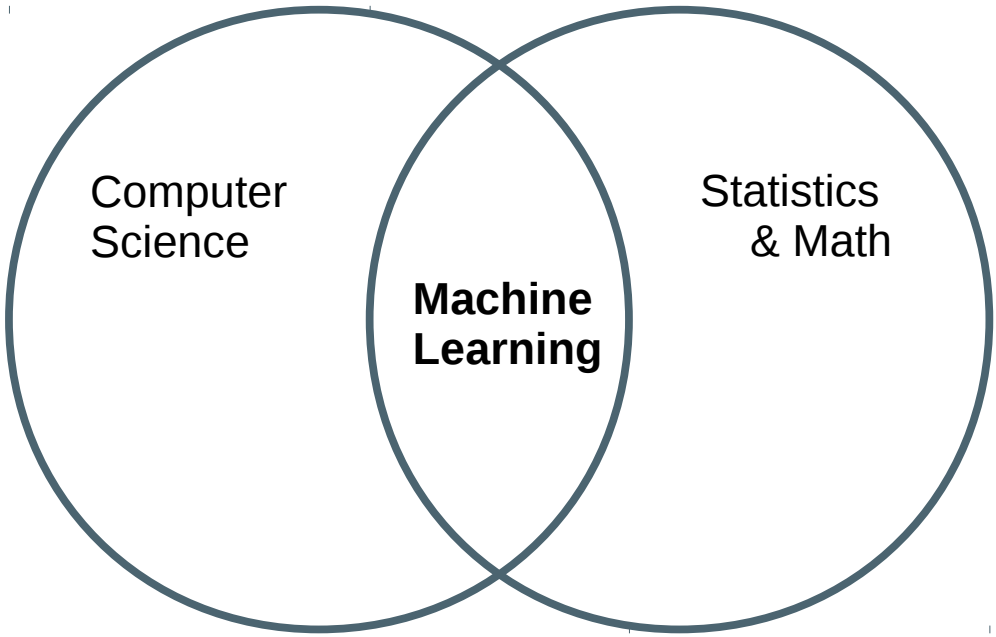
Brungard, C. W., Boettinger, J. L., Duniway, M. C., Wills, S. A., and Edwards Jr., T. C.: Machine learning for predicting soil classes in three semi-arid landscapes, Geoderma, 239–240, 68–83, doi:10.1016/j.geoderma.2014.09.019, 2015.

Nussbaum, M., Spiess, K., Baltensweiler, A., Grob, U., Keller, A., Greiner, L., Schaepman, M. E., and Papritz, A.: Evaluation of digital soil mapping approaches with large sets of environmental covariates, SOIL, 4, 1-22, https://doi.org/10.5194/soil-4-1-2018, 2018.

Content of lecture

- Terms and concepts
- Spatial modelling: requirements?
  - Side note: overfitting
- Overview of ML and their strategies
  - Bootstrap
  - Boosting
  - Model averaging
- Model selection
  - Why model selection?
  - Selection with lasso
  - Selection with random forest
- Model interpretation

Introduction: some terms



→ For digital soil mapping maybe better:
statistical learning or **computational statistics**

Definitions of machine learning are very wide!

They may include artificial intelligence, pattern recognition, even normal linear regression.

Examples:

Machine Learning at its most basic is the practice of using algorithms to parse data, learn from it, and then make a determination or prediction about something in the world. (Nvidia)

Machine learning is the science of getting computers to act without being explicitly programmed. (Stanford)

Machine learning explores the study and construction of algorithms that can learn from and make predictions on data (Kohavi 1998)

Machine learning is a method of data analysis that automates analytical model building (www.sas.com)
.

→ *the last definition suits best for what we intend to use it in this course.*

Introduction: some terms

- **Unsupervised learning**
  No response, just “covariates”
  → e.g. clustering of satellite image by similar values
  → operates “blind”

- **Supervised learning**
  For each covariate value x_i there is also a response value y_i
  → e.g. random forest
  → what we usually do for digital soil mapping

  - Regression: continuous responses, e.g. clay content
  - Classification: categorical responses (binary or multinomial)

Response = output variable = dependent variable = e.g. pH, clay, presence/absence of a diagnostic horizon, drainage classes, soil types

Covariates = input variable = independent variables = features = predicors = e.g. topographic wetness index, slope, substrate classes from geological maps

Huge topic, hence further reading advised:

Gareth et al. 2017, very nice and solid introduction, easy accessible.
Gareth, James, Witten, Daniela, Hastie, Trevor and Tibshirani, Robert. An introduction to statistical learning : With applications in R. 8 edn. New York: Springer, 2017.

Hastie et al. 2009, very good and detailed book, but rather advanced.
Hastie, T., Tibshirani, R., and Friedman, J.: The Elements of Statistical Learning; Data Mining, Inference and Prediction, Springer, New York, 2 edn., 2009. with examples and data in R package ElemStatLearn, https://cran.r-project.org/web/packages/ElemStatLearn/index.html

Kuhn et al. 2013, form the author of the R package caret, focuses a bit more on classification
Kuhn, M., Johnson, K.: Applied predictive modeling, Springer, New York, 2013.

Hothorn, 2018, overview of R packages for ML:
Hothorn, Torsten. CRAN Task View: Machine Learning & Statistical Learning, https://CRAN.R-project.org/view=MachineLearning, 2018.

Possibly further interessting to you:

Tutz, 2012
very good book on modelling categorical responses, mostly parametric methods, also examples on machine learning (but not main focus), R package “catdata” with all examples.
Tutz, G.: Regression for Categorical Data, Cambridge University Press, doi:10.1017/cbo9780511842061, 2012.

Wilks, 2011, Chapter 8.
Validation measures for classification results, also validation of uncertainty, all very well explained. R package “verification”.
Wilks, D. S.: Statistical Methods in the Atmospheric Sciences, Academic Press, 3 edn., 2011.

Davidson and Hinkley, 1997
Extended book on boostrapping methods (a bit dry though), for simulation of confidence or prediction intervals.
Davison, A. C. and Hinkley, D. V.: Bootstrap Methods and Their Applications, Cambridge University Press, Cambridge, doi:10.1017/cbo9780511802843, 1997.

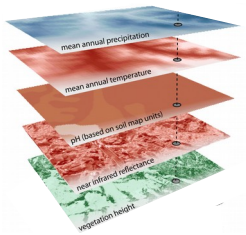
Step back: What do we need for spatial predictions?

My situation …

texture
density
gravel
soil depth
drainage
pH, ECEC
SOC

300-1400
locations with soil properties in

2–4 soil depth
3 study areas



300-500
environmental covariates

48
statistical models

Requirements

A spatial prediction method should …

- model **nonlinear** relations
- consider **spatial** autocorrelation
- model continuous and categorical responses
- handle **numerous** correlated **covariates** without overfitting calibration data
- **automatically** build models with **good predictive power**
- preferably result in **sparse model**
- accurately quantify **accuracy** of **predictions**
- give prediction **uncertainty**

Side note:
Overfitting? Bias-Variance trade-off

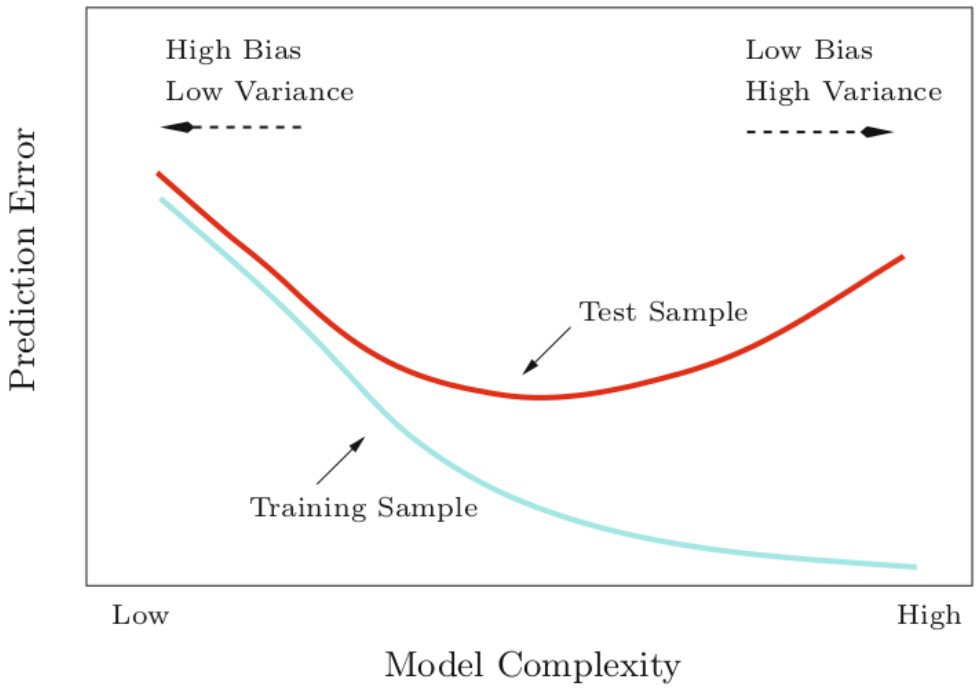
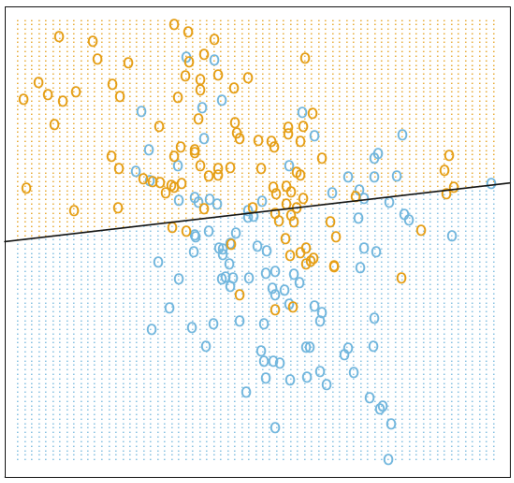


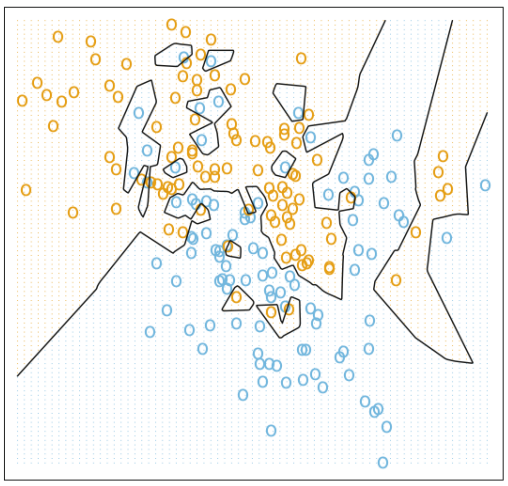
FIGURE 2.11. *Test and training error as a function of model complexity.*

Hastie et al. 2009, p. 38.

Side note:
Overfitting? Bias-Variance trade-off



Linear regression
high bias, but stable



1-nearest neighbours
low bias, high variance

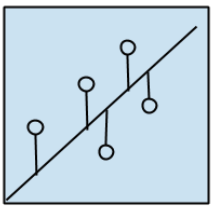
Hastie et al. 2009, Chap. 2.3.

$$\mathrm{E}\left[\left(y-\hat{f}\left(x\right)\right)^2\right]=\mathrm{Bias}\left[\hat{f}\left(x\right)\right]^2+\mathrm{Var}\left[\hat{f}\left(x\right)\right]+\sigma^2$$

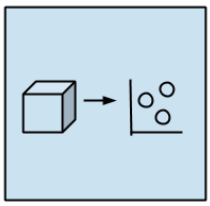
Bias: Erroneous assumptions in the model, relevant relationship are missed, e.g. non-linear dependence of response on covariate (underfitting).

Variance: Sensitivity to small fluctuations in the calibration data. Small change in the data results in big change in the model.
Algorithm models random noise in calibration data, instead of just relevant relationship (overfitting).

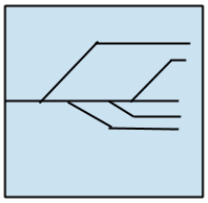
I tried to tidy up ...



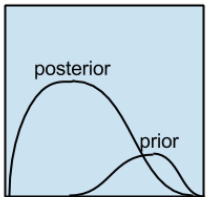
Regression
linear and non-linear models, geostatistics



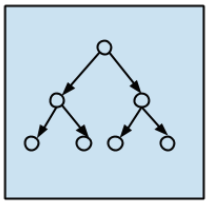
Dimension reduction
PCA, PLS



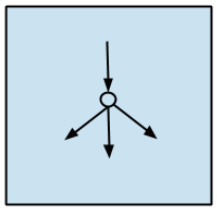
Regularisation Shrinkage
Lasso



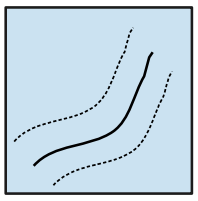
Bayes methods



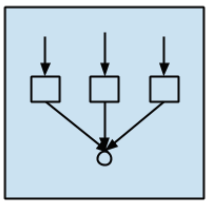
Decision trees
CART



Neuronal networks



Support vector machines
kernel methods



Ensembles
bootstrap, boosting, model averaging

Overview of machine learning methods

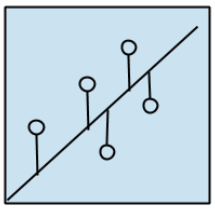
Methods often used in digital soil mapping literature, grouped by underlying concept

Regression: linear models like ordinary least squares, logistic regression. I would also put geostatistics (external-drift kriging, regression kriging) here, because the trend is most often estimated by a linear model. Further, also generalized additive models (GAM, non-linear regression based on splines) go in this category, because like linear models they make assumptions about the distribution of the errors (**parametric methods**, e.g. errors have to follow normal distribution, otherwise we need to transform the response).
Following the rather narrow definition of machine learning from SAS (see above) regressions are not per se machine learning methods, because no automatic model selection is performed and multi-collinearity has to be avoided before model fitting. No fit possible for p > n.

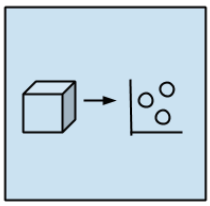
Dimension reduction: Methods, that transform the covariates to reduce the dimensions of the covariate space (principal component analysis, PCA) and apply a linear model on these transformed covariates (partial least squares, PLS).

Regularization, shrinkage: linear models that put a penalty on the coefficients to shrink them; can deal with multi-collinearity, some of those methods select covariates (e.g. lasso, ridge regression, least angle regression).

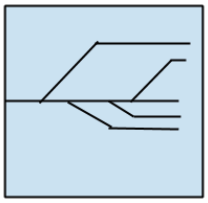
I tried to tidy up ...



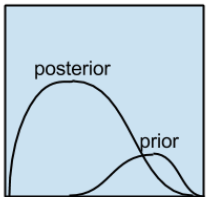
Regression
linear and non-linear models, geostatistics



Dimension reduction
PCA, PLS

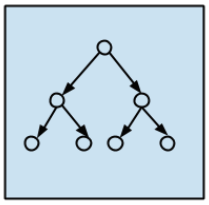


Regularisation Shrinkage
Lasso

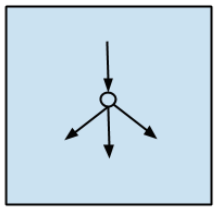


Bayes methods

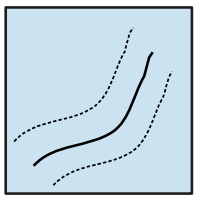
Linear (more or less)



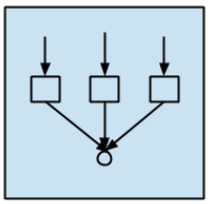
Decision trees
CART



Neuronal networks



Support vector machines
kernel methods



Ensembles
bootstrap, boosting, model averaging

High complexity

Overview continued ..

Bayes methods: models that work with prior assumptions, not that often used for digital soil mapping, but for being complete still included in the overview.

Decision trees: stepwise splitting of dataset based on a rule, recursive partitioning; easy interpretation, but decision trees are sensitive to small changes in the data (high variance method), e.g. classification and regression trees (CART), rule based models (e.g. Cubist, a decision tree with linear regressions at the nodes of the tree).

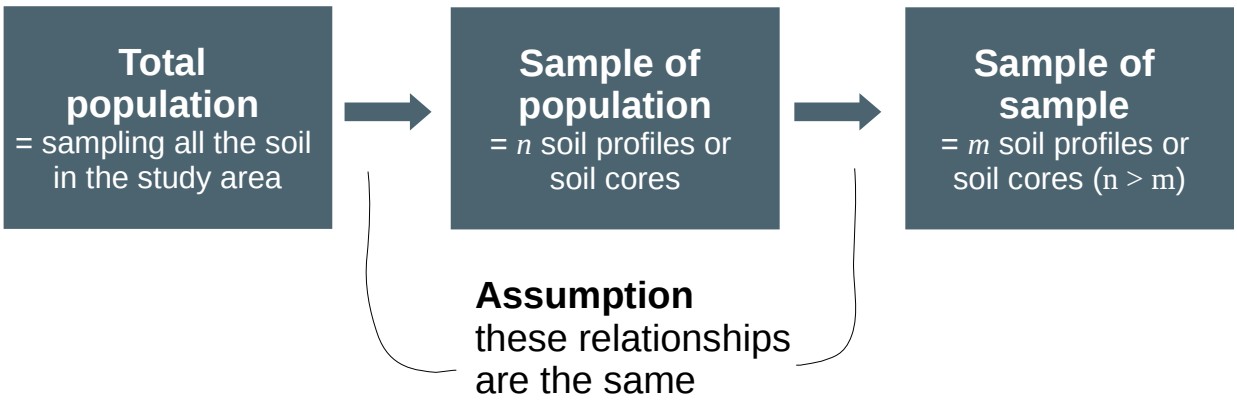
Neural networks: extracts linear combinations of the covariates and then models the response as a nonlinear function of these linear combinations; often used for unsupervised learning (artificial intelligence).

Support vector machines: splits the covariate space with a plane that optimally separates the values; instead of a plane kernel functions can be used to split the covariate space in a non-linear way; also called kernel methods; classification only, good predictions expected, not sensitive to outliers.

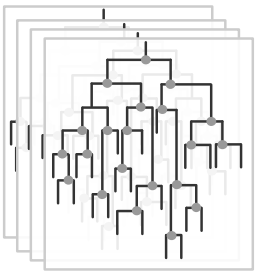
Ensemble methods: methods that combine several realisations of a single or multiple procedures/method; most often successful for procedures that have a high variance and low bias like decision trees; often used for digital soil mapping. e.g. bagging, boosting, bootstrapping, model averaging.

Pictures (besides support vect. machines) from: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Bootstrap – concept



- Idea: by **resampling** our sample many times (e.g. 1000x) we can approximate properties of the **distribution** of the total population.
- Useful to
  - create model ensembles
    bagging = boostrap aggregation
  - estimate uncertainty for any model



Bootstrap, also Bootstrapping

Resampling = draw from your sample at random, usually with replacement

Bagging = bootstrap aggregation.
Uniform *r*esampling the data with replacement (no change of response distribution), fit the data to each resampled set,
Model prediction is calculated as the average of all single predictions.

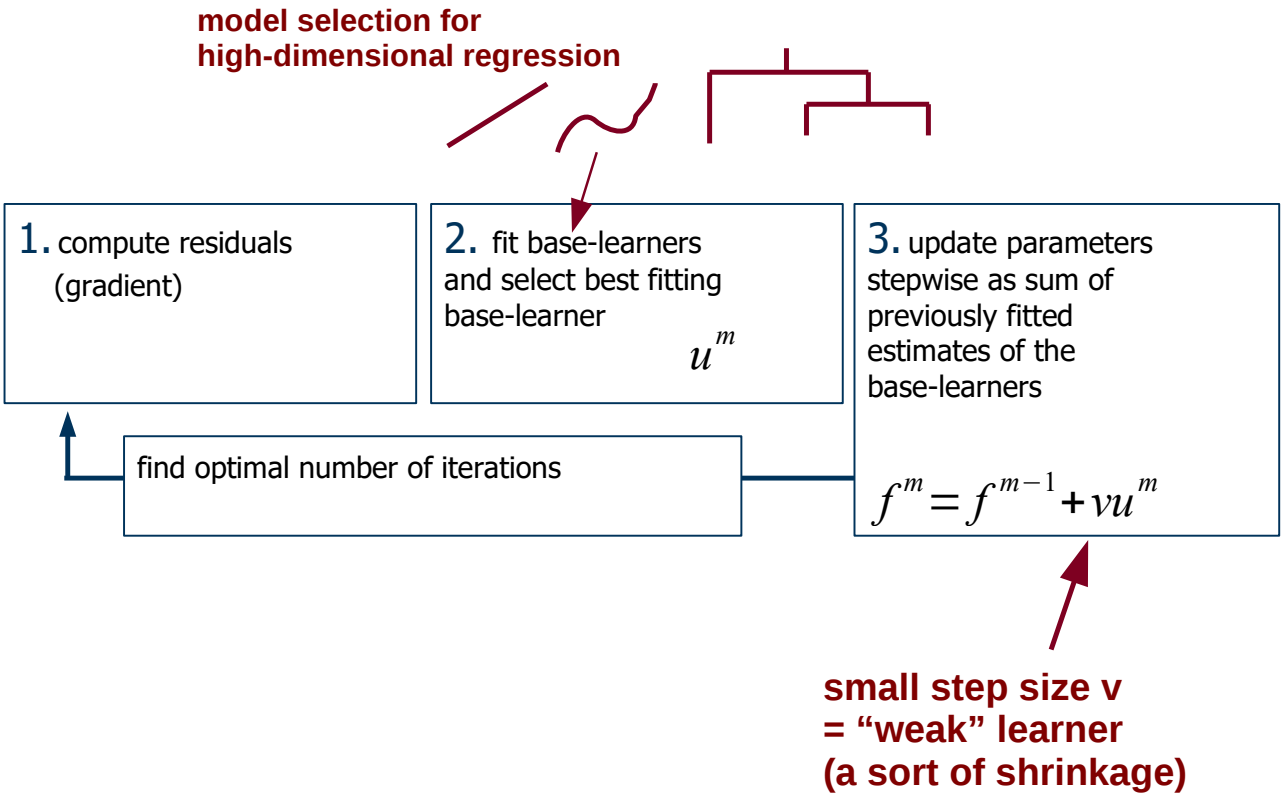
Uncertainty approximation: Bootstrap computes full predictive distribution for each prediction, e.g. we can then derive 80 % or 95 %-prediction intervals.

Non-parametric bootstrap: resample the original data (as in figure on slide)

Model-based bootstrap: fit a model and with the assumption that this model is "true", just resample the model errors.

Very powerful statistical tool, but
→ computationally intensive (CPU load)
→ invest in parallel programming!

Gradient boosting: Algorithm



Gradient boosting = Stepwise forward procedure

- «weak» learning algorithm – small step size v
- One covariate can be added multiple times

→ covariate selection for high-dimensional regression!

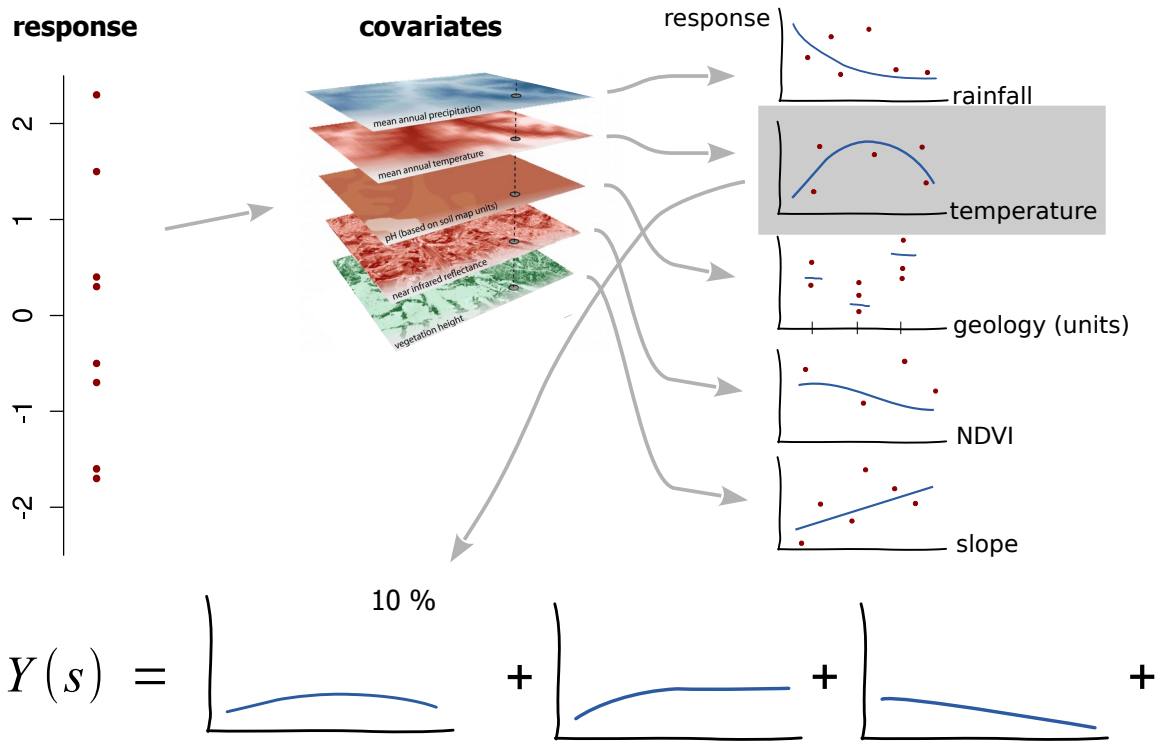
Different base-learners (base procedures for the stepwise fit) possible e.g.
- linear terms as in a linear model
- penalized splines as in an additive model
- trees as in random forest

To calculate predictions for linear and splines baselearners: sum up base-learners

→ Easy to interpret, if we use linear or splines terms (residual plots possible).

→ For trees we can compute variable importance as for random forest.

Gradient boosting: mini example



1. Calculate mean of response (center)
2. Fit all possible covariates
3. Add the best fitting one to the model, by a reduced amount (e.g. 10 %)

Repeat ..

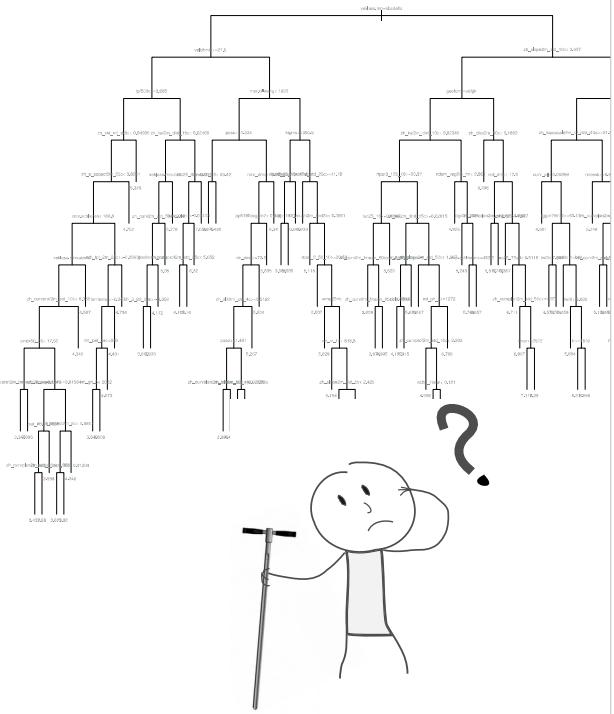
→ Number of repetition is main tuning parameter.

Model averaging
Ensembles of different methods

- Create predictions from different (ML) methods and **combine them**.
- Idea: each (ML) method as a mean of **reducing dimensions** in the dataset capturing different properties of the dataset → used methods should not be similar.
- Mathematical proofs show that combinations of different linear models result always in better performance. For other methods thats not a priori given, but very likely.
- Strategies
  - just take mean for every prediction
  - weighted mean, weights from model performance e.g. $\frac{1}{MSE}$
  - local weights with uncertainties of each method and prediction
  - linear fit with predictions as covariates and original data as response → but take car, never fit on validation set!!
  - or stacked generalisation, Bayesian approach

Model selection, why?!

- ✔ Model interpretation, likely better acceptance of final data product
- ✔ Better just use relevant covariates for prediction
- ✔ Computational effort for predictions (just prepare 12 instead of 300 rasters)
- ✔ Maybe reduce effort for future data collection and modelling on same topic
- ✖ However, theoretical statisticians do not recommenced selection, because it is often biased, difficult to find the true model..
- ✖ We might loose prediction accuracy...



Is there a reason for model selection?
Or is it enough to do model building?

Model selection = reduce the initial covariate set
Model building = find relationships between covariates and response

Model flexibility vs. model complexity

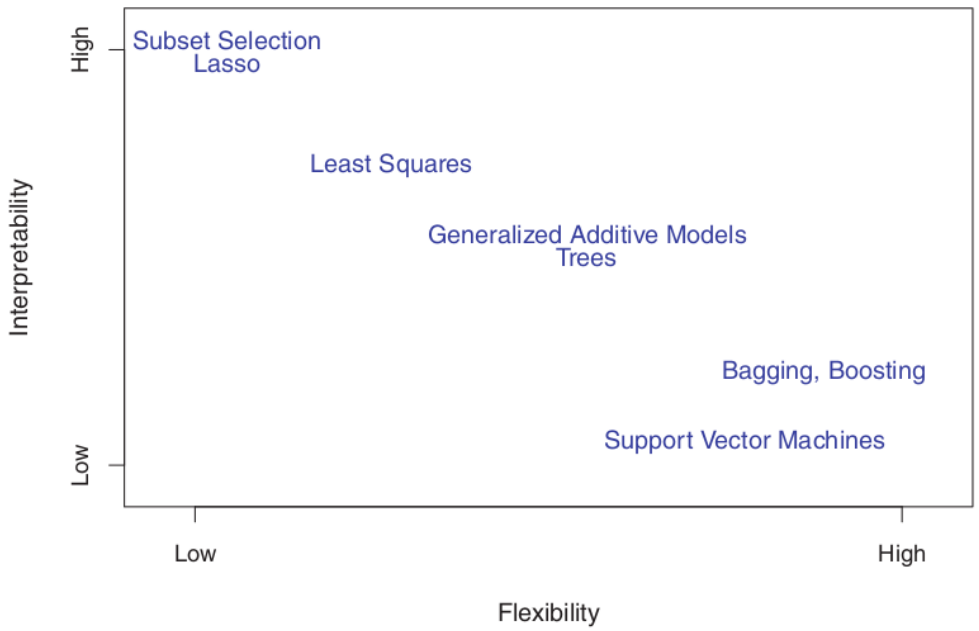


FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

Gareth et al. 2013, p. 25

Model selection – strategies

- Ask an soil surveyor that is familiar with the area
- Remove n worst covariates
- Stepwise addition or removal (see later)
- Test all possible models
- Shrinkage (e.g. by boosting or lasso, see later).

Strategies

- **Ask an soil surveor that is familiar with the area**
  Ask experts who know the area of interest. Only use the data layers that seem relevant for soil forming in this area. Further ask for meaningful derivates form your geodata (e.g. agregate geological map in a meaningful way).
  Problem: most likely biased.. according to literature other model selection procedures yielded better model performance.

- **Remove n worst covariates**
  Fast, just one more model fit.
  Problem: unclear how to define threshold (which are n worst covariates?)
- **Stepwise addition or removal** (see below)
- **Test all possible models, best subset selection**
  Problem: might be extremly time consuming, a bit arbitrary which model will end up beeing the best (no clear path of model optimization)
- **Shrinkage / Regulariation**
  e.g. by gradient boosting, for lasso see below

Model selection for linear regression

Usually used for linear models (e. g. OLS):

- Forward selection
  - Start with a model with just an intercept
  - Try all possible covariates and add the one that results in the best fit (e.g. R^2)
  - Add covariates until increase in R^2 is only very small (threshold) or evaluate the gained models by cross validation.

- Backward elimination
  - Fit the full model with all covariates
  - Remove the covariate that will cause the smallest decrease in R^2
  - Continue removing until the drop in R2 gets too big (threshold) or evaluate the fitted models by cross validation

OLS: ordinary least squares fit for linear models

Model selection for linear regression

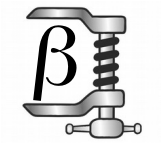
Evaluation of forward and backward selection:

- ✔ Straightforward, easy to understand
- ✔ Follows a selection path, hence much more efficient and less arbitrary than best subset selection
- ✖ Binary selection: A covariate is either in or out.
  Being in by e.g. 30 % is not possible.
- ✖ Multi-collinearity, unstable fits!
  We need to remove correlated covariates beforehand.
- ✖ Likely overfits the data, biased model selection.
  Most often does not find true model.
- ✖ We can not fit p > n

→ Possible solution: regularization / shrinkage with e.g. **lasso**.

p: number of coefficients
n: number of observations.

Model selection with lasso



- **Shrinkage:** include a covariate, but with smaller / down-weighted coefficients
- Different approaches (ridge regression etc.), most promising: Lasso: least absolute shrinkage and selection operator

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

OLS **Lasso penalty**

- Thus the lasso does a kind of continuous subset selection.
- Tuning Parameter λ, find by cross validation

Model selection with lasso

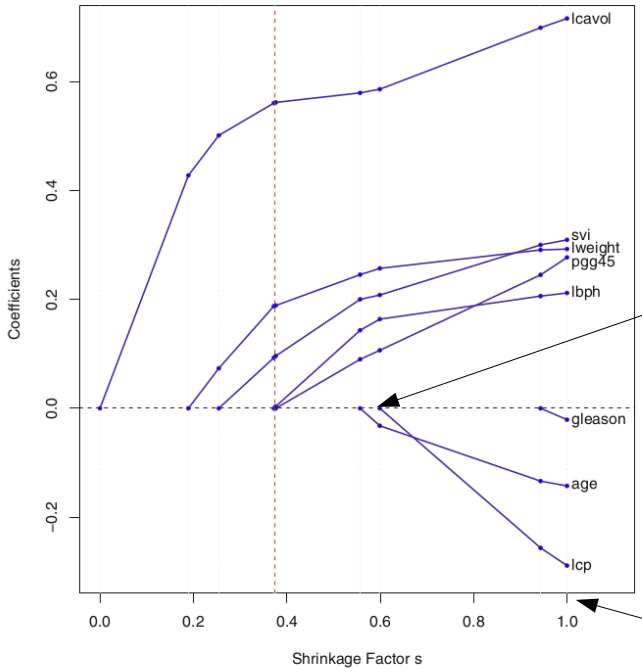


FIGURE 3.10. *Profiles of lasso coefficients, as the tuning parameter t is varied. Coefficients are plotted versus $s = t/\sum_1^p |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.36$, the value chosen by cross-validation. Compare Figure 3.8 on page 65; the lasso profiles hit zero, while those for ridge do not. The profiles are piece-wise linear, and so are computed only at the points displayed; see Section 3.4.4 for details.*

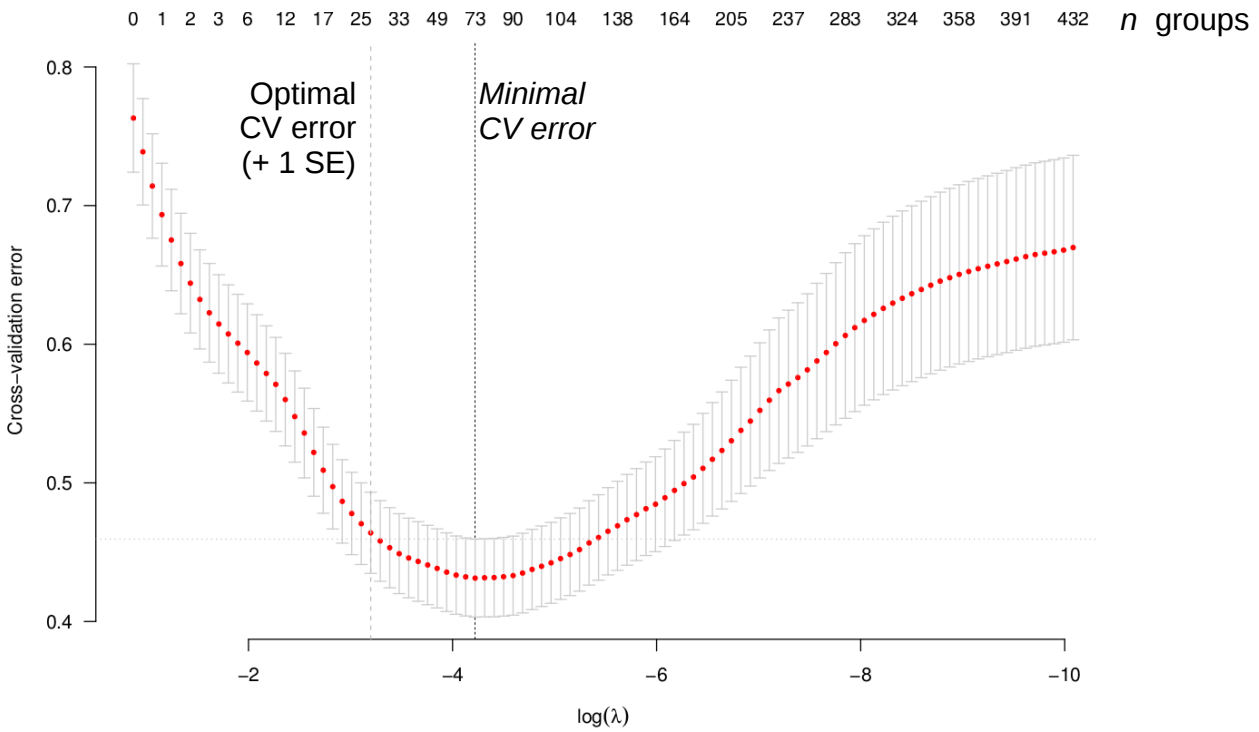
Hastie et al. 2009, p. 70

Path of coefficients for increasing tuning parameter

Coefficient becomes 0, meaning the covariate is removed from the model

With lambda = 1, there is no shrinkage, and we have the normal ordinary least squares linear model fit

Model selection with lasso: tuning



Berne data set, subsoil pH, >400 partly highly correlated and noisy covariates

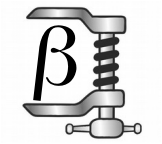
We have to find the optimal lambda. We just try a series of lambdas and compute the cross validation (CV) error.

This error has a certain variation. Instead of using lambda at the minium CV error, we try to exclude as many covariates as possible and choose an optimal CV error (e.g. plus one standard error, as in figure).

Hint on group lasso:

n groups: number of covariates, if we use categorical covariates, these are coded as dummy variables (with 1 and 0 for each category). In “group lasso” the coefficients of these dummy variables are set to 0 as groups at the same time. Hence, the complete categorical covariate is excluded at once. In non-group lasso, some categories can remain in the model while others are excluded.

Is lasso useful for DSM?



- ✔ Very fast
- ✔ Selects covariates
- ✔ No problems with collinearity
- ✔ Easy interpretation (linear relationships)
- ✔ Linear regression with a lot of covariates, even *n>>p*

- ✖ Linear only, no interactions if not added explicitly
  (if p >>> n becomes nonlinear again)
- ✖ Take care, not always stable
- ✖ Rather underfitting
  (possible solution: relaxed Lasso with a second fit on non-zero covariates only)
- ✖ Standard errors not defined, prediction uncertainty only with bootstrap
- ✖ No direct spatial modelling, only via workaround

Model selection with random forest

What again was this **out-of-bag (OOB) error**?

For each observation $z_i = (x_i, y_i)$, *construct its random forest predictor by averaging* only *those trees corresponding to bootstrap samples in which* z_i did not *appear.*

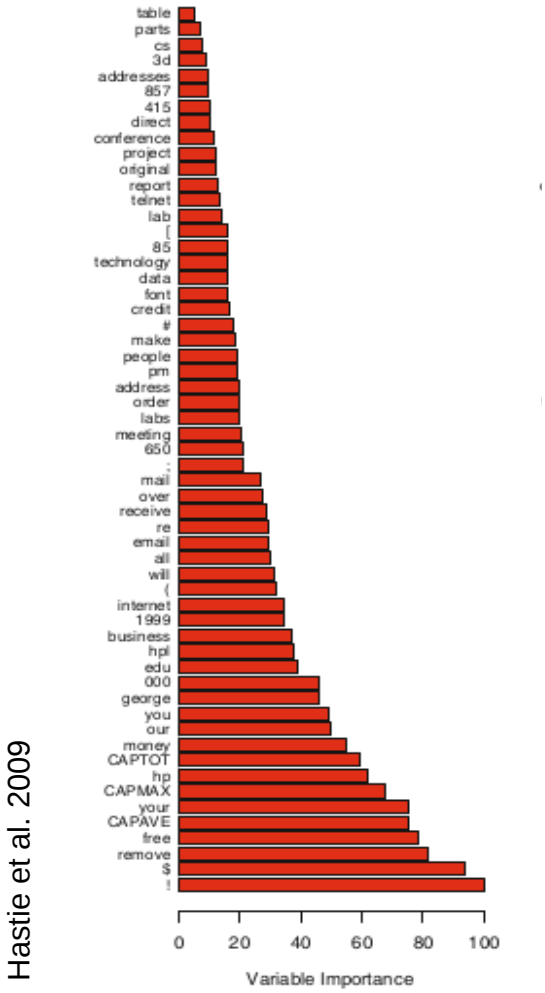
Hastie et al. 2009, p. 593

Covariate importance in random forest:

- How much worse do OOB predictions get if we randomly permute a covariate?

  → Hint: removing a covariate and refit random forest is not the same, other correlated covariate could replace its “predictive capacity”

Covariate importance (importance(type = 1))
Mean decrease in accuracy. Based on permutation of the covariate, oriented on predictions.



Hastie et al. 2009

Model selection for tree based methods, very simple:

Recursive backward elimination

1) Remove covariate(s) with lowest importance

2) Refit random forest with remaining covariates

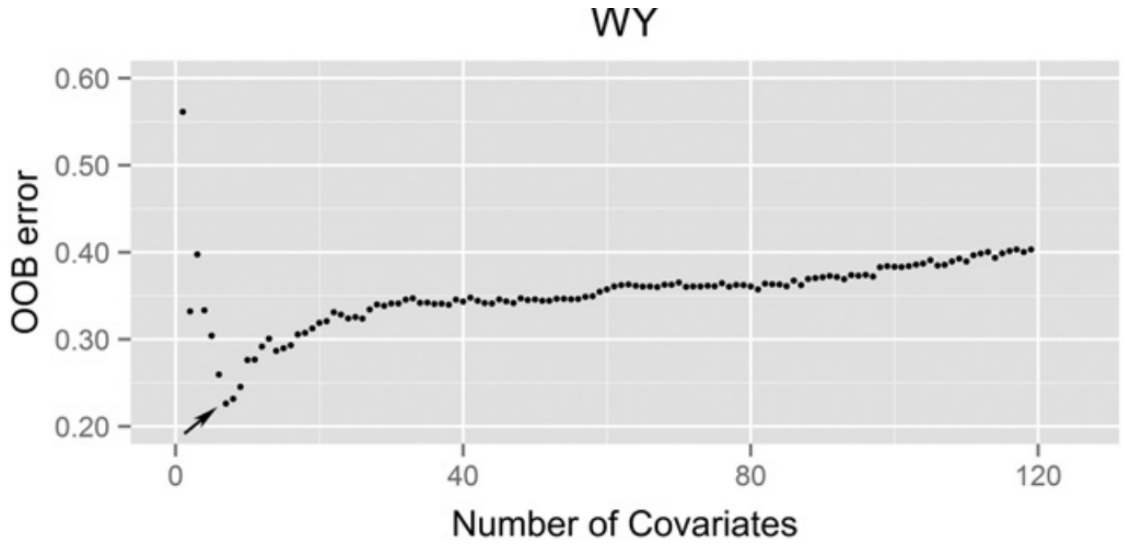
Find optimum number of covariates by minimizing OOB error.

Main problem of this approach:

Correlated covariates remain, because of randomisation at each split based on the tuning parameter m_{try} .

Interpretation needs to account for that.

Model selection with random forest
Example



Brungard et al. 2015

Model selection with random forest

- ✔ Straightforward approach
- ✔ Easy to implement
- ✔ From my experience: efficient
  (meaning a lot of covariates are removed)

- ✖ Artefacts in predictions possible
- ✖ Stability unclear?
  (Small changes in covariate or response values might change result drastically)
- ✖ Biased?
  (maybe biased as other backward elimination methods)
- ✖ Time consuming
  (iterative method, no paralell computing possible)
- ✖ Possibly: A lot of effort for a small result

Content of lecture

- Terms and concepts
- Spatial modelling: requirements?
  - Side note: overfitting
- Overview of ML and their strategies
  - Boostrap
  - Boosting
  - Model averaging
- Model selection
  - Why model selection?
  - Selection with lasso
  - Selection with random forest
- Model interpretation

Covariate interpretation for any model

Partial residual plots (see e.g. Wikipedia)
For regression based methods

Plot residuals of full model plus the covariate effect $\hat{\beta}_i X_i$ against the values of covariate X_i, for better interpretation center $\hat{\beta}_i X_i$

$$Residuals \; + \; \hat{\beta}_i X_i \;\; versus \;\; X_i$$

Partial dependence plots Hastie et al. 2009, chapt. 10.13.2
For any “black box” learning model

Dependence of covariate on response after *accounting* (not *ignoring*) for the effects of all other covariates. Approximation of function by:

$$\bar{f}_S(X_S) = \frac{1}{N} \sum_{i=1}^{N} f(X_S, x_{iC}),$$

Partial residual plot

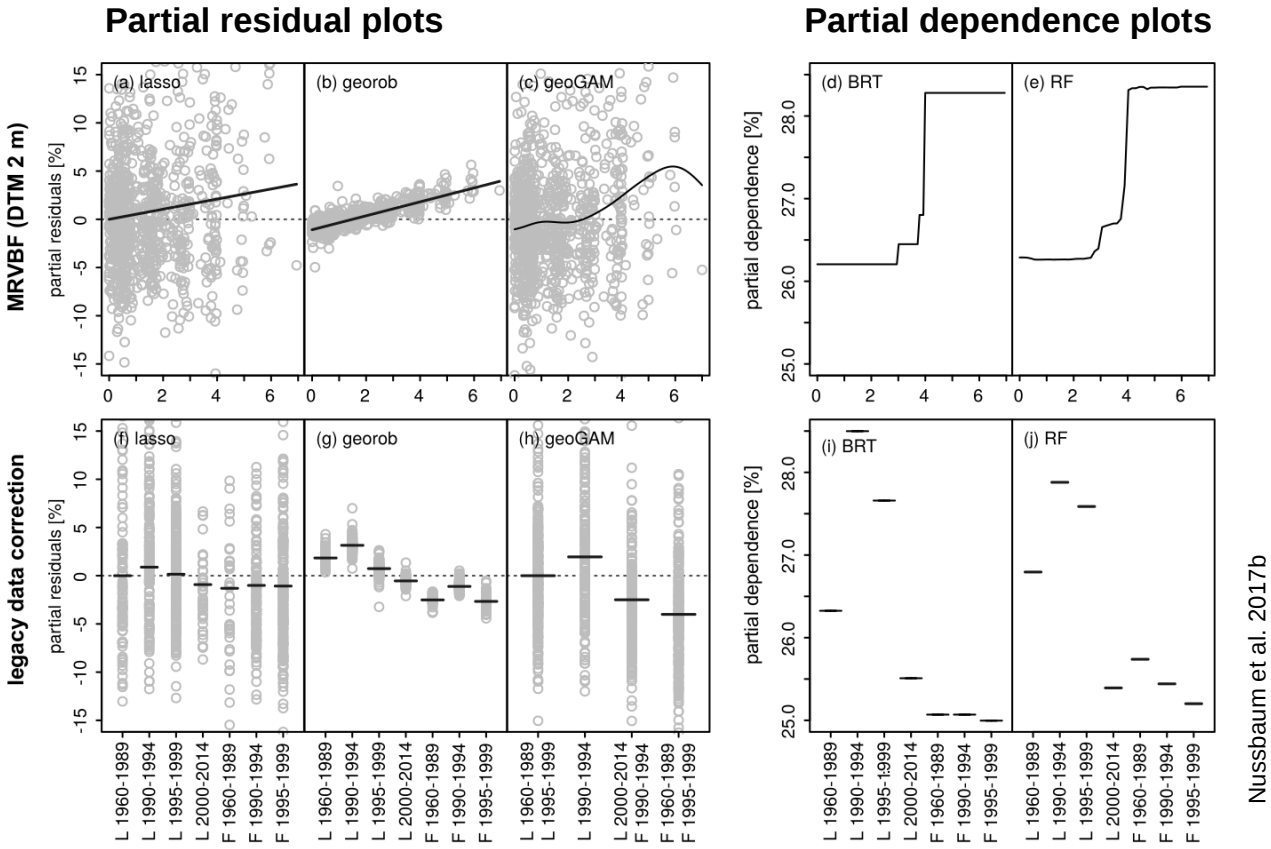
residuals: residuals from the full model

$\hat{\beta}_i$: regression coefficient of the covariate *i* you are interested in,
this coefficient can be estimated by ordinary least squares, by a robust estimator, additive model, lasso etc.

X_i : the original values of this covariate *i*

Partial dependence plot

Covariate interpretation for any model



Modelled response: topsoil clay content (0-10 cm)

lasso: model fitted by least absolute shrinkage and selection operator

georob: robust geostatistical model

geoGAM: geo-additive model, a non-linear regression that considers the spatial autocorrelation by including a smooth spatial surface based on the coordinates.

BRT: boosted regression trees, gradient boosting with trees as baselearners

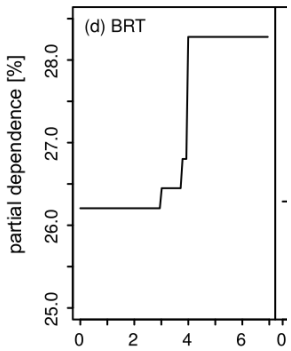
RF: random forest

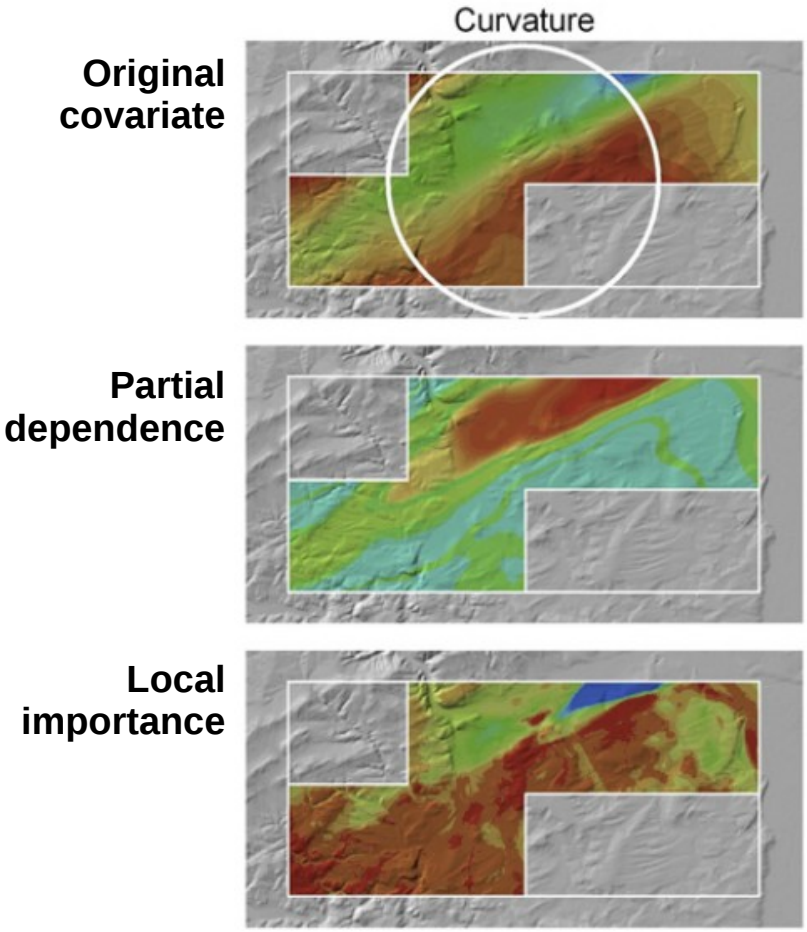
MRVBF: multi-resolution valley bottom flatness, a terrain attribute indicating erosion sites (small values) and accumulation sites (large values).

Legacy data correction (timeset in the berne dataset, see practical training): The soil profiles were grouped according their survey period and method (L: laboratory measurement, F: field estimate). With this factor we tried to account for the time variation and the differences in sampling protocol.

Spatial covariate interpretation

Create maps from relationship:





Behrens et al. 2014

Partial dependence map

If we create a continuous partial dependence plot (e.g. lowess), we can use this function to create a map for every point of our study area. This allows to interpret the local dependence of one covariate.

Take care with interpretation of partial residual and dependence plots:

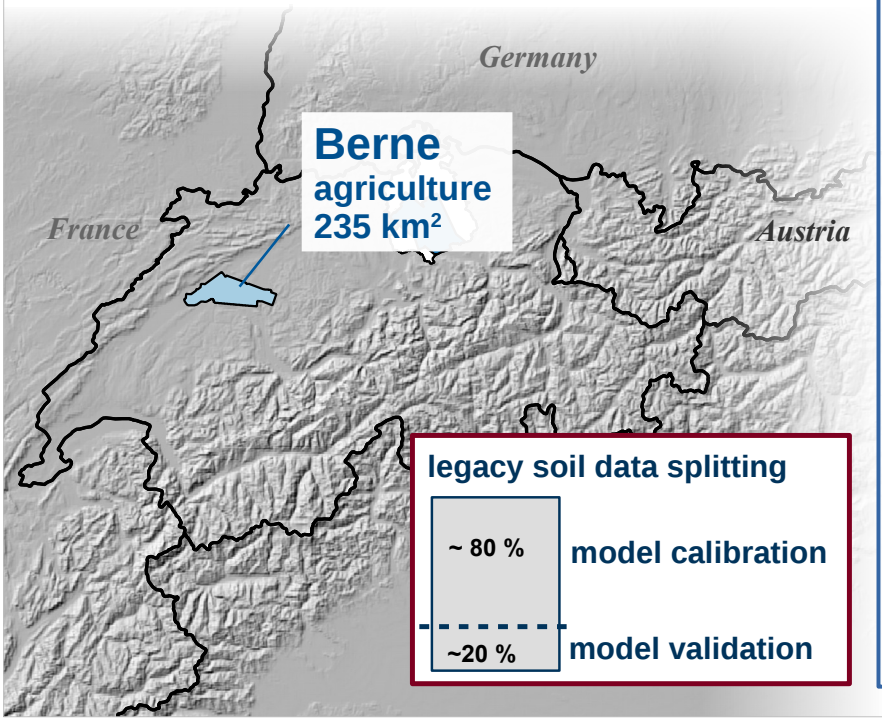
If there are many (correlated) covariates, it is difficult to choose which to interpret.
If the covariates are correlated they might replace each other in the model.

My personal tips for your digital soil mapper career:

- Do not believe there is the single one solution (e.g. one model that never fails.). So make sure you understand advantages and disadvantages.
- Do not neglect classical statistics. Without understanding linear models you will never master machine learning!
- If a method is hard to understand to you, please use a simpler one, that you feel secure with! Your credibility and credibility of your data product is at stake.
- Do not neglect soil knowledge. Most likely consulting a soil surveyor might improve your model more than the 100redst tuning of the latest fancy method.

Practical training: Berne soil mapping study

~ 1000 sites with legacy soil data from 1970-1980
Nussbaum et al. 2017b



Numerous covariates

Climate
different data sets
(monthly resolution)

Soil
soil overview map
historic wetlands
anthropogenic soil interventions
drainage networks

Parent material
(hydro)geological maps
and derivates

Vegetation
Landsat, SPOT5, DMC mosaic
forest vegetation map and
species composition

Terrain
90 derived attributes
(multiple scales)

Dataset (with even more covariates) used for publication:

Nussbaum, M., Spiess, K., Baltensweiler, A., Grob, U., Keller, A., Greiner, L., Schaepman, M. E., and Papritz, A.: Evaluation of digital soil mapping approaches with large sets of environmental covariates, SOIL, 4, 1-22, https://doi.org/10.5194/soil-4-1-2018, 2018.

Practical training

You will learn:

lasso

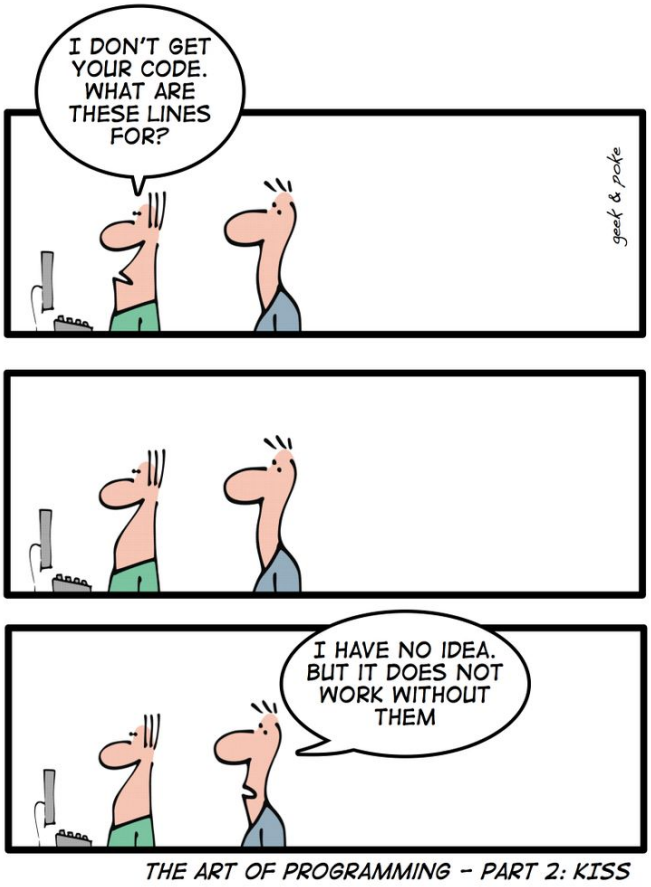
model selection

model interpretation

and much more!

until 12:00

then we discuss
the questions



http://geekandpoke.typepad.com/geekandpoke/2009/07/the-art-of-programming-part-2.html

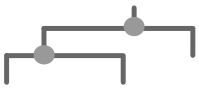
Summary
You will learn to ..

- fit a lasso model for a continuous and a categorical response,
- select covariates with random forest
- interpret partial residual and dependence plots
- and, if you like, to fine tune your style of R programming or documentation, apply model averaging, fit gradient boosting models etc.

Documents for the exercises:

- ISRIC-module-ML-2-training.**pdf:** The instructions.
- ISRIC-module-ML-2-training.**R:** The plain code.
- ISRIC-module-ML-2-training.**Rnw**: KnitR file that was used to create .PDF and .R file.

Aditional material on advanced tasks: Should I use boosted trees or random forests?



Boosted trees

- ✔ Selects covariates weakly
- ✔ Covariate importance
  for interpretation and maybe selection
- ✖ Predictive accuracy slightly lower than random forest
- ✖ Prediction uncertainty only by bootstrapping
- ✔ Reduces bias by fitting on residuals

Speed?
Do some benchmarking if interested ;-)

Random forest

- ✖ Does not select covariates
- ✔ Covariate importance
  for interpretation and maybe selection
- ✔ From my datasets on average best performance
  (up to 50 different responses tested)
- ✔ Prediction uncertainty with quantile regression forest
- ✖ Always fits on data with same distribution