# MA8701 Advanced methods in statistical inference and learning

## ~~L3-4~~: Model selection and assessment

L4:

Mette Langaas

~~1/17/23~~ 2023.01.20

# Cross-validation (CV)

(ESL Ch 7.10, 7.12 - most should be known from TMA4268)

The aim is to estimate $\text{Err}_T$, but from simulation analyses (ESL Ch 7.12) it turns out that cross-validation estimates $\text{Err}$ "the best".

The starting point for the method is that we only have one training set - and try to use that for either model selection or model assessment (not both).

What to do when both is needed, is not covered in this chapter. Nested cross-validations aka two-layers of cross-validation is one possibility. Another is to set aside data for a test set for model assessment, but use the training set in cross-validation for model selection.

# CROSS-VALIDATION (ESL 7.10)

We will estimate $Err = \underset{XY}{E}(L(Y, \hat{f}(X)))$ extra sample error

NB: $Err$, not $Err_{\mathcal{T}}$!

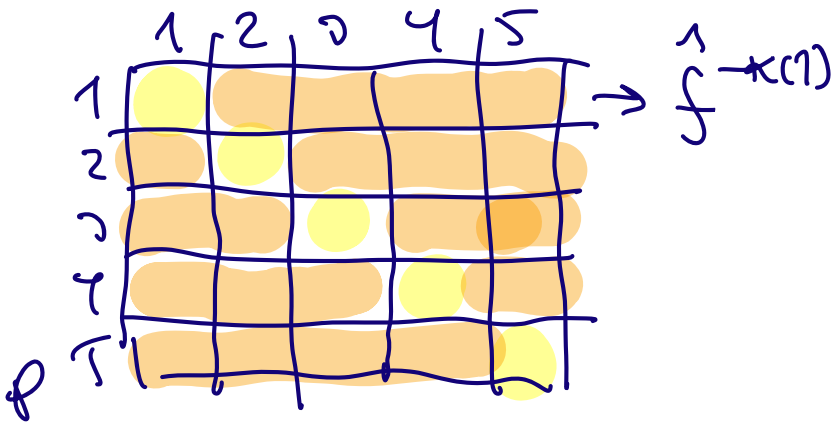
## K-fold CV

randomized

$\kappa : \{1, .., N\} \to \{1, ..., K\}$

indexing function

all folds of approx. the same size and no overlap between folds



$\hat{f}^{-k(i)}(x)$ : estimator based on all data except on fold $k$

Err:

$$CV(\hat{f}_\alpha) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, \hat{f}^{-k(i)}_\alpha(x_i)\right)$$

$\alpha$: tuning parameter

$-k(i)$: the fold where $x_i$ was not a part of the "training set"

Model selection: choose $\alpha$ where $CV(\hat{f}_\alpha)$ is minimum or use 1-SD-rule

## Formal set-up for model assessment

- The allocation of observation $\{1, \ldots, N\}$ to folds $\{1, \ldots, K\}$ is done using an indexing function $\kappa : \{1, \ldots, N\} \to \{1, \ldots, K\}$, that for each observation allocate the observation to one of $K$ folds.
- Further, $\hat{f}^{-k}(x)$ is the fitted function, computed on the observations except the $k$th fold (the observations from the $k$th fold is removed).
- The CV estimate of the expected prediction error $\text{Err} = \text{E}_T\text{E}_{X^0,Y^0}[L(Y^0, \hat{f}(X^0)) \mid T]$ is then

$$\text{CV}(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i))$$

Efron and Hastie (2016) page 218: "$\text{CV}(\hat{f})$ is estimating the average prediction error of the algorithm producing $\hat{f}$, not $\hat{f}$ itself".

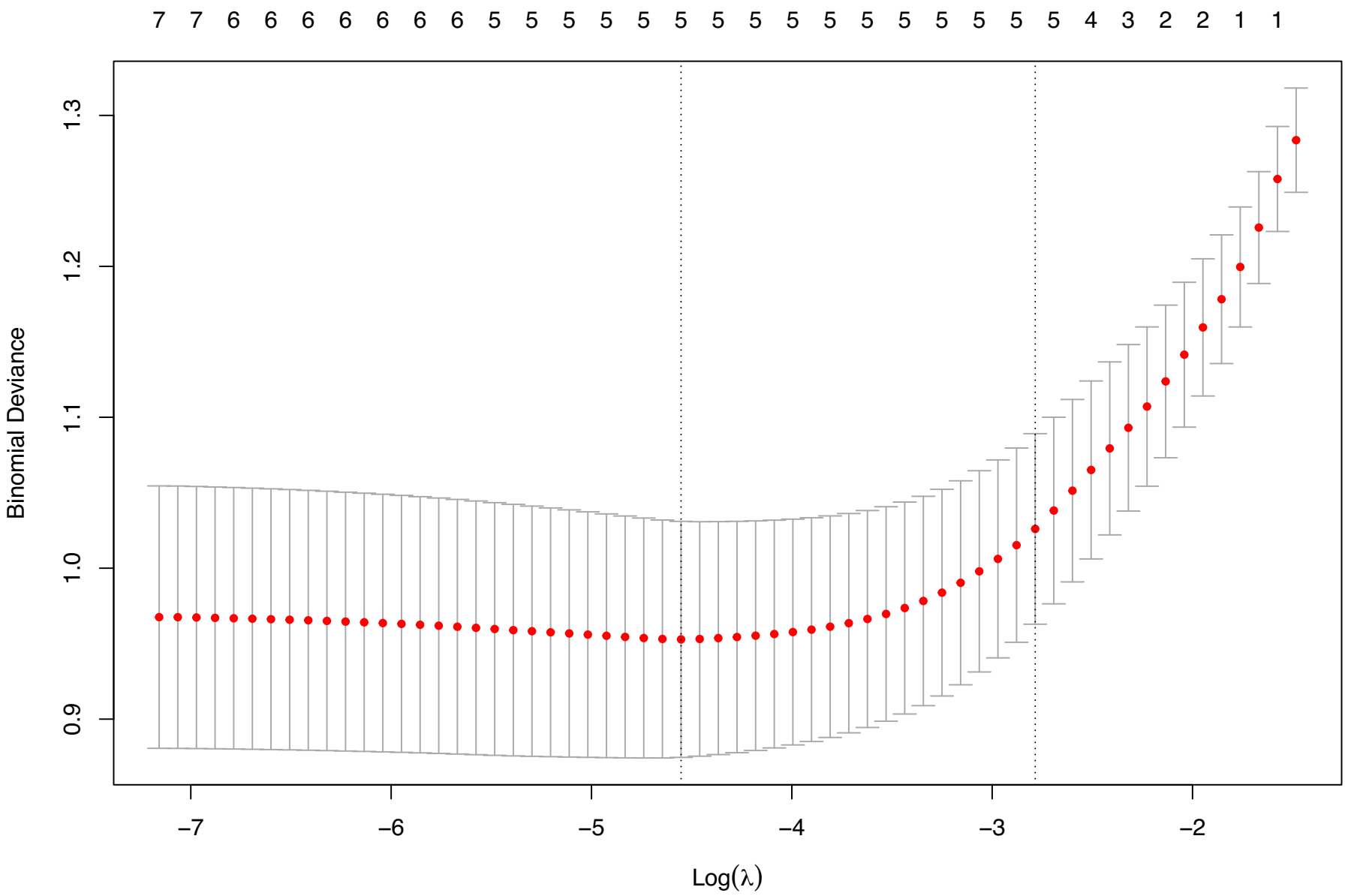
## Pima indian example

We will use the classical data set of *diabetes* from a population of women of Pima Indian heritage in the US, available in the R `MASS` package. This version of the data has no missing values. The following information is available for each woman:

- diabetes: 0= not present, 1= present
- npreg: number of pregnancies
- glu: plasma glucose concentration in an oral glucose tolerance test
- bp: diastolic blood pressure (mmHg)
- skin: triceps skin fold thickness (mm)
- bmi: body mass index (weight in kg/(height in m)$^2$)
- ped: diabetes pedigree function.
- age: age in years

We will use the default division into training and test in the MASS library, with 200 observations for training and 332 for testing.

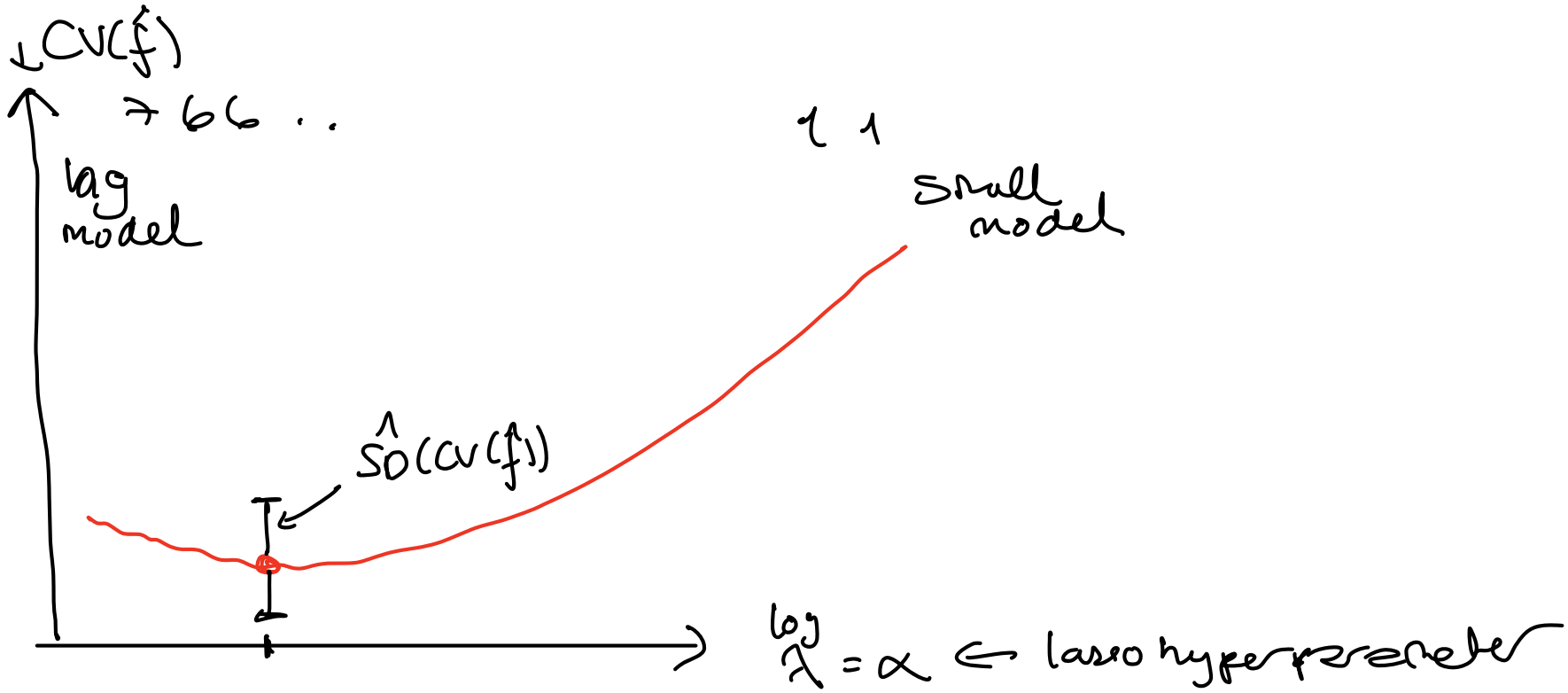
Only training set used here.


7 7 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 4 3 2 2 1 1
Binomial Deviance
0.9
1.0
1.1
1.2
1.3
−7
−6
−5
−4
−3
−2
Log(λ)

## Group discussion

The lasso logistic regression (to be studied in Part 2) was used to fit the data, and some loss function is plotted on the vertical axis (more in Part 2) and on the horisontal axis the loss for different fits for different choices of a complexity parameter is given. 10-fold crossvalidation is used. (Just assume that a generic prediction is used, this is not meant to be specific for the lasso.)

- What are the red dots and how have they been calculated?
- What are the the vertical bars sticking out of each red dot, and how have they been calculated? What do they picture?
- What (your choice) is the optimal choice of the complexity parameter?




Red dot: $CV(\hat{f}(\alpha))$, Vertical bar: $\hat{SD}(CV(\hat{f}))$

$$\hat{Err}: CV(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i)) = \frac{1}{K}\sum_{j=1}^{K} \hat{CV}_j$$

$$\frac{1}{N_j}\sum_{i \in \kappa(i)} L(y_i, \hat{f}^{-\kappa(x_i)}(x_i))$$

Want: $\widehat{\mathrm{Var}}(\widehat{\mathrm{Err}}) = \mathrm{Var}\left(\frac{1}{K}\sum_{j=1}^{K} \widehat{CV}_j\right) = \frac{1}{K^2}\sum_{j=1}^{K} \mathrm{Var}(\widehat{CV}_j)$

$= \frac{1}{K}\,\mathrm{Var}(\widehat{CV}_j)$

$\widehat{\mathrm{Var}}(\widehat{CV}(\hat{f})) = \frac{1}{K}\cdot\frac{1}{K-1}\sum_{j=1}^{K}(\widehat{CV}_j - \widehat{CV}(\hat{f}))^2$

$\widehat{SD}(\widehat{CV}(\hat{f}))$ is shown as vertical bar

"We choose the most parsimonious model whose error is no more than one SE above the error of the best model"

} 1 SD-error rule

## Choice of $K$

- Popular choices are 5 and 10 based on observations in simulation studies- and arguments similar to a bias-variance trace off.
- $K = N$ is called *leave-one-out* cross-validation LOOCV, and gives the lowest bias for estimating the Err.

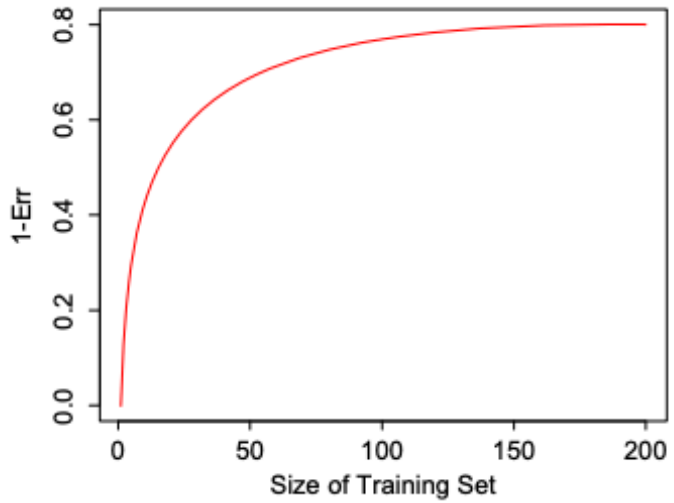



**FIGURE 7.8.** *Hypothetical learning curve for a classifier on a given task: a plot of* $1-\text{Err}$ *versus the size of the training set* $N$*. With a dataset of* 200 *observations,* 5*-fold cross-validation would use training sets of size* 160*, which would behave much like the full set. However, with a dataset of* 50 *observations fivefold cross–validation would use training sets of size* 40*, and this would result in a considerable overestimate of prediction error.*

## Generalized cross-validation (GCV)

For LOOCV with squared loss and linear fitting. Remember

$$\hat{\mathbf{y}} = \mathbf{S}y$$

For many fitting methods (including MLR)

$$\frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{f}^{-i}(x_i)]^2 = \frac{1}{N}\sum_{i=1}^{N}[\frac{y_i - \hat{f}(x_i)}{1 - S_{ii}}]^2$$

where $S_{ii}$ is the $i$th diagonal element of $\mathbf{S}$. This leads to the GCV approximation:

$$\text{GCV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N}[\frac{y_i - \hat{f}(x_i)}{1 - \text{tr}(\mathbf{S})/N}]^2$$

where we recognise the effective number of parameters trace$(\mathbf{S})$. In some settings the trace$(\mathbf{S})$ is computed more easily than the individual elements $S_{ii}$.

## The wrong and the right way to do cross-validation

In short: make sure that all part of the model fit process is "inside" the CV.
See learning material from TMA4268: Module 5: Resampling, and I also recommend to work on Problem 3 with solutions

## Group discussion

Can you give one example of a right way to do cross-valiation and also a wrong way? If you want you may used the Pima-indians as an example, but other examples may also be used.

- randomized allocation to folds, maybe also stratified
- remember also standardization part of CV-loop

# Selection bias in gene extraction on the basis of microarray gene-expression data

Article by Christophe Ambroise and Geoffrey J. McLachlan, PNAS 2002: Direct quotation from the abstract of the article follows.

- In the context of cancer diagnosis and treatment, we consider the problem of constructing an accurate prediction rule on the basis of a relatively small number of tumor tissue samples of known type containing the expression data on very many (possibly thousands) genes.
- Recently, results have been presented in the literature suggesting that it is possible to construct a prediction rule from only a few genes such that it has a negligible prediction error rate.
- However, in these results the test error or the leave-one-out cross-validated error is calculated without allowance for the selection bias.

---

- There is no allowance because the rule is either tested on tissue samples that were used in the first instance to select the genes being used in the rule or because the cross-validation of the rule is not external to the selection process; that is, gene selection is not performed in training the rule at each stage of the crossvalidation process.
- We describe how in practice the selection bias can be assessed and corrected for by either performing a crossvalidation or applying the bootstrap external to the selection process.
- We recommend using 10-fold rather than leave-one-out cross-validation, and concerning the bootstrap, we suggest using the so-called .632 bootstrap error estimate designed to handle overfitted prediction rules.
- Using two published data sets, we demonstrate that when correction is made for the selection bias, the cross-validated error is no longer zero for a subset of only a few genes.

---

0.632+ ?

# BOOTSTRAP METHODS (ESL 7.11)

We study nonparametric bootstrap – sampling from data – and focus on estimating Err.

$Z_1 = (x_1, y_1), Z_2 = (x_2, y_2), \dots, Z_N = (x_N, y_N)$

are independent observation pairs from the training set

$Z = \{Z_1, \dots, Z_N\}$

↓ draw with replacement N obs. pairs $Z^{*b}$ ← bth sample

↓ repeat B times

$Z^{*1}, Z^{*2}, \dots, Z^{*B}$

↓

calculate the quantity of interest: $\hat{f}^{*b}$ for each boot sample

How to use $\hat{f}^{*b}$ to estimate Err?

First try: $\hat{Err}_{boot} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \hat{f}^{*b}(x_i))$

$Z^{*b}$: act like a training set

$Z_{org\ data}$: act like test set

Quality of $\hat{Err}_{boot}$? Too low ↙ $\hat{Err}_{boot}$ too low!

Because overlap between "train" and "test" data!

However - $\widehat{\text{Err}}_{\text{boot}}$ is not a good estimator: bootstrap datasets are acting as training data and the original data as a test sample - and the two samples have observations in common.

This overlap can make predictions too good. Remember, in CV we have no overlap.

**Q:** What is the probability that observation $i$ is included in bootstrap sample $b$?

In one draw:

$P(\text{draw } x_i) = \frac{1}{N}$, $P(\text{not draw } x_i) = 1 - \frac{1}{N}$

In $N$ (independent) draws: $P(\text{not draw } x_i \text{ in any draw}) = \left(1 - \frac{1}{N}\right)^N$

$P(x_i \text{ is included in boot sample } b) = 1 - (1 - \frac{1}{N})^N$

$\xrightarrow[n \to \infty]{} \approx 1 - e^{-1} = 0.632$

$P(\text{not included}) = 0.368$

Ex: two-class problem where $x$ and $y$ ($0/1$) are independent ($y$: random assignment)

$Err \simeq 0.5$ if use

1-NN method: obs $i$ is in boot $b$: $I(y_i \hat{f}^{-b}(x_i)) = 0$

else random guess

$0.632 \cdot 0 + 0.368 \cdot 0.5 = 0.184$ too low!

## Second try: leave-one-out bootstrap aka out of bag OOB

$$\hat{Err}^{(1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} L(y_i, \hat{f}^{*b}(x_i))$$

- $|C^{-i}|$: how many such
- $b \in C^{-i}$: boot samples where $z_i$ not included
- $L(y_i, \hat{f}^{*b}(x_i))$: no overlap between $y_i$ and $\hat{f}$ by construction!

$\hat{f}^{*b}$ is based on bootstrap sample $b$, with $N$ observations, but what is the average number of distinct obs in $Z^{*b}$? (distinct: we assume $z$ has $N$ distinct obs)

$E$(number of distinct obs drawn)

$= E(I_1 + I_2 + \cdots + I_N) = N\cdot(1-(1-\frac{1}{N})^N)$

$\approx N \cdot 0.632$

$I_i = \begin{cases} 0 & \text{not drawn} \\ 1 & \text{drawn} \end{cases}$

$E(I_i) = P(I_i = 1) = 1-(1-\frac{1}{N})^N$

The estimator $\hat{f}^{*b}$ will then be using very little of the original training data $\Rightarrow$ $\hat{Err}^{(1)}$ too big!

Now - let us use $\hat{Err}^{(1)}$ (too big) together with

$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$ (too low) - in a weighted fashion

↑ What is this weight?

Third try:

$$\hat{Err}^{(0.632)} = 0.368\,\overline{err} + 0.632 \cdot \hat{Err}^{(1)}$$

good performance in light fitting situations

but not for "overfitted situations".

Ex: $\overline{err} = 0$ because 1NN always in bootstrap

$\hat{Err}^{(1)} = 0.5$ $\quad \hat{Err}^{(0.632)} = 0.632 \cdot 0.5 + 0 = 0.316$ true 0.5

According to ESL (page 251): the derivation of the .632 estimator is complex, and the estimator is expected to work well in situation where the data is not overfitted, but may break down in overfit situations.

According to CASI (page 323) the .632 rule is less variable than the leave-one-out CV.

Example of this on page 251-252: two equal size classes where predictors independent of class, classification with $1$NN gives $\overline{\text{err}} = 0$ and $\widehat{\text{Err}}^{(1)} = 0.5$ and thus $\widehat{\text{Err}}^{(.632)} = 0.632 \cdot 0.5 = 0.316$, where here the true error rate is $0.5$.

Fourth try:

$$\widehat{Err}^{(0.632+)} = (1-\hat{w})\,\overline{err} + \hat{w}\,\widehat{Err}^{(1)}$$

$\hat{w}$ is decided by evaluating the amount of (something) overfitting.

There is an improved version of the estimator - taking into account the amount of overfitting, leading to an adjustment to the weight $w = 0.632$ (and $1 - w = 0.368$) dependent on a socalled *no-information error rate*$=\gamma=$the error rate of the prediction rule when predictors and class labels are independent.

$$\hat{\gamma} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} L(y_i, \hat{f}(x_{i'}))$$

obs $i$; prediction at a different $x_{i'}$

Further the *relative overfitting rate* is defined to be

$$\hat{R} = \frac{\widehat{\mathrm{Err}}^{(1)} - \overline{\mathrm{err}}}{\hat{\gamma} - \overline{\mathrm{err}}}$$

$\hat{R} = 0 \Leftrightarrow \hat{\mathrm{Err}}^{(1)} = \overline{\mathrm{err}}$ so no overfitting and $\hat{w} = 0.632$ used

$\hat{R} = 1 \Leftrightarrow \hat{\mathrm{Err}}^{(1)} = \hat{\gamma}$ no-info case and $\hat{w} = 1$

Finally, the ".632+"-estimator is

$$\widehat{\mathsf{Err}}^{(.632+)} = (1 - \hat{w})\overline{\mathsf{err}} + \hat{w}\widehat{\mathsf{Err}}^{(1)}$$

where $\hat{w} = \frac{0.632}{1-0.368\hat{R}}$.

For details on this approach consult ESL page 252-253.

The 0.632+ bootstrap estimator is a method for estimating the performance of a machine learning model on unseen data, based on its performance on a sample of data. It is an extension of the 0.632 bootstrap estimator, which was proposed by Efron in 1983.

The 0.632 bootstrap estimator is based on the idea that the performance of a model on the training set is likely to be better than its performance on unseen data. To estimate the performance on unseen data, the 0.632 bootstrap estimator takes the average of the performance on the training set and the performance on a validation set, with a weight of 0.632 given to the performance on the training set.

The 0.632+ bootstrap estimator extends this idea by taking into account the variability of the performance on the validation set. Instead of taking a fixed weight of 0.632 for the performance on the training set, the 0.632+ bootstrap estimator estimates the weight dynamically, by taking into account the variability of the performance on the validation set.

The 0.632+ bootstrap estimator is calculated as follows:

- Take a sample of the dataset with replacement, called bootstrap sample.
- Train the model on the bootstrap sample and test it on the remaining data.
- Repeat this process B times.
- Calculate the average performance of the model across all B runs.
- Let the performance of the model on the bootstrap sample be denoted by T and the performance on the remaining data be denoted by V.
- The 0.632+ bootstrap estimator is given by: T + (V - T) * (1 - B^(-1/2))

The 0.632+ bootstrap estimator is a more robust method for estimating the performance of a model on unseen data, as it takes into account the variability of the performance on the validation set. It is particularly useful when the sample size is small or the performance of the model is highly variable.

Regenerate response

openai ChatGPT:
Explain the 0.632+ bootstrap estimator

Can we think of the bootstrap sample as a validation set? maybe?

The OOB data is more of a traning set than a test set in this setting? The sum in the OOB bootstrap estimator is not this

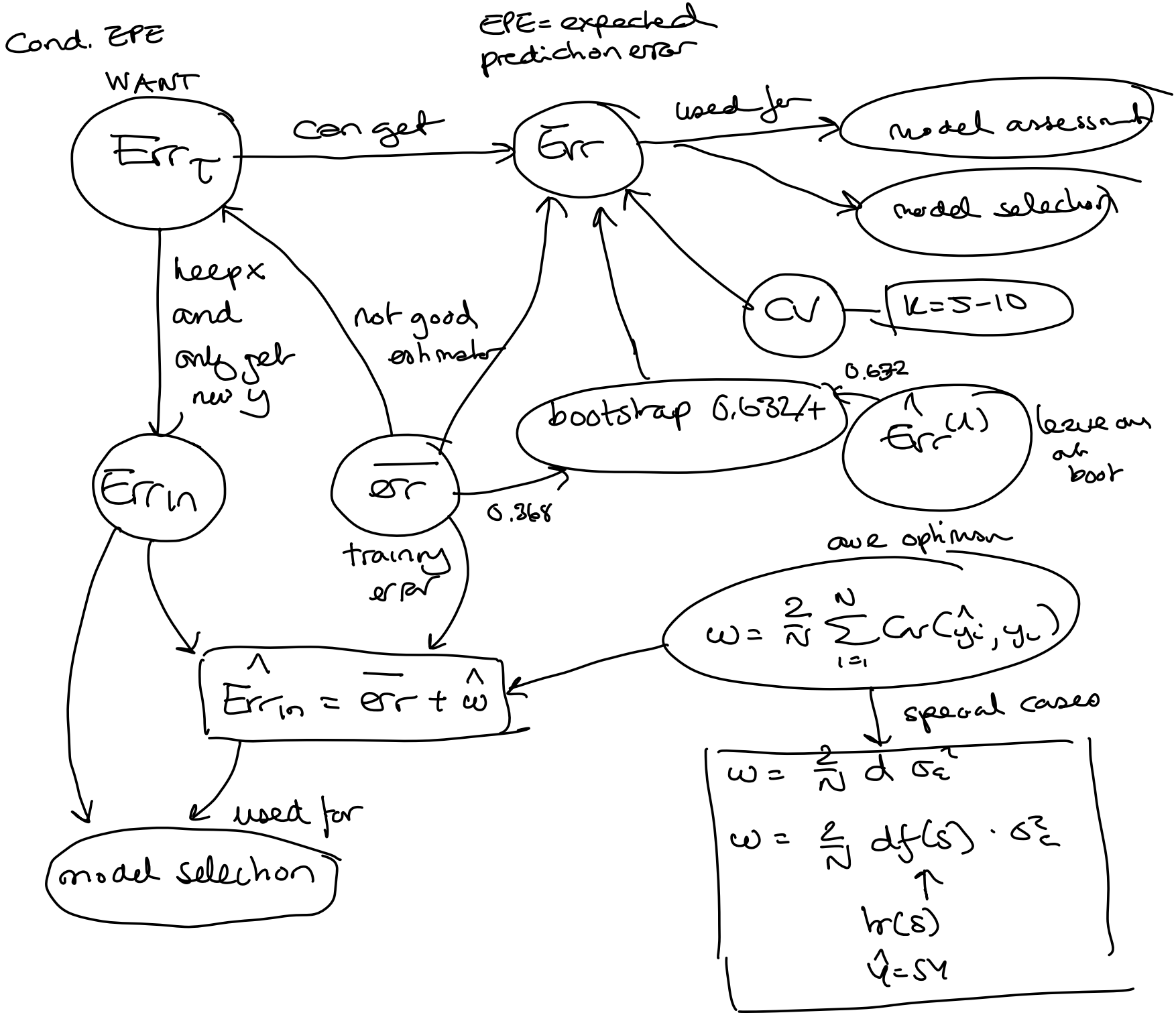
GRADE: E/F ?

# Conclusions

## Group discussion:

Construct a "mind map" or "overview sheet" or "concept map" (mind map with verbs on arrows between entities) for the "Model assessment and selection" topics, and write down important take home messages!

Some concept that could be in the map:
Err,$\text{Err}_T$,$\text{Err}_{\text{in}}$,$\widehat{\text{Err}}_{\text{in}}$,$\overline{\text{err}}$,$\omega$,$\text{Cov}(\hat{y}_i, y_i)$,trace$(S)$,df,CV$(\hat{f})$, bootstrap, $\widehat{\text{Err}}^{(1)}$, $0.632$, $0.368$, model assessment, model selection.


Cond. EPE
WANT
EPE= expected prediction error
$Err_T$
can get
$\overline{Err}$
used for
model assessment
model selection
keep x and only get new y
not good estimate
CV
K=5-10
0.632
bootstrap 0.632+
$\overline{Err}^{(1)}$
leave one out boot
$Err_{in}$
$\overline{err}$
0.368
training error
our optimism
$\omega = \frac{2}{N}\sum_{i=1}^{N} Cov(\hat{y}_i, y_i)$
$\hat{Err}_{in} = \overline{err} + \hat{\omega}$
special cases
$\omega = \frac{2}{N} d\, \sigma_\varepsilon^2$
$\omega = \frac{2}{N} df(S) \cdot \sigma_\varepsilon^2$
$h r(S)$
$\hat{y} = Sy$
used for
model selection

## Final remarks

- In a perfect world we would be rich on data and can divide available data into sets for training, validation and testing
- We have derived cool covariance-result on expected optimism for training error related to in-sample prediction error (the covariance) - that is used for finding model selection criteria (but not for model assessment). If we can´t calculate a formula for the covariance, bootstrapping can be used to do this (Efron and Hastie (2016) Equation 12.64 on åage 224).
- Estimating expected prediction (test) error for a particular training set is not easy in general (if we only have this one training set), but cross-validation and bootstrapping may provide reasonable estimates of the expected test error Err.
- If resampling needed for model assessment: take average of many 10-fold CV Err estimates?

**Week 3: How to handle missing data (new this year)**

Reading list: online-book of van Buuren - given chapters
and Handbook of missing data (oria) parts of chapter 12 on multiple imputation.

Monday 23.01: Motivation, missing types, different solutions, starting on multiple imputation
Friday 27.01: Mainly multiple imputation

**Friday 27.01 after class: meeting with reference group.**

Usual agenda for such meetings:
We go through the learning outcome and compare with the learning plan/ resources and evaluation - and see if they are in alignment (samstemt undervisning)

Comment on lectures, exercises, text book, plan for the course in general.

For Part 4: should we be full digital the two times that Kjersti presents from Oslo, or hybrid (then difficult for Kjersti to help in break-out rooms). The third time Kjersti is in Trondheim (3 hrs physical).

Data analysis project 1: based on Part 2. Which deadlines should we have?
Article presentation: During Part 3 and also possibly Part 5 (after Easter).