MA8701 Advanced methods in statistical inference and learning

L3-4: Model selection and assessment

Mette Langaas

1/17/23

Cross-validation (CV)

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

The aim is to estimate Err_T, but from simulation analyses (ESL Ch 7.12) it turns out that cross-validation estimates 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.

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 kth fold (the observations from the kth 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".

Formal set-up for model selection

- The indexing function κ is unchanged, and for the fitting function we add a tuning parameter α: $f(x, \alpha)$ such that $\hat{f}^{-k}(x, \alpha)$ is the fitted function using tuning parameter α, with the kth fold removed from the model fitting.
- The expected prediction error is estimated by

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

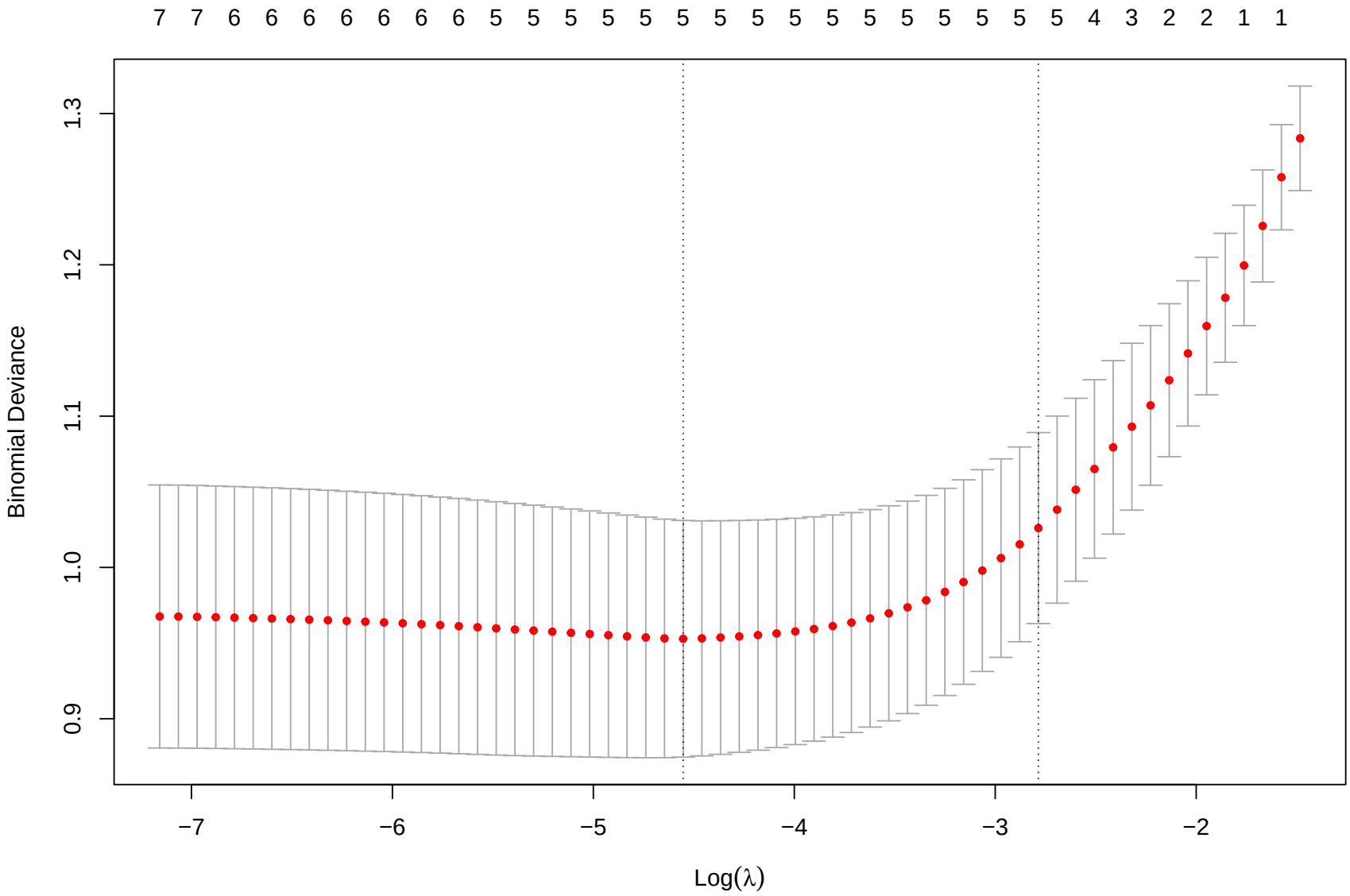
- We find the best tuning parameter $\hat{\alpha}$ that minimize the $\text{CV}(\hat{f}, \alpha)$. Alternatively the *one-standard error rule* can be used: choose the most parsimonious ("smallest") model whose error is no more than one standard error above the error of the best model.
- This best chosen model is then fit to all the data. (ESL page 242).

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.


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?

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.

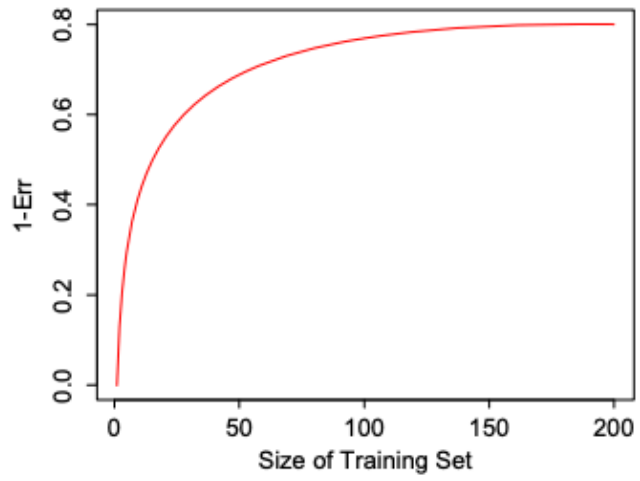


FIGURE 7.8. *Hypothetical learning curve for a classifier on a given task: a plot of* $1-\text{Err}$ *versus the size of the training set* N*. With a dataset of* 200 *observations,* 5*-fold cross-validation would use training sets of size* 160*, which would behave much like the full set. However, with 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 ith 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.

Bootstrap methods

(ESL Ch 7.11 - bootstrapping is known from TMA4268 and TMA4300, but not the special case of estimating Err). Bootstrap in TMA4268: Module 5

Notation: $\mathbf{Z} = (z_1, \ldots, z_N)$ is the training set with $z_i = (x_i, y_i)$.

Aim: Of interest is some quantity calculated from the data $\mathbf{Z}$, denoted $S(\mathbf{Z})$. We will have focus on the expected prediction error.

Resampling: We draw with replacement from $\mathbf{Z}$ a total of N observations into $\mathbf{Z}^{*b}$. We repeat this B times.

Estimator for expected predicted error Err:

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

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?

The problem is given in TMA4268 Module 5 as Problem 1 with (handwritten) solutions.

The answer is $1 - (1 - \frac{1}{N})^N \approx 1 - e^{-1} = 0.632$.

Why is this relevant?

What if we try to change the bootstrap Err estimator - so that we for each observation i only keep predictions from bootstrap samples where this observation is not present? Then we would mimic the CV-estimator.

The *leave-one-out* bootstrap estimate:

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

where C^{-i} are the indices in the bootstrap sample b that do not contain observation i, and $|C^{-i}|$ is the number of samples. (B must be large enough that we do not get any C^{-i}s that are empty, or leave out these zero sets in the formula.)

Comment: this is also called out-of-bootstrap, and is closely connected to the popular out-of-bag estimate for random forests.

There is an addition fix to make the estimate even better.

Since the average number of distinct observations in each bootstrap sample is approximately $0.632N$ - and the bootstrap sample behaves like a training set - this gives a socalled training-set-size bias (similar to C with $K = 2$), meaning that the leave-one-out bootstrap estimator will be *biased upwards*. This can be fixed by weighing together the leave-one-out boostrap estimator with the training error.

The ".632" estimator:

$$\widehat{\mathsf{Err}}^{(.632)} = 0.368\overline{\mathsf{err}} + 0.632\widehat{\mathsf{Err}}^{(1)}$$

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 1NN 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.

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*=γ=the error rate of the prediction rule when predictors and class labels are independent.

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

Further the *relative overfitting rate* is defined to be

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

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.

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,Err_T,Err_{in},$\widehat{\text{Err}}_{\text{in}}$,$\overline{\text{err}}$,$\omega$,$\text{Cov}(\hat{y}_i, y_i)$,trace$(S)$,df,$\text{CV}(\hat{f})$, bootstrap, $\widehat{\text{Err}}^{(1)}$, 0.632, 0.368, model assessment, 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?