Numerical Estimation Techniques

Cross-Validation
Bootstrap
Jackknife
Model selection

October 28, 2019

How complex should a model be?

Estimate the generalization error for ...

- hyper parameters tuning, e.g. estimate λ in ridge regression,
- choosing a kernel for Gaussian processes,
- more general: **model selection**.

Side effect:

- Create variability in models/hypotheses to construct **ensembles**.

Numerical Estimation Techniques

Goal: Select the hypothesis $\hat{f}(x) \in \mathcal{F}$ with minimal prediction error in the hypothesis class $\mathcal{F}$

a) Cross-Validation: Split the data in training, validation and test data; estimation of the prediction error $\mathcal{R}$!

$$\begin{array}{rrcl} \text{regression} & \mathcal{R}(\hat{f}) & = & \mathbb{E}\left[(\hat{f}(x)-y)^2\right] \\ \text{classification} & \mathcal{R}(\hat{c}) & = & \mathbb{E}\left[\mathbb{I}_{\{\hat{c}(x)\neq y\}}\right] = \mathbf{P}\left[\hat{c}(x)\neq y\right] \end{array}$$

b) Bootstrap: Resampling with Replacement of the training data yields different bootstrap sample sets; numerical calculation of the estimation error by the empirical distribution.

c) Jackknife: Method to compensate for systematic estimation errors (bias reduction)

d) Model selection: Neyman Pearson Test, AIC, BIC, ...

Core Problem of Statistical Inference

Dilemma: We want to compute the **expectation** of some statistic, e.g., the expected loss in regression

$$\mathcal{R}(f) := \mathbb{E}_{x,y}\left[\big(y - f(x)\big)^2\right] = \int_{\Omega}\int_{\mathbb{R}} \big(y - f(x)\big)^2 \, \mathbf{P}(x,y)\, dy\, dx.$$

We only (!) can estimate the **empirical mean** of this statistic

$$\hat{\mathcal{R}}(f) := \frac{1}{n}\sum_{i \le n} \big(y_i - f(x_i)\big)^2.$$

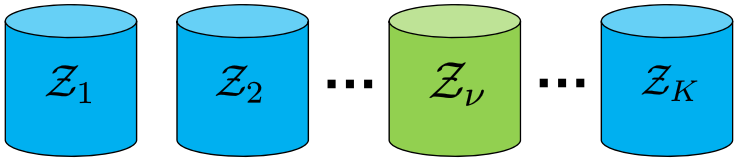
Questions:

1. How far apart are the two values $\mathcal{R}(f),\ \hat{\mathcal{R}}(f)$ for given f?
2. How different are the functions $f^{\text{opt}}(x) \in \arg\min_f \mathcal{R}(f)$, $\hat{f}(x) \in \arg\min_f \hat{\mathcal{R}}(f)$ which minimize the two cost functions $\mathcal{R}(f), \hat{\mathcal{R}}(f)$ over the hypothesis class $\mathcal{F} \ni f$?

Recall: K-fold Cross Validation

1) Initialization

Split data in K approximately equally sized subsets, i.e.,
$\mathcal{Z} = \mathcal{Z}_1 \bigcup \mathcal{Z}_2 \bigcup \cdots \bigcup \mathcal{Z}_\nu \bigcup \cdots \bigcup \mathcal{Z}_K,\ \forall \nu \neq \mu\ :\ \mathcal{Z}_\nu \cap \mathcal{Z}_\mu = \emptyset$;
the map $\kappa : \{1, \ldots, n\} \to \{1, \ldots, K\}$ denotes the subset where datum (x_i, y_i) is an element of.

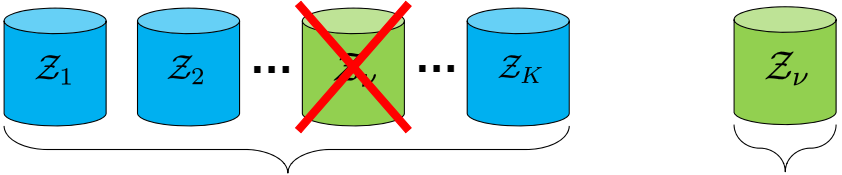


For every partition of a data set in K subsets, we can define K training data sets with approximately $n\frac{K-1}{K}$ data samples.

Frequently, K-fold cross-validation is repeated for several partitionings of the data set. This procedure introduces correlations and might cause a too optimistic estimate of the validation error.

2) ν-th step

Adapt a model $\hat{f}^{-\nu}(x)$ to the $K-1$ data subsets (learning step); validate the resulting model with the not yet used subset $\mathcal{Z}_\nu$



Training data set to learn $\hat{f}^{-\nu}(x)$ Validation data $\Rightarrow \hat{\mathcal{R}}_\nu$

$$\hat{f}^{-\nu} \in \arg\min_{f \in \mathcal{F}} \frac{1}{|\mathcal{Z} \setminus \mathcal{Z}_\nu|} \sum_{i \notin \mathcal{Z}_\nu} (y_i - f(x_i))^2$$

3) Estimation of the prediction error

Combine all K estimates $\hat{f}^{-\nu},\ 1 \le \nu \le K$ of $\hat{\mathcal{R}}$, i.e.,

$$\hat{\mathcal{R}}^{\mathrm{cv}} = \tfrac{1}{n} \textstyle\sum_{i \le n} \big(y_i - \hat{f}^{-\kappa(i)}(x_i)\big)^2$$

Note that $\boxed{\hat{f}^{-\kappa(i)}(x)}$ is the predictor which has been trained by omitting the data $(x_i, y_i) \in \mathcal{Z}_{\kappa(i)}$.

Problem: prediction quality is determined for a model which has been trained on approximately $n(K-1)/K$ data; there exists a systematic tendency to underfit since the adapted model is not as complex as it could have been using the full data set.

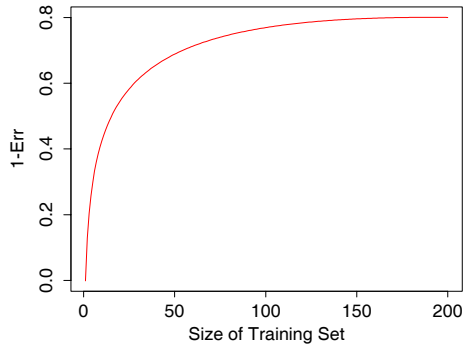
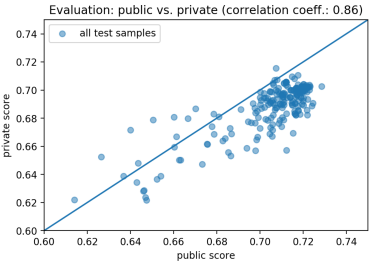


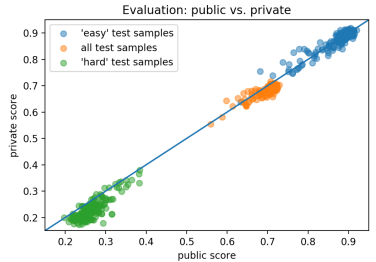
Fig. 7.8 (Hastie et al. 2009)
Hypothetical learning curve for a classifier with 5-fold cross-validation ($n = 200$).

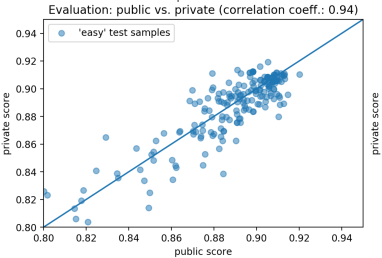
Performance of HS18 student cohort

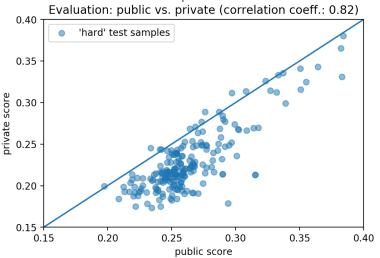
The public score performance of machine learning programs written by AML students in HS18 are compared to the private score. The “hard” test samples with a correlation value of 0.82 document that students overfit to the validation data in the model design cycle by reusing the cross-validation data repeatedly in the model selection process. The ML algorithms perform on “easy” samples for the public score almost as well as for the private score with correlation 0.94.

Stronger regularization of learning would have improved the private score of the students which was used for grading.









"Recall: Leave-one-out" Method

- *Cross-Validation* with $K = n$: Model estimate with $n - 1$ data and n estimates of the prediction error.

$$\hat{f}^{-i} \in \arg\min_{f \in \mathcal{F}} \frac{1}{n-1} \sum_{j \leq n: j \neq i} (y_j - f(x_j))^2$$

Problem: The *leave-one-out* method is unbiased w.r.t. the true prediction error but the variance can be very large due to highly correlated training sets (bias-variance dilemma).

Engineer's solution: Choose K in K-fold cross validation as

$$\min\{\sqrt{n}, 10\}$$

Feature Selection by Cross-validation

Example setting: Linear classification problem in 20 dimensions.

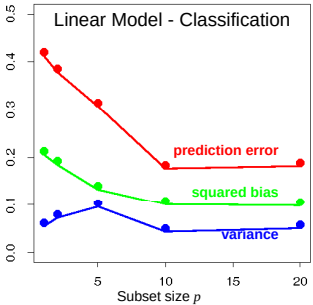
Classification rule to be learned from data: $Y = \begin{cases} 1 & \sum_{j=1}^{10} x_j \geq 5 \\ 0 & \text{otherwise} \end{cases}$

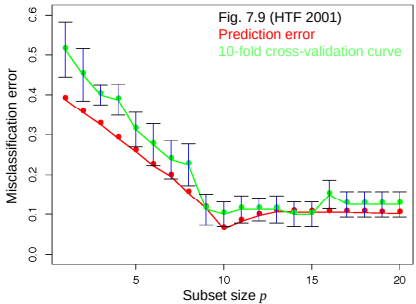
Select relevant features by cross-validation, i.e., adapt model size p by subset selection and choose hypotheses $Y = \begin{cases} 1 & \sum_{j=1}^{p} x_j \geq 5 \\ 0 & \text{otherwise} \end{cases}$; $1 \leq p \leq 20$ and control it by minimizing the prediction error estimated via cross-validation.

Feature selection is a special form of model selection where we prefer one model over alternative models due to its minimal validation error. The validation error is determined based on the out-of-sample data $\mathcal{Z}_{\kappa(i)}$ for classifier $\hat{c}^{-\kappa(i)}(x)$.

Note that it is important to calculate the final prediction error on data which have not been used in model fitting (training) nor in model selection (testing for parameter adaptation).

Experimental Result for Feature Selection





Left figure: Classification with 0-1 loss of the rule $Y = \begin{cases} 1 & \sum_{j=1}^{p} x_j \geq 5 \\ 0 & \text{otherwise} \end{cases}$ by best subset selection of size p.

Right figure: The 10-fold cross-validation error overestimates the "true" prediction error since the fitted models are trained on only 90 percent of the data. Both curves indicate that 10 features are sufficient and additional features will not reduce the prediction error.

“Bootstrapping”

(B. Efron, “Bootstrap methods: another look at the jackknife” (1979)

The "Bootstrap" Method

Idea of Bootstrap

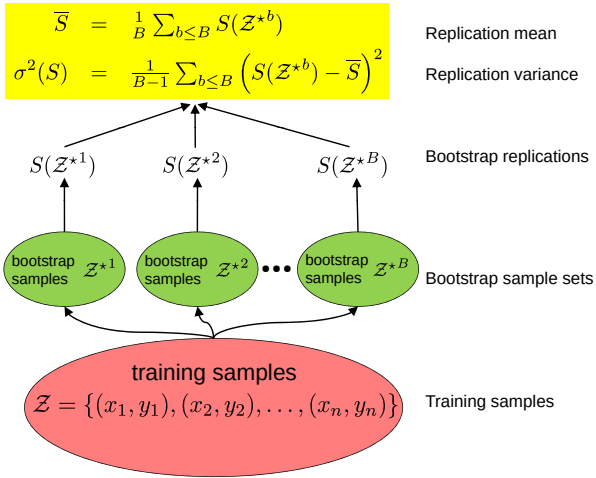
Estimate the uncertainty of a statistic $S(\mathcal{Z})$ by resampling with replacement. The Bootstrap method is intended to derive properties of the distribution of $S(\mathcal{Z})$, i.e., its mean $\mathbb{E}_F[S(\mathcal{Z})]$ and its variance $\mathbb{V}_F[S(\mathcal{Z})]$ where $F(Z)$ is the distribution function.

Cross-validation "sacrifices" samples for validation which yields a bias towards too simple models. Bootstrap replaces the true data source by the empirical data source $\mathcal{Z}$ where we sample B bootstrap sample sets $(\mathcal{Z}^{\star b})|_{b=1}^{B}$. For continuous feature vectors X, $F_X(x) = \mathbf{P}\{X \leq x\}$ is the cumulative distribution function (CDF) and $\hat{F}_X(x) = \#\{x_i \in \mathcal{X} : x_i \leq x\}$ is the empirical CDF.

Goal of Bootstrap

Calculate numerically the estimation error of a statistic $S(\mathcal{Z})$, e.g., the regression risk $\hat{\mathcal{R}}(\hat{f})$ for regression function $\hat{f}(x)$ based on the empirical distribution function $\hat{F}_X(x)$.

The “Bootstrap” method graphically



Bootstrap samples: $\mathcal{Z}^\star = \{Z_1^\star, \ldots, Z_n^\star\}$, with $Z_i^\star$ being independently drawn from $\hat{F}$. Note that $Z_i^\star = Z_j^\star$ for $i \neq j$ is possible since we use sampling with replacement.

Approximation by Bootstrap: (i) data source F_X is approximated by empirical data source $\hat{F}_X$;
(ii) variance is approximated by bootstrap average.

$$\mathbb{V}_F[S(\mathcal{Z})] \approx \mathbb{V}_{\hat{F}}[S(\mathcal{Z})] \approx \mathbb{V}_{\text{boot}}$$

Consistency of the bootstrap estimate: $\mathbb{V}_{\text{boot}} \equiv$

$$\lim_{B\to\infty} \frac{1}{B-1} \sum_{b\leq B} \left(S(\mathcal{Z}^{\star b}) - \frac{1}{B} \sum_{\beta \leq B} S(\mathcal{Z}^{\star \beta}) \right)^2 = \mathbb{V}_{\hat{F}}[S(\mathcal{Z})]$$

Remark: Bootstrap estimates often show a too small bias! $B = 200$ is sufficient in most cases to estimate the error.

When Does "Bootstrap" Work?

Let F be an unknown distribution function, $\hat{F}$ be an empirical distribution function and $\mathcal{R}_n^{\text{str}}$ a statistical functional for sample set size n ($\text{str} \in \{\text{err}, \text{bias}, \text{std}\}$).

Error distribution: $\mathcal{R}_n^{\text{err}}(F, \hat{F}) = \mathbf{P}\left(\sqrt{n}\big(S(\hat{F}) - S(F)\big)\right)$

Bias: $\mathcal{R}_n^{\text{bias}}(F, \hat{F}) = \mathbb{E}_F[S(\hat{F})] - S(F)$

Standard error: $\mathcal{R}_n^{\text{std}}(F, \hat{F}) = \sqrt{\mathbb{E}_F\left[(S(\hat{F}) - S(F))^2\right]}$

Bootstrap works ...

... if the deviation between empirical and bootstrap estimator ($\hat{F}^\star$ being the Bootstrap CDF) converges in probability to the deviation between true parameter value and the empirical estimator, i.e.,

$$\mathcal{R}_n^{\text{str}}(\hat{F}, \hat{F}^\star) - \mathcal{R}_n^{\text{str}}(F, \hat{F}) \xrightarrow{\mathbf{P}} 0 \quad \text{for} \quad n \to \infty,\ \text{str} \in \{\text{err}, \text{bias}, \text{std}\}.$$

Improvement of the "Bootstrap" Method

Bootstrap estimation of classification error is **too optimistic**!

Naive Bootstrap error estimation: use the bootstrap estimator and evaluate it on original data $\mathcal{Z}$.

$$\hat{\mathcal{R}}^{\star} = \frac{1}{B}\frac{1}{n}\sum_{b=1}^{B}\sum_{i=1}^{n} l(y_i, \hat{f}^{\star b}(x_i))$$

Problem: Overlap of training and test set.

Question: What is the probability that a sample Z_i appears at least once in a specific bootstrap sample set $\mathcal{Z}^{\star b}, 1 \leq b \leq B$ of size n?

$$\begin{aligned} \forall b \quad \mathbf{P}\{Z_i \in \mathcal{Z}^{\star b}\} &= 1 - \left(1 - \frac{1}{n}\right)^n \\ &\overset{n\to\infty}{\longrightarrow} 1 - \frac{1}{e} = 0.632. \end{aligned}$$

Test: Example of 1-nn

Classification problem: Labels $Y \in \{0, 1\}$ and $p(Y = 1) = 0.5$. $P(X, Y) = P(X)P(Y)$, i.e., the features are statistically independent of the class labels.

Loss: Let l be the $0/1$-loss, i.e. $l(y, c(x)) = \mathbb{I}_{\{c(x) \neq y\}}$.

Classifier: $c(x)$ trained by 1-nearest neighbor method.

Error: True error: 0.5. The naive Bootstrap error estimates 0 on the training set and 0.5 error rate only on the not selected data, that means on 36.8% of the data; $\Rightarrow$ expected error rate is approximately 18.4% rather than 50%.

Improvement of plain bootstrap

Leave-one-out bootstrap: $\hat{\mathcal{R}}^{(1)} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|\mathcal{C}^{-i}|} \sum_{b \in C^{-i}} l(y_i, f^{\star b}(x_i))$

$C^{-i} \subset \{1, \cdots, B\}$ contains all the bootstrap indices b so that $\mathcal{Z}^{\star b}$ does not contain observation (x_i, y_i).

Leave-one-out bootstrap solves the naive bootstrap overfitting problem. However, as in CV, less data is used for training: on average, $0.632 \times n$ samples. The introduced bias is similar as for 2-fold CV.

.632 bootstrap: $\hat{\mathcal{R}}^{(.632)} = 0.368 \cdot \hat{\mathcal{R}}_{\text{train}} + 0.632 \cdot \hat{\mathcal{R}}^{(1)}$

This reduces the bias of Leave-one-out bootstrap.Yet, in our $1nn$ example, we have $\hat{\mathcal{R}}_{\text{train}} = 0$ and $\hat{\mathcal{R}}^{(1)} = 0.5$. Thus, the error is estimated as $0.632 \times 0.5 < 0.5$.

.632+ bootstrap: $\hat{\mathcal{R}}^{(.632+)} = (1 - \hat{w}) \cdot \hat{\mathcal{R}}_{\text{train}} + \hat{w} \cdot \hat{\mathcal{R}}^{(1)}$

with $\hat{w} = \frac{0.632}{1 - 0.368\hat{G}}$, $\hat{G} = \frac{\hat{\mathcal{R}}^{(1)} - \hat{\mathcal{R}}^{\star}}{\hat{\gamma} - \hat{\mathcal{R}}^{\star}}$, $\hat{\gamma} = \frac{1}{n^2}\sum_{i=1}^{N}\sum_{j=1}^{N} l(y_i, \hat{f}(x_j))$.

More information can be found in "The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, Jerome Friedman.

Sketching distributions

How to encode uncertainty?

Functions $S(\mathcal{Z})$ that depend on stochastic data $\mathcal{Z}$ are random variables!
They are characterized by their distributions $\mathbf{P}(S(\mathcal{Z}))$. Therefore, we have to develop algorithmic techniques to represent these distributions.

Moment methods: Cross-validation and bootstrap provide numerical techniques to estimate moments of some statistic $S(\mathcal{Z})$.

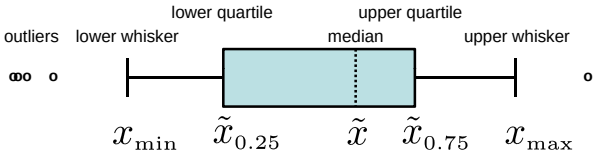
Graphical sketch: Box-Plots sketch a distribution by quartiles and encode more information on the distribution than mean and variance of $S(\mathcal{Z})$.

Density estimation: The most information of $S(\mathcal{Z})$ is captured by its probability distribution.

Graphical Statistics: Box-Plots

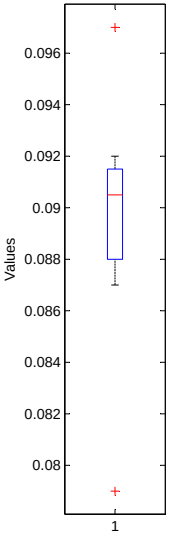
Box-plots ...

... provide a simplified representation of an empirical distribution with outliers (○), lower and upper quartiles and median.



Example: (Matlab)

```
>> x = [0.09, 0.087, 0.092, 0.079, 0.091,
        0.097, 0.089, 0.091]
>> boxplot(x)
```



The "Jackknife" Method (Quenouille 1949, Tukey 1958)

Goal: **Numerical estimate of the bias of an estimator**

Assume that you want to use an estimator $\hat{S}_n(x_1, \ldots, x_n)$ with a considerable bias. How can we reduce this bias?
The Jackknife method uses the leave-one-out estimator $\hat{S}_{n-1}^{(-i)}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ to estimate the bias of $\hat{S}_n$.

Expansion of the bias of $\hat{S}_n$ (continuity of the estimator is assumed):

$$\mathbb{E}[\hat{S}_n] - S = \frac{a_1}{n} + \frac{a_2}{n^2} + \ldots$$

Bias of leave-one-out estimator:

$$\mathbb{E}[\hat{S}_{n-1}^{(-i)}] - S = \frac{a_1}{n-1} + \frac{a_2}{(n-1)^2} + \ldots$$

Jackknife bias estimator of $\hat{S}_n$

$$\text{bias}^{\text{JK}} := (n-1)(\tilde{S}_n - \hat{S}_n) \quad \text{with} \quad \tilde{S}_n = \frac{1}{n}\sum_{i=1}^{n} \hat{S}_{n-1}^{(-i)}$$

Asymptotic expansion of the expected Jackknife bias

$$\begin{aligned}
\mathbb{E}[\text{bias}^{\text{JK}}] &= (n-1)(\mathbb{E}[\tilde{S}_n] - \mathbb{E}[\hat{S}_n]) \\
&= (n-1)\left(\tfrac{1}{n}\textstyle\sum_i (\mathbb{E}[\hat{S}_{n-1}^{(-i)}] - S) - \mathbb{E}[\hat{S}_n] + S\right) \\
&= (n-1)\left(\frac{a_1}{n-1} + \frac{a_2}{(n-1)^2} - \frac{a_1}{n} - \frac{a_2}{n^2} + \mathcal{O}(n^{-3})\right) \\
&= \frac{a_1}{n} + \frac{(2n-1)a_2}{(n-1)n^2} + \mathcal{O}(n^{-2}) \;=\; \frac{a_1}{n} + \mathcal{O}(n^{-2})
\end{aligned}$$

This equation shows that the expectation $\mathbb{E}[\text{bias}^{\text{JK}}]$ estimates the bias up to order n^{-2}. Therefore we can define a Jackknife estimator by subtracting bias^{JK} from $\hat{S}_n$.

Jackknife estimator: $\hat{S}^{\text{JK}} = \hat{S}_n - \text{bias}^{\text{JK}}$

“Jackknife” Debiasing

Shao & Tu, “*The Jackknife and Bootstrap*, Springer Verlag, (1995)

Goal: Construction of an estimator with small bias (*debiasing*), i.e.,

$$\overline{S} = \hat{S} - \hat{\text{bias}}$$

For bootstrap bias $\hat{\text{bias}}_{\text{boot}} = \frac{1}{B}\sum_{b \leq B} \hat{S}^{\star}(b) - \hat{S}$ it holds

$$\overline{S} = 2\hat{S} - \frac{1}{B}\sum_{b \leq B} \hat{S}^{\star}(b)$$

Example: ML plug-in classifiers: $n\hat{\mathcal{R}} - \frac{n-1}{n}\sum_i \hat{\mathcal{R}}^{-i}$

Warning: Bias corrected estimators can have a considerably larger variance than uncorrected estimators!

Model selection

Neyman-Pearson Test
Complexity based model selection:
Minimum Description Length
Bayesian Information Criterion
Akaikes Information Criterion
Takeuchi Information Criterion

(partially adapted from Charles J. Geyer (2016), Lecture Slides on Model Selection)

October 28, 2019

Neyman-Pearson Test

Select a decision rule which minimizes the error of the second kind while fixing the error of the first kind!

Problem setting: Let the random variables

$$X_1, X_2, \ldots, X_n \text{ i.i.d} \sim \mathbf{Q}(x).$$

Consider the hypotheses $\mathcal{H}_0 : \mathbf{Q} = \mathbf{P}_0$ and $\mathcal{H}_1 : \mathbf{Q} = \mathbf{P}_1$.

Consider the general decision function $g(x_1, \ldots, x_n)$, where

$$g(x_1, \ldots, x_n) = \begin{cases} 0 & \mathcal{H}_0 \text{ is accepted,} \\ 1 & \mathcal{H}_1 \text{ is accepted.} \end{cases}$$

Error probabilities: $(A = \{\mathbf{x} : g(x_1, \ldots, x_n) = 0\})$

$$\begin{aligned} \text{Type I error (false positive):} \quad \alpha &= \mathbf{P}\{g(X_1, \ldots, X_n) = 1 | \mathcal{H}_0 \text{ true}\} = \mathbf{P}_0^n(A^c) \\ \text{Type II error (false negative):} \quad \beta &= \mathbf{P}\{g(X_1, \ldots, X_n) = 0 | \mathcal{H}_1 \text{ true}\} = \mathbf{P}_1^n(A) \end{aligned}$$

Theorem: (Neyman-Pearson Lemma)
Let the random variables $X_1, X_2, \ldots, X_n$ be i.i.d. according to $\sim \mathbf{Q}(x)$. Consider the decision problem to accept the hypotheses $\mathbf{Q} = \mathbf{P}_0$ vs. $\mathbf{Q} = \mathbf{P}_1$.
For $T \geq 0$, define the region

$$A_n(T) = \left\{ \frac{\mathbf{P}_0(x_1, \ldots, x_n)}{\mathbf{P}_1(x_1, \ldots, x_n)} > T \right\}.$$

Let $\alpha^* = \mathbf{P}_0^n(A_n^c(T)),\ \beta^* = \mathbf{P}_1^n(A_n(T))$ be the respective error probabilities for the decision region A_n. Let B_n be an arbitrary other decision region with error probabilities $\alpha,\ \beta$.

Then it holds $\beta \geq \beta^*$, if $\alpha \leq \alpha^*$.

Remark: The Neyman-Pearson lemma characterizes the likelihood ratio test as optimal test!

Proof: Let $A = A_n(T)$ be the acceptance region according to the *likelihood ratio test* and assume that $B \in \mathcal{X}^n$ is an arbitrary second acceptance region. Φ_A and Φ_B are the indicator functions for these regions. Then for all $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$ the following inequality holds:

$$\underbrace{(\Phi_A(\mathbf{x}) - \Phi_B(\mathbf{x}))}_{\{0,1\},\text{ or } \{-1,0\}} \underbrace{(\mathbf{P}_0\{\mathbf{x}\} - T\mathbf{P}_1\{\mathbf{x}\})}_{\geq 0;\text{ or } \leq 0} \geq 0 \,.$$

Multiplication and integration over the data space yields

$$\begin{aligned} 0 &\leq \sum_{\mathbf{x}} (\Phi_A \mathbf{P}_0 - T\Phi_A \mathbf{P}_1 - \mathbf{P}_0 \Phi_B + T\mathbf{P}_1 \Phi_B) \\ &= \sum_{A} (\mathbf{P}_0 - T\mathbf{P}_1) - \sum_{B} (\mathbf{P}_0 - T\mathbf{P}_1) \\ &= (1 - \alpha^*) - T\beta^* - (1 - \alpha) + T\beta \\ &= T(\beta - \beta^*) - (\alpha^* - \alpha). \end{aligned}$$

Since $T \geq 0$ has been assumed, the inequality implies $\beta \geq \beta^*$ for $\alpha \leq \alpha^*$.

□

Complexity-Based Model Selection

Assume that we have (nested) models with k parameters $p(\mathbf{X}|\theta_k)$.

Problem: more parameters lead to better fit!

Strategy: add a complexity term to the goodness of fit term.

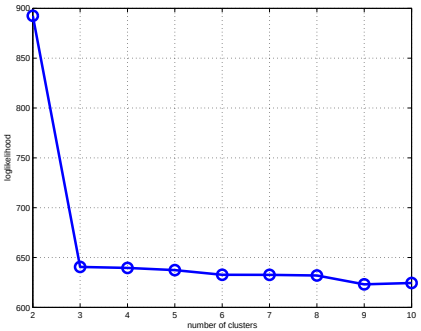
Occam's razor: Choose the simplest model that explains the data.

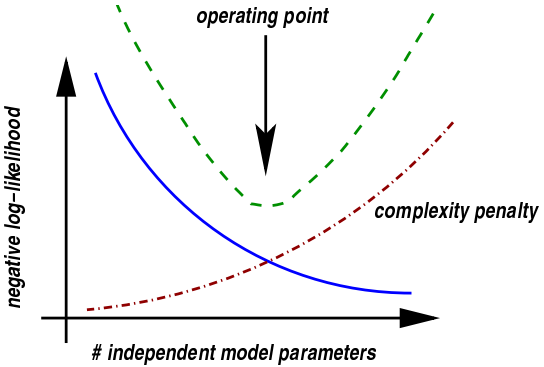
Minimize:

$$\text{objective} = \text{loss} + \text{model complexity}$$

Note: Cross-validation (CV) and Bootstrap do not assume explicit knowledge of model complexity, but are often used to tune regularization parameters (i.e. to adjust the model complexity term). Goal in all cases: Choose model complexity penalty so that optimal model also generalizes best.
Next: Assume that we know the log likelihood function (loss) of the models in our model class. We aim at deriving criteria that are computationally less intensive than CV or Bootstrap and make use of the full dataset available (in contrast to CV). They further correct for overfitting of Maximum Likelihood estimation.

Underlying Principle





Negative log-likelihood

... usually decreases with increasing model complexity. More parameters support a better fit!

Correct this tendency to select complex models with a complexity penalty.

Bayesian Perspective

Assume: Candidate model classes $\mathcal{M}_1 \ldots \mathcal{M}_m$

Bayes Factor to select between two classes $\mathcal{M}_k$ and $\mathcal{M}_l$

$$\frac{p(\mathbf{X}|\mathcal{M}_k)}{p(\mathbf{X}|\mathcal{M}_l)}$$

where

$$p(\mathbf{X}|\mathcal{M}_k) = \int p(\mathbf{X}|\theta_k, \mathcal{M}_k) p(\theta_k|\mathcal{M}_k)\, d\theta_k$$

Using Laplace Approximation of $p(\mathbf{X}|\mathcal{M}_k)$ (cf. Ripley, p. 64)

$$\log p(\mathbf{X}|\mathcal{M}_k) = \log p(\mathbf{X}|\hat{\theta}_k, \mathcal{M}_k) - \frac{k'}{2} \log n + O(1)$$

Bayesian Information Criterion (BIC) (Schwartz ,1978)

BIC drops $O(1)$ terms and hence:

$$\begin{aligned}\log\left(\Pr(\mathcal{M}_k \mid \mathbf{X})\right) &\propto \log\left(\Pr(\mathcal{M}_k)\right) + \log\left(\Pr(\mathbf{X}|\mathcal{M}_k)\right) \\ &\approx \text{const} + \log(\hat{\mathrm{p}}(\mathbf{X} \mid \hat{\theta}_k, \mathcal{M}_k)) - \frac{k'}{2}\log n\end{aligned}$$

$$\text{BIC} := -2\log(\hat{\mathrm{p}}(\mathbf{X} \mid \hat{\theta}_k, \mathcal{M}_k)) + k' \log n$$

Information theoretic view: Minimum Description Length

MDL minimizes the overall description length of the data

$$\underbrace{-\log p(\mathbf{X}|\theta_k)}_{\text{data}} \underbrace{-\log p(\theta_k)}_{\text{model}}$$

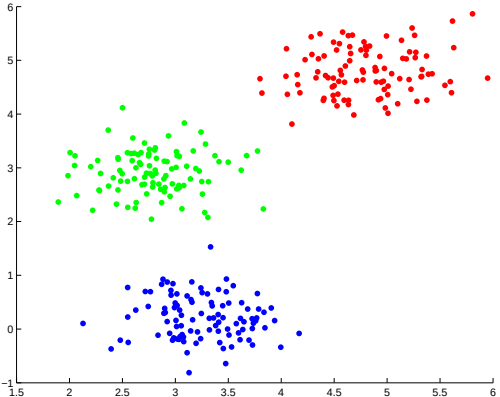
Asymptotic approximation (one possible)

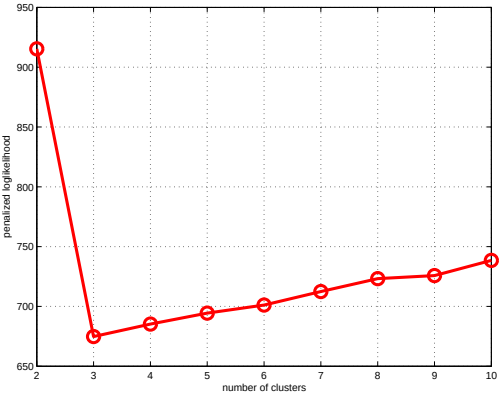
$$\hat{k} \in \underset{1 \le k \le K_{\max}}{\operatorname{argmin}} \left(\underbrace{-\log(\mathrm{p}(\mathbf{X}|\hat{\theta}_k))}_{\text{negative loglikelihood}} + \underbrace{\frac{1}{2} k' \log n}_{\text{complexity penalty}} \right)$$

where k' is the number of independent parameters in the model encoded by (maximum likelihood estimate) $\hat{\theta}_k$.

MDL and **BIC** are consistent as a model and asymptotically equivalent.

A Toy Example: K-means Clustering





MDL/BIC estimates the correct number of classes for this data set.

Akaike Information Criterion (AIC)

Akaike 1973: $AIC = -2\log(\hat{\mathrm{p}}(\mathbf{X} \mid \hat{\theta}_k)) + 2k$

- approximates the Kullback-Leibler (KL) divergence:

$$D(p \mid\mid \hat{p}) = -\int p(x) \log\left(\frac{\hat{p}(x \mid \hat{\theta}_k)}{p(x)}\right) dx$$

  between the true model $p(x)$ and the estimate $\hat{p}(x \mid \hat{\theta}_k)$.

Note: AIC is asymptotically equivalent to Leave-one-out cross-validation for ordinary linear regression models (and mixed effect models). It can also be derived in a Bayesian framework, but with a different prior than for BIC.
The penalty on model complexity is smaller for AIC, as it misses the factor $\log(n)$. For small sample sizes, AIC has the tendency to select large models, i.e. to overfit, while BIC has the tendency to underfit.

Maximum Likelihood Revisited

Recall, when the model is correct, maximum likelihood is consistent, asymptotically normal, efficient and

$$\hat{\theta}_n \xrightarrow{P} \theta_0,$$

$$\sqrt{n}(\theta_n - \theta_0) \xrightarrow{D} \mathcal{N}(0, I^{-1}(\theta_0)),$$

where $I(\theta_0)$ denotes the Fisher Information. When the model is not correct, maximum likelihood is not consistent, since there is no θ that corresponds to the true distribution of the data. In this case

$$\hat{\theta}_n \xrightarrow{P} \theta^*$$

where θ^* minimizes the KL divergence with respect to the true distribution of the data.

Correction of Maximum Likelihood

When the considered model class does not contain the true model, the log likelihood derivative identities no longer hold. Fisher information can no longer be defined two ways. Instead, we have

$$I_n(\theta) = \mathbb{V}\left(\nabla l_n(\theta)\right)$$
$$J_n(\theta) = -\mathbb{E}\left(\nabla^2 l_n(\theta)\right)$$

with the negative log-likelihood $l_n(\theta) = \log(\hat{\mathrm{p}}(\mathbf{X_1}, \ldots, \mathbf{X_n} \mid \hat{\theta}))$.
Maximum likelihood leads to:

$$\sqrt{n}(\theta_n - \theta^*) \xrightarrow{D} \mathcal{N}(0, J_1^{-1}(\theta^*) I_1(\theta^*) J_1^{-1}(\theta^*)).$$

Takeuchi Information Criterion (TIC)

Consistency: BIC is a consistent, AIC minimizes the KL divergence to the true model.

TIC: correction of AIC when true model is not an element of the model class. TIC reduces to AIC if the true model is an element of the model class.

$$TIC = -2\log(\hat{\mathrm{p}}(\mathbf{X} \mid \hat{\theta}_k)) + 2\mathrm{trace}[I_1(\theta_k)J_1^{-1}(\theta_k)]$$

Summary

- Well-motivated model selection schemes.
- BIC/MDL, AIC for model selection rely on likelihood optimization.
- Hence, validation is not generally applicable.
- Many more information criteria have been developed: corrections for small sample sizes, model specific ones, etc. The best choice depends on the assumed and true model class.